The Mesa 3D Graphics Library Documentation

Release latest

Brian Paul

Oct 21, 2020

Documentation

Both professional and volunteer developers contribute to Mesa.

VMware employs several of the main Mesa developers including Brian Paul and Keith Whitwell.

In the past, Tungsten Graphics contracts implemented many Mesa features including:

- DRI drivers for Intel i965, i945, i915 and other chips
- Advanced memory manager and framebuffer object support
- Shading language compiler and OpenGL 2.0 support
- MiniGLX environment

Other companies including Intel and RedHat also actively contribute to the project. Intel has recently contributed the new GLSL compiler in Mesa 7.9.

LunarG can be contacted for custom Mesa / 3D graphics development.

Volunteers have made significant contributions to all parts of Mesa, including complete device drivers.

CHAPTER 1

Platforms and Drivers

Mesa is primarily developed and used on Linux systems. But there's also support for Windows, other flavors of Unix and other systems such as Haiku. We're actively developing and maintaining several hardware and software drivers.

The primary API is OpenGL but there's also support for OpenGL ES 1, ES2 and ES 3, OpenCL, VDPAU, XvMC and the EGL interface.

Hardware drivers include:

- Intel GMA, HD Graphics, Iris. See Intel's Website
- AMD Radeon series. See RadeonFeature
- NVIDIA GPUs (Riva TNT and later). See Nouveau Wiki
- Qualcomm Adreno A2xx-A6xx. See Freedreno Wiki
- Broadcom VideoCore 4, 5. See This Week in V3D
- ARM Mali Utgard. See Lima Wiki
- ARM Mali Midgard, Bifrost. See Panfrost Site
- Vivante GCxxx. See Etnaviv Wiki
- NVIDIA Tegra (K1 and later).

Software drivers include:

- *llvmpipe* - uses LLVM for x86 JIT code generation and is multi-threaded
- softpipe - a reference Gallium driver
- *svga* - driver for VMWare virtual GPU
- swr - x86-optimized software renderer for visualization workloads
- virgl - research project for accelerated graphics for qemu guests
- swrast - the legacy/original Mesa software rasterizer

Additional driver information:

- DRI hardware drivers for the X Window System
- *Xlib / swrast driver* for the X Window System and Unix-like operating systems
- Microsoft Windows

1.1 Deprecated Systems and Drivers

In the past there were other drivers for older GPUs and operating systems. These have been removed from the Mesa source tree and distribution. If anyone's interested though, the code can be found in the Git repo. The list includes:

- 3dfx/glide
- Matrox
- ATI R128
- Savage
- VIA Unichrome
- SIS
- 3Dlabs gamma
- DOS
- fbdev
- DEC/VMS
- Mach64
- Intel i810

CHAPTER 2

License and Copyright

2.1 Disclaimer

Mesa is a 3-D graphics library with an API which is very similar to that of OpenGL[1]. To the extent that Mesa utilizes the OpenGL command syntax or state machine, it is being used with authorization from Silicon Graphics, Inc.(SGI). However, the author does not possess an OpenGL license from SGI, and makes no claim that Mesa is in any way a compatible replacement for OpenGL or associated with SGI. Those who want a licensed implementation of OpenGL should contact a licensed vendor.

Please do not refer to the library as *MesaGL* (for legal reasons). It's just *Mesa* or *The Mesa 3-D graphics library*.

2.2 License / Copyright Information

The Mesa distribution consists of several components. Different copyrights and licenses apply to different components. For example, the GLX client code uses the SGI Free Software License B, and some of the Mesa device drivers are copyrighted by their authors. See below for a list of Mesa's main components and the license for each.

The core Mesa library is licensed according to the terms of the MIT license. This allows integration with the XFree86, Xorg and DRI projects.

The default Mesa license is as follows:

```
Copyright (C) 1999-2007  Brian Paul   All Rights Reserved.

Permission is hereby granted, free of charge, to any person obtaining a
copy of this software and associated documentation files (the "Software"),
to deal in the Software without restriction, including without limitation
the rights to use, copy, modify, merge, publish, distribute, sublicense,
and/or sell copies of the Software, and to permit persons to whom the
Software is furnished to do so, subject to the following conditions:
```

(continues on next page)

[1] OpenGL is a trademark of Silicon Graphics Incorporated.

(continued from previous page)

```
The above copyright notice and this permission notice shall be included
in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS
OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT.  IN NO EVENT SHALL
THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
SOFTWARE.
```

2.3 Attention, Contributors

When contributing to the Mesa project you must agree to the licensing terms of the component to which you're contributing. The following section lists the primary components of the Mesa distribution and their respective licenses.

2.4 Mesa Component Licenses

Component	Location	License
Main Mesa code	src/mesa/	MIT
Device drivers	src/mesa/drivers/*	MIT, generally
Gallium code	src/gallium/	MIT
Ext headers	include/GL/glext.h, include/GL/glxext.h	Khronos
GLX client code	src/glx/	SGI Free Software License B
C11 thread emulation	include/c11/threads*.h	Boost (permissive)

In general, consult the source files for license terms.

CHAPTER 3

Frequently Asked Questions

Last updated: 19 September 2018

3.1 1. High-level Questions and Answers

3.1.1 1.1 What is Mesa?

Mesa is an open-source implementation of the OpenGL specification. OpenGL is a programming library for writing interactive 3D applications. See the OpenGL website for more information.

Mesa 9.x supports the OpenGL 3.1 specification.

3.2 1.2 Does Mesa support/use graphics hardware?

Yes. Specifically, Mesa serves as the OpenGL core for the open-source DRI drivers for X.org.

- See the DRI website for more information.
- See 01.org for more information about Intel drivers.
- See nouveau.freedesktop.org for more information about Nouveau drivers.
- See www.x.org/wiki/RadeonFeature for more information about Radeon drivers.

3.3 1.3 What purpose does Mesa serve today?

Hardware-accelerated OpenGL implementations are available for most popular operating systems today. Still, Mesa serves at least these purposes:

- Mesa is used as the core of the open-source X.org DRI hardware drivers.

- Mesa is quite portable and allows OpenGL to be used on systems that have no other OpenGL solution.
- Software rendering with Mesa serves as a reference for validating the hardware drivers.
- A software implementation of OpenGL is useful for experimentation, such as testing new rendering techniques.
- Mesa can render images with deep color channels: 16-bit integer and 32-bit floating point color channels are supported. This capability is only now appearing in hardware.
- Mesa's internal limits (max lights, clip planes, texture size, etc) can be changed for special needs (hardware limits are hard to overcome).

3.4 1.4 What's the difference between "Stand-Alone" Mesa and the DRI drivers?

Stand-alone Mesa is the original incarnation of Mesa. On systems running the X Window System it does all its rendering through the Xlib API:

- The GLX API is supported, but it's really just an emulation of the real thing.
- The GLX wire protocol is not supported and there's no OpenGL extension loaded by the X server.
- There is no hardware acceleration.
- The OpenGL library, `libGL.so`, contains everything (the programming API, the GLX functions and all the rendering code).

Alternately, Mesa acts as the core for a number of OpenGL hardware drivers within the DRI (Direct Rendering Infrastructure):

- The `libGL.so` library provides the GL and GLX API functions, a GLX protocol encoder, and a device driver loader.
- The device driver modules (such as `r200_dri.so`) contain a built-in copy of the core Mesa code.
- The X server loads the GLX module. The GLX module decodes incoming GLX protocol and dispatches the commands to a rendering module. For the DRI, this module is basically a software Mesa renderer.

3.5 1.5 How do I upgrade my DRI installation to use a new Mesa release?

This wasn't easy in the past. Now, the DRI drivers are included in the Mesa tree and can be compiled separately from the X server. Just follow the Mesa *compilation instructions*.

3.6 1.6 Are there other open-source implementations of OpenGL?

Yes, SGI's OpenGL Sample Implementation (SI) is available. The SI was written during the time that OpenGL was originally designed. Unfortunately, development of the SI has stagnated. Mesa is much more up to date with modern features and extensions.

Vincent is an open-source implementation of OpenGL ES for mobile devices.

miniGL is a subset of OpenGL for PalmOS devices. The website is gone, but the source code can still be found on sourceforge.net.

TinyGL is a subset of OpenGL.

SoftGL is an OpenGL subset for mobile devices.

Chromium isn't a conventional OpenGL implementation (it's layered upon OpenGL), but it does export the OpenGL API. It allows tiled rendering, sort-last rendering, etc.

ClosedGL is an OpenGL subset library for TI graphing calculators.

There may be other open OpenGL implementations, but Mesa is the most popular and feature-complete.

3.7 2. Compilation and Installation Problems

3.7.1 2.1 What's the easiest way to install Mesa?

If you're using a Linux-based system, your distro CD most likely already has Mesa packages (like RPM or DEB) which you can easily install.

3.7.2 2.2 I get undefined symbols such as bgnpolygon, v3f, etc...

You're application is written in IRIS GL, not OpenGL. IRIS GL was the predecessor to OpenGL and is a different thing (almost) entirely. Mesa's not the solution.

3.7.3 2.3 Where is the GLUT library?

GLUT (OpenGL Utility Toolkit) is no longer in the separate `MesaGLUT-x.y.z.tar.gz` file. If you don't already have GLUT installed, you should grab freeglut.

3.7.4 2.4 Where is the GLw library?

GLw (OpenGL widget library) is now available from a separate git repository. Unless you're using very old Xt/Motif applications with OpenGL, you shouldn't need it.

3.8 2.5 What's the proper place for the libraries and headers?

On Linux-based systems you'll want to follow the Linux ABI standard. Basically you'll want the following:

`/usr/include/GL/gl.h` the main OpenGL header

`/usr/include/GL/glu.h` the OpenGL GLU (utility) header

`/usr/include/GL/glx.h` the OpenGL GLX header

`/usr/include/GL/glext.h` the OpenGL extensions header

`/usr/include/GL/glxext.h` the OpenGL GLX extensions header

`/usr/include/GL/osmesa.h` the Mesa off-screen rendering header

`/usr/lib/libGL.so` a symlink to `libGL.so.1`

`/usr/lib/libGL.so.1` a symlink to `libGL.so.1.xyz`

`/usr/lib/libGL.so.xyz` the actual OpenGL/Mesa library. xyz denotes the Mesa version number.

When configuring Mesa, there are three meson options that affect the install location that you should take care with: `--prefix`, `--libdir`, and `-D dri-drivers-path`. To install Mesa into the system location where it will be available for all programs to use, set `--prefix=/usr`. Set `--libdir` to where your Linux distribution installs system libraries, usually either `/usr/lib` or `/usr/lib64`. Set `-D dri-drivers-path` to the directory where your Linux distribution installs DRI drivers. To find your system's DRI driver directory, try executing `find /usr -type d -name dri`. For example, if the `find` command listed `/usr/lib64/dri`, then set `-D dri-drivers-path=/usr/lib64/dri`.

After determining the correct values for the install location, configure Mesa with `meson configure --prefix=/usr --libdir=xxx -D dri-drivers-path=xxx` and then install with `sudo ninja install`.

3.9 3. Runtime / Rendering Problems

3.9.1 3.1 Rendering is slow / why isn't my graphics hardware being used?

If Mesa can't use its hardware accelerated drivers it falls back on one of its software renderers. (eg. classic swrast, softpipe or llvmpipe)

You can run the `glxinfo` program to learn about your OpenGL library. Look for the `OpenGL vendor` and `OpenGL renderer` values. That will identify who's OpenGL library with which driver you're using and what sort of hardware it has detected.

If you're using a hardware accelerated driver you want `direct rendering: Yes`.

If your DRI-based driver isn't working, go to the DRI website for trouble-shooting information.

3.9.2 3.2 I'm seeing errors in depth (Z) buffering. Why?

Make sure the ratio of the far to near clipping planes isn't too great. Look here for details.

Mesa uses a 16-bit depth buffer by default which is smaller and faster to clear than a 32-bit buffer but not as accurate. If you need a deeper you can modify the parameters to `glXChooseVisual` in your code.

3.9.3 3.3 Why Isn't depth buffering working at all?

Be sure you're requesting a depth buffered-visual. If you set the `MESA_DEBUG` environment variable it will warn you about trying to enable depth testing when you don't have a depth buffer.

Specifically, make sure `glutInitDisplayMode` is being called with `GLUT_DEPTH` or `glXChooseVisual` is being called with a non-zero value for `GLX_DEPTH_SIZE`.

This discussion applies to stencil buffers, accumulation buffers and alpha channels too.

3.9.4 3.4 Why does `glGetString()` always return `NULL`?

Be sure you have an active/current OpenGL rendering context before calling `glGetString`.

3.9.5 3.5 `GL_POINTS` and `GL_LINES` don't touch the right pixels

If you're trying to draw a filled region by using `GL_POINTS` or `GL_LINES` and seeing holes or gaps it's because of a float-to-int rounding problem. But this is not a bug. See Appendix H of the OpenGL Programming Guide - "OpenGL Correctness Tips". Basically, applying a translation of (0.375, 0.375, 0.0) to your coordinates will fix the problem.

3.10 4. Developer Questions

3.10.1 4.1 How can I contribute?

First, join the *mesa-dev mailing list*. That's where Mesa development is discussed.

The OpenGL Specification is the bible for OpenGL implementation work. You should read it.

Most of the Mesa development work involves implementing new OpenGL extensions, writing hardware drivers (for the DRI), and code optimization.

3.10.2 4.2 How do I write a new device driver?

Unfortunately, writing a device driver isn't easy. It requires detailed understanding of OpenGL, the Mesa code, and your target hardware/operating system. 3D graphics are not simple.

The best way to get started is to use an existing driver as your starting point. For a classic hardware driver, the i965 driver is a good example. For a Gallium3D hardware driver, the r300g, r600g and the i915g are good examples.

The DRI website has more information about writing hardware drivers. The process isn't well document because the Mesa driver interface changes over time, and we seldom have spare time for writing documentation. That being said, many people have managed to figure out the process.

Joining the appropriate mailing lists and asking questions (and searching the archives) is a good way to get information.

3.10.3 4.3 Why isn't `GL_EXT_texture_compression_s3tc` implemented in Mesa?

Oh but it is! Prior to 2nd October 2017, the Mesa project did not include s3tc support due to intellectual property (IP) and/or patent issues around the s3tc algorithm.

As of Mesa 17.3.0, Mesa now officially supports s3tc, as the patent has expired.

In versions prior to this, a 3rd party plug-in library was required.

CHAPTER 4

Release Notes

The release notes summarize what's new or changed in each Mesa release.

- *10.5.9 release notes*
- *10.6.1 release notes*
- *10.5.8 release notes*
- *10.6.0 release notes*
- *10.5.7 release notes*
- *10.5.6 release notes*
- *10.5.5 release notes*
- *10.5.4 release notes*
- *10.5.3 release notes*
- *10.5.2 release notes*
- *10.4.7 release notes*
- *10.5.1 release notes*
- *10.5.0 release notes*
- *10.4.6 release notes*
- *10.4.5 release notes*
- *10.4.4 release notes*
- *10.4.3 release notes*
- *10.4.2 release notes*
- *10.3.7 release notes*
- *10.4.1 release notes*
- *10.3.6 release notes*
- *10.4 release notes*
- *10.3.5 release notes*
- *10.3.4 release notes*
- *10.3.3 release notes*
- *10.3.2 release notes*
- *10.3.1 release notes*
- *10.2.9 release notes*
- *10.3 release notes*
- *10.2.8 release notes*
- *10.2.7 release notes*
- *10.2.6 release notes*
- *10.2.5 release notes*
- *10.2.4 release notes*
- *10.2.3 release notes*
- *10.2.2 release notes*

- *7.0 release notes*
- *6.5.3 release notes*
- *6.5.2 release notes*
- *6.5.1 release notes*
- *6.5 release notes*
- *6.4.2 release notes*
- *6.4.1 release notes*
- *6.4 release notes*

Versions of Mesa prior to 6.4 are summarized in the *versions file* and the following release notes.

- 6.3.2 release notes
- 6.3.1 release notes
- 6.3 release notes
- 6.2.1 release notes
- 6.2 release notes
- 6.1 release notes
- 6.0.1 release notes
- 6.0 release notes
- 5.1 release notes
- 5.0.2 release notes
- 5.0.1 release notes
- 5.0 release notes
- 4.1 release notes
- 4.0.3 release notes
- 4.0.2 release notes
- 4.0.1 release notes
- 4.0 release notes
- 3.5 release notes
- 3.4.2 release notes
- 3.4.1 release notes
- 3.4 release notes
- 3.3 release notes
- 3.2.1 release notes
- 3.2 release notes
- 3.1 release notes

4.1 Mesa 20.1.10 Release Notes / 2020-10-14

Mesa 20.1.10 is a bug fix release which fixes bugs found since the 20.1.9 release.

Mesa 20.1.10 implements the OpenGL 4.6 API, but the version reported by glGetString(GL_VERSION) or glGetIntegerv(GL_MAJOR_VERSION) / glGetIntegerv(GL_MINOR_VERSION) depends on the particular driver being used. Some drivers don't support all the features required in OpenGL 4.6. OpenGL 4.6 is **only** available if requested at context creation. Compatibility contexts may report a lower version depending on each driver.

Mesa 20.1.10 implements the Vulkan 1.2 API, but the version reported by the apiVersion property of the VkPhysicalDeviceProperties struct depends on the particular driver being used.

4.1.1 SHA256 checksum

```
b1dba69910adac9fcb4cbdfd7833d99a4a8c75b91f3d2e97f0fd0a3cd8c6ee9f  mesa-20.1.10.tar.xz
```

4.1.2 New features

- None

4.1.3 Bug fixes

- [RADV/ACO] Star Citizen Lighting/Shadow Issue
- Graphics corruption in Super Mega Baseball 2 with RADV on Navi
- RADV ACO - ground line corruption in Path of Exile with Vulkan renderer
- omx/tizonia build broken with latest mesa git
- [hsw][bisected][regression] gpu hangs on dEQP-VK.subgroups.(shufflelquad) tests
- TGL B0 Stepping gpu hangs on many dEQP-VK.subgroups.quad nonconst tests

4.1.4 Changes

Alyssa Rosenzweig (1):

- pan/bi: Handle vector moves

Anuj Phogat (1):

- intel/gen9: Enable MSC RAW Hazard Avoidance

Bas Nieuwenhuizen (2):

- radv: Use atomics to read query results.
- radv: Fix mipmap extent adjustment on GFX9+.

Dylan Baker (1):

- glsl/xxd.py: fix imports

Eric Engestrom (4):

- docs/relnotes: add sha256 sums to 20.1.9

- .pick_status.json: Update to 68daac28df1b2f50a43740d1905932cfde0ddf1a
- .pick_status.json: Mark d78df70c2a85fd846d40b71b9e213122347bea1b as denominated
- .pick_status.json: Mark c02e933de4a9a644410384f815c84d1c08107b82 as applied

Jason Ekstrand (5):

- nir/cf: Better handle intra-block splits
- intel/fs: NoMask initialize the address register for shuffles
- nir/opt_load_store_vectorize: Use bit sizes when checking mask compatibility
- intel/fs: Don't use NoDDClk/NoDDClr for split SHUFFLEs
- intel/nir: Don't try to emit vector load_scratch instructions

Lionel Landwerlin (1):

- intel/perf: fix crash when no perf queries are supported

Lucas Stach (1):

- etnaviv: stop leaking the dummy texure descriptor BO

Marek Olšák (1):

- radeonsi: Fix dead lock with aux_context_lock in si_screen_clear_buffer.

Nanley Chery (2):

- iris: Fix a fast-clear skipping optimization
- anv: Enable multi-layer aux-map init for HIZ+CCS

Pierre-Eric Pelloux-Prayer (1):

- omx/tizonia: fix build

Rhys Perry (4):

- spirv: add and use a generator id enum
- spirv: replace discard with demote for incorrect HLSL->SPIR-V translations
- android: fix SPIR-V -> NIR build
- scons: fix SPIR-V -> NIR build

Timothy Arceri (1):

- glsl: don't duplicate state vars as uniforms in the NIR linker

Tony Wasserka (1):

- aco/isel: Always export position data from VS/NGG

Vinson Lee (1):

- freedreno: Move rsc NULL check to before rsc dereferences.

4.2 Mesa 20.2.1 Release Notes / 2020-10-14

Mesa 20.2.1 is a bug fix release which fixes bugs found since the 20.2.0 release.

Mesa 20.2.1 implements the OpenGL 4.6 API, but the version reported by glGetString(GL_VERSION) or glGetIntegerv(GL_MAJOR_VERSION) / glGetIntegerv(GL_MINOR_VERSION) depends on the particular driver being used.

Some drivers don't support all the features required in OpenGL 4.6. OpenGL 4.6 is **only** available if requested at context creation. Compatibility contexts may report a lower version depending on each driver.

Mesa 20.2.1 implements the Vulkan 1.2 API, but the version reported by the apiVersion property of the VkPhysicalDeviceProperties struct depends on the particular driver being used.

4.2.1 SHA256 checksum

```
d1a46d9a3f291bc0e0374600bdcb59844fa3eafaa50398e472a36fc65fd0244a  mesa-20.2.1.tar.xz
```

4.2.2 New features

- None

4.2.3 Bug fixes

- RADV ACO - ground line corruption in Path of Exile with Vulkan renderer
- Graphics corruption in Super Mega Baseball 2 with RADV on Navi
- Running Amber test leads to VK_DEVICE_LOST
- omx/tizonia build broken with latest mesa git
- [hsw][bisected][regression] gpu hangs on dEQP-VK.subgroups.(shufflelquad) tests
- TGL B0 Stepping gpu hangs on many dEQP-VK.subgroups.quad nonconst tests
- [spirv-fuzz] Shader generates a wrong image

4.2.4 Changes

Alyssa Rosenzweig (2):

- pan/bi: Handle vector moves
- pan/bi: Fix simple txl test

Anuj Phogat (1):

- intel/gen9: Enable MSC RAW Hazard Avoidance

Bas Nieuwenhuizen (3):

- radv,radeonsi: Disable compression on interop depth images
- radv: Use atomics to read query results.
- radv: Fix mipmap extent adjustment on GFX9+.

Christian Gmeiner (1):

- etnaviv: simplify linear stride implementation

Connor Abbott (1):

- nir/lower_io_arrays: Fix xfb_offset bug

Danylo Piliaiev (1):

- intel/fs: Disable sample mask predication for scratch stores

Dylan Baker (12):

- docs: add release notes for 20.2.0
- docs: Add sh256 sums for 20.2.0
- .pick_status.json: Update to 291cfb1e41513008a5be08be95399373a7de206d
- meson/anv: Use variable that checks for –build-id
- .pick_status.json: Update to 7dbb1f7462433940951ce6c3fa22f6368aeafd50
- .pick_status.json: Update to e3b814d5e9e414839d5e4de3a76bb2899cbb7249
- .pick_status.json: Update to b32a8f83dce3b8789f2e8790ab41b8a63c9bedc6
- .pick_status.json: Mark b23013db0aa6845d661c2da5d4003615b064e01f as denominated
- .pick_status.json: Mark 4790811d78011d45830d9543ad6e7401391cfb15 as denominated
- glsl/xxd.py: fix imports
- .pick_status.json: Update to e1efc534e6c452e3e606d663864896a654acc185
- retab ac_surface.h so that backports apply

Eric Engestrom (1):

- radv: add missing u_atomic.h include

Erik Faye-Lund (1):

- st/mesa: use roundf instead of floorf for lod-bias rounding

Jason Ekstrand (6):

- nir/liveness: Consider if uses in nir_ssa_defs_interfere
- nir/cf: Better handle intra-block splits
- intel/fs: NoMask initialize the address register for shuffles
- nir/opt_load_store_vectorize: Use bit sizes when checking mask compatibility
- intel/fs: Don't use NoDDClk/NoDDClr for split SHUFFLEs
- intel/nir: Don't try to emit vector load_scratch instructions

Jose Maria Casanova Crespo (3):

- vc4: Avoid negative scissor caused by no intersection
- nir/algebraic: optimize iand/ior of (n)eq zero when umax/umin not available
- vc4: Enable lower_umax and lower_umin

Lionel Landwerlin (1):

- intel/perf: fix crash when no perf queries are supported

Lucas Stach (1):

- etnaviv: stop leaking the dummy texure descriptor BO

Marek Olšák (4):

- radeonsi: fix indirect dispatches with variable block sizes
- radeonsi: Fix dead lock with aux_context_lock in si_screen_clear_buffer.

- gallium/u_threaded_context: fix use-after-free in transfer_unmap
- ac/surface: fix valgrind warnings in DCC retile tile lookups

Nanley Chery (3):

- blorp: Ensure aligned HIZ_CCS_WT partial clears
- iris: Fix a fast-clear skipping optimization
- anv: Enable multi-layer aux-map init for HIZ+CCS

Philipp Zabel (1):

- meson: fix power8 option

Pierre-Eric Pelloux-Prayer (3):

- gallium/vl: do not call transfer_unmap if transfer is NULL
- gallium/vl: add chroma_format arg to vl_video_buffer functions
- omx/tizonia: fix build

Rhys Perry (4):

- spirv: add and use a generator id enum
- android: fix SPIR-V -> NIR build
- scons: fix SPIR-V -> NIR build
- spirv: replace discard with demote for incorrect HLSL->SPIR-V translations

Samuel Pitoiset (1):

- aco: implement missing nir_op_unpack_half_2x16_split_{x,y}_flush_to_zero

Timothy Arceri (1):

- glsl: don't duplicate state vars as uniforms in the NIR linker

Vinson Lee (2):

- gallium/dri2: Move image->texture assignment after image NULL check.
- freedreno: Move rsc NULL check to before rsc dereferences.

4.3 Mesa 20.2.0 Release Notes / 2020-09-28

Mesa 20.2.0 is a new development release. People who are concerned with stability and reliability should stick with a previous release or wait for Mesa 20.2.1.

Mesa 20.2.0 implements the OpenGL 4.6 API, but the version reported by glGetString(GL_VERSION) or glGetIntegerv(GL_MAJOR_VERSION) / glGetIntegerv(GL_MINOR_VERSION) depends on the particular driver being used. Some drivers don't support all the features required in OpenGL 4.6. OpenGL 4.6 is **only** available if requested at context creation. Compatibility contexts may report a lower version depending on each driver.

Mesa 20.2.0 implements the Vulkan 1.2 API, but the version reported by the apiVersion property of the VkPhysicalDeviceProperties struct depends on the particular driver being used.

4.3.1 SHA256 checksum

```
63f0359575d558ef98dd78adffc0df4c66b76964ebf603b778b7004964191d30  mesa-20.2.0.tar.xz
```

4.3.2 New features

- GL_ARB_compute_variable_group_size on Iris.
- GL_ARB_gpu_shader5 on llvmpipe
- GL_ARB_post_depth_coverage on llvmpipe
- GLES 3.2 on llvmpipe
- GL_EXT_shader_group_vote on GLES3.
- GL_EXT_texture_shadow_lod on llvmpipe
- VK_AMD_texture_gather_bias_lod on RADV.
- VK_AMD_gpu_shader_half_float on RADV/ACO.
- VK_AMD_gpu_shader_int16 on RADV/ACO.
- VK_EXT_extended_dynamic_state on ANV and RADV.
- VK_EXT_image_robustness on RADV.
- VK_EXT_private_data on ANV and RADV.
- VK_EXT_custom_border_color on ANV and RADV.
- VK_EXT_pipeline_creation_cache_control on ANV and RADV.
- VK_EXT_shader_demote_to_helper_invocation on RADV/LLVM.
- VK_EXT_subgroup_size_control on RADV/ACO.
- VK_GOOGLE_user_type on ANV and RADV.
- VK_KHR_shader_subgroup_extended_types on RADV/ACO.
- GL_ARB_gl_spirv on nvc0/nir.
- GL_ARB_spirv_extensions on nvc0/nir.
- RADV now uses ACO per default as backend
- RADV_DEBUG=llvm option to enable LLVM backend for RADV
- VK_EXT_image_robustness for ANV
- VK_EXT_shader_atomic_float on ANV
- VK_EXT_4444_formats on ANV and RADV.
- VK_KHR_memory_model on RADV.
- GL 4.5 on llvmpipe
- EGL_KHR_swap_buffers_with_damage on X11 (DRI3)

4.3.3 Bug fixes

- [Regression][Bisected][20.2][radeonsi] American Truck Simulator continually allocates memory until OOM
- anv: dEQP-VK.robustness.robustness2.* failures on gen12
- [RADV] Problems reading primitive ID in fragment shader after tessellation
- Massive memory leak (at least AMD, others unknown)
- Substance Painter 6.1.3 black glitches on Radeon RX570
- vkCmdCopyImage broadcasts subsample 0 of MSAA src into all subsamples of dst on RADV
- Crash in ruvd_end_frame when calling vaBeginPicture/vaEndPicture without rendering anything
- X-Plane 11 Installer crashes on startup since *glsl: declare gl_Layer/gl_ViewportIndex/gl_ViewportMask as vs builtins*
- Horizon Zero Dawn graphics corruption with with radv
- Amber test opt_peel_loop_initial_if: Assertion failed
- Dirt Rally: Flickering glitches on certain foliage since Mesa 20.1.0 caused by MSAA
- [BRW] WRC 5 asserts with gallium nine and iris.
- radv: Corruption in "The Surge 2"
- [RADV] Detroit: Become Human Demo game lock-ups with RADV
- Road Redemption certain graphic effects rendered white color
- vulkan/wsi/x11: deadlock with Xwayland when compositor holds multiple buffers
- [RADV/ACO] Death Stranding cause a GPU hung (*ERROR* Waiting for fences timed out!)
- lp_bld_init.c:172:7: error: implicit declaration of function 'LLVMAddConstantPropagationPass'; did you mean 'LLVMAddCorrelatedValuePropagationPass'? [-Werror=implicit-function-declaration]
- Intel Vulkan driver crash with alpha-to-coverage
- EGL_KHR_swap_buffers_with_damage support on X11
- radv: blitting 3D images with linear filter
- [ACO] Compiling pipelines from RPCS3's shader interpreter spins forever in ACO code
- Intel Vulkan driver assertion with small xfb buffer
- [spirv-fuzz] SPIR-V parsing failed "src->type->type == dest->type->type"
- radeonsi: radeonsi crashes in Chrome on chromeos
- [RADV] commit d19bc94e4eb94 broke gamescope with Navi
- 4e3a7dcf6ee4946c46ae8b35e7883a49859ef6fb breaks Gamescope showing windows properly.
- anv: crashes in CTS test dEQP-VK.subgroups.*.framebuffer.*_tess_eval
- Intel Vuikan (anv) crash in copy_non_dynamic_state() when using validation layer
- Mafia 3: Trees get rendered incorrectly
- radv: dEQP-VK.synchronization.op.multi_queue.timeline_semaphore.write_clear_attachments_*_concurrent fail when forcing DCC.
- Crash on GTA 5 through proton 5.0.9 and GE versions
- Mesa 20.2.0-rc1 fails to build for AMD

- Assertion failure compiling shader from Zigguart
- Panfrost locks for waiting fence when running Source engine games
- ci: *-Dtools=panfrost* should be build-tested
- panfrost: Register allocation fails for Firefox WebRender shaders
- VRAM leak with vuilkan external memory + opengl memory objects
- [vulkan/build] Recent build system changes made VK_EXT_acquire_xlib_display unnecessarily depend on GBM
- ci: Capture devcoredumps on chezas
- Possible array out of bounds in brw_vec4_nir.cpp
- freedreno/a6xx: incorrect rendering in asphalt 9
- [tgl][bisected][regression][iris] failure on dEQP-EGL.functional.wide_color.pbuffer_8888_colorspace_default
- Multiply defined symbols compiling with gcc@10.1.0
- shrinking descriptor pool on intel+vulkan
- dEQP-VK.renderpass2.dedicated_allocation.attachment.1.12 fails on NAVI14
- turnip: binning and indirect dependency
- Amber test leads to NIR validation failed after nir_opt_if (on spirv-fuzz shader)
- Unable to compile mesa-git from b559d26c
- Ambient light too bright with ACO in AC: Odyssey
- Multiple issues with Detroit Become Human
- ci: Capture artifacts in baremetal mode
- turnip/ir3: fine derivatives
- panfrost: regression: Major stuttering and low compositor FPS with glmark2
- khr_debug-push-pop-group_gl: ../src/util/simple_mtx.h:86: simple_mtx_lock: Assertion 'c != _SIMPLE_MTX_INVALID_VALUE' failed.
- freedreno/a6xx: skai/skqp fails
- SPIR-V parsing fails in src/compiler/spirv/spirv_to_nir.c
- SPIR-V parsing fails in src/compiler/spirv/vtn_cfg.c
- Weird GLSL bug
- iris driver is broken in Freedesktop 19.08
- LLVM not properly shutdown in *si_pipe.c*?
- Panfrost: add current status to docs/features.txt
- Opengl incorrect rendering on yuzu Amd
- RADV: VK_ACCESS_MEMORY_READ/WRITE_BIT is not implemented
- [bisected][regression][all platforms] multiple deqp-gles31/glescts/piglit failures
- 7406ea37, "ac/surface: require that gfx8 doesn't have DCC in order to be displayable", breaks Gamescope being able to launch games on RX580, and possibly other gfx8 cards
- vkGetSemaphoreCounterValue doesn't update without vkWaitSemaphores calls on Intel UHD 620

- [RADV] System crash when playing XCOM Chimera Squad because of commit #7a5e6fd2
- [RADV] Non-precise occlusion queries return non-zero when all fragments are discarded
- [DXVK] Project Cars rendering problems
- ADDRLIB ODR Violation
- Build fails with current mesa from git "undefinierter Verweis auf »nir_lower_clip_disable«"
- KDE Compositor stuttering after Check for window destruction in dri3_wait_for_event_locked
- Add fallthrough to prevent errors caused by missing break
- i965/20.1: gray rendering with torcs racing
- glBindBufferRange call seems to be ignored by one of two shader-programs on radeon cards
- [bisected][g33] piglit.spec.ext_framebuffer_object.fbo-cubemap failure
- Increase GL_MAX_COMPUTE_SHADER_STORAGE_BLOCKS to greater value.
- nir: st_nir_lower_builtin fails for gl_LightSource[i]
- Sometimes VLC player process gets stuck in memory after closure if video output used is Auto or OpenGL
- Double unlock in rbug_context.c
- Double copy for TexSubImage
- [v3d] corruption when GS omits some vertices
- Iris crashes when reading from multisampled front buffer on platforms without front buffer
- freedreno: subway surfers crash when repeatedly toggling fullscreen
- [RADV/GFX8] Performance drop in DOOM Eternal when "Present from compute" is enabled
- freedreno: multiple applications crash on a5xx
- Use-after-free crash innv50_ir::GCRA::RIG_Node::init()
- intel: Sample mask writes need to be honored in Vulkan
- [RADV] - Path of Exile (238960) - Map outline, landscape and markers are missing with the Vulkan renderer.
- ASTC texture decompression fails when using software fallback
- [i965][iris][regression][bisected] multiple piglit and glcts failures on all platforms
- please publish GPG keyring used to sign new releases
- [BISECTED] compiling shader causes crash
- Missing render Information on Stellaris
- freedreno/ir3: allow copy-propagate from array
- Zink + GALLIUM_HUD SIGSEGV
- piglit spec@egl_ext_device_base@conformance fails LLVM 11 Git assertion since "llvmpipe/fs: add caching support"
- llvmpipe: 1x1 framebuffer with a 2x2 viewport
- [regression] nir build failure
- ci: need to end baremetal tests after kernel panic/instaboot
- If-statement body is executed for false condition

- freedreno/a6xx: broken rendering in playcanvas "after the flood"
- [regression] performance drop on Dota 2, CS:GO, and gfxbench GL benchmarks on ICL/Iris
- [amd] C++ ODR violatation for union GB_ADDR_CONFIG
- Zink reports incorrect amount of video memory
- [RADV/LLVM]: void llvm::ICmpInst::AssertOK(): Assertion 'getOperand(0)->getType() == getOperand(1)->getType() && "Both operands to ICmp instruction are not of the same type!"' failed.
- glsl-1.50-gs-max-output hangs on Navi10 + NGG
- anv: Runs out of binding tables with PPSSPP during long runs
- Segfault in Panfrost with waypipe
- ci: Use rsync instead of rm -rf ; cp for baremetal rootfs
- i965: Rendering problems replaying a trace of "Refunct" after mesa-20.1.0-rc1 release [bisected]
- Panfrost (rk3399 NanoPi M4) hang/crash on playing video on Kodi/X11
- gallium/winsys/radeon/drm fails assertion on 32bit
- NIR validation failed after glsl to nir, before function inline, wrong {src,dst}->type ?
- nir/spirv asin() function not precise enough
- Mesa 20.0.7 / 20.1.0-rc4 regression, extremally long shader compilation time in NIR
- Android build error after 689acc73
- freedreno/a6xx: gpu hangs in google earth
- Mesa-git build fails on Fedora Rawhide
- Doom Eternal 1.1 performs very poorly on RADV
- iris/i965: possible regression in 20.0.5 due to changes in buffer manager sharing across screens (firefox/mozilla#1634213)
- iris/i965: possible regression in 20.0.5 due to changes in buffer manager sharing across screens (firefox/mozilla#1634213)
- Incorrect _NetBSD__ macro inside execmem.c
- Possible invalid sizeof in device.c
- YUV FP16 lowering validation failing
- GLSL compiler assertion is_float() failed in glsl/ir_validate.cpp, visit_leave on specific WebGL shader
- [RADV] - Doom Eternal (782330) & Metro Exodus (412020) - Title requires 'RADV_DEBUG=zerovram' to eliminate colorful graphical aberrations.
- [RADV] - Doom Eternal (782330) & Metro Exodus (412020) - Title requires 'RADV_DEBUG=zerovram' to eliminate colorful graphical aberrations.
- mesa trunk master vulkan overlay-layer meson.build warning empty configuration_data() object
- [meson] increase minimum required version
- Kicad fails to render 3D PCB models.
- freedreno: minetest: alpha channel issue on a6xx
- Reproduceable i915 gpu hang Intel Iris Plus Graphics (Ice Lake 8x8 GT2)
- 7 Days to Die - "Reflection Quality" setting broken, results in environment rendered black

- glsl: regression affecting shader compilation time
- freedreno: glamor issue with x11 desktops
- finish converting from fnv1a to xxhash
- Hang in iris_dri in kitty
- Setting twice value to output_stream in radv_nir_to_llvm.c
- Overwriting value of *jit_tex->sample_stride* in lp_setup.c
- [AMDGPU][OpenGL] apitrace of kernel/firmware crash that requires a reboot
- Flickering in Superposition benchmark
- Double lock in fbobject.c
- Possible typo in aco_insert_waitcnt.cpp
- [bisected] Steam crashes when newest Iris built with LTO
- Freeing null pointer inside radv_amdgpu_cs.c
- Duplicated sub expression in radv_nir_to_llvm.c
- i965/vec4: opt_cse_local cause the out of bound array access
- NIR: Regression on shader using 8/16-bit integers
- ACO: Compiler segfault on 8/16-bit integers.
- lp_bld_intr.c:70:16: error: use of undeclared identifier 'LLVMFixedVectorTypeKind'; did you mean 'LLVMVectorTypeKind'?
- recent seqno changes causing surfaceflinger crash
- [radeonsi] [glthread] Crash with glthread enabled
- Deadlock in anv_timelines_wait()
- [gles3] supertuxkart: some textures are incorrect
- post_version.py does not work with release candidates
- post_version.py does not work with release candidates
- radv regression on android
- ogl: Set mesa_glthread=true as default on the RPCS3 emulator
- [iris] android deqp dEQP-EGL.functional.robustness.negative_context#invalid_notification_strategy_enum fails
- zink: conditional rendering
- [RadeonSI] Glitches on VEGA8 + RX 560X after MR 4863
- RadeonSI OpenGL broken for GFX8 after unify code for overriding offset
- freedreno/turnip: Don't request fragcoord components we don't use
- Make check fails in ANV
- srcutilmeson.build:294:4: ERROR: Program or command 'winepath' not found or not executable
- Please add Zink to features.txt
- llvmpipe: assert triggers in LLVM
- debug builds are massively broken on Windows

- ci: Report flakes on IRC from baremetal tests
- heavy glitches on amd ryzen 5 since version 20.x
- zink asserts with 32-bit boolean
- OpenGL: Surviving Mars black screen late-game (possible shader problem)
- Kerbal Space Program (KSP) hangs entire Navi system
- Dirt: Showdown bad performance and broken rendering with enabled advanced lightning
- gravit & Firefox WebGL broken since 3dc2ccc14c0e035368fea6ae3cce8c481f3c4ad2 "ac/surface: replace RADEON_SURF_OPTIMIZE_FOR_SPACE with !FORCE_SWIZZLE_MODE"
- mesa 20.0.5 causing kitty to crash
- radeonsi: "Torchlight II" trace showing regression on mesa-20.0.6 [bisected]
- [RADV/LLVM/ACO/Regression] After mesa commit a3dc7fffbb7be0f1b2ac478b16d3acc5662dff66 all games stucks at start
- Android building error after commit 2ab45f41
- freedreno/a6xx: pubg rendering glitches
- iris: Crash when trying to capture window in OBS Studio
- lp_test_format failure with llvm-11

4.3.4 Changes

Abhishek Kumar (1):

- egl: Limit the EGL ver for android

Adam Jackson (1):

- glx: Fix build and warnings with -Dglx=dri -Dglx-direct=false

Alejandro Piñeiro (9):

- v3d/tex: only look up the 2nd texture gather offset for 1d non-arrays
- v3d/tex: set up default values for Configuration Parameter 1 if possible
- v3d/tex: use TMUSLOD register if possible
- v3d: moving v3d simulator to src/broadcom
- v3d/tex: handle correctly coordinates for cube/cubearrays images
- vulkan/util: add struct vk_pipeline_cache_header
- nir/lower_tex: handle query lod with nir_lower_tex_packing_16 at lower_tex_packing
- v3d/packet: fix typo on Set InstanceID/PrimitiveID packet
- v3d: set instance id to 0 at start of tile

Alyssa Rosenzweig (475):

- pan/mdg: Track more types
- pan/mdg: Be a bit more pedantic in invert passes
- panfrost: Enumify bifrost blend types
- pan/bi: Add texture indices to IR

- pan/bi: Pipe multiple textures through
- pan/bi: Pack round opcodes (FMA, either 16 or 32)
- pan/bit: Add framework forinterpreting double vs float
- pan/bit: Interpret ROUND
- pan/bit: Add round tests
- panfrost: Fix texture field size
- panfrost: Fix size of bifrost sampler descriptor
- panfrost: Fix sampler wrap/filter field orders
- panfrost: Fix norm coords on bifrost sampler
- panfrost: Fix tiled texture "stride"s on Bifrost
- pan/decode: Don't crash on missing payload
- pan/bi: Enable lower_mediump_outputs NIR pass
- panfrost: Update Bifrost fields in mali_shader_meta
- pan/bi: Lower for now sincos
- pan/mdg: Ingest actual isub ops
- pan/mdg: Rename .one to .sat_signed
- pan/mdg: Move constant switch opts to algebraic pass
- pan/mdg: Drop forever todo
- pan/mdg: Drop *opt* in name of midgard_opt_cull_dead_branch
- pan/mdg: Enable nir_opt_algebraic_distribute_src_mods
- panfrost: Update dEQP expectation list
- panfrost: Setup gl_FragCoord as sysval on Bifrost
- pan/bi: Add clause type for gl_FragCoord.zw load
- pan/bi: Abort on unknown op packing
- pan/bi: Abort on unhandled intrinsics
- pan/bi: Futureproof COMBINE lowering against non-u32
- pan/bi: Print bad instruction on src packing fail
- pan/bi: Passthrough direct ld_var addresses
- pan/bi: Lower gl_FragCoord
- pan/bi: Set clause type for gl_FragCoord.z
- pan/bi: Fix double-abs flipping
- pan/bi: Fix missing swizzle
- pan/bi: Fix incorrectly flipped swizzle
- pan/bi: Disable CSEL4 emit for now
- pan/bi: Fix DISCARD ops in disasm
- pan/bi: Structify DISCARD

- pan/bi: Remove BI_GENERIC
- pan/bi: Unwrap BRANCH into CONDITIONAL class
- pan/bi: Handle discard_if in NIR->BIR naively
- pan/bi: Emit discard (not if)
- pan/bi: Add float-only mode to condition fusing
- pan/bi: Fuse conditions into discard_if
- pan/bi: Handle discard/branch in get_component_count
- pan/bi: Pack ADD.DISCARD
- pan/bi: Structify ADD ICMP 16
- pan/bi: Pack ADD ICMP 32
- pan/bi: Pack ADD ICMP 16
- pan/bi: Don't pack ICMP on FMA
- pan/bit: Add swizzles to round tests
- pan/bit: Add more 16-bit fmod tests
- pan/bit: Add ICMP tests
- pan/bi: Rename BI_ISUB to BI_IMATH
- pan/bi: Use IMATH for nir_op_iadd
- pan/bi: Pack FMA IADD/ISUB 32
- pan/bi: Pack ADD IADD/ISUB for 8/16/32
- pan/bi: Add SUB.v2i16/SUB.v4i8 opcodes to disasm
- pan/bi: Don't schedule <32-bit IMATH to FMA
- pan/bit: Interpret IMATH
- pan/bit: Interpret v4i8 ops
- pan/bit: Remove test names
- pan/bit: Use swizzle helper for round
- pan/bit: Factor out identity swizzle helper
- pan/bit: Add IMATH packing tests
- pan/decode: Fix flags_hi printing
- pan/mdg: Explain helper invocations dataflow theory
- pan/mdg: Analyze helper invocation termination
- pan/mdg: Analyze helper execution requirements
- pan/mdg: Use the helper invo analyze passes
- pan/mdg: Use analysis to set .cont/.last flags
- pan/mdg: Remove texture_op_count
- pan/mdg: Set types for derivatives
- pan/mdg: Fix derivative swizzle

- panfrost: Run dEQP-GLES3.functional.shaders.derivate.* on CI
- pan/decode: Use a page table for tracking mmaps
- pan/decode: Fix min/max_tile_coord mixup
- pan/mfbd: Add format codes for PIPE_FORMAT_B5G5R5A1_UNORM
- panfrost: Switch formats to table
- panfrost: Fix Z24 vs Z32 mixup
- panfrost: Enable AFBC for Z24X8
- nir: Add fsat_signed opcode
- nir: Add fclamp_pos opcode
- panfrost: Add modifier detection helpers
- pan/mdg: Remove .pos propagation pass
- pan/mdg: Drop nir_lower_to_source_mods
- pan/mdg: Prepare for modifier helpers
- pan/mdg: Ingest fsat_signed/fclamp_pos
- pan/mdg: Apply abs/neg modifiers
- pan/mdg: Treat inot as a modifier
- pan/mdg: Remove invert optimizations
- pan/mdg: Use helpers for branch/discard inversion
- pan/mdg: Apply outmods
- pan/mdg: Emit fcsel when beneficial
- pan/mdg: Optimize pipelining logic
- pan/mdg: Precompute mir_special_index
- pan/mdg: Optimize liveness computation in DCE
- pan/mdg: Handle comparisons in fp16 path
- pan/mdg: Fix constant combining crash
- pan/mdg: Remove mir_*size routines
- pan/mdg: Remove mir_get_alu_src
- pan/mdg: Include more types
- pan/mdg: Handle dest up/lower correctly with swizzles
- pan/mdg: Respect !32-bit sizes in RA
- pan/mdg: Explain ld/st sign/zero extension
- pan/mdg: Add abs/neg/shift modifiers to IR
- pan/mdg: Use src_types to determine size in scheduling
- pan/mdg: Use type to determine triviality of a move
- pan/mdg: Identify scalar integer mods
- pan/mdg: Promote imov to fmov on a NIR level

- pan/mdg: Remove promote_float pass
- pan/mdg: Defer modifier packing until emit time
- pan/mdg: Remove redundant redundancy
- pan/mdg: Streamline dest_override handling
- pan/mdg: Implement b2f16
- pan/mdg: Don't generate conversions for fp16 LUTs
- pan/mdg: Ignore dest.type when offseting load swizzle
- pan/lcra: Remove unused alignment parameters
- pan/lcra: Allow per-variable bounds to be set
- pan/mdg: Use type size to determine alignment
- pan/mdg: Eliminate load_64
- pan/mdg: Set RA bounds for fp16
- pan/mdg: Print mask when dest=0
- pan/mdg: Round up bytemasks when spilling
- pan/mdg: Print constant vectors less wrong
- pan/mdg: Factor out mir_adjust_constant
- pan/mdg: Only combine 16-bit constants to lower half
- pan/mdg: Separately pack constants to the upper half
- pan/mdg: Fix type checking issues with compute
- pan/mdg: Pack barriers correctly
- pan/mdg: Use shifts instead of division for RA sizes
- pan/mdg: Implement vector constant printing for 8-bit
- pan/mdg: Implement condense_writemask for 8-bit
- pan/mdg: Pack 8-bit swizzles in 16-bit ops
- panfrost: Guard experimental fp16 behind debug flag
- panfrost: Keep cached BOs mmap'd
- panfrost: Remove deadcode
- panfrost: Fill in SCALED formats to format table
- panfrost: Don't set PIPE_CAP_VERTEX_BUFFER_STRIDE_4BYTE_ALIGNED_ONLY
- panfrost: Don't zero staging buffer for tiling
- panfrost: Allow bpp24 tiling
- panfrost: Allow tiling on RECT textures
- panfrost: Limit blend shader work count
- panfrost: Remove dated comment about leaks
- panfrost: Disable tib read/write when colourmask = 0x0
- panfrost: Avoid redundant shader executions with mask=0x0

- panfrost: Don't set CAN_DISCARD for MFBD
- panfrost: Fix transform feedback types
- pan/mdg: Cleanup comments that look like division
- pan/mdg: Eliminate expand_writemask division
- pan/mdg: Eliminate 64-bit swizzle packing division
- pan/mdg: Avoid division in printing helpers
- pan/mdg: Eliminate remaining divisions from compiler
- panfrost: Fix dated comment
- panfrost: Use _mesa_roundevenf when packing clear colours
- panfrost: Handle !independent_blend for blend shaders
- pan/mdg: Add pack_colour_32 opcode
- pan/mdg: Lower shifts to 32-bit
- pan/mdg: Ensure we don't DCE into impossible masks
- pan/mdg: Allow DCE on ld_color_buffer masks
- panfrost: Add debug print before query flushes
- panfrost: Only run batch debug when specifically asked
- nir: Add un/pack_32_4x8 opcodes
- util: Add SATURATE macro
- util/format: Use SATURATE
- mesa: Use SATURATE
- mesa/swrast: Use SATURATE
- gallium/draw: Use SATURATE
- glsl: Use SATURATE
- panfrost: Use SATURATE
- softpipe: Use SATURATE
- intel: Use SATURATE
- i965: Use SATURATE
- iris: Use SATURATE
- etnaviv: Use SATURATE
- nouveau: Use SATURATE
- pan/decode: Fix unused variable warning
- pan/decode: Fix tiler warning
- pan/decode: Dump missing field on Bifrost
- pan/decode: Dump unknown2
- panfrost: Fix Bifrost blending with depth-only FBO
- panfrost: Adjust null_rt for Bifrost

- panfrost: Tweak zsbuf magic numbers for Bifrost
- panfrost: Tweak Bifrost colour buffer magic
- panfrost: Force Z/S tiling on Bifrost
- panfrost: Share MRT blend flag calculation with Bifrost
- panfrost: Set unk2 to accomodate blending
- panfrost: Identify Bifrost texture format swizzle
- panfrost: Ensure nonlinear strides are 16-aligned
- panfrost: Document Midgard Inf/NaN suppress bit
- panfrost: Add defines for bifrost unk1 flags
- panfrost: Identify MALI_BIFROST_EARLY_Z flag
- panfrost: Set MALI_BIFROST_EARLY_Z as necessary
- pan/decode: Decode Bifrost shader flags
- pan/bi: Add TEX.vtx opcode for vertex texturing
- pan/bi: Also add compact vertex texturing
- pan/bi: Document compute_lod bit for compact tex
- pan/bi: Allow vertex txl with lod=0 as compact
- pan/bi: Add f16 TEXC.vtx op
- pan/bi: Pack compact vertex texturing
- pan/bi: Add CSEL.16 packing tests
- pan/bi: Suppress inf/nan for now
- panfrost: Don't generate gl_FragCoord varying on Bifrost
- panfrost: Set reads_frag_coord as a sysval
- panfrost: Preload gl_FragCoord on Bifrost
- pan/bi: Remove FMA? parameter from get_src
- pan/bi: Remove comment about old scheduler design
- pan/bi: Move bi_registers to common IR structures
- pan/bi: Move bi_registers to bi_bundle
- pan/bi: Drop *struct* from bi_registers
- pan/bi: Add FILE* argument to bi_print_registers
- pan/bi: Move bi_flip_ports out of port assignment
- pan/bi: Document constant count invariant
- pan/bi: Disassemble pos=0xe
- pan/bi: Add MUL.i32 to disasm
- pan/bi: Remove more artefacts of 2-pass scheduling
- pan/bi: Add bi_layout.c for clause layout helpers
- pan/bi: Add helper to measure clause size

- pan/bi: Remove schedule_barrier
- pan/bi: Allow printing branches without targets
- pan/bi: Fix emit_if successor assignment
- pan/bi: Only rewrite COMBINE dest if not SSA
- pan/bi: Fix CONVERT component counting
- pan/bi: Fix branch condition typesize
- pan/bi: Passthrough ZERO in branch packing
- pan/bi: Add branch constant field to IR
- pan/bi: Pack branch offset constants
- pan/bi: Set branch_constant if there is a branch
- pan/bi: Assign constant port for branch offsets
- pan/bi: Preliminary branch packing
- pan/bi: Link clauses back to their blocks
- pan/bi: Add bi_foreach_clause_in_block_from{_rev} helpers
- pan/bi: Measure distance between blocks
- pan/bi: Pack proper clause offsets
- pan/bi: Set branch_conditional if b2b is set
- pan/bi: Set back-to-back bit more accurately
- pan/bi: Set branch conditional bit
- pan/bi: Pack unconditional branch
- pan/bi: Defer block naming until after emit
- pan/bi: Add bi_foreach_block_from_rev helper
- pan/bi: Measure backwards branches as well
- pan/bi: Allow two successors in header packing
- pan/bi: Passthrough deps of the branch target
- panfrost: Disable QUAD_STRIP/POLYGON on Bifrost
- panfrost: Add GPU IDs for G31/G52
- panfrost: Probe G31/G52 if PAN_MESA_DEBUG=bifrost
- pan/mdg: Handle un/pack opcodes as moves
- pan/mdg: Add pack_unorm_4x8 via 8-bit
- pan/mdg: Treat packs “specially”
- pan/mdg: Handle bitsize for packs
- pan/mdg: Print 8-bit constants
- pan/mdg: Drop the u8 from the colorbuf op names
- pan/mdg: Implement raw colourbuf loads on T720
- panfrost: Add theory for new framebuffer lowering

- panfrost: Determine unpacked type for formats
- panfrost: Add quirks for blend shader types
- panfrost: Determine load classes for formats
- panfrost: Determine classes for stores
- panfrost: Stub out lowering boilerplate
- panfrost: Un/pack pure 32-bit
- panfrost: Un/pack pure 16-bit
- panfrost: Un/pack pure 8-bit
- panfrost: Un/pack 8-bit UNORM
- panfrost: Flesh out dispatch
- panfrost: Un/pack UNORM 4
- panfrost: Un/pack RGB565 and RGB5A1
- panfrost: Un/pack RGB10_A2_UNORM
- panfrost: Un/pack RGB10_A2_UINT
- panfrost: Un/pack R11G11B10
- panfrost: Un/pack sRGB via NIR
- panfrost: Switch to pan_lower_framebuffer
- panfrost: Conditionally allow fp16 blending
- panfrost: Account for differing types in blend lower
- panfrost: Let Gallium pack colours
- panfrost: Check for large tilebuffer requirements
- panfrost: Add separate_stencil BO to batch
- panfrost: Use internal_format throughout
- panfrost: Update fails list
- pan/mdg: Handle 16-bit ld_vary
- pan/mdg: Fuse f2f16 into load_interpolated_input
- panfrost: Fix PRESENT flag mix-up
- panfrost: Permit AFBC of RGB8
- panfrost: Use VTX tag for vertex texturing
- panfrost: Don't flush explicitly when mipmapping
- panfrost: Remove unused nir_lower_framebuffer pass
- pan/mdg: Disassemble out-of-order bits
- pan/mdg: Add quirk for missing out-of-order support
- pan/mdg: Enable out-of-order execution after texture ops
- nir: Fold f2f16(b2f32(x)) to b2f16(x)
- pan/mdg: Don't double-replicate blend on T720

- pan/mdg: Distinguish blend shaders in internal shader-db
- pan/mdg: Add roundmode enum
- pan/mdg: Add opcode roundmode property
- pan/mdg: Lower roundmodes
- pan/mdg: Implement *_rtz conversions with roundmode
- pan/mdg: Fold roundmode into applicable instructions
- pan/mdg: Handle f2u8
- pan/mdg: Allow f2u8 and friends thru
- pan/mdg: Handle regular nir_intrinsic_load_output
- panfrost: Passthrough NATIVE loads/stores
- pan/bi: Handle SEL with vec3 16-bit
- pan/bi: Fix SEL.16 swizzle
- pan/bi: Pack second argument of F32_TO_F16
- pan/bi: Passthrough second argument of F32_TO_F16
- pan/bi: Handle vectorized load_const
- panfrost: Update MALI_EARLY_Z description
- panfrost: Document MALI_WRITES_GLOBAL bit
- panfrost: Handle writes_memory correctly
- panfrost: Readd MIDGARD_SHADERLESS quirk to t760
- panfrost: Explicitly convert to 32-bit for logic-ops
- pan/bi: Disassemble gl_PointCoord reads.
- panfrost: Prefer sysval for gl_PointCoord on Bifrost
- panfrost: Fix gl_PointSize out of GL_POINTS
- panfrost: Mark point sprites as todo on Bifrost
- pan/mdg: Legalize inverts with constants
- pan/mdg: Ensure ld_vary_16 is aligned
- panfrost: Ensure we have ro before using it
- nir: Remove nir_intrinsic_output_u8_as_fp16_pan
- pan/mdg: Avoid fusing ld_vary_16 with non-zero component
- panfrost: Calculate varying size by format
- panfrost: Add panfrost_streamout_offset helper
- panfrost: Introduce bitfields for tracking varyings
- panfrost: Determine varying buffer presence
- panfrost: Emit unlinked varyings
- panfrost: Emit special varyings
- panfrost: Emit xfb records

- panfrost: Add helper to determine if we are capturing
- panfrost: Add high-level varying emit
- panfrost: Use new varying linking
- panfrost: Remove unused routines
- panfrost: Allow R/RG/RGB varyings
- panfrost: Only store varying formats
- panfrost: Use shader_info harder
- panfrost: Override varying format to minimal precision
- panfrost: Demote mediump varyings to fp16
- pan/mdg: Explicitly type 64-bit uniform moves
- pan/mdg: Analyze types for 64-bitness in RA
- pan/mdg: Prefer type over regmode for schedule constraints
- pan/mdg: Precolour blend inputs
- panfrost: Merge bifrost_bo/midgard_bo
- panfrost: Update sampler view in Bifrost path
- panfrost: Fix level_2
- panfrost: Correctly calculate tiled stride
- panfrost: Enable AFBC for RGB565
- panfrost: Simplify AFBC format check
- pan/mdg: Factor out unit check
- pan/mdg: Allow scheduling "x + x" to multipliers
- pan/mdg: Canonicalize (x * 2.0) to (x + x)
- pan/mdg: Reassociate adds for multiply-by-two
- nir: Propagate *2*16 conversions into vectors
- panfrost: Specify stack_shift on SFBD
- pan/mdg: Defer nir_fuse_io_16 until after opts
- pan/mdg: Don't assign destination in writeout block to r1
- pan/mdg: Remove bundle interference code
- pan/mdg: Schedule writeout to VLUT
- pan/mdg: Defer smul, vlut until after writeout moves
- pan/mdg: Allow Z/S writes to use any 2nd stage unit
- pan/mdg: Prioritize non-moves on VADD/VLUT
- pan/mdg: Skip r1.w write where possible
- pan/mdg: Schedule based on liveness
- pan/mdg: Respect type/mask in mir_lower_special_reads
- pan/mdg: Fix indirect UBO swizzles

- pan/decode: Fix MSAA texture decoding
- pan/decode: Identify layered MSAA flag
- pan/mdg: Allow ignoring move mode
- pan/mdg: Handle GLSL_SAMPLER_DIM_MS
- pan/mdg: Handle nir_tex_src_ms_index
- pan/mdg: Handle nir_texop_txf_ms
- pan/mdg: Use _VTX tag for texelFetch in frag shaders
- panfrost: Set depth to sample_count for MSAA 2D
- panfrost: Identify layer_stride
- panfrost: Allocate space for multisampling
- panfrost: Index texture by sample
- panfrost: Include pointer for each sample
- panfrost: Set layer_stride for multisampled rendering
- panfrost: Don't advertise MSAA 2x
- panfrost: Identify coverage_mask
- panfrost: Pass sample_mask to the hardware
- panfrost: Implement alpha-to-coverage
- panfrost: Identify depth/stencil layer strides
- panfrost: Set depth/stencil_layer_stride accordingly
- panfrost: Enable MSAA if we render to such a surface
- panfrost: Save sample_mask before blitting
- panfrost: Expose MSAA 4x
- glsl: Handle 16-bit types in loop analysis
- docs/features: Track Panfrost
- panfrost: Introduce pan_pool struct
- panfrost: Allocate pool BOs against the pool
- panfrost: Track the device through the pool
- panfrost: Expose pool-based allocation API
- panfrost: Move debug flags into the device
- panfrost: Drop Gallium-local pan_bo_create wrapper
- panfrost: Move pool routines to common code
- panfrost: Factor out scoreboarding state
- panfrost: Pass polygon_list to tiler init function
- panfrost: Drop batch from scoreboard routines
- panfrost: Move scoreboarding routines to common
- panfrost: Handle PIPE_FORMAT_X24S8_UINT

- panfrost: Handle PIPE_FORMAT_S8_UINT
- panfrost: Move panfrost_translate_texture_type
- panfrost: Report blend shader work count
- panfrost: Clamp pure int pixels
- panfrost: Generate shader variants on framebuffer bind
- panfrost: Always use SOFTWARE for pure formats
- panfrost: Extend fetched framebuffer results
- panfrost: Fix fence leak
- panfrost: Fix write to free'd memory
- panfrost: Add a sparse array to map GEM handles to BOs
- panfrost: Index BOs from the BO map sparse array
- panfrost: Merge PAN_BO_IMPORTED/PAN_BO_EXPORTED
- panfrost: Remove PAN_BO_COHERENT_LOCAL
- panfrost: Remove PAN_BO_DONT_REUSE
- panfrost: Remove panfrost_bo_access type
- panfrost: Compact unused BO flag bits
- panfrost: Add format codes for new compressed textures
- panfrost: Pipe in compressed texture feature mask
- panfrost: Filter compressed texture formats
- panfrost: Map PIPE_{DXT, RGTC, BPTC} to MALI_BCn
- docs/features: Update ASTC entries for Panfrost
- pan/mdg: Bump compiler RT maximum
- pan/mdg: Identify per-sample interpolation mode
- pan/mdg: Implement gl_SampleID
- panfrost: Force Z/S writeback
- panfrost: Expose panfrost_get_blend_shader
- panfrost: Add MALI_PER_SAMPLE bit
- panfrost: Include sample count in payload estimates
- panfrost: Identify zs_samples field
- panfrost: Add rectangle subtraction algorithm
- panfrost: Handle per-sample shading
- panfrost: Set zs_samples as necessary
- panfrost: Track surfaces drawn per-batch
- panfrost: Extract panfrost_batch_reserve_framebuffer
- panfrost: Use Midgard-specific reloads
- panfrost: Call util_blitter_save_fragment_constant_buffer_slot

- panfrost: Overhaul tilebuffer allocations
- panfrost: Set PIPE_CAP_MIXED_COLORBUFFER_FORMATS
- panfrost: Fix sRGB clear colour packing
- panfrost: Implement Z32F_S8 blits
- panfrost: Abort on unsupported blit
- panfrost: Avoid integer underflow in rt_count_1
- panfrost: Honour cso->compare_mode
- panfrost: Fix faults with RASTERIZER_DISCARD
- panfrost: Report CAPs more honestly
- panfrost: Enable Chromium
- panfrost: Revert "Disable frame throttling"
- docs/features: Mark trivial missed feature
- panfrost: Enable FP16 by default
- panfrost: Avoid wait=true flushing all batches
- panfrost: Remove wait parameter to flush_all_batches
- panfrost: Skip specifying in_syncs
- panfrost: Allocate syncobjs in panfrost_flush
- panfrost: Remove unused batch_fence->signaled
- panfrost: Remove unused batch_fence->ctx
- pan/bit: Update f32->f16 convert test
- pan/bit: Remove BI_SHIFT stub
- pan/mdg: Mask spills from texture write
- pan/mdg: Test for SSA before chasing addresses
- docs/features: Add GL_EXT_multisampled_render_to_texture
- panfrost: Add MSAA mode selection field
- panfrost: Implement EXT_multisampled_render_to_texture
- panfrost: Set STRIDE_4BYTE_ALIGNED_ONLY
- panfrost: Fix WRITES_GLOBAL bit
- pan/mdg: Ensure barrier op is set on texture
- panfrost: Fix blend leak for render targets 5-8
- panfrost: Free cloned NIR shader
- panfrost: Free NIR of blit shaders
- panfrost: Free hash_to_temp map
- pan/mdg: Free previous liveness
- panfrost: Use memctx for sysvals
- panfrost: Free batch->dependencies

- pan/mdg: Fix discard encoding
- pan/mdg: Fix perspective combination
- pan/bit: Set d3d=true for CMP tests

Andreas Baierl (1):

- nir/ lower_int_to_float: Handle umax and umin

Andres Gomez (10):

- .mailmap: add an alias for Iago Toral Quiroga
- .mailmap: add an alias for Andres Gomez
- gitlab-ci: update tracie README after changes in main script
- scripts: remove unittest.mock dependency when not used
- gitlab-ci: create always the "results" directory with tracie
- gitlab-ci: correct tracie behavior with replay errors
- gitlab-ci: build gfxreconstruct from the "dev" branch
- gitlab-ci: get the last frame from a gfxr trace using gfxrecon-info
- gitlab-ci/traces: updated paths and checksums for POLARIS10 traces
- gitlab-ci: Test AMD's Raven with traces

Andrey Vostrikov (1):

- egl/x11: Free memory allocated for reply structures on error

Andrii Simiklit (3):

- glsl_type: don't serialize padding bytes from glsl_struct_field
- i965/vec4: Ignore swizzle of VGRF for use by var_range_end()
- glsl: fix crash on glsl macro redefinition

Ani (1):

- drirc: Enable glthread for rpcs3

Anuj Phogat (6):

- intel/devinfo: Add is_dg1 to device info
- intel/l3: Add DG1 L3 configuration
- intel/ehl: Use GEN11_URB_MIN_MAX_ENTRIES in device info
- intel/ehl: Use macro GEN11_LP_FEATURES in device info
- intel/ehl: Rename gen_device_info struct
- intel/ehl: Add new PCI-IDs

Arcady Goldmints-Orlov (4):

- anv: increase minUniformBufferOffsetAlignment to 64
- intel/compiler: fix alignment assert in nir_emit_intrinsic
- nir/spirv/glsl450: increase asin(x) precision
- intel/compiler: Always apply sample mask on Vulkan.

Axel Davy (19):

- st/nine: Set correctly blend max_rt
- gallium/util: Fix leak in the live shader cache
- ttn: Add new allow_disk_cache parameter
- ttn: Implement disk cache
- st/nine: Enable ttn cache
- radeonsi: Enable tgsi to nir disk cache
- st/nine: Add checks for pure device
- st/nine: Return error when setting invalid depth buffer
- st/nine: Do not return invalidcall on getrenderstate
- st/nine: Pass more adapter formats for CheckDepthStencilMatch
- st/nine: Improve return error code in CheckDeviceFormat
- st/nine: Fix uninitialized variable in BEM()
- st/nine: Fix a crash if the state is not initialized
- st/nine: Add missing NULL checks
- st/nine: Increase available GPU memory
- st/nine: Retry allocations after freeing some space
- st/nine: Improve pDestRect handling
- st/nine: Ignore pDirtyRegion
- st/nine: Handle full pSourceRect better

Bas Nieuwenhuizen (80):

- radv: Fix implicit sync with recent allocation changes.
- radv: Extend tiling flags to 64-bit.
- radv: Provide a better error for permission issues with priorities.
- radv: Support VK_PIPELINE_COMPILE_REQUIRED_EXT.
- radv: Support VK_PIPELINE_CREATE_EARLY_RETURN_ON_FAILURE_BIT_EXT.
- radv: Support VK_PIPELINE_CACHE_CREATE_EXTERNALLY_SYNCHRONIZED_BIT_EXT.
- radv: Expose VK_EXT_pipeline_creation_cache_control.
- radv/winsys: Finish mapping for sparse residency.
- radv/winsys: Remove extra sizeof multiply.
- radv: Handle failing to create .cache dir.
- radv: Remove dead code.
- radv: Do not close fd -1 when NULL-winsys creation fails.
- radv: Implement vkGetSwapchainGrallocUsage2ANDROID.
- frontend/dri: Implement mapping individual planes.
- util/format: Add VK_FORMAT_D16_UNORM_S8_UINT.

- util/format: Use correct pipe format for VK_FORMAT_G8_B8_R8_3PLANE_420_UNORM.
- util/format: Add more multi-planar formats.
- gallium/dri: Remove lowered_yuv tracking for plane mapping.
- radeonsi: Explicitly map Z16_UNORM_S8_UINT to None for GFX10.
- amd/common,radeonsi: Move gfx10_format_table to common.
- radeonsi: Define gfx10_format in the common header.
- radv: Include gfx10_format_table.h only from a single source file.
- radv: Use common gfx10_format_table.h
- radv: Use ac_surface to determine fmask enable.
- radv: Pass no_metadata_planes info in to ac_surface.
- radv: Enforce the contiguous memory for DCC layers in ac_surface.
- radv: Rely on ac_surface for avoiding cmask for linear images.
- radv: Use offsets in surface struct.
- radv: Disable DCC in ac_surface.
- radv: Disable HTILE in ac_surface.
- radv: Allocate values/predicates at the end of the image.
- amd/common: Add total alignment calculation.
- radv: Use ac_surface to allocate aux surfaces.
- vulkan/wsi/x11: Ensure we create at least minImageCount images.
- radv/winsys: Deal with realloc failures in BO lists.
- radv: Handle mmap failures.
- radv/winsys: Distinguish device/host memory errors.
- radv: Make radv_alloc_shader_memory static.
- turnip: semaphore support.
- meson: Do not require shader cache for radv.
- amd/addrlib: fix another C++ one definition rule violation
- radv: Set handle types in Android semaphore/fence import.
- radv: Always enable PERFECT_ZPASS_COUNTS.
- Revert "radv: add support for MRTs compaction to avoid holes"
- radv: Use correct semaphore handle type for Android import.
- amd/llvm: Mark pointer function arguments as 32-byte aligned.
- amd/common: Cache intra-tile addresses for retile map.
- amd/addrlib: Clean up unused colorFlags argument
- amd/registers: add RLC_PERFMON_CLK_CNTL for pre-GFX10
- radeonsi: Inhibit clock-gating for perf counters.
- meson: Add mising git_sha1.h dependency.

- amd: Add detection of timeline semaphore support.
- radv/winsys: Add binary syncobj ABI changes for timeline semaphores.
- radv: Add thread for timeline syncobj submission.
- radv: Add winsys support for submitting timeline syncobj.
- radv: Add winsys functions for timeline syncobj.
- radv: Add timeline syncobj for timeline semaphores.
- radv: Fix uninitialized variable in renderpass.
- vulkan/wsi/x11: report device-group present rectangles with prime.
- vulkan/wsi: Convert usage of -1 to UINT32_MAX.
- radv: Fix host->host signalling with legacy timeline semaphores.
- mesa/st: Actually free the driver part of memory objects on destruction.
- radv: Don't use both DCC and CMASK for single sample images.
- radv: Fix assert that is too strict.
- radv: Do not consider layouts fast-clearable on compute queue.
- radv: When importing an image, redo the layout based on the metadata.
- radv: Use getter instead of setter to extract value.
- driconf: Support selection by Vulkan applicationName.
- radv: Override the uniform buffer offset alignment for World War Z.
- radv: Fix handling of attribs 16-31.
- radv: Remove conformance warnings with ACO.
- radv: Update CTS version.
- radv: Fix 3d blits.
- radv: Fix threading issue with submission refcounts.
- radv: Avoid deadlock on bo_list.
- spirv: Deal with glslang not setting NonUniform on constructors.
- radeonsi: Work around Wasteland 2 bug.
- spirv: Deal with glslang bug not setting the decoration for stores.
- ac/surface: Fix depth import on GFX6-GFX8.
- st/mesa: Deal with empty textures/buffers in semaphore wait/signal.

Ben Skeggs (38):

- nir: use bitfield_insert instead of bfi in nir_lower_double_ops
- nvir: bump max encoding size of instructions
- nvir: introduce OP_LOP3_LUT
- nvir: introduce OP_WARPSYNC
- nvir: introduce OP_BREV with lowering to EXTBF_REV for current GPUs
- nvir: introduce OP_SHF

- nvir: introduce OP_BMSK
- nvir: introduce OP_SGXT
- nvir: introduce OP_FINAL
- nvir: add constant folding for OP_PERMT
- nvir: run replaceZero() before replaceCvt()
- nvir/nir: fix fragment program output when using MRT
- nvir/nir: move nir options to codegen
- nvir/nir: flesh out options
- nvir/nir: turn on lower_rotate
- nvir/nir: implement nir_op_extract_u8
- nvir/nir: implement nir_op_extract_i8
- nvir/nir: implement nir_op_extract_u16
- nvir/nir: implement nir_op_extract_i16
- nvir/nir: implement nir_op_urol
- nvir/nir: implement nir_op_uror
- nvir/nir: nir expects the shift amount to wrap, rather than clamp
- nvir/nir: use nir_lower_idiv
- nvir/gm107: implement OP_PERMT
- nvir/gm107: replace SHR+AND+AND with PRMT+PRMT in PFETCH lowering
- nvir/gm107: separate out header for sched data calculator
- nvir/nir/gm107: split nir shader compiler options from gf100
- nvir/nir/gm107: turn on nir_lower_extract64
- nvir/nir/gm107: switch off lower_extract_byte
- nvir/nir/gm107: switch off lower_extract_word
- nvir/gv100: initial support
- nvir/gv100: enable support for tu1xx
- nvc0: use NVIDIA headers for GK104->GM2xx compute QMD
- nvc0: use NVIDIA headers for GP100- compute QMD
- nvc0: move setting of entrypoint for a shader stage to a function
- nvc0: remove hardcoded blitter vertprog
- nvc0: initial support for gv100
- nvc0: initial support for tu1xx

Benjamin Cheng (1):

- drirc: Add picom to adaptive_sync exclusion list

Benjamin Tissoires (3):

- CI: reduce bandwidth for git pull

- gitlab-ci: update ci-fairy minio to latest upstream
- gitlab-ci: do not run full CI on scheduled pipelines

Blaž Tomažič (1):

- radeonsi: Fix omitted flush when moving suballocated texture

Boris Brezillon (14):

- spirv: Split the vtn_emit_scoped_memory_barrier() logic
- nir: Replace the scoped_memory barrier by a scoped_barrier
- intel/compiler: Extract control barriers from scoped barriers
- spirv: Use scoped barriers for SpvOpControlBarrier
- nir: Add new rules to optimize NOOP pack/unpack pairs
- nir: Use a switch in build_deref_offset()/deref_instr_get_const_offset()
- nir: Allow casts in nir_deref_instr_get[_const]_offset()
- freedreno: Initialize lower_int64_options to a proper value
- nir: Stop passing an options arg to nir_lower_int64()
- nir: Extend nir_lower_int64() to support i2f/f2i lowering
- intel: Set int64_options to ~0 when lowering 64b ops
- nir: Get rid of __[u]int64_to_fp32() and __fp32_to_[u]int64()
- nir: Fix i64tof32 lowering
- spirv: Add a vtn_get_mem_operands() helper

Boyuan Zhang (2):

- radeon/vcn/enc: Re-write PPS encoding for HEVC
- radeon/vcn: bump vcn3.0 encode major version to 1

Brian Ho (14):

- turnip: Execute ir3_nir_lower_gs pass again
- turnip: Fill out VkPhysicalDeviceSubgroupProperties
- nir: Support sysval tess levels in SPIR-V to NIR
- nir: Add an option for lowering TessLevelInner/Outer to vecs
- turnip: Lower shaders for tessellation
- turnip: Offset by component when lowering gl_TessLevel*
- turnip: Parse tess state and support PATCH primtype
- turnip: Allocate tess BOs as a function of draw size
- turnip: Update VFD_CONTROL with tess system values
- turnip: Emit HS/DS user consts as draw states
- turnip: Support tess for draws
- turnip: Force sysmem for tessellation
- ir3: Unconditionally enable MERGEDREGS on a6xx

- turnip: Enable tessellationShader physical device feature

Caio Marcelo de Oliveira Filho (32):

- intel/dev: Bail when INTEL_DEVID_OVERRIDE is not valid
- intel/fs: Clean up variable group size handling in backend
- intel/fs: Add an option to lower variable group size in backend
- intel/fs: Add and use a new load_simd_width_intel intrinsic
- intel: Let drivers call brw_nir_lower_cs_intrinsics()
- iris: Implement ARB_compute_variable_group_size
- util/list: Add list_foreach_entry_from_safe
- nir: Use deref intrinsics to set writes_memory when gathering info
- intel/fs: Use writes_memory from shader_info
- nir: Consider atomic counter intrinsics when setting writes_memory
- intel/fs: Remove unused emission of load_simd_with_intel
- intel/fs: Remove unused state from brw_nir_lower_cs_intrinsics
- intel/fs: Early return when can't satisfy explicit group size
- intel/fs: Remove redundant assert()
- intel/fs: Remove min_dispatch_width spilling decision from RA
- intel/fs: Support INTEL_DEBUG=no8,no32 in compute shaders
- intel/fs: Add helper to get prog_offset and simd_size
- i965: Use new helper functions to pick SIMD variant for CS
- iris: Set CS KernelStatePointer at dispatch
- iris: Use new helper functions to pick SIMD variant for CS
- anv: Use new helper functions to pick SIMD variant for CS
- intel/fs: Generate multiple CS SIMD variants for variable group size
- iris, i965: Drop max_variable_local_size
- iris, i965: Update limits for ARB_compute_variable_group_size
- intel: Add helper to calculate GPGPU_WALKER::RightExecutionMask
- nir: Fix printing execution scope of a scoped barrier
- spirv: Memory semantics is optional for OpControlBarrier
- intel/fs: Add Fall-through comment
- nir: Fix logic that ends combine barrier sequence
- spirv: Handle most execution modes earlier
- nir: Filter modes of scoped memory barrier in nir_opt_load_store_vectorize
- spirv: Propagate explicit layout only in types that need it

Charmaine Lee (1):

- llvmpipe: do not enable tessellation shader without llvm coroutines support

Chris Forbes (12):

- bifrost: Set RTZ rounding mode for f2i conversion
- bifrost: Lower x->bool conversions to != 0
- bifrost: Emit "d3d" variant of comparison instructions
- bifrost: Document d3d/gl comparison control bit
- bifrost: Add lowering for b2i32
- bifrost: Add support for nir_op_inot
- bifrost: Add support for nir_op_ishl
- bifrost: Add support for nir_op_uge
- bifrost: Add support for nir_op_imul
- bifrost: Add support for nir_op_iabs
- bifrost: Honor src swizzle in special math ops
- bifrost: Fix packing of ADD_FEXP2_FAST

Chris Wilson (6):

- iris: Place a seqno at the end of every batch
- iris: Convert fences to using lightweight seqno
- iris: Store a seqno for each batch in the fence
- iris: Initialise stub iris_seqno to 0
- iris: Rename iris_seqno to iris_fine_fence
- iris: Fixup copy'n'paste mistake in Makefile.sources

Christian Gmeiner (31):

- etnaviv: fix SAMP_ANISOTROPY register value
- etnaviv: do not use int filter when anisotropic filtering is used
- ci: bare-metal: make it possible to use a script for serial
- ci: extend expect-output.sh
- ci: add U-Boot specific fetch strings
- etnaviv: drop translate_blend(..)
- ci: add arm_test-base docker image
- ci: use separate docker images for baremetal builds
- ci: fix possible spuriously run of jobs
- etnaviv: delete not used struct
- etnaviv: convert enums
- etnaviv: move etna_lower_io(..) to etnaviv_nir.c
- etnaviv: get rid of etna_compile dependency
- etnaviv: move etna_lower_alu(..) to etnaviv_nir.c
- etnaviv: drop OPT_V define

- etnaviv: make more use of compile_error(..)
- etnaviv: move liveness related stuff into own file
- etnaviv: merge struct etna_compile and etna_state
- etnaviv: drop emit macro
- etnaviv: move functions that generate asm to own file
- etnaviv: move nir compiler related stuff into .c file
- etnaviv: move ra into own file
- etnaviv: replace prims-emitted query
- ci: bare-metal: use nginx to get results from DUT
- etnaviv: explicitly set nir_variable_mode
- etnaviv: introduce struct etna_compiler
- etnaviv: move shader_count to etna_compiler
- etnaviv: do register setup only once
- etnaviv: fix nir validation problem
- etnaviv: call nir_lower_bool_to_bitsize
- etnaviv: completely turn off MSAA

Christopher Egert (2):

- radv: use util_float_to_half_rtz
- r600: Use TRUNC_COORD on samplers

Clément Guérin (1):

- radv: Always expose non-visible local memory type on dedicated GPUs

Con Kolivas (1):

- Linux: Change minimum priority threads from SCHED_IDLE to nice 19 SCHED_BATCH.

Connor Abbott (88):

- tu: Support pipelines without a fragment shader
- tu: Add a "scratch bo" allocation mechanism
- tu: Add noubwc debug flag to disable UBWC
- tu: Implement fallback linear staging blit for CopyImage
- freedreno/a6xx: Document dual-src blending enable bits
- ir3: Fixup dual-source blending slot
- tu: Move RENDER_COMPONENTS setting to pipeline state
- tu: Implement dual-src blending
- tu: Advertise COLOR_ATTACHMENT_BLEND_BIT for blendable formats
- tu: Always initialize image_view fields for blit sources
- tu: Fall back to 3d blit path for BC1_RGB_* formats
- tu: Fix buffer compressed pitch calculation with unaligned sizes

- tu: Support VK_FORMAT_FEATURE_BLIT_SRC_BIT for texture-only formats
- tu: Fix IBO descriptor for cubes
- tu: Respect VK_IMAGE_CREATE_MUTABLE_FORMAT_BIT
- tu: Add missing storage image/texel buffer bits
- tu: Remove useless post-binning flushes
- tu: Don't actually track seqno's for events
- tu: Remove useless event_write helpers
- tu: Rewrite flushing to use barriers
- tu: Fix context faults loading unused descriptor sets
- ir3: Pass reserved_user_consts to ir3_shader_from_nir()
- tu: Remove num_samp hack
- tu: Use the ir3 shader API
- tu: Remove tu_shader_compile_options
- tu: Set num_components to 0 when building bindless intrinsics
- ir3: Don't calculate num_samp ourselves
- tu: Actually remove dead variables after io lowering
- ir3: Split out variant-specific lowering and optimizations
- ir3, freedreno: Round up constlen earlier
- ir3: Include ir3_compiler from ir3_shader
- ir3: Support variants with different constlen's
- ir3: Add ir3_trim_constlen()
- tu: Share constlen between different stages properly
- freedreno: Refactor ir3_cache shader compilation
- freedreno: Share constlen between different stages properly
- freedreno: On a5xx+ INDX_SIZE is MAX_INDICES
- freedreno/registers: Label firstIndex field in CP_DRAW_INDX_OFFSET
- tu: Pass firstIndex directly to CP_DRAW_INDX_OFFSET
- freedreno/a6xx: use firstIndex field
- nir: Refactor load/store intrinsic helper
- nir: add vec2_index_32bit_offset address format
- tu: Rewrite variable lowering
- tu: Enable KHR_variable_pointers
- ir3: Add layer_zero variant bit
- tu: Force gl_Layer to 0 when necessary
- freedreno/a6xx: Force gl_Layer to 0 when necessary
- freedreno: Include adreno_pm4.xml.h before adreno_a6xx.xml.h

- freedreno: Sync registers with envytools
- freedreno/a6xx: Rename and document HLSQ_UPDATE_CNTL
- freedreno/a6xx: Add some documentation for shared consts
- tu: Don't invalidate irrelevant state when changing pipeline
- freedreno/a6xx: Add stencilref register info
- ir3: Handle gl_FragStencilRefARB
- tu: Enable VK_EXT_shader_stencil_export
- freedreno: Add a helper for computing guardband sizes
- tu: Use common guardband helper
- freedreno: Use common guardband helper
- freedreno/ir3: Fix SSBO size for bindless SSBO's
- tu: Enable VK_EXT_depth_clip_enable
- freedreno: Clean up CP_DRAW_MULTI_INDIRECT definition
- freedreno: Add INDIRECT_COUNT CP_DRAW_INDIRECT_MULTI variants
- tu: Integrate WFI/WAIT_FOR_ME/WAIT_MEM_WRITES with cache tracking
- tu: Add missing wfi to tu6_emit_hw()
- tu: Implement VK_KHR_draw_indirect_count
- tu: Fix empty blit scissor case
- tu: Fix hangs for DS with no output
- tu: Detect invalid-for-binning renderpass dependencies
- tu: Enable vertex & fragment stores & atomics
- tu: Fix descriptor update templates with input attachments
- ir3: Validate bindless samp_tex correctly
- ir3: Remove redundant samp_tex validation
- ir3: Fix incorrect src flags for samp_tex
- tu: Enable resource dynamic indexing
- freedreno/rnn: Return success when parsing addvariant
- tu: Dump CP_DRAW_INDIRECT_MULTI draw BO's
- freedreno/rnn: Support stripes in rnndec_decodereg
- freedreno/cffdec: Handle CP_DRAW_INDIRECT_MULTI like other draws
- freedreno: Add trace for CP_DRAW_INDIRECT_MULTI
- freedreno/a6xx: Fix CP_BIN_SIZE_ADDRESS name
- freedreno/rnn: Make rnn_decode_enum() respect variants
- freedreno/cffdec: Stop open-coding enum parsing
- freedreno/afuc: Add missing rnn_prepdb()
- freedreno/afuc: Fix PM4 enum parsing

- tu: Fix DST_INCOHERENT_FLUSH copy/paste error
- freedreno: Document draw predication packets
- tu: Reset has_tess after renderpass
- tu: Implement VK_EXT_conditional_rendering

D Scott Phillips (4):

- intel/fs: Update location of Render Target Array Index for gen12
- anv,iris: Fix input vertex max for tcs on gen12
- intel/dump_gpu: Fix name of LD_PRELOAD in env append logic
- anv/gen11+: Disable object level preemption

Daniel Schürmann (54):

- aco: either copy-propagate or inline create_vector operands
- aco: coalesce parallelcopies during register allocation
- nir: add nir_intrinsic_elect to divergence analysis
- nir: refactor divergence analysis state
- nir: rework phi handling in divergence analysis
- nir: simplify phi handling in divergence analysis
- nir: reset ssa-defs as non-divergent during divergence analysis instead of upfront
- aco: fix WQM coalescing
- aco: restrict copying of create_vector operands to GFX9+
- aco: don't move create_vector subdword operands to unsupported register offsets
- aco: fix corner case in register allocation
- aco: don't allow unaligned subdword accesses on GFX6/7
- aco: fix register assignment for p_create_vector on GFX6/7
- aco: simplify statistics collection for copies
- aco: use full-register instructions to implement subdword packing on GFX6/7
- aco: Workarounds subdword lowering on GFX6/7
- aco: adjust GFX6 subdword lowering workarounds for 8bit
- aco: add and use scratch SGPR to lower subdword p_create_vector on GFX6/7
- aco: coalesce copies more aggressively when lowering to hw
- aco: skip partial copies on first iteration when lowering to hw
- aco: optimize packing of 16bit subdword registers on GFX6/7
- aco: remove unnecessary split- and create_vector instructions for subdword loads
- aco: fix shared subdword loads
- aco: reorder calls to aco_validate() and cleanup aco_compile_shader()
- aco: don't allow SGPRs on logical phis
- aco: fix WQM handling in nested loops

- radv/aco: implement logic64 instead of lowering
- aco: align swap operations to 4 bytes on GFX6/7
- aco: don't allow partial copies on GFX6/7
- radv: introduce RADV_DEBUG=llvm option
- radv: change use_aco -> use_llvm
- radv: enable ACO by default
- aco: fix partial copies on GFX6/7
- aco: remove superflous (bool & exec) if the result comes from VOPC
- nir: also move vecN in case of nir_move_copies
- nir: refactor nir_can_move_instr
- nir/algebraic: optimize bcsel(a, 0, 1) to b2i
- nir: also move b2i in case of nir_move_copies
- nir/algebraic: optimize iand/ior of (n)eq zero
- nir/algebraic: add optimizations for fsign/isign
- nir/algebraic: add some more unop + bcsel optimizations
- nir/algebraic: optimize fmul(x, bcsel(c, -1.0, 1.0)) -> bcsel(c, -x, x)
- nir/algebraic: optimize (a < 0.0) ? -a : a -> fabs(a)
- nir/algebraic: add distributive rules for ior/iand
- nir/algebraic: propagate b2i out of ior/iand
- nir/algebraic: fold some nested bcsel
- aco: fix scratch loads which cross element_size boundaries
- aco: ensure to not extract more components than have been fetched
- aco: don't split store data if it was already split into more elements
- aco: prevent infinite recursion in RA for subdword variables
- aco: ensure readfirstlane subdword operands are always dword aligned
- radv: call radv_nir_lower_ycbcr_textures after first optimizations
- aco: add GFX6/7 subdword lowering tests
- aco: execute branch instructions in WQM if necessary

Daniel Stone (13):

- CI: Disable Panfrost T7x0 jobs
- CI: Re-enable Panfrost T7x0 jobs
- llvmpipe: Expect increased exp precision on Windows
- CI: Windows: Build LLVM and llvmpipe
- CI: Disable Panfrost T720/T760
- Revert "CI: Disable Panfrost T720/T760"
- CI: Enable assertions on Windows

- CI: Try shared libraries on Windows
- CI: Correct build-directory path on Windows, and keep it
- CI: Re-enable the Windows VS2019 build job
- CI: Temporarily disable Panfrost T860 jobs
- CI: Re-enable Panfrost T860 jobs
- CI: Disable Windows build due to unstable infrastructure

Danylo Piliaiev (25):

- glsl: rename has_implicit_uint_to_int_conversion to *_int_to_uint_*
- i965: Fix out-of-bounds access to brw_stage_state::surf_offset
- anv: Translate relative timeout to absolute when calling anv_timelines_wait
- anv: Fix deadlock in anv_timelines_wait
- meson: Disable GCC's dead store elimination for memory zeroing custom new
- mesa: Fix double-lock of Shared->FrameBuffers and usage of wrong mutex
- st/mesa: Clear texture's views when texture is removed from Shared->TexObjects
- intel/fs: Work around dual-source blending hangs in combination with SIMD16
- glsl: Don't replace lrp pattern with lrp if arguments are not floats
- glsl: inline functions with unsupported return type before converting to nir
- i965: Work around incorrect usage of glDrawRangeElements in UE4
- st/mesa: account for "loose", per-mipmap level textures in CopyImageSubData
- iris: Honor scanout requirement from DRI
- iris: Fix fast-clearing of depth via glClearTex(Sub)Image
- nir/opt_if: Fix opt_if_simplification when else branch has jump
- nir/tests: Add tests for opt_if_simplification
- st/mesa: Treat vertex outputs absent in outputMapping as zero in mesa_to_tgsi
- anv/nir: Unify inputs_read/outputs_written between geometry stages
- spirv: Only require bare types to match when copying variables
- glsl: Eliminate out-of-bounds triop_vector_insert
- intel/compiler: Fix pointer arithmetic when reading shader assembly
- glsl: Eliminate assigments to out-of-bounds elements of vector
- nir/lower_io: Eliminate oob writes and return zero for oob reads
- nir/large_constants: Eliminate out-of-bounds writes to large constants
- nir/lower_samplers: Clamp out-of-bounds access to array of samplers

Daryl W. Grunau (1):

- prevent multiply defined symbols

Dave Airlie (199):

- i965: add support for gen 5 pipelined pointers to dump

- i965: disable shadow batches when batch debugging.
- draw/tess: free tessellation control shader i/o memory.
- llvmpipo/nir: free compute shader NIR
- llvmpipe: simple texture barrier implementation.
- gallivm/sample: add multisample support for texel fetch
- gallivm/sample: add multisample image operation support
- gallivm/nir/tgsi: add multisample texture sampling.
- gallivm/nir: add multisample support to image size
- gallivm/nir: add multisample image operations
- draw: introduce sampler num samples + stride members
- draw: add support for num_samples + sample_stride to the image paths
- llvmpipe: add num_samples/sample_stride support to jit textures
- llvmpipe: add samples support to image jit
- util: add a resource wrapper to get resource samples
- llvmpipe: add multisample support to texture allocator.
- llvmpipe: add a max samples define set to 4.
- gallium/util: split out zstencil clearing code.
- llvmpipe: fix race between draw and setting fragment shader.
- llvmpipe: add get_sample_position support (v2)
- llvmpipe/jit: pass fragment sample mask via jit context.
- llvmpipe: pass incoming sample_mask into fragment shader context.
- llvmpipe: add internal multisample texture mapping path.
- llvmpipe: add multisample resource copy region support.
- llvmpipe: add clear texture support for multisample textures.
- llvmpipe: handle multisample render target clears
- draw: disable point/line smoothing for multisample (v2)
- llvmpipe: pass color and depth sample strides into fragment shader.
- llvmpipe: record sample info for color/depth buffers in scene
- llvmpipe/rast: fix tile clearing for multisample color and depth tiles
- llvmpipe: plumb multisample state bit into setup code.
- llvmpipe: add multisample bit to fragment shader key.
- llvmpipe: change mask input to fragment shader to 64-bit.
- llvmpipe: add cbuf/zsbuf + coverage samples to the fragment shader key.
- gallivm: add sample id/pos intrinsic support
- gallivm: add mask api to force mask
- nir/tgsi: translate the interp location

- llvmpipe: pass interp location into interpolation code.
- llvmpipe: add centroid interpolation support.
- llvmpipe: add per-sample interpolation.
- llvmpipe: move getting mask value out of depth code. (v2)
- llvmpipe: add per-sample depth/stencil test
- llvmpipe: move some fs code around
- llvmpipe: multisample sample mask + early/late depth pass
- llvmpipe: handle multisample early depth test/late depth write
- llvmpipe: interpolate Z at sample points for early depth test.
- llvmpipe: handle multisample color stores.
- llvmpipe: hook up sample position system value
- llvmpipe: add multisample alpha to coverage support.
- llvmpipe: add multisample alpha to one support
- llvmpipe: handle gl_SampleMask writing.
- llvmpipe: don't allow branch to end for early Z with multisample
- llvmpipe: pass mask store into interp for centroid interpolation
- llvmpipe: move color storing earlier in frag shader
- llvmpipe: fix multisample occlusion queries.
- llvmpipe: disable opaque variant for multisample
- llvmpipe: add new rast api to pass full 64-bit mask.
- llvmpipe: add fixed point sample positions to scene.
- llvmpipe: build 64-bit coverage mask in rasterizer
- llvmpipe: fixup multisample coverage masks for covered tiles
- llvmpipe: generate multisample triangle rasterizer functions (v2)
- llvmpipe: choose multisample rasterizer functions per triangle (v2)
- llvmpipe: choose correct position for multisample
- llvmpipe: don't choose pixel centers for multisample
- drisw: add multisample support to sw dri layer.
- llvmpipe: enable 4x sample MSAA + texture multisample
- gallivm/sample: add num samples query for txqs (v2)
- gallivm/nir: hooks up texture samples queries
- llvmpipe: enable GL_ARB_shader_texture_image_samples
- llvmpipe: add min samples support to the fragment shader.
- llvmpipe: enable ARB_sample_shading
- llvmpipe: make sample position a global array.
- zink: enable conditional rendering if available

- r600: enable TEXCOORD semantic for TGSI.
- r600/sfn: plumb the chip class into the instruction emission
- r600/sfn: fix cayman float instruction emission.
- r600/sfn: cayman fix int trans op2
- r600/sfn: add callstack non-evergreen support
- r600/sfn: add emit if start cayman support
- llvmpipe: don't use sample mask with 0 samples
- llvmpipe: use per-sample position not sample id for interp
- llvmpipe/interp: fix interpolating frag pos for sample shading
- llvmpipe: remove non-simple interpolation paths.
- gallivm/nir: add an interpolation interface.
- llvmpipe/interp: refactor out use of pixel center offset
- llvmpipe/interp: refactor out centroid calculations
- llvmpipe: add interp instruction support
- llvmpipe/fs: hook up the interpolation APIs.
- gallivm/nir: add sample_mask_in support
- llvmpipe: add gl_SampleMaskIn support.
- r600/sfn: fix nop channel assignment.
- llvmpipe: compute shaders work better with all the threads.
- llvmpipe: move coroutines out of noopt case
- ci: bump virglrenderer to latest version
- util/disk_cache: add fallback for disk_cache_get_function_identifier
- llvmpipe/cs: overhaul cs variant key state.
- llvmpipe/draw: drop variant number from function names.
- gallivm: rework coroutine malloc/free callouts.
- gallivm: rework debug printf hook to use global mapping.
- gallivm: add support for a cache object
- gallivm: skip operations if we have a cached object.
- gallivm: add cache interface to mcjit
- llvmpipe: add infrastructure for disk cache support
- gallivm: don't cache shaders that use fetch functions.
- llvmpipe/fs: add caching support
- llvmpipe/cs: add shader caching
- draw: add disk cache callbacks for draw shaders
- llvmpipe: hook draw disk cache up
- draw: add disk caching for draw shaders

- draw/gs: fix emitting inactive primitives crash
- draw/gs: add more info to debugging.
- gallivm/nir: add group barrier support
- llvmpipe: fix subpixel bits reporting.
- gallivm/format: convert unsigned values to float properly.
- gallivm/conv: enable conversion min code. (v2)
- gallivm/sample: fix texel type for stencil 8-bit
- llvmpipe/setup: add planes for draw regions if no scissor.
- gallivm/cache: don't require a null terminator for cache data.
- mesa/gles3: add support for GL_EXT_shader_group_vote
- virgl: change vendor id to reflect reality more.
- llvmpipe: change vendor to be more generic.
- softpipe: change vendor name to something more generic.
- gallivm/nir: fix const loading on big endian systems
- glsl: fix constant packing for 64-bit big endian.
- gallivm/nir: fix big-endian 64-bit splitting/merging.
- llvmpipe: fix occlusion queries on big-endian.
- mesa/get: fix enum16 big-endian getting.
- draw/llvm: fix big-endian mask adjusting
- draw: pass nr_samplers into llvm sample state creation.
- llvmpipe: pass number of samplers into llvm sampler code.
- gallivm/sample: change texture function generator api
- gallivm: add indirect texture switch statement builder.
- draw: add support for indirect texture access
- llvmpipe: add support for indirect texture access.
- gallivm/nir: add texture unit indexing
- gallivm/nir: handle non-uniform texture offsets
- gallivm/sample: pass indirect offset into texture/image units
- llvmpipe/draw: wire up indirect offset
- gallivm/sample: handle size unit offset
- llvmpipe: enable ARB_gpu_shader5
- draw: pass number of images to image soa create
- llvmpipe: pass number of images into image soa create
- gallivm/nir: support passing image index into image code.
- gallivm/nir: refactor image operations for indirect support.
- gallivm/img: refactor out the texel return type (v2)

- gallivm/nir: add support for indirect image loading
- draw/sample: add support for indirect images
- llvmpipe: handle indirect images properly
- ci: fixup tests after all indirect images fixes.
- docs: update llvmpipe GL 4.0 status
- draw/clip: cleanup viewport index handling code.
- draw/clip: fix viewport index for geometry shaders
- mesa/version: only enable GL4.1 with correct limits.
- llvmpipe: bump texture/scene limits to enable GL 4.1
- llvmpipe: bump to GL support to GL 4.1
- llvmpipe: enable GL 4.2
- gallivm/nir: call end prim at end on all GS streams.
- draw: emit so primitives before ending empty pipeline.
- draw/gs: fix up current verts in output fetching.
- gallivm/draw/gs: pass vertex stream count into shader build
- draw/gs: only allocate memory for streams needed.
- gallivm/gs_iface: pass stream into end primitive interface.
- gallivm/nir: don't access stream var outside bounds
- gallivm/nir: end primitive for all streams.
- draw: account primitive lengths for all streams.
- draw/gs: reverse the polarity of the invocation/prims execution
- draw: use common exit path in pipeline finish.
- draw: free vertex info from geometry streams.
- draw/gs: use mask to limit vertex emission.
- ci/virgl: update results after streams fixes.
- llvmpipe: add ARB_post_depth_coverage support.
- llvmpipe: denote NEW fs when images change.
- llvmpipe: flush resources on sampler view binding
- llvmpipe/cs: fix image/sampler binding for compute
- nouveau: avoid LTO ODR warning (v2)
- gallivm/sample: always square rho before fast log2
- llvmpipe/format: fix snorm conversion
- mesa: change dsa texture error codes for GL 4.6
- ci: bump piglit checkout for dsa tests
- llvmpipe: fix stencil only formats.
- llvmpipe: fix position offset interpolation

- llvmpipe/cs: respect render condition
- llvmpipe: add framebuffer fetching support (v1.1)
- ci/llvmpipe: reenable gpu shader5 tests
- llvmpipe: enable EXT_texture_shadow_lod
- llvmpipe/draw: handle constant buffer limits and robustness (v1.1)
- drisw: add robustness extension support.
- glx/drisw: add robustness support
- llvmpipe: add device reset query context hook.
- llvmpipe: enable robust buffer access + GL 4.3, GLES 3.2 and robust buffer access behaviour
- llvmpipe/ms: fix sign extension bug in rasterizer.
- Revert "llvmpipe: Use the default behavior of ALLOW_MAPPED_BUFFERS."
- radv: cleanup locking around timeline waiting.
- llvmpipe: only read 0 for channels being read
- llvmpipe/blit: for 32-bit unorm depth blits just copy 32-bit
- llvmpipe: enable GL 4.5
- llvmpipe/cs: update compute counters not fragment shader.
- llvmpipe: include gallivm perf flags in shader cache.
- gallivm: disable brilinear for lod bias and explicit lod.

David McFarland (1):

- radv: link with ld_args_build_id

David Stevens (2):

- nir: Add colorspace support to YUV lowering pass
- i965/i915: Add colorspace support to YUV sampling

Denys (1):

- gitlab: Ask about reproduction rate in the issue template

Dmitriy Nester (8):

- mesa: check draw buffer completeness on glClearBufferfv/glClearBufferuiv
- nir: replace fnv1a hash function with xxhash
- freedreno: replace fnv1a hash function with xxhash
- i965: replace fnv1a hash function with xxhash
- util/hash_table: replace fnv1a hash function with xxhash
- r600: replace fnv1a hash function with xxhash
- zink: replace fnv1a hash function with xxhash
- util: delete fnv1a hash function

Duncan Hopkins (1):

- zink. Changed sampler default name.

Dylan Baker (41):

- docs: Add release notes for 20.0.6
- docs: Add SHA256 sums for 20.0.6
- docs: update calendar, add news item, and link releases notes for 20.0.6
- docs: Add release notes for 20.0.7
- docs/relnotes Add sha256 sums to 20.0.7
- docs: update calendar, add news item, and link releases notes for 20.0.7
- tests: Make tests aware of meson test wrapper
- meson: Bump required version to 0.52.0
- meson: Use the check_header function
- meson: Use build_always_stale instead of build_always
- meson: Use builtins for checking gnu __attributes__
- drm-shim/meson: The name of the target is a string not a list
- drm-shim/meson: Use portable override_options for setting C standard
- meson: use gnu_symbol_visibility argument
- meson: use 2 space not 3 space indent
- meson: deprecated 'true' and 'false' in combo options for 'enabled' and 'disabled'
- vulkan-overlay/meson: use install_data instead of configure_file
- docs: Add release notes for 20.0.8
- docs: Add sha256sums for 20.0.8
- docs: update calendar, add news item, and link releases notes for 20.0.8
- mesa/swrast: use logf2 instead of util_fast_log2
- VERSION: bump for 20.2.0-rc1
- .pick_status.json: Update to 9333a8570d2174b73da63c3ee6f1a740ae487ab8
- .pick_status.json: Update to 1e28745bc0d3528c1dfc25459456849feb58d407
- meson/freedreno: Fix lua requirement
- .pick_status.json: Update to fdb97d3d2914c8f887a7968432db4fdbd35d8376
- bump version for 20.2.0-rc2
- .pick_status.json: Update to 61042b1bdb199f98dd34085ed29a8c492ed9b2a3
- .pick_status.json: Update to 6d28270968e0728bf8bdf48a6abd261c50d9ef07
- .pick_status.json: Update to ca7d66e847d08914cec0a5e003b400da9c0a2695
- VERSION: bump for 20.2.0-rc3
- .pick_status.json: Update to 7fbded8b5821a47c26245b181446f972f920a96e
- .pick_status.json: Mark e93979ba599355c42df01a89073362b970489a3a as denominated
- .pick_status.json: Update to b9927c8c8d0c105699306a68773c015930ff9509
- VERSION: bump for 20.2.0-rc4

- .pick_status.json: Update to ef980ac0c1cd65993ba0c1d20e1c09b45bfef99d
- fix: gallivm: disable brilenear for lod bias and explicit lod.
- .pick_status.json: Update to a1f46d7b6943699e5efb60fbcfdd1450db85adb1
- amd/ac_surface: convert tabs to 3 spaces
- .pick_status.json: Update to 90b98c06493f8a9759e5496d5ec91fb60edf7b92
- .pick_status.json: Update to 472a20c5fc0feda0f074b4ff95fd7c7a6305c8cd

Eduardo Lima Mitev (2):

- freedreno: Centralize UUID generation into new files freedreno_uuid.c/h
- freedreno/uuid: Generate meaningful device and driver UUID

Elie Tournier (12):

- virgl: implement ARB_clear_texture
- virgl: Enable CAP_CLEAR_TEXTURE if host supports it
- docs/features: Add ARB_clear_texture to virgl
- gallium: add TGSI_PROPERTY_FS_BLEND_EQUATION_ADVANCED
- glsl_to_tgsi: Set TGSI_PROPERTY_FS_BLEND_EQUATION_ADVANCED
- virgl: Reserved last caps of capability_bits
- gallium: Add PIPE_CAP_BLEND_EQUATION_ADVANCED
- st: expose KHR_blend_equation_advanced if PIPE_CAP_BLEND_EQUATION_ADVANCED
- glsl_to_ir: do lower_blend_equation if PIPE_CAP_FBFETCH
- virgl: Use alpha_src_factor to store blend_equation_advenced value
- virgl: Encode barrier for blend_equation_advanced
- virgl: set PIPE_CAP_BLEND_EQUATION_ADVANCED

Emmanuel (3):

- meson: Do not enable USE_ELF_TLS for FreeBSD
- iris: Explicitly cast value to uint64_t
- i965: Explicitly cast value to uint64_t

Emmanuel Gil Peyrot (2):

- util/rand_xor: use getrandom() when available
- Expose EGL_KHR_platform_* when EXT is supported

Emmanuel Vadot (1):

- meson: Add versioning for xvmc tracker

Eric Anholt (228):

- freedreno/ir3: Initialize the unused dwords of the immediates consts.
- freedreno/ir3: Drop redundant IR3_REG_HALF setup in ALU ops.
- freedreno/ir3: Leave bools as 1-bit, storing them in full regs.
- freedreno/ir3: Set up the block predecessors for a3xx TF

- freedreno/ir3: Fix the a3xx TF outputs stores.
- freedreno/ir3: Fix register allocation assertion failures.
- freedreno: Stop doing binning shaders other than the VS in shader-db.
- freedreno/ir3: Skip tess epilogue if the program is missing stores.
- freedreno: Fix assertion failures on GS/tess shaders with shader-db enabled.
- freedreno/ir3: Remove unused half precision shader key flag.
- freedreno: Emit debug messages when doing draw-time recompiles of shaders.
- freedreno/ir3: Improve shader key normalization.
- freedreno/ir3: Stop initializing regid of so->outputs during setup.
- freedreno/ir3: Set up outputs for multi-slot varyings.
- freedreno: Immediately compile a default variant of shaders.
- freedreno/ir3: Set the FS .msaa flag to true during precompiles.
- freedreno/ir3: Add some more tests of cat6 disasm.
- freedreno/ir3: Sync some new changes from envytools.
- freedreno/ir3: Define the bindful uniform/nonuniform desc modes for cat6 a6xx.
- freedreno/ir3: Disable sin/cos range reduction for mediump.
- ci: Clean up setup of the job-specific env vars in baremetal testing.
- ci: Enable IRC flake reporting on freedreno baremetal boards.
- ci: Improve the flakes reports on IRC.
- ci: Fix the nick used in IRC reporting.
- freedreno: Deduplicate ringbuffer macros with computerator/fdperf
- freedreno: Clean up tests around ORing in the reloc flags.
- freedreno: Rename append_bo() in case it doesn't get inlined.
- freedreno: Initialize the bo's iova at creation time.
- freedreno: Start moving relocs flags into the BOs.
- freedreno: Replace OUT_RELOCD with permanently flagging shader BOs for it.
- freedreno: Mark all ringbuffer BOs as to be dumped on crash.
- freedreno: Tell the kernel that all BOs are for writing.
- freedreno: Replace OUT_RELOCW with OUT_RELOC.
- freedreno: Drop the "write" arg to emit_const_bo now relocs don't care.
- nir: Fix count when we didn't lower load_uniforms but did shift load_ubos.
- freedreno: Fix non-constbuf-upload UBO block indices and count.
- freedreno: Add a nohw flag to skip submitting to the kernel.
- freedreno: Split the fd_batch_resource_used by read vs write.
- freedreno: Add an early out for preparing to read a resource.
- freedreno: Move the resource_read early out to an inline.

- freedreno: Skip taking the lock for resource usage if it's already flagged.
- freedreno/a4xx+: Increase max texture size to 16384.
- freedreno/a6xx: Improve layout testcase logging for UBWC fails.
- freedreno/a6xx: Add a testcase for UBWC buffer sharing.
- freedreno: Pull the tile_alignment lookup for a layout to a helper.
- freedreno/a6xx: Fix UBWC blockheight for RG8.
- freedreno/a6xx: Fix UBWC mipmap sizing.
- freedreno/a6xx: Fix UBWC mipmapping height alignment.
- nir: Include num_ubos in the printed shader (if nonzero).
- freedreno/ir3: Clean up a silly nir_src_for_ssa(src.ssa).
- freedreno/ir3: Leave the cursor alone during ir3_nir_try_propagate_bit_shift.
- freedreno/ir3: Move i/o offset lowering after analyze_ubo_ranges.
- freedreno: Trim num_ubos to just the ones we haven't lowered to constbuf.
- freedreno/a6xx: Use LDC for UBO loads.
- freedreno: Drop the noubo fails list for CI, since there aren't any now.
- freedreno: Fix attempts to push UBO contents past the constlen on pre-a6xx.
- freedreno: Fix resource layout dump loop.
- freedreno: Avoid duplicate BO relocs in FD_RINGBUFFER_OBJECTs.
- ci: Move cross file generation to a shared script.
- ci: Autodetect whether we need cross setup in lava_arm builds.
- ci: Make cmake toolchain file for deqp cross build setup.
- ci: Make the create-rootfs more resilient.
- ci: Update versions of packages to remove from rootfses.
- ci: Switch the baremetal runner to be an x86 docker image.
- ci: Disable SMP on the a5xx boards.
- ci: Make a530's GLES3/31 fractional runs much more complete.
- freedreno/a5xx: Move resource layout to fdl.
- freedreno/fdl: Separate the list of a6xx testcases from the the test code.
- freedreno/a5xx: Add the outline of a unit test for a5xx layout.
- freedreno/a5xx: Set MIN_LAYERSZ on 3D textures like we do on a6xx.
- freedreno/a5xx: Define the 2D blit UBWC pitch fields
- ci: Fix DEQP_CASELIST_FILTER (used by a630 noubo run)
- ci: Do an explicit NIR validation-enabled pass on freedreno a630.
- ci: Don't forget to set NIR_VALIDATE in baremetal runs.
- ci: Enable a fractional run with UBO-to-constbuf disabled on a3xx.
- ci: Improve baremetal's logging of the job env var passthrough.

- freedreno/a6xx: Fix the size of buffer image views.
- freedreno: Fix printing of unused src in disasm of cat6 RESINFO.
- freedreno: Add more resinfo/ldgb testcases.
- freedreno: Fix resinfo asm, which doesn't have srcs besides IBO number.
- freedreno: Set the immediate flag in a4/a5xx resinfos.
- freedreno/ir3: Refactor out IBO source references.
- freedreno/ir3: Move handle_bindless_cat6 to compiler_nir and reuse.
- freedreno/ir3: Use RESINFO for a6xx image size queries.
- ci: Drop double ".txt" suffix on the unexpected results file.
- ci: Drop old comment about enabling –deqp-watchdog.
- ci: Auto-detect the architecture for VK ICD filenames.
- ci: Add DEQP_EXPECTED_RENDERER support for VK tests.
- ci: Move baremetal DEQP_NO_SAVE_RESULTS setup to the yml.
- ci: Quick exit qpa extraction for non-matching qpas.
- ci: Disable the firmware loader user helper option in arm64 kernels.
- ci: Build a cheza kernel.
- ci: Add scripts for controlling bare-metal chezas.
- ci: Switch cheza (freedreno a630) testing to baremetal.
- ci: Don't build an arm_test container now that the last user is gone.
- ci: Rename x86_cross_arm_test to just arm_test.
- turnip: Move vertex buffer bindings to SET_DRAW_STATE.
- turnip: Don't bother clamping VB size.
- turnip: Simplify vertex buffer bindings.
- turnip: Use tu_cs_emit_regs() for BLEND_CONTROL.
- turnip: Add support for alphaToOne.
- freedreno/a6xx: Add support for ALPHA_TO_ONE.
- freedreno: Upload gallium constbufs as needed when referenced as a UBO.
- freedreno/ir3: Refactor ir3_cp's lower_immed().
- freedreno/ir3: Stop pushing immediates once we've filled the constbuf.
- freedreno/ir3: Drop unnecessary alignment of pushed UBO size.
- freedreno/ir3: Stop shifting UBO 1 down to be UBO 0.
- freedreno/ir3: Account for driver params in UBO max const upload.
- freedreno/ir3: Drop the max_const on a6xx to 512.
- freedreno/ir3: Handle cases where we decide not to lower UBO 0 loads.
- turnip: Fix crashes in compute with no descriptors to load.
- ci: Bump up to the current version of the VK CTS.

- ci: Disable shader cache on vulkan CI runs.
- ci: Build the full VK CTS for baremetal testing.
- ci: Enable pre-merge fractional vulkan CTS runs on the turnip driver.
- ci: Use rsync for initial nfsroot population on cheza.
- turnip: Expose robustBufferAccess.
- freedreno/a6xx: Fix clip_halfz support.
- ci: Leave a note as to what might be going on with a test.
- ci: Fix weird filesystem globs appearing in failed test .qpa files.
- ci: Disable some flaky tests on turnip.
- ci/bare-metal: Reword the final output of the init script on the board.
- ci/bare-metal: Make which test to run configurable.
- ci/bare-metal: Use the deqp-runner bits straight out of the artifacts.
- ci/bare-metal: Stop fetching the git tree.
- ci/bare-metal: Terminate the job with an error on kernel panic.
- docs: Replace ancient swrast conformance docs with more current information.
- docs: Add dri-devel to the mailing lists and drop the DRI wiki link.
- ci: disable the windows tests until the runner can be stabilized again
- ci: Bump vulkan CTS to 1.2.3.0.
- ci: Enable NIR validation on a630 GLES2 and VK tests.
- ci/bare-metal: Skip setting of unset variables at startup.
- ci/bare-metal: Don't include dev packages in arm*test.
- ci/tracie: Print the path if the trace isn't found.
- ci/tracie: Fix apitrace dump using "less" which isn't in the ARM rootfs.
- ci: Add a freedreno a630 tracie run.
- freedreno/a6xx: Define the register fields for polygon fill mode.
- turnip: Add support for polygon fill modes.
- freedreno/a6xx: Add support for polygon fill mode (as long as front==back).
- ci: Remove a stray "always" on the freedreno traces job.
- ci/bare-metal: Fail early when we get stuck powering on a cheza.
- ci/baremetal: Bump the kernel to a recent drm-msm-fixes for msm semaphores.
- turnip: Do better TU_DEBUG=startup logging of drmGetDevices2() failure.
- turnip: Fix error handling of DRM_MSM_GEM_INFO ioctls.
- turnip: Properly return VK_DEVICE_LOST on queuesubmit failures.
- gallium/util: Add a helper function for point sprite handling.
- vc4: Enable PIPE_CAP_TGSI_TEXCOORD.
- v3d: Enable PIPE_CAP_TGSI_TEXCOORD.

- v3d: Fix -Wmaybe-uninitialized compiler warning in the v33 code.
- ci: Disable pixmark-piano trace on a630 due to GPU hangs.
- util: Avoid strict aliasing bugs in xxhash.
- util: Mark util_format_description() as a const function.
- softpipe: Clean up softpipe's SSBO load/store interpreting instructions.
- util: Remove unused util_format_planar_is_supported().
- etnaviv: Use the util_pack_color_union() helper.
- gallium/util: Fix location of the comment about S8_UINT handling.
- gallium/util: Clean up the Z/S tile write path.
- gallium/util: Move the Z/S handling to the outside of get_tile().
- svga: Reuse util_format_unpack_rgba().
- util: Merge util_format_write_4* functions.
- util: Merge util_format_read_4* functions.
- util: Use designated initializers to clean up the format tables' pack/unpack.
- llvmpipe: Generalize "could llvmpipe fetch this format" check in unit testing.
- util: Remove the stub pack/unpack functions for YUV formats.
- util: Share a single function pointer for the 4-byte rgba unpack function.
- docs: Move the current CI .rst doc to docs/ci/ and link to it from .gitlab-ci.
- docs: Move the conformance and the CI docs to a top level Testing section.
- docs: Move the gitlab-ci docs to RST.
- docs: Relax the expectations of HW CI farms.
- docs: Document how to interact with docker containers.
- freedreno/ir3_cmdline: Fix an uninit var warning.
- freedreno/ir3: Fix uninit var warning.
- intel: Fix release-build warnings about sf_entry_size.
- intel/perf: Fix unused var warning in release builds.
- intel/perf: Move perf query register programming to static tables.
- freedreno/a2xx: Fix compiler warning in disasm.
- meson: Enable GCing of functions and data from compilation units by default.
- freedreno/ir3: Fix duplicated fine derivatives instructions.
- freedreno/ir3: Add unit tests for derivatives disasm.
- ci: Use FDO_CI_CONCURRENT as our -j flags when present in the runner env.
- freedreno/ir3: Add a note about the instructions in the disasm test.
- freedreno/ir3: Add a bunch more tests for cat6 opcodes.
- freedreno/ir3: Refactor cat6 general dst printing.
- freedreno/ir3: Fix disasm of register offsets in ldp/stp.

- freedreno/ir3: Add missing ld_args_build_id to the ir3_delay unit test.
- ci: Set XDG_CACHE_HOME to tmpfs for bare-metal runners to avoid NFS.
- ci: Update checksums for freedreno traces.
- llvmpipe: Remove a bunch of default handling of pipe caps.
- llvmpipe: Use the default behavior of ALLOW_MAPPED_BUFFERS.
- softpipe: Remove a bunch of default handling of pipe caps.
- softpipe: Use the default behavior of ALLOW_MAPPED_BUFFERS.
- virgl: Remove a bunch of default handling of pipe caps.
- swr: Remove a bunch of default handling of pipe caps.
- swr: Use the default behavior of ALLOW_MAPPED_BUFFERS.
- svga: Remove a bunch of default handling of pipe caps.
- i915: Remove a bunch of default handling of pipe caps.
- softpipe: Refactor pipe_shader_state setup.
- softpipe: Convert to comma-separated SOFTPIPE_DEBUG for debug options.
- softpipe: Add support for reporting shader-db output.
- softpipe: Enable PIPE_CAP_TGSI_TEXCOORD.
- softpipe: Enable PIPE_CAP_TGSI_ANY_REG_AS_ADDRESS;
- ci/bare-metal: Capture the first devcoredump a job produces.
- drm-shim: Return -EINVAL instead of abort()ing on unknown ioctls.
- docs: Explain how to set up a personal gitlab runner.
- nir: Add a pass to cut the trailing ends of vectors.
- i965: Enable vector shrinking in the vec4 backend.
- amd: Swap from nir_opt_shrink_load() to nir_opt_shrink_vectors().
- nir: Remove the old nir_opt_shrink_load.
- freedreno: Fix “Offset of packed bitfield changed” warnings:
- nir/lower_amul: Use num_ubos/ssbos instead of recomputing it.
- nir: Add a little more docs about NIR’s constant_data.
- nir: Print the constant data size associated with a shader.
- freedreno/ir3: Fix the type of half-float indirect uniform loads.
- freedreno/a6xx: Document the bit for the magic 32bit-uniforms-as-16b mode.
- freedreno/computerator: Set SP_MODE_CONTROL to the same value as vulkan/GL
- freedreno/ir3: Merge the redundant immediate_idx/immediates_count fields
- freedreno/ir3: Simpify the immediates from an array of vec4 to array of dwords.
- freedreno: Rename emit_const_bo() to emit_const_ptrs().
- freedreno: Split ir3_const’s user buffer and indirect upload APIs.
- freedreno/ir3: Clean up instrlen setup.

- freedreno: Increase the NUM_UNIT on compute's consts in indirect dispatch.
- freedreno: Add more asserts for DST_OFF/NUM_UNIT in indirect const uploads.
- freedreno/ir3: Fix assertion failures dumping CS high full regs.
- turnip: Make sure we include the build id.
- gallium/tgsi_exec: Fix up NumOutputs counting
- freedreno: Make the pack struct have a .qword for wide addresses.
- turnip: Fix truncation of CS shader iovas to 32 bits.
- turnip: Fix truncation of iovas to 32 bits in queries.

Eric Engestrom (146):

- cut 20.1 branch
- docs: update calendar for 20.1.0-rc2
- post_version.py: fix branch name construction for release candidates
- post_version.py: invert *is_point* into *is_first_release* to make its purpose clearer
- post_version.py: stop adding release candidates to the index and relnotes
- docs: update calendar for 20.1.0-rc3
- gitlab-ci: exclude scripts that don't affect the build
- util/rand_xor: make it clear that {,s_}rand_xorshift128plus take *exactly 2* uint64_t
- util/rand_xor: drop unused header
- util/rand_xor: fallback Linux to time-based instead of fixed seed
- util/rand_xor: extend the urandom path to all non-Windows platforms
- docs: update calendar for 20.1.0-rc4
- anv: pass the fd directly to anv_gem_reg_read()
- anv: replace magic | *1* with already #define'd name
- anv: disable VK_EXT_calibrated_timestamps when the timestamp register is unreadable
- git_sha1_gen.py: fix out-of-date comment
- git_sha1_gen.py: fix code style
- git_sha1_gen.py: fix whitespace
- compiler: delete leftover autotools test wrapper
- no_extern_c.h: fix typo in comment
- tree-wide: fix deprecated GitLab URLs
- docs: drop no-longer-relevant comment about bugzilla
- docs: Add release notes for 20.1.0
- docs: update calendar, add news item, and link releases notes for 20.1.0
- meson: remove "empty array"/"array of an empty string" confusion
- glapi: remove deprecated .getchildren() that has been replace with an iterator
- intel/genxml: drop sort_xml.sh and move the loop directly in gen_sort_tags.py

- intel: fix gen_sort_tags.py
- docs: Add release notes for 20.1.1
- docs: update calendar, add news item, and link releases notes for 20.1.1
- v3d: add missing unlock() in error path
- intel/genxml: drop python 2 support for gen_sort_tags.py
- intel/genxml: replace gen_sort_tags.py MIT licence with SPDX equivalent
- docs: update the blocks of unused EGL enums assigned to us
- i965: drop dead #include "config.h"
- iris: drop dead #include "config.h"
- gen_release_notes.py: update script to the new rST way of things
- post_version.py: update script to the new rST way of things
- intel/tools: rewrite run-test.sh in python
- intel/tools: make test aware of the meson test wrapper
- khronos-update.py: add script to simplify update of Khronos headers & xml files
- docs: remove plain-text copy of versions.rst
- util/os_file: replace broken windows-detection code with detect_os.h
- util: introduce os_dupfd_cloexec() helper
- replace all F_DUPFD_CLOEXEC with os_dupfd_cloexec()
- vulkan/wsi: replace all dup() with os_dupfd_cloexec()
- radv: replace all dup() with os_dupfd_cloexec()
- anv: replace all dup() with os_dupfd_cloexec()
- iris: replace all dup() with os_dupfd_cloexec()
- i965: replace all dup() with os_dupfd_cloexec()
- egl: replace all dup() with os_dupfd_cloexec()
- etnaviv: replace all dup() with os_dupfd_cloexec()
- freedreno: replace all dup() with os_dupfd_cloexec()
- svga: replace all dup() with os_dupfd_cloexec()
- virgl: replace all dup() with os_dupfd_cloexec()
- docs: publish our release maintainers' keys
- docs: remind release maintainers to sign the tarballs and publish their key
- docs: suggest alternative installation methods for meson
- docs: stop considering *Cc: mesa-stable* as an email address
- docs: reword "sending a patch revision" to "updating a merge request"
- docs: drop *git sendemail* instructions
- docs: prefer *Fixes:* over *Cc: mesa-stable*
- docs: add some formatting to the "backport merge request" option

- docs: reword a sentence a bit
- docs: make it clear that the tags needs to be in the commit message
- docs: move *Fixes:* tag explanation to its own section
- docs: move "stable" tag explanation next to *Fixes:*
- driconf: drop 28% catalan translation
- driconf: drop 15% german translation
- driconf: drop 26% spanish translation
- driconf: drop 6% french translation
- driconf: drop 8% dutch translation
- driconf: drop 9% swedish translation
- driconf: drop now unused translation facility
- util: rename xmlpool.h to driconf.h
- gitlab-ci: drop gettext from the build images
- docs: drop deleted file from extra sphinx files
- docs: cat maintainer keys to a single file
- docs: add some padding to the release calendar
- docs: add planning for 20.2
- bin/symbols-check: explain C++ symbols workaround
- docs: Add release notes for 20.1.2
- docs: update calendar and link releases notes for 20.1.2
- docs: fix 20.1.2 relnotes
- docs: add a page explaining the GitLab CI and the Intel CI
- mesa/glformats: make _mesa_gles_error_check_format_and_type() more consistent
- docs: add release notes for 20.1.3
- docs: update calendar and link releases notes for 20.1.3
- docs: fix a bunch of typos
- egl: always compile surfaceless
- vulkan: automatically compile the *display* platform when available
- meson: move xlib-lease block further down
- egl: automatically compile the *drm* platform when available
- introduce *commit_in_branch.py* script to help devs figure this out
- bin/gen_release_notes.py: drop new_features.txt when we release XX.Y.0
- egl/wayland: add missing newline between functions
- glx: drop always-true #ifdef
- docs/submittingpatches: add more than one *Cc: mesa-stable* example to the examples list
- meson/intel: add missing dep on git_sha1.h

- meson: fix android vulkan build
- egl: inline fallback for create_pixmap_surface
- egl: inline fallback for create_pbuffer_surface
- egl: drop unused fallback function
- egl: inline fallback for swap_buffers_with_damage
- egl: inline fallback for swap_buffers_region
- egl: inline fallback for post_sub_buffer
- egl: inline fallback for copy_buffers
- egl: inline fallback for query_buffer_age
- egl: inline fallback for create_wayland_buffer_from_image
- egl: inline fallback for get_sync_values
- egl: drop now empty egl_dri2_fallbacks.h
- egl: mark the rest of the callbacks as mandatory or optional
- egl: inline _EGLAPI into _EGLDriver
- docs: add release notes for 20.1.4
- docs: update calendar and link releases notes for 20.1.4
- post_version.py: don't generate relnotes twice
- post_version.py: drop incorrect conf.py changes
- post_version.py: stop using non-existent functions and fix commit message
- post_version.py: update the files in the current worktree, not the one with the script that we run
- post_version.py: fix relnotes links
- bin/gen_release_notes: automatically commit release notes
- docs/releasing: improve wording
- bin/khronos-update: having a folder in include/ is not a requirement
- bin/khronos-update: add support for the SPIRV files
- bin/khronos-update: add workaround for python bug 9625
- egl: replace _eglInitDriver() with a simple variable
- egl: drop unnecessary _eglGetDriver()
- egl: fix _eglMatchDriver() return type
- egl: inline _eglMatchAndInitialize() and refactor _eglMatchDriver()
- egl: rename _eglMatchDriver() to _eglInitializeDisplay()
- egl: drop left-over function prototype
- egl: const _eglDriver
- egl/haiku: drop overwritten preset of EGL version
- egl: consistently use dri2_egl_display() helper macro
- meson: fix *-D xlib-lease=auto* detection

- docs: add release notes for 20.1.5
- docs: update calendar and link releases notes for 20.1.5
- pick-ui: specify git commands in "resolve cherry pick" message
- egl/entrypoint-check: split sort-check into a function
- egl/entrypoint-check: add check that GLVND and plain EGL have the same entrypoints
- driconf: fix force_gl_vendor description
- meson: bump required glvnd version
- egl/x11_dri3: enable & require xfixes 2.0
- egl/x11_dri3: implement EGL_KHR_swap_buffers_with_damage
- meson: don't advertise TLS support if glx wasn't build with it
- meson: drop leftover PTHREAD_SETAFFINITY_IN_NP_HEADER

Erico Nunes (16):

- lima/ppir: introduce liveness internal live set
- lima/ppir: fix lod bias register codegen
- lima/ppir: do not assume single src for pipeline outputs
- lima/ppir: combine varying loads in node_to_instr
- lima/ppir: duplicate intrinsics in nir
- lima/ppir: duplicate consts in nir
- lima/ppir: remove unused clone functions
- lima/ppir: rework emit nir to ppir
- lima/ppir: rework store output
- lima/ppir: add fallback mov option for const scheduler
- lima/ppir: rework select conditions
- lima/ppir: handle failures on all ppir_emit_cf_list paths
- lima/ppir: improve handling for successors in other blocks
- lima/ppir: rework tex lowering
- lima/ppir: optimize tex loads with single successor
- lima/ppir: use a ready list in node_to_instr

Erik Faye-Lund (124):

- compiler/nir: move tan-calculation to helper
- vtn/opencl: add native_tan-support
- vtn/opencl: native variants of sin/cos
- vtn/opencl: native divide support
- vtn/opencl: native powr support
- vtn/opencl: native recip support
- vtn/opencl: native rsqrt support

- vtn/opencl: native sqrt support
- compiler/glsl: explicitly store NumUniformBlocks
- mesa/st: consider NumUniformBlocks instead of num_ubos when binding
- zink: use nir_lower_uniforms_to_ubo
- zink: lower b2b to b2i
- util/os_memory: never use os_memory_debug.h
- st/wgl: pass st_context_iface into stw_st_framebuffer_present_locked
- st/wgl: allocate and resolve msaa-textures
- docs/features: add zink features
- zink: load vk_GetMemoryFdKHR while creating screen
- zink: add a GET_PROC_ADDR macro to simplify load_device_extensions
- docs/features: mark GL_NV_conditional_render as done for zink
- zink: disable vkCmdResolveImage when respecting render-condition
- zink: do not expose real value for PIPE_CAP_MAX_VIEWPORTS
- zink: correct PIPE_SHADER_CAP_MAX_SHADER_IMAGES
- zink: mark depth-component cube-maps as done
- zink: implement i2b1
- docs: fix broken release-calendar
- zink: hammer in an explicit wait when retrieving buffer contents for reading
- zink: use samples from state
- zink: do not dig into resource for nr_samples
- zink: pass batch instead of context for queries
- zink: implement nir_texop_txf_ms
- zink: expose PIPE_CAP_TEXTURE_MULTISAMPLE
- docs/features: mark GL_ARB_texture_multisample as done for zink
- zink: use general-layout when blitting to/from same resource
- zink: Use store_dest_raw instead of storing an uint
- nir: reuse existing psiz-variable
- zink: emulate B8G8R8X8_SRGB with B8G8R8A8_SRGB
- zink: assert that image-view format isn't undefined
- zink: only report device-local memory as video-memory
- gallium/hud: do not specify potentially invalid depth-range
- TEMP: add rst-conversion scripts
- docs: convert articles to reructuredtext
- TEMP: remove rst-conversion scripts
- docs: delete no longer needed file

- docs: fixup botched table
- docs: escape double colons
- docs: escape asterisks
- docs: escape trailing underscores properly
- docs: fixup broken rst
- docs: fixup heading-levels
- docs: use sphinx
- docs: disable syntax-highlighting by default
- docs: use code-block with caption instead of table
- docs: format notes as rst-notes
- docs: use code-blocks
- docs: drop open-coded toc for articles
- docs: add xlibdriver to table-of-contents
- docs: do not copy source-files to site
- docs: use rst footnotes instead of manual ones
- docs: reformat license table as rst table
- docs: use rst-note for highlighted text
- docs: bundle extra files
- docs: include specs into the generated docs
- gitlab-ci: build and deploy docs
- docs: drop news in favour of the introduction as index-page
- README: update references to internal docs
- docs: update internal references
- docs/relnotes: update internal references
- radv: update internal reference
- bin/perf-annotate-jit.py: update internal reference
- docs/release-calendar: restore missing id
- nir: do not try to merge xfb-outputs
- Revert "gallium/hud: don't use user vertex buffers"
- gallium/hud: don't use user vertex buffers
- zink: enable cull-distance if supported
- zink: expose GLSL 1.30
- docs: update internal references
- docs/relnotes: update internal references
- docs: fixup relnotes after rst-conversion
- docs/features: mark GL3 as complete for zink

- docs/features: update ARB_texture_buffer_object line
- docs/features: remove driver-list for forward-compatible context
- mesa/main: fix inverted condition
- gallium/os: call "ANSI" version of GetCommandLine
- graw/gdi: do not depend on UNICODE macro
- gallium/util: limit STACK_LEN on Windows
- gallium/util: add missing include
- docs: update favicon
- docs: remove non-existent reference
- docs: restore accidentally dropped labels
- docs: fix internal references
- docs: use ref-links for internal references
- gallium/docs: update to recent sphinx
- gallium/docs: fixup formatting of numbered lists
- gallium/docs: remove reference to non-existent label
- gallium/docs: use none for highlight_language
- gallium/docs: prefix exts dir with underscore
- gallium/docs: remove non-existent static dir
- gallium/docs: remove unused imgmath extension
- ci: only build docs in the upstream-repo
- ci: only build docs if any docs changed
- ci: test docs for non-master builds
- ci: move deploy-stage later in the pipeline
- ci: move test-docs to container stage
- ci: add graphviz to the .docs-base template
- merge gallium docs into main docs
- docs: clean up gallium index-file
- docs: add an extension to generate redirects
- docs: move gallium specific docs into gallium folder
- docs: use svg for graphviz output
- docs: fixup envvar output
- zink: expose depth-clip if supported
- mesa/main: factor out one-time-init into a helper
- mesa/main: use call_once instead of open-coding
- gallium/util: do not use _MTX_INITIALIZER_NP on Windows
- mesa/main: use p_atomic_inc_return instead of locking

- mesa: do not use bitfields for advanced-blend state
- mesa: treat Color._AdvancedBlendMode as enum
- zink: use ralloc in nir-to-spirv
- zink: use ralloc for plain malloc-calls
- zink: pass mem_ctx to ralloc_size-call
- zink: use ralloc for spirv_builder as well
- mesa/program: fix shadow property for samplers
- docs: add some very basic documentation about zink
- mesa: handle GL_FRONT after translating to it

Francisco Jerez (23):

- intel/ir: Update performance analysis parameters for memory fence codegen changes.
- iris: Simplify iris_batch_prepare_noop().
- iris: Extend iris_context dirty state flags to 128 bits.
- iris: Add batch-local synchronization book-keeping to iris_bo.
- iris: Add infrastructure to partition batch into sync boundaries.
- iris: Bracket batch operations which access memory within sync regions.
- iris: Annotate all BO uses with domain and sequence number information.
- iris: Drop redundant iris_address::write flag.
- iris: Report use of any in-flight buffers on first draw call after sync boundary.
- iris: Introduce cache coherency matrix for batch-local memory ordering.
- iris: Update cache coherency matrix on PIPE_CONTROL.
- iris: Implement buffer-local memory barrier based on cache coherency matrix.
- iris: Insert buffer barrier in existing cache flush helpers.
- iris: Remove batch argument of iris_resource_prepare_access() and friends.
- iris: Perform compute predraw flushes from compute batch.
- iris: Remove depth cache set tracking and synchronization.
- iris: Remove render cache hash table-based synchronization.
- iris: Open-code iris_cache_flush_for_read() and iris_cache_flush_for_depth().
- iris: Emit single render target flush PIPE_CONTROL on format mismatch.
- iris: Remove iris_flush_depth_and_render_caches().
- OPTIONAL: iris: Perform BLORP buffer barriers outside of iris_blorp_exec() hook.
- iris/icl+: Report same caching domain as main surface for clear color BO.
- intel/ir/gen12+: Work around FS performance regressions due to SIMD32 discard divergence.

Frank Binns (2):

- docs: change “Fixes:” tag example to match git fixes output
- egl/dri2: only take a dri2_dpy reference when binding a new context/surfaces

Frédéric Bonnard (2):

- clover: Fix types collision between c++ and altivec
- meson: Revert commit overriding C++ standard with gnu++11 on ppc64el

Gert Wollny (66):

- r600: Annotate some case fallthroughs
- r600: remove unused static functions
- r600/sb: replace memset by using member initialization/assignment
- r600: remove some unused variables to silence warnings
- r600: Fix warning regarding mixing enums and unsigned in ?: expression
- r600: Fix nir compiler options, i.e. don't lower IO to temps for TESS
- r600/sfn: Unify semantic name and index query and use TEXCOORD semantic
- r600/sfn: Fix printing vertex fetch instruction flags
- r600: Lower int64 ops from TGSI-to-NIR shaders too
- r600: Lower lerp after tgsi_to_nir
- r600: Add support for loading index register from other than chan X
- r600/sfn: Handle CF index loading from non-X channel
- r600/sfn: rework getting a vector and uniforms from the value pool
- r600/sfn: Skip move instructions if they are only ssa and without modifiers
- r600/sfn: re-use an allocated register in lookup
- r600/sfn: skip copying LOD if the target register is is the same
- r600/sfn: Fix memring print output
- r600/sfn: Fix RING instruction assembly emission
- r600/sfn: Fix GDS assembly emission
- r600/sfn: Fix RAT instruction assembly emission
- r600/sfn: Make allocate_reserved_registers forward to a virtual function
- r600/sfn: Fix handling of output register index
- r600/sfn: Make 3vec loads skip possible moves
- r600/sfn: Add support for viewport index output
- r600/sfn: Take FOGC, and backcolors into account im GS outputs
- r600/sfn: Handle loading sample_pos
- r600/sfn: Add FS output sample_mask
- r600/sfn: Don't reject VARYING_SLOT_PCNT
- r600/sfn: remove pointless check
- r600/sfn: assert when alu dest is missing
- r600/sfn: support indirect sampler buffer reads.
- r600/sfn: Add support for texture_samples

- r600/sfn: use the per shader atomic base
- r600/sfn: SSBO: Fix query of dest components
- r600/sfn: Fix clip vertex output as possible stream variable
- r600/sfn: Fix splitting constants that come from different kcache banks.
- r600/sfn: Don't reorder outputs by location
- r600/sfn: Fix printing ALU op without dest
- r600: Fix duplicated subexpression in r600_asm.c
- r600/sfn: Fix mapping for f32tof64 and f64tof32
- r600/sfn: use modern c++ in printing LDS read instruction
- r600/sfn: Correctly update the number of literals when forcing a new group
- r600/sfn: remove debug output leftover
- nir: lower_tex: Don't normalize coordinates for TXF with RECT
- r600/sfn: lower image derefs
- r600/sfn: Add imageio support
- r600/sfn: Add support for image_size
- r600/sfn: Add support for reading cube image array dim.
- r600/sfn: Take SSBO buffer ID offset into account
- r600/sfn: Handle memory_barrier
- r600/sfn: Add lowering pass for shared IO
- r600/sfn: Add support for shared atomics
- r600/sfn: Don't set num_components on TESS sysvalue intrinsics
- r600/sfn: lower rotate ALU ops
- r600/sfn: Pipe through requesting a register at a given channel
- r600/sfn: emit texture instructions in one block
- r600/sfn: Add option to get a temp value for a specific channel
- r600/sfn: correct handling of loading vec4 with fetching constants
- r600/sfn: Add a forced output swizzle for depth write
- r600/sfn: Fix Ring output swizzle masks
- r600/sfn: Fix default z swizzle for GDS instructions
- r600: Add shader key item to identify when the sample mask should be used
- r600/sfn: Only use sample mask if the according shader key is set
- r600/sfn: Make the pin_to_channel generic
- d600/sfn: write stream outputs to correct mem ring
- gallivm/nir: Lower uniforms to UBOs in llvm draw if the driver didn't request this already

Greg V (1):

- gallium,util: undef ALIGN on FreeBSD to prevent name clash

Guido Günther (2):

- etnaviv: drm: Use NSEC_PER_SEC
- etnaviv: drm: Normalize nano seconds

Gurchetan Singh (1):

- virgl: apply bgra dest swizzle and add Portal 2

Hanno Böck (1):

- Properly check mmap return value

Hyunjun Ko (6):

- freedreno,tu: Don't request fragcoord components not being read.
- tu,radv: fix potentially wrong offset of flexible array.
- vulkan: Adds helpers for vk_object (de)alloation and (de)initialization.
- tu: Fix wrong copies of sampler descriptor.
- turnip: Use the common base object type and struct.
- turnip: implement VK_EXT_private_data

Iago Toral Quiroga (7):

- v3d/compiler: don't rewrite unused temporaries to point to NOP register
- v3d/compiler: fix spill offset
- v3d/compiler: fix image size for 1D arrays
- nir/lower_clip: make the pass compatible with Vulkan semantics
- v3d/compiler: handle compact varyings
- v3d/compiler: request fragment shader clip lowering to be vulkan compatible.
- nir/lower_tex: skip lower_tex_packing for the texture samples query

Ian Romanick (24):

- nir/algebraic: Recognize open-coded byte or word extract from bfe
- nir/algebraic: Split ibfe and ubfe with two constant sources
- nir/algebraic: Optimize some bfe patterns
- nir/algebraic: Optimize ushr of pack_half, not ishr
- nir/algebraic: Add some half packing optimizations for pack_half_2x16_split
- nir/algebraic: Eliminate useless extract before unpack
- i965: Assert that blorp always handles color blits
- meta: Make _mesa_meta_texture_object_from_renderbuffer static
- meta: Make _mesa_meta_setup_sampler static
- meta: Remove support for clearing integer buffers
- mesa: Add matrix utility functions to load matrices
- mesa: Add function to calculate an orthographic projection
- meta: Stop frobbing MatrixMode

- meta: Use same vertex coordinates for GLSL and FF clears
- meta: Coalesce the GLSL and FF paths in meta_clear
- meta: Remove support for multisample blits
- anv/tests: Don't rely on assert or changing NDEBUG in tests
- anv/tests: Silence unused parameter warnings in main
- anv: Silence unused parameter warning in anv_image_get_clear_color_addr
- intel: Silence unused parameter warning in __intel_log_use_args
- intel/drm-shim: Add noop ioctl handler for set_tiling
- intel/drm-shim: Return correct values for I915_PARAM_HAS_ALIASING_PPGTT
- glsl: Remove integer matrix support from ir_dereference_array::constant_expression_value
- nir/algebraic: Don't distrubte absolute-value into dot-products

Icecream95 (78):

- pan/midgard: Fix old style shadows
- panfrost: Fix background showing when using discard
- panfrost: Enable PIPE_CAP_VERTEX_COLOR_UNCLAMPED
- panfrost: Decode AFBC flag bits
- panfrost: Only use AFBC YTR with RGB and RGBA
- pan/midgard: Use a signed value for checking inline constants
- Revert "panfrost: Keep cached BOs mmap'd"
- panfrost: Mark PIPE_BUFFER BOs as not renderable
- pan/mdg: Add a macro for printing instruction source information
- pan/mdg: Move r1.w writeout to branch->dest
- pan/mdg: Remove old zs store lowering
- pan/mdg: Remove old depth writeout code
- pan/mdg: Remove writeout case from bytemask_of_read_components
- nir: Replace the zs_output_pan intrinsic with combined_output_pan
- pan/mdg: Replace writeout booleans with a single value
- pan/mdg: Add new depth writeout code
- pan/mdg: Move search_var to earlier in midgard_compile.c
- pan/mdg: Add depth/stencil support to emit_fragment_store
- pan/mdg: Add new depth store lowering
- pan/mdg: Print writeout sources in mir_print_instruction
- panfrost: Add writes_stencil to the EARLY_Z disable list
- panfrost: Move sampler view bo creation to a separate function
- panfrost: Create a new sampler view bo when the layout changes
- panfrost: Tiled to linear layout conversion

- panfrost: Clean up panfrost_frag_meta_rasterizer_update
- panfrost: Implement ARB_depth_clamp
- pan/decode: Fix helper invocations when tracing
- pan/decode: Add missing wrap modes
- pan/mdg: Fix max_comp calculation for constant printing
- panfrost: RGBA4 and RGB5_A1 framebuffer support
- panfrost: Update sampler views when the texture bo changes
- panfrost: Copy resources when mapping to avoid waiting for readers
- panfrost: Only copy resources when they are in a pending batch
- panfrost: Add PAN_MESA_DEBUG=gl3 flag
- panfrost: Do fine-grained flushing for occlusion query results
- pan/mdg: Vectorize vlut operations
- pan/decode: Make mapped memory read-only while decoding
- nir: Add a base value to load_raw_output_pan
- panfrost: Fix MALI_READS_TILEBUFFER
- pan/mdg: Handle tilebuffer wait loops
- pan/mdg: Use the writeout tag for tilebuffer wait loops
- panfrost: Add rt formats to shader state
- panfrost: Add a bitset of render targets read by shaders
- pan/mdg: Do the pan_lower_framebuffer pass later
- pan/mdg: Emit a tilebuffer wait loop when needed
- pan/mdg: Handle non-blend framebuffer lowering
- pan/mdg: Support MRT in output load lowering
- pan/mdg: Set the z/s store intrinsic base correctly
- pan/mdg: Use a 32-bit ld_color_buffer op when needed
- panfrost: Implement texture_barrier
- panfrost: Stop keying on rt format when using native loads
- panfrost: Use f2fmp for framebuffer lowering conversions
- panfrost: Enable framebuffer fetch
- pan/mdg: Fix non-debug compiliation
- compiler: Add dual-source factors to blend_factor
- gallium: Dual source support in blend_factor_to_shader
- pan/mdg: Add a nir pass to reorder store_output intrinsics
- pan/mdg: Dual source blend input/writeout support
- pan/mdg: Skip z/s combining for dual-source writes
- panfrost: Dual source blend support

- pan/decode: Open the dump file later
- pan/mdg: Don't disassemble blit shaders
- panfrost: Rename lower_store to is_blend in pan_lower_framebuffer
- pan/mdg: Do per-sample framebuffer loads
- panfrost: Do per-sample shading when outputs are read
- nir: Add a face_sysval argument to nir_lower_two_sided_color
- nir: Fix lower_two_sided_color when the face is an input
- panfrost: Report TEXTURE_BUFFER_OBJECTS cap when gl3 flag set
- panfrost: Set depth_enabled when stencil is enabled
- nir: Set the alignment for SSBO lowering
- panfrost: Make panfrost_bo_wait take a wait_readers bool
- panfrost: Fix calls to panfrost_flush_batches_accessing_bo
- panfrost: Fake RGTC support
- panfrost: Use more tilebuffer sizes
- panfrost: 8x MRT support
- pan/mdg: Use the blend RT for blend shader framebuffer fetches
- panfrost: Allow PIPE_TEXTURE_1D_ARRAY textures
- pan/mdg: Fix spilling of non-32-bit types

Icenowy Zheng (1):

- panfrost: signal syncobj if nothing is going to be flushed

Ilia Mirkin (14):

- freedreno/a3xx: there's no r8i/ui rb format, only rg8i/rg8ui
- freedreno/a3xx: reinstate rgb10_a2ui texture format
- freedreno/ir3: avoid applying (sat) on bary.f
- freedreno/a3xx: fix const footprint
- freedreno: fix off-by-one in assertions checking for const sizes
- freedreno/a3xx: parameterize ubo optimization
- freedreno/a3xx: fix rasterizer discard
- nouveau: allow invalidating coherent/persistent buffer backings
- st/mesa: allow R8 to not be exposed as renderable by driver
- a4xx: add noperspective interpolation support
- a4xx: add polygon offset clamp, fix units
- ir3: mark ucp_enables as allowed values on all keys
- a4xx: hook up centroid ij coords
- ir3: use empirical size for params as used by the shader

Indrajit Kumar Das (2):

- st/mesa: use fragment shader to copy stencil buffer
- st/mesa: optimize DEPTH_STENCIL copies using fragment shader

Italo Nicola (17):

- panfrost: Fix outmods on int to float conversions
- pan/mdg: fix src_type in instructions that need a implicit zero
- pan/mdg: prepare effective_writemask()
- pan/mdg: eliminate references to ins->alu.op
- pan/mdg: eliminate references to ins->alu.reg_mode
- pan/mdg: fix comment
- pan/mdg: eliminate references to ins->alu.outmod
- pan/mdg: apply float outmods to textures
- pan/mdg: eliminate references to ins->texture.op
- pan/mdg: eliminate references to ins->load_store.op
- pan/mdg: defer register packing
- pan/mdg: externalize mir_pack_mod
- pan/mdg: remove ins->alu
- pan/mdg: refactor emit_alu_bundle
- pan/mdg: defer branch packing
- pan/mdg: remove ins->br_compact and ins->branch_extended
- pan/mdg: emit REGISTER_UNUSED on unused ALU src2

Iván Briano (9):

- anv: use the correct format on Android
- anv: Disable B5G6R5_UNORM_PACK16
- anv: Add a way to reserve states from a pool
- anv: Implement VK_EXT_custom_border_color
- anv: support externally synchronized pipeline caches
- anv: implement VK_PIPELINE_CREATE_FAIL_ON_PIPELINE_COMPILE_REQUIRED_BIT_EXT
- anv: enable VK_EXT_pipeline_creation_cache_control
- anv: Add VK_EXT_custom_border_color to relnotes
- anv: fix allocation of custom border color pool

James Park (1):

- amd/llvm: Reorder LLVM headers

James Zhu (1):

- ac/gpu_info: Correct Acturus cu bitmap

Jan Beich (5):

- drm-uapi: Add sync_file.h

- anv,iris: unbreak on BSDs after 812cf5f522ab,abf8aed68047
- util: enable futex usage on BSDs after 7dc2f4788288
- meson: unbreak sysctl.h detection on BSDs
- anv: disable i915_perf warning on non-Linux

Jan Palus (1):

- targets/opencl: fix build against LLVM>=10 with Polly support

Jan Zielinski (1):

- gallium/swr: Fix crashes in sampling code

Jason Ekstrand (167):

- intel/eu: Use non-coherent mode (BTI=253) for stateless A64 messages
- Revert "anv/gen12: Temporarily disable VK_KHR_buffer_device_address (and EXT)"
- vulkan: Allow destroying NULL debug report callbacks
- vulkan,anv: Add a common base object type for VkDevice
- anv: Stop clflushing events
- anv: Allocate CPU-side memory for events
- vulkan,anv: Add a base object struct type
- vulkan,anv: Move the DEFINE_HANDLE_CASTS macros to vk_object.h
- anv: Refactor setting descriptors with immutable sampler
- vulkan: Add run-time object type asserts in handle casts
- vulkan/wsi: Make wsi_swapchain inherit from vk_object_base
- anv/allocator: Add a start_offset to anv_state_pool
- vulkan/object: Always include the type
- anv,vulkan: Implement VK_EXT_private_data
- vulkan: Handle vkGet/SetPrivateDataEXT on Android swapchains
- nir: Make "divergent" a property of an SSA value
- util/list: Add a list pair iterator
- util/vma: Add an option to configure high/low preference
- util/vma: Add a debug print helper
- util/ra: Add [de]serialization support
- anv: Set 3DSTATE_VF_INSTANCING on the SVGS element
- anv: Set MOCS in 3DSTATE_CONSTANT_* on Gen9+
- nir: Add some docs to the metadata types
- anv: Call vk_object_base_finish for image views
- anv: Fix descriptor set clean-up on BO allocation failure
- nir: Use 8-bit types for most info fields
- anv:gpu_memcpy: Emit 3DSTATE_VF_INDEXING on Gen8+

- nir: Validate jump instructions as an instruction type
- nir: Use a switch statement in nir_handle_add_jump
- nir: Add documentation for each jump instruction type
- nir/clone: Re-use clone_alu for nir_alu_instr_clone
- nir: Add a new helper for iterating phi sources leaving a block
- nir: Add a store_reg helper and use the builder in phis_to_regs
- nir: Add const to nir_intrinsic_src_components
- nir/lower_double_ops: Rework the if (progress) tree
- nir/opt_deref: Report progress if we remove a deref
- nir/copy_prop_vars: Record progress in more places
- nir: Fix sources for image atomic fadd
- intel/vec4: Stomp the return type of RESINFO to UINT32
- intel/fs: Fix unused texture coordinate zeroing on Gen4-5
- intel/fs: Emit HALT for discard on Gen4-5
- anv/allocator: Compare to start_offset in state_pool_free_no_vg
- nir: Add a nir_metadata_all enum value
- nir: Add a nir_shader_preserve_all_metadata helper
- nir: Call nir_metadata_preserve on !progress
- nir: Properly preserve metadata in more cases
- intel/nir: Call nir_metadata_preserve on !progress
- iris: Better handle metadata in NIR passes
- anv: Add an anv_batch_set_storage helper
- anv: Add anv_pipeline_init/finish helpers
- nir/intrinsics: Put the _intel intrinsics together at the end
- anv: Use resolve_device_entrypoint for dispatch init
- vulkan: Update Vulkan XML and headers to 1.2.145
- anv: Bump the advertised patch version to 145
- intel/fs: Expose a couple of NIR lowering helpers
- intel/fs: Break wm_prog_data setup into a helper
- intel/fs: Move more prog_data setup into populate_wm_prog_data
- intel/compiler: Expose brw_texture_offset to C
- intel/eu: Add a brw_urb_dest_msg_type helper
- intel/eu: Set the right subnr for ALIGN16 destinations
- intel/eu: Add the RNDU opcode
- vulkan/wsi: Don't consider VK_SUBOPTIMAL_KHR to be an error condition
- wsi/x11: Log swapchain status changes

- freedreno: Only call nir_lower_io on shader_in/out
- lima: Only call nir_lower_io on shader_in/out
- nouveau: Only call nir_lower_io on shader_in/out
- vc4: Only call nir_lower_io on shader_in/out
- v3d: Only call nir_lower_io on shader_in/out
- panfrost: Only call nir_lower_io on shader_in/out
- nir: Assert that nir_lower_io is only called with allowed modes
- nir: Remove shared support from lower_io
- nir: Add docs to nir_lower[_explicit]_io
- anv: Handle clamping of inverted depth ranges
- nir/validate: Don't abort() until after the shader has printed
- spirv: Skip phis in unreachable blocks in the second phi pass
- spirv: Allow block-decorated struct types for constants
- vulkan: Update Vulkan XML and headers to 1.2.148
- anv: Advertise VK_EXT_image_robustness
- spirv: Update headers and grammar json
- spirv: Add support for SPV_EXT_shader_atomic_float
- intel/fs: Use the correct logical op for global float atomics
- anv: Advertise support for VK_EXT_shader_atomic_float
- nir: Allow for system values with variable numbers of destination components
- nir/lower_io: Choose to set access based on intrinsic metadata
- nir/lower_io: Use b2b for shader and function temporaries
- nir/lower_io: Add support for global scratch addressing
- spirv: Simplify our handling of NonUniform
- spirv: Drop the void *ptr from vtn_value
- spirv: Fix indentation in vtn_handle_ptr
- spirv: Clean up OpSignBitSet
- spirv: Use nir_bany/ball for OpAny/All
- spirv: Add a helpers for getting types of values
- spirv: Rename push_value_pointer to push_pointer
- spirv: Add a vtn_push_nir_ssa helper
- spirv/amd: Use vtn_push_nir_ssa
- spirv: Add a vtn_get_nir_ssa helper
- spirv: Use the new helpers in OpConvertUToPtr/PtrToU
- spirv: Refactor vtn_push_ssa
- spirv/alu: Use vtn_push_ssa_value

- spirv/glsl450: Use vtn_push_ssa_value
- spirv/subgroups: Stop incrementing w
- spirv/subgroups: Refactor to use vtn_push_ssa
- spirv: Simplify vtn_ssa_value creation
- spirv: Hand-roll fewer vtn_ssa_value creations
- spirv: Add better checks for SSA value types
- spirv: Drop the sampled boolean from vtn_type
- spirv: Give atomic counters their own variable mode
- spirv: Add a helper for getting the NIR type of a vtn_type
- spirv: Remove a dead case in function parameter handling
- spirv: More heavily use vtn_ssa_value in function parameter handling
- anv,turnip,radv,clover,glspirv: Run nir_copy_prop before nir_opt_deref
- spirv: Rework our handling of images and samplers
- spirv: Also copy over binding information for atomic counters
- nir: Take a mode in remove_unused_io_vars
- nir/dead_variables: Respect the modes passed to remove_dead_vars
- nir: Add nir_foreach_shader_in/out_variable helpers
- nir: Add a nir_foreach_function_temp_variable helper
- nir: Add a nir_foreach_uniform_variable helper
- nir: Add a nir_foreach_gl_uniform_variable helper for GL linking
- nir: Add and use a nir_variable_list_for_mode helper
- nir: Take a nir_shader and variable mode in assign_var_locations
- nir: Take a shader and variable mode in nir_assign_io_var_locations
- nir/linking: Rework some internal helpers
- st/nir: Rework fixup_varying_slots
- nir/split_vars: Add mode checks to list walks
- nir: Split nir_index_vars into two functions
- nir/lower_amul: Add a variable mode check
- nir: Use a nir_shader and mode in lower_clip_cull_distance_arrays
- nir/lower_io_to_temporaries: Use a separate list for new inputs
- nir/io_to_vector: Use nir_foreach_variable_with_modes
- nir/lower_two_sided_color: Use nir_variable_create
- nir/lower_uniforms_to_ubo: Use nir_foreach_variable_with_modes
- nir/split_per_member_structs: Use nir_variable_with_modes_safe
- nir/lower_variable_initializers: Restrict the modes we lower
- nir/gl_nir_linker: Use nir_foreach_variable_with_modes

- freedreno/ir3_lower_tess: Rework var list helpers
- lima/standalone: Rework i/o variable fixup
- freedreno/ir3_cmdline: Rework i/o variable fixup
- r600/sfn/lower_tess_io: Rework get_tcs_varying_offset
- r600/sfn/lower_tex: Get rid of the lower_sampler vector
- r600/sfn: Use nir_foreach_variable_with_modes in IO vectorization
- panfrost/midgard: Make search_var take a nir_shader and mode
- panfrost: Use nir_foreach_variable_with_modes in pan_compile
- aco: Use nir_foreach_variable_with_modes to walk SSBOs
- mesa/ptn: Use nir_variable_create
- gallium/ttn: Use variable create/add helpers
- nir: Use a single list for all shader variables
- nir/split_per_member_structs: Inline split_variables_in_list
- nir/gl_nir_linker: Call add_vars_with_modes once for GL_PROGRAM_INPUT
- nir: Add a find_variable_with_[driver_]location helper
- vulkan: Update Vulkan XML and headers to 1.2.149
- anv: Implement VK_EXT_4444_formats
- nir/deref: Don't try to compare derefs containing casts
- compiler/types: Add a struct_type_is_packed wrapper
- spirv: Do more complex unwrapping in get_nir_type
- anv: Advertise shaderIntegerFunctions2
- spirv: Don't emit RMW for vector indexing in shared or global
- clover/spirv: Don't call llvm::regularizeLlvmForSpirv
- intel/nir: Pass the nir_builder by reference in lower_alpha_to_coverage
- intel/nir: Rewrite the guts of lower_alpha_to_coverage
- intel/fs: Fix MOV_INDIRECT and BROADCAST of Q types on Gen11+
- intel/fs: Don't copy-propagate stride=0 sources into ddx/ddy
- iris: Re-emit push constants if we have a varying workgroup size
- spirv: Run repair_ssa if there are discard instructions
- nir: More NIR_MAX_VEC_COMPONENTS fixes
- intel/fs/swsb: SCHEDULING_FENCE only emits SYNC_NOP
- radeonsi: Only call nir_lower_var_copies at the end of the opt loop

Jesse Natalie (10):

- nir_lower_io: Add addr_format_is_offset helper
- nir: When nir_lower_vars_to_explicit_types is run on temps, update scratch_size
- nir: Support load/store of temps as scratch in nir_lower_explicit_io

- nir: Support vec8/vec16 in nir_lower_bit_size
- nir: Support algebraic opts on vectors larger than 4
- nir: Support 8 and 16 component vectors for reduceable intrinsics
- nir/vtn: Add support for 8 and 16 vector ball/bany
- u_debug_stack_test: Fix MSVC compiling by using ATTRIBUTE_NOINLINE
- nir: More NIR_MAX_VEC_COMPONENTS fixes
- glsl_type: Add packed to structure type comparison for hash map

JibbityJobbity (1):

- drirc: Enable glthread for PCSX2

Jon Turney (1):

- glthread: Fix use of alloca() without #include "c99_alloca.h"

Jonathan Gray (13):

- util: unbreak endian detection on OpenBSD
- util/anon_file: add OpenBSD shm_mkstemp() path
- meson: build with _ISOC11_SOURCE on OpenBSD
- meson: don't build with USE_ELF_TLS on OpenBSD
- meson: conditionally include -ldl in gbm pkg-config file
- util: futex fixes for OpenBSD
- util/u_thread: include pthread_np.h if found
- anv: use os_get_total_physical_memory()
- util/os_misc: add os_get_available_system_memory()
- anv: use os_get_available_system_memory()
- util/os_misc: os_get_available_system_memory() for OpenBSD
- radv: remove seccomp includes
- vulkan: make VK_TIME_DOMAIN_CLOCK_MONOTONIC_RAW_EXT conditional

Jonathan Marek (135):

- turnip: update "fetchsize" value to match fdl6_layout changes
- turnip: enable tiling for compressed formats
- util/format: translate 422_UNORM and 420_UNORM vulkan formats
- freedreno/registers: document 422_UNORM and 420_UNORM formats
- turnip: implement VK_KHR_sampler_ycbcr_conversion
- turnip: enable 422_UNORM formats
- freedreno: move a4xx specific layout code to a4xx code
- freedreno/a5xx: remove unused reference to gmem_alignw in layout code
- freedreno/a6xx: don't use gmem_alignw for imported buffers
- freedreno/a6xx: split up gmem/tile alignment requirements

- freedreno: reduce extra height alignment in a6xx layout
- freedreno/a6xx: use RESOLVE_TS event
- freedreno: add adreno 650
- freedreno/layout: add explicit offset/pitch argument to fdl6_layout
- turnip: support VkImageDrmFormatModifierExplicitCreateInfoEXT
- turnip: fix RENDER_COMPONENTS value
- turnip: move HLSQ_UPDATE_CNTL write to before xs config writes
- turnip: update some properties based on blob driver
- turnip: clamp sampler minLod/maxLod
- freedreno/a6xx: use nonbinning VS when GS is used
- turnip: correctly emit non-binning vs in transform feedback case
- turnip: fix HW binning with geometry shader
- turnip: use common emit_xs_cntl to fill a6xx_sp_xs_ctrl_reg0
- turnip: fix VFD_CONTROL for binning pass
- turnip: pipeline program state refactor
- turnip: share code between 3D blit/clear path and tu_pipeline
- turnip: add layered 3D path clear for CmdClearAttachments
- turnip: add emit renderpass cache flushes for sysmem 3D CmdClearAttachments
- turnip: remove some dead/redundant code
- freedreno/ir3: fix ir3_nir_move_varying_inputs
- turnip: remove duplicated stage2opcode and stage2shaderdb
- turnip: simplify stage2 helpers
- turnip: set VFD_INDEX_OFFSET in 3D clear/blit path
- turnip: fix 3D path always being used for CmdBlitImage
- turnip: fix cubic filtering with CmdBlitImage
- turnip: compute and graphics have completely separate state
- turnip: move descriptor set BO tracking to CmdBindDescriptorSets
- turnip: improve dirty bit handling a bit
- turnip: delete dead dynamic state code
- turnip: refactor draw states and dynamic states
- turnip: input attachment descriptor set rework
- turnip: use draw states for input attachments
- turnip: use u_format for packing gmem clear values
- freedreno/a6xx: FETCHSIZE is PITCHALIGN
- freedreno/fdl6: rework layout code a bit (reduce linear align to 64 bytes)
- turnip: fix a crash when rasterizerDiscardEnable is set

- turnip: fix a sample shading case
- turnip: fix renderpass gmem configs when there are too many attachments
- turnip: set the API version
- turnip: move enum translation functions to a common header
- freedreno/a6xx: VSC "STRM_ARRAY_PITCH" is "STRM_LIMIT"
- freedreno/a6xx: remove unnecessary OVERFLOW_FLAG_REG check
- turnip: remove unnecessary OVERFLOW_FLAG_REG check
- freedreno/a4xx: restore pitch to bytes change to layout code
- freedreno/a4xx: simplify setup_slices
- turnip: rework streamout state and add missing counter buffer read/writes
- turnip: refactor CmdDraw* functions (and a few fixes)
- turnip: enable VK_EXT_index_type_uint8
- turnip: implement CmdDrawIndirectByteCountEXT
- turnip: fix ts_cs_memory typo
- turnip: use pipeline cs for shader programs instead of separate bo
- freedreno/registers: a6xx depth bounds test registers
- turnip: implement depthBounds
- turnip: translate CreateRenderPass to CreateRenderPass2
- turnip: replace a memset(0) with zalloc in CreateRenderPass
- turnip: use RenderPassCreateInfo for render_pass_add_implicit_deps
- turnip: move some logic out of create_render_pass_common
- turnip: implement VK_EXT_vertex_attribute_divisor
- turnip: fix empty scissor case
- turnip: fix update_stencil_mask
- turnip: disable early_z for VK_FORMAT_S8_UINT
- freedreno/registers: add CP_DRAW_INDIRECT_MULTI
- freedreno/ir3: add support for load_draw_id
- turnip: implement VK_KHR_shader_draw_parameters
- turnip: fix VK_STRUCTURE_TYPE_PHYSICAL_DEVICE_VULKAN_1_1_FEATURES
- turnip: fix huge scissor min/max case
- freedreno/ir3: fix resinfo wrmask
- freedreno/regs: add extra bits for UBWC array pitch
- turnip: enable largePoints
- turnip: enable depthBiasClamp
- freedreno/registers: update varying-related registers
- freedreno/a3xx: support LINEAR_PIXEL/PERSP_CENTROID/LINEAR_CENTROID sysvals

- freedreno/a4xx: fake LINEAR_PIXEL varying support for u_blitter
- freedreno/ir3: add generic get_barycentric()
- freedreno/a5xx: set missing bary sysvals
- freedreno/a6xx: set missing bary sysvals
- turnip: set missing bary sysvals
- freedreno/ir3: add support for INTERP_MODE_NOPERSPECTIVE
- turnip: make tiling config part of framebuffer state
- turnip: rework render_tiles loop
- turnip: vsc improvements
- turnip: fix tess param bo size calculation
- turnip: clear_blit: pass aspect mask to setup function
- turnip: support multi-image layouts
- turnip: enable 420_UNORM formats
- freedreno/layout: fix explicit layout offset not added to slice offset
- freedreno/ir3: fix/rework tess levels
- Revert "nir: Add an option for lowering TessLevelInner/Outer to vecs"
- Revert "nir: Support sysval tess levels in SPIR-V to NIR"
- freedreno/regs: document SS6_UBO state src
- turnip: use global bo for clear blit shaders
- freedreno/ir3: add support for a650 tess shared storage
- freedreno/regs: document CS shared storage size bit
- freedreno/a2xx: fix compressed textures
- freedreno: add a fd_resource_pitch helper
- freedreno/layout: layout simplifications and pitch from level 0 pitch
- turnip: fix active_desc_sets not being set for compute pipeline
- freedreno/ir3: fix setup_input for sparse vertex inputs
- freedreno/ir3: run nir_opt_loop_unroll in optimization loop
- freedreno: fix layout pitchalign field not being set for imported buffers
- freedreno/regs: update primitive output related registers
- turnip: clean up primitive output state
- turnip: drop GS clear path
- turnip: use DIRTY SDS bit to avoid making copies of pipeline load state ib
- turnip: emit compute pipeline directly in CmdBindPipeline
- turnip: fix inconsistencies with tu6_load_state_size
- turnip: remove use of tu_cs_entry for draw states
- gitlab-ci: re-enable arm64_a630_vk

- freedreno/regs: update a6xx GRAS registers
- freedreno/regs: update a6xx RB regs
- freedreno/regs: update a6xx VPC regs
- freedreno/regs: update a6xx PC regs
- turnip: disable tiling for NV12/IYUV formats
- turnip: remove extra gmem alignment
- freedreno/ir3: fix wrong local_primitive_id_start type
- turnip: move WFI out of draw state to fix a650 hangs
- turnip: use patchControlPoints for HS_INPUT_SIZE value
- turnip: fix SP_HS_UNKNOWN_A831 value for A650
- turnip: workaround for a630 d24_unorm_s8_uint fails
- turnip: fix sysmem CmdClearAttachments 3D fallback breaking GMEM path flush
- turnip: delete tu_clear_sysmem_attachments_2d
- turnip: add support for D32_SFLOAT_S8_UINT
- turnip: rework extended formats to allow more extended formats
- util/format: translate A4R4G4B4_UNORM and A4B4G4R4_UNORM vulkan formats
- turnip: implement VK_EXT_4444_formats

Jordan Justen (17):

- intel/dev: Split .num_subslices out of GEN12_FEATURES macro
- intel/dev: Add device info for RKL
- intel/l3: Don't rely on cfg entry URB size being 0 as a sentinal
- intel/l3: Allow platforms to have no l3 configurations
- iris/l3: Enable L3 full way allocation when L3 config is NULL
- anv: Set L3 full way allocation at context init if L3 cfg is NULL
- intel/dev: Add device info for DG1
- iris: Make use of devinfo has_aux_map field
- anv: Make use of devinfo has_aux_map field
- anv/pipeline: Split VFE/INTERFACE_DESCRIPTOR out to emit_media_cs_state
- anv/cmd_buffer: Split GPGPU_WALKER out to emit_gpgpu_walker
- iris: Split walker and state update into iris_upload_gpgpu_walker
- iris/compute: Split out iris_load_indirect_location
- intel/compiler/cs: Allow simd32 in some more cases with no8 and/or no16
- intel/compiler/fs: Still attempt simd32 when INTEL_DEBUG=no16 is used
- iris: Add missing break in switch in modifier_is_supported
- anv, iris: Set MediaSamplerDOPClockGateEnable for gen12+

Jose Maria Casanova Crespo (4):

- v3d: Fix swizzle in DXT3 and DXT5 formats
- v3d: Include supported DXT formats to enable s3tc/dxt extensions
- vc4: don't relay on intr->num_components for non-vectorized intrinsics
- nir: only uniforms with dynamically_uniform offset are dynamically_uniform

Joshua Ashton (7):

- anv: Remove RANGE_SIZE usage
- radv: Remove RANGE_SIZE usage
- turnip: Remove RANGE_SIZE usage
- vulkan: Update Vulkan XML and headers to 1.2.140
- radv: Implement VK_EXT_custom_border_color
- radeonsi: Use TRUNC_COORD on samplers
- radv: Implement VK_EXT_4444_formats

José Fonseca (3):

- glthread: Add GLAPIENTRY to _mesa_marshal_MultiDrawArrays.
- appveyor: Upgrade pip.
- appveyor: Use Python3.

Karol Herbst (50):

- nir/deref: copy ptr_stride when rematerializing
- nir/validate: validate the stride for deref_ptr_as_array
- Revert "nir/validate: validate the stride for deref_ptr_as_array"
- nvir/nir: use component helpers instead of insn->num_components
- st/mesa: lower images when needed
- nir/lower_images: fix for array of arrays
- nir/lower_images: handle dec and inc
- nv50/ir/nir: move away from image_deref intrinsics
- nv50/ir/nir: handle image atomic inc and dec
- nv50/ir/nir: remove image uniform hack
- gv100/ir: fix atom cas
- gv100/ir: fix shift lowering
- gv100/ir: fix OP_TXG for shadow textures
- nv50/ir/nir: add workaround for double vertex attribs
- nv50/ir/print: add missing VIEWPORT_MASK handling
- nv50/ir/nir: fix ext_demote_to_helper_invocation
- nv50/ir/nir: fix nv_viewport_array2
- nvc0: enable spirv caps with nir
- nv50/ir/nir: don't emit a restart with set a stream_id

- nv50/ir/nir: handle clip vertex for tess eval shaders
- nv50/ir/nir: rework input output handling
- nv50/ir/nir: rework CFG handling
- nv50/ir/ra: convert some for loops to Range-based for loops
- nv50/ir/ra: fix memory corruption when spilling
- nv50/ir/nir: fix interpolation on explicit operations
- gv100/ir: implement sample shading
- gv100/ir: fix coherent and volatile memory access
- nv50/ir/nir: fix cache mode conversion
- nv50/ir: fix memset on non trivial types warning
- nv50/ir/tgsi: move call to tgsi_scan_shader inside Source constructor
- nvc0: set local mem size for compute on gv100
- nvc0: set sampler index mode to independently on gv100 compute
- gv100/ir: set ftz bit on floating point operations
- ci: bump libdrm to 2.4.102
- nouveau: enable HMM
- gallium: add PIPE_CAP_RESOURCE_FROM_USER_MEMORY_COMPUTE_ONLY
- nvc0: support PIPE_CAP_RESOURCE_FROM_USER_MEMORY_COMPUTE_ONLY
- nouveau: expose HMM
- ci: need to install wget in order to download libdrm
- ci: bump libdrm to 2.4.102
- nouveau: enable HMM
- gallium: add PIPE_CAP_RESOURCE_FROM_USER_MEMORY_COMPUTE_ONLY
- nvc0: support PIPE_CAP_RESOURCE_FROM_USER_MEMORY_COMPUTE_ONLY
- nouveau: expose HMM
- st/mesa: fix st_CopyPixels without support for stencil exports
- nv50/ir/tgsi: silence warning about unhandled GS_INPUT_PRIM property
- nv50/ir: initialize persampleInvocation to false
- nir/lower_io: assert that offsets are used for shader_in
- nv50/ir/nir: fix global_atomic_comp_swap
- spirv: extract switch parsing into its own function

Kenneth Graunke (20):

- iris: Include linux/sync_file.h instead of cut and pasting contents
- anv: Include linux/sync_file.h instead of cut and pasting contents
- iris: Rename iris_syncpt to iris_syncobj for clarity.
- iris: Give up on not passing ice to iris_init_batch

- iris: Destroy transfer slab after batches
- iris: Flush any current work in iris_fence_await before adding deps
- intel: Move anv_gem_supports_syncobj_wait to common code.
- iris: Detect DRM_SYNCOBJ_WAIT_FLAGS_WAIT_FOR_SUBMIT kernel support
- iris: Implement PIPE_FLUSH_DEFERRED support.
- intel: Delete hardcoded devinfo->urb.size values for Gen7+ (sans DG1).
- iris: Delete useless #define
- intel/eu: Add a brw_urb_desc helper
- CI: Disable Panfrost Mali-T820, Lima Mali-400 and Lima Mali-450 jobs
- intel: Disable loading drivers on DG1 devices for now
- nir: Fix divergence analysis for tessellation input/outputs
- iris: Implement pipe->texture_subdata directly
- iris: Fix CCS check in iris_texture_subdata().
- iris: Delete shader variants when deleting the API-facing shader
- iris: Reorder the loops in iris_fence_await() for clarity.
- iris: Drop stale syncobj references in fence_server_sync

Kristian Høgsberg (73):

- freedreno/ir3: Pass stream output info to ir3_shader_from_nir
- freedreno/ir3: Rename ir3_nir_lower_to_explicit_io
- freedreno/ir3: Add ir3_nir_lower_to_explicit_input() pass
- freedreno/ir3: Lower GS builtins before lowering IO
- freedreno/ir3: Drop hack to clean up split vars
- freedreno/fdl: Align after dividing by block size
- freedreno/a6xx: Set tfetch correctly for compressed formats
- freedreno/ir3: Drop wrmask for ir3 local and global store intrinsics
- freedreno/a6xx: Create shader dependent streamout state at compile time
- freedreno/a6xx: Map inputs to VFD entries up front
- freedreno/a6xx: Allocate ringbuffer based on VFD count
- freedreno/a6xx: Emit VFD setup as array writes
- freedreno/a6xx: Avoid stalling for occlusion queries
- freedreno: Use the right amount of &'s
- freedreno: Use explicit *_NONE enum for undefined formats
- turnip: Use hw enum when emitting A6XX_RB_STENCIL_CONTROL
- turnip: Use tu6_reduction_mode() to avoid warning
- turnip: Use {} initializer to silence warning
- freedreno/ir3: Avoid {0} initializer for struct reginfo

- src/util: Remove out-of-range comparison
- mapi: Fix a couple of warning in generated code
- mesa/st: Use memset to zero out struct
- egl/android: Move get_format under HAVE_DRM_GRALLOC guard where it's used
- egl/android: Drop unused variable
- freedreno/a6xx: Move per element offset to VFD_DECODE
- freedreno/a6xx: Decouple VFD_FETCH and VFD_DECODE
- freedreno/a6xx: Create stateobj for VFD_DECODE
- freedreno/a6xx: Program VFD_DEST_CNTL from program stateobj
- freedreno/a6xx: Turn on robustness extensions
- docs/features.txt: Update for freedreno
- freedreno/a6xx: Fix VFD_CONTROL emit
- freedreno/a6xx: Don't write REG_A6XX_RB_SRGB_CNTL in restore
- freedreno/a6xx: Set index buffer size to bo size
- freedreno: Handle DRM_FORMAT_MOD_INVALID in shared code
- turnip: Put VK_KHR_external_fence_fd stubs back
- freedreno/a6xx: Don't blit with R2D_RAW
- freedreno/a6xx: Move fd6_ifmt into fd6_blitter.c
- freedreno/a6xx: Split out src and dst setup helpers for blit
- freedreno/a6xx: Don't set unknown bit when tiling differs
- freedreno/a6xx: Set src and dst rects outside blit loop
- freedreno/a6xx: Program SP_2D_SRC_FORMAT outside blit loop
- freedreno/a6xx: Consolidate computing blit_cntl
- freedreno/a6xx: Don't emit src state when clearing
- freedreno/a6xx: Separate stencil sysmem clear fix
- freedreno/a6xx: Enable FMT6_10_10_10_2_UNORM blitting
- freedreno/a6xx: Make blit_control helper a little more helpful
- freedreno/a6xx: Program A6XX_SP_2D_SRC_FORMAT_COLOR_FORMAT based on dst format
- freedreno/a6xx: Move REG_A6XX_SP_2D_SRC_FORMAT programming to helper
- freedreno/a6xx: Move CP_SET_MARKER to setup helper
- freedreno/a6xx: Program RB_UNKNOWN_8C01 in setup helper
- freedreno/a6xx: Don't take pipe_blit_info in emit_blit_dst
- freedreno/a6xx: Split clear and blit texture into different functions
- freedreno/registers: Rename SP_2D_SRC_FORMAT
- turnip: Move device enumeration and feature discovery to tu_drm.c
- turnip: Move tu_bo functions to tu_drm.c

- turnip: Collapse some tu_drm wrappers
- turnip: Move remaining drm code to tu_drm.c
- turnip: Only include msm_drm in tu_drm.c
- egl/android: Remove unused variable
- mapi/test: Change type to unsigned for offset
- gallium: Switch u_debug_stack/symbol.c to util/hash_table.h
- util: Move stack debug functions to src/util
- util: Add unit test for stack backtrace caputure
- gallium/android: Rewrite backtrace helper for android
- ci: Include enough Android headers to let us compile test EGL
- mapi: Mark TLS symbols as optional in glapi-symbols.txt
- turnip: Make tu_android.c compile again
- meson: Define ANDROID and ANDROID_API_LEVEL when compiling for Android
- anv: Pass device to setup_gralloc0_usage for error reporting
- anv: Add stub for anv_gem_get_tiling() for Android
- vulkan: Allow global symbol HMI for Android
- radv/android: Remove unused variable
- ci: Add a build test for the Android platform

Krzysztof Raszkowski (1):

- gallium/swr: Fix building swr with MSVC

Laura Ekstrand (3):

- docs: include meson in the toctree
- docs: Remove version.
- docs: Add the favicon to the new page.

Leo Liu (3):

- radeon/vcn: reset the decode flags from message buffer
- radeon/vcn: add Sienna to use internal register offset
- radeon/vcn/dec: add db_aligned_height to message buffer

Lepton Wu (3):

- mapi: x86: Fix dynamic entries in x86 tsd stubs.
- mapi: Return NULL function pointers for GL_EXT_debug_marker
- egl: Allow software rendering for vgem/virtio_gpu in platform_device

Lionel Landwerlin (60):

- drm-shim: move handle lock to shim_fd
- drm-shim: don't create a memfd per BO
- drm-shim: silence warnings

- intel/dev: print out error when platform is not found by name
- intel: add stub_gpu tool
- ci: Add intel to shaderdb runs
- iris: don't assert on unfinished aux import in copy paths
- anv: don't expose VK_INTEL_performance_query without kernel support
- anv: fix alignments for uniform buffers
- genxml: run sorting script
- genxml: fix invalid end value for video fields
- genxml: factor out utility functions
- genxml: pack: deal with default field not being simple integers
- intel/genxml: fix bits generation for MI_LOAD_REGISTER_IMM
- intel/mi-builder: add framework for self modifying batches
- anv: don't reserve a particular register for draw count
- anv: add a new execution mode for secondary command buffers
- intel/genxml: add PIPE_CONTROL command cache invalidate bit
- intel/perf: make pipeline statistic query loading optional
- intel/perf: store the appropriate OA formats in queries
- intel/perf: update generated code to ralloc all data
- intel/perf: create a unique list of counters
- intel/perf: compute number of passes for a set of counters
- intel/perf: emit counter units in generated code
- intel/perf: add helper to compute metrics from counters
- intel/perf: add counter category to generated code
- intel/perf: report whether the platform supported
- anv: use a query filled by the perf code
- intel/perf: reuse offset specified in the query
- anv: Implement VK_KHR_performance_query
- intel/perf: repurpose INTEL_DEBUG=no-oaconfig
- anv: fixup unwinding of device create failure
- blorp: rename workaround address function
- anv: store the workaround address
- iris: store workaround address
- i965: store workaround_bo offset
- intel: add identifier for debug purposes
- iris: add identifier BO
- i965: add identifier BO

- anv: add identifier BO
- intel/aub_error_decoder: print driver identifier if found
- iris: fix BO destruction in error path
- i965: don't forget to set screen on duped image
- iris: fix export of GEM handles
- i965: fix export of GEM handles
- anv: add an option to disable secondary command buffer calls
- anv: garbage collect timeline semaphore when querying value
- iris: fix fallback to swrast driver
- anv: fix uninitialized variable access
- anv: properly handle fence import of sync_fd = -1
- anv: fix descriptor set free
- anv: fix incorrect realloc failure handling
- anv: centralize vk to gen arrays
- anv: fix up dynamic clip emission
- anv: don't fail userspace relocation with perf queries
- anv: fix transform feedback surface size
- anv: VK_INTEL_performance_query interaction with VK_EXT_private_data
- intel/perf: store query symbol name
- intel/perf: fix raw query kernel metric selection
- intel/compiler: fixup Gen12 workaround for array sizes

Liviu Prodea (1):

- util: Make process_test path compatible with mingw native toolchains

Louis-Francis Ratté-Boulianne (1):

- nir: Always create UBO variable when lowering uniforms to ubo

Lucas Stach (3):

- etnaviv: generalize FE stall before loading shader and sampler states
- etnaviv: retarget transfer to render resource when necessary
- etnaviv: don't expose timer queries

Luigi Santivetti (3):

- dri2: dri2_make_current() fold multiple if blocks
- dri2: do not conflate unbind and bindContext() failure
- egl/dri2: try to bind old context if bindContext failed

Marcin Ślusarz (24):

- i965: remove unused variable
- glsl_to_tgsi: add fallthrough comments

- glsl: cleanup vertex shader input checks
- iris: remove unused iris_bo->swizzle_mode
- intel/compiler: fix Android build
- st/mesa: fix reporting of float perf counters max value
- iris: return max counter value for AMD_performance_monitor
- iris: remove iris_monitor_config
- intel/perf: move query_mask and location out of gen_perf_query_counter
- iris: propagate error from gen_perf_begin_query to glBeginPerfQueryINTEL
- i965: propagate error from gen_perf_begin_query to glBeginPerfQueryINTEL
- util: fix possible fd leaks in os_socket_listen_abstract
- glsl: catch out of bounds access in the debug version
- util: fix possible buffer overflow in util_get_process_exec_path
- util/format: initialize non-important components to 0
- mesa: fix out of bounds access in glGetFramebufferParameterivEXT
- mesa: quiet down static analyzers
- iris: quiet down static analyzers
- intel/vec4: fix out of bounds read
- intel/perf: fix performance counters availability after glFinish
- anv: refresh cached current batch bo after emitting some commands
- anv: fix minor gen_ioctl(I915_PERF_IOCTL_CONFIG) error handling issue
- intel/perf: split load_oa_metrics
- intel/perf: export performance counters sorted by [group|set] and name

Marek Olšák (226):

- mesa: optimize glPush/PopClientAttrib by removing malloc overhead
- mesa: don't call _mesa_update_state for _mesa_get_clamp_fragment_color
- mesa: don't set unnecessary program flags in _mesa_update_state
- mesa: don't update shaders on fixed-func state changes if user shaders are bound
- mesa,st/mesa: add a fast path for non-static VAOs
- mesa: inline vbo_context inside gl_context to remove vbo_context dereferences
- mesa: add glInternalBufferSubDataCopyMESA for glthread
- mesa: add _mesa_InternalBind{ElementBuffer,VertexBuffers} for glthread
- glthread: do glBufferSubData as unsynchronized upload + GPU copy
- glthread: don't use atomics for refcounting to decrease overhead on AMD Zen
- glthread: track pointers and strides for Pointer & EXT_dsa attrib functions
- glthread: track instance divisor changes
- glthread: track primitive restart state

- glthread: initialize VAOs properly
- glthread: handle POS vs GENERIC0 aliasing
- glthread: handle gl{Push,Pop}ClientAttrib{DefaultEXT} for glthread states
- glthread: upload non-VBO vertices and indices for non-Indirect non-IBM draws
- tgsi_to_nir: handle TGSI_SEMANTIC_BLOCK_SIZE
- tgsi_to_nir: handle TGSI_OPCODE_BARRIER
- radeonsi: unify and align down the max SSBO/TBO/UBO buffer binding size
- radeonsi: clean up and deduplicate code around internal compute dispatches
- radeonsi: bind shader images after DCC is disabled for image stores
- radeonsi: add SI_IMAGE_ACCESS_DCC_OFF to ignore DCC for shader images
- radeonsi: implement and use compute-based DCC decompression on gfx9-10
- radeonsi: add a workaround to fix KHR-GL45.texture_view.view_classes on gfx9
- radeonsi: fix si_compute_clear_render_target with render condition enabled
- radeonsi: revert an accidental change in si_clear_buffer
- Revert "ac/surface: remove RADEON_SURF_TC_COMPATIBLE_HTILE and assume it's always set"
- Revert "ac: reassociate FP expressions for inexact instructions for radeonsi"
- ac/surface: fix MSAA crash with FORCE_SWIZZLE_MODE on gfx9
- radeonsi: don't wait for idle at the end of gfx IBs
- ac/surface: unset RADEON_SURF_TC_COMPATIBLE_HTILE if HTILE hasn't been computed
- radeonsi/gfx9: always use IMG_DATA_FORMAT_S8_32 for 8-bit stencil
- radeonsi: allow tc_compatible_htile to be mutable
- radeonsi: enable TC-compatible HTILE on demand for best Z/S performance
- tgsi_to_nir: translate non-vec4 image stores correctly
- radeonsi: fix compilation of monolithic PS
- amd: update amdgpu_drm.h
- amd: remove duplicated definitions from amdgpu_drm.h
- amd: assume CMASK is always rb/pipe_aligned, remove ac_surface.u.gfx9.cmask
- amd: assume HTILE is always rb/pipe_aligned, remove ac_surface.u.gfx9.htile
- ac/surface,radeonsi: move the set/get_bo_metadata code to ac_surface.c
- ac/surface,radeonsi: move the set/get_umd_metadata code into ac_surface.c
- amd: unify code for overriding offset and stride for imported buffers
- ac/surface: override all offsets including metadata offsets
- ac/surface: fix broken pitch override on gfx8
- gallium: rename 'state tracker' to 'frontend'
- gallium: change comments to remove 'state tracker'
- gallium: rename PIPE_RESOURCE_FLAG_ST_PRIV to FRONTEND_PRIV

- gallium: remove more “state tracker” occurences
- radeonsi: also enable tgsi_to_nir caching for compute shaders
- glthread: stop using GLenum16 to get correct GL errors for out-of-bounds enums
- radeonsi: don’t expose 16xAA on chips with 1 RB due to an occlusion query issue
- ac/nir: honor ACCESS_STREAM_CACHE_POLICY for L1 and L0 caches too
- radeonsi: use correct clear value size for EQAA in expand_fmask
- radeonsi: optimize access pattern for compute blits with linear textures
- radeonsi: tweak clear/copy_buffer limits when to use compute
- radeonsi: simplify setting resource usage for si_init_temp_resource_from_box
- radeonsi: rename SI_RESOURCE_FLAG_TRANSFER to FORCE_LINEAR
- radeonsi: use vi_dcc_enabled instead of using tex->surface.dcc_offset directly
- radeonsi: use display_dcc_offset for setting displayable_dcc_cb_mask
- winsys/amdgpu: add RADEON_FLAG_UNCACHED for faster blits over PCIe
- radeonsi: disable the L2 cache for most CPU mappings of textures
- radeonsi: disable the L2 cache for CPU read mappings of buffers
- radeonsi: compute perf tests - don’t test 1 wave/SA limit, test no limit first
- radeonsi: test uncached clear/copy buffer performance with compute shaders
- gallium/u_threaded: execute transfer_unmap with THREAD_SAFE directly
- ac/gpu_info: compute the best safe IB alignment
- ac/surface: don’t compute single-sample CMASK if it’s unaligned
- radeonsi: don’t use INDIRECT_BUFFER within IBs
- radeonsi: decrease the max GS invocation count to 32
- Revert “radeonsi: don’t wait for idle at the end of gfx IBs”
- ac: update register and packet definitions for preemption
- radeonsi: move resetting tracked registers into a new function
- radeonsi: split si_all_descriptors_begin_new_cs and rename functions
- radeonsi: don’t enable TC-compatible HTILE for stencil if stencil doesn’t use it
- radeonsi/gfx8: enable TC-compatible HTILE from the beginning as before
- radeonsi: don’t hardcode most perf counter block counts
- ac/gpu_info: replace num_good_cu_per_sh with min/max_good_cu_per_sa
- amd: replace SH -> SA (shader array) in comments
- radeonsi/gfx10: implement most performance counters
- glthread: don’t upload for glDraw inside a display list and always sync
- nir: add i2imp and u2ump opcodes for conversions to mediump
- nir: add int16 and uint16 type helpers
- nir: lower int16 and uint16 in nir_lower_mediump_outputs

- nir: fix lower_wpos for 16-bit fddy
- nir: add options::vectorize_vec2_16bit to limit vectorization to vec2 16
- glsl: treat lowp as mediump when lowering builtins
- glsl: handle int16 and uint16 types and add instructions for mediump
- glsl: lower mediump integer types to int16 and uint16
- glsl: lower mediump partial derivatives
- glsl: lower the precision of imageLoad
- glsl: lower samplers with highp coordinates correctly
- gallium: add shader caps INT16 and FP16_DERIVATIVES
- ac: rename has_double_rate_fp16 -> has_packed_math_16bit
- ac/nir: use more types from ac_llvm_context
- ac/nir: support vector types in the type suffix of overloaded intrinsics
- ac/nir: remove type and num_channels args from ac_build_buffer_store_common
- ac/nir: support 16-bit data in buffer_load_format opcodes
- ac/nir: support 16-bit data in image opcodes
- ac/nir: handle nir_op_[fiu]2[fiu]mp opcodes
- ac/nir: select v_cvt_pkrtz for all conversions from f32 to f16 for radeonsi
- ac/nir: set the second v_cvt_pkrtz argument to undef if it's unused
- ac/nir: support v2f16 derivatives
- nir: don't count samplers and images in interface blocks
- nir: gather which images are buffers
- nir: gather which images are MSAA
- radeonsi: remove unused leftover code for INDIRECT_BUFFER inside IBs
- radeonsi: remove const_buffers_declared hacks
- radeonsi: pass at most 3 images and/or shader buffers via user SGPRs for compute
- radeonsi: add a hack to disable TRUNC_COORD for shadow samplers
- gallium/u_vbuf: get rid of some pointer dereferences
- gallium/u_vbuf: add a faster path for uploading non-interleaved attribs
- glthread: sync in glFlush for multiple contexts
- radeonsi: enable ARB_sparse_buffer
- ac,radeonsi: replace == GFX10 with >= GFX10 where it's needed
- ac,radeonsi: start adding support for gfx10.3
- ac/surface: add displayable DCC code for gfx10.3
- radeonsi: honor a user-specified pitch on gfx10.3
- radeonsi: enable larger SDMA clears and copies on gfx10.3
- radeonsi: implement R9G9B9E5 render target and image store support on gfx10.3

- radeonsi: move L2_CACHE_CONTROL registers into si_emit_framebuffer_state
- radeonsi: set BIG_PAGE fields on gfx10.3
- radeonsi: don't set any XNACK options on gfx10.3
- ac: align num_vgprs for gfx10.3
- radeonsi: add support for Sienna Cichlid
- radeonsi: require LLVM 11 for gfx10.3
- ac/surface: don't recompute the DCC retile map for imported textures
- amd/addrlib: don't recompute DCC info for every ComputeDccAddrFromCoord call
- amd/addrlib: remove unused members of ADDR2_COMPUTE_DCC_ADDRFROMCOORD_INPUT
- ac/surface: add a wrapper structure to hold ADDR_HANDLE
- ac/surface: cache DCC retile maps (v2)
- amd/addrlib: fix the C++ one definition rule violation
- ac/surface: don't set is_displayable if displayable DCC is missing
- ac/surface: require that gfx8 doesn't have DCC in order to be displayable
- ac/surface: enable DCC for the first level in the mip tail on gfx10
- ac/surface: don't free dcc_retile_map on failure
- radeonsi: compact MRTs to save PS export memory space
- ac/nir: fix 64-bit division for GL CTS
- glapi: fix incorrect param names in ARB_vertex_attrib_binding functions
- glthread: rename non_vbo_attrib_mask -> user_buffer_mask, attribs -> buffers
- glthread: handle ARB_vertex_attrib_binding
- radeonsi: don't wait for idle at the end of gfx IBs
- radeonsi: replace ctx->screen with sscreen in si_flush_gfx_cs
- glsl,driconf: add allow_glsl_120_subset_in_110 for SPECviewperf13
- driconf: add workarounds for SPECviewperf13
- amd: add proper definitions for NOP packets
- ac,winsys/amdgpu: align IBs the same as the kernel
- radeonsi: don't add the border color buffer into the init_config state
- radeonsi: rename init_config states to cs_preamble states
- radeonsi: don't add the tess ring buffers into the cs_preamble state
- radeonsi: make wait_mem_scratch unmappable
- radeonsi: disallow adding BOs into si_pm4_state except 1 shader BO per state
- radeonsi: make si_pm4_cmd_begin/end static and simplify all usages
- radeonsi: clear per-context buffers at the end of si_create_context
- radeonsi: remove tabs
- radeonsi: don't flush in fence_server_sync

- ac/gpu_info: fix num_physical_sgprs_per_simd for gfx10
- radeonsi: fix NGG culling for Wave64
- radeonsi: always use Wave32 for GS fast launch, because Wave64 hangs
- radeonsi: always use Wave64 for HS/GS/VS shader stages (except GS fast launch)
- radeonsi: don't try to enable NGG culling for GS
- radeonsi: add a debug option to enable NGG culling for tessellation
- glsl: make print_type non-static for debugging
- glsl: print precision qualifiers in IR dumps
- glsl: print constant initializers
- glsl: fix the type of ir_constant_data::u16
- glsl: fix evaluating float16 constant expression matrices
- glsl: run validate_ir_tree if GLSL_VALIDATE=1 regardless of the build config
- glsl: validate more stuff
- glsl: convert reusable lower_precision util code into helper functions
- glsl: remove the return type from lower_precision
- glsl: cleanups in lower_precision
- glsl: flatten a tautological conditional in lower_precision
- glsl: don't lower precision of textureSize
- glsl: don't lower builtins to mediump that don't allow it
- glsl: lower builtins to mediump that ignore precision of certain parameters
- glsl: lower builtins to mediump that always return mediump or lowp
- glsl: add capability to lower mediump array types
- glsl: lower mediump temporaries to 16 bits except structures (v2)
- gallium: add PIPE_SHADER_CAP_GLSL_16BIT_TEMPS for LowerPrecisionTemporaries
- Revert "ac/surface: require that gfx8 doesn't have DCC in order to be displayable"
- glsl: don't validate array types in ir_dereference_variable
- radeonsi: prevent a gfx10_ngg_calculate_subgroup_info failure for TES+NGG GS
- radeonsi: add missing initialization of registers
- radeonsi/gfx10: set the correct value for OFFCHIP_BUFFERING
- radeonsi: sort registers in si_emit_initial_compute_regs according to GPU gen
- radeonsi: sort registers in si_init_cs_preamble_state according to GPU gen
- ac: add helper ac_get_register_name
- ac: add tables for CP register shadowing
- winsys/amdgpu: make amdgpu_bo_unmap non-static
- radeonsi: make cs_preamble_state optional
- radeonsi: reorder code in update_gs_ring_buffers and init_tess_factor_ring

- radeonsi: implement CP register shadowing
- radeonsi: add reg shadowing codepaths to GS and tess ring setup
- radeonsi: add debug code for register shadowing
- radeonsi: don't restore states at the beginning of IBs if they're shadowed
- radeonsi: set up IBs for preemption
- radeonsi: enable preemption if the kernel enabled it
- amd: rename SIENNA -> SIENNA_CICHLID
- amd: add support for Navy Flounder
- amd: enable displayable DCC for everything newer than Navi1x
- radeonsi: disable SDMA on gfx9
- radeonsi: reorder NIR optimizations
- radeonsi: call nir_split_array_vars/shrink_vec_array_vars/opt_find_array_copies
- glsl: lower_precision - fix assertion failure with dereferences of constants
- glsl: fix constant expression evaluation for 16-bit types
- glsl: don't lower atomic functions to mediump
- glsl: don't create conversion opcodes for array types
- glsl: don't lower to mediump for desktop OpenGL
- glsl: improve precision determination for calls
- Revert "radeonsi: honor a user-specified pitch on gfx10.3"
- radeonsi: use correct wave size in gfx10_ngg_calculate_subgroup_info
- radeonsi: use the same units for esgs_ring_size and ngg_emit_size
- radeonsi: increase minimum NGG vertex count requirement per workgroup on gfx 10.3
- radeonsi: fix applying the NGG minimum vertex count requirement
- radeonsi: don't count unusable vertices to the NGG LDS size
- radeonsi: add a common function for getting the size of gs_ngg_scratch
- radeonsi: remove the NGG hack decreasing LDS usage to deal with overflows
- radeonsi: various fixes for gfx10.3
- radeonsi: disable NGG culling on gfx10.3 because of hangs
- st/mesa: don't generate NIR for ARB_vp/fp if NIR is not preferred
- radeonsi: fix tess levels coming as scalar arrays from SPIR-V
- gallivm: fix build on LLVM 12 due to LLVMAddConstantPropagationPass removal
- ac/llvm: fix unaligned VS input loads on gfx10.3
- Revert "ac: generate FMA for inexact instructions for radeonsi"

Marek Vasut (3):

- etnaviv: Disable seamless cube map on GC880
- etnaviv: Remove etna_resource_get_status()

- etnaviv: Add lock around pending_ctx

Mario Kleiner (1):

- vulkan/wsi: Really terminate DRM lease in wsi_release_display().

Mathias Fröhlich (2):

- st/mesa: Move _NEW_FRAG_CLAMP to NewFragClamp driver flag.
- mesa: set _NEW_FRAG_CLAMP only when needed

Matt Turner (22):

- intel/compiler: Drop opt_sampler_eot()
- intel/tools: Remove unnecessary reg number checking
- intel/tools: Drop srctype from ipreg
- intel/tools: Require explicit regions/types for special regs
- intel/tools: Disallow control subregisters > 3
- intel/tools: Add assembler tests for the cr0 register
- intel/compiler: Add assert that set bits are within mask
- intel/compiler: Don't emit no-op cr0 changes
- intel/tools: Fix typos
- intel/tools: Remove stray newline
- intel/tools: Don't allow empty type specifier
- intel/tools: Simplify register type handling
- intel/tools: Make swizzle an integer
- intel/tools: Make writemask an integer
- intel/tools: Simplify immediate handling
- intel/tools: Simplify dstregion
- intel/compiler: Relax SENDS regioning assertions
- intel/tools: Pass integers, not enums, to stride()
- intel/tools: Manually set ARF register file/nr/subnr
- intel/tools: Don't hardcode notification register
- intel/tools: Simplify notification register handling
- intel/tools: Test notification subregisters

Mauro Rossi (17):

- android: iris: add iris_seqno.{c,h} to Makefile.sources
- freedreno/drm: android: add libfreedreno_registers static dependency
- freedreno: android: add adreno-pm4-pack.xml.h generation to android build
- android: util: fix build for GL4.1 support
- android: svga: fix build for GL4.1 support
- android: aco: add aco_ir.cpp to Makefile.sources

- android: nvir/gv100: update sources in Makefile.sources
- android: freedreno: add fd5_layout.c to Makefile.sources
- android: freedreno/ir3: add missing generated sources and rules
- android: freedreno/ir3: simplify generated sources rules
- android: panfrost/encoder: add libmesa_nir static dependency
- radv: fix build on Android 7 (v2)
- android: freedreno/registers: fix generated headers rules
- android: freedreno/ir3: fix include paths
- android: freedreno/common: add support for libfreedreno_common static
- android: freedreno: move a2xx disasm out of gallium
- android: freedreno/common: add libmesa_git_sha1 static dependency

Michel Dänzer (38):

- gitlab-ci: Use YAML anchor for llvmpipe paths in virgl rules
- gitlab-ci: Update to current templates
- gitlab-ci: Move down container_pre_build.sh invocation in x86_build.sh
- gitlab-ci: Add Debian testing repository for x86_build image
- gitlab-ci: Install WINE from Debian testing
- gitlab-ci: Move lib{drm,pciaccess}-dev cross packages out of loop
- gitlab-ci: Install g++-mingw-w64-x86-64-win32 instead of mingw-w64
- Revert "ac,radeonsi: fix compilations issues with LLVM 11"
- Revert "gallium/gallivm: fix compilation issues with llvm 11"
- gitlab-ci: Enable -Werror in *meson-s390x* job
- gitlab-ci: Also list arm/x86_build in needs: of test jobs
- gitlab-ci: x86_test-base image as common base for x86_test-gl/vk
- gitlab-ci: Pull in GCC 9 from Debian testing in x86_test-gl/vk images
- gitlab-ci: Move LLVM/clang 6/7 packages to the x86_build_old image
- gitlab-ci: Use Debian 10 wine-development packages
- gitlab-ci: Stop using packages from Debian testing
- gitlab-ci: Move meson back to x86_test-gl/vk ephemeral packages lists
- gitlab-ci: Add x86_build-base docker image
- gitlab-ci: Use separate docker images for cross builds
- loader/dri3: Add dri3_wait_for_event_locked full_sequence out parameter
- loader/dri3: Use dri3_wait_for_event_locked in loader_dri3_wait_for_msc
- loader/dri3: Check for window destruction in dri3_wait_for_event_locked
- gitlab-ci: Automatically run pipelines for Marge Bot pre-merge only
- gitlab-ci: Use rules: instead of except:/only: for test-docs job

- gitlab-ci: Extend .ci-run-policy template for docs jobs
- gitlab-ci: Do not create the "success" job when the test-docs job exists
- ci: Use "when: always" for pages job
- ci: Move deploy stage between container & build stages
- Revert "loader/dri3: Check for window destruction in dri3_wait_for_event_locked"
- gitlab-ci: Remove indirect dependencies from needs:
- gitlab-ci: Drop dependencies:
- Revert https://gitlab.freedesktop.org/mesa/mesa/-/merge_requests/4580
- gitlab-ci: Fix "triggered by Marge for a merge request" rule
- gitlab-ci: Only trigger test-docs job automatically for MRs
- ci: Use FDO_CI_CONCURRENT in run-shader-db.sh as well
- ci: Do not mark container / pages jobs as interruptible
- ci: Use half as many parallel softpipe / virgl test jobs
- ci: Use ignore_scheduled_pipelines anchor in .radeonsi-rules

Michel Zou (1):

- swr: fix build with mingw

Mike Blumenkrantz (73):

- zink: explicitly zero some arrays in ntv
- zink: add SpvId returns to a couple ntv functions
- zink: flush active queries on destroy and free query object
- zink: fix vkCmdResetQueryPool usage
- zink: reset query on-demand when beginning a new query from resume
- zink: always use logical eq ops in ntv with 1bit inputs
- zink: track program usages for each shader
- zink: emit interpolation decorations for ntv outputs
- zink: handle more glsl->spirv builtin translation
- zink: rework input/output location emission
- zink: use '2' variants for device props/feats, check features for ext enabling
- zink: add spirv builder util functions for emitting xfb decorations
- zink: add spirv_builder methods for OpVectorExtractDynamic and OpVectorInsertDynamic
- zink: implement streamout and xfb handling in ntv
- zink: implement transform feedback support to finish off opengl 3.0
- zink: set PIPE_CAP_VIEWPORT_TRANSFORM_LOWERED and remove POS special casing
- zink: switch to passing VkPhysicalDeviceFeatures2 in VkDeviceCreateInfo
- zink: enable xfb extension in screen creation
- zink: use int assignment for vk int type

- zink: use correct define value for reserved slot count in ntv
- zink: clamp VkImageCreateInfo.arrayLayers to 1 for image resource creation
- zink: unify code for setting resource barriers
- zink: handle signed and unsigned min/max ops in ntv
- zink: add ult handling for ntv
- zink: add bitfield_reverse handling to ntv
- zink: lower byte/word extract ops in nir
- zink: handle ixor in ntv
- zink: handle isign alu in ntv
- zink: set lower_mul_high and lower_rotate in ntv compiler options
- zink: use OpFUnordNotEqual for nir_op_fne
- zink: set lower_uadd_carry in nir options
- zink: implement Vk_EXT_index_type_uint8
- nir: add lowering pass for clip plane enabling
- st/program: use nir_lower_clip_disable instead of nir_lower_clip_vs conditionally
- nir: add lowering pass for fragcolor -> fragdata
- zink: translate gl_FragColor to gl_FragData before ntv to fix multi-rt output
- u_prim_restart: handle user buffers in util_translate_prim_restart_ib()
- nir: allow nir_lower_point_size_mov to run in geometry shader
- nir: allow nir_lower_clip_halfz to run in geometry shaders
- zink: rework query handling
- zink: use #define for number of queries per-pool
- zink: only stall during query destroy for xfb queries
- zink: properly handle query pool overflows
- zink: only reset query pool on query end if current batch isn't in renderpass
- zink: use right vulkan type for GL_PRIMITIVES_GENERATED queries
- zink: handle ntv case of nested loop instructions more permissively
- zink: add lengthy comment and remove assert from discard_if ntv pass
- zink: use type of src[0] for ntv store and load ops
- zink: try copy_region hook for blits where we can't do a regular blit or resolve
- zink: block vkCmdBlitImage usage for multi sampled blits
- zink: block resolve blits for depth/stencil buffers
- zink: handle empty attachments
- zink: try to handle multisampled null buffers
- zink: enable tgsi texcoord pipe cap
- zink: destroy gfx program when a shader is freed

- zink: destroy descriptor pools on context destroy
- zink: free pipeline cache during program destroy
- zink: free all ntv allocations after creating shader module
- zink: use helper function to handle uvec/bvec types
- zink: handle texelFetchOffset with offsets
- zink: add some asserts for building access chains in ntv
- zink: omit Lod image operand in ntv when not using an image texture dim
- nir: allow lower_psiz_mov to run in tessellation stages
- nir_ allow nir_lower_clip_halfz to run in tess eval shader
- u_prim_restart: handle indirect draws
- zink: add extension loading framework for spirv builder
- zink: implement VK_EXT_robustness2
- zink: clamp PIPE_SHADER_CAP_MAX_SHADER_BUFFERS to PIPE_MAX_SHADER_BUFFERS
- zink: handle VK_EXT_vertex_attribute_divisor setup
- zink: store valid timestamp bits onto zink_screen
- zink: implement handling for VK_EXT_calibrated_timestamps
- u_prim_restart: add inline function for getting restart index based on index size
- zink: reorder create_stream_output_target to fix failure case leak

Miklós Máté (1):

- docs: add some missing stuff to sourcetree.rst

Nanley Chery (18):

- iris: Drop can_fast_clear_color's format parameter
- iris: Remove the CCS_D fallback
- iris: Avoid fast-clear with incompatible view
- iris: Disable sRGB fast-clears for non-0/1 values
- intel: Add ISL_AUX_USAGE_GEN12_CCS_E
- iris: Don't support sRGB + Y_TILED_CCS on gen9
- iris: Use ISL_AUX_USAGE_GEN12_CCS_E on gen12
- isl/drm: Support I915_FORMAT_MOD_Y_TILED_GEN12_RC_CCS
- gallium/dri2: Support I915_FORMAT_MOD_Y_TILED_GEN12_RC_CCS
- iris: Handle importing aux-enabled surfaces on TGL
- iris: Refactor modifier_is_supported for gen12
- iris: Support I915_FORMAT_MOD_Y_TILED_GEN12_RC_CCS
- iris: Zero the add-on clear color BO on import
- dri_util: Update internal_format to GL_RGB8 for MESA_FORMAT_B8G8R8X8_UNORM
- iris: Don't call SET_TILING for dmabuf imports

- gallium/dri2: Report correct YUYV and UYVY plane count
- iris: Fix aux assertion in resource_get_handle
- blorp: Fix alignment test for HIZ_CCS_WT fast-clears

Nataraj Deshpande (3):

- anv: Limit vulkan version to 1.1 for Android
- anv: Disable extensions based on Android versions
- dri_util: Update internal_format to GL_RGB8 for MESA_FORMAT_R8G8B8X8_UNORM

Neha Bhende (6):

- util: Initialize pipe_shader_state for passthrough and transform shaders
- util: Add util functionality for GL4.1 support
- winsys/drm: Add GL4.1 support in drm winsys
- svga/include: Headers for GL4.1 support
- svga: Add GL4.1(compatibility profile) support in svga driver
- svga: Performance fixes

Neil Armstrong (2):

- Revert "CI: Disable Lima jobs due to lab unhealthiness"
- Revert "CI: Disable Panfrost Mali-T820 jobs"

Neil Roberts (26):

- nir/scheduler: Handle nir_intrinsic_load_per_vertex_input
- v3d: Remove unused member of v3d_compile
- nir/schedule: Store a pointer to the scoreboard in nir_deps_state
- nir/scheduler: Add an option to specify what stages share memory for I/O
- v3d: Let scheduler know GS doesn't have shared I/O memory
- gallium: Add pipe cap for primitive restart with fixed index
- mesa: Add PrimitiveRestartFixedIndex to gl_constants
- v3d: Disable PIPE_CAP_PRIMITIVE_RESTART
- v3d: Add missing macro for stvpmd instruction
- v3d: Use stvpmd for non-uniform offsets in GS
- compiler: Add a system value for the line coord
- v3d: Implement the line coord intrinsic
- nir: Add intrinsics for the line width
- v3d: Handle the line width intrinsics
- v3d: Add a lowering pass for line smoothing
- v3d: Enable perpendicular line caps when line smoothing
- broadcom/qpu: set VC5_QPU_RADDR_A out of the switch at _pack_branch
- v3d/compiler: Fix sorting the gs and fs inputs

- v3d/compiler: Lower geometry output store base into offset src
- nir/scheduler: Move nir_scheduler to its own header
- nir/schedule: Store a pointer to the options struct in scoreboard
- nir/schedule: Add a callback for backend-specific dependencies
- v3d: Mark scheduling dependency for prim id and first output
- nir/schedule: Add an option for a fallback scheduling algorithm
- v3d: Changed v3d_compile:failed to an enum
- v3d: Retry with the fallback scheduler when RA fails

Oschowa (5):

- radv: Don't take absolute value of unsigned type.
- aco: Don't declare 'Block' as class, but define as struct.
- aco: Don't std::move temporary object.
- aco: Use correct reference type in for-range-loop.
- radv: Explicitly cast TIMESTAMP_NOT_READY value to uin32_t where needed.

Pablo Saavedra (5):

- ci: TRACES_DB_PATH and RESULTS_PATH defined as relative paths
- ci: ArgumentParser receives the args from the main parameters
- ci: Migrate tracie tests done in shell script to pytest
- ci: Split test_tracie_skips_traces_without_checksum in separate cases
- ci: Fix TypoError error when traces in traces.yml is an empty list

Pavel Asyutchenko (1):

- vulkan/overlay: fix crash on destroying NULL swapchain

Peter Seiderer (3):

- vc4_bufmgr: fix time_t printf
- pan_bo.h: add time.h include for time_t
- v3d_bufmgr: fix time_t printf

Pierre Moreau (4):

- clover/nir: Check the result of spirv_to_nir
- clover/api: Address missing braces for subobj init
- clover: Address unnecessary copy warnings
- clover/spirv: Remove unused tuple header

Pierre-Eric Pelloux-Prayer (62):

- radeonsi: fix export count
- mesa: add gl_coontext::ForceIntegerTexNearest
- driconf: add force_integer_tex_nearest option
- radeonsi: add workaround for issue 2647

- radeonsi: don't print gs_copy_shader stats for shaderdb
- glsl: init gl_FragColor if zero_init=true
- glsl: rework zero initialization
- glsl: add a is_implicit_initializer flag
- mesa: extend GLSLZeroInit semantics
- gallium: add a new cap PIPE_CAP_GLSL_ZERO_INIT
- ac/nir: export some undef as zero
- ac/surface: remove shadowing declaration
- amdgpu/radeon: add secure api
- radeonsi: add AMD_DEBUG=tmz option
- radeon: add RADEON_CREATE_ENCRYPTED flag
- radeonsi: allocate framebuffer texture as secure when using tmz
- amdgpu: add encrypted slabs support
- radeonsi: force using staging texture when uploading to secure texture
- radeonsi/sdma: implement tmz support
- gallium: PIPE_RESOURCE_FLAG_ENCRYPTED
- radeonsi: add support for PIPE_RESOURCE_FLAG_ENCRYPTED
- amdgpu: use AMDGPU_IB_FLAGS_SECURE when requested
- radeonsi: determine secure flag must be set for gfx IB
- radeonsi: do not use cmask with encrypted texture
- amd/addrlib: fix forgotten char -> enum conversions
- radeonsi: fix inversed arguments in si_test_gds_memory_management
- amdgpu: fix unitialized variable
- radeonsi/sdma: remove useless compare
- radeonsi/drirc: enable zerovram option for 7 Days to Die
- winsys/radeon: do not cast bo->va as void*
- radeonsi: add return value to gfx10_ngg_calculate_subgroup_info
- radeonsi/ngg: try GS multi-cycling mode if default mode failed
- ac/surface: set SCANOUT if surf->is_displayable
- ac/surface: fix epitch when modifying surf_pitch
- ac/llvm: load 1 byte at a time if unaligned on gfx10
- st/mesa: make texture views inherit compressed_data storage
- radeonsi: bump SI_NUM_SHADER_BUFFERS to 32
- st/mesa: do not clear NewDriverState for inactive states
- glsl: reject size1x8 for image variable with floating-point data types
- ac/llvm: remove the -1 hack from ac_atomic_inc_wrap

- glsl: don't expose imageAtomicIncWrap for signed image
- glsl: only allow 32 bits atomic operations on images
- glsl: declare gl_Layer/gl_ViewportIndex/gl_ViewportMask as vs builtins
- st/mesa: set compressed_data to NULL when freed
- bin/symbols-check.py: add –ignore-symbol argument
- ac/llvm: export ac_init_llvm_once in targets
- mesa: rename _mesa_free_errors_data
- mesa: add bool param to _mesa_free_context_data
- mesa/st: release debug_output after destroying the context
- ac/surface: adapt surf_size when modifying surf_pitch
- radeonsi: adjust epitch for PIPE_FORMAT_R8G8_R8B8_UNORM
- radeonsi: extend workaround for KHR-GL45.texture_view.view_classes on gfx9
- ac/llvm: handle static/shared llvm init separately
- mesa/st: introduce PIPE_CAP_NO_CLIP_ON_COPY_TEX
- radeonsi: enable PIPE_CAP_NO_CLIP_ON_COPY_TEX
- ac/llvm: add option to clamp division by zero
- radeonsi,driconf: add clamp_div_by_zero option
- radeonsi: use radeonsi_clamp_div_by_zero for SPECviewperf13, Road Redemption
- glsl: fix per_vertex_accumulator::fields size
- r600/uvd: set dec->bs_ptr = NULL on unmap
- radeon/vcn: set dec->bs_ptr = NULL on unmap
- mesa: fix glUniform* when a struct contains a bindless sampler

Pierre-Loup A. Griffais (2):

- radv: fix null descriptor for dynamic buffers
- radv: fix vertex buffer null descriptors

Qiang Yu (6):

- radeonsi: remove emacs style config file
- panfrost: don't always build bifrost_compiler
- radeonsi: fix syncobj wait timeout
- radeonsi: fix user fence space when MCBP is enabled
- radeonsi: fix max syncobj wait timeout
- radeonsi: fix user fence GPU address

Rafael Antognolli (8):

- intel: Store the aperture size in devinfo.
- intel/isl: Update mocs for DG1
- intel/l3: Return the URB size from devinfo for DG1

- intel/devinfo: Add function to check for DRM_I915_GEM_GET_TILING.
- iris/bufmgr: Do not use map_gtt or use set/get_tiling on DG1
- anv/dg1: Don't use SET_TILING kernel uapi.
- iris: Align last_seqnos to 64 bits.
- anv: Align "used" attribute to 64 bits.

Rhys Kidd (5):

- nv50_2d: regenerate envytools-based rnndb headers
- nv50_2d,nvc0_2d: Document SET_PIXELS_FROM_MEMORY_SAFE_OVERLAP from rnndb
- nvc0_2d: Document SET_PIXELS_FROM_MEMORY_CORRAL_SIZE from rnndb
- nvc0: fix macro define for NVE4_COPY()
- nvc0: add documentation for nve4+ (Kepler) COPY class

Rhys Perry (174):

- aco: remove use of f-strings
- aco: add message to static_assert
- nir: add missing group_memory_barrier handling
- compiler/spirv: flag nclamp/nmin/nmax as exact
- nir: make fsat return 0.0 with NaN instead of passing it through
- docs: add src/amd/ to sourcetree.html
- docs/envvars: document ACO_DEBUG
- docs/envvars: update RADV_FORCE_FAMILY
- aco: simplify consecutive ordered vmem/lds writes optimization
- aco: fix consecutively written vgprs from vmem instructions
- aco: mark phi definitions as last-seen phi operands
- aco: consider affinities when creating v_mac_f32
- aco: improve phi affinities with p_split_vector
- aco: split operations that use a swap's definition
- aco: fix disassembly with LLVM 11
- nir/opt_if: run opt_peel_loop_initial_if after all other optimizations
- nir/opt_if: use nir_src_as_bool in opt_peel_loop_initial_if helper
- aco: fix typo in insert_waitcnt's kill()
- nir: fix lowering to scratch with boolean access
- aco: fix interaction with 3f branch workaround and p_constaddr
- aco: consider SDWA during value numbering
- aco: check instruction format before waiting for a previous SMEM store
- aco: preserve more fields when combining additions into SMEM
- aco: don't reorder barriers in the scheduler

- aco: fix 64-bit shared_atomic_exchange
- docs: add missing "shader_" in VK_KHR_shader_subgroup_extended_types
- radv: set keep_statistic_info with RADV_DEBUG=shaderstats
- ac/gpu_info, radv: set max_wave64_per_simd to 20 on GFX10
- aco: use v_xor3_b32
- aco: validate instructions reading/writing upper halves/bytes
- aco: p_extract_vector in 64-bit u2f16/i2f16
- aco: allow reading/writing upper halves/bytes when possible
- aco: prefer 4-byte aligned definitions
- aco: add Info::{operand_size,definition_size}
- aco: use Info::definition_size instead of definition's regclass
- aco: fix moving sub-dword values out of a register for a fixed definition
- aco: use num_opcodes instead of last_opcode
- aco: improve code for f2{i,u}{8,16}
- aco: use p_as_uniform in emit_vop1_instruction
- aco: add and set precise flag
- aco: create mads when signed zeros should be preserved
- aco: try to use fma instead of mad when denormals are enabled
- aco: create 16-bit mad/fma
- aco: update comment about preserving fp16/fp64 denormals
- aco: create 16-bit input and output modifiers
- aco: improve sub-dword check for sgpr/constant propagation
- aco: fix half_pi constant for 16-bit fsin/fcos
- aco: use 32-bit inline constants for 16-bit integer instructions
- aco: improve 8/16-bit constants
- aco: copy-propagate constants through p_extract_vector/p_split_vector
- aco: optimize 16-bit and 64-bit float comparisons
- aco: validate sub-dword pseudo instructions
- aco: add more opcodes to can_swap_operands
- aco: allow GFX9 partial writes with instructions which use opsel
- aco: improve check for moving temporaries out of fixed definitions
- aco: fix encoding of certain s_setreg_imm32_b32 instructions
- aco: fix validation error from vgpr spill/restore code
- aco: fix sub-dword opsel/sdwa checks
- aco: fix validation of opsel when set for the definition
- aco: shrink ssa_info

- aco: make ssa_info::label 64-bit
- aco: shrink mad_info
- aco: fix edge check with sub-dword temporaries
- aco: use the same regclass as the definition for undef phi operands
- radv: add new drirc option radv_no_dynamic_bounds
- radv: enable radv_no_dynamic_bounds for Path of Exile
- radv: enable radv_no_dynamic_bounds for more Path of Exile executables
- nir: slight correction to cube_face_coord constant folding
- spirv: set variables to restrict by default
- radv: fix image variable types in meta shaders
- aco: only use SMEM if we can prove it's safe
- aco: allow SMEM for some sub-dword accesses
- radv/aco,aco: allow SMEM SSBO loads on GFX6/7
- aco: fix copy+paste error in split_buffer_store
- aco: don't store byte-aligned short stores
- aco: add missing bld.scc() in byte_align_scalar()
- aco: don't create byte-aligned short loads
- aco: fix when sub-dword create_vector operand cannot be placed perfectly
- aco: improve vectorization of 8/16-bit loads/stores
- aco: ignore blocked registers when checking edges in get_reg_impl()
- aco: remove outdated assert in handle_operands()
- radv: enable zerovram for Quantic Dream games
- aco: use VOP2 version of v_mbcnt_hi_u32_b32 on GFX6/7
- aco: rework boolean phi pass
- aco: create better code for boolean phis with constant operands
- aco: optimize boolean phis with uniform selections
- aco: don't create phis with undef operands in the boolean phi pass
- aco: read 0 from inactive lanes when using dpp
- aco: optimize some masked swizzles to DPP
- aco: implement <32-bit masked_swizzle_amd
- nir/lower_subgroups: pass options struct to lower_shuffle
- nir/lower_subgroups: add lower_shuffle_to_swizzle_amd
- radv: use lower_shuffle_to_swizzle_amd
- aco: add 32-bit integer addition to can_swap_operands
- aco: fix underestimated pressure in spiller when a phi has a killed def
- aco: rewrite graph coloring in spiller

- aco: use unordered_set for spill id interferences
- aco: add add_interference() helper
- aco: use s_round_mode/s_denorm_mode
- aco: flush denormals before fp16 fabs/fneg if needed
- aco: fix nir_op_f2f16_rtne with non-default rounding modes
- aco: set tcs_in_out_eq=false if float controls of VS and TCS stages differ
- radv: enable more float_controls features
- aco: properly recognize that s_waitcnt mitigates VMEMtoScalarWriteHazard
- aco: use s_waitcnt_depctr to mitigate VMEMtoScalarWriteHazard
- spirv: don't split memory barriers
- nir/lower_int64: lower 64-bit amul
- aco: always set FI on GFX10
- radv: replace discard with demote for Quantic Dream games
- aco: implement b2i8/b2i16
- aco: be more careful combining additions that could wrap into loads/stores
- aco: allow overflow for some SMEM instructions
- aco: add NUW flag
- nir: add nir_unsigned_upper_bound and nir_addition_might_overflow
- aco: use nir_addition_might_overflow to combine additions into SMEM
- aco: move some setup code into helpers
- aco: make validate() usable in tests
- aco: print ACO IR before scheduling instead of after
- radv: fix invalid conversion warnings in vk_format.h
- aco: fix copy of uninitialized boolean
- aco: fix includes in aco_ir.cpp
- aco: add missing add_to_hazard_query
- aco: rework barriers and replace can_reorder
- radv/aco,aco: use scoped barriers
- aco: consider intrinsic access in visit_{load,store}_image
- nir,radv/aco: add and use pass to lower make available/visible barriers
- aco: enable value numbering of s_buffer_load_*
- aco: use storage_scratch
- aco: improve sync_info for TCS output stores
- aco: improve workgroup-scope and lower vmem/smem barriers
- aco: create acq+rel barriers instead of acq/rel
- nir/load_store_vectorize: fix indentation

- ac/nir: implement scoped_barrier
- radv: use scoped barriers
- aco: remove isel for GLSL-style barriers
- aco: add framework for unit testing
- aco: add a few tests for the assembler and optimizer
- aco: add framework for testing isel and integration tests
- ci: enable ACO tests
- aco/tests: add tests for sub-dword swaps
- aco: optimize swizzled SALU 8/16-bit conversions
- aco: fix waitcnt insertion on GFX10.3
- aco: don't create v_mad_f32 on GFX10.3
- aco: update bug workarounds for GFX10_3
- aco: fix max_waves_per_simd on Polaris, VegaM and GFX10.3
- aco: update vgpr_alloc_granule for GFX10.3
- aco: implement subgroup shader_clock on GFX10.3
- aco: update aco_opcodes.py for GFX10.3
- aco: disable SMEM stores on GFX10.3
- aco: replace MADs in isel with FMA on GFX10.3
- spirv: set ACCESS_COHERENT for ssbo/global/image atomic load/store
- radv/aco: enable VK_KHR_memory_model
- ac/nir: consider an image load/store intrinsic's access
- ac/nir: fix coherent global loads/stores
- radv/llvm: enable VK_KHR_memory_model
- aco: fix C++11/C++14 compilation
- aco: set constant_data_offset correctly in the case of merged shaders
- aco: don't move memory accesses to before control barriers
- aco: fix non-rtz pack_half_2x16
- aco: consider branch definitions in spiller
- aco: don't consider the first partial spill if it's the wrong type
- aco: don't fix break condition for break+discard to exec
- aco: fix regclass checks when fixing to vcc/exec with Builder
- aco: fix spills_entry heuristic for branch blocks in init_live_in_vars()
- aco: keep loop live-through variables spilled
- aco: reserve 2 sgprs for each branch
- aco: create long jumps
- aco: fix byte_align_scalar for 3 dword vectors

- aco: fix one-off error in Operand(uint16_t)
- nir/opt_if: fix opt_if_merge when destination branch has a jump
- aco: fix v_writelane_b32 with two sgprs
- aco: don't apply constant to SDWA on GFX8
- radv: initialize with expanded cmask if the destination layout needs it
- radv,aco: fix reading primitive ID in FS after TES

Rob Clark (265):

- util/simple_mtx: add assert_locked()
- freedreno: add screen lock wrappers
- freedreno: switch to simple_mtx
- freedreno: fix buffer import
- gallium: extract out logicop helper
- freedreno/drm: drop atomic refcnts
- freedreno/drm: inline the things
- freedreno/a6xx: small query cleanup
- freedreno/a6xx: avoid unnecessary clearing VS DP state
- freedreno/a6xx: move const state to single stateobj
- freedreno/a6xx: move scissor state to stateobj
- freedreno/a6xx: limit PROG_FB_RAST state emit
- freedreno/a6xx: limit LRZ state emit
- freedreno/a6xx: move blend-color to stateobj
- freedreno/a6xx: combine sample mask into blend state
- freedreno/a6xx: skip unnecessary MRT blend state
- freedreno/a6xx: add OUT_PKT()
- freedreno/a6xx: convert draw packet to OUT_PKT()
- freedreno/a6xx: split out const emit
- freedreno/ir3: inline const emit
- freedreno/a6xx: convert const emit to OUT_PKT()
- freedreno: scissor vs disabled scissor micro-opt
- freedreno/a6xx: more OUT_REG()
- freedreno: sync registers with envytools
- freedreno/a6xx: don't set SP_FS_CTRL_REG0.VARYING for fragcoord
- freedreno/a6xx: fix LRZ hang
- freedreno/a6xx: add some more formats
- freedreno: we don't need aligned vbo's
- freedreno/a6xx: compressed blit fixes

- freedreno/a6xx: enable tiled compressed textures
- freedreno/gmem: don't assume scissor opt when estimating # of bins
- freedreno: initialize max_scissor
- freedreno/gmem: add div_align() helper
- freedreno/gmem: add helper to dump GMEM layout
- freedreno: add gmemtool
- freedreno/gmem: relax alignment on a6xx
- freedreno/gmem: rework gmem layout algo
- freedreno/ir3: don't allow negative const_offset
- freedreno/ir3: fix indirect cb0 load_ubo lowering
- freedreno/ir3: limit # of tex prefetch by shader size
- freedreno/ir3/postsched: reset sfu_delay on sync
- freedreno/ir3/postsched: try to avoid (sy) syncs
- freedreno/ir3/sched: avoid scheduling outputs
- freedreno/ir3/sched: try to avoid syncs
- freedreno/a6xx: fix max-scissor opt
- freedreno/ir3: use const_index accessors
- nir: fix indices for ir3 ssbo_atomic intrinsics
- nir: add helper to copy const_index[]
- nir: add pass to lower disjoint wrmask's
- freedreno/ir3: use lower_wrmasks pass
- freedreno/fdperf: add dependency on generated headers
- freedreno/drm: don't pass thru 'DUMP' flag on older kernels
- freedreno/drm: handle ancient kernels
- freedreno/ir3: remove Sethi-Ullman numbering pass
- freedreno/ir3: juggle around ir3_debug_print()
- freedreno/ir3/dce: report progress
- freedreno/cf: report progress
- freedreno/ir3/cp: report progress
- freedreno/ir3/deps: report progress
- freedreno/ir3/group: report progress
- freedreno/ir3/legalize: report progress
- freedreno/ir3/postsched: report progress
- freedreno/ir3: add IR3_PASS() macro
- freedreno/ir3: move where we preserve binning pass inputs
- freedreno/ir3: be iterative

- freedreno/ir3: make foreach_src declare cursor ptr
- freedreno/ir3: make foreach_ssa_src declar cursor ptr
- freedreno/ir3: make input/output iterators declare cursor ptr
- freedreno/ir3/group: fix for half-regs
- freedreno/ir3: fix mismatched flags on split
- freedreno/ir3/cf: handle multiple cov's properly
- freedreno/ir3: fix immed type in create_addr0()
- freedreno/ir3/print: print cat2 condition
- freedreno/ir3/cp: fix cmps folding
- freedreno/ir3: fix mismatched wrmask for overlapping VS inputs
- freedreno/ir3: add simple validate pass
- freedreno/ir3: add helpers to deal with src/dst types
- freedreno/ir3/validate: add checking for types and opcodes
- freedreno/drm: disallow exported buffers in bo cache
- freedreno: add batch debugging
- freedreno: clear last_fence after resource tracking
- freedreno: handle PIPE_TRANSFER_MAP_DIRECTLY
- freedreno/gmem: make noscis debug actually do something on a6xx
- freedreno/gmemtool: make GMEM alignment per-gen
- freedreno/gmemtool: add a405
- freedreno/gmemtool: add verbose mode
- freedreno/gmem: add some asserts
- freedreno/gmem: fix nbins_x/y mismatch
- freedreno/gmem: split out helper to calc # of bins
- freedreno/a6xx: LRZ fix for alpha-test
- freedreno/a6xx: document LRZ flag buffer
- freedreno/a6xx: fix vsc assert
- nir: get_base_type() should return enum type
- nir: extract out convert_to_bitsize() helper
- nir/builder: add bitsize conversion helpers
- nir/lower_tex: fixes for fp16 yuv lowering
- freedreno/ir3: split kill from no_earlyz
- freedreno/a6xx: sync registers from envytools
- freedreno/a6xx: update depth-plane control regs
- freedreno/a6xx: re-work LRZ state tracking
- freedreno/a6xx: add early-lrz-late-z mode

- freedreno/a6xx: also consider alpha-test for ztest-mode
- freedreno/a6xx: more early-z
- freedreno/computerator: fix missing dependency on generated header
- nir/print: print tex dest type
- freedreno/ir3: add debug code to print conflicting half-regs
- freedreno/ir3: respect tex prefetch limits
- freedreno/ir3: remove RA "q-values" optimization
- freedreno/ir3: limit pre-fetched tex dest
- freedreno/ir3: unify shader create/delete paths
- freedreno/ir3: move the libdrm dependency out of shared code
- turnip: drop linking libfreedreno_drm
- freedreno/ir3: don't rely on intr->num_components
- radv: don't set num_components for non-vectorized intrinsics
- nir/builder: don't set intr->num_components
- nir/lower-atomics-to-ssbo: don't set num_components
- spriv: don't set num_components for non-vectorised intrinsics
- v3d: don't use intr->num_components for non-vectorized intrinsics
- nir/validate: validate intr->num_components
- freedreno/log-parser: fix compute times
- freedreno/sched: reset delay counters at start of block
- freedreno/ir3/validate: also check instr->address
- freedreno/ir3/cp: properly handle already-folded RELATIV
- freedreno: splitup emit_string_marker
- freedreno/a6xx: emit shader names in debug builds
- freedreno/ir3/legalize: don't allow (nopN) if (rptN)
- freedreno/ir3/print: print (r) flag
- freedreno/ir3: add test for delay slot calculation
- freedreno/ir3/delay: calculate delay properly for (rptN)'d instructions
- freedreno/ir3: add helpers to move instructions
- freedreno/ir3: delay test support for vectorish instructions
- freedreno/ir3/cp: extract valid_flags
- freedreno/ir3: add post-scheduler cp pass
- freedreno/ir3: convert regmask_t to struct
- freedreno/ir3: move mergedreg state out of reg
- freedreno/ir3: decouple regset from gpu gen
- freedreno/ir3: pass variant to postsched

- freedreno/ir3: re-work assembler API
- freedreno/ir3: make mergedregs a property of the variant
- freedreno/a6xx: set .MERGEREGS based on variant
- turnip: set .MERGEDREGS based on variant
- freedreno/computerator: MERGEDREGS update
- freedreno/ir3: update obsolete comment
- spirv: atomic_counter_read_deref is not vectorized
- spirv: drop some dead code
- glsl_to_nir: fix is_helper_invocation
- glsl_to_nir: fix shader_clock
- glsl_to_nir: fix vote_any/vote_all
- freedreno/ir3: refactor out helper to compile shader from asm
- freedreno/ir3: add accessor for const_state
- freedreno/a6xx: defer userconst cmdstream size calculation
- freedreno/ir3: move ubo_state into const_state
- freedreno/ir3: drop shader->num_ubos
- freedreno/ir3: constify shader key
- freedreno/ir3: pass variant to ir3_create()
- freedreno/ir3: convert over to ralloc
- freedreno/ir3: move num_reserved_user_consts out of const_state
- freedreno/ir3: un-embed const_state
- freedreno/ir3: move const_state back to variant
- freedreno/ir3: move output_loc to variant
- freedreno/ir3: split out ubo info from range
- freedreno/ir3: splitup get_existing_range()
- freedreno/ir3: split ubo analysis/lowering passes
- ci: remove some freedreno a6xx skips
- freedreno/ir3: add helper to determine point-coord inputs
- freedreno/a6xx: de-duplicate vinterp/vpsrepl state building
- freedreno/a6xx: use point-coord helper
- freedreno/a5xx: use point-coord helper
- freedreno/a4xx: use point-coord helper
- freedreno/a3xx: use point-coord helper
- freedreno: convert builtin blit VS prog to ureg builder
- freedreno/ir3: switch PIPE_CAP_TGSI_TEXCOORD
- freedreno: make foreach_bit() declare it's cursor

- freedreno: split out batch draw tracking helper
- freedreno: split out batch clear tracking helper
- freedreno: handle batch flush in resource tracking
- freedreno/ir3/ra: fix pre-color edge case
- freedreno/ir3: add ir3_finalize_nir()
- freedreno/ir3: move finalize_nir to pscreen hook
- freedreno/ir3: add ir3_compiler_destroy()
- freedreno/ir3: shuffle some variant fields
- freedreno/a6xx+ir3: stop generating pointless binning shaders
- freedreno/ir3: build binning variant at same time as draw variant
- freedreno/ir3: disk-cache support
- freedreno/ir3: move nir finalization to after cache miss
- freedreno/fdperf: fix print of base address
- freedreno/fdperf: better compatible string matching
- freedreno/fdperf: prefer render node
- gitlab-ci: reduce a630 runner load
- freedreno/ir3: add missing VS driver params
- freedreno/ir3: make compile fails more visible
- freedreno/a6xx: bail instead of crash for compile fails
- freedreno/ir3/ra: be better at failing
- freedreno/a6xx: don't enable early-z/lrz if no z-test
- freedreno/ir3: DCE unused arrays
- driconf: allowlist/denylist
- gitlab-ci: re-enable all a630 jobs
- freedreno: small comment re-word
- freedreno: whitespace fix
- freedreno/ir3/parser: half-precision relative regs
- freedreno/ir3: set array precision on creation
- freedreno/ir3: fix half-reg array stores
- freedreno/ir3/ra: debug msgs tweak
- freedreno/ir3/ra: assign vreg names to all array elements
- freedreno/ir3/ra: fix array conflicts for split/merged
- freedreno: sync registers from envytools
- freedreno: make gen_header.py check parent directory
- freedreno: slurp in rnndb
- freedreno: slurp in rnn

- freedreno: slurp in decode tools
- freedreno: slurp in afuc
- freedreno/rnn: warnings cleanup
- freedreno/decode: warnings cleanup
- freedreno/afuc: warnings cleanup
- freedreno: add CI for envytools tools
- freedreno/ir3: split out regmask
- freedreno: drop shader_t
- freedreno: deduplicate a3xx+ disasm
- freedreno: move a2xx disasm out of gallium
- freedreno: deduplicate a2xx disasm
- freedreno/ci: add a2xx trace to CI job
- freedreno/tools: check rnn parse status
- freedreno/rnn: split out helper to find files
- freedreno/rnn: add error helper
- freedreno/rnn: rename schema file
- freedreno/rnn: update schema for 'pos'
- freedreno/rnn: add relaxed boolean type
- freedreno/rnn: add high/low/pos to registers
- freedreno/rnn: add radix/align
- freedreno/rnn: relax Hexadecimal to HexOrNumber
- freedreno/rnn: add variants/varset to domain
- freedreno/registers/a2xx: fix validation error
- freedreno/registers/a4xx: fix validation error
- freedreno/registers/adreno_pm4: fix validation errors
- freedreno/rnn: describe copyright element in schema
- freedreno/rnn: add "addvariant" to schema
- freedreno/rnn: allow name to be optional in arrays
- freedreno/rnn: fix use-group
- freedreno/registers/mdp5: fix validation error
- freedreno/rnn: schema updates for dynamic/irregular offsets
- freedreno/rnn: add schema validation
- freedreno/rnn: headergen2 warnings cleanup
- freedreno/decode: cffdec warnings cleanup
- freedreno/ir3: add missing track_ubo_use()
- freedreno/a6xx: don't emit a bogus size for empty cb slots

- freedreno/a6xx: fixup draw state earlier
- freedreno/rnn: also look for .xml.gz
- freedreno/rnn: rework RNN_DEF_PATH construction
- freedreno/registers: add .gitignore
- freedreno/registers: split header build into subdirs
- freedreno/registers: install gzip'd register database
- freedreno/decode: move dependencies up a level
- freedreno: allow fence_fd fences to be recycled
- freedreno/ir3: ir3_cmdline updates
- freedreno/ir3: lower local_index using local_id
- glsl/lower_precision: split out const lowering
- gallium: replace 16BIT_TEMPS cap with 16BIT_CONSTS
- glsl: remove LowerPrecisionTemporaries
- glsl: don't inline intrinsics for mediump
- glsl_to_nir: fix bitfield_extract with 16-bit operands
- freedreno/registers: add some missing regs to build
- freedreno/crashdec: handle section name typos
- freedreno/a6xx: fix occlusion query with more than one tile
- freedreno: handle case of shadowing current render target
- freedreno/gmemtool: add tile_alignw/h and a650

Rohan Garg (3):

- iris: Fix documentation for _iris_batch_flush
- ci: Include trace replay support in ARM rootfses.
- gitlab-ci: Replay traces on lava devices

Roland Scheidegger (1):

- gallivm: fix half to float conversions with llvm 11

Roman Gilg (2):

- vulkan/wsi/x11: add sent image counter
- vulkan/wsi/x11: wait for acquirable images in FIFO mode

Roman Stratiienko (5):

- egl: Build surfaceless platform on Android
- Android: Fixes for Q and R
- panfrost: Android build fixes 2020 week 31
- lima: Fix lima_screen_query_dmabuf_modifiers()
- android: freedreno: Another build fix

Sagar Ghuge (3):

- iris: Use modfiy disables for 3DSTATE_WM_DEPTH_STENCIL command
- intel/compiler: Optimize integer add with 0 into mov
- intel/compiler: Remove unnecessary optimization for MUL

Samuel Pitoiset (235):

- ci: fix reporting the number of unexpected/flakes
- ci: add lists of expected failures & skipped tests for RAVEN with ACO
- aco: remove unecessary p_split_vector with v2b reg class
- radv: enable shaderInt16 unconditionally with LLVM and only GFX8+ with ACO
- radv: cleanup radv_CreateInstance()
- radv: rename radv_devices() to radv_enumerate_physical_devices()
- radv: fix a memleak if the physical device initialization failed
- radv: report INITIALIZATION_FAILED when the amdgpu winsys init failed
- radv: don't report error with other vendor DRM devices
- radv: use a linked list for physical devices
- radv: display an error message if the winsys init failed
- radv/winsys: do not count visible VRAM buffers twice in the budget
- ci: remove unused .test-radv-fossilize rule
- ci: set ACO_DEBUG=validateir,validatera global for RADV testing
- ci: run radv-fossils with Pitcairn (GFX6) and Bonaire (GFX7) too
- radv: remove the LLVM version string when ACO is used
- radv: do not print the LLVM version string twice in hang reports
- radv: report correct backend IR in hang reports when ACO is used
- aco: fix 64-bit trunc with negative exponents on GFX6
- nir: do not vectorize load/store if offset can overflow and robustness enabled
- aco: prevent invalid loads/stores vectorization if robustness is enabled
- radv: limit the Vulkan version to 1.1 for Android
- radv: handle different Vulkan API versions correctly
- radv: update the list of allowed Android extensions
- aco: optimize add/sub(a, cndmask(b, 0, 1, cond)) -> addc/subbrev_co(0, a, b)
- radv: use the common base object type for VkDevice
- radv: use the base object struct types
- radv: implement VK_EXT_private_data
- vulkan: import common code for generating extensions
- radv: use the common code for generating extensions and dispatch tables
- anv: use the common code for generating extensions and dispatch tables
- turnip: use the common code for generating extensions and dispatch tables

- radv: add a LLVM version string workaround for SotTR and ACO
- aco: remove useless check for nir_tex_src_bias
- aco: add support for texturing with clamped LOD
- ac/llvm: add support for texturing with clamped LOD
- radv: enable shaderResourceMinLod
- spirv: handle OpCopyObject correctly with any types
- radv: fix missing break in radv_GetPhysicalDeviceProperties2()
- aco: store 16-bit temporary outputs as v2b
- aco: convert 16-bit values before exporting MRTs
- aco: allow to load/store 16-bit values in VMEM for tess and geom
- aco: implement 8-bit/16-bit mov's with p_create_vector
- aco: implement 16-bit vertex fetches with tbuffer_load_format_d16_*
- aco: validate v_interp_*_f16 as VOP3 instructions instead of VINTRP
- aco: emit v_interp_*_f16 instructions as VOP3 instead of VINTRP
- aco: implement 16-bit interp
- aco: fix off-by-one error with 16-bit MTBUF opcodes on GFX10
- radv/aco: enable storageInputOutput16 on GFX9+
- aco: fix missing break in label_instruction()
- radv: fix missing break in radv_GetPhysicalDeviceFeatures2()
- radv: fix duplicated expression in ac_setup_rings()
- radv/winsys: remove useless free in radv_amdgpu_create_bo_list()
- aco: declare 8-bit/16-bit reduce operations
- aco: implement 8-bit/16-bit reductions
- aco: validate 8-bit/16-bit VGPR operands for readfirstlane/readlane/writelane
- aco: implement 8-bit/16-bit nir_intrinsic_read_first_invocation
- aco: implement 8-bit/16-bit nir_intrinsic_{shuffle,_read_invocation}
- aco: implement 8-bit/16-bit nir_intrinsic_quad_*
- aco: use a temporary SGPR for 8-bit/16-bit literal reduction identities
- aco: sign-extend the input and identity for 8-bit subgroup operations
- radv: do not return from radv_GetPhysicalDeviceFeatures2()
- radv: cleanup physical device features
- radv: remove useless assignment in build_streamout_vertex()
- spirv: add ReadClockKHR support with device scope
- aco: implement nir_intrinsic_shader_clock with device scope
- ac/nir: fix shader clock with subgroup scope
- ac/nir: implement nir_intrinsic_shader_clock with device scope

- radv: advertise shaderDeviceClock on GFX8+
- spirv: add SpvCapabilityImageGatherBiasLodAMD
- spirv: add support for bias/lod with OpImageGather
- ac/nir: add support for bias/lod with texture gather
- aco: add support for bias/lod with texture gather
- radv: add support for querying which formats support texture gather LOD
- radv: advertise VK_AMD_texture_gather_bias_lod
- spirv,radv,anv: implement no-op VK_GOOGLE_user_type
- radv/aco: enable VK_EXT_subgroup_size_control
- aco: fix register allocation for subdword instructions on GFX10
- aco: implement 8-bit/16-bit reductions on GFX10
- aco: allocate a temp VGPR for some 8-bit/16-bit reduction ops on GFX10
- aco: allow gfx10_wave64_bpermute with 8-bit/16-bit input
- aco: sign-extend input/indentity for 32-bit reduce ops on GFX10
- radv/aco: enable VK_KHR_subgroup_extended_types on GFX8+
- radv: enable zero VRAM for Doom Eternal
- radv: enable zero VRAM for all VKD3D (DX12->VK) games
- aco: implement 16-bit reduce operations on GFX6-GFX7
- aco: implement 16-bit nir_intrinsic_quad_* on GFX6-GFX7
- aco: fix subdword copies on GFX6-GFX7
- aco: sign-extend input/identity for 16-bit subgroup ops on GFX6-GFX7
- radv/aco: enable 64-bit atomic features if RADV is linked with LLVM 8
- aco: use v_bfe_u32 for unsigned reductions sign-extension on GFX6-GFX7
- aco: fix sign-extend 8-bit subgroup operations on GFX6-GFX7
- aco: fix nir_intrinsic_quad_* with 8-bit in GFX6-GFX7
- radv/aco: enable VK_KHR_shader_subgroup_extended_types on GFX6-GFX7
- ac/nir: adjust an assertion for D16 on GFX6-GFX7
- nir/lower_explicit_io: fix NON_UNIFORM access for UBO loads
- radv/llvm: expose VK_EXT_shader_demote_to_helper_invocation with LLVM 9+
- aco: implement 8-bit/16-bit conversions on GFX6-GFX7
- aco: fix alignment of vectors with 4 elements
- radv/aco: enable 8-bit/16-bit storage on GFX6-GFX7
- radv/aco: enable shaderInt16 on GFX6-GFX7
- radv/aco: enable shaderInt8 and VK_KHR_shader_float16_int8 on GFX6-GFX7
- ac/nir: fix integer comparisons with pointers
- radv: set DB_SHADER_CONTROL.CONSERVATIVE_Z_EXPORT correctly

- radv: add new drirc option radv_enable_mrt_output_nan_fixup
- aco: implement radv_enable_mrt_output_nan_fixup workaround
- radv/llvm: implement radv_enable_mrt_output_nan_fixup workaround
- radv: enable radv_enable_mrt_output_nan_fixup for RAGE 2
- ac: add ac_choose_spi_color_formats() to common code
- spirv: fix using OpSampledImage with OpUndef instead of OpType{Image,Sampler}
- aco: allow to swap operands for some 16-bit float instructions
- spirv: do not set num_components for non-vectorized mbcnt_amd intrinsic
- radv/aco: enable FP16 features/extensions on GFX9+
- radv: lower discards to demote to workaround a RDR2 game bug
- radv: make sure to set CB_SHADER_MASK correctly for internal CB operations
- radv: compute CB_SHADER_MASK from the fragment shader outputs
- radv: only requires LLVM 9 for GFX10 if not using ACO
- radv: replace == GFX10 with >= GFX10 where it's needed
- aco: replace == GFX10 with >= GFX10 where it's needed
- radv: add support for Sienna Cichlid
- radv: require LLVM 11+ for GFX 10.3 if not using ACO
- aco: fix printing ASM on GFX6-7 if clrxdisasm is not found
- aco: improve validation checks for readlane/writelane
- aco: fix printing ASM on GFX6-7 again
- gitlab-ci: stop testing RADV with LLVM
- gitlab-ci: update the list of expected CTS failures for RADV/ACO
- gitlab-ci: update the list of expected failures for Pitcairn
- radv: fix checking the return value of cs_finalize()
- gitlab-ci: add parallel-rdp fossils
- radv: lower 64-bit drcp/dsqrt/drsq for fixing precision issues
- radv: lower 64-bit dfloor on GFX6 for fixing precision issues
- gitlab-ci: add a list of expected failures for RADV/ACO on NAVI14
- gitlab-ci: set the number of Fossilize threads to 4
- gitlab-ci: append Fossilize stdout/stderr to a file to reduce spam
- gitlab-ci: attach the Fossilize log file as artifact on failure
- radv: remove the shader ballot workaround for Youngblood with LLVM
- radv: remove the load/store workaround for Monster Hunter World with LLVM
- radv: enable VK_AMD_shader_ballot on GFX6-7 with both compiler backends
- radv: adjust CB_SHADER_MASK for dual-source blending in the shader info pass
- radv: rework 8/16-bit color attachment formats detection

- radv: use SPI_SHADER_ZERO for non-written color attachments
- radv: add support for MRTs compaction to avoid holes
- radv: fix wide points and lines
- radv: fix wide lines with multisample enabled
- Revert "vulkan/wsi/x11: Ensure we create at least minImageCount images."
- radv,vulkan: add a new x11 wsi drirc workaround for DOOM Eternal
- radv: disable FMASK compression when drawing with GENERAL layout
- radv: set depth/stencil enable values correctly for the meta clear path
- radv: implement missing VK_ACCESS_MEMORY_{READ,WRITE}_BIT
- radv: store the primitive topology hardware value in the pipeline
- radv: adjust IA_MULTI_VGT_PARAM.WD_SWITCH_ON_EOP at draw time
- radv: adjust IA_MULTI_VGT_PARAM.PARTIAL_VS_WAVE at draw time
- radv: compute prim_vertex_count at draw time
- aco: fix more validation errors from vgpr spill/restore code
- radv: return VK_ERROR_DEVICE_LOST if wait-for-idle failed or expired
- radv: remove the secure compile support feature
- radv: rework dynamic viewports/scissors support
- radv: add VK_EXT_extended_dynamic_state but leave it disabled
- radv: declare new extended dynamic states
- radv: add support for dynamic cull mode and front face
- radv: add support for dynamic primitive topology
- radv: add support for dynamic and scissor count
- radv: add support for dynamic depth/stencil states
- radv: add support for dynamic vertex input binding stride
- radv: advertise VK_EXT_extended_dynamic_state
- radv: add the custom border color BO to the list of buffers
- radv: destroy the base object if VkCreateQueryPool() failed
- radv: destroy the base object if VkCreateRenderPass*() failed
- radv: destroy the base object if VkCreateImage() failed
- radv: destroy the base object if VkCreateBuffer() failed
- radv: destroy the base object if VkCreateEvent() failed
- radv: destroy the base object if VkCreateSemaphore() failed
- radv: destroy the base object if VkCreateFence() failed
- radv: destroy the base object if VkAllocateCommandBuffers() failed
- radv: destroy the base object if VkCreateInstance() failed
- radv/winsys: replace alloca() by malloc() everywhere

- radv/winsys: pass the buffer list via the CS ioctl for less CPU overhead
- radv: fix destroying the syncobj when exporting a fence FD
- radv: fix the error code when exporting a semaphore/fence fails
- radv: fix the error code when allocating a fresh imported syncobj fails
- radv: optimize creating signaled syncobj with amdgpu_cs_create_syncobj2()
- radv: split fence into two parts as enum+union.
- radv: remove one useless goto in radv_queue_submit_deferred()
- radv: improve the error messages when a CS submission failed
- radv: return better Vulkan error codes when VkQueueSubmit() fails
- radv: disable CPU caching for IBS to reduce fetch latency
- radv/winsys: always allow GTT placements on APUs
- radv: advertise VK_EXT_image_robustness
- radv: do not perform read-modify-write with the upload BO
- radv: disable CPU caching for the upload BO to reduce fetch latency
- aco: add support for nir_intrinsic_shared_atomic_fadd
- ac/nir: add support for nir_intrinsic_shared_atomic_fadd
- radv: advertise VK_EXT_shader_atomic_float
- radv: add missing return values check for some winsys calls
- radv/winsys: check more allocation failures
- radv/winsys: remove useless check when binding virtual buffers/images
- radv/winsys: return a Vulkan error code when binding virtual buffers/images
- radv/winsys: be more robust when a CS failed during recording
- radv: remove declared but unused radv_pipeline::is_dual_src
- radv: remove set but unused radv_pipeline::vertex_elements
- radv: remove outdated TODO related to PA_SU_VTX_CNTL.PIX_CENTER
- radv: emit more invariant registers as part of the initial gfx state
- radv: emit PA_SC_LINE_CNTL as part of the rasterization state
- radv: clean up VGT_SHADER_STAGES_EN emission
- radv: clean up PA_SC_CLIPRECT_RULE emission
- radv: reduce the number of allocated dwords for compute CS
- radv: clean up radv_compute_generate_pm4()
- radv: remove unnecessary radv_tessellation_state::num_patches
- radv: remove no-op si_multiwave_lds_size_workaround()
- radv: remove one unnecessary param to radv_generate_graphics_pipeline_key()
- radv: align the LDS size in calculate_tess_lds_size()
- radv: set LDS TCS size at shaders creation for GFX9+

- radv: remove unnecessary radv_tessellation_state::lds_size
- radv: clean up tessellation state emission
- radv: add radv_pipeline_init_input_assembly_state()
- radv: add radv_pipeline_generate_vgt_gs_out()
- radv: clean up adjusting MSAA state if conservative rast is enabled
- radv: clean up binning state initialization
- radv: assign pipeline gfx fields before PM4 emission
- radv: constify all radv_pipeline_generate_*() helpers
- radv: add radv_pipeline_init_shader_stages_state()
- radv: remove useless return value to radv_pipeline_scratch_init()
- radv: clean up remaining pipeline init functions
- radv: print warnings for famous RADV_PERFTEST options that no longer exist
- radv: do not honor a user-specified pitch on GFX 10.3
- radv: increase minimum NGG vertex count requirement per workgroup on GFX 10.3
- radv: fix sample shading on GFX 10.3
- radv: set BYPASS_VTX_RATE_COMBINER_GFX103 on GFX 10.3
- radv/gfx10: add missing initialization of registers
- radv: limit LATE_ALLOC_GS to prevent a GPU hang on GFX10
- radv: fix emitting the border color pointer on the compute queue
- nir/algebraic: mark some optimizations with fsat(NaN) as inexact
- aco: handle unaligned loads on GFX10.3
- spirv: fix emitting switch cases that directly jump to the merge block
- radv: fix transform feedback crashes if pCounterBufferOffsets is NULL

Satyajit Sahu (1):

- frontends/va: Handle dynamic resolution/SVC for VP9

Satyeshwar Singh (1):

- intel/dev: Don't consider all TGL SKUs as GT1 only

Serge Martin (3):

- amd/common: Fix incorrect use of asprintf instead of vasprintf
- clover: add more cl_mem_object_type to pipe_texture_target mapping
- clover: implements clEnqueueFillBuffer

Shawn Guo (1):

- freedreno/a4xx: fix *_NONE enum conversion

Simon Ser (3):

- EGL: sync headers with Khronos
- gbm: document that gbm_bo_map exposes a linear view

- radv: use bitshifts for debug enum values

SureshGuttula (1):

- radeon/vcn: Corrected vp9 ref associated data incase of target->codec is NULL

Tapani Pälli (14):

- st/mesa: destroy only own program variants when program is released
- anv: call base finish only if pass given in DestroyRenderPass
- anv: add VK_EXT_extended_dynamic_state but leave it disabled
- anv: add new dynamic states
- anv: consider dynamic state when creating pipeline
- anv: handle dynamic viewport count
- anv: add support for dynamic cull mode and winding order
- anv: add support for dynamic viewport and scissor with count
- anv: add support for dynamic primitive topology change
- anv: depth/stencil dynamic state support
- anv: dynamic vertex input binding stride and size support
- anv: toggle on VK_EXT_extended_dynamic_state
- anv: add a check for depthStencilState before using it
- anv: null check for buffer before reading size

Thong Thai (8):

- radeon: Fix whitespaces
- gallium/auxiliary/vl: Fix compute shader scaling for non-square pixels
- gallium/auxiliary/vl: Fix compute shader scale_y for interlaced videos
- frontends/va: Fix deinterlace bottom field first flag
- frontends/vdpau: Default destination rect to source rect
- radeon/vcn: add vcn 3.0 encode support
- radeonsi: use PIPE_FORMAT_P010 for 10-bit VP9 decoding
- radeon/vcn: increase render_pic_list size

Timothy Arceri (69):

- glsl: stop cascading errors if process_parameters() fails
- glsl: fix slow linking of uniforms in the nir linker
- radv: fix regression with builtin cache
- nir: add glsl_get_ifc_packing() helper
- nir: add callback to nir_remove_dead_variables()
- glsl: add can_remove_uniform() helper to the NIR linker
- glsl: remove dead uniforms in the nir linker
- glsl/spirv: remove dead uniforms in spirv nir linker

- gitlab-ci: bump piglit checkout commit
- i965: call brw_nir_lower_uniforms() after uniform linking is complete
- util: add BITSET_LAST_BIT() helper
- glsl: add struct to gather more info about uniform array access
- glsl: add update_array_sizes() helper to the NIR uniform linker
- glsl: gather uniform dereference info before main linking loop
- glsl: when NIR linker enable use it to resize uniform arrays
- glsl: fix potential slow compile times for GLSLOptimizeConservatively
- glsl: fix incorrect optimisation in opt_constant_variable()
- glsl: fix uniform array resizing in the nir linker
- glsl: small optimisation fix for uniform array resizing
- st_glsl_to_nir: fix potential use after free
- mesa: remove _mesa prefix from static function
- mesa: add _mesa_program_state_value_size() helper
- glsl: define gl_LightSource members in ARB_vertex_program order
- st/glsl_to_nir: disable st_nir_lower_builtin() when packing supported
- glsl: remove stale FIXME
- i965: add and fix fallthrough comments
- llvmpipe: add missing fallthrough comments
- gallivm: add missing break
- anv: update fallthrough comment so gcc sees it
- intel/compiler: add and fix up fallthrough comments for gcc warnings
- iris: add missing fallthrough comment
- egl: move fallthrough comment so gcc can see it
- nir: add missing break to nir_opt_access()
- mesa: fix fallthrough in glformats
- mesa: add fallthrough comments to glformats.c
- mesa: add fallthrough comments to get.c
- nir: fix implicit fallthrough warnings
- mesa: add fallthrough comments to COPY_SZ_4V()
- radeonsi: add missing fallthrough comment
- glx: add missing fallthrough comment
- glsl: move fallthrough comment to where gcc can see it
- radeon: add missing fallthrough comments
- spirv: add missing fallthrough comments
- mesa/vbo: add some missing fallthrough comments

- mesa: add missing fallthrough comment to teximage.c
- mesa: fix unintended fallthrough in glIsEnabled()
- r300: add and fix up fallthrough comments
- svga: add missing fallthrough comments
- mesa: update fallthrough comment so gcc can see it
- nv30: add missing fallthrough comment
- meson: turn on Wimplicit-fallthrough project wide
- nouveau: fix pointer-sign warning
- gitlab-ci: Enable -Werror in *meson-classic* job
- r600/radeonsi: silence zero-length-bounds gcc warnings
- radeonsi: fix SI_NUM_ATOMS
- iris: fix maybe-uninitialized warning for initial_state variable
- iris: silence maybe-uninitialized for stc_dst_aux_usage variable
- nouveau/nvc0: silence maybe-uninitialized warning
- panfrost: add some missing fallthrough comments
- panfrost: hide more unused code in bi_lower_combine.c
- panfrost: add some missing fallthrough comments to bi_pack.c
- freedreno: fix missing fallthrough comments
- v3d: remove redefine of VG(x)
- zink: fix missing fallthrough comment
- nine: remove unused var
- etnaviv: add missing fallthrough comments
- lima: add missing fallthrough comments
- lima: add missing break
- gitlab-ci: Enable -Werror in *meson-gallium* job

Timur Kristóf (4):

- aco/gfx10: Refactor of GFX10 wave64 bpermute.
- aco: Implement subgroup shuffle on GFX6-7.
- radv/aco: Always enable subgroup shuffle.
- aco: Fix emit_boolean_exclusive_scan in wave32 mode.

Tomeu Vizoso (55):

- panfrost: Emit blend descriptors on Bifrost
- panfrost: Don't leak temporary descriptors array
- pan/decode: Check for correct unknown field
- pan/decode: Use correct printf modifier for long int
- panfrost: Split bit out of format.unk3

- panfrost: Create additional BO for the checksum of imported BOs (Bifrost)
- panfrost: Add a bit more info about some tiler fields
- pan/bi: Print shaders only if BIFROST_MESA_DEBUG=shaders
- pan/decode: Trace to stderr with PANDECODE_DUMP_FILE=stderr
- panfrost: GPUs newer than G-71 don't have swizzles…
- panfrost: mali_attr_meta.unknown1 is zero on Bifrost
- panfrost: Add Bifrost texture trampoline BO to batch
- pan/decode: Properly print tripped zeroes
- virgl: Properly check for encode_stride when encoding transfers
- panfrost: Add checksum BOs to batch
- panfrost: Don't trample on top of Bifrost-specific unions
- panfrost: Handle MALI_RGB8_UNORM in panfrost_format_to_bifrost_blend
- gitlab-ci: Run more dEQP tests for virgl
- gitlab-ci: Add manual tests for Virgl using GLES on the host
- gitlab-ci: Test virgl with Khronos' OpenGL CTS
- gitlab-ci: Update CTS runner
- ci: Don't call renderdoc's ReplayController.Shutdown()
- ci: Move ARM rootfses to stable
- gitlab-ci: Build kernel drivers for a few ethernet USB dongles
- gitlab-ci: More stable URL for kernel and ramdisk artifacts, for LAVA
- gitlab-ci: Remove left-behind rules:
- gitlab-ci: Don't rebuild kernels and rootfs if they have been already built in mainline
- gitlab-ci: Run all of GLES3 tests for Panfrost
- gitlab-ci: Re-add kernels for bare-metal
- gitlab-ci: Download traces from MinIO
- gitlab-ci: Upload tracie artifacts to MinIO
- gitlab-ci: Fix needs: of the arm64 LAVA test jobs
- ci: Upload images of failed replays to MinIO
- ci: Use smaller glxgears trace
- ci: Prefix tracie artifacts with the device name
- ci: Test with more traces
- ci: Disable trace testing on Mali T760
- ci: Fix the overwriting of traces.yml for baremetal
- ci: Namespace trace artifacts to the job number
- ci: Always print status code of HTTP uploads in tracie
- ci: Print load stats after running dEQP

- ci: Fix URL for glslang
- ci: Don't ship vk-build-programs after building dEQP
- ci: Split building of libdrm to its own script
- ci: Build kernels and rootfs for x86 devices
- ci: Upload reference images for traces
- ci: Print URL to image diff when a trace replay fails
- ci: Generate MinIO credentials within LAVA jobs
- ci: Set date in LAVA DUTs from NTP servers
- ci: Build-test Panfrost tools
- ci: Upload traces' reference and actual images to MinIO
- ci: Download traces from MinIO in baremetal runs
- ci: Remove kernel module build that slipped in
- ci: Actually upload trace artifacts to MinIO for baremetal
- ci: Use a rootfs tarball for NFS root, instead of a ramdisk (for LAVA)

Tony Wasserka (4):

- nir/lower_idiv: Port recent LLVM fixes to emit_udiv
- radv: Fix various non-critical integer overflows
- aco: Fix integer overflows when emitting parallel copies during RA
- amd/common: Fix various non-critical integer overflows

Vinson Lee (25):

- freedreno: Add missing break statement.
- llvmpipe: Fix variable name.
- r600/sfn: Initialize VertexStageExportForGS m_num_clip_dist member variable.
- panfrost: Ensure final.no_colour is initialized.
- r600/sfn: Use correct setter method.
- freedreno: Add missing va_end.
- pan/bi: Initialize struct fma_op_info member extended.
- zink: Check fopen result.
- etnaviv: Fix memory leak on error path.
- panfrost: Fix printf format specifier.
- r300g: Remove extra printf format specifiers.
- vdpau: Fix wrong calloc sizeof argument.
- mesa: Fix NetBSD compiler macro.
- Switch from cElementTree to ElementTree.
- intel/genxml: Migrate from deprecated xml.etree.ElementTree getchildren.
- rbug: Fix rbug_delete_vs_state lock acquisition.

- nir: Add nir_lower_clip_disable.c to SCons build.
- util: Fix SCons build.
- util: Fix memory leaks in unit test.
- meson: Fix lmsensors warning message.
- vulkan: Fix memory leaks.
- freedreno: Fix file descriptor leak.
- svga: Fix unused printf argument.
- freedreno: Check file descriptor before write.
- panfrost: Delete debug allocated syncobj.

Yevhenii Kharchenko (1):

- st/mesa: fix corrupted texture levels, when adding more levels than expected

Yevhenii Kolesnikov (5):

- glsl: subroutine signatures must match exactly
- nvir: don't use designated initialisers in C++ code
- intel/compiler: don't propagate cmp to add if add is saturated
- mesa: change error code of *TextureSubImage* for incorreect target
- nine: fix incorrect calculation of layer count for 3D textures

jzielins (2):

- gallium/swr: Fix compilation warnings
- swr: Bump maximum 2D texture size to 16kx16k

mmenzyns (1):

- nv50: Clear nv50_ir_prog_info of dead and codegen specific variables

4.4 Mesa 20.1.9 Release Notes / 2020-09-30

Mesa 20.1.9 is a bug fix release which fixes bugs found since the 20.1.8 release.

Mesa 20.1.9 implements the OpenGL 4.6 API, but the version reported by glGetString(GL_VERSION) or glGetIntegerv(GL_MAJOR_VERSION) / glGetIntegerv(GL_MINOR_VERSION) depends on the particular driver being used. Some drivers don't support all the features required in OpenGL 4.6. OpenGL 4.6 is **only** available if requested at context creation. Compatibility contexts may report a lower version depending on each driver.

Mesa 20.1.9 implements the Vulkan 1.2 API, but the version reported by the apiVersion property of the VkPhysicalDeviceProperties struct depends on the particular driver being used.

4.4.1 SHA256 checksum

```
b251ca0769b722058986640d48f8457c596142cfbee1a83cba91b83391427382  mesa-20.1.9.tar.xz
```

4.4.2 New features

- None

4.4.3 Bug fixes

- Horizon Zero Dawn graphics corruption with with radv
- Running Amber test leads to VK_DEVICE_LOST
- [spirv-fuzz] Shader generates a wrong image
- anv: dEQP-VK.robustness.robustness2.* failures on gen12
- [RADV] Problems reading primitive ID in fragment shader after tessellation
- Substance Painter 6.1.3 black glitches on Radeon RX570
- vkCmdCopyImage broadcasts subsample 0 of MSAA src into all subsamples of dst on RADV

4.4.4 Changes

Bas Nieuwenhuizen (3):

- amd/common: Cache intra-tile addresses for retile map.
- ac/surface: Fix depth import on GFX6-GFX8.
- st/mesa: Deal with empty textures/buffers in semaphore wait/signal.

Christian Gmeiner (1):

- etnaviv: simplify linear stride implementation

Connor Abbott (1):

- nir/lower_io_arrays: Fix xfb_offset bug

Danylo Piliaiev (4):

- nir/lower_io: Eliminate oob writes and return zero for oob reads
- nir/large_constants: Eliminate out-of-bounds writes to large constants
- nir/lower_samplers: Clamp out-of-bounds access to array of samplers
- intel/fs: Disable sample mask predication for scratch stores

Dylan Baker (1):

- meson/anv: Use variable that checks for –build-id

Eric Engestrom (9):

- docs/relnotes: add sha256 sums to 20.1.8
- .pick_status.json: Update to d74fe47101995d2659b1e59495d2f77b9dc14f3d
- .pick_status.json: Update to c669db0b503c10faf2d1c67c9340d7222b4f946e
- .pick_status.json: Update to a3543adc2628461818cfa691a7f547af7bc6f0fb
- .pick_status.json: Mark 802d3611dcec8102ef75fe2461340c2997af931e as denominated
- .pick_status.json: Mark e98c7a66347a05fc166c377ab1abb77955aff775 as denominated

- .pick_status.json: Mark 6b1a56b908e702c06f55c63b19b695a47f607456 as denominated
- .pick_status.json: Mark 89401e58672e1251b954662f0f776a6e9bce6df8 as denominated
- .pick_status.json: Update to efaea653b5766427701817ab06c319902a148ee9

Erik Faye-Lund (2):

- mesa: handle GL_FRONT after translating to it
- st/mesa: use roundf instead of floorf for lod-bias rounding

Jason Ekstrand (2):

- intel/fs/swsb: SCHEDULING_FENCE only emits SYNC_NOP
- nir/liveness: Consider if uses in nir_ssa_defs_interfere

Jesse Natalie (1):

- glsl_type: Add packed to structure type comparison for hash map

Karol Herbst (1):

- spirv: extract switch parsing into its own function

Lionel Landwerlin (1):

- intel/compiler: fixup Gen12 workaround for array sizes

Marek Olšák (1):

- radeonsi: fix indirect dispatches with variable block sizes

Nanley Chery (1):

- blorp: Ensure aligned HIZ_CCS_WT partial clears

Pierre-Eric Pelloux-Prayer (3):

- mesa: fix glUniform* when a struct contains a bindless sampler
- gallium/vl: do not call transfer_unmap if transfer is NULL
- gallium/vl: add chroma_format arg to vl_video_buffer functions

Pierre-Loup A. Griffais (2):

- radv: fix null descriptor for dynamic buffers
- radv: fix vertex buffer null descriptors

Rhys Perry (2):

- radv: initialize with expanded cmask if the destination layout needs it
- radv,aco: fix reading primitive ID in FS after TES

Samuel Pitoiset (2):

- radv: fix transform feedback crashes if pCounterBufferOffsets is NULL
- spirv: fix emitting switch cases that directly jump to the merge block

4.5 Mesa 20.1.8 Release Notes / 2020-09-16

Mesa 20.1.8 is a bug fix release which fixes bugs found since the 20.1.7 release.

Mesa 20.1.8 implements the OpenGL 4.6 API, but the version reported by glGetString(GL_VERSION) or glGetIntegerv(GL_MAJOR_VERSION) / glGetIntegerv(GL_MINOR_VERSION) depends on the particular driver being used. Some drivers don't support all the features required in OpenGL 4.6. OpenGL 4.6 is **only** available if requested at context creation. Compatibility contexts may report a lower version depending on each driver.

Mesa 20.1.8 implements the Vulkan 1.2 API, but the version reported by the apiVersion property of the VkPhysicalDeviceProperties struct depends on the particular driver being used.

4.5.1 SHA256 checksum

```
df21351494f7caaec5a3ccc16f14f15512e98d2ecde178bba1d134edc899b961  mesa-20.1.8.tar.xz
```

4.5.2 New features

- None

4.5.3 Bug fixes

- Crash in ruvd_end_frame when calling vaBeginPicture/vaEndPicture without rendering anything
- khr_debug-push-pop-group_gl: ../src/util/simple_mtx.h:86: simple_mtx_lock: Assertion 'c != _SIMPLE_MTX_INVALID_VALUE' failed.
- Amber test opt_peel_loop_initial_if: Assertion failed
- Dirt Rally: Flickering glitches on certain foliage since Mesa 20.1.0 caused by MSAA
- [BRW] WRC 5 asserts with gallium nine and iris.

4.5.4 Changes

Andrey Vostrikov (1):

- egl/x11: Free memory allocated for reply structures on error

Bas Nieuwenhuizen (2):

- radv: Fix threading issue with submission refcounts.
- radeonsi: Work around Wasteland 2 bug.

Danylo Piliaiev (1):

- intel/compiler: Fix pointer arithmetic when reading shader assembly

Dave Airlie (1):

- gallivm: disable brilinear for lod bias and explicit lod.

Eric Anholt (2):

- gallium/tgsi_exec: Fix up NumOutputs counting
- freedreno: Make the pack struct have a .qword for wide addresses.

Eric Engestrom (4):

- docs/relnotes: add sha256 sums to 20.1.7
- .pick_status.json: Update to 123bdb61cc0be64ee29b63cf27565ec98c2a0ab8
- .pick_status.json: Mark c8ac01af33a5bc63822915f08f89a7dbaf7d433f as denominated
- meson: drop leftover PTHREAD_SETAFFINITY_IN_NP_HEADER

Icecream95 (1):

- pan/mdg: Fix spilling of non-32-bit types

Jason Ekstrand (3):

- intel/fs: Don't copy-propagate stride=0 sources into ddx/ddy
- spirv: Run repair_ssa if there are discard instructions
- nir: More NIR_MAX_VEC_COMPONENTS fixes

Jordan Justen (1):

- anv, iris: Set MediaSamplerDOPClockGateEnable for gen12+

Marek Olšák (1):

- Revert "ac: generate FMA for inexact instructions for radeonsi"

Michel Zou (1):

- swr: fix build with mingw

Nanley Chery (2):

- iris: Fix aux assertion in resource_get_handle
- blorp: Fix alignment test for HIZ_CCS_WT fast-clears

Pierre-Eric Pelloux-Prayer (5):

- mesa: rename _mesa_free_errors_data
- mesa: add bool param to _mesa_free_context_data
- mesa/st: release debug_output after destroying the context
- r600/uvd: set dec->bs_ptr = NULL on unmap
- radeon/vcn: set dec->bs_ptr = NULL on unmap

Qiang Yu (2):

- radeonsi: fix syncobj wait timeout
- radeonsi: fix max syncobj wait timeout

Rhys Perry (2):

- aco: fix byte_align_scalar for 3 dword vectors
- nir/opt_if: fix opt_if_merge when destination branch has a jump

Timur Kristóf (1):

- aco: Fix emit_boolean_exclusive_scan in wave32 mode.

Tony Wasserka (3):

- radv: Fix various non-critical integer overflows

- aco: Fix integer overflows when emitting parallel copies during RA
- amd/common: Fix various non-critical integer overflows

4.6 Mesa 20.1.7 Release Notes / 2020-09-02

Mesa 20.1.7 is a bug fix release which fixes bugs found since the 20.1.6 release.

Mesa 20.1.7 implements the OpenGL 4.6 API, but the version reported by glGetString(GL_VERSION) or glGetIntegerv(GL_MAJOR_VERSION) / glGetIntegerv(GL_MINOR_VERSION) depends on the particular driver being used. Some drivers don't support all the features required in OpenGL 4.6. OpenGL 4.6 is **only** available if requested at context creation. Compatibility contexts may report a lower version depending on each driver.

Mesa 20.1.7 implements the Vulkan 1.2 API, but the version reported by the apiVersion property of the VkPhysicalDeviceProperties struct depends on the particular driver being used.

4.6.1 SHA256 checksum

```
4de9dd0cb2ca367606cada421db62fbf6d276dfd533e522bfab6001ff9aba288  mesa-20.1.7.tar.xz
```

4.6.2 New features

- None

4.6.3 Bug fixes

- Road Redemption certain graphic effects rendered white color
- Intel Vulkan driver crash with alpha-to-coverage
- error: 'static_assert' was not declared in this scope
- vulkan/wsi/x11: deadlock with Xwayland when compositor holds multiple buffers
- [RADV/ACO] Death Stranding cause a GPU hung (*ERROR* Waiting for fences timed out!)
- lp_bld_init.c:172:7: error: implicit declaration of function 'LLVMAddConstantPropagationPass'; did you mean 'LLVMAddCorrelatedValuePropagationPass'? [-Werror=implicit-function-declaration]
- radv: blitting 3D images with linear filter
- <<MESA crashed>> Array Index Out of Range with Graphicsfuzz application
- Intel Vulkan driver assertion with small xfb buffer

4.6.4 Changes

Alejandro Piñeiro (2):

- v3d/packet: fix typo on Set InstanceID/PrimitiveID packet
- v3d: set instance id to 0 at start of tile

Alyssa Rosenzweig (6):

- panfrost: Fix blend leak for render targets 5-8

- panfrost: Free hash_to_temp map
- pan/mdg: Free previous liveness
- panfrost: Use memctx for sysvals
- panfrost: Free batch->dependencies
- pan/mdg: Fix perspective combination

Bas Nieuwenhuizen (1):

- radv: Fix 3d blits.

Danylo Piliaiev (3):

- glsl: Eliminate out-of-bounds triop_vector_insert
- ir_constant: Return zero on out-of-bounds vector accesses
- glsl: Eliminate assigments to out-of-bounds elements of vector

Emil Velikov (1):

- radv: restrict exported symbols with static llvm

Eric Engestrom (11):

- docs/relnotes: add sha256 sums to 20.1.6
- .pick_status.json: Update to e94c22429b64f419d9a66f04fa5ecdad33f7f5ef
- .pick_status.json: Mark 9146f596ed1e8854a2a6c9137396a902bc92946c as denominated
- .pick_status.json: Mark da6d0e3facfe0eb5c7db2d75d6992643d929caff as denominated
- .pick_status.json: Mark b5558f2d2aa738d90b9e039144ae3ca69bdf92ca as denominated
- .pick_status.json: Mark c9858fb941ce7e903f608e537b3657c946f86980 as denominated
- .pick_status.json: Mark ee77951714ff4373261befde6e84f592cc1c769c as denominated
- .pick_status.json: Mark 7c226116c6c0793d6d9a7dec52ac7cf54b82b57f as denominated
- .pick_status.json: Mark d7d7687829875e401690219d4a72458fb2bbe4de as denominated
- scons: bump c++ standard to 14 to match meson
- docs: add release notes for 20.1.7

Jason Ekstrand (5):

- clover/spirv: Don't call llvm::regularizeLlvmForSpirv
- intel/nir: Pass the nir_builder by reference in lower_alpha_to_coverage
- nir: Add a nir_metadata_all enum value
- intel/nir: Rewrite the guts of lower_alpha_to_coverage
- intel/fs: Fix MOV_INDIRECT and BROADCAST of Q types on Gen11+

Jonathan Gray (11):

- util: unbreak endian detection on OpenBSD
- util/anon_file: add OpenBSD shm_mkstemp() path
- meson: build with _ISOC11_SOURCE on OpenBSD
- meson: conditionally include -ldl in gbm pkg-config file

- util: futex fixes for OpenBSD
- util/u_thread: include pthread_np.h if found
- anv: use os_get_total_physical_memory()
- util/os_misc: add os_get_available_system_memory()
- anv: use os_get_available_system_memory()
- util/os_misc: os_get_available_system_memory() for OpenBSD
- vulkan: make VK_TIME_DOMAIN_CLOCK_MONOTONIC_RAW_EXT conditional

Lionel Landwerlin (3):

- anv: fix transform feedback surface size
- intel/perf: store query symbol name
- intel/perf: fix raw query kernel metric selection

Marek Olšák (3):

- st/mesa: don't generate NIR for ARB_vp/fp if NIR is not preferred
- radeonsi: fix tess levels coming as scalar arrays from SPIR-V
- gallivm: fix build on LLVM 12 due to LLVMAddConstantPropagationPass removal

Marek Vasut (2):

- etnaviv: Remove etna_resource_get_status()
- etnaviv: Add lock around pending_ctx

Nanley Chery (1):

- gallium/dri2: Report correct YUYV and UYVY plane count

Pierre Moreau (1):

- clover/spirv: Remove unused tuple header

Pierre-Eric Pelloux-Prayer (5):

- mesa/st: introduce PIPE_CAP_NO_CLIP_ON_COPY_TEX
- radeonsi: enable PIPE_CAP_NO_CLIP_ON_COPY_TEX
- ac/llvm: add option to clamp division by zero
- radeonsi,driconf: add clamp_div_by_zero option
- radeonsi: use radeonsi_clamp_div_by_zero for SPECviewperf13, Road Redemption

Rhys Perry (1):

- aco: fix non-rtz pack_half_2x16

Rob Clark (1):

- freedreno: handle case of shadowing current render target

Roman Gilg (2):

- vulkan/wsi/x11: add sent image counter
- vulkan/wsi/x11: wait for acquirable images in FIFO mode

Samuel Pitoiset (1):

- nir/algebraic: mark some optimizations with fsat(NaN) as inexact

Vinson Lee (1):

- vulkan: Fix memory leaks.

4.7 Mesa 20.1.6 Release Notes / 2020-08-19

Mesa 20.1.6 is a bug fix release which fixes bugs found since the 20.1.5 release.

Mesa 20.1.6 implements the OpenGL 4.6 API, but the version reported by glGetString(GL_VERSION) or glGetIntegerv(GL_MAJOR_VERSION) / glGetIntegerv(GL_MINOR_VERSION) depends on the particular driver being used. Some drivers don't support all the features required in OpenGL 4.6. OpenGL 4.6 is **only** available if requested at context creation. Compatibility contexts may report a lower version depending on each driver.

Mesa 20.1.6 implements the Vulkan 1.2 API, but the version reported by the apiVersion property of the VkPhysicalDeviceProperties struct depends on the particular driver being used.

4.7.1 SHA256 checksum

```
23bed40114b03ad640c95bfe72cc879ed2f941d0d481b77b5204a1fc567fa93c  mesa-20.1.6.tar.xz
```

4.7.2 New features

- None

4.7.3 Bug fixes

- [spirv-fuzz] SPIR-V parsing failed "src->type->type == dest->type->type"
- [RADV] commit d19bc94e4eb94 broke gamescope with Navi
- 4e3a7dcf6ee4946c46ae8b35e7883a49859ef6fb breaks Gamescope showing windows properly.
- anv: crashes in CTS test dEQP-VK.subgroups.*.framebuffer.*_tess_eval
- Mafia 3: Trees get rendered incorrectly
- radv: dEQP-VK.synchronization.op.multi_queue.timeline_semaphore.write_clear_attachments_*_concurrent fail when forcing DCC.
- Assertion failure compiling shader from Zigguart
- Panfrost locks for waiting fence when running Source engine games

4.7.4 Changes

Bas Nieuwenhuizen (6):

- radv: Do not consider layouts fast-clearable on compute queue.
- radv: When importing an image, redo the layout based on the metadata.
- radv: Use getter instead of setter to extract value.
- driconf: Support selection by Vulkan applicationName.

- radv: Override the uniform buffer offset alignment for World War Z.
- radv: Fix handling of attribs 16-31.

Christian Gmeiner (1):

- etnaviv: completely turn off MSAA

Daniel Schürmann (1):

- aco: execute branch instructions in WQM if necessary

Danylo Piliaiev (3):

- st/mesa: Treat vertex outputs absent in outputMapping as zero in mesa_to_tgsi
- anv/nir: Unify inputs_read/outputs_written between geometry stages
- spirv: Only require bare types to match when copying variables

Dave Airlie (2):

- llvmpipe: only read 0 for channels being read
- llvmpipe/cs: update compute counters not fragment shader.

Eric Engestrom (8):

- docs/relnotes: add sha256 sums to 20.1.5
- .pick_status.json: Update to a880f97d593a461bdcce27e526423a9b1d6834b4
- .pick_status.json: Mark e03622e50fcebbcc32b2fd403b1a729c73cb49d5 as denominated
- pick-ui: specify git commands in "resolve cherry pick" message
- egl/entrypoint-check: split sort-check into a function
- egl/entrypoint-check: add check that GLVND and plain EGL have the same entrypoints
- driconf: fix force_gl_vendor description
- docs: add release notes for 20.1.6

Icenowy Zheng (1):

- panfrost: signal syncobj if nothing is going to be flushed

Jason Ekstrand (2):

- anv: Advertise shaderIntegerFunctions2
- spirv: Don't emit RMW for vector indexing in shared or global

Karol Herbst (1):

- nv50/ir/nir: fix global_atomic_comp_swap

Lionel Landwerlin (1):

- anv: fix incorrect realloc failure handling

Marcin Ślusarz (1):

- intel/perf: fix performance counters availability after glFinish

Marek Olšák (2):

- radeonsi: use correct wave size in gfx10_ngg_calculate_subgroup_info
- radeonsi: fix applying the NGG minimum vertex count requirement

Nanley Chery (1):

- dri_util: Update internal_format to GL_RGB8 for MESA_FORMAT_B8G8R8X8_UNORM

Rhys Perry (1):

- aco: set constant_data_offset correctly in the case of merged shaders

Samuel Pitoiset (2):

- radv/gfx10: add missing initialization of registers
- radv: limit LATE_ALLOC_GS to prevent a GPU hang on GFX10

Tony Wasserka (1):

- nir/lower_idiv: Port recent LLVM fixes to emit_udiv

4.8 Mesa 20.1.5 Release Notes / 2020-08-05

Mesa 20.1.5 is a bug fix release which fixes bugs found since the 20.1.4 release.

Mesa 20.1.5 implements the OpenGL 4.6 API, but the version reported by glGetString(GL_VERSION) or glGetIntegerv(GL_MAJOR_VERSION) / glGetIntegerv(GL_MINOR_VERSION) depends on the particular driver being used. Some drivers don't support all the features required in OpenGL 4.6. OpenGL 4.6 is **only** available if requested at context creation. Compatibility contexts may report a lower version depending on each driver.

Mesa 20.1.5 implements the Vulkan 1.2 API, but the version reported by the apiVersion property of the VkPhysicalDeviceProperties struct depends on the particular driver being used.

4.8.1 SHA256 checksum

```
fac1861e6e0bf1aec893f8d86dbfb9d8a0f426ff06b05256df10e3ad7e02c69b  mesa-20.1.5.tar.xz
```

4.8.2 New features

- None

4.8.3 Bug fixes

- panfrost: Register allocation fails for Firefox WebRender shaders
- VRAM leak with vuilkan external memory + opengl memory objects
- Possible array out of bounds in brw_vec4_nir.cpp
- [tgl][bisected][regression][iris] failure on dEQP-EGL.functional.wide_color.pbuffer_8888_colorspace_default
- Multiply defined symbols compiling with gcc@10.1.0
- shrinking descriptor pool on intel+vulkan
- dEQP-VK.renderpass2.dedicated_allocation.attachment.1.12 fails on NAVI14

4.8.4 Changes

Alyssa Rosenzweig (2):

- pan/mdg: Mask spills from texture write
- pan/mdg: Test for SSA before chasing addresses

Bas Nieuwenhuizen (3):

- radv: Fix uninitialized variable in renderpass.
- radv: Fix host->host signalling with legacy timeline semaphores.
- mesa/st: Actually free the driver part of memory objects on destruction.

Daniel Schürmann (3):

- aco: fix scratch loads which cross element_size boundaries
- aco: don't split store data if it was already split into more elements
- aco: prevent infinite recursion in RA for subdword variables

Daryl W. Grunau (1):

- prevent multiply defined symbols

Eric Engestrom (6):

- docs/relnotes: add sha256 sums to 20.1.4
- .pick_status.json: Update to caa98246a0e180a96f3fcdcd3bfcbef0b136bc11
- .pick_status.json: Mark bd75e9923302a3d389469b7b233968576a46f4de as denominated
- .pick_status.json: Mark 1b3be07b5faf867f698668080b060a270c5f795e as denominated
- bin/gen_release_notes: automatically commit release notes
- docs: add release notes for 20.1.5

Francisco Jerez (1):

- intel/ir/gen12+: Work around FS performance regressions due to SIMD32 discard divergence.

Frank Binns (1):

- egl/dri2: only take a dri2_dpy reference when binding a new context/surfaces

Jason Ekstrand (2):

- intel/eu: Use non-coherent mode (BTI=253) for stateless A64 messages
- nir/deref: Don't try to compare derefs containing casts

Kenneth Graunke (1):

- iris: Delete shader variants when deleting the API-facing shader

Kristian Høgsberg (1):

- anv: Pass device to setup_gralloc0_usage for error reporting

Lionel Landwerlin (1):

- anv: fix descriptor set free

Marcin Ślusarz (6):

- iris: propagate error from gen_perf_begin_query to glBeginPerfQueryINTEL

- i965: propagate error from gen_perf_begin_query to glBeginPerfQueryINTEL
- util: fix possible fd leaks in os_socket_listen_abstract
- util: fix possible buffer overflow in util_get_process_exec_path
- mesa: fix out of bounds access in glGetFramebufferParameterivEXT
- intel/vec4: fix out of bounds read

Marek Olšák (4):

- ac: update register and packet definitions for preemption
- radeonsi: add missing initialization of registers
- radeonsi/gfx10: set the correct value for OFFCHIP_BUFFERING
- radeonsi: disable SDMA on gfx9

Mauro Rossi (1):

- radv: fix build on Android 7 (v2)

Yevhenii Kolesnikov (1):

- nine: fix incorrect calculation of layer count for 3D textures

4.9 Mesa 20.1.4 Release Notes / 2020-07-22

Mesa 20.1.4 is a bug fix release which fixes bugs found since the 20.1.3 release.

Mesa 20.1.4 implements the OpenGL 4.6 API, but the version reported by glGetString(GL_VERSION) or glGetIntegerv(GL_MAJOR_VERSION) / glGetIntegerv(GL_MINOR_VERSION) depends on the particular driver being used. Some drivers don't support all the features required in OpenGL 4.6. OpenGL 4.6 is **only** available if requested at context creation. Compatibility contexts may report a lower version depending on each driver.

Mesa 20.1.4 implements the Vulkan 1.2 API, but the version reported by the apiVersion property of the VkPhysicalDeviceProperties struct depends on the particular driver being used.

4.9.1 SHA256 checksum

```
6800271c2be2a0447510eb4e9b67edd9521859a4d565310617c4b359eb6799fe  mesa-20.1.4.tar.xz
```

4.9.2 New features

- None

4.9.3 Bug fixes

- Amber test leads to NIR validation failed after nir_opt_if (on spirv-fuzz shader)
- Multiple issues with Detroit Become Human
- panfrost: regression: Major stuttering and low compositor FPS with glmark2
- SPIR-V parsing fails in src/compiler/spirv/spirv_to_nir.c

- SPIR-V parsing fails in src/compiler/spirv/vtn_cfg.c
- iris driver is broken in Freedesktop 19.08

4.9.4 Changes

Alyssa Rosenzweig (3):

- panfrost: Fix fence leak
- panfrost: Fix write to free'd memory
- panfrost: Revert "Disable frame throttling"

Bas Nieuwenhuizen (1):

- meson: Add mising git_sha1.h dependency.

Danylo Piliaiev (1):

- nir/opt_if: Fix opt_if_simplification when else branch has jump

Eric Engestrom (3):

- docs/relnotes: add sha256 sums to 20.1.3
- .pick_status.json: Update to fd20e986249f88129d81353d79dd248d7664953b
- docs: add release notes for 20.1.4

Erik Faye-Lund (1):

- mesa/program: fix shadow property for samplers

Jason Ekstrand (3):

- spirv: Skip phis in unreachable blocks in the second phi pass
- spirv: Allow block-decorated struct types for constants
- intel/fs: Use the correct logical op for global float atomics

Jonathan Marek (1):

- freedreno/a2xx: fix compressed textures

Lionel Landwerlin (2):

- iris: fix fallback to swrast driver
- anv: properly handle fence import of sync_fd = -1

Luigi Santivetti (3):

- dri2: dri2_make_current() fold multiple if blocks
- dri2: do not conflate unbind and bindContext() failure
- egl/dri2: try to bind old context if bindContext failed

Pierre-Eric Pelloux-Prayer (5):

- glsl: reject size1x8 for image variable with floating-point data types
- glsl: don't expose imageAtomicIncWrap for signed image
- glsl: only allow 32 bits atomic operations on images
- st/mesa: set compressed_data to NULL when freed

- ac/surface: adapt surf_size when modifying surf_pitch

Rhys Kidd (1):

- nvc0: fix macro define for NVE4_COPY()

Rhys Perry (2):

- nir/lower_int64: lower 64-bit amul
- radv: replace discard with demote for Quantic Dream games

Samuel Pitoiset (1):

- radv: fix destroying the syncobj when exporting a fence FD

4.10 Mesa 20.1.3 Release Notes / 2020-07-08

Mesa 20.1.3 is a bug fix release which fixes bugs found since the 20.1.2 release.

Mesa 20.1.3 implements the OpenGL 4.6 API, but the version reported by glGetString(GL_VERSION) or glGetIntegerv(GL_MAJOR_VERSION) / glGetIntegerv(GL_MINOR_VERSION) depends on the particular driver being used. Some drivers don't support all the features required in OpenGL 4.6. OpenGL 4.6 is **only** available if requested at context creation. Compatibility contexts may report a lower version depending on each driver.

Mesa 20.1.3 implements the Vulkan 1.2 API, but the version reported by the apiVersion property of the VkPhysicalDeviceProperties struct depends on the particular driver being used.

4.10.1 SHA256 checksum

```
9872b8d46bee822177ffbe4292addff7bdb84cefc1fe776b8e6b2881a8362bf1  mesa-20.1.3.tar.xz
```

4.10.2 New features

- None

4.10.3 Bug fixes

- vkGetSemaphoreCounterValue doesn't update without vkWaitSemaphores calls on Intel UHD 620
- [v3d] corruption when GS omits some vertices
- [RADV] Non-precise occlusion queries return non-zero when all fragments are discarded
- [DXVK] Project Cars rendering problems
- Add fallthrough to prevent errors caused by missing break
- i965/20.1: gray rendering with torcs racing
- glBindBufferRange call seems to be ignored by one of two shader-programs on radeon cards
- [bisected][g33] piglit.spec.ext_framebuffer_object.fbo-cubemap failure
- Double unlock in rbug_context.c
- ci: Report flakes on IRC from baremetal tests

4.10.4 Changes

Bas Nieuwenhuizen (4):

- meson: Do not require shader cache for radv.
- radv: Set handle types in Android semaphore/fence import.
- radv: Always enable PERFECT_ZPASS_COUNTS.
- radv: Use correct semaphore handle type for Android import.

Christian Gmeiner (1):

- etnaviv: replace prims-emitted query

Danylo Piliaiev (1):

- iris: Fix fast-clearing of depth via glClearTex(Sub)Image

Dave Airlie (6):

- gallivm/nir: fix const loading on big endian systems
- glsl: fix constant packing for 64-bit big endian.
- gallivm/nir: fix big-endian 64-bit splitting/merging.
- llvmpipe: fix occlusion queries on big-endian.
- mesa/get: fix enum16 big-endian getting.
- draw/llvm: fix big-endian mask adjusting

Dylan Baker (1):

- mesa/swrast: use logf2 instead of util_fast_log2

Emmanuel (3):

- meson: Do not enable USE_ELF_TLS for FreeBSD
- iris: Explicitly cast value to uint64_t
- i965: Explicitly cast value to uint64_t

Emmanuel Vadot (1):

- meson: Add versioning for xvmc tracker

Eric Anholt (26):

- ci: Clean up setup of the job-specific env vars in baremetal testing.
- ci: Enable IRC flake reporting on freedreno baremetal boards.
- ci: Improve the flakes reports on IRC.
- ci: Fix the nick used in IRC reporting.
- ci: Move cross file generation to a shared script.
- ci: Autodetect whether we need cross setup in lava_arm builds.
- ci: Make cmake toolchain file for deqp cross build setup.
- ci: Make the create-rootfs more resilient.
- ci: Update versions of packages to remove from rootfses.
- ci: Switch the baremetal runner to be an x86 docker image.

- ci: Disable SMP on the a5xx boards.
- ci: Fix DEQP_CASELIST_FILTER (used by a630 noubo run)
- ci: Do an explicit NIR validation-enabled pass on freedreno a630.
- ci: Improve baremetal's logging of the job env var passthrough.
- ci: Drop double ".txt" suffix on the unexpected results file.
- ci: Drop old comment about enabling –deqp-watchdog.
- ci: Auto-detect the architecture for VK ICD filenames.
- ci: Add DEQP_EXPECTED_RENDERER support for VK tests.
- ci: Move baremetal DEQP_NO_SAVE_RESULTS setup to the yml.
- ci: Quick exit qpa extraction for non-matching qpas.
- ci: Disable the firmware loader user helper option in arm64 kernels.
- ci: Build a cheza kernel.
- ci: Add scripts for controlling bare-metal chezas.
- ci: Switch cheza (freedreno a630) testing to baremetal.
- ci: Don't build an arm_test container now that the last user is gone.
- ci: Rename x86_cross_arm_test to just arm_test.

Eric Engestrom (6):

- docs/relnotes: add sha256 sums to 20.1.2
- .pick_status.json: Update to 0ca7bd73c6f1f59dcb41ead7a3923c55040377d9
- gitlab-ci: exclude scripts that don't affect the build
- .pick_status.json: Mark 293221dddaedb410781d39fdecf3c93bb111475b as denominated
- docs: add release notes for 20.1.3
- VERSION: bump to release 20.1.3

Erik Faye-Lund (2):

- gallium/docs: fixup formatting of numbered lists
- gallium/docs: remove reference to non-existent label

Frédéric Bonnard (2):

- clover: Fix types collision between c++ and altivec
- meson: Revert commit overriding C++ standard with gnu++11 on ppc64el

Greg V (1):

- gallium,util: undef ALIGN on FreeBSD to prevent name clash

Iago Toral Quiroga (1):

- v3d/compiler: fix spill offset

Ian Romanick (1):

- nir/algebraic: Don't distrubte absolute-value into dot-products

Ilia Mirkin (3):

- freedreno/a3xx: there's no r8i/ui rb format, only rg8i/rg8ui
- freedreno/a3xx: reinstate rgb10_a2ui texture format
- freedreno/ir3: avoid applying (sat) on bary.f

Jason Ekstrand (2):

- vulkan/wsi: Don't consider VK_SUBOPTIMAL_KHR to be an error condition
- anv: Handle clamping of inverted depth ranges

Lepton Wu (1):

- mapi: x86: Fix dynamic entries in x86 tsd stubs.

Lionel Landwerlin (1):

- anv: garbage collect timeline semaphore when querying value

Marcin Ślusarz (2):

- st/mesa: fix reporting of float perf counters max value
- iris: return max counter value for AMD_performance_monitor

Marek Olšák (1):

- radeonsi: don't flush in fence_server_sync

Michel Dänzer (8):

- gitlab-ci: Use YAML anchor for llvmpipe paths in virgl rules
- gitlab-ci: Move down container_pre_build.sh invocation in x86_build.sh
- gitlab-ci: Add Debian testing repository for x86_build image
- gitlab-ci: Install WINE from Debian testing
- gitlab-ci: Move lib{drm,pciaccess}-dev cross packages out of loop
- gitlab-ci: Install g++-mingw-w64-x86-64-win32 instead of mingw-w64
- gitlab-ci: Enable -Werror in 'meson-s390x' job
- gitlab-ci: Also list arm/x86_build in needs: of test jobs

Neil Armstrong (2):

- Revert "CI: Disable Lima jobs due to lab unhealthiness"
- Revert "CI: Disable Panfrost Mali-T820 jobs"

Neil Roberts (3):

- v3d: Add missing macro for stvpmd instruction
- v3d: Use stvpmd for non-uniform offsets in GS
- v3d/compiler: Fix sorting the gs and fs inputs

Pablo Saavedra (5):

- ci: TRACES_DB_PATH and RESULTS_PATH defined as relative paths
- ci: ArgumentParser receives the args from the main parameters
- ci: Migrate tracie tests done in shell script to pytest
- ci: Split test_tracie_skips_traces_without_checksum in separate cases

- ci: Fix TypoError error when traces in traces.yml is an empty list

Pavel Asyutchenko (1):

- vulkan/overlay: fix crash on destroying NULL swapchain

Pierre-Eric Pelloux-Prayer (1):

- st/mesa: do not clear NewDriverState for inactive states

Rhys Perry (1):

- radv: enable zerovram for Quantic Dream games

Rob Clark (1):

- freedreno/fdperf: better compatible string matching

Samuel Pitoiset (3):

- Revert "vulkan/wsi/x11: Ensure we create at least minImageCount images."
- radv,vulkan: add a new x11 wsi drirc workaround for DOOM Eternal
- radv: disable FMASK compression when drawing with GENERAL layout

Timothy Arceri (6):

- gallivm: add missing break
- nir: add missing break to nir_opt_access()
- mesa: fix fallthrough in glformats
- mesa: fix unintended fallthrough in glIsEnabled()
- nouveau: fix pointer-sign warning
- radeonsi: fix SI_NUM_ATOMS

Vinson Lee (1):

- rbug: Fix rbug_delete_vs_state lock acquisition.

4.11 Mesa 20.1.2 Release Notes / 2020-06-24

Mesa 20.1.2 is a bug fix release which fixes bugs found since the 20.1.1 release.

Mesa 20.1.2 implements the OpenGL 4.6 API, but the version reported by glGetString(GL_VERSION) or glGetIntegerv(GL_MAJOR_VERSION) / glGetIntegerv(GL_MINOR_VERSION) depends on the particular driver being used. Some drivers don't support all the features required in OpenGL 4.6. OpenGL 4.6 is **only** available if requested at context creation. Compatibility contexts may report a lower version depending on each driver.

Mesa 20.1.2 implements the Vulkan 1.2 API, but the version reported by the apiVersion property of the VkPhysicalDeviceProperties struct depends on the particular driver being used.

4.11.1 SHA256 checksum

```
283dff72814c8a80ce1ff8271e3f055895d26f4da3f4362acc49193e635780cb  mesa-20.1.2.tar.xz
```

4.11.2 New features

- None

4.11.3 Bug fixes

- [RADV/GFX8] Performance drop in DOOM Eternal when "Present from compute" is enabled
- freedreno: multiple applications crash on a5xx
- [RADV] - Path of Exile (238960) - Map outline, landscape and markers are missing with the Vulkan renderer.
- ASTC texture decompression fails when using software fallback
- [BISECTED] compiling shader causes crash
- Zink + GALLIUM_HUD SIGSEGV
- If-statement body is executed for false condition

4.11.4 Changes

Bas Nieuwenhuizen (3):

- vulkan/wsi/x11: Ensure we create at least minImageCount images.
- radv/winsys: Deal with realloc failures in BO lists.
- radv: Handle mmap failures.

Daniel Schürmann (1):

- aco: fix WQM handling in nested loops

Danylo Piliaiev (1):

- st/mesa: account for "loose", per-mipmap level textures in CopyImageSubData

Eric Anholt (2):

- freedreno/ir3: Fix register allocation assertion failures.
- freedreno/ir3: Fix register allocation assertion failures.

Eric Engestrom (3):

- docs/relnotes: add sha256 sums to 20.1.0
- .pick_status.json: Update to 4fc0499049fcdb7f892f99ce7abf9d739730138e
- v3d: add missing unlock() in error path

Erik Faye-Lund (3):

- Revert "gallium/hud: don't use user vertex buffers"
- gallium/hud: don't use user vertex buffers
- mesa/main: fix inverted condition

Gert Wollny (1):

- r600/sfn: Don't set num_components on TESS sysvalue intrinsics

Jan Beich (2):

- util: enable futex usage on BSDs after 7dc2f4788288
- meson: unbreak sysctl.h detection on BSDs

Jose Maria Casanova Crespo (1):

- nir: only uniforms with dynamically_uniform offset are dynamically_uniform

Kristian Høgsberg (1):

- freedreno: Handle DRM_FORMAT_MOD_INVALID in shared code

Krzysztof Raszkowski (1):

- gallium/swr: Fix building swr with MSVC

Marek Olšák (7):

- ac/surface: don't recompute the DCC retile map for imported textures
- amd/addrlib: don't recompute DCC info for every ComputeDccAddrFromCoord call
- amd/addrlib: remove unused members of ADDR2_COMPUTE_DCC_ADDRFROMCOORD_INPUT
- ac/surface: add a wrapper structure to hold ADDR_HANDLE
- ac/surface: cache DCC retile maps (v2)
- ac/surface: don't free dcc_retile_map on failure
- ac/nir: fix 64-bit division for GL CTS

Mario Kleiner (1):

- vulkan/wsi: Really terminate DRM lease in wsi_release_display().

Pierre-Eric Pelloux-Prayer (1):

- st/mesa: make texture views inherit compressed_data storage

Rhys Perry (3):

- radv: add new drirc option radv_no_dynamic_bounds
- radv: enable radv_no_dynamic_bounds for Path of Exile
- radv: enable radv_no_dynamic_bounds for more Path of Exile executables

Samuel Pitoiset (3):

- radv: set DB_SHADER_CONTROL.CONSERVATIVE_Z_EXPORT correctly
- spirv: fix using OpSampledImage with OpUndef instead of OpType{Image,Sampler}
- radv: lower discards to demote to workaround a RDR2 game bug

Timothy Arceri (2):

- glsl: fix incorrect optimisation in opt_constant_variable()
- st_glsl_to_nir: fix potential use after free

4.12 Mesa 20.0.8 Release Notes / 2020-06-11

Mesa 20.0.8 is a bug fix release which fixes bugs found since the 20.0.7 release.

Mesa 20.0.8 implements the OpenGL 4.6 API, but the version reported by glGetString(GL_VERSION) or glGetIntegerv(GL_MAJOR_VERSION) / glGetIntegerv(GL_MINOR_VERSION) depends on the particular driver being used.

Some drivers don't support all the features required in OpenGL 4.6. OpenGL 4.6 is **only** available if requested at context creation. Compatibility contexts may report a lower version depending on each driver.

Mesa 20.0.8 implements the Vulkan 1.2 API, but the version reported by the apiVersion property of the VkPhysicalDeviceProperties struct depends on the particular driver being used.

4.12.1 SHA256 checksum

```
6cf0c010df89680f9b2bc6432ff01400031795e39bceda7535fa00af06740b6c  mesa-20.0.8.tar.xz
```

4.12.2 New features

- VK_GOOGLE_user_type on ANV and RADV.

4.12.3 Bug fixes

- iris/i965: possible regression in 20.0.5 due to changes in buffer manager sharing across screens (firefox/mozilla#1634213)
- [RADV] - Doom Eternal (782330) & Metro Exodus (412020) - Title requires 'RADV_DEBUG=zerovram' to eliminate colorful graphical aberrations.
- [RADV] - Doom Eternal (782330) & Metro Exodus (412020) - Title requires 'RADV_DEBUG=zerovram' to eliminate colorful graphical aberrations.
- NIR validation failed after glsl to nir, before function inline, wrong {src,dst}->type ?
- Mesa 20.0.7 / 20.1.0-rc4 regression, extremally long shader compilation time in NIR
- Mesa-git build fails on Fedora Rawhide
- iris/i965: possible regression in 20.0.5 due to changes in buffer manager sharing across screens (firefox/mozilla#1634213)
- Incorrect _NetBSD__ macro inside execmem.c
- Possible invalid sizeof in device.c
- mesa trunk master vulkan overlay-layer meson.build warning empty configuration_data() object
- 20.0.7: mesa still is not ready to gcc 10 default settings
- [Gen9/icl] [Bisected] [Regression] dEQP-GLES3.functional.shaders.loops.short_circuit.do_while_fragment fail
- Reproduceable i915 gpu hang Intel Iris Plus Graphics (Ice Lake 8x8 GT2)
- Double lock in fbobject.c
- [bisected] Steam crashes when newest Iris built with LTO
- freedreno: glamor issue with x11 desktops
- Deadlock in anv_timelines_wait()

4.12.4 Changes

Bas Nieuwenhuizen (3):

- radv/winsys: Remove extra sizeof multiply.
- radv: Handle failing to create .cache dir.
- radv: Provide a better error for permission issues with priorities.

D Scott Phillips (1):

- anv/gen11+: Disable object level preemption

Danylo Piliaiev (6):

- anv: Translate relative timeout to absolute when calling anv_timelines_wait
- anv: Fix deadlock in anv_timelines_wait
- meson: Disable GCC's dead store elimination for memory zeroing custom new
- mesa: Fix double-lock of Shared->FrameBuffers and usage of wrong mutex
- intel/fs: Work around dual-source blending hangs in combination with SIMD16
- glsl: inline functions with unsupported return type before converting to nir

Dave Airlie (1):

- llvmpipe: compute shaders work better with all the threads.

Dylan Baker (10):

- docs/relnotes Add sha256 sums to 20.0.7
- .pick_status.json: Update to ceae09da156309327d7ba6f4a59d3a2e9b8837d9
- .pick_status.json: Update to a887ad7c84e14fdad7907037a39e9fee9d504bf3
- .pick_status.json: Update to 4504d6374dbe2aa40af519c16765457bcbf81b84
- .pick_status.json: Update to f0c102c075f8ac76629bb34619187262ccc3e9d8
- tests: Make tests aware of meson test wrapper
- .pick_status.json: Update to e58112bc08f99861ac634ede8db0f98cd497fc14
- radonsi/si_state.c: retab
- .pick_status.json: Update to 0795241dde1507e0c6a3f9ef07c281ad4f2acf7b
- vulkan-overlay/meson: use install_data instead of configure_file

Eric Engestrom (3):

- tree-wide: fix deprecated GitLab URLs
- glapi: remove deprecated .getchildren() that has been replace with an iterator
- intel: fix gen_sort_tags.py

Erik Faye-Lund (2):

- zink: use general-layout when blitting to/from same resource
- nir: reuse existing psiz-variable

Gert Wollny (1):

- nir: lower_tex: Don't normalize coordinates for TXF with RECT

Ian Romanick (1):

- anv/tests: Don't rely on assert or changing NDEBUG in tests

Ilia Mirkin (1):

- nouveau: allow invalidating coherent/persistent buffer backings

Jan Palus (1):

- targets/opencl: fix build against LLVM>=10 with Polly support

Jason Ekstrand (6):

- anv:gpu_memcpy: Emit 3DSTATE_VF_INDEXING on Gen8+
- nir/lower_double_ops: Rework the if (progress) tree
- nir/opt_deref: Report progress if we remove a deref
- nir/copy_prop_vars: Record progress in more places
- intel/vec4: Stomp the return type of RESINFO to UINT32
- intel/fs: Fix unused texture coordinate zeroing on Gen4-5

Jonathan Marek (1):

- freedreno/a6xx: use nonbinning VS when GS is used

Joshua Ashton (1):

- radeonsi: Use TRUNC_COORD on samplers

Lionel Landwerlin (4):

- iris: fix BO destruction in error path
- i965: don't forget to set screen on duped image
- i965: fix export of GEM handles
- iris: fix export of GEM handles

Lucas Stach (1):

- etnaviv: retarget transfer to render resource when necessary

Marek Olšák (2):

- radeonsi: don't expose 16xAA on chips with 1 RB due to an occlusion query issue
- radeonsi: add a hack to disable TRUNC_COORD for shadow samplers

Marek Vasut (1):

- etnaviv: Disable seamless cube map on GC880

Michel Dänzer (1):

- util: Change os_same_file_description return type from bool to int

Nataraj Deshpande (1):

- dri_util: Update internal_format to GL_RGB8 for MESA_FORMAT_R8G8B8X8_UNORM

Neha Bhende (1):

- util: Initialize pipe_shader_state for passthrough and transform shaders

Pierre-Eric Pelloux-Prayer (1):

- omx: fix build with gcc 10

Rhys Perry (4):

- nir: fix lowering to scratch with boolean access
- aco: fix interaction with 3f branch workaround and p_constaddr
- aco: check instruction format before waiting for a previous SMEM store
- aco: preserve more fields when combining additions into SMEM

Rob Clark (1):

- freedreno: clear last_fence after resource tracking

Samuel Pitoiset (4):

- spirv,radv,anv: implement no-op VK_GOOGLE_user_type
- nir/lower_explicit_io: fix NON_UNIFORM access for UBO loads
- radv: enable zero VRAM for Doom Eternal
- radv: enable zero VRAM for all VKD3D (DX12->VK) games

Timothy Arceri (3):

- glsl: stop cascading errors if process_parameters() fails
- radv: fix regression with builtin cache
- glsl: fix potential slow compile times for GLSLOptimizeConservatively

Vinson Lee (4):

- zink: Check fopen result.
- r300g: Remove extra printf format specifiers.
- vdpau: Fix wrong calloc sizeof argument.
- mesa: Fix NetBSD compiler macro.

Yevhenii Kolesnikov (1):

- intel/compiler: fix cmod propagation optimisations

4.13 Mesa 20.1.1 Release Notes / 2020-06-10

Mesa 20.1.1 is a bug fix release which fixes bugs found since the 20.1.0 release.

Mesa 20.1.1 implements the OpenGL 4.6 API, but the version reported by glGetString(GL_VERSION) or glGetIntegerv(GL_MAJOR_VERSION) / glGetIntegerv(GL_MINOR_VERSION) depends on the particular driver being used. Some drivers don't support all the features required in OpenGL 4.6. OpenGL 4.6 is **only** available if requested at context creation. Compatibility contexts may report a lower version depending on each driver.

Mesa 20.1.1 implements the Vulkan 1.2 API, but the version reported by the apiVersion property of the VkPhysicalDeviceProperties struct depends on the particular driver being used.

4.13.1 SHA256 checksum

```
3ea6e46ea7881c656f7b4724639eaa4672d4e0e0b70869651e8f955ebae3d476  mesa-20.1.1.tar.xz
```

4.13.2 New features

- None

4.13.3 Bug fixes

- i965: Rendering problems replaying a trace of "Refunct" after mesa-20.1.0-rc1 release [bisected]
- gallium/winsys/radeon/drm fails assertion on 32bit
- NIR validation failed after glsl to nir, before function inline, wrong {src,dst}->type ?
- Mesa 20.0.7 / 20.1.0-rc4 regression, extremally long shader compilation time in NIR
- Mesa-git build fails on Fedora Rawhide
- Doom Eternal 1.1 performs very poorly on RADV
- iris/i965: possible regression in 20.0.5 due to changes in buffer manager sharing across screens (firefox/mozilla#1634213)
- iris/i965: possible regression in 20.0.5 due to changes in buffer manager sharing across screens (firefox/mozilla#1634213)
- Incorrect _NetBSD__ macro inside execmem.c
- Possible invalid sizeof in device.c
- GLSL compiler assertion is_float() failed in glsl/ir_validate.cpp, visit_leave on specific WebGL shader
- [RADV] - Doom Eternal (782330) & Metro Exodus (412020) - Title requires 'RADV_DEBUG=zerovram' to eliminate colorful graphical aberrations.
- [RADV] - Doom Eternal (782330) & Metro Exodus (412020) - Title requires 'RADV_DEBUG=zerovram' to eliminate colorful graphical aberrations.
- mesa trunk master vulkan overlay-layer meson.build warning empty configuration_data() object

4.13.4 Changes

Alyssa Rosenzweig (1):

- pan/bi: Fix emit_if successor assignment

Andrii Simiklit (1):

- glsl: fix crash on glsl macro redefinition

Charmaine Lee (1):

- llvmpipe: do not enable tessellation shader without llvm coroutines support

Clément Guérin (1):

- radv: Always expose non-visible local memory type on dedicated GPUs

Danylo Piliaiev (3):

- glsl: Don't replace lrp pattern with lrp if arguments are not floats
- glsl: inline functions with unsupported return type before converting to nir
- i965: Work around incorrect usage of glDrawRangeElements in UE4

Dave Airlie (1):

- llvmpipe: move coroutines out of noopt case

Dylan Baker (1):

- vulkan-overlay/meson: use install_data instead of configure_file

Eric Engestrom (5):

- docs/relnotes add sha256 sums to 20.1.0
- docs: drop new_features.txt
- .pick_status.json: Update to 3a1a40b4431d505fa6487cd012ddb4b64387aee5
- glapi: remove deprecated .getchildren() that has been replace with an iterator
- intel: fix gen_sort_tags.py

Erik Faye-Lund (2):

- zink: Use store_dest_raw instead of storing an uint
- nir: reuse existing psiz-variable

Gert Wollny (1):

- nir: lower_tex: Don't normalize coordinates for TXF with RECT

Ilia Mirkin (1):

- nouveau: allow invalidating coherent/persistent buffer backings

Jason Ekstrand (2):

- intel/vec4: Stomp the return type of RESINFO to UINT32
- intel/fs: Fix unused texture coordinate zeroing on Gen4-5

Jonathan Marek (1):

- freedreno/a6xx: use nonbinning VS when GS is used

Lionel Landwerlin (4):

- iris: fix BO destruction in error path
- i965: don't forget to set screen on duped image
- i965: fix export of GEM handles
- iris: fix export of GEM handles

Marek Olšák (1):

- radeonsi: add a hack to disable TRUNC_COORD for shadow samplers

Neha Bhende (1):

- util: Initialize pipe_shader_state for passthrough and transform shaders

Peter Seiderer (3):

- vc4_bufmgr: fix time_t printf

- pan_bo.h: add time.h include for time_t
- v3d_bufmgr: fix time_t printf

Pierre-Eric Pelloux-Prayer (3):

- winsys/radeon: do not cast bo->va as void*
- ac/surface: set SCANOUT if surf->is_displayable
- ac/surface: fix epitch when modifying surf_pitch

Rhys Perry (4):

- aco: fix interaction with 3f branch workaround and p_constaddr
- aco: consider SDWA during value numbering
- aco: check instruction format before waiting for a previous SMEM store
- aco: preserve more fields when combining additions into SMEM

Rob Clark (1):

- freedreno/computerator: fix missing dependency on generated header

Samuel Pitoiset (5):

- spirv,radv,anv: implement no-op VK_GOOGLE_user_type
- aco: fix register allocation for subdword instructions on GFX10
- radv: enable zero VRAM for Doom Eternal
- radv: enable zero VRAM for all VKD3D (DX12->VK) games
- nir/lower_explicit_io: fix NON_UNIFORM access for UBO loads

Satyeshwar Singh (1):

- intel/dev: Don't consider all TGL SKUs as GT1 only

Timothy Arceri (2):

- radv: fix regression with builtin cache
- glsl: fix potential slow compile times for GLSLOptimizeConservatively

Vinson Lee (8):

- pan/bi: Initialize struct fma_op_info member extended.
- zink: Check fopen result.
- etnaviv: Fix memory leak on error path.
- r300g: Remove extra printf format specifiers.
- vdpau: Fix wrong calloc sizeof argument.
- mesa: Fix NetBSD compiler macro.
- intel/genxml: Migrate from deprecated xml.etree.ElementTree getchildren.
- Switch from cElementTree to ElementTree.

4.14 Mesa 20.1.0 Release Notes / 2020-05-27

Mesa 20.1.0 is a new development release. People who are concerned with stability and reliability should stick with a previous release or wait for Mesa 20.1.1.

Mesa 20.1.0 implements the OpenGL 4.6 API, but the version reported by glGetString(GL_VERSION) or glGetIntegerv(GL_MAJOR_VERSION) / glGetIntegerv(GL_MINOR_VERSION) depends on the particular driver being used. Some drivers don't support all the features required in OpenGL 4.6. OpenGL 4.6 is **only** available if requested at context creation. Compatibility contexts may report a lower version depending on each driver.

Mesa 20.1.0 implements the Vulkan 1.2 API, but the version reported by the apiVersion property of the VkPhysicalDeviceProperties struct depends on the particular driver being used.

4.14.1 SHA256 checksum

```
2109055d7660514fc4c1bcd861bcba9db00c026119ae222720111732dba27c83  mesa-20.1.0.tar.xz
```

4.14.2 New features

- GL_ARB_compute_variable_group_size on i965.
- GL_EXT_depth_bounds_test on Iris.
- GL_EXT_texture_shadow_lod on radeonsi, nvc0.
- GL_NV_alpha_to_coverage_dither_control on radeonsi
- GL_NV_copy_image on all gallium drivers.
- GL_NV_pixel_buffer_object on all gallium drivers, i915, i965, swrast.
- GL_NV_viewport_array2 on nvc0 (GM200+).
- GL_NV_viewport_swizzle on nvc0 (GM200+).
- VK_AMD_memory_overallocation_behavior on RADV.
- VK_KHR_shader_non_semantic_info on Intel, RADV.
- GL_EXT_draw_instanced on gles2
- VK_KHR_8bit_storage for ACO on GFX8+
- VK_KHR_16bit_storage for ACO on GFX8+ (storageInputOutput16 is still unsupported)
- shaderInt16 for ACO on GFX9+
- VK_KHR_shader_float16_int8 for ACO on GFX8+ (shaderFloat16 is still unsupported)
- VK_EXT_robustness2 on Intel, RADV.
- Add Rocket Lake (RKL) support on anvil and iris.

4.14.3 Bug fixes

- Reproduceable i915 gpu hang Intel Iris Plus Graphics (Ice Lake 8x8 GT2)
- glsl: regression affecting shader compilation time
- freedreno: glamor issue with x11 desktops

- [gles3] supertuxkart: some textures are incorrect
- Double lock in fbobject.c
- [bisected] Steam crashes when newest Iris built with LTO
- i965/vec4: opt_cse_local cause the out of bound array access
- NIR: Regression on shader using 8/16-bit integers
- lp_bld_intr.c:70:16: error: use of undeclared identifier 'LLVMFixedVectorTypeKind'; did you mean 'LLVMVectorTypeKind'?
- Deadlock in anv_timelines_wait()
- post_version.py does not work with release candidates
- post_version.py does not work with release candidates
- radv regression on android
- srcutilmeson.build:294:4: ERROR: Program or command 'winepath' not found or not executable
- debug builds are massively broken on Windows
- heavy glitches on amd ryzen 5 since version 20.x
- zink asserts with 32-bit boolean
- Dirt: Showdown bad performance and broken rendering with enabled advanced lightning
- gravit & Firefox WebGL broken since 3dc2ccc14c0e035368fea6ae3cce8c481f3c4ad2 "ac/surface: replace RADEON_SURF_OPTIMIZE_FOR_SPACE with !FORCE_SWIZZLE_MODE"
- mesa 20.0.5 causing kitty to crash
- radeonsi: "Torchlight II" trace showing regression on mesa-20.0.6 [bisected]
- [RADV/LLVM/ACO/Regression] After mesa commit a3dc7fffbb7be0f1b2ac478b16d3acc5662dff66 all games stucks at start
- Android building error after commit 2ab45f41
- iris: Crash when trying to capture window in OBS Studio
- Properly annotate control flow convergence points
- intel/compiler: Register coalesce doesn't move conditional modifiers
- [bisected] [iris] mpv under wayland: failed to import supplied dmabufs: Unsupported buffer format 808669784
- [Bisected][Iris] piglit.spec.!opengl 1_1.max-texture-size crashes on x32 platform
- anv : android deqp assert dEQP-VK.api.external.memory.android_hardware_buffer.dedicated.image#export_import_bind_bind
- GL cts gtf30.GL3Tests.sgis_texture_lod.sgis_texture_lod_basic_getter failure
- freedreno/a6xx: texture cache vs realloc_bo()
- [Bisected] dEQP-VK.subgroups.ballot_mask.ext_shader_subgroup_ballot.* failures
- dEQP-VK.subgroups.size_control.compute.* crashes on HSW and TGL
- zink: framebuffer and pipeline caches accumulate due to zink_create_surface()
- FTBFS due to LLVM commit 2dea3f129878 (LLVMVectorTypeKind is gone)
- [r600/Turks] 20.0.2: modesetting/radeon driver SIGABRT at loading X (kernel 5.5.10, ppc64)
- piglit spec.!opengl 1.0.gl-1.0-fpexceptions crash on Iris

- ci: Update the Wine version
- SPIR-V: Failure in dEQP-VK.graphicsfuzz.control-flow-switch
- SPIR-V: OpConvertUToPtr from spec constant fails to compile
- ACO: Regression: Texture corruption
- radv: Reading ViewportIndex in fragment shader returns garbage
- piglit spec.arb_gpu_shader_fp64.execution.arb_gpu_shader_fp64-vs-non-uniform-control-flow-ssbo crash on Iris
- piglit spec/arb_gpu_shader_fp64/execution/built-in-functions/vs-sign-neg-abs.shader_test failure on IVB
- [ANV] gfxbench Aztec Ruins misrenders on gen11+
- glxinfo cmd crashed
- radeonsi: GL_LINES rendering is affected by GL_POINT_SPRITE
- nir: nir_lower_returns can't handle nested loops
- Graphic artifacts with Mesa 20.0.4 on intel HD 510 GPU
- [Iris] [Bisected] Some KHR-GL46.arrays_of_arrays_gl. tests are failing
- Mesa 20 regression makes Lightsprint demos crash
- metro redux games crash upon loading certain levels on amdgpu
- dri_common.h:58:8: error: unknown type name '__GLXDRIdrawable'
- Graphical glitches on Intel Graphics when Xorg started on Iris driver
- GL/GLES test crashes on G33/i915 platforms
- GL/GLES test crashes on G33/i915 platforms
- GL/GLES test crashes on G33/i915 platforms
- SIGSEGV src/compiler/glsl/ast_function.cpp:53
- manywin aborts with "i965: Failed to submit batchbuffer: Invalid argument"
- manywin aborts with "i965: Failed to submit batchbuffer: Invalid argument"
- manywin aborts with "i965: Failed to submit batchbuffer: Invalid argument"
- manywin aborts with "i965: Failed to submit batchbuffer: Invalid argument"
- v3d: transform feedback issue
- radv: Enable TC-compat HTILE in VK_IMAGE_LAYOUT_GENERAL.
- radv: dEQP-VK.binding_model.descriptorset_random.sets4.noarray.ubolimitlow.sbolimitlow.imglimitlow.noiub.comp.noia.0 segfault
- radv: RAVEN fails dEQP-VK.pipeline.timestamp.misc_tests.reset_query_before_copy
- buffer overflow in nouveau driver on mesa 20.0.2
- xmlconfig sha1 code has overflow and possible bug
- enable storageBuffer16BitAccess feature in radv for SI and CIK
- Build Fails with Clang Shared Library
- Thousands of 32 bit regressions in VulkanCTS and GL test suites due to handling of cross-invocation
- anv: isl assert when running dEQP-VK.geometry.layered.3d.*.readback

- Weston drm-backend.so seems to fail with Mesa master and LIBGL_ALWAYS_SOFTWARE=1
- freedreno/turnip: Don't request pixlodenable when we don't use it
- VulkanCTS uniform_buffer_block_geom spins forever
- freedreno: dEQP-GLES3.functional.fbo.msaa.4_samples.r16f flakiness in CI
- srcutilmeson.build:291:4: ERROR: Program or command 'winepath' not found or not executable
- RADV: flickering textures in Q.U.B.E. 2 through Proton
- Missing ENDBR in entry_x86-64_tls.h, entry_x86_tls.h and entry_x86_tsd.h
- [regression][bisected] Android build test fails: marshal_generated.c', missing and no known rule to make it
- Missing ENDBR in rtasm_x86sse.c
- src/intel/tools/aubinator_viewer.cpp:383:52: error: format '%lx' expects argument of type 'long unsigned int', but argument 5 has type 'uint64_t {aka long long unsigned int}' [-Werror=format=]
- src/compiler/glsl/ast_to_hir.cpp:2134: ir_rvalue* ast_expression::do_hir(exec_list*, _mesa_glsl_parse_state*, bool): Assertion 'result != NULL || !needs_rvalue' failed.
- process_test fails on macOS
- Vulkan Overlay is blinking
- Regression: 9d64ad2fe79 broke Rocket League
- GameMaker games (Memoranda and Undertale) + amdgpu — Segmentation fault on launch
- Civilization VI - Animated leader characters small black squares artifacts
- [ACO] Reliable crash with RPCS3 that is not present with LLVM
- [RADV] vkCmdBindTransformFeedbackBuffersEXT pSizes optional parameter not handled
- [RadeonSI] - Curse of the Dead Gods (1123770) - Lighting is not rendering correctly.
- soft-fp64: __fsat64 incorrectly returns NaN for a NaN input. It should return zero.
- Hang when using glWaitSync with multithreaded shared GL contexts
- RPCS3 / Persona 5 - Performance regression [RADV / Navi]
- [ANV] Rendering corruption in Shadow of the Tomb Raider
- src/compiler/glsl/glcpp/glcpp-parse.y:1297: _token_print: Assertion '!"Error: Don't know how to print token."' failed.
- [CTS] dEQP-VK.descriptor_indexing.* fails on RADV/LLVM
- Unigine Valley failure / assert
- [Gen9/icl] [Bisected] [Regression] dEQP-GLES3.functional.shaders.loops.short_circuit.do_while_fragment fail
- [RadeonSI][gfx10/navi] Kerbal Space Program crash: si_draw_vbo: Assertion '0' failed
- Budget Cuts hits VK_AMD_shader_fragment_mask assert
- Follow-up from "i965/blorp: Don't resolve HiZ unless we're reinterpreting"
- crash in vc4_write_uniforms with shaders involving YUV textures
- Corrupted output with vaapi 10 bit -> 8 bit transcoding on AMD RAVEN
- tessellator.cpp:78:7: error: 'fmin' is missing exception specification 'noexcept'

- Please add Raspberry Pi 4 to features.txt
- Build failure with bison 2.3.
- Mesa build fails on 32 bit architecture
- Mesa build fails on 32 bit architecture
- Incorrect rendering with vaapi + uyvy422
- V3D/Broadcom (Raspberry Pi 4) - GLES 3.1 - GL_EXT_texture_norm16 advertised, but not usable
- mesa-20.0.0/src/amd/compiler/aco_instruction_selection.cpp:7221:55: style: Same expression on both sides of '&&
- i965 assertion failure in fallback_rgbx_to_rgba
- vaapi bob deinterlacer produces wrong output height on AMD
- Compute copies do not handle SUBSAMPLED formats
- Please document RADV_TEX_ANISO variable in envvars.html
- unexpected CI failure
- Multiple glapi_mapi_tmp.h
- drisw crashes on calling NULL putImage on EGL surfaceless platform (pbuffer EGLSurface)
- VRAM leak with vuilkan external memory + opengl memory objects
- [radeonsi][vaapi][bisected] invalid VASurfaceID when playing interlaced DVB stream in Kodi
- [RADV] GPU hangs while the cutscene plays in the game Assassin's Creed Origins
- ACO: The Elder Scrolls Online crashes on startup (Navi)
- Broken rendering of glxgears on S/390 architecture (64bit, BigEndian)
- aco: sun flickering with Assassins Creeds Origins
- !1896 broke ext_image_dma_buf_import piglit tests with radeonsi
- aco: wrong geometry with Assassins Creed Origins on GFX6
- valgrind errors since commit a8ec4082a41
- src/broadcom/qpu/qpu_pack.c:962:25: error: implicit declaration of function 'ffs' is invalid in C99 [-Werror,-Wimplicit-function-declaration] mux_b = ffs(desc->mux_b_mask) - 1;
- X fails to start with amdgpu and Mesa 20.1 on Fedora
- GPU hangs in Factorio on Radeon RX 5700 XT (MSI GAMING X)
- OSMesa osmesa_choose_format returns a format not supported by st_new_renderbuffer_fb
- Build error with VS on WIN
- Using EGL_KHR_surfaceless_context causes spurious "libEGL warning: FIXME: egl/x11 doesn't support front buffer rendering."
- !3460 broke texsubimage test with piglit on zink+anv
- VERSION needs to be bumped for trunk master
- The screen is black when using ACO

4.14.4 Changes

Abhishek Kumar (1):

- anv/android: fix assert in anv_import_ahw_memory

Adam Jackson (1):

- gallium: enable EGL_EXT_image_dma_buf_import_modifiers unconditionally

Albert Astals Cid (5):

- cube_face_coord: Use fabsf instead of fabs since we know it's floats
- cube_face_index: Use fabsf instead of fabs since we know it's floats
- aco: Minor optimization in spill_ctx constructor
- aco: pass vars by const &
- Fix promotion of floats to doubles

Alejandro Piñeiro (7):

- docs/features: add v3d driver
- nir/linker: remove reference to just SPIR-V linking
- v3d/tex: don't configure tmu config 1 if not needed
- v3d/tex: Configuration Parameter 1 can be only skipped if P2 can be skipped too
- v3d/packet: fixing TMU_Config_Parameter_2 definition
- nir: add nir_tex_instr_need_sampler helper
- v3d: support for textureQueryLOD

Alexandros Frantzis (3):

- gitlab-ci: Automated testing with OpenGL traces
- gitlab-ci: Fix traces caching in tracie
- gitlab-ci: Check the Mesa version used for tracie tests

Alyssa Rosenzweig (505):

- pan/midgard: Break out one-src read_components
- pan/midgard: Implement mixed-type constant packing
- panfrost: Avoid overlapping copy
- pan/midgard: Check for null consts
- pan/midgard: Remove unused variable
- panfrost: Use size0 when calculating the offset to a depth level
- pan/midgard: Fix scheduling issue with csel + render target reference
- panfrost: Simplify swizzle translation
- panfrost: Update comment about magic number relating to barriers
- panfrost: Ensure compute shader_meta is zeroed
- panfrost: Identify mali_shared_memory structure
- panfrost: Unify bifrost_scratchpad with mali_shared_memory

- panfrost: Rename bifrost_framebuffer->mali_framebuffer
- panfrost: Rename unknown2_8 to padding
- panfrost: Allocate RAM backing of shared memory
- pan/midgard: Track pressure when scheduling ld/st
- pan/midgard: Fix missing prefixes
- pan/midgard: Fix swizzles harder
- pan/midgard: Implement barriers
- pan/midgard: Allow jumping out of a shader
- pan/midgard: Fix 32/64 mixed swizzle packing
- pan/midgard: Use dummy tag for empty shaders
- pan/midgard: Improve barrier disassembly
- pan/midgard: Overhaul tag handling
- pan/midgard: Imply next tags
- pan/midgard: Infer tags entirely
- pan/midgard: Set xyzx swizzle for load_compute_arg
- pan/midgard: Identify stack barrier flag
- pan/midgard: Don't crash with constants on unknown ops
- pan/midgard: Use fprintf instead of printf for constants
- pan/decode: Remove extraneous newline
- pan/decode: Add 'minimal' mode
- pan/decode: Cleanup pandecode_jc
- panfrost: Implement PAN_DBG_SYNC with pandecode/minimal
- panfrost: Print synced traces to stderr
- panfrost: Rewrite scoreboarding routines
- panfrost: Update scoreboarding notes
- panfrost: Cleanup transfer_map
- panfrost: Avoid reading GPU memory when packing vertices
- panfrost: Debitfieldize mali_uniform_buffer_meta
- panfrost: Remove enum panfrost_memory_layout
- panfrost: Remove dirty tracking
- panfrost: Remove old comment
- panfrost: Remove old hack
- panfrost: Remove flush_frontbuffer
- pan/midgard: Identify clamp(x, -1.0, 1.0) flag
- panfrost: Move checksum routines to root panfrost
- panfrost: Move pan_afbc.c to root

- panfrost: Move format translation to root
- panfrost: Rewrite texture descriptor creation logic
- nir: Add SSBO->global lowering pass
- pan/midgard: Lower SSBOs in NIR
- pan/midgard: Implement nir_intrinsic_get_buffer_size
- pan/midgard: Implement load/store_shared
- panfrost: Combine get_index_buffer with bound computation
- panfrost: Implement index buffer cache
- pan/decode: Dump scratchpad size if present
- pan/midgard: Don't spill near a branch
- panfrost: Fix gl_VertexID/InstanceID
- panfrost: Fix padded_vertex_count generation
- panfrost: Update spilling comment framebuffer->shared
- panfrost: Don't set shared->unk0
- panfrost: Fix param getting
- panfrost: Default to 256 threads for TLS
- panfrost: Reserve an extra page for spilling
- panfrost: Simplify stack shift calculation
- panfrost: Expose PIPE_CAP_PRIMITIVE_RESTART
- panfrost: Add PAN_MESA_DEBUG=gles3 option
- panfrost: Increase SSBO/image limit from 4->8
- pan/midgard: Allow inverted inverted ops
- pan/midgard: Allow fusing inverted sources for inverted ops
- pan/midgard: Partially fix 64-bit swizzle alignment
- pan/midgard: Extract nir_ssa_index helper
- pan/midgard: Add LDST_ADDRESS property
- pan/midgard: Fix load/store argument sizing
- pan/midgard: Round up bytemasks when promoting uniforms
- pan/midgard: Force address alignment
- pan/midgard: Add address analysis framework
- pan/midgard: Use address analysis for globals, etc
- pan/decode: Calm an assert to a pandecode error
- pan/decode: Restore bifrost sample_locations
- pan/decode: Fix tiler weights printing
- pan/decode: Skip analysis for Bifrost tiler structures
- pan/bi: Add discard ops

- pan/bi: Add ICMP.GL.NEQ op
- pan/bi: Move notes on FMA opcodes from disassembler
- pan/bi: Introduce CSEL4 class
- pan/bi: Move notes on ADD ops to notes file
- pan/bi: Decode FMA_SHIFT properly
- pan/bi: Add v4i8 mode to FMA_SHIFT
- pan/bi: Identify extended FMA opcodes
- pan/bi: Decode ADD_SHIFT properly
- pan/bi: Combine LOAD_VARYING_ADDRESS instructions by type
- pan/bi: Squash LD_ATTR ops together
- pan/bi: Structify FMA_FADD
- pan/bi: Move some definitions from disasm to bifrost.h
- panfrost: Add note about preloaded varyings
- pan/bi: Gut old compiler
- pan/bi: Stub out new compiler
- pan/bi: Add the control flow graph
- pan/bi: Add src/dest fields to bifrost_instruction
- pan/bi: Add class properties
- pan/bi: Add modifiers to bi_instruction
- pan/bi: Add BI_GENERIC property
- pan/bi: Factor out enum bifrost_minmax_mode
- pan/bi: Add a bifrost_roundmode field
- pan/bi: Add bifrost_minmax_mode field
- pan/bi: Add bi_load structure
- pan/bi: Pull out bifrost_load_var
- pan/bi: Add bi_load_vary structure
- pan/bi: Add PAN_SCHED_* flags
- pan/bi: Add bi_clause, bi_bundle abstractions
- pan/bi: Add dest_type field to bifrost_instruction
- pan/bi: Add special indices
- pan/bi: Add constant field to bi_instruction
- pan/bi: Add class-specific ops
- pan/bi: Add clause header fields to bi_clause
- pan/bi: Clarify special op scheduling
- pan/bi: Add swizzles
- pan/bi: Add source type for conversions

- pan/bi: Add EXTRACT, MAKE_VEC synthetic ops
- pan/bi: Add constants to bi_clause
- pan/bi: Add pred/successors to build CFG
- pan/bi: Extract bifrost_branch structure
- pan/bi: Add bi_branch data
- pan/bi: Add CSEL condition
- pan/bi: Add high-latency property for classes
- pan/bi: Add quirks system
- pan/bi: Add IR iteration macros
- pan/bi: Move some print routines out of the disasm
- pan/bi: Add BIR manipulation routines to bir.c
- pan/bi: Move bi_interp_mode_name to bi_print
- pan/bi: Add bi_instruction printing
- pan/bi: Add bi_print_bundle for printing bi_bundle
- pan/bi: Add bi_print_clause
- pan/bi: Add bi_print_block
- pan/bi: Add bi_print_shader
- pan/bi: Lower and optimize NIR
- pan/bi: Walk through the NIR control flow graph
- pan/bi: Improve block printing
- pan/bi: Don't print types for unconditional branches
- pan/bi: Print branch target
- pan/bi: Add instruction emit/remove helpers
- pan/bi: Call nir_lower_io_to_temporaries in cmdline
- pan/bi: Add support for if-else blocks
- pan/bi: Handle loops when ingesting CFG
- pan/bi: Handle jumps (breaks, continues)
- pan/bi: Fix destination printing
- pan/bi: Implement nir_intrsinic_load_interpolated_input
- pan/bi: Add blend_location to IR for BI_BLEND
- pan/bi: Add bi_schedule_barrier helper
- pan/bi: Implement store_output for fragment shaders
- pan/bi: Implement load_input for vertex shaders
- pan/bi: Add helpers for creating temporaries
- pan/bi: Implement store_vary for vertex shaders
- pan/bi: Add preliminary LOAD_UNIFORM implementation

- pan/bi: Implement load_const
- pan/bi: Add dummy scheduler
- pan/bi: Rename next-wait to simply 'wait'
- pan/bi: Fix Android.mk
- panfrost: Move mir_to_bytemask to common code
- pan/bi: Generalize swizzles to avoid extracts
- pan/bi: Introduce writemasks
- pan/bi: Remove bi_load
- pan/bi: Lower vec* to writemasks in NIR
- pan/bi: Add initial handling of ALU ops
- pan/bi: Allow inlining constants
- pan/bi: Implement fsat as mov.sat
- pan/bi: Add a bunch of ALU ops
- pan/bi: Add BI_SPECIAL_* enum
- pan/bi: Handle special ops in NIR->BIR
- pan/bi: Implement fabs, fneg as fmov with mods
- pan/bi: Disable lower_sub
- pan/bi: Add isub op
- pan/bi: Import algebraic pass from midgard
- pan/bi: Implement nir_op_bcsel
- pan/bi: Lower b2f to bcsel
- pan/bi: Specify comparison op for BI_CMP
- pan/bi: Print source types unconditionally
- pan/bi: Implement comparison opcodes via BI_CMP
- panfrost: Promote midgard_program to panfrost/util
- pan/midgard: Remove unused iterators
- pan/midgard: Adjust sysval-related prototypes
- pan/midgard: Remove indexing dependency of sysvals
- pan/midgard: Decontextualize midgard_nir_assign_sysval_body
- pan/midgard: Remove dest_override sysval argument
- panfrost: Move Midgard sysval code to common Panfrost
- pan/bi: Switch to panfrost_program
- pan/bi: Implement sysvals
- pan/midgard: Localize 'visited' tracking
- pan/midgard: Decontextualize liveness analysis core
- pan/midgard: Sync midgard_block field names with Bifrost

- pan/midgard: Subclass midgard_block from pan_block
- panfrost: Move liveness analysis to root panfrost/
- panfrost: Sync Midgard/Bifrost control flow
- pan/bi: Paste over bi_has_arg
- pan/bi: Add bi_bytemask_of_read_components helpers
- pan/bi: Add bi_next/prev_op helpers
- pan/bi: Add bi_max_temp helper
- pan/bi: Add liveness analysis pass
- pan/bi: Add dead code elimination pass
- pan/bi: Implement nir_op_ffma
- pan/bi: Fix swizzle for second argument to ST_VARY
- panfrost: Move lcra to panfrost/util
- pan/midgard: Remove incorrect comment in RA
- pan/bi: Minor fixes in iteration macros
- pan/bi: Fix vector handling of readmasks
- pan/bi: Fix missing src_types
- pan/bi: Add register allocator
- pan/bi: Interpret register allocation results
- pan/bi: Setup initial clause packing
- pan/bi: Sketch out instruction word packing
- pan/bi: Add packing for register control field
- pan/bi: Pack register fields
- pan/bi: Add missing __attribute__((packed))
- pan/bi: Assign registers to ports
- pan/bi: Route through first_instruction field
- pan/bi: Model 3-bit Bifrost srcs in IR
- pan/bi: Add struct bifrost_fma_fma
- pan/bi: Pack BI_FMA ops
- pan/bi: Pack fadd32
- pan/bi: List ADD classes in bi_pack_add
- pan/bi: Generalize bi_get_src a bit
- pan/bi: Pass second src for load_vary ops
- pan/bi: Emit load_vary ops
- pan/bi: Skip over data registers in port assignment
- pan/bi: Route through clause header
- pan/bi: Pretty-print clause types in disassembler

- pan/bi: Don't hide SCHED_ADD inside HI_LATENCY
- pan/bi: Track clause types during scheduling
- pan/bi: Flesh out ATEST in IR
- pan/bi: Add ATEST packing
- pan/bi: Flesh out BI_BLEND
- pan/bi: Pack BI_BLEND
- pan/bi: Implement FMA/MOV without modifiers
- pan/bi: Add bi_emit_before helper
- pan/bi: Add move lowering pass
- pan/bi: Pack a constant quadword
- pan/bi: Document constant related errata(?)
- pan/bi: Index out constants in instructions
- pan/bi: Include UBO index for sysval reads
- pan/bi: Add bi_load32_components helper
- pan/bi: Pack ld_ubo ops
- pan/bi: Pack ld_var_addr
- pan/bi: Flesh out st_vary IR
- pan/bi: Generalize data register setting
- pan/bi: Add store_channels property
- pan/bi: Pack st_vary
- pan/bi: Pack LD_ATTR
- pan/bi: Lower bool to ints
- pan/bi: Remove hacks for 1-bit booleans in IR
- pan/bi: Add 'soft' NIR->BIR condition translation
- pan/bi: Implement csel fusing
- pan/bi: Respect shift when printing immediates
- pan/bi: Use bi_lookup_immediate when packing
- pan/bi: Default csel to "!= 0" mode
- pan/bi: Pack csel4 opcodes
- pan/bi: Ingest vecN directly (again)
- pan/bi: Lower combines to rewrites for scalars
- pan/bi: Rewrite aligned vectors as well
- panfrost: Split panfrost_device from panfrost_screen
- panfrost: Isolate panfrost_bo_access_for_stage to pan_cmdstream.c
- panfrost: Inline reference counting routines
- panfrost: Move pan_bo to root panfrost

- pan/bit: Link standalone compiler with en/decoder
- panfrost: Move device open/close to root panfrost
- pan/bit: Open up the device
- panfrost: Stub out G31/G52 quirks
- pan/bit: Submit a WRITE_VALUE job as a sanity check
- pan/bit: Begin generating a vertex job
- pan/bi: Fix overzealous write barriers
- pan/bi: Fix off-by-one in scoreboarding packing
- pan/bi: Enable precision lowering in standalone compiler
- panfrost: Enable PIPE_SHADER_CAP_FP16 on Bifrost
- pan/bi: Handle f2f* opcodes
- pan/bi: Ignore swizzle in unwritten component
- pan/bi: Finish FMA structures
- pan/bi: Fix missing type for fmul
- pan/bi: Add FMA16 packing
- pan/bi: Pack outmod and roundmode with FMA
- pan/bi: Expand out FMA conversion opcodes
- pan/bi: Enumerate conversions
- pan/bi: Handle standard FMA conversions
- pan/bi: Add bifrost_fma_2src generic
- pan/bi: Add one-source f32->f16 op
- pan/bi: Assert out i16 related converts for now
- pan/bi: Handle round opcodes in frontend
- pan/bi: Add v2f16 versions of rounding ops
- pan/bi: Structify fadd/min/max16
- pan/bi: Handle core faddminmax16 packing
- pan/bi: Handle abs packing for fp16/FMA add/min
- pan/bi: Handle fp16/abs scheduling restriction
- pan/bi: Fix handling of constants with COMBINE
- pan/bit: Add ‘run‘ mode to the cmdline
- pan/bit: Wire through I/O
- pan/bi: Fix writes_component for VECTOR
- pan/bi: Use STAGE srcs for scheduler nops
- pan/bi: Don’t set the back-to-back bit yet
- pan/bi: Add cmdline option for verbose disassembly
- pan/bi: Fix unused port swapping

- pan/bi: Handle fmov class ops
- pan/bi: Fix outmod/roundmode flip
- pan/bi: Export bi_class_name
- pan/bi: Fix duplicated source in ADD.v2f16
- pan/bi: Fix negation in ADD.v2f16
- pan/bi: Don't gobble zero ports
- pan/bi: Allow BI_FMA to take mods
- pan/bi: Handle BIFROST_FIRST_WRITE_FMA_P2_READ_P3
- pan/bi: Add helper to debug port assignment
- pan/bi: Match CSEL argument order with hw
- pan/bit: Stub out BIR interpreter
- pan/bit: Handle read/write
- pan/bit: Add preliminary FMA/ADD/MOV implementations
- pan/bit: Implement outmods
- pan/bit: Implement floating source mods
- pan/bit: Add packing test framework
- pan/bit: Add helper for generating floating mod tests
- pan/bit: Add verbose printing for tests
- pan/bit: Add 16-bit fmod tests
- pan/bit: Add FMA tests
- pan/bit: Add CSEL to interpreter
- pan/bit: Add csel tests
- pan/bit: Make run more useful
- pan/bit: Add mode to run unit tests
- pan/bi: Remove nontrivial SPECIAL ops
- pan/bi: Add 32-bit _FAST packing
- pan/bi: Add fp16 support for frcp/frsq
- pan/bit: Add special op interpreting
- pan/bit: Add special unit test
- pan/bi: Implement min/max on FMA
- pan/bi: Structify ADD unit add/min/max
- pan/bi: Add ADD add/min/max fp32 packing
- pan/bi: Set BI_MODS for MINMAX
- pan/bi: Fix incorrect abs flip in fma/fadd16
- pan/bi: Force ADD scheduling for MINMAX
- pan/bit: Unify test frontends

- pan/bit: Add min/max support to interpreter
- pan/bit: Enable more debug for ‘run‘
- pan/bit: Add fmin/max16 tests
- pan/bit: Wire up add/add op+test
- panfrost: Add IS_BIFROST quirk
- panfrost: Populate bifrost-specific structs within mali_shader_meta
- panfrost: Staticize a few cmdstream functions
- panfrost: Unify vertex/tiler structures
- panfrost: Set mfbd.msaa.sample_locations on Bifrost
- panfrost: Call the Bifrost compiler on bi devices
- pan/bi: Fix nondeterministic register packing
- pan/midgard: Remove unused max_varying variable
- panfrost: Move varying linking to cmdstream
- panfrost: Move uniform_count to pan_assemble
- panfrost: Pass compiler-appropriate options
- pan/bi: Fix backwards registers ports
- panfrost: Fix BI_BLEND packing
- pan/bi: Let !b2b imply branch_cond
- pan/decode: Print Bifrost blend descriptor
- panfrost: Drop dependency on nonexistant write_value
- pan/bi: Lower fsqrt
- pan/midgard: Fix f2u naming confusion
- pan/bi: Set BI_ROUNDMODE for BI_CONVERT
- pan/bi: Fix incorrect swizzle packing assert
- pan/bi: Rewrite conversion packing
- pan/bi: ADD packing for CONVERT
- pan/bit: Add BI_CONVERT interpretation
- pan/bit: Add BI_CONVERT tests
- pan/bi: Add disasm for ADD.i8
- pan/bi: Disable FMA scheduling for CONVERT
- pan/bi: Add BI_TABLE for fast table accesses
- pan/bi: Add special op for exp2
- pan/bi: Add op for ADD_FREXPM
- pan/bi: Add FLOG2_U op to disassembler
- pan/bi: Add log_frexpe op to IR
- pan/bi: Add frexp_log packing

- pan/bi: Add bi_pack_fma_2src helper
- pan/bi: Pack ADD_FREXPM
- pan/bi: Add log2_help packing
- pan/bi: Add _MSCALE flag for FMA/ADD
- pan/bi: Structify FMA_MSCALE
- pan/bi: Pack FMA_MSCALE
- pan/bi: Add fexp2_fast packing
- pan/bi: Split src/dest index printing
- pan/bi: Ensure CONSTANT srcs have types
- pan/bi: Fix bi_get_immediate with multiple imms
- pan/bi: Fix packing with multiple constants
- pan/bi: Fix packing with low-nibble-set on hi constant
- pan/bi: Fix lower_combine swizzle rewrite
- pan/bi: Add fexp2 implementation
- pan/bi: Implement flog2
- pan/bi: Fix vec2/3 handling
- pan/bi: Handle st_vary with <4 components
- pan/bi: Try to reuse constants in ALU
- pan/bi: Workaround constant packing errata
- pan/bi: Structify add and min/max fp16 ADD
- pan/bi: Pack ADD.v2f16
- pan/bi: Pack MAX.v2f16
- pan/bi: Dump extra bits for disasm
- pan/bi: Round constants to 32-bit
- pan/bi: Lower special ops to 32-bit
- pan/bit: Add FREXP interp support
- pan/bit: Add frexp_log test
- pan/bit: Add BI_REDUCE_FMA interp
- pan/bit: Add FMA_REDUCE test
- pan/bit: Add log2 helper interp
- pan/bit: Add BI_TABLE test
- pan/bit: _MSCALE interp
- pan/bit: Add FMA_MSCALE test
- pan/bit: Add fexp2_fast interp
- pan/bit: Add fexp2_fast test
- pan/bit: Add constants test

- pan/bit: Add fp16 min/max tests
- pan/bi: Print tex_compact coordinates
- pan/bi: Document when dual-tex is triggered
- pan/bi: Disassemble f16 dual tex
- pan/bi: Structify TEX compact
- pan/bi: Include TEX_COMPACT f16 opcode
- pan/bi: Feed data register to BI_TEX
- pan/bi: Add normal/compact/dual switch to IR
- pan/bi: Stub out tex_compact logic
- pan/bi: Generate TEX_COMPACT instruction
- pan/bi: Pack TEX compact instructions
- pan/bi: Assert out multiple textures
- panfrost: Fix crashes with small BOs
- panfrost: Assert on unimplemented fragcoord etc
- panfrost: Set clear_color_[12] in the extra fb desc
- panfrost: Add tentative bifrost_texture_descriptor
- panfrost: decode textures and samplers on bifrost
- pan/decode: Remove is_zs weirdness
- panfrost: Identify texture layout field
- panfrost: The texture descriptor has a pointer to a trampoline
- pan/bi: Pack fp16 ATEST
- pan/bi: Passthrough type for ATEST
- pan/bi: Passthrough blend types
- pan/bi: Assign blend descriptor for BLEND op
- pan/bi: Add missing BI_VECTOR
- pan/bi: Fix ADD.v4i8 opcode
- pan/bi: Eliminate writemasks in the IR
- pan/bi: Rename BI_SWIZZLE to BI_SELECT
- pan/bi: Pack FMA SEL16
- pan/bi: Pack FMA SEL8
- pan/bi: Pack ADD SEL16
- pan/bi: Force BI_SELECT arguments scalar
- pan/bit: Interpret BI_SELECT
- pan/bit: Add SELECT tests
- pan/bi: Fix RA wrt 16-bit swizzles
- pan/bi: Implement 16-bit COMBINE lowering

- nir: Move nir_lower_mediump_outputs from ir3
- ir3: Use shared mediump output lowering
- pan/bi: Add bool->float opcodes
- pan/bi: Add CSEL.64 opcode
- pan/bi: Add some 8-bit compares
- pan/bi: Add 64-bit int compares
- pan/bi: Add FCMP.GL.v2f16 on ADD opcode
- pan/bi: Add CSEL.8 opcode
- pan/bi(t): Fix SELECT tests
- pan/bi: Deduplicate csel/cmp cond
- pan/bi: Remove bi_round_op
- pan/bi: Structify FMA FCMP
- pan/bi Strucitfy ADD FCMP 32
- pan/bi: Structify FMA FCMP16
- pan/bi: Structify ADD FCMP16
- pan/bi: Structify FMA ICMP 32
- pan/bi: Structify FMA ICMP 16
- pan/bi: Structify ADD ICMP 32
- pan/bi: Fix source mod testing for CMP
- pan/bi: Pack FMA 32 FCMP
- pan/bi: Factor out fp16 abs logic
- pan/bi: Pack fma.fcmp16
- pan/bi: Relax double-abs condition
- pan/bit: Prepare condition evaluation for vectors
- pan/bit: Interpret CMP
- pan/bi: Add initial fcmp test
- pan/bi: Add bitwise modifiers
- pan/bi: Pack BI_BITWISE
- pan/bi: Handle iand/ior/ixor in NIR->BIR
- pan/bit: Interpret BI_BITWISE
- pan/bit: Add BITWISE test
- panfrost: Fix BO reference counting
- panfrost: Move Bifrost IR indexing to common
- pan/bi: Use common IR indices
- pan/mdg: Remove nir_alu_src_index
- pan/mdg: Use PAN_IS_REG

- pan/mdg: SSA_FIXED_MINIMUM already covered by PAN_IS_REG
- pan/mdg: Don't break SSA
- pan/mdg: Remove goofy 16-bit comment
- pan/mdg: Remove old hack
- pan/mdg: Set lower_flrp16
- pan/bi: Share ALU type printing
- pan/mdg: Add type fields to IR
- pan/mdg: Track ALU src types
- pan/mdg: Track ALU dest type
- pan/mdg: Another goofy comment gone
- pan/mdg: Track a primary type for I/O
- pan/mdg: Denoise prints
- pan/mdg: Track v_mov type (force uint32 for now?)
- pan/mdg: Track texture types
- pan/mdg: Set texture full fields at pack time
- pan/mdg: Move sampler_type emission to pack time
- pan/mdg: Lower specials to 32-bit
- pan/mdg: Specialize swizzle to type
- pan/mdg: Always print the mask
- pan/mdg: Make some branch targets more explicit
- pan/mdg: Don't crash on unknown branch target
- pan/mdg: Pass through some types from scheduling
- pan/mdg: Move condense_writemask to disasm
- pan/mdg: Ensure fdot is scalar out in disasm
- pan/mdg: Replicate 16-bit swizzles

Andreas Baierl (8):

- lima/parser: Fix RSW depth test parsing
- lima/parser: Extend AUX0 findings
- lima/parser: Change value name in RSW parser
- lima/parser: Extend rsw parsing showing strings instead of numbers
- gitlab-ci: lima: Add flaky tests to the skips list
- gitlab-ci: Enable the lima job again
- gitlab-ci: Add add a set of lima flakes
- lima: Add etc1 support

Andres Gomez (27):

- tracie: correct typo

- gitlab-ci: add missing popd to the build-deqp-vk.sh script
- gitlab-ci: build gfxreconstruct into the Vulkan testing container
- gitlab-ci: build VulkanTools into the Vulkan testing container
- gitlab-ci: Change devices format to <api-vendor-deviceId>
- gitlab-ci: Add gfxreconstruct traces support
- gitlab-ci: Add jobs to be able to test Vulkan
- gitlab-ci: Fix indentation and dangerous "" in the last multiline line
- gitlab-ci: Remove unneeded python3-pilkit dependency
- gitlab-ci: Sort packages to install alphabetically
- gitlab-ci: add python3-requests to the test-vk container
- gitlab-ci/traces: Add Vulkan sample entries for POLARIS10
- gitlab-ci: Don't use buster-backports packages by default for x86_test-vk
- gitlab-ci: add Wine, win64's apitrace and DXVK to the Vulkan testing container
- gitlab-ci: add apitrace's DXGI traces support
- gitlab-ci: replay apitrace traces in headless mode
- gitlab-ci: add Wine and DXVK env variables to Vulkan's tracie runner
- gitlab-ci/traces: Add D3D11 sample entry for POLARIS10
- gitlab-ci: Vulkan tracie runner to return last command exit code
- gitlab-ci: protect usage of shell variables with double quotes
- gitlab-ci: make explicit tracie is gitlab specific
- gitlab-ci: adapt query_traces_yaml to gitlab specific changes
- gitlab-ci: install winehq-stable to get 5.0 instead of 4.0
- Revert "meson,ci: Disable sparse_array tests on windows"
- gitlab-ci: update tracie README after changes in main script
- gitlab-ci: create always the "results" directory with tracie
- gitlab-ci: correct tracie behavior with replay errors

Andrii Simiklit (2):

- Revert "glx: convert glx_config_create_list to one big calloc"
- i965/vec4: Ignore swizzle of VGRF for use by var_range_end()

Anuj Phogat (2):

- intel/gen12+: Reserve 4KB of URB space per bank for Compute Engine
- intel/gen12+: Set way_size_per_bank to 4

Arcady Goldmints-Orlov (7):

- compiler/nir: Add support for variable initialization from a pointer
- compiler/spirv: Add support for non-constant initializers
- Rename nir_lower_constant_initializers to nir_lower_variable_initalizers

- spirv: Remove outdated SPIR-V decoration warnings
- nir: Lower returns correctly inside nested loops
- anv: increase minUniformBufferOffsetAlignment to 64
- intel/compiler: fix alignment assert in nir_emit_intrinsic

Axel Davy (1):

- gallium/util: Fix leak in the live shader cache

Bas Nieuwenhuizen (29):

- radv: Allow non-dedicated linear images and buffer.
- radv: Do not set SX DISABLE bits for RB+ with unused surfaces.
- radv: Optimize emitting index buffer changes.
- radv: Do not redundantly set the RB+ regs on pipeline switch.
- radeonsi: Fix compute copies for subsampled formats.
- amd/llvm: Fix divergent descriptor indexing. (v3)
- amd/llvm: Fix divergent descriptor regressions with radeonsi.
- radv: Store 64-bit availability bools if requested.
- radv: Consider maximum sample distances for entire grid.
- radv: Whitespace fixup.
- radv: Use correct buffer count with variable descriptor set sizes.
- winsys/amdgpu: Retrieve WC flags from imported buffers.
- drm-uapi,radv,radeonsi: Add amdgpu_drm.h header.
- vulkan/wsi: Add callback to set ownership of buffer.
- radv: Add WSI buffers to BO list only if they can be used.
- st/dri: Set next in template instead of after creation. (v2)
- radeonsi: Count planes for imported textures.
- radv: Use actual memory type count for setting app-visible bitset.
- radv: Stop using memory type indices.
- radv/winsys: Add function to get domains/flags from fd.
- radv: Determine memory type for import based on fd.
- radv: Expose 4G element texel buffers.
- radv: Fix implicit sync with recent allocation changes.
- radv: Extend tiling flags to 64-bit.
- radv: Provide a better error for permission issues with priorities.
- radv/winsys: Remove extra sizeof multiply.
- radv: Handle failing to create .cache dir.
- radv: Do not close fd -1 when NULL-winsys creation fails.
- radv: Implement vkGetSwapchainGrallocUsage2ANDROID.

Bernd Kuhls (1):

- util/os_socket: Include unistd.h to fix build error

Blaž Tomažič (1):

- radeonsi: Fix omitted flush when moving suballocated texture

Boris Brezillon (45):

- pan/midgard: Add an enum to describe the render targets
- pan/midgard: Make sure we pass the right RT id to emit_fragment_store()
- pan/midgard: Lower bitfield extract to shifts
- pan/midgard: Don't check 'branch && branch->writeout' twice in mir_schedule_alu()
- pan/midgard: Stop leaking instruction objects in mir_schedule_alu()
- panfrost: Fix the damage box clamping logic
- pan/midgard: Turn Z/S stores into zs_output_pan intrinsics
- pan/midgard: Add nir_intrinsic_store_zs_output_pan support
- panfrost: Z24 variants should be sampled as R32UI
- panfrost: Add the MALI_WRITES_{Z,S} flags
- panfrost: Set the MALI_WRITES_{Z,S} flags when needed
- Revert "panfrost: Z24 variants should be sampled as R32UI"
- panfrost: Pass the sampler view format when creating a tex descriptor
- panfrost: Assign primitive_size.pointer only if writes_point_size() returns true
- panfrost: Add an helper to retrieve the currently active shader state
- panfrost: Move the batch stack size adjustment out of panfrost_queue_draw()
- panfrost: Move viewport desc emission out of panfrost_emit_for_draw()
- panfrost: Move the const buf emission logic out of panfrost_emit_for_draw()
- panfrost: Move shared mem desc emission out of panfrost_launch_grid()
- panfrost: Dissociate shader meta patching from the desc emission
- panfrost: Move panfrost_attach_vt_framebuffer() to pan_cmdstream.c
- panfrost: Stop using panfrost_emit_for_draw() for compute jobs
- panfrost: Simplify panfrost_emit_for_draw() and make it private
- panfrost: Add an helper to update the occclusion query part of a tiler job desc
- panfrost: Add an helper to update the rasterizer part of a tiler job desc
- panfrost: Prepare things to get rid of panfrost_shader_state.tripipe
- panfrost: Prepare shader_meta descriptors at emission time
- panfrost: Add a panfrost_sampler_desc_init() helper
- panfrost: Move sampler/tex descs emission helpers to pan_cmdstream.c
- panfrost: Add an helper to emit a pair of vertex/tiler jobs
- panfrost: Drop initial mali_attr_meta.src_offset assignment

- panfrost: Ignore BO start addr when adjusting src_offset
- panfrost: Prepare attribute for builtins at state creation time
- panfrost: Emit attribute descriptors after patching the templates
- panfrost: Move the mali_attr.src_offset adjustment to a sub-function
- panfrost: Rename panfrost_stage_attributes()
- panfrost: Move streamout offset update out of panfrost_draw_vbo()
- panfrost: Move vertex/tiler payload initialization out of panfrost_draw_vbo()
- panfrost: Inline panfrost_queue_draw() and panfrost_emit_for_draw()
- panfrost: Move panfrost_emit_vertex_data() to pan_cmdstream.c
- panfrost: Move panfrost_emit_varying_descriptor() to pan_cmdstream.c
- panfrost: Re-init the VT payloads at draw/launch_grid() time
- panfrost: Use ctx->active_prim in panfrost_writes_point_size()
- panfrost: Get rid of ctx->payloads[]
- vtn/opencl: add rint-support

Brian Ho (17):

- turnip: Promote tu_cs_get_size/is_empty to header
- turnip: Execute main cs for secondary command buffers
- turnip: Advertise 8 bit subpixel precision
- ir3: Disable copy prop for immediate ldlw offsets
- turnip: Set has_gs in ir3_shader_key
- turnip: Emit geometry shader obj and related consts
- turnip: Configure VPC for geometry shaders
- turnip: Configure VFD_CONTROL with gsheader and primitiveid
- turnip: Set up REG_A6XX_SP_GS_CONFIG
- turnip: Selectively configure GRAS_LAYER_CNTL
- turnip: Update maxGeometryShaderInvocations to match blob
- turnip: Populate tu_pipeline.active_stages
- turnip: Enable geometry shaders for CP_DRAWs
- turnip: Enable geometryShader device feature
- turnip: Correctly set layer stride for 3D images
- turnip: Emit geometry shader descriptor consts
- freedreno/turnip: Update GRAS_LAYER_CNTL to GRAS_MAX_LAYER_INDEX

Caio Marcelo de Oliveira Filho (46):

- anv: Advertise VK_KHR_shader_non_semantic_info
- radv: Advertise VK_KHR_shader_non_semantic_info
- intel/gen12: Take into account opcode when decoding SWSB

- spirv: Be consistent when checking for Shader/Kernel
- anv: Use intel_debug_flag_for_shader_stage()
- anv: Add pipe_state_for_stage() helper
- nir/builder: Add nir_scoped_memory_barrier()
- nir: Add the alias NIR_MEMORY_ACQ_REL
- nir/tests: Use nir_scoped_memory_barrier() helper
- nir, intel: Move use_scoped_memory_barrier to nir_options
- anv: Remove unused field xfb_used from anv_pipeline
- anv: Remove unused field ‘urb.total_size‘
- nir: Don’t skip a bit in nir_memory_semantics
- nir: Reorder nir_scopes so wider scope has larger numeric value
- nir: Add pass to combine adjacent scoped memory barriers
- intel/fs: Combine adjacent memory barriers
- anv: Add a new enum to identify the pipeline type
- anv: Use pipeline type to decide whether or not lower multiview
- anv: Use a dynamic array for storing executables in pipeline
- anv: Keep the shader stage in anv_shader_bin
- anv: Pass the right pipe_state to flush_descriptor_sets()
- anv: Remove redundant check in flush_descriptor_sets() helpers
- anv: Decouple flush_descriptor_sets() helpers from pipeline struct
- anv: Decouple flush_descriptor_sets() from pipeline struct
- anv: Use a separate field in the pipeline for compute shader
- anv: Split graphics and compute bits from anv_pipeline
- anv: Reduce compute pipeline batch_data size
- anv: Remove duplicate code in anv_cmd_buffer_bind_descriptor_set
- intel/blorp: Plumb the stage through blorp upload_shader
- mesa/main: Fix overflow in validation of DispatchComputeGroupSizeARB
- nir: Add per_view attribute to nir_variable
- intel/gen12: Add XML description for 3DSTATE_PRIMITIVE_REPLICATION
- intel/fs: Allow multiple slots for position
- anv/gen12: Lower VK_KHR_multiview using Primitive Replication
- intel/compiler: Replace cs_prog_data->push.total with a helper
- anv: Stop using cs_prog_data->threads
- iris: Stop using cs_prog_data->threads
- intel/compiler: Remove cs_prog_data->threads
- intel/fs,vec4: Properly account SENDs in IVB memory fence

- spirv: Fix propagation of OpVariable access flags
- spirv: Handle instruction aliases in vtn_gather_types
- spirv: Update the headers from latest Khronos master
- intel/fs: Allow FS_OPCODE_SCHEDULING_FENCE stall on registers
- intel/fs,vec4: Pull stall logic for memory fences up into the IR
- intel/fs: Only stall after sending all memory fence messages
- i965: Use correct constant for max_variable_local_size

Chad Versace (12):

- anv: Drop unused anv_image_get_surface_for_aspect_mask()
- anv: Rename param make_surface::dev to device
- anv: Delete anv_image::ccs_e_compatible
- anv: Clarify behavior of anv_image_aspect_to_plane()
- anv: Respect ISL_SURF_USAGE_DISABLE_AUX_BIT in make_surface()
- turnip: Add magic register values to tu_physical_device
- turnip: Add a618 support
- anv: Drop anv_image.c:get_surface()
- anv: Add anv_image_plane_needs_shadow_surface() (v2)
- anv: Refactor creation of aux surfaces (v2)
- anv: Flatten the logic add_aux_surface_if_supported (v3)
- anv: Use isl_drm_modifier_get_default_aux_state()

Chia-I Wu (2):

- egl/android: require ANDROID_native_fence_sync for buffer age
- egl/android: enable/disable KHR_partial_update correctly

Chris Lord (2):

- vc4: fix vc4_yuv_blit overwriting fragment constant buffer slot 0
- vc4: Fix query_dmabuf_modifiers mis-reporting external_only property

Chris Wilson (1):

- iris: Fix import sync-file into syncobj

Christian Gmeiner (44):

- etnaviv: enable texture upload memory throttling
- etnaviv: update headers from rnndb
- etnaviv: fix alpha test on GC3000
- etnaviv: add etna_constbuf_state object
- etnaviv: ask kernel for max number of supported varyings
- etnaviv: update headers from rnndb
- etnaviv: increase number of supported varyings to 16

- etnaviv: implement emit_string_marker
- etnaviv: get rid of etna_spec in etna_context
- etnaviv: enable shareable shaders
- freedreno: calculate modified bit mask only once
- freedreno: simplify fd_set_shader_buffers(..)
- freedreno: ssbo: keep track if a buffer gets written
- freedreno: ssbo: mark resource read or written depending on usage
- etnaviv: get rid of SE_CLIP_*
- etnaviv: rework clippling calculation to be a derived state
- etnaviv: do the left shift by 16 at emit time
- etnaviv: get rid of struct compiled_scissor_state
- etnaviv: s/scissor_s/scissor
- etnaviv: compiled_framebuffer_state: get rid of SE_SCISSOR_*
- etnaviv: rename hw queries to acc queries
- etnaviv: rework etna_acc_sample_provider
- etnaviv: explicitly call resource_written(..)
- etnaviv: reset no_wait_cnt after triggered flush
- etnaviv: rework wait/flush logic
- etnaviv: extend acc query provider with supports(..) function
- etnaviv: make use of a fixed size array to track of all acc query provider
- etnaviv: extend result(..) to return if data is ready
- etnaviv: extend acc sample provide with an allocate(..)
- etnaviv: move generic perfmon functionality into own file
- etnaviv: convert perfmon queries to acc queries
- etnaviv: drop redundant calls to etna_acc_query_suspend(..)
- etnaviv: change begin_query(..) to a void function
- etnaviv: remove the "active" member of queries
- etnaviv: anisotropic filtering is supported starting with HALTI0
- etnaviv: update headers from rnndb
- etnaviv: add anisotropic filter support
- docs/features: mark GL_ARB_texture_filter_anisotropic as done for etnaviv
- etnaviv: drop default state for FE_HALTI5_ID_CONFIG
- etnaviv: call util_blitter_save_fragment_constant_buffer_slot(..)
- etnaviv: support for using generic blit path
- ci: bare-metal: power down device after tests
- etnaviv: fix SAMP_ANISOTROPY register value

- etnaviv: do not use int filter when anisotropic filtering is used

Christopher Egert (1):

- radv: use util_float_to_half_rtz

Christopher James Halse Rogers (1):

- egl/wayland: Fix zwp_linux_dmabuf usage

Connor Abbott (55):

- freedreno: Fix CP_COND_REG_EXEC bit positions
- freedreno: Add CP_REG_WRITE documentation
- freedreno: Fix CP_COND_EXEC
- tu: Move vsc_data and vsc_data2 allocation into the device
- tu: Don't emit initial render target state in tile_load_ib
- tu: Properly set UBWC flags in RB_RENDER_CNTL
- tu/blit: Support blits in secondary cmdstreams
- tu: Support multisample image clears
- tu: Disable linear depth attachments
- tu: Sysmem rendering
- tu: Add helper for CP_COND_REG_EXEC
- tu: Handle vkCmdClearAttachments() with sysmem
- tu: Support resolve ops with sysmem rendering
- tu: Support input attachments with sysmem
- tu: Force sysmem with mipmapped non-aligned linear stores
- tu: Rewrite border color handling
- lima/gpir: Make lima_gpir_node_insert_child() useful
- lima/gpir: Optimize conditional break/continue
- lima/gpir: Optimize nots created from branch lowering
- tu: Fix border color with compute shaders
- freedreno/fdl: Add base_align
- tu: Return the correct alignment for images
- freedreno: Cleanup event names
- freedreno: Rename RB_DONE_TS
- tu: Dump out shader assembly when requested
- tu: ir3: Emit push constants directly
- freedreno/a6xx: Add UBO size field
- freedreno/a6xx: Add registers for the bindless model
- ir3: Add bindless instruction encoding
- ir3: Plumb through support for a1.x

- ir3: Also don't propagate immediate offset with LDC
- ir3: LDC also has a destination
- ir3: Plumb through bindless support
- ir3: Rewrite UBO push analysis to support bindless
- tu: Switch to the bindless descriptor model
- tu: Emit CP_LOAD_STATE6 for descriptors
- tu: Add missing code for immutable samplers
- tu: Implement descriptor set update templates
- ir3: Fix txs with bindless
- ir3: Fix LDC offset units
- ir3: Handle load_ubo_ir3 when promoting to constants
- tu: Align GMEM resolve blit scissor
- tu: Use tu_cs_add_entries() with non-render-pass secondaries
- ir3/ra: Fix off-by-one issues with live-range extension
- freedreno/a6xx: Expand various varying-count bitfields
- tu: Fix the advertised maxFragmentInputComponents
- ir3: Don't double-insert the first block
- ir3: Fix bug with shaders that only exit via discard
- freedreno/a6xx: Document PrimID passthrough registers
- ir3: Skip missing VS outputs in VS out map when linking
- tu: Implement PrimID passthrough
- freedreno/a6xx: Implement PrimID passthrough
- st/nir: Fix assigning PointCoord location with !PIPE_CAP_TEXCOORD
- ir3: Remove VARYING_SLOT_PNTC remapping hack
- tu: Don't invert point coords

D Scott Phillips (6):

- intel/tools/aubinator_error_decode: read HW Context before other batches
- intel/tools/aubinator_error_decode: Decode ring buffers from HEAD to TAIL
- util/sparse_array: don't stomp head's counter on pop operations
- intel/fs: Update location of Render Target Array Index for gen12
- anv,iris: Fix input vertex max for tcs on gen12
- anv/gen11+: Disable object level preemption

Daniel Schürmann (73):

- aco: fix image_atomic_cmp_swap
- nir: gather info whether a shader uses demote_to_helper
- nir: add pass to lower discard() to demote()

- amd/llvm: implement nir_intrinsic_demote(_if) and nir_intrinsic_is_helper_invocation
- radeonsi: lower discard to demote when FS_CORRECT_DERIVS_AFTER_KILL is enabled
- radv: use nir_lower_discard_to_demote to work around game bugs
- amd: join emit_kill() from radv and radeonsi in ac_nir_to_llvm
- nir: fix unpack_64_4x16 in lower_alu_to_scalar()
- aco: add comparison operators for PhysReg
- aco: add sub-dword regclasses
- aco: refactor regClass setup for subdword VGPRs
- aco: validate p_create_vector with subdword elements properly
- aco: validate register alignment of subdword operands and definitions
- aco: validate uninitialized operands
- aco: validate RA of subdword assignments
- aco: print subdword registers
- aco: fix Temp and assignment of renamed operands during RA
- aco: remove unnecessary reg_file.fill() operation in get_reg_create_vector()
- aco: add notion of subdword registers to register allocator
- aco: create helper function to collect variables from register area
- aco: adapt register allocation for subdword registers
- aco: align subdword registers during RA when necessary
- aco: small refactoring of shuffle code lowering
- aco: add builder function for subdword copy()
- aco: lower subdword shuffles correctly.
- aco: don't propagate SGPRs into subdword PSEUDO instructions
- aco: don't assume split_vector(create_vector) has the same number of elements when optimizing
- aco: don't vectorize 8/16bit load/store_ssbo
- aco: add missing conversion operations for small bitsizes
- aco: add byte_align_scalar() & trim_subdword_vector() helper functions
- aco: prepare helper functions for subdword handling
- aco: implement vec2/3/4 with subdword operands
- aco: implement storagePushConstant8 & storagePushConstant16
- aco: implement 8bit/16bit load_buffer
- aco: implement 8bit/16bit store_ssbo
- aco: use MUBUF to load subdword SSBO
- aco: guarantee that Temp fits in 4 bytes
- aco: add explicit padding for all Instruction sub-structs
- aco: improve hashing for value numbering

- aco: improve register assignment when live-range splits are necessary
- aco: replace assignment hashmap by std::vector in register allocation
- aco: during RA only insert into renames table if a variable got renamed
- aco: improve speed of live_var_analysis
- aco: refactor try_remove_trivial_phi() in RA
- aco: change some std::map to std::unordered_map in register_allocation
- aco: change live_out variables to std::unordered_set
- aco: move all needed helper containers to ra_ctx
- aco: RA - move all std::function objects into proper functions
- aco: setup subdword regclasses for ssa_undef & load_const
- aco: ensure correct bit representation of subdword constants
- aco: don't constant-propagate into subdword PSEUDO instructions
- aco: lower subdword phis with SGPR operands
- aco: rename aco_lower_bool_phis() -> aco_lower_phis()
- aco: make some reg_file helpers private and fix their uses
- aco: fix p_extract_vector optimization in presence of unequally sized vector operands
- aco: use v_subrev_f32 for fsub with an sgpr operand in src1
- aco: fix 64bit fsub
- aco: move src1 to vgpr instead of using VOP3 for VOP2 instructions during isel
- aco: simplify operand handling in RA
- aco: refactor get_reg() to take Temp instead of RegClass
- aco: refactor get_reg() to also handle affinities
- aco: create pseudo dummy instruction in RA to be used for live-range splits
- aco: create and use DefInfo struct in RA
- aco: use DefInfo in more places to simplify RA
- aco: move attempt to find strided register into get_reg_simple()
- aco: allocate full register for subdword definitions if HW doesn't support it
- aco: don't create vector affinities for operands which are not killed or are duplicates
- aco: refactor get_reg_simple() to return early on exact matches
- aco: stop get_reg_simple after reaching max_used_gpr
- aco: try to always find a register with stride for even sizes
- aco: use upper part of gap in register file if it is beneficial for striding
- aco: coalesce v_mad's accumulator with definition's affinities
- aco: either copy-propagate or inline create_vector operands

Daniel Stone (15):

- Revert "gitlab-ci: disable panfrost runners"

- egl/wayland: Don't invalidate buffers on no-op resize
- util/test: Use MAX_PATH on Windows
- CI: Add native Windows VS2019 build
- CI: Windows: Fix Docker tag argument inversion
- CI: Disable Panfrost Mali-T820 jobs
- CI: Avoid htz4 runner for VS2019
- meson: Add VS 4624 warning exclusion to remove piles of LLVM warnings
- CI: Re-enable Windows VS2019 builds
- EGL: Add eglSetDamageRegionKHR to GLVND dispatch list
- meson: Make shared-llvm into a tri-state boolean
- CI: Disable Windows/VS2019 builds
- Revert "CI: Disable Windows/VS2019 builds"
- ci/windows: Make Chocolatey installs more reliable
- CI: Disable Lima jobs due to lab unhealthiness

Danylo Piliaiev (29):

- i965: Do not set front_buffer_dirty if there is no front buffer
- st/mesa: Handle the rest renderbuffer formats from OSMesa
- osmesa/tests: Cover OSMESA_RGB GL_UNSIGNED_BYTE case
- st/nir: Unify inputs_read/outputs_written before serializing NIR
- brw_nir: Cast bitshift to unsigned
- brw_fs: Avoid zero size vla
- intel/compiler: Do not qsort zero sized array
- intel/bufmgr: Cast bitshift to unsigned
- glsl/blob: Do not call memcpy if there is nothing to copy
- iris: Do not dereference nullptr with pipe_reference
- i965: Do not generate D16 B5G6R5_UNORM configs on gen < 8
- intel/tools: Fix compilation with UBSan
- glsl: do not crash if string literal is used outside of #include/#line
- st/mesa: Fix signed integer overflow when using util_throttle_memory_usage
- intel/aub_viewer: Fix format specifier for uint64_t
- nir: Fix breakage of foreach_list_typed_safe assumptions in loop unrolling
- anv: Do not sample from 3d depth image with HiZ
- glsl/list: Fix undefined behaviour of foreach_* macros
- st/mesa: Update shader info of ffvp/ARB_vp after translation to NIR
- st/mesa: Re-assign vs in locations after updating nir info for ffvp/ARB_vp
- spirv: Expand workaround for OpControlBarrier on old GLSLang

- st/mesa: Treat vertex inputs absent in inputMapping as zero in mesa_to_tgsi
- iris/bufmgr: Check if iris_bo_gem_mmap failed
- i965: Fix out-of-bounds access to brw_stage_state::surf_offset
- anv: Translate relative timeout to absolute when calling anv_timelines_wait
- anv: Fix deadlock in anv_timelines_wait
- meson: Disable GCC's dead store elimination for memory zeroing custom new
- mesa: Fix double-lock of Shared->FrameBuffers and usage of wrong mutex
- intel/fs: Work around dual-source blending hangs in combination with SIMD16

Dave Airlie (69):

- llvmpipe/query: add support for indexed queries
- gallivm/swr: add stream_id to geom epilogue emit
- gallivm/nir: add support for multiple vertex streams
- draw: change geom shader output to an array of outputs.
- draw/gs: track emitted prims + verts per stream.
- draw: emit multiple streams to streamout.
- draw: don't emit vertex to streams with no outputs
- llvmpipe: advertise 4 vertex streams
- gallivm/s390: fix pass init order on s390 with llvm 8 (v2)
- ci: bump debian image and change llvm deps to 8
- dri: add another get shm variant.
- glx/drisw: add getImageShm2 path
- glx/drisw: return false if shmid == -1
- glx/drisw: fix shm put image fallback
- gallivm/tgsi: fix stream id regression
- gallivm/nir: fix integer divide SIGFPE
- gallivm/nir: handle mod 0 better.
- gallium/auxiliary: add the microsoft tessellator and a pipe wrapper.
- gallivm/nir: split out 64-bit splitting code
- gallivm/nir: add support for tess system values
- gallivm/nir: align store_var param order with load_var
- gallivm/tgsi/swr: add mask vec to the tcs store
- gallivm/nir: add tessellation i/o support.
- draw: add JIT context/functions for tess stages.
- draw: add main tessellation code
- draw: hook up final bits of tessellation
- gallium/nir/tgsi: only scan fragment shader inputs for usage_mask

- llvmpipe: add support for tessellation shaders
- gallivm/tessellator: use private functions for min/max to avoid namespace issues
- gallium: fix build with latest meson and gcc10
- gallivm/s3tc: split out dxt5 alpha code
- gallivm: add support for rgtc/latc fetches.
- gallium/llvmpipe: add an optimised 32-bit memset
- gallivm/rgtc: fix the truncation to 8-bit
- gallivm/rgtc: enable fast path for snorm types.
- Revert "gallivm: disable rgtc/latc SNORM accellerated fetches"
- llvmpipe: fixup context leaks.
- draw: collect tessellation invocations statistics
- llvmpipe: report tessellation shader statistics.
- llvmpipe/query: fix transform feedback overflow any queries.
- gallivm: fix left over shader vote debug
- gallivm/nir: lower implicit lod to tex.
- gallivm/draw: calloc prim id toavoid undef
- llvmpipe: fix no tokens detections.
- draw: fix tessellation stats query
- llvmpipe/setup: move line stats collection earlier.
- draw/cull: run pipeline for culled points.
- draw: fix user culling pipeline order. (v2)
- u_blitter: fix stencil blitting
- draw: free the NIR IR.
- draw/tess: free the NIR
- llvmpipe/nir: free the nir shader
- nir/linking: fix issue with two compact variables in a row. (v2)
- gallivm/nir: fix image store conversions
- gallivm/nir: add helper invocation support
- util/indirect: handle stride less than number of parameters.
- llvmpipe: bump max images to 16
- llvmpipe: fix ssbo alignment
- draw/tess: fix TES patch vertices in.
- llvmpipe: fix d32 unorm depth conversions.
- llvmpipe/setup: add point size clamping
- llvmpipe: enable stencil only formats. (v2)
- llvmpipe: clamp color storage for integer types.

- gallivm: fix stencil border
- vulkan: add initial device selection layer. (v6.1)
- ci: add llvmpipe paths to virgl rules
- draw/tess: free tessellation control shader i/o memory.
- llvmpipo/nir: free compute shader NIR
- llvmpipe: compute shaders work better with all the threads.

David Stevens (1):

- egl/android: set window usage flags

Denys (1):

- gitlab: add bug report template

Dominik Behr (1):

- meson: fix debug build on Android

Drew Davenport (1):

- radv: Filter extensions not whitelisted for Android

Duncan Hopkins (2):

- zink. Added storage CISto descriptor pool. Added storage in descriptor pool for combined image samplers as well as uniform buffers. Stops some shaders from running through a pools storage faster than zinks internal tracking.
- zink: zero out zink_render_pass_state

Dylan Baker (48):

- docs/release-calendar: 20.0.0-rc1 has been released
- docs: Mark 20.0-rc2 as done
- docs: Add release notes for 19.3.4
- docs: Add SHA256 sum for 19.3.4
- docs: Mark 19.3.4 as done
- docs: Mark 20.0.0-rc3 as done
- Docs: Add 20.0.0 release notes
- docs: Update index, relnotes, and release-calendar for 20.0
- docs: Update stable process around using fixes: and gitlab
- docs/submittingpatches: Fix confusing typo + missing pronoun
- docs: Update release notes with current process
- bin/post_version.py: Update the release calendar as well
- bin/post_version.py: Pretty print the html
- bin/post_version.py: Make the git commit as well.
- docs: update releasing to cover updated post_version.py
- docs: add relnotes for 20.0.1
- docs: Add sha256sums for 20.0.1

- docs: update news, calendar, and link release notes for 20.0.1
- Docs: Add release notes for 20.0.2
- docs/relnotes: Add sha256 sums for 20.0.2
- docs: update calendar, add news item, and link releases notes for 20.0.2
- docs/release-calendar: Add calendar for 20.1 Release candidates
- bin/gen_release_notes.py: Fix version detection for .0 release
- bin/pick-ui: Add a new maintainer script for picking patches
- replace _mesa_is_pow_two with util_is_power_of_two_*
- replace _mesa_next_pow_two_* with util_next_power_of_two_*
- replace _mesa_logbase2 with util_logbase2
- replace LOG2 with util_fast_log2
- u_math: add x86 optimized version of ifloor
- replace IFLOOR with util_ifloor
- Replace IROUND_POS with _mesa_roundevenf
- mesa/main: remove unused IROUNDD
- replace IROUND with util functions
- move windows strtok_r define to u_string
- Replace IS_INF_OR_NAN with util_is_inf_or_nan
- replace malloc macros in imports.h with u_memory.h versions
- util: Add an aligned realloc function
- replace imports memory functions with utils memory functions
- mesa|mapi: replace _mesa_[v]snprintf with [v]snprintf
- mesa: move ADD_POINTERS to macros.h
- dri/nouveau: replace assert with unreachable
- remove final imports.h and imports.c bits
- meson: update llvm dependency logic for meson 0.54.0
- docs: Add relnotes for 20.0.5
- docs: Add sha256 sums for 20.0.5
- docs: update calendar, add news item, and link releases notes for 20.0.5
- mesa: Follow OpenGL conversion rules for values that exceed storage size
- tests: Make tests aware of meson test wrapper

Edmondo Tommasina (1):

- radv/sqtt: fix RADV_THREAD_TRACE_BUFFER_SIZE spelling

Eduardo Lima Mitev (3):

- turnip/pipeline: Don't assume tu_shader is a valid object
- turnip: Instance can be NULL resolving 'GetInstanceProcAddr' entry point

- anv/radv: Resolving 'GetInstanceProcAddr' should not require a valid instance

Eli Schwartz (1):

- docs: fix typo in v20 release notes

Elie Tournier (3):

- spirv2nir: print nir shader if translation succed
- spirv2nir: Add kernel spirv support
- docs/features: Update virgl OpenGL 4.5 features GL_ARB_clip_control and GL_KHR_robustness are now expose in the guest.

Emil Velikov (11):

- meson: glx: drop with_glx == dri check
- glx: set the loader_logger early and for everyone
- egl/drm: reinstate (kms_)swrast support
- Revert "egl/dri2: Don't dlclose() the driver on dri2_load_driver_common failure"
- loader: use a maximum of 64 drmDevices
- loader: simplify loader_get_user_preferred_fd()
- loader: simplify codeflow in drm_get_pci_id_for_fd
- loader: move "using driver…" message to loader_get_kernel_driver_name
- loader: fallback to kernel name, if PCI fails
- glx: omit loader_loader() for macOS
- egl: simplify client/platform extension handling

Emmanuel Gil Peyrot (1):

- Expose EGL_KHR_platform_* when EXT is supported

Eric Anholt (144):

- gallium/osmesa: Fix a typo in the unit test's test names.
- gallium/osmesa: Fix MakeCurrent of non-8888 contexts.
- gallium/osmesa: Fill out other format tests.
- gallium/osmesa: Try to fix the test for big-endian.
- util: Make helper functions for pack/unpacking pixel rows.
- mesa/st: Use direct util_format_pack/unpack instead of u_tile.
- gallium/util: Remove pipe_get_tile_z/put_tile_z.
- softpipe: Drop the raw_to* part of the tile cache interface.
- softpipe: Refactor pipe_get/put_tile_rgba_* paths.
- gallium: Add and use a helper for packing uc from a color_union.
- gallium: Refactor some single-pixel util_format_read/writes.
- util: Drop unpacking from int signed to unsigned and vice versa.
- freedreno: Move the layout debug under FD_MESA_DEBUG=layout.

- freedreno: Include the layer size in layout debug.
- freedreno: Rename the UBWC layer size field and store it as bytes.
- freedreno/a6xx: Disable the core layer-size setup.
- freedreno: Swap the whole resource layout in shadowing.
- freedreno: Blit all array levels when uncompressing UBWC.
- freedreno: Disable UBWC on Z24S8 if not TEXTURE_2D.
- freedreno: Allow UBWC on textures with multiple mipmap levels.
- mesa: Clean up some endianness adapters for shader image formats.
- intel/isl: Move iris's pipe-to-isl format function to isl.
- glsl,nir: Switch the enum representing shader image formats to PIPE_FORMAT.
- mesa/st: Move the SYSTEM_VALUE -> TGSI_SEMANTIC map to tgsi_from_mesa.
- nouveau: Reuse tgsi_get_sysval_semantic().
- nouveau: reuse tgsi_get_gl_frag_result_semantic().
- nouveau: Reuse tgsi_get_gl_varying_semantic().
- u_tile: Skip the packed temporary and just store tiles directly.
- ci: Disable a bunch of tests on freedreno a630.
- ci: Bump the GLES CTS version to 3.2.6.1.
- Revert "gallium: Fix big-endian addressing of non-bitmask array formats."
- ci: Extend the a630 flake list to reduce spurious failures.
- radv: Squelch possibly-undefined warning
- llvmpipe: Fix real uninitialized use of "atype" for SEMANTIC_FACE
- llvmpipe: Silence "possibly uninitialized value" warning for ssbo_limit.
- llvmpipe: Silence uninitialized variable warning about "chan"
- llvmpipe: Fix warning about uninitialized "op" in the NIR path.
- llvmpipe: Silence uninitialized variable warning about "vals"
- llvmpipe: Silence uninitialized variable warning about "scissor"
- llvmpipe: Fix another uninitialized value warning, on init_val.
- gallium: Only define PIPE_ALIGNSTACK on x86.
- ci: prepare-artifacts: Make the indent here match previously in the file
- ci: Make sure that we have a proper shell prompt for LAVA.
- ci: Make LAVA job fails emit the full list of unexpected test results.
- ci: Document how LAVA runners work.
- ci: Don't bother generating deqp junit results since we don't present it.
- ci: Remove a useless filtering of the lava logs.
- nir: Rename gl_nir_lower_bindless_images.c in preparation for extending it.
- nir: Make image lowering optionally handle the !bindless case as well.

- gallium: Add a cap for enabling lowering of image load/store intrinsics.
- v3d: Ask the state tracker to lower image accesses off of derefs.
- glsl: Factor out the sampler dim coordinate components switch statement.
- spirv_to_nir: Reuse glsl_sampler_dim_coordinate_components().
- freedreno/ir3: Reuse glsl_get_sampler_dim_coordinate_components() in tex_info.
- tgsi_to_nir: Reuse glsl_get_sampler_dim_coordinate_components().
- prog_to_nir: Reuse glsl_get_sampler_dim_coordinate_components().
- freedreno/ir3: Fix the arg to ir3_get_num_components_for_image_format()
- nir: Move intel's intrinsic_image_coordinate_components() to core nir.
- freedreno: Switch to using lowered image intrinsics.
- ci: Blacklist another freedreno flaky test.
- meson: Disable bison's -Wdeprecated since we still support old bison.
- turnip: Fix compiler warning about casting a nondispatchable handle.
- freedreno/computerator: Fix defined-but-not-used warnings from lex/yacc.
- ci: Remove LLVM from ARM test drivers.
- ci: Stop disabling ACPI in the LAVA arm64 kernel build.
- ci: Shrink the arm64 kernel build a bit.
- ci: Include db410c support in the ARM container.
- aco: Fix signed-vs-unsigned warning.
- ci: Enable -Werror on meson-vulkan and meson-testing.
- ci: Switch testing on db410c over to LAVA.
- ci: Add a disabled-by-default job for GLES3 testing on db410c.
- ci: Flip db410c back to docker mode.
- ci: Print the renderer/version that our dEQP invocation is using.
- ci: Fix installation of firmware for db410c's nic.
- ci: Make a simple little bare-metal fastboot mode for db410c.
- glsl/tests: Catch mkdir errors to help explain when they happen.
- glsl/tests: Fix waiting for disk_cache_put() to finish.
- ci: Update the ci-templates commit.
- ci: Enable ccache in the container builds.
- ci: Enable ccaching of CMake builds as well.
- ci: Enable testing GLES2-3 on a530 (Dragonboard 820c).
- freedreno/a5xx: Fix min-vs-mag filtering decisions on non-mipmap tex.
- gallium/util: Switch util_float_to_half to _mesa_float_to_half()'s impl.
- ci: Ban the recent popular freedreno a630 flakes.
- ci: Disable tests that showed intermittent fails on a530 in day 1.

- ci: Only run the freedreno baremetal tests when freedreno/core changes.
- freedreno: Switch to exposing only half-integer pixel centers.
- ci: Move db820c and db410c's gles3 tests to manual, like radv did.
- glsl: Restore the IsES flag on the shader when reading from cache.
- ci: Ban the recent popular freedreno a630 intermittent failure.
- freedreno: Remove always-true return from per-gen begin_query.
- freedreno: Remove the "active" member of queries.
- freedreno: Fix acc query handling in the presence of batch reordering.
- freedreno: Associate the acc query bo with the batch.
- freedreno: Count blits in GL_TIME_ELAPSED and perf counter queries.
- freedreno/a6xx: Fix timestamp queries.
- freedreno: Rename "is_blit" to "is_discard_blit"
- freedreno: Fix detection of being in a blit for acc queries.
- freedreno: Work around UBWC flakiness.
- freedreno: Drop an unnecessary include marked "this should go away"
- freedreno/turnip: Use the NIR info to decide if we need helper invocations.
- loader: Warn when we fail to open a device node due to permissions.
- ci: Consistently use -j4 across x86 build jobs and -j8 on ARM.
- freedreno/a6xx: Sink the per-level size temps inside the loop.
- freedreno/a6xx: Remove the "aligned_height" temporary.
- freedreno/a6xx: Drop the "alignment" layout temporary.
- freedreno: Add the outline of a test for a6xx texture layout.
- freedreno/a6xx: Set a level's pitch based on minified level0 pitch, not width0.
- freedreno: Fix leak of binning shader variants.
- freedreno/ir3: Stop doing b2n on the SEL condition.
- freedreno/ir3: CSE the up/downconversion of SEL's cond's size.
- freedreno/a5xx+: Skip compiling the old gmem blit programs.
- freedreno/drm-shim: Add support for faking other adreno chips.
- freedreno/ir3: Drop handling FRAG_RESULT_DEPTH writing to .z
- freedreno: Introduce a "cpp_shift" value for cpp divs/muls.
- freedreno: Make the slice pitch be bytes, not pixels.
- drm-shim: Let the driver choose to overwrite the first render node.
- nir/lower_two_sided_color: Fix picking of new driver location.
- nir/lower_clip: Fix picking of unused driver locations.
- gallium: Fix setup of pstipple frag coord var.
- freedreno/ir3: Fix driver_location of the added vertex_flags varying.

- freedreno/ir3: Fix sizing of the inputs/outputs array.
- vc4: Use NIR shader's num_outputs for generating our new output.
- ci: Drop redundant freedreno stage specification.
- ci: Enable GLES3 testing on db410c/db820c (freedreno a306 and a530).
- freedreno: Fix derivatives without texturing on a3xx-a5xx.
- ci: Enable GLES 3.1 testing on db820c (a530).
- freedreno/ir3: Fix the disasm of half-float STG dests.
- freedreno/ir3: Print a space after nop counts, like qcom's disasm.
- freedreno/ir3: Add a unit test for our disassembler.
- freedreno/ir3: Convert remaining disasm src prints to reginfo.
- freedreno/ir3: Refactor out print_reg_src().
- freedreno/ir3: Add support for disasm of cat2 float32 immediates.
- ci: Enable –compact-display false on all dEQP runs.
- ci: Add sanity checking that dEQP gets the expected GL_RENDERER.
- freedreno: Fix calculation of the const buffer cmdstream size.
- ci: Allow namespacing of dEQP run results files.
- ci: Clean up some excessive use of pipes in dEQP results processing.
- ci/freedreno: Add a test run of a few driver options.
- util/ra: Sanity check that the driver selected a valid reg.
- util/ra: Sanity check that we're adding a valid reg to a class.
- util/ra: Use util_dynarray for the adjacency list.
- util/ra: Use util_dynarray for handling the conflict lists.
- util/ra: Improve ra_set_finalize() performance.

Eric Engestrom (58):

- VERSION: bump after 20.0 branch point
- egl: put full path to libEGL_mesa.so in GLVND json
- gitlab-ci: disable a630 tests as mesa-cheza is down
- util/os_socket: fix header unavailable on windows
- freedreno/perfcntrs: fix fd leak
- dri: delete gen-symbol-redefs.py
- util/disk_cache: check for write() failure in the zstd path
- meson: don't bother trying 'python2'
- Revert "egl: put full path to libEGL_mesa.so in GLVND json"
- egl: directly access static members instead of using _egl{Get,Set}ConfigKey()
- meson: explicitly disallow unsupported build directory layout
- docs: fix typos in the release docs

- bin/gen_release_notes.py: fix commit list command
- gen_release_notes: fix vulkan version reported
- docs/relnotes/19.3: fix vulkan version reported
- docs/relnotes/20.0: fix vulkan version reported
- Revert "docs/relnotes/19.3: fix vulkan version reported"
- docs: trivial fix for html structure
- docs/releasing: add missing tags
- docs: add release notes for 19.3.5
- docs: update calendar, add news item, and link releases notes for 19.3.5
- vulkan/wsi: fix cleanup when dup() fails
- gen_release_notes: fix version in "you should wait" message
- gen_release_notes: resolve ambiguity by renaming 'version' to 'previous_version' and 'next_version' to 'this_version'
- meson: use existing variables in inc_common
- meson: inline 'inc_common'
- vulkan: drop unused include directories
- intel: drop unused include directories
- scons: prune unused Makefile.sources
- docs: add release notes for 20.0.3
- docs/relnotes: add sha256sum for 20.0.3
- docs: update calendar, add news item, and link releases notes for 20.0.3
- docs: add release notes for 20.0.4
- docs/relnotes: add sha256sum for 20.0.4
- docs: update calendar, add news item, and link releases notes for 20.0.4
- glx: fix 630 times -Wlto-type-mismatch when building with LTO enabled
- glx: use anonymous namespace to avoid -Wodr issues when building with LTO enabled
- pick-ui: auto-scroll the feedback window
- pick-ui: compute .pick_status.json path only once
- pick-ui: make .pick_status.json path relative to the git root instead of the script
- pick-ui: show commit sha in the pick list
- VERSION: bump to 20.1.0-rc1
- .pick_status.json: Update to af55bdd05d94eda59ee1c9331a50045000da5db5
- .pick_status.json: Update to 57796946985de60204189426ca8eb7bbfa97c396
- .pick_status.json: Mark 3fac55ce0d066d767d6c6c8308f79d0c3e566ec0 as denominated
- .pick_status.json: Update to 29da52128090a1ef8ef782188c0f67c7f5ec8d19
- VERSION: bump to 20.1.0-rc2

- .pick_status.json: Update to 772b15ad3227e08bb4e18932ac9ecf4c29271160
- .pick_status.json: Update to 56f955e4850035d915a2a87e2ebea7fa66ab5e19
- .pick_status.json: Update to c1c0cf7a66905e8d7ad506842a41b0ad0c5b10da
- VERSION: bump to 20.1.0-rc3
- .pick_status.json: Update to 5a6beb6a24aa084adfd6c57edd0a64f0a044611a
- post_version.py: fix branch name construction for release candidates
- post_version.py: invert ‘is_point‘ into ‘is_first_release‘ to make its purpose clearer
- post_version.py: stop adding release candidates to the index and relnotes
- VERSION: bump to 20.1.0-rc4
- .pick_status.json: Update to a91306677c613ba7511b764b3decc9db42b24de1
- tree-wide: fix deprecated GitLab URLs

Erik Faye-Lund (154):

- zink: enable texture-buffer objects
- zink: implement load_instance_id
- zink: implement support for derivative-control
- zink: be more careful about the mask-check
- zink: disallow depth-stencil blits with format-change
- st/mesa: use uint-result for sampling stencil buffers
- zink: lower away fdph
- zink: fixup sampler-usage
- zink: replace unset buffer with a dummy-buffer
- zink: emit blend-target index
- zink: only inspect dual-src limit if feature enabled
- Revert “nir: Add a couple trivial abs optimizations”
- zink: do not use SpvDimRect
- zink: fix binding-usage
- zink: do not report texture-samplers for unsupported stages
- zink/spirv: do not reinvent store_dest
- zink/spirv: prefer store_dest over store_dest_uint
- zink/spirv: rename functions a bit
- zink/spirv: unit_value -> raw_value
- zink/spirv: uint -> raw
- zink: do not convert bools to/from uint
- util: promote u_debug_memory.c to src/util
- util: move debug_memory_{begin,end} to os_memory_debug.h
- gallium/util: do not use debug_print_format

- gallium/util: remove unused debug_print_foo helpers
- zink/spirv: do not use bitwise operations on booleans
- pipebuffer: clean up cast-warnings
- rbug: clean up cast-warnings
- rbug: do not return void-value
- vtn/opencl: fully enable OpenCLstd_Clz
- compiler/nir: move build_exp helper into builtin-builder
- compiler/nir: move build_log helper into builtin-builder
- vtn/opencl: add native exp/log-support
- vtn/opencl: add native exp10/log10-support
- vtn/opencl: add native exp2/log2-support
- nv50: remove unused variable
- meson: disable some more warnings on msvc
- mesa/main: correct extension-checks for GL_BLACKHOLE_RENDER_INTEL
- mesa/main: clean-up extension-checks for point-sprites
- mesa/main: clean up extension-check for GL_VERTEX_PROGRAM
- mesa/main: clean up extension-check for GL_VERTEX_PROGRAM_TWO_SIDE
- mesa/main: clean up extension-check for GL_VERTEX_PROGRAM_POINT_SIZE
- mesa/main: clean up extension-check for GL_TEXTURE_RECTANGLE
- mesa/main: clean up extension-check for GL_STENCIL_TEST_TWO_SIDE
- mesa/main: clean up extension-check for GL_DEPTH_BOUNDS_TEST
- mesa/main: clean up extension-check for AMD_depth_clamp_separate
- mesa/main: clean up extension-check for GL_FRAGMENT_SHADER_ATI
- mesa/main: clean up extension-check for GL_TEXTURE_CUBE_MAP_SEAMLESS
- mesa/main: clean up extension-check for GL_RASTERIZER_DISCARD
- mesa/main: clean up extension-check for GL_TEXTURE_EXTERNAL
- mesa/main: remove unused macro
- wgl: drop pointless debug_printf
- wgl: drop unused member
- wgl: move screen-init to a helper
- wgl: do not create screen from DllMain
- st/dri: make sure software color-buffers are linear
- zink: be less picky about tiled resources
- .mailmap: add an alias for Alan Swanson
- .mailmap: add an alias for Alyssa Rosenzweig
- .mailmap: add an alias for Andrii Simiklit

- .mailmap: add an alias for Anuj Phogat
- .mailmap: add an alias for Axel Davy
- .mailmap: add an alias for Boris Brezillon
- .mailmap: add an alias for Bruce Cherniak
- .mailmap: update aliases for Carl-Philip Hänsch
- .mailmap: add an alias for Chad Versace
- .mailmap: add a couple of aliases for Chandu Babu Namburu
- .mailmap: add alias for Chenglei Ren
- .mailmap: add an alias for Christian Gmeiner
- .mailmap: add an alias for Christian Inci
- .mailmap: add a few aliases for Christoph Haag
- .mailmap: add an alias for Colin McDonald
- .mailmap: specify spelling for Constantine Kharlamov
- .mailmap: add an alias for Craig Stout
- .mailmap: add an alias for Daniel Schürmann
- .mailmap: add an alias for Danylo Piliaiev
- .mailmap: add an alias for Dave Airlie
- .mailmap: add an alias for Dylan Baker
- .mailmap: add a couple of aliases for Dylan Noblesmith
- .mailmap: add an alias for Emmanuel Gil Peyrot
- .mailmap: add an alias for Erik Faye-Lund
- .mailmap: specify spelling for Francesco Ansanelli
- .mailmap: specify spelling for Gurchetan Singh
- .mailmap: add an alias for Haihao Xiang
- .mailmap: add an alias for Harish Krupo
- .mailmap: specify spelling for Heinrich Fink
- .mailmap: specify spelling for Henri Verbeet
- .mailmap: add an alias for Igor Gnatenko
- .mailmap: add an alias for Illia Iorin
- .mailmap: specify spelling for James Zhu
- .mailmap: add an alias for Jan Beich
- .mailmap: clean up aliases for Jeremy Huddleston
- .mailmap: add an alias for Julien Isorce
- .mailmap: add a few aliases for Karol Herbst
- .mailmap: add a few aliases for Kevin Rogovin
- .mailmap: add a few aliases for Kristian Høgsberg

- .mailmap: add an alias for Lionel Landwerlin
- .mailmap: specify spelling for Liviu Prodea
- .mailmap: update aliases for Marc-André Lureau
- .mailmap: add alias for Matthias Groß
- .mailmap: add an alias for Neha Bhende
- .mailmap: add an alias for Neil Roberts
- .mailmap: specify spelling for Nian Wu
- .mailmap: add an alias for Nicholas Bishop
- .mailmap: update aliases for Nicolai Hähnle
- .mailmap: add an alias for Philipp Zabel
- .mailmap: update aliases for Pierre-Eric Pelloux-Prayer
- .mailmap: add an alias for Plamena Manolova
- .mailmap: add an alias for Qiang Yu
- .mailmap: specify spelling for Randy Xu
- .mailmap: add an alias for Renato Caldas
- .mailmap: add an alias for Rob Clark
- .mailmap: add an alias for Rodrigo Vivi
- .mailmap: add an alias for Samuel Li
- .mailmap: add an alias for Sergii Romantsov
- .mailmap: specify spelling for Sonny Jiang
- .mailmap: add a couple of aliases for Steinar H. Gunderson
- .mailmap: add a couple of aliases for Suresh Guttula
- .mailmap: add an alias for Thierry Reding
- .mailmap: add an alias for Timo Aaltonen
- .mailmap: add a couple of aliases for Timothy Arceri
- .mailmap: add an alias for Tim Wiederhake
- .mailmap: add an alias for Tom Stellard
- .mailmap: add an alias for Tomasz Figa
- .mailmap: add an alias for Topi Pohjolainen
- .mailmap: add an alias for Vadym Shovkoplias
- .mailmap: add an alias for Varad Gautam
- .mailmap: specify spelling for Vivek Kasireddy
- .mailmap: specify spelling for Wladimir J. van der Laan
- .mailmap: add an alias for Xavier Bouchoux
- .mailmap: add an alias for Yaakov Selkowitz
- .mailmap: add alias for Zhaowei Yuan

- .mailmap: add an alias for Zhongmin Wu
- meson: use override_options to change warning-level
- wgl: silence some cast-warnings
- util/tests: initialize variable
- mesa: fixup cast expression
- vbo: avoid including wingdi.h on win32
- meson: tell flex that we support c99
- gtest: Update to 1.10.0
- meson: do not disable incremental linking for debug-builds
- docs: remove outdated sentence
- mesa/gallium: do not use enum for bit-allocated member
- meson: correct windows-version define
- mesa/main: do not store unrecognized extensions in context
- mesa/main: do not pass context to one-time extension init
- mesa/main: do not init remap-table per api
- mesa/main: Do not pass context to one_time_init
- mesa/main: one_time_init() -> _mesa_initialize()
- mesa/st: call _mesa_initialize() early
- zink: lower b2b to b2i
- util/os_memory: never use os_memory_debug.h
- zink: implement i2b1
- zink: use general-layout when blitting to/from same resource

Francisco Jerez (57):

- intel/fs/cse: Make HALT instruction act as CSE barrier.
- intel/fs/gen7: Fix fs_inst::flags_written() for SHADER_OPCODE_FIND_LIVE_CHANNEL.
- intel/fs: Add virtual instruction to load mask of live channels into flag register.
- intel/fs/gen12: Workaround unwanted SEND execution due to broken NoMask control flow.
- intel/fs/gen12: Fixup/simplify SWSB annotations of SIMD32 scratch writes.
- intel/fs/gen12: Workaround data coherency issues due to broken NoMask control flow.
- intel/fs: Set src0 alpha present bit in header when provided in message payload.
- intel/fs/gen11: Work around dual-source blending hangs in combination with SIMD32.
- intel/fs: Make sample_mask_reg() local to brw_fs.cpp and use it in more places.
- intel/fs: Use helper for discard sample mask flag subregister number.
- intel/fs/gen7+: Swap sample mask flag register and FIND_LIVE_CHANNEL temporary.
- intel/fs: Refactor predication on sample mask into helper function.
- intel/fs: Return consistent UW types from sample_mask_reg() in fragment shaders.

- intel/fs/gen7+: Implement discard/demote for SIMD32 programs.
- intel/compiler: Move base IR definitions into a separate header file
- intel/compiler: Reverse inclusion dependency between brw_cfg.h and brw_shader.h
- intel/compiler: Nest definition of live variables block_data structures
- intel/compiler: Reverse inclusion dependency between brw_fs_live_variables.h and brw_fs.h
- intel/compiler: Reverse inclusion dependency between brw_vec4_live_variables.h and brw_vec4.h
- intel/compiler: Introduce simple IR analysis pass framework
- intel/compiler: Introduce backend_shader method to propagate IR changes to analysis passes
- intel/compiler: Define more detailed analysis dependency classes
- intel/compiler: Pass detailed dependency classes to invalidate_analysis()
- intel/compiler: Mark virtual_grf_interferes and vars_interfere as const
- intel/compiler: Move all live interval analysis results into fs_live_variables
- intel/compiler: Move all live interval analysis results into vec4_live_variables
- intel/compiler: Restructure live intervals computation code
- intel/compiler: Pass single backend_shader argument to the fs_live_variables constructor
- intel/compiler: Pass single backend_shader argument to the vec4_live_variables constructor
- intel/compiler/fs: Add live interval validation pass
- intel/compiler/vec4: Add live interval validation pass
- intel/compiler/fs: Switch liveness analysis to IR analysis framework
- intel/compiler/vec4: Switch liveness analysis to IR analysis framework
- intel/compiler: Drop invalidate_live_intervals()
- intel/compiler: Move idom tree calculation and related logic into analysis object
- intel/compiler: Move dominance tree data structure into idom_tree object
- entel/compiler: Simplify new_idom reduction in dominance tree calculation
- intel/compiler: Move register pressure calculation into IR analysis object
- intel/compiler: Calculate num_instructions in O(1) during register pressure calculation
- intel/fs: Fix workaround for VxH indirect addressing bug under control flow.
- intel/fs/gen12: Fix interaction of SWSB dependency combination with EU fusion workaround.
- intel/fs/gen12: Fix hangs with per-sample SIMD32 fragment shader dispatch.
- intel/fs/gen12: Work around dual-source blending hangs in combination with SIMD32.
- intel/fs/gen12: Fix Render Target Read header setup for new thread payload layout.
- intel/ir: Add missing initialization of backend_reg::offset during construction.
- intel/fs: Rename half() helpers to quarter(), allow index up to 3.
- intel/fs: Fix constness of argument of fs_instruction_scheduler::is_compressed().
- intel/fs: Replace fs_visitor::bank_conflict_cycles() with stand-alone function.
- intel/vec4: Fix constness of vec4_instruction::reads_flag() and ::writes_flag().

- intel/ir: Import shader performance analysis pass.
- intel/fs: Heap-allocate fs_visitors in brw_compile_fs().
- intel/fs: Implement performance analysis-based SIMD32 heuristic for fragment shaders.
- intel/fs: Add INTEL_DEBUG=no32 debugging flag.
- intel/ir: Use brw::performance object instead of CFG cycle counts for codegen stats.
- intel/ir: Pass block cycle count information explicitly to disassembler.
- intel/ir: Remove scheduling-based cycle count estimates.
- intel/ir: Update performance analysis parameters for memory fence codegen changes.

Fritz Koenig (3):

- Revert "gitlab-ci: disable a630 tests as mesa-cheza is down"
- Revert "gitlab-ci: disable a630 tests as mesa-cheza is down (again)"
- freedreno: allow FMT6_8_UNORM as a UBWC format

Georg Lehmann (3):

- Correctly wait in the fragment stage until all semaphores are signaled
- Vulkan Overlay: Don't try to change the image layout to present twice
- Vulkan overlay: use the corresponding image index for each swapchain

Gert Wollny (63):

- r600: force new CF with TEX only if any texture value is written
- r600: Increase space for IO values to agree with PIPE_MAX_SHADER_IN/OUTPUTS
- r600: Add NIR compiler options
- r600: Update state code to accept NIR shaders
- r600/sfn: Add a basic nir shader backend
- r600: enable NIR backend DEBUG flag for supported architectures
- r600/sfn: Add the VS in and FS out vectorization
- r600/sfn: Add the WaitAck instruction
- r600/sfn: add live range evaluation for the GPR
- r600/sfn: add register remapping
- r600/sfn: Add lowering arrays to scratch and according instructions
- r600/sfn: Add a load GDS result instruction
- r600/sfn: Add MemRingOut instructions
- r600/sfn: add emitVertex instructions
- r600/sfn: Add support for geometry shader
- r600/sfn: Add VS for TCS shader skeleton
- r600/sfn: Add compute shader skeleton
- r600/sfn: Add GDS instructions
- r600/sfn: Add lowering UBO access to r600 specific codes

- r600: Make sure LLVM is not used for DRAW
- r600/sfn: Add support for atomic instructions
- r600/sfn: Add support for SSBO load and store
- r600/sfn: Add .editorconfig file
- r600/sfn: Add some documentation
- r600/sfn: Avoid using dynamic_cast to identify type
- r600/sfn: Use static_cast when type is already known
- r600/sfn: Don't try to catch exceptions, the driver doesn't throw any
- gallium/tgsi_to_nir: Set nir_intrinsic_align_mul to 16 and offset to 0
- r600: Dump a few more variables when requested
- r600/sfn: Reduce array limit for scratch usage
- r600/sfn: Fix setting alignments when lowering UBOs
- r600/sfn: Implementing instructions blocks
- r600/nir: Pin interpolation results to channel
- r600/sfn: Fix null pointer deref in live range evalation
- r600/sfn: Handle b2b1 like it was a mov
- r600/sfn: Fix handling of GS inputs
- r600/sfn: Fix using the result of a fetch instruction in next fetch
- r600/sfn: Count only literals that are not inline to split instruction groups
- r600/sfn: use new temp register allocation when loading single value temporaries
- nir: Add r600 specific intrinsics for tesselation shader IO
- nir: Add umad24 and umul24 opcodes
- r600: Handle texcoord semantics in LDS index evaluation
- r600/sfn: simplify UBO lowering pass
- r600/sfn: Don't emit inline constants in the r600 IR
- r600/sfn: Add LDS IO instructions to r600 IR
- r600/sfn: Add LDS instruction to assembly conversion
- r600/sfn: Add TF write instruction
- r600/sfn: Add IR instruction to fetch the TESS parameters
- r600/sfn: Handle umul24 and umad24
- r600/sfn: Emit some LDS instructions
- r600/sfn: Move emission of barrier from compute shader to shader base
- r600/sfn: Add methods to valuepool to get a vector of values
- r600/sfn: Move some shader base methods to the public interface
- r600/sfn: extract class to handle the VS export to different stages
- r600/sfn: derive the GS from the vertex stage for a common interface

- r600/sfn: Handle LDS output in VS
- r600/sfn: Move removing of unused variables
- r600/sfn: Add lowering passes for Tesselation IO
- r600/sfn: Add tesselation shaders
- r600: Enable tesselation for NIR
- r600: Fix nir compiler options, i.e. don't lower IO to temps for TESS
- r600/sfn: Fix printing vertex fetch instruction flags
- r600: Fix duplicated subexpression in r600_asm.c

Greg V (3):

- amd/addrlib: fix build on non-x86 platforms
- r600: add missing <array> include
- svga: fix build on FreeBSD

H.J. Lu (2):

- x86_init_func_common: Add ENDBR at function entry
- x86: Add ENDBR at function entries

Hanno Böck (1):

- Properly check mmap return value

Hyunjun Ko (27):

- freedreno/ir3: fix printing half constant registers.
- freedreno/ir3: Add cat4 mediump opcodes
- freedreno/ir3: put the conversion back for half const to the right place.
- freedreno/ir3: Fold const only when the type is float
- freedreno/ir3: Add new ir3 pass to fold out fp16 conversions
- nir: Add optimization for doing removing f16/f32 conversions
- freedreno/ir3: handle half registers for arrays during register allocation.
- turnip: support indirect draw
- glsl: Handle fp16 unary operations when lowering matrix operations
- glsl/lower_instructions: Handle fp16 for MOD_TO_FLOOR
- turnip: Gather information for transform feedback
- turnip: Define structs for transform feedback
- turnip: Setup stream-output when linking program
- turnip: Implement stream-out emit and vkApis for transform feedback
- turnip: Implement an empty function vkCmdDrawIndirectByteCountEXT
- turnip: Enable VK_EXT_transform_feedback
- turnip: Add tu6_control struct.
- turnip: Fix wrong assignment of xfb output's offset.

- turnip: Do gathering xfb info after nir_remove_dead_variables
- freedreno: Enable mediump lowering
- freedreno/ir3: enable nir_opt_loop_unroll on a6xx
- nir: fix wrong assignment to buffer in xfb_varyings_info
- turnip: make the struct slot_value of queries get 2 values
- turnip: Implement and enable VK_QUERY_TYPE_TRANSFORM_FEEDBACK_STREAM_EXT
- turnip : Fix wrong offset calculation for xfb buffer.
- turnip: Skip unused regs when setting up streamout buffers
- turnip: Fix crashes when geometry shader constants aren't used

Iago Toral Quiroga (1):

- nir: add a bool bitsize lowering pass

Ian Romanick (62):

- intel/fs: Don't count integer instructions as being possibly coissue
- nir: Mark fmin and fmax as commutative and associative
- mesa/draw: Make sure all the unused fields are initialized to zero
- nir/search: Use larger type to hold linearized index
- intel/fs: Correctly handle multiply of fsign with a source modifier
- intel/fs: Do cmod prop again after scheduling
- intel/fs: Allow NOT instructions in conditional discard optimization
- intel/fs: Fix NULL destinations on 3-source instructions again after late DCE
- nir/algebraic: Simplify logic to detect sign of an integer
- nir/algebraic: optimize ior(ine(a, 0), ine(b, 0)) to ine(ior(a, b), 0)
- nir/algebraic: Generalize some and-of-shift-right patterns [v2]
- nir/algebraic: Constant reassociation for bitwise operations too
- nir/algebraic: Simplify a contradiction that can occur in __flt64_nonnan
- soft-fp64/b2f: Reimplement using bitwise logic ops
- soft-fp64: Don't open-code umulExtended
- soft-fp64: Simplify __countLeadingZeros32 function
- soft-fp64: Pick a single idiom for treating sign value as a Boolean
- soft-fp64: Store sign value as 0 or 0x80000000
- soft-fp64/fneg: Don't treat NaN specially
- soft-fp64/flt: Perform checks in a different order
- soft-fp64/fsat: Correctly handle NaN
- soft-fp64/fsat: Micro-optimize x < 0 test
- soft-fp64/fsat: Micro-optimize x >= 1 test
- soft-fp64: Relax the way NaN is propagated

- soft-fp64/ffloor: Simplify the >= 0 comparison
- soft-fp64: Optimize __fmin64 and __fmax64 by using different evaluation order [v2]
- soft-fp64/fadd: Instead of tracking "b < a", track sign of the difference
- soft-fp64/fadd: Massively split the live range of zFrac0 and zFrac1
- soft-fp64/fadd: Pick zero or non-zero result based on subtraction result
- soft-fp64/fadd: Just let the subtraction happen when the result will be zero
- soft-fp64/fadd: Delete a redundant condition check
- soft-fp64/fadd: Reformat after previous commit
- soft-fp64/fadd: Combine an if-statement into the preceeding else-clause
- soft-fp64/fadd: Rename aFrac and bFrac variables
- soft-fp64/fadd: Use absolute value of expDiff
- soft-fp64/fadd: Move common code out of both branches of an if-statement
- soft-fp64/fadd: Common code optimization for differing sign case
- soft-fp64: Split a block that was missing a cast on a comparison
- intel/vec4: Allow late copy propagation on vec4
- nir/algebraic: Change the default cursor location when replacing a unary op
- nir/algebraic: Distribute source modifiers into instructions
- nir/algebraic: Use value range analysis to convert fmax to fsat
- nir/algebraic: Remove a redundant fabs pattern
- tnl: Don't dereference NULL obj pointer in bind_indices
- tnl: Don't dereference NULL obj pointer in replay_init
- tnl: Don't dereference NULL obj pointer in t_rebase_prims
- tnl: Silence unused parameter 'attrib' warning in convert_half_to_float
- tnl: Silence unused parameter warnings in _tnl_draw_prims
- tnl: Silence unused parameter warnings in dump_draw_info
- tnl: Silence unused parameter warnings in _tnl_split_inplace
- tnl: Code formatting in t_draw.c
- tnl: Code formatting in t_rebase.c
- intel/compiler: Silence unused parameter warnings in vec4_tcs_visitor
- intel/compiler: Silence unused parameter warning in fs_live_variables::setup_one_read
- intel/compiler: Silence unused parameter warning in update_inst_scoreboard
- intel/compiler: Only GE and L modifiers are commutative for SEL
- intel/compiler: CSEL can do saturate
- intel/compiler: Fixup operands in fs_builder::emit() that takes array
- nir/algebraic: Detect some kinds of malformed variable names
- nir/algebraic: Require operands to iand be 32-bit

- nir/algebraic: Optimize ushr of pack_half, not ishr
- anv/tests: Don't rely on assert or changing NDEBUG in tests

Icecream95 (16):

- panfrost: Fix non-debug builds
- panfrost: Inline panfrost_get_default_swizzle
- panfrost: LogicOp support
- nir: Allow nir_format conversions to work on 32-bit values
- panfrost: LogicOp fixes and non 8-bit format support
- mesa/format_utils: Add a fast-path for RGBA to BGRA
- panfrost: Extend the tiled store fast-path to loads
- panfrost: Mark 64-bit formats as unsupported
- panfrost: Add support for B5G5R5X1
- st/mesa: Fall back on R3G3B2 for R3_G3_B2
- panfrost: Add support for R3G3B2
- panfrost: Correctly identify format 0x4c
- pan/midgard: Fix a divide by zero in emit_alu_bundle
- panfrost: Fix GL_EXT_vertex_array_bgra
- panfrost: Enable PIPE_CAP_VERTEX_COLOR_UNCLAMPED
- panfrost: Fix background showing when using discard

Icenowy Zheng (3):

- lima: remove its hash table entry when invalidating a resource
- lima: expose fragment shader derivatives capability
- lima: implement zsbuf reload

Ilia Mirkin (24):

- nv50: report max lod bias of 15.0
- gitlab-ci: disable panfrost runners
- mesa: fix _mesa_draw_nonzero_divisor_bits to return nonzero divisors
- nv50,nvc0: add newly added PIPE_CAP's to list
- st/mesa: allow TXB2/TXL2 to work with cube array shadow textures
- nvc0: enable EXT_texture_shadow_lod
- st/vdpau: avoid asserting on new VDP_YCBCR_* formats
- st/vdpau: make query test for 2D support
- nv50: don't try to upload MSAA settings for BUFFER textures
- gallium: add viewport swizzling state and cap
- mesa: add GL_NV_viewport_swizzle support
- st/mesa: add NV_viewport_swizzle support

- nvc0: add NV_viewport_swizzle support for GM200+
- compiler: add VARYING_SLOT_VIEWPORT_MASK
- glsl: add NV_viewport_array2 support
- mesa: add NV_viewport_array2 enable, attach to glsl
- gallium: add TGSI_SEMANTIC_VIEWPORT_MASK
- gallium: add TGSI_PROPERTY_LAYER_VIEWPORT_RELATIVE
- gallium: add PIPE_CAP_VIEWPORT_MASK
- st/mesa: add support for GL_NV_viewport_array2
- nvc0: enable GL_NV_viewport_array2
- nv50,nvc0: update with latest caps
- docs: update for recently-added nvc0 features
- mesa: add interaction between compute derivatives and variable local sizes

Indrajit Kumar Das (4):

- glapi/copyimage: Implement CopyImageSubDataNV
- gallium: prepare framework for supporting AlphaToCoverageDitherControlNV
- mesa: add support for AlphaToCoverageDitherControlNV
- radeonsi: enable support for AlphaToCoverageDitherControlNV

Ivan Molodetskikh (1):

- egl: allow INVALID format for linux_dmabuf

James Xiong (2):

- iris: handle the failure of converting unsupported yuv formats to isl
- gallium: let the pipe drivers decide the supported modifiers

James Zhu (1):

- radeonsi: fix Segmentation fault during vaapi enc test

Jan Palus (1):

- targets/opencl: fix build against LLVM>=10 with Polly support

Jan Vesely (2):

- clover: Use explicit conversion from llvm::StringRef to std::string
- clover: Check if the detected clang libraries are usable

Jan Zielinski (8):

- gallium/swr: Fix various asserts and security issues
- gallium/swr: fix corruptions in Unigine Heaven
- gallium/swr: use ElementCount type arguments for getSplat()
- gallium/gallivm: Remove workaround disabling AVX code for newer CPUs
- gallium/gallivm: fix compilation issues with llvm 11
- gallium/gallivm: remove unused header include for newer LLVM

- gallium/swr: Fix LLVM 11 compilation issues
- gallium/swr: Fix crashes and failures in vertex fetch

Jason Ekstrand (202):

- genxml: Add a new 3DSTATE_SF field on gen12
- anv,iris: Set 3DSTATE_SF::DerefBlockSize to per-poly on Gen12+
- intel/genxml: Drop SLMEnable from L3CNTLREG on Gen11
- iris: Set SLMEnable based on the L3$ config
- iris: Store the L3$ configs in the screen
- iris: Use the URB size from the L3$ config
- i965: Re-emit l3 state before BLORP executes
- intel: Take a gen_l3_config in gen_get_urb_config
- intel/blorp: Always emit URB config on Gen7+
- iris: Consolodate URB emit
- anv: Emit URB setup earlier
- intel/common: Return the block size from get_urb_config
- intel/blorp: Plumb deref block size through to 3DSTATE_SF
- anv: Plumb deref block size through to 3DSTATE_SF
- iris: Plumb deref block size through to 3DSTATE_SF
- anv: Always fill out the AUX table even if CCS is disabled
- intel/eu/validate: Don't validate regions of sends
- intel/disasm: SEND has two sources on Gen12+
- intel/tools: Handle strides better when dumping buffers
- intel/fs: Write the address register with NoMask for MOV_INDIRECT
- anv/blorp: Use the correct size for vkCmdCopyBufferToImage
- anv: No-op submit and wait calls when no_hw is set
- anv: Reject modifiers on depth/stencil formats
- vulkan: Update the XML and headers to 1.2.133
- nir: Fix the nir_builder include path for nir_builtin_builder
- nir/builder: Return an integer from nir_get_texture_size
- intel/isl: Add isl_aux_info.c to Makefile.sources
- anv: Always enable the data cache
- nir: Drop nir_tex_instr::texture_array_size
- anv: Use the PIPE_CONTROL instead of bits for the CS stall W/A
- anv: Use a proper end-of-pipe sync instead of just CS stall
- anv: Do end-of-pipe sync around MCS/CCS ops instead of CS stall
- nir: Flush to zero with OOB low exponents in ldexp

- isl: Set 3DSTATE_DEPTH_BUFFER::Depth correctly for 3D surfaces
- iris: Allow HiZ on blit sources
- blorp: Write to depth/stencil images as depth/stencil when possible
- anv: Enable HiZ for VK_IMAGE_LAYOUT_TRANSFER_DST_OPTIMAL
- iris: Enable CCS for copies from HiZ+CCS depth buffers
- iris: Enable HiZ and stencil CCS for blorp blit destinations
- iris: Don't skip fast depth clears if the color changed
- anv: Parse VkPhysicalDeviceFeatures2 in CreateDevice
- anv: Mark max_push_range UNUSED and simplify the code
- anv: Pass buffer addresses into emit_push_constant*
- anv: Delete some pointless break statements
- anv: Align UBO sizes to 32B
- anv: Add an align_down_u32 helper
- anv: Bounds-check pushed UBOs when robustBufferAccess = true
- vulkan/wsi: Don't leak the FD when GetImageDrmFormatModifierProperties fails
- vulkan/wsi: Return an error if dup() fails
- intel/isl: Clean up some aux surface logic
- intel/isl: Add a separate ISL_AUX_USAGE_HIZ_CCS_WT
- intel/blorp: Allow HIZ_CCS_WT in copy sources
- iris: Use ISL_AUX_USAGE_HIZ_CCS_WT to indicate write-through HiZ
- intel/isl: Require ISL_AUX_USAGE_HIZ_CCS_WT for HZ+CCS WT mode
- intel/isl: Add a separate ISL_AUX_USAGE_STC_CCS
- intel/blorp: Allow STC_CCS in blit sources
- iris: Use ISL_AUX_USAGE_STC_CCS for stencil CCS
- intel: Require ISL_AUX_USAGE_STC_CCS for stencil CCS
- intel/isl: Set DepthStencilResource based on aux usage
- anv: Dump push ranges via VK_KHR_pipeline_executable_properties
- anv: Fix the comparison in an assert
- anv: Push UBO ranges relative to the start of the binding
- anv: Do an end-of-pipe sync before updating AUX table entries
- intel/isl: Don't align linear images to 64K on Gen12+
- intel/blorp: Add support for swizzling fast-clear colors
- anv: Swizzle fast-clear values
- intel/iris: Always initialize CCS to 0
- anv: Only add END_OF_PIPE_SYNC if we actually have AUX_INVAL
- util/sparse_array: Finish the sparse_array in the tests

- util/sparse_array: Add a node_size_log2 temporary
- meson,ci: Disable sparse_array tests on windows
- util/sparse_array: Stash the node level in the node pointer
- anv: Stop fetching the timestamp frequency ourselves
- intel/dump_gpu: Add an ensure_device_info helper
- intel/dump_gpu: Handle a bunch of getparam in the no-HW case
- intel/nir: Run copy-prop and DCE after lower_bool_to_int32
- nir: Add b2b opcodes
- aco: Implement b2b32 and b2b1
- nir: Use b2b opcodes for shared and constant memory
- nir: Insert b2b1s around booleans in nir_lower_to
- anv: Set alignments on descriptor and constant loads
- nir: Validate that memory load/store ops work on whole bytes
- nir: Set UBO alignments in lower_uniforms_to_ubo
- nir/opt_loop_unroll: Fix has_nested_loop handling
- nir/lower_int64: Lower 8 and 16-bit downcasts with nir_lower_mov64
- nir/algebraic: Add downcast-of-pack opts
- nir: Add a nir_op_is_vec helper
- nir: Copy propagate through vec8s and vec16s
- nir: Handle vec8/16 in bool_to_bitsize
- nir: Handle vec8/16 in gather_ssa_types
- nir: Handle vec8/16 in lower_phis_to_scalar
- nir: Handle vec8/16 in lower_regs_to_ssa
- nir: Handle vec8/16 in opt_split_alu_of_phi
- nir: Treat vec8/16 as select in opt_peephole_select
- nir: Handle vec8/16 in opt_undef_vecN
- nir: Handle vec8/16 in nir_shrink_array_vars
- anv: Account for the header in anv_state_stream_alloc
- anv/allocator: Use util_dynarray for blocks in anv_state_stream
- spirv: Implement OpCopyObject and OpCopyLogical as blind copies
- Revert “spirv: Implement OpCopyObject and OpCopyLogical as blind copies”
- anv/image: Use align_u64 for image offsets
- nir/from_ssa: Only chain movs when a src is also a dest
- intel/fs: Choose memory message type based on bit size
- anv: Improve brw_nir_lower_mem_access_bit_sizes
- iris: Set alignments on cbuf0 and constant reads

- intel/nir: Lower memory access bit sizes later
- nir/load_store_vectorize: Fix shared atomic info
- nir/load_store_vectorize: Use nir_iadd_imm for offsets
- nir/load_store_vectorize: Add support for nir_var_mem_global
- intel/nir: Enable load/store vectorization
- spirv: Add a vtn_block() helper
- spirv: Add cast and loop helpers for vtn_cf_node
- spirv: Make vtn_case a vtn_cf_node
- spirv: Make vtn_function a vtn_cf_node
- spirv: Add a parent field to vtn_cf_node
- spirv: Rewrite CFG construction
- Revert "spirv: Rewrite CFG construction"
- nir: Assert memory loads are aligned
- anv: Advertise SEND count through VK_EXT_pipeline_executable_properties
- anv: Fix UBO range detection in anv_nir_compute_push_layout
- nir: Add an alignment to nir_intrinsic_load_constant
- nir: Add some sanity assertions in opt_large_constants
- intel: Add _const versions of prog_data cast helpers
- anv: Report correct SLM size
- intel/batch_decoder: Stop printing to stdout
- intel/cfg: Add first/last_block helpers
- anv: Emit pushed UBO bounds checking code in the back-end compiler
- intel/blorp: Delete an unused enum
- spirv: Handle OOB vector extract operations
- spirv,nir: Add a better vector_insert
- spirv: Error if OpCompositeInsert/Extract has OOB indices
- nir/builder: Handle any bit-size selector in nir_extract
- spirv: Call nir_builder directly for vector_extract
- spirv,nir: Move the SPIR-V vector insert code to NIR
- anv: Move vb_emit setup closer to where it's used in flush_state
- anv: Apply any needed PIPE_CONTROLs before emitting state
- nir/dominance: Better handle unreachable blocks
- nir/gcm: Loop over blocks in pin_instructions
- nir/gcm: Use an array for storing the early block
- nir/gcm: Move block choosing into a helper function
- nir/gcm: Add a real concept of "progress"

- nir/gcm: Delete dead instructions
- nir/gcm: Prefer the instruction's original block
- intel/fs: Rename block to scan_block in can_coalesce_vars
- intel/fs: Coalesce when the src live range is contained in the dst
- glsl: Hard-code noise to zero in builtin_functions.cpp
- nir: Delete the fnoise opcodes
- meta,i965: Rip GL_EXT_texture_multisample_blit_scaled support out of meta
- spirv: Allow constants and NULLs in SpvOpConvertUToPtr
- anv: Properly handle all sizes of specialization constants
- radv: Properly handle all sizes of specialization constants
- turnip: Properly handle all sizes of specialization constants
- spirv: Use nir_const_value for spec constants
- nir/opt_deref: Remove certain sampler type casts
- spirv: Fix passing combined image/samplers through function calls
- anv: Drop an assert
- nir/lower_subgroups: Mask off unused bits in ballot ops
- anv: Add a vk_image_layout_to_usage_flags helper
- anv: Move vk_image_layout_is_read_only higher
- anv: Be more conservative about image view usage
- anv: Rework anv_layout_to_aux_state
- anv/blorp: Do less hard-coding of aux usages
- anv: Generalize some aux usage checks
- intel/blorp: Allow more HiZ usages in hiz_clear_depth_stencil
- anv: Simplify a case in layout_to_aux_usage
- anv/cmd_buffer: Move anv_image_init_aux_tt higher
- intel/isl: Delete a misleading comment
- intel/isl: Refactor isl_surf_get_ccs_surf
- anv: Add support for HiZ+CCS
- spirv: Rewrite CFG construction
- intel/devinfo: Compute the correct L3$ size for Gen12
- anv: Expose CS workgroup sizes based on a maximum of 64 threads
- anv: Return an error if allocating attachment memory fails
- anv: Add TRANSFER_SRC to pass usage not subpass usage
- anv: Stop filling out the clear color in compute_aux_usage
- anv: Assert surface states are valid
- anv: Use ANV_FROM_HANDLE for pInheritanceInfo fields

- anv: Mark images written in end_subpass
- anv: Split command buffer attachment setup in three
- anv: Allocate surface states per-subpass
- intel: Move swizzle_color_value from blorp to ISL
- anv: Disallow fast-clears which require format-reinterpretation
- anv: Stop allowing non-zero clear colors in input attachments
- anv: Refactor cmd_buffer_setup_attachments
- anv: Rework depth_stencil_attachment_compute_aux_usage
- anv: Split color_attachment_compute_aux_usage in two
- anv: Use anv_layout_to_aux_usage for color during render passes
- anv: Allow all clear colors for texturing on Gen11+
- vulkan: Update Vulkan XML and headers to 1.2.139
- nir/copy_prop_vars: Handle volatile better
- nir/copy_prop_vars: Report progress when deleting self-copies
- nir/dead_write_vars: Handle volatile
- nir/combine_stores: Handle volatile
- anv: Handle NULL descriptors
- anv: Handle null vertex buffer bindings
- anv: Claim VK_EXT_robustness2 support
- intel/fs: Don't delete coalesced MOVs if they have a cmod
- vulkan: Allow destroying NULL debug report callbacks
- anv:gpu_memcpy: Emit 3DSTATE_VF_INDEXING on Gen8+
- nir/lower_double_ops: Rework the if (progress) tree
- nir/opt_deref: Report progress if we remove a deref
- nir/copy_prop_vars: Record progress in more places

Jesse Natalie (3):

- wgl: add official gldrv.h header-file
- wgl: use gldrv.h instead of stw_icd.h
- util/ralloc: fix ralloc alignment on Win64

John Stultz (7):

- freedreno: Add ir3_cf.c and ir3_delay.c to Makefile.sources
- panfrost: Move pan_afbc.c file to the the right Makefile.source file
- gallium: hud_context: Fix scalar initializer warning.
- Android.mk: Tweak MESA_ENABLE_LLVM checks
- etnaviv: Avoid shift overflow
- vc4_bufmgr: Remove duplicative VC definition

- r600: Fix build error in sfn_nir_lower_fs_out_to_vector.cpp

Jon Turney (1):

- Fix util/process test on Cygwin

Jonathan Marek (79):

- freedreno/a6xx: use single format enum
- freedreno/a6xx: fix Z24_UNORM_S8_UINT_AS_R8G8B8A8
- freedreno: name sysmem color/depth flush events
- freedreno/a6xx: document some unknown bits
- turnip: add option to force use of hw binning
- turnip: fix COND_EXEC reserved size in tu_query
- turnip: add tu_device pointer to tu_cs
- turnip: automatically reserve cmdstream space in emit_pkt4/emit_pkt7
- turnip: remove marker seqno
- turnip: make cond_exec helper easier to use
- turnip: move tile_load_ib/sysmem_clear_ib into draw_cs
- hud: add GALLIUM_HUD_SCALE
- turnip: enable sampleRateShading feature
- turnip: enable fullDrawIndexUint32/independentBlend/dualSrcBlend/logicOp
- etnaviv: disable INT_FILTER for ASTC
- util/format: add missing BC4/BC5 vulkan formats
- turnip: rework format table to support r5g5b5a1_unorm/b5g5r5a1_unorm
- turnip: add r5g5b5a1_unorm/b5g5r5a1_unorm formats
- turnip: check the right alignment requirement on shader iova
- turnip: move some constant state to tu6_init_hw
- turnip: remove unecessary MRT_CONTROL fill
- turnip: minify image_view extent
- turnip: fix hw binning + render_area offset interaction
- turnip: fix srgb MRT
- turnip: don't hardcode gmem base for input attachment
- turnip: remove unnecessary fb size check
- turnip: fall back to sysmem when attachments don't fit into gmem
- turnip: increase array sizes in tu_descriptor_map
- turnip: improve binning pipe layout config
- turnip: fix tile->slot calculation
- etnaviv: nir: add compile_check_limits
- freedreno/registers: more GRAS_CL_CNTL bits, Z_CLAMP

- turnip: fix znear clipping
- turnip: implement depth clamp
- turnip: implement timestamp query
- turnip: fix compute shaders crashing after geometry shader change
- turnip: improve vertex input handling
- turnip: use buffer size instead of bo size for VFD_FETCH_SIZE
- freedreno/registers: add RB_CCU_CNTL bitfields
- freedreno/a6xx: set bypass RB_CCU_CNTL value for blitter
- turnip: RB_CCU_CNTL fixes
- turnip: split up gmem/tile alignment
- turnip: fix nir validate failure from push constant lowering
- turnip: disable 8x msaa
- turnip: save attachment samples in renderpass state
- turnip: use dirty bits for dynamic viewport/scissor state
- turnip: rework format helpers
- turnip: add vk_format_is_snorm/is_float
- turnip: new clear/blit implementation with shader path fallback
- freedreno/computerator: support nop prefix
- freedreno/computerator: support bindless sampler instructions
- freedreno/ir3: fix emit_tex_info split_dest
- freedreno/ir3: don't overwrite wrmask in ir3_SAM
- turnip: compute render_components/srgb_cntl at renderpass creation time
- turnip: don't limit framebuffer size to image size
- turnip: image_view rework
- nir: add common convert_ycbcr for vulkan csc
- nir: convert_ycbcr: preserve alpha channel
- anv: use common nir_convert_ycbcr
- radv: use common nir_convert_ycbcr
- turnip: fix GMEM resolve in CmdNextSubpass
- turnip: disable depth test for S8_UINT attachment
- turnip: improve GMEM load/store logic
- turnip: enable VK_FORMAT_S8_UINT as stencil format
- turnip: set shader key msaa field
- turnip: implement VK_EXT_sample_locations
- turnip: implement VK_EXT_filter_cubic
- turnip: enable cube arrays

- turnip: implement VK_EXT_sampler_filter_minmax
- turnip: divide cube map depth by 6
- freedreno/ir3: fix 16-bit ssbo access
- freedreno/ir3: set even bit for f2f16_rtne
- freedreno/ir3: fix incorrect conversion folding
- turnip: remove unused RB_UNKNOWN_8E04_blit
- turnip: use RESOLVE_TS event
- turnip: add adreno 650
- nir: add pack_32_2x16_split/unpack_32_2x16_split lowering
- freedreno/ir3: run nir_lower_pack
- turnip: fix wrong substream size in parse_multisample_and_color_blend

Jordan Justen (6):

- intel/compiler: Restrict cs_threads to 64
- intel: Update TGL PCI strings
- intel: Add TGL PCI ID
- intel/dev: Split .num_subslices out of GEN12_FEATURES macro
- intel/dev: Add device info for RKL
- docs/relnotes/new_features.txt: Add RKL to 20.1 release notes

Jose Maria Casanova Crespo (5):

- broadcom: Fix implicit declaration of ffs for Android build
- v3d: Sync on last CS when non-compute stage uses resource written by CS
- v3d: Primitive Counts Feedback needs an extra 32-bit padding.
- v3d: Fix swizzle in DXT3 and DXT5 formats
- v3d: Include supported DXT formats to enable s3tc/dxt extensions

Joshua Ashton (3):

- radv: Use TRUNC_COORD on samplers
- radv: Pass logical device to si_emit_graphics
- radeonsi: Use TRUNC_COORD on samplers

José Fonseca (4):

- meson: Avoid duplicate symbols.
- scons: Prune out unnecessary targets.
- gitlab-ci: Prune all SCons jobs except scons-win64, and allows failures.
- appveyor: Remove Meson job.

Juan A. Suarez Romero (6):

- nir/lower_double_ops: add note for lowering mod
- nir/lower_double_ops: relax lower mod()

- nir/algebraic: coalesce fmod lowering
- anv: use urb_setup_attribs in SBE
- intel/compiler: store the FS inputs in WM prog data
- anv/pipeline: allow more than 16 FS inputs

Karol Herbst (18):

- clover: add trivial clCreateCommandQueueWithProperties implementation
- nir/lower_ssbo: handle atomics
- gallium: make handles of set_global_binding 64 bit
- Revert "gallium: make handles of set_global_binding 64 bit"
- nv50, nvc0: fix must_check warning of util_dynarray_resize_bytes
- clover: fix build with single library clang build
- gallium: add PIPE_CAP_SYSTEM_SVM
- clover: add stubs for SVM
- clover: implement CL_DEVICE_SVM_CAPABILITIES
- clover: implement clSetKernelArgSVMPointer
- clover: implement SVM functions for devices with fine grained system SVM support
- clover: implement cl_arm_shared_virtual_memory
- clover: expose cl_arm_shared_virtual_memory for devices with SVM support
- nvc0: enable ASTC and ETC on GM20B
- mesa: fix enum value of VIEWPORT_SWIZZLE_POSITIVE_W_NV
- gallium: initialize viewport swizzle in cso_set_viewport_dims
- Revert "nvc0: fix line width on GM20x+"
- st/mesa: properly guard fallback_copy_texsubimage aginst failed maps

Kenneth Graunke (14):

- intel/genxml: Drop "reserved" enum
- isl: Fix the android build.
- iris: Dump frame markers with INTEL_DEBUG=submit
- iris: Trim "../../src/gallium/drivers/iris/" out of debug dump filenames
- iris: Make mocs an inline helper in iris_resource.h
- iris: Fix BLORP vertex buffers to respect ISL MOCS settings
- iris: Set MOCS for constant packets on Gen12+
- intel/compiler: Drop nir_lower_to_source_mods() and related handling.
- intel/compiler: Put back saturate on [iu]add_sat opcodes
- intel/compiler: Don't copy prop source mods into PICK_HIGH_32BIT
- intel/compiler: Delete abs/neg handling in fsign code
- intel/compiler: Don't create 64-bit src1 immediates in opt_peephole_sel

- nir: Actually do load/store vectorization beyond vec2
- iris: Fix downcast of bound_vertex_buffers from uint64_t to int

Konrad Dybcio (1):

- freedreno/a4xx: enable A405

Kristian Høgsberg (39):

- nir: Delete unused is_var_constant() helper
- nir: Make unroll pragma work on clang
- freedreno/fdperf: Cast away some ignored return values
- spirv/opencl: Cast opcode up front to avoid warnings
- glsl: Use 'using' to be explicit about visitor overloads
- nir: Remove always-true assert
- turnip: Be explicit about converting vk compare func to a6xx
- freedreno/a6xx: Add fd6_resource_screen_init()
- freedreno: Set up supported modifiers in fd*_resource_screen_init()
- freedreno: Add layout_resource_for_modifier screen vfunc
- freedreno/a6xx: Implement layout for DRM_FORMAT_MOD_QCOM_COMPRESSED
- turnip: Drop explicit configure opt-in for turnip
- ci: Drop turnip opt-in option
- freedreno/ir3: Set IR3_REG_HALF flag on src as well in immediate MOV
- Mark a few static inline helpers with ASSERTED
- main/get: Converted type conversion macros to inline functions
- nir/types: Add glsl_float16_type() helper
- freedreno/ir3: Lower output precision
- Revert "glsl: Use a simpler formula for tanh"
- Revert "spirv: Use a simpler and more correct implementaiton of tanh()"
- freedreno/ir3: Don't fold conversions into sign
- glsl: Add ir_constant constructor for fp16
- glsl: Add fp16 case for ir_triop_lrp optimization
- glsl: Implement constant propagation for fp16
- glsl: Expand fp16 to float before constant expression evaluation
- glsl: Add type queries for fp16+float and fp16+float+double
- glsl/lower_instructions: Handle fp16 for FDIV_TO_MUL_RCP
- radeonsi: Stop exposing PIPE_SHADER_CAP_FP16
- turnip: Add missing VKAPI_ATTR annotations
- turnip: Stub out VK_KHR_external_{fence,semaphore}_fd
- turnip: Make Android platform build

- turnip: Drop dep_llvm from dependencies
- freedreno/ir3: Fix sz vs class confusion
- freedreno/computerator: Decouple ir3 assembler
- freedreno/ir3: Move ir3 assembler to backend compiler
- freedreno/ir3: Parse, but ignore @in, @out and @tex headers
- freedreno/ir3: Reset lex line number when we start parsing
- freedreno/ir3: Print @tex write mask using 0x%x
- freedreno: Use the right amount of &'s

Krzysztof Raszkowski (10):

- gallium/swr: fix gcc warnings
- gallium/swr: Fix gcc 4.8.5 compile error
- gallium/swr: Fix llvm11 compilation issues
- gallium/swr: simplify environmental variabled expansion code
- gallium/swr: fix rdtsc debug statistics mechanism
- gallium/swr: Fix min/max range index draw
- Revert "gallium/swr: Fix min/max range index draw"
- gallium/swr: Fix vcvtph2ps llvm intrinsic compile error
- gallium/swr: Fix array stride problem.
- gallium/swr: Re-enable scratch space for client-memory buffers

Leandro Ribeiro (1):

- i965: remove duplicated comment

Leo Liu (1):

- radeon/jpeg: fix the jpeg dt_pitch with YUYV format

Lepton Wu (1):

- virgl: Use ETC2 formats directly when possible.

Lionel Landwerlin (49):

- iris: implement gen12 post sync pipe control workaround
- anv: implement gen9 post sync pipe control workaround
- anv: implement gen12 post sync pipe control workaround
- anv: set MOCS on push constants
- mesa: add INTEL_blackhole_render
- i965: enable INTEL_blackhole_render
- st: add support for INTEL_blackhole_render
- iris: add support INTEL_blackhole_render
- intel/tools/aub_dump: move aub file initialization to maybe_init()
- intel/tools/aub_dump: fix crash when using the default legacy context

- intel/aub_dump: stub the waits when overriding the device
- intel/tools/dump_gpu: fix getparam values
- anv: stop storing prog param data into shader blobs
- intel/decoder: don't consider header fields past dword0
- isl: implement linear tiling row pitch requirement for display
- isl: properly filter supported display modifiers on Gen9+
- isl: only apply main surface ccs pitch constraint with CCS
- isl: drop min row pitch alignment when set by the driver
- intel: add new TGL pci ids
- i965/iris: fix crash when calling GetPerfQueryDataINTEL
- vulkan/overlay: Add a workaround semaphore for application presenting without one
- intel/perf: move register definition to special file
- intel/perf: break GL query stuff away
- intel/perf: move mdapi query definitions to their own file
- intel/perf: document meaning of query field
- intel/perf: store the probed i915-perf version
- isl: set bpb for Y8_UNORM
- isl: don't warn in physical extent calculation for yuv formats
- intel/aub_viewer: fix access to freed memory
- drm-shim: return device platform as specified
- drm-shim: stub libdrm's use of realpath()
- iris: properly free resources on BO allocation failure
- iris: share buffer managers accross screens
- iris: make resources take a ref on the screen object
- i965: store DRM fd on intel_screen
- i965: share buffer managers across screens
- iris: drop cache coherent cpu mapping for external BO
- intel/perf: Enable MDAPI queries for Gen12
- anv: skip writing perfcntr in results on Gen12+
- util/sparse_free_list: manipulate node pointers using atomic primitives
- iris: fail screen creation when kernel support is not there
- include/drm-uapi: bump headers
- intel/perf: store default sseu configuration
- intel/perf: specify sseu configuration when supported
- anv: force whole EU array to be powered for perf queries
- drm-shim: provide a valid fake syncobj handle at creation

- drm-shim: stub syncobj wait ioctl
- iris: don't assert on unfinished aux import in copy paths
- anv: don't expose VK_INTEL_performance_query without kernel support

Liviu Prodea (2):

- scons/windows: Support build with LLVM 10.
- util: Make process_test path compatible with mingw native toolchains

Louis-Francis Ratté-Boulianne (7):

- glsl/linker: add DisableTransformFeedbackPacking workaround
- glsl/linker: handle array/struct members for DisableXfbPacking
- glsl/linker: add xfb workaround for modified built-in variables
- gallium: add PIPE_CAP_PACKED_STREAM_OUTPUT
- gallium: add PIPE_CAP_VIEWPORT_TRANSFORM_LOWERED
- gallium: add PIPE_CAP_PSIZ_CLAMPED
- panfrost: fix transform feedback

Lucas Stach (1):

- etnaviv: retarget transfer to render resource when necessary

Marek Olšák (254):

- vbo: move GLvertexformat initialization into a template header file for reuse
- vbo: use the template for noop GLvertexformat initialization
- vbo: use the template for save GLvertexformat initialization
- vbo: move reusable code from vbo_attrib_tmp.h into vbo_util.h
- mesa: implement missing display list functions while switching to the template
- radeonsi: don't report that multi-plane formats are supported
- radeonsi: fix the DCC MSAA bug workaround
- radeonsi: don't update states for the DCC MSAA bug on GFX6-7
- glx: print FPS with 2 decimal places
- mesa: fix incorrect uses of FLUSH_CURRENT
- mesa: remove FLUSH_CURRENT calls that have no effect
- mesa: import PIPE_CAP_SIGNED_VERTEX_BUFFER_OFFSET handling
- vbo: create the immediate mode buffer only in vbo_exec_vtx_map
- vbo: skip FlushMappedBufferRange for glBegin/End by using a persistent mapping
- vbo: don't unmap persistent buffer mappings for glBegin/End
- vbo: remove immediate mode code that doesn't do anything and simplify stuff
- vbo: interleave attrsz, attrtype, and active_sz in memory
- vbo: remove a funky recursive call in glBegin
- vbo: don't check ctx->NewState twice in glBegin

- vbo: keep the immediate mode buffer always mapped for simplicity
- vbo: don't set FLUSH_UPDATE_CURRENT for glVertex
- vbo: pass only either uint32_t or uint64_t into ATTR_UNION
- vbo: don't store glVertex values temporarily into exec
- vbo: optimize resizing vertex attributes during immediate mode
- vbo: fix resizing 64-bit vertex attributes
- vbo: use FlushVertices flags properly and clear NeedFlush correctly
- vbo: increase the size of the immediate mode buffer to decrease draw count
- vbo: add/update unlikely statements in ATTR_UNION
- vbo: delay flagging FLUSH_STORED_VERTICES until glEnd
- vbo: also map the immediate mode buffer for read
- vbo: clean up resetting vertex attribs
- vbo: merge use_buffer_objects into vbo_CreateContext to skip the big malloc
- í965: don't use _mesa_prim::is_indirect
- mesa: remove unused _mesa_prim::is_indirect
- mesa: don't use bitfields in _mesa_prim
- st/mesa: optimize st_update_array with ALWAYSINLINE
- radeonsi: don't wait for shader compilation to finish when destroying a context
- mesa: translate into gallium vertex formats in mesa/main
- mesa: remove unused _mesa_draw_indirect
- st/mesa: always inline the code setting non-64bit vertex elements
- st/mesa: simplify determination whether a draw has user vertex buffers
- st/mesa: simplify determination whether a draw needs min/max index
- st/mesa: change some loops from while to do..while in st_atom_array.c
- st/mesa: make st_setup_current static
- st/mesa: simplify releasing the current attrib buffer
- gallium/u_upload_mgr: reduce dereferences by adding buffer_size
- gallium/u_upload_mgr: don't do align twice in the u_upload_alloc fast path
- gallium/u_vbuf: adjust the heuristic for unrolling indices
- gallium/cso_hash: inline a bunch of functions
- gallium/cso_hash: make cso_hash declared within structures instead of alloc'd
- gallium/cso_hash: remove always constant variable nodeSize
- gallium/cso_hash: cosmetic changes, no behavior changes
- gallium/cso_hash: remove another layer of pointer indirection
- st/mesa: try to fix MSVC build failure due to ALWAYS_INLINE
- vbo: remove dead code in vbo_can_merge_prims

- vbo: remove redundant code in vbo_exec_fixup_vertex
- mesa: document _mesa_prim::begin/end
- mesa: don't use memset in glDrawArrays
- mesa: fix immediate mode with tessellation and varying patch vertices
- gallium/util: remove unused u_surfaces.c/h
- util: remove the dependency on kcmp.h
- nir: fix gl_nir_lower_images for bindless images
- tgsi_to_nir: set num_images and num_samplers with holes correctly
- gallium/hash_table: consolidate hash tables with pointer keys
- gallium/hash_table: consolidate hash tables with FD keys
- gallium/hash_table: use the same callback signatures as util/hash_table
- gallium/hash_table: turn it into a wrapper around util/hash_table
- gallium/hash_table: remove some function wrappers
- mesa: remove leftovers from ARB_shadow_ambient
- mesa: call FLUSH_VERTICES before updating CoordReplace
- i965: stop using "indirect" parameter from Driver.Draw (non-indirect)
- mesa: remove unused "indirect" parameter from Driver.Draw
- gallium/cso_hash: pack cso_node better
- gallium/cso_hash: inline struct cso_hash_data
- gallium: pass cso_velems_state into cso_context instead of pipe_vertex_element
- gallium/u_threaded: fix uploading user indices with start != 0
- gallium/u_threaded: convert dividing by index_size to a bit shift
- mesa/i965: remove _mesa_prim::indirect_offset
- mesa: remove redundant _mesa_prim::is_indexed
- mesa: move num_instances and base_instance out of _mesa_prim
- mesa: clean up glMultiDrawElements code, use alloca for small draw count (v2)
- mesa: don't unroll glMultiDrawElements if one count is 0
- mesa: optimize glMultiDrawArrays, call Draw only once (v2)
- mesa: fix incorrect prim.begin/end for glMultiDrawElements
- nir: replace GCC unroll with an option that works on GCC < 8.0
- gallivm: fix 5 warnings
- nir: fix 5 warnings
- mesa: fix 11 warnings
- gallium/u_vbuf: silence a warning by using unreachable
- mesa: add index_size_shift = log2(index_size) into _mesa_index_buffer
- mesa: replace some index_size multiplications and divisions with shifts

- vbo: don't look at the second draw's count when merging 2 glBegin/End draws
- vbo: deduplicate copy_vertices functions
- vbo: clean up vbo_copy_vertices
- vbo: handle GS and tess primitive types when splitting Begin/End
- vbo: clean up conditional blocks in ATTR_UNION
- vbo: fold code from vbo_exec_fixup_vertex to vbo_exec_wrap_upgrade_vertex
- Revert "mesa: check for z=0 in _mesa_Vertex3dv()"
- mesa: remove _mesa_index_buffer::index_size in favor of index_size_shift
- mesa: optimize get_index_size
- mesa: deduplicate draw indirect functions
- vbo: merge more primitive types for glBegin/End (v2)
- vbo: merge draws even when begin==0 or end==0
- glthread: don't generate the sync fallback if the call size is not variable
- glthread: don't prefix variable_data with const
- glthread: inline _mesa_unmarshal_dispatch_cmd and convert the switch to a table
- glthread: reduce pointer dereferences in glthread_unmarshal_batch
- glthread: use int instead of size_t where it's OK
- glthread: simplify repeated function sequences in marshal_generated.c
- glthread: don't insert _mesa_post_marshal_hook into every function
- glthread: don't increment variable_data if it's the last variable-size param
- glthread: add GL_DRAW_INDIRECT_BUFFER tracking and generator support
- glthread: add/update count and marshal fields for many GL functions
- glthread: handle complex pointer parameters and support GL functions with strings
- glthread: check the size of all variable params and clean up the code
- glthread: replace custom ClearBuffer marshalling with generated one
- glthread: add support for TexParameteri and SamplerParameteri functions
- glthread: add support for glFog, glLight, glLightModel, glTexEnv, glTexGen
- glthread: add support for glClearNamedFramebuffer, glMaterial, glPointParameter
- glthread: add support for glCallLists, glPatchParameterfv
- glthread: add support for glMemoryObjectParameteriv, glSemaphoreParameterui64v
- glthread: don't insert an empty line after (void) cmd;
- glthread: add marshal_call_after and remove custom glFlush and glEnable code
- glthread: track for each VAO whether the user has set a user pointer
- glthread: sync instead of disabling glthread for non-VBO pointers
- glthread: replace custom glBindBuffer marshalling with generated one
- glthread: merge glBufferData and glNamedBufferData into 1 set of functions

- glthread: merge glBufferSubData and glNamedBufferSubData into 1 set of functions
- glthread: add custom marshalling for glNamedBuffer(Sub)DataEXT
- glthread: fix a crash with incorrect glShaderSource parameters
- glthread: fall back if a param size is non-zero and a pointer param is NULL
- radeonsi: add a bug workaround for NGG - LATE_ALLOC_GS
- ac: add a bug workaround for the 100% NGG culling case
- radeonsi: determine uses_bindless_samplers correctly
- st/mesa: flush the bitmap cache before st/dri and vbo flushes
- st/mesa: fix a possible crash with selection and feedback modes
- gallium/cso_context: remove cso_delete_xxx_shader helpers to fix the live cache
- st/mesa: keep serialized NIR instead of nir_shader in st_program
- vbo: use vbo_exec_wrap_upgrade_vertex for glVertex in ATTR_UNION
- vbo: fix transitions from glVertexN to glVertexM where M < N
- vbo: fix vbo_copy_vertices for GL_PATCHES and adjacency primitive types
- gallium: add PIPE_CAP_DRAW_INFO_START_WITH_USER_INDICES
- mesa: don't unroll glMultiDrawElements with user indices for gallium
- radeonsi/gfx10: cache metadata in L2 on small chips
- radeonsi: set better tessellation tunables on gfx9 and gfx10
- radeonsi: tune primitive binning for small chips
- ac: add radeon_info::use_late_alloc to control LATE_ALLOC globally
- ac: disable late alloc on small gfx10 chips
- gallium/u_threaded: don't sync the thread for all unsychronized mappings
- gallium/u_vbuf: simplify the first if statement in u_vbuf_upload_buffers
- ac: unify denorm setting enforcement
- ac: set new LLVM denormal flags
- ac: don't set old denormals flags with LLVM >= 11
- nir: fix clip/cull_distance_array_size in nir_lower_clip_cull_distance_arrays
- mesa: use vbo_attrib_tmp.h to generate display list vertex attrib functions
- mesa: remove redundant api_loopback functions
- glthread: align the batch buffer to 8 bytes for pointers and doubles again
- glthread: enable display lists
- glthread: track VAOs created by CreateVertexArrays
- glthread: don't execute any custom VAO and BindBuffer code in the Core profile
- glthread: remove debug_print_marshal function
- glthread: clean up debug_print_sync code
- glthread: don't declare unmarshal functions as inline

- winsys/radeon: change to 3-space indentation
- driconf: enable glthread for "From The Depths"
- glthread: remove _mesa_post_marshal_hook, because it's not very useful
- glthread: simplify printing safe_mul in gl_marshal.py
- glthread: autogenerate prototypes for custom-marshalled functions
- glthread: move buffer functions into glthread_bufferobj.c
- glthread: rename marshal.h/c to glthread_marshal.h and glthread_shaderobj.c
- mesa: put gl_thread_state inside gl_context to remove pointer indirection
- glthread: handle buffer unbinding via glDeleteBuffers
- glthread: rename non_vbo helper functions
- glthread: track which vertex array attribs are enabled
- glthread: ignore vertex arrays with user pointers if they're disabled
- glthread: remove the marshal_fail XML attribute
- vbo,gallium: make glBegin/End buffer size configurable by drivers
- ac: fix fast division
- st/mesa: fix use of uninitialized memory due to st_nir_lower_builtin
- glthread: inline SET_func and add -O1 to build _mesa_create_marshal_table faster
- glthread: declare marshal and unmarshal functions as non-static
- glthread: compile marshal_generated.c faster by breaking it up into 8 files
- nir: add and gather shader_info::writes_memory
- glsl_to_tgsi: set shader_info::writes_memory
- mesa: allow out-of-order drawing to optimize immediate mode if it's safe
- radeonsi: enable full out-of-order drawing when allow_draw_out_of_order is set
- mesa: try to fix the android build
- Move compiler.h and imports.h/c from src/mesa/main into src/util
- mesa: don't use <> for including internal headers
- util: stop including files from mesa/main
- radv: stop including files from mesa/main
- util: don't include p_defines.h and u_pointer.h from gallium
- util: remove duplicated MALLOC_STRUCT and CALLOC_STRUCT
- radeonsi: remove obsolete TODO comment related to compute-based culling
- radeonsi: fix incorrect ordered_wave_id initilization for compute-based culling
- radeonsi: set amdgpu-gds-size for mode == 2 of compute-based culling
- radeonsi: always create wait_mem_scratch for compute-based culling
- radeonsi: add num_vbos_in_user_sgprs into the shader cache key
- radeonsi/gfx10: don't use NGG culling if compute-based culling is used

- radeonsi/gfx10: fix ds.ordered.add intrinsic for compute-based culling
- radeonsi/gfx10: user correct ACQUIRE_MEM packet for compute-based culling
- radeonsi/gfx10: fix the wave size for compute-based culling
- radeonsi/gfx10: fix descriptors and compute registers for compute-based culling
- gallium/u_threaded: call the driver to pin threads to L3 immediately
- st/mesa: add environment variable pin_app_thread for faster glthread on AMD Zen
- driconf: whilelist more games for glthread
- mesa: optimize initialization of new VAOs
- mesa: don't ever set NullBufferObj in gl_vertex_array_binding
- mesa: don't ever bind NullBufferObj for glBindBuffer targets
- mesa: don't ever bind NullBufferObj to glBindBuffer(Base,Range) slots
- mesa: remove NullBufferObj
- mesa: remove no longer needed _mesa_is_bufferobj function
- mesa: precompute _mesa_primitive_restart_index during state changes
- mesa: split _mesa_primitive_restart_index into a function without gl_context
- vbo: expose helper function vbo_get_minmax_index_mapped for glthread
- util: move and adjust the vertex upload heuristic equation from u_vbuf
- st/mesa: fix a crash due to passing a draw vertex shader into the driver
- ac: out-of-order rasterization is not supported on gfx10
- ac,radeonsi: simplify checking for Navi1x chips
- radeonsi: use pipe_blend_state::max_rt to update fewer blend registers
- ac: force enable -structurizecfg-skip-uniform-regions for LLVM 11
- ac: update and document fast math flags used by radeonsi
- ac: generate FMA for inexact instructions for radeonsi
- ac: reassociate FP expressions for inexact instructions for radeonsi
- mesa: replace _NEW_EVAL with vbo_exec_update_eval_maps
- mesa: reset primitive restart state in glClientAttribDefaultEXT
- mesa: remove exec="dynamic" from Draw functions that are not really dynamic
- glthread: use 32-bit align instead of 64-bit ALIGN
- glthread: reduce dereferences of the next batch
- glthread: use GLenum16 in batch buffers to save space
- glthread: sort variables in marshal structures to pack them optimally
- gallium: add PIPE_CAP_MAP_UNSYNCHRONIZED_THREAD_SAFE for glthread
- mesa: add Const.BufferCreateMapUnsynchronizedThreadSafe & MESA_MAP_THREAD_SAFE
- mesa: add offset_is_int32 param into _mesa_bind_vertex_buffer for glthread
- mesa: extend _mesa_bind_vertex_buffer to take ownership of the buffer reference

- mesa: replace GLenum target with gl_shader_stage in NewProgram
- ac/surface: rename micro tile mode enums like gfx10 uses them
- ac/surface: remove RADEON_SURF_TC_COMPATIBLE_HTILE and assume it's always set
- ac/surface: replace RADEON_SURF_OPTIMIZE_FOR_SPACE with !FORCE_SWIZZLE_MODE
- ac/surface: match get_display_flag() with expectations for is_displayable
- ac/surface: don't compute DCC if it's unsupported by DCN on gfx9+
- ac/surface: move non-displayable DCC to the end of the buffer
- ac/surface: add code for gfx10 displayable DCC
- ac/surface: validate that DCC is enabled correctly on gfx9+
- ac: enable displayable DCC on Navi12 & Navi14
- mesa: report GL_INVALID_OPERATION for invalid glTextureBuffer target
- st/mesa: expose more SPIR-V capabilities
- radeonsi: unify and align down the max SSBO/TBO/UBO buffer binding size
- radeonsi: revert an accidental change in si_clear_buffer
- Revert "ac/surface: remove RADEON_SURF_TC_COMPATIBLE_HTILE and assume it's always set"
- Revert "ac: reassociate FP expressions for inexact instructions for radeonsi"
- ac/surface: fix MSAA crash with FORCE_SWIZZLE_MODE on gfx9
- radeonsi: fix compilation of monolithic PS
- radeonsi: don't expose 16xAA on chips with 1 RB due to an occlusion query issue

Marek Vasut (4):

- etnaviv: Destroy rsc->pending_ctx set in etna_resource_destroy()
- etnaviv: Emit PE.ALPHA_COLOR_EXT* on GPUs with half-float support
- etnaviv: Fix depth stencil ops on GC880/GC2000
- etnaviv: Disable seamless cube map on GC880

Mark Janes (2):

- nir: check shader type before writing to shaderinfo.tess union
- nir: place aligned members after bitfields in shader_info.tess

Mark Menzynski (2):

- util/blob: Add overwrite function for uint8
- tgsi/util: Change boolean for bool

Martin Fuzzey (3):

- freedreno: android: fix build failure on android due to python version
- freedreno: android: add a6xx-pack.xml.h generation to android build
- freedreno: android: fix build of perfcounters.

Mathias Fröhlich (19):

- egl: Implement getImage/putImage on pbuffer swrast.

- mesa: Fix FLUSH_VERTICES in SubpixelPrecisionBiasNV.
- egl: Fix A2RGB10 platform_{device,surfaceless} PBuffer configs.
- egl: Factor out dri2_add_pbuffer_configs_for_visuals {device,surfaceless}.
- mesa: Check for OpenGL state change before flushing vertices.
- mesa: Flush vertices before changing the OpenGL state.
- i965: Move down genX_upload_sbe in profiles.
- iris: Move down iris_emit_sbe_swiz in profiles.
- i965: Use 32 bit u_bit_scan for vertex attribute setup.
- i965: Use the VAOs binding information in array setup.
- i965: Test original vertex array pointer to skip array upload.
- i965: Split merge_inputs and clear_buffers.
- i965: Reorder workaround flags computation.
- i965: Remove glbinding from brw_vertex_element.
- mesa: Remove now unused _mesa_draw_attrib_and_binding.
- mesa: Remove now unused _mesa_draw_attrib.
- mesa: Provide gl_vertex_format accessors.
- i965: Make use of the vertex format functions in i965.
- i965: Use gl_vertex_format in brw_vertex_element.

Matt Turner (11):

- intel/tools: Do not print type/qualifiers/name for c_literal
- intel/vec4: Make implied_mrf_writes() a vec4_instruction method
- intel/compiler: Remove unnecessary local variables
- intel/compiler: Make instructions_to_schedule a local variable
- intel/compiler: Mark some methods and parameters const
- intel/compiler: Mark visitor parameters to scheduler const
- intel/compiler: Pass backend_shader * to cfg_t()
- intel/compiler: Pass shader_stats for each SIMD mode
- intel/compiler: Discount NOPs from instruction counts
- isl: Avoid EXPECT_DEATH in unit tests
- meson: Specify the maximum required libdrm in dri.pc

Mauro Rossi (5):

- android: gallium/auxiliary: fix “Unused source files” in tesselator
- android: aco: fix PIPE_FORMAT related building errors
- android: r600/sfn: fix includes and libmesa_nir dependency
- android: r600/sfn: Add GDS instructions
- android: aco: add various compiler statistics

Michel Dänzer (33):

- gitlab-ci: Update to latest ci-templates HEAD
- gitlab-ci: Pass -j4 to make
- gitlab-ci: Merge ccache and libxml2-utils into main apt-get install
- gitlab-ci: Add ppc64el and s390x cross-build jobs
- gitlab-ci: Build radeonsi & RADV in the ppc64el job
- llvmpipe: Bump test timeout to 180 seconds
- gitlab-ci: Only use gstreamer runners for the s390x job for now
- gitlab-ci: Sort random failure softpipe skips
- gitlab-ci: Add three more dEQP-GLES31 tests to softpipe skips
- st/vdpau: Only call is_video_format_supported hook if needed
- winsys/amdgpu: Make local variable r signed
- util: Change os_same_file_description return type from bool to int
- gitlab-ci: Drop "test-" prefix from llvmpipe/softpipe job names
- gitlab-ci: Distribute jobs across more stages
- gitlab-ci: Always name artifacts archive after the job producing it
- gitlab-ci: Don't restrict ppc64el/s390x build jobs to gstreamer runners
- gitlab-ci: Don't use buster-backports packages by default for x86_build
- gitlab-ci: Fold scons-swr job into scons job
- gitlab-ci: Move classic driver testing to a new meson-classic job
- llvmpipe: Use uintptr_t for pointer values
- gitlab-ci: Enable more Gallium drivers in meson-i386 job
- gitlab-ci: Restrict s390x/ppc64el jobs to packet runners
- gitlab-ci: Update to current templates
- gitlab-ci: Rename "paths" YAML anchor to "all_paths"
- gitlab-ci/lava: Add needs: for container image to test jobs (again)
- gitlab-ci: Don't require triggering build/test jobs manually
- gitlab-ci: Run merge request pipelines automatically only for Marge Bot
- gitlab-ci: Use all_paths in .test-manual rules
- gbm/dri: Propagate queryDmaBufModifiers return value
- amd/addrlib: Use enum instead of sparse chars to identify dimensions
- mesa: Skip 3-byte array formats in _mesa_array_format_flip_channels
- Revert "ac,radeonsi: fix compilations issues with LLVM 11"
- Revert "gallium/gallivm: fix compilation issues with llvm 11"

Mike Blumenkrantz (6):

- zink: set UBO alignments in nir_intrinsic_load_uniform lowering

- zink: remove framebuffer cache
- zink: explicitly unref old fb object when setting new one
- iris: move iris_vtable to iris_screen
- gallium: add pipe cap for scissored clears and pass scissor state to clear() hook
- iris: handle PIPE_CAP_CLEAR_SCISSORED

Nanley Chery (6):

- isl: Add a module which manages aux resolves
- iris: Use isl_aux_usage_has_fast_clear()
- iris: Use ISL's access preparation functions
- iris: Use isl_aux_state_transition_write()
- i965: Use ISL's access preparation functions
- i965: Use isl_aux_state_transition_write()

Nataraj Deshpande (1):

- dri_util: Update internal_format to GL_RGB8 for MESA_FORMAT_R8G8B8X8_UNORM

Neha Bhende (2):

- svga: fix size of format_conversion_table[]
- svga: Use pipe_shader_state_from_tgsi to set shader state

Neil Armstrong (4):

- gitlab-ci/lava: fix handling of lava tags
- Revert "ci: Remove T820 from CI temporarily"
- gitlab-ci: add FILES_HOST_URL and move FILES_HOST_NAME into jobs
- gitlab-ci: re-enable mali400/450 and t820 jobs

Neil Roberts (17):

- nir/opcodes: Add nir_op_f2fmp
- glsl: Add support for float16 types in the IR tree
- glsl: Add IR conversion ops for 16-bit float types
- glsl: Add b2f16 and f162b conversion operations
- glsl: Add ir_unop_f2fmp
- glsl/validate: Allow float16 in the expression tree
- glsl/lower_instructions: Use float16 constants when appropriate
- glsl/opt_minmax: Add support for float16
- glsl: Add a method to get precision from a deref instruction
- glsl/hierarchical_visitor: Call leave_callback on leaf nodes
- glsl: Add an IR lowering pass to convert mediump operations to 16-bit
- glsl/standalone: Add an option to lower the precision
- glsl: Add unit tests for the lower_precision pass

- freedreno/ir3: Lower bools to bitsize
- glsl: Inline builtins in a separate pass
- glsl/lower_precision: Lower builtins depending on arguments
- glsl/lower_precision: Use vector.back() instead of vector.end()[-1]

Paulo Zanoni (8):

- intel: fix the gen 11 compute shader scratch IDs
- intel: fix the gen 12 compute shader scratch IDs
- intel/device: bdw_gt1 actually has 6 eus per subslice
- anv: multiply the scratch space by 4 on gen9-10 like iris and i965
- iris: remove hole from struct iris_bo
- iris: remove unnecessary forward declaration
- iris: remove useless bo->gtt_offset assignment
- iris: make BATCH_SZ smaller by BATCH_RESERVED bytes

Peng Huang (1):

- radeonsi: make si_fence_server_signal flush pipe without work

Pierre Moreau (1):

- clover/nir: Check the result of spirv_to_nir

Pierre-Eric Pelloux-Prayer (44):

- radeonsi/ngg: add VGT_FLUSH when enabling fast launch
- radeonsi: test subsampled format in testdma
- format: add format_to_chroma_format
- gallium/video: remove pipe_video_buffer.chroma_format
- gallium/vl: add 4:2:2 support
- radeonsi: fix surf_pitch for subsampled surface
- st/va: enable 4:2:2 chroma format
- st/va: add support YUY2
- radeonsi: remove AMD_DEBUG=sisched option
- omx: fix build with gcc 10
- meson: enable -fno-common by default
- gitlab-ci: rules:changes to test on tested drivers changes
- vdpau: remove bogus assert
- st/mesa: disallow deferred flush if there are multiple contexts
- radeonsi: enable glsl_zero_init for Curse of the Dead Gods
- radeonsi: clarify the conditions when FLUSH_AND_INV_DB is needed
- util/os_file: extend os_read_file to return the file size
- util/u_process: add util_get_process_exec_path

- util/xmlconfig: add new sha1 application attribute
- radeonsi: enable workarounds for YoYo engine based games
- util/u_process: fix Windows build
- nir: update uses_demote flag in discard_to_demote pass
- ac: fix ac_build_is_helper_invocation when postponed_kill is null
- util: fix process_test path
- ddebug: add missing forward declaration
- radeon: fix includes
- radeonsi: switch to 3-spaces style
- radeon: switch to 3-spaces style
- gallium/util: let shader live cache users know if a hit occured
- radeonsi: dump shader stats when hitting the live cache
- util/xmlconfig: fix sha1 comparison code
- mesa: update pipeline when re-linking a program in use
- gallium/u_threaded: flush batch when hitting mapping limit
- radeonsi: use thread_context::bytes_mapped_limit
- radeonsi: don't assume ctx is always a threaded_context
- radeonsi: skip vs output optimizations for some outputs
- mesa: fix crash in find_value
- gallium/utils: silence strncpy warning
- st/omx: fix gcc warnings
- radeonsi: fix export count
- mesa: add gl_coontext::ForceIntegerTexNearest
- driconf: add force_integer_tex_nearest option
- radeonsi: don't print gs_copy_shader stats for shaderdb
- amd/addrlib: fix forgotten char -> enum conversions

Plamena Manolova (2):

- intel/compiler: Add support for variable workgroup size
- i965: Implement ARB_compute_variable_group_size

Qiang Yu (35):

- lima: remove definition of lima_is_scanout
- lima: use util_copy_framebuffer_state
- lima: always add texture bo to submit
- lima: remove lima_ctx_buff_va submit flags (v2)
- lima: pass array as parameter to PLBU and VS command macros
- lima: delay add plb buffer to submit when flush

- lima: delay plbu head command generation to flush stage (v2)
- lima: add render target to submit by dirty buffer flags
- lima: add missing resolve check for damage and reload
- lima: move syncobj from lima_submit to lima_context
- lima: merge gp/pp submit
- lima: put hardware related info to lima_gpu.h
- lima: move flush code to lima_submit.c
- lima: pass submit parameter for functions in lima_submic.c (v2)
- lima: add lima_submit_create_stream_bo
- lima: adjust pp_stream to use lima_submit_create_stream_bo
- lima: use lima_submit_create_stream_bo for plbu/vs_cmd and pp_stack
- lima: add lima_submit_get
- lima: make lima_submit one time use drop data (v3)
- lima: track write submits of context (v3)
- lima: move plbu/vs_cmd_array into lima_submit
- lima: move resolve into lima_submit
- lima: move pp_max_stack_size to lima_submit
- lima: move damage_rect into lima_submit
- lima: move clear into submit (v2)
- lima: move framebuffer info to lima_submit
- lima: use per submit dump file
- lima: optinal flush submit in lima_clear
- lima: enable multi submit optimization
- lima: move dump check to macro for lima_dump_command_stream_print
- lima: rename lima_submit to lima_job
- lima: fix buffer import with offset
- lima: also check tiled and depth case when import
- lima: set offset when export resource
- panfrost: don't always build bifrost_compiler

Quentin Glidic (1):

- meson: Use dependency.partial_dependency()

Rafael Antognolli (18):

- intel: Load the driver even if I915_PARAM_REVISION is not found.
- intel/tools: Update aubinator_error_decode.
- intel/blorp: Implement GEN:BUG:1605967699.
- iris: Apply the flushes when switching pipelines.

- anv: Wait for the GPU to be idle before invalidating the aux table.
- iris: Split aux map initialization from invalidation.
- iris: Wait for the GPU to be idle before invalidating the aux table.
- intel/isl: Implement D16_UNORM workarounds.
- intel/gen12+: Disable mid thread preemption.
- iris: Enable EXT_depth_bounds_test extension.
- drm-uapi: Update headers from Linux 5.7-rc1.
- i965/bufmgr: Factor out GEM_MMAP ioctl from mmap_cpu and mmap_wc.
- iris/bufmgr: Factor out GEM_MMAP ioctl from mmap_cpu and mmap_wc.
- i965/bufmgr: Add support for MMAP_OFFSET ioctl.
- iris/bufmgr: Add support for MMAP_OFFSET ioctl.
- anv: Add anv_device parameter to anv_gem_munmap.
- anv: Add support for new MMAP_OFFSET ioctl.
- anv: Enable HiZ on multi-layer depth buffers.

Rhys Perry (118):

- aco: fix gfx10_wave64_bpermute
- aco: gfx10_wave64_bpermute reduce op to print_ir
- aco: disable some instruction combining if it could change an exec operand
- aco: improve SCC handling in some SALU combines
- nir: fix nir_const_value_as_uint bit size in load/store vectorizer tests
- gitlab-ci: remove load_store_vectorizer from expected s390x test failures
- aco: add RegisterFile
- aco: add some helpers for filling/testing register ranges
- aco: improve GFX9 1D ddx/ddy assertion
- spirv: improve creation of memory_barrier
- spirv: fix memory_barrier_tcs_patch emission
- aco: keep track of which events are used in a barrier
- aco: fix carry-out size for wave32 v_add_co_u32_e64
- aco: handle v_add_co_u32_e64 in parse_base_offset()
- aco: add new NOP insertion pass for GFX6-9
- aco: improve get_wait_states()
- aco: consider non-hazard writes in handle_raw_hazard_internal
- aco: improve control flow handling in GFX6-9 NOP pass
- aco: only reserve sgprs for vcc if it's used
- aco: fix uninitialized data error in waitcnt pass
- glsl/list: use uintptr_t for exec_node_data()'s subtraction

- aco: add helpers for moving instructions for scheduling
- aco: add helpers for ensuring correct ordering while scheduling
- aco: allow barriers to be skipped during scheduling
- aco: don't stop scheduling at exports
- aco: move some register demand helpers into aco_live_var_analysis.cpp
- aco: add a late kill flag
- aco: set late kill for v_interp_p1_f32 for some APUs
- aco: fix instruction encoding for LS VGPR init bug workaround
- aco: fix operand order for LS VGPR init bug workaround
- nir/gather_info: handle emit_vertex_with_counter
- radv: call nir_shader_gather_info again
- radv/winsys: set has_syncobj_wait_for_submit in the null winsys
- aco: set has_divergent_branch for discards in loops
- aco: handle missing second predecessors at merge block phis
- aco: handle when ACO adds new continue edges
- aco: skip NIR in unreachable merge blocks
- aco: improve check for unreachable loop continue blocks
- aco: emit IR in IF's merge block instead if the other side ends in a jump
- aco: fix boolean undef regclass
- nir/gather_info: fix per-vertex handling in try_mask_partial_io
- aco: remove dead code in handle_operands()
- aco: implement 64-bit VGPR constant copies in handle_operands()
- aco: look at p_{extract,split}_vector's definitions in pred_by_exec_mask()
- glsl: fix race in instance getters
- util/u_queue: fix race in total_jobs_size access
- radv: add code for exposing compiler statistics
- aco: add various compiler statistics
- aco: add vmem/smem score statistic
- radv, aco: collect statistics if requested but executables are not
- radv: fix null winsys gpu_info array
- aco: make PhysReg in units of bytes
- aco: add SDWA_instruction
- aco: print and validate opsel
- aco: add emission support for register-allocated sdwa sels
- aco: remove divergence check in sanitize_if()
- aco: zero-initialize Temp

- aco: improve vector optimization with sub-dword vectors
- aco: fix p_extract_vector validation
- aco: improve p_create_vector RA for sub-dword operands
- aco: clear moved operands in get_reg_create_vector()
- aco: fix 1D textureGrad() on GFX9
- aco: implement various 8/16-bit conversions
- aco: add missing scc clobber to nir_op_unpack_32_2x16_split_y
- aco: fix copy statistic for 64-bit vgpr constant copy
- aco: add VOP3P_instruction
- aco: implement sub-dword swaps
- aco: implement 64-bit sgpr swaps
- nir/lower_bit_size: fix lowering of shifts
- nir/lower_bit_size: fix lowering of {imul,umul}_high
- nir/algebraic: don't undo lowering of 8/16-bit comparisons to 32-bit
- aco: decrease the uses of other copy operations after splitting/removing
- aco: copy-propagate p_create_vector copies of vectors
- aco: remove copy in load_input_from_temps()
- aco: move call to store_output_to_temps in store_ls_or_es_output earlier
- aco: combine VALU and SALU into various VOP3 instructions
- aco: improve code for 32-bit isign
- aco: fix v_or(s_lshl) and v_add(s_lshl) optimizations
- aco: fix outdated label_vec from p_create_vector labelling
- radv: align buffer descriptor sizes to dword
- radv: allocate larger shader memory slabs if needed
- aco: be more careful about using SMEM for load_global
- aco: add and use RegClass::get() helper
- aco: add emit_load helper
- aco: refactor load_lds to use new helpers
- aco: use emit_load helper for VMEM/SMEM loads
- aco: add helpers for splitting stores
- aco: refactor store_lds() to use new helpers
- aco: refactor store_vmem_mubuf() to use new helpers
- aco: refactor visit_store_ssbo() to use new helpers
- aco: refactor visit_store_global() to use new helpers
- aco: refactor visit_store_scratch() to use new helpers
- aco: add and use get_buffer_store_op() helper

- aco: allow 8/16-bit shared loads
- aco: vectorize global loads/stores
- aco: handle undef p_create_vector operands in the optimizer
- aco: clobber scc in s_bfe_u32 in get_alu_src()
- aco: improve sub-dword emit_split_vector() with sgprs
- aco: lower 8/16-bit integer arithmetic
- radv/aco: enable 8/16-bit storage and int8/int16 on GFX8+
- aco: make RegisterFile::block() take a regclass
- aco: check alignment of non-subdword registers in get_reg_specified()
- aco: fix neighboring register check in get_reg_simple()
- aco: split self-intersecting copies instead of swapping
- aco: don't recurse in sub-dword get_reg_simple()
- aco: improve RA for uneven p_split_vector
- aco: add missing adjust_max_used_regs()
- aco: fix sub-dword out-of-bounds check in RA validator
- aco: fix sub-dword overwrite check in RA validator
- aco: add various GFX10 int16 opcodes
- aco: improve clamped integer addition disassembly workaround
- aco: fix vgpr nir_op_vecn with sgpr operands
- aco: consider blocks unreachable if they are in the logical cfg
- aco: remove use of f-strings
- aco: add message to static_assert
- nir: add missing group_memory_barrier handling
- nir/opt_if: run opt_peel_loop_initial_if after all other optimizations
- nir: fix lowering to scratch with boolean access

Rob Clark (147):

- freedreno/drm: readonly cmdstream
- freedreno/ir3: shuffle a few ir3_register fields
- freedreno/ir3: cleanup after lower_locals_to_regs
- freedreno/ir3: fix crash when no non-input instructions
- freedreno/ir3: split out delay helpers
- freedreno/ir3: move nop padding to legalize
- freedreno/ir3: move block-scheduling into legalize
- freedreno/ir3: move atomic fixup after RA
- freedreno/ir3: a bit more optmsgs debug
- freedreno/ir3/ra: make use()/def() functions instead of macros

- freedreno/ir3: fix kill scheduling
- freedreno/ir3: post-RA sched pass
- freedreno/ir3: number instructions from one
- freedreno/ir3: add is_tex_or_prefetch()
- freedreno/ir3: don't precolor unused inputs
- freedreno/ir3: two pass register allocation
- freedreno/a6xx: fix lrz overflow
- freedreno/ir3: add RA sanity check
- freedreno/ir3: remove unused tex arg harder
- freedreno/ir3: create fragcoord instructions in input block
- freedreno/ir3: simplify split from collect
- freedreno/ir3: fix a dirty lie
- freedreno: allow ctx->batch to be NULL
- freedreno/ir3: fold const conversion into consumer
- freedreno: allow INVALID modifier
- freedreno/registers: teach gen_header.py about a3xx_regid
- freedreno/a6xx: few register updates
- freedreno: quiet INFO_MSG
- freedreno/registers: cleanup CP_SET_MARKER
- freedreno/computerator: import parser/lexer from fdre-a3xx
- freedreno/computerator: polish out some of the rust
- freedreno/computerator: rename prefix asm->ir3
- freedreno/ir3: allow block->predecessors to be null
- freedreno/computerator: add computerator
- freedreno/computerator: fix build dependency
- freedreno/ir3: remove from_tgsi
- freedreno/a6xx: remove unused param
- freedreno/a6xx: emit LRZ clear in sysmem too
- freedreno/a6xx: whitespace fix
- freedreno/a6xx: don't emit YIELD packet
- freedreno/a6xx: enable SKIP_IB2_ENABLE properly
- freedreno: honor FD_MESA_DEBUG=nogrow
- freedreno/ir3: remove regmask_set_if_not()
- freedreno/ir3: rewrite regmask to better support a6xx+
- freedreno/ir3: don't hide latency when there is none to hide
- freedreno/ir3: track half-precision live values

- freedreno/ir3: update SFU delay
- freedreno/ir3: fix crash with samgq workaround
- freedreno/ir3: don't precolor unassigned inputs
- freedreno/ir3: fix assert with getinfo
- freedreno/ir3: add assert
- nir/print: show variable precision
- freedreno/ir3: also lower lowp frag outputs
- freedreno/computerator: add hrsq/hlog2/hexp2
- freedreno/ir3: remove extra nops inserted in scheduler
- freedreno/ir3: add simplified stall estimation
- freedreno: fix FD_MESA_DEBUG=inorder
- util/ra: spiff out select_reg_callback
- util/ra: move NO_REG to header
- freedreno/ir3: split out has_latency_to_hide()
- freedreno/ir3: fix has_latency_to_hide
- freedreno/ir3: track register usage in first RA pass
- freedreno/ir3: round-robin RA
- freedreno/ir3: try to avoid syncs
- freedreno/computerator: add performance counter support
- freedreno/fdperf: set locale
- freedreno/a6xx: register update
- freedreno/ir3: small cleanup and comments
- freedreno/ir3: add bary_ij as src for meta:tex_prefetch
- freedreno/ir3: remove unused helper
- freedreno/ir3: fix bogus register footprint with tess/gs
- freedreno/ir3: reformat disasm output
- freedreno/ir3: convert debug bitfield to BITFIELD_BIT()
- freedreno/ir3/ra: add debug option for RA debug msgs
- freedreno/ir3/ra: split-up
- freedreno/ir3/ra: add helper to map name to instruction
- freedreno/ir3/ra: fix target register calculation
- freedreno/ir3/ra: add helper to map name to array
- freedreno/ir3/ra: drop extending output live-ranges
- freedreno/ir3/ra: add def/use iterators
- freedreno/ir3/ra: fix array liveranges
- freedreno/ir3/ra: compute register target from liveranges

- freedreno/ir3/ra: pick higher numbered scalars in first pass
- freedreno/ir3/ra: split building regs/classes and conflicts
- freedreno/ir3/ra: re-work a6xx merged register file conflicts
- gitlab-ci: disable vs2019 build
- freedreno: remove some obsolete debug options
- util: fix u_fifo_pop()
- freedreno: add logging infrastructure
- freedreno/a6xx: timestamp logging support
- freedreno: add some initial fd_log tracepoints
- freedreno/a6xx: add some more tracepoints
- freedreno/log: avoid duplicate ts's
- util: move ALIGN/ROUND_DOWN_TO to u_math.h
- freedreno/ir3: fix android build
- freedreno/log: fix build error
- nir: fix definition of imadsh_mix16 for vectors
- freedreno/ir3/cf: handle widening too
- freedreno/ir3: fixup cat3 32b vs 16b
- freedreno/ir3/cf: skip array load/store
- freedreno/ir3: add a pass to collect SSA uses
- freedreno/ir3/cf: use ssa-uses
- freedreno/a6xx: add some compute logging
- freedreno: fix missing locking
- freedreno/ir3: also precompile compute shaders for shaderdb
- freedreno: limit fp16 to frag and compute
- glsl: don't limit fp16 lowering to frag
- nir: add some swizzle helpers
- nir/lower_amul: fix slot calculation
- freedreno/log: android support
- freedreno/log: spiff out parser some more
- freedreno/log: better decoding for multiple chunks per batch
- freedreno/ir3: spiff out disasm a bit
- freedreno/ir3: make falsedep use's optional
- freedreno/ir3: simplify grouping pass
- freedreno/ir3: fix location of inserted mov's
- freedreno/ir3: new pre-RA scheduler
- freedreno/ir3/sched: awareness of partial liveness

- freedreno/ir3/postsched: remove some leftovers
- freedreno/ir3/postsched: avoid moving tex ahead of kill
- freedreno/ir3: add mov/cov stats
- freedreno/ir3/ra: handle array case for SFU select_reg opt
- freedreno/ir3: better cleanup when removing unused instructions
- freedreno/ir3: rename depth->dce
- freedreno/ir3/ra: cleanup some leftovers
- mesa: avoid redundant VBO updates
- mesa/st: avoid u_vbuf for GLES
- gallium: add # of MRT to blend state
- freedreno/computer: add script to test widening/narrowing
- freedreno/ir3/ra: remove unused variable
- freedreno/ir3/ra: use ir3_debug_print helper
- freedreno/ir3/ra: split out helper for array assignment
- freedreno/ir3/ra: only assign array base in first pass
- freedreno/a6xx+tu: rename VSC_DATA/VSC_DATA2
- freedreno: add helper to estimate # of bins per pipe
- freedreno/a6xx: pre-calculate expected vsc stream sizes
- freedreno/log-parser: support to read gzip'd logs
- freedreno: small whitespace fix
- freedreno: don't realloc idle bo's
- freedreno: mark more state dirty when rebinding resources
- freedreno: optimize rebind_resource()
- freedreno: rebind resource in all contexts
- freedreno: rebind_resource() *before* bo changes
- freedreno/a6xx: invalidate tex state cache entries on rebind
- freedreno: fix buffer import
- freedreno/ir3: fix indirect cb0 load_ubo lowering
- freedreno: clear last_fence after resource tracking

Rohan Garg (5):

- ci: Split out radv build-testing on arm64
- ci: Drop the git dependency in tracie
- tracie: Switch to using shutil.move for cross filesystem moves
- tracie: Print results in a machine readable format
- tracie: Reformat code to fix indentation

Roland Scheidegger (7):

- gallivm: fix crash with bptc border color sampling
- gallivm: fix crash in emit_get_buffer_size
- gallivm: disable rgtc/latc SNORM accellerated fetches
- gallium/util: Add back (and rename) util_float_to_half implementation
- gallivm: fix rgtc2 format
- gallivm: switch the mask6/mask7 cases for signed rgtc formats
- gallivm: fix stream id fetch

Roman Stratiienko (3):

- panfrost: Align Android makefiles with recent changes
- lima: Add missing source file to Android.mk
- panfrost: Align Android makefiles with recent changes

Sagar Ghuge (13):

- intel/isl: Move get_format_encoding function to isl
- intel/isl: Switch to R8_UNORM format for compatiblity
- intel/tools: Handle illegal instruction
- intel/tools: Handle STATE_REG in typed source operand
- intel/tools: Set correct address register file and number in i965_asm
- intel/tools: Add test for address register as source
- intel/tools: Add test for state register as source
- intel/tools: Print c_literals 4 byte wide
- intel/tools: Allow i965_disasm to disassemble c_literal input type
- intel/genxml: Add patch count threshold field on gen12
- intel/compiler: Track patch count threshold
- anv: Set patch count threshold in 3DSTATE_HS
- iris: Set patch count threshold in 3DSTATE_HS

Samuel Iglesias Gonsálvez (2):

- radv: check buffer size in vkCreateBuffer()
- radv: set sparseAddressSpaceSize to RADV_MAX_MEMORY_ALLOCATION_SIZE

Samuel Pitoiset (197):

- aco: fix MUBUF VS input loads when expanding vec3 to vec4 on GFX6
- aco: do not use ds_{read,write}2 on GFX6
- gitlab-ci: disable a630 tests as mesa-cheza is down (again)
- aco: fix waiting for scalar stores before “writing back” data on GFX8-GFX9
- radv: make sure to not submit any IBs when RADV_FORCE_FAMILY is set
- radv: set the chip name to GCN-NOOP when RADV_FORCE_FAMILY is set
- aco: fix creating v_madak if v_mad_f32 has two sgpr literals

- nir: do not use De Morgan's Law rules for flt and fge
- radv: fix line width range and granularity
- radv: implement VK_EXT_line_rasterization
- radv: remove LLVM sicheduler enable for The Talos Principle
- radv: remove RADV_DEBUG=nosisched and RADV_PERFTEST=sisched
- radv: remove unused RADV_HASH_SHADER_IS_GEOM_COPY_SHADER
- radv: remove unnecessary RADV_DEBUG=nobatchchain option
- docs/new_features: empty the feature list for the 20.1 cycle
- radv: enable shaderStorageImageMultisample on GFX6-GFX7
- radv: enable VK_EXT_sampler_filter_minmax on GFX6
- radv: enable VK_NV_compute_shader_derivatives on GFX6-GFX7
- radv: add a comment about VK_AMD_mixed_attachment_samples on GFX6-GFX7
- docs/envvars: document RADV_TEX_ANISO
- radv/winsys: add a new flag that requests zerovram allocations
- radv: use RADEON_FLAG_ZERO_VRAM when creating the trace BO
- radv: add the trace BO to the BO list at submit time
- radv: implement a dummy winsys for creating devices without AMDGPU
- ac,radeonsi: add ac_gpu_info::lds_size_per_cu
- ac: add more ac_gpu_info related shader fields
- radv/gfx10: adjust the number of simd per compute unit
- radv/gfx10: adjust SGPRs/VGPRs related info
- radv/gfx10: adjust the LDS size used to compute waves
- radv/gfx10: adjust the number of VGPRs used to compute waves
- radv: make use of ac_gpu_info::max_wave64_per_simd
- radv: fix creating null devices if KHR_display is enabled
- ac/llvm: fix 64-bit fmed3
- ac/llvm: fix 16-bit fmed3 on GFX8 and older gens
- ac/llvm: flush denorms for nir_op_fmed3 on GFX8 and older gens
- ac: add more fields to ac_gpu_info
- ac/registers: add definitions for thread trace
- radv: add a small helper that allows to submit internal CS
- radv: add initial SQ Thread Trace support for GFX9
- radv: emit thread trace markers after every draw/dispatch call
- radv: add initial SQTT files generation support
- radv: allow to capture SQTT traces with RADV_THREAD_TRACE=<start_frame>
- radv: fix 32-bit build failure in radv_queue_internal_submit()

- radv: fix size of sqtt_file_chunk_asic_info on 32-bit system
- radv/rgp: adjust trace memory/shader clocks to fix frame duration
- radv/sqtt: do not assume that the number of shader engines is 4
- radv/sqtt: update SPI_CONFIG_CNTL.EXP_PRIORITY_ORDER value
- ac/registers: add definitions for thread trace on GFX10
- radv/sqtt: add support for GFX10
- radv: update entrypoints generation from ANV
- ac: rename lds_size_per_cu to lds_size_per_workgroup
- ac: rename vgpr_alloc_granularity to wave64_vgpr_alloc_granularity
- ac: rename min_vgpr_alloc to min_wave64_vgpr_alloc
- aco: fix image load/store with lod and 1D images
- gitlab-ci: build Fossilize in the test image for VK
- gitlab-ci: add Fossilize support to detect compiler regressions
- gitlab-ci: enable building the test image for VK unconditionally
- gitlab-ci: add a job that runs Fossilize on RADV/Polaris10
- radv/winsys: fix missing initializations of shader info in the null device
- radv/sqtt: fix wrong check in radv_is_thread_trace_complete()
- radv/sqtt: tidy up radv_emit_thread_trace_{start,stop}
- radv/sqtt: add radv_copy_thread_trace_info_regs() helper
- ac/registers: adjust some definitions for thread trace on GFX8
- radv/sqtt: add support for GFX8
- radv/sqtt: abort if SQTT is used on GFX6-GFX7
- ac: add ac_gpu_info::cu_mask to store bitmask of compute units
- radv/rgp: report correct cu_mask info
- radv/rgp: report correct system ram size
- nir/lower_input_attachments: remove bogus assert in try_lower_input_texop()
- radv/entrypoints: declare a driver internal layer for SQTT
- radv: use device entrypoints from the SQTT layer if enabled
- radv/sqtt: add a helper that emits thread trace userdata markers
- radv: initial implementation of the driver internal layer SQTT
- radv/sqtt: describe begin/end command buffers with user markers
- radv/sqtt: describe draw/dispatch and emit event markers
- radv/sqtt: describe render pass color/depthstencil clears
- radv/rgp: bump the instrumentation spec version to 1
- radv/sqtt: describe pipeline and wait events barriers
- gitlab-ci: add rules:changes for RADV

- radv: do not recursively begin/end render pass for meta operations
- radv: fix 32-bits build (again)
- gitlab-ci: build RADV in meson-i386 to avoid 32-bit build failures
- ac/llvm: add missing optimization barrier for 64-bit readlanes
- radv/sqtt: describe begin/end subpass barriers with user markers
- radv/sqtt: describe layout transitions with user markers
- radv/gfx10: cache metadata in L2 on small chips
- radv: use better tessellation tunables on GFX9+
- radv: tune primitive binning for small chips
- radv: rewrite late alloc computation
- radv: use ac_gpu_info::use_late_alloc
- radv: cleanup occurences of use_aco everywhere
- radv: remove radv_shader_variant::aco_used
- radv: remove unnecessary LLVM includes
- radv: add llvm_compiler_shader() helper
- gitlab-ci: remove useless 'patch' package in the VK test image
- gitlab-ci: allow deqp-runner to use the maximum number of jobs
- gitlab-ci: do not set the number of deqp-parallel jobs for RADV CTS
- gitlab-ci: bump Vulkan CTS to 1.2.1.0
- radv/sqtt: handle thread trace capture in sqtt_QueuePresentKHR()
- radv: only inject implicit subpass dependencies if necessary
- radv/gfx10: fix required subgroup size with VK_EXT_subgroup_size_control
- radv/gfx10: fix required ballot size with VK_EXT_subgroup_size_control
- radv: fix random depth range unrestricted failures due to a cache issue
- radv: remove wrong assert that checks compute subgroup size
- radv: fix optional pSizes parameter when binding streamout buffers
- radv/winsys: fix wrong PCI ID for Vega10 in the null winsys
- radv/winsys: spoof some values for num_render_backends in the null winsys
- gitlab-ci: compile fossils with both RADV compiler backends (LLVM/ACO)
- gitlab-ci: compile fossils with more ASICs
- gitlab-ci: add a new stage for RADV CI
- gitlab-ci: add a bunch of new fossils from the Sascha Vulkan demos
- radv/llvm: fix subgroup shuffle for chips without bpermute
- radv: enable VK_KHR_8bit_storage on GFX6-GFX7
- ac/nir: use llvm.amdgcn.rcp for nir_op_frcp
- ac/nir: use llvm.amdgcn.rsq for nir_op_frsq

- ac/nir: use llvm.amdgcn.rcp in ac_build_fdiv()
- nir/algebraic: add fexp2(fmul(flog2(a), 0.5) -> fsqrt(a) optimization
- aco: only break SMEM clauses if XNACK is enabled (mostly APUs)
- aco: always optimize v_mad to v_madak in presence of literals
- ac/nir: split 8-bit load/store to global memory on GFX6
- ac/nir: split 8-bit SSBO stores on GFX6
- radv/llvm: enable 8-bit storage features on GFX6-GFX7
- ac/nir: split 16-bit load/store to global memory on GFX6
- ac/nir: split 16-bit SSBO stores on GFX6
- radv/llvm: enable 16-bit storage features on GFX6-GFX7
- radv: rename decompress/resummarize depth/stencil functions
- radv: rename extra graphics pipeline decompress/resummarize fields
- radv: cleanup creating the decompress/resummarize pipelines
- radv: remove radv_layout_has_htile() helper
- radv: enable lowering of GS intrinsics for the LLVM backend
- ac,radv: add ac_gpu_info::has_double_rate_fp16
- radv: only expose shaderFloat16 for chips with double rate fp16
- radv: only expose storageInputOutput16 for chips with double rate fp16
- radv: only expose fp16 control features for chips with double rate fp16
- radv: only enable TC-compat HTILE for images readable by a shader
- radv: allow TC-compat HTILE with GENERAL outside of render loops
- aco: implement 16-bit nir_op_frexp_sig/nir_op_frexp_exp
- aco: implement 16-bit nir_op_ffract
- aco: implement 16-bit nir_op_fexp2/nir_op_flog2
- aco: implement 16-bit nir_op_ftrunc/nir_op_fround_even
- aco: implement 16-bit nir_op_fsqrt/nir_op_frcp/nir_op_frsq
- aco: implement 16-bit nir_op_ffloor/nir_op_fceil
- aco: implement 16-bit nir_op_fmax/nir_op_fmin
- aco: implement 16-bit nir_op_fabs/nir_op_fneg
- aco: implement 16-bit nir_op_fsub/nir_op_fadd
- aco: implement 16-bit nir_op_fcos/nir_op_fsin
- aco: implement 16-bit nir_op_fmul
- aco: implement 16-bit nir_op_fsat
- aco: implement 16-bit nir_op_fsign
- aco: implement 16-bit nir_op_bcsel
- aco: implement 16-bit nir_op_f2i32/nir_op_f2u32

- aco: implement 16-bit nir_op_ldexp
- aco: implement 16-bit nir_op_fmax3/nir_op_fmin3/nir_op_fmed3
- aco: implement 16-bit comparisons
- aco: implement nir_op_b2f16/nir_op_i2f16/nir_op_u2f16
- aco: fix f2i64/f2u64 with sgprs if the exponent computation overflow
- aco: implement 16-bit nir_op_f2i64/nir_op_f2u64
- aco: fix nir_op_pack_32_2x16_split if one operand is a constant
- radv: add radeon_set_context_reg_rmw() helper
- radv: use RMW packets for updating the maximum sample distance
- aco: fix nir_op_frexp_exp with 16-bit floats and negative exponents
- radv/aco: do not advertise VK_KHR_shader_subgroup_extended_types
- aco: implement nir_op_f2i8/nir_op_f2u8
- aco: fix emitting stream output with tess eval shaders
- radv: do not abort with unknown/unimplemented descriptor types
- radv: fix geometry shader primitives query with ACO on GFX10
- radv: set missing SHARED_VGPR_CNT for NGG VS and ACO
- radv/llvm: fix exporting the viewport index if the fragment shader needs it
- aco: fix exporting the viewport index if the fragment shader needs it
- nir/lower_int64: lower imin3/imax3/umin3/umax3/imed3/umed3
- nir/opt_algebraic: lower 64-bit fmin3/fmax3/fmed3
- gitlab-ci: add a list of excluded tests for RADV
- radv: make sure to export the viewport index if FS needs it
- radv: simplify checking for Navi1x chips
- radv: adjust the supported subgroup stages
- radv: fix robust_buffer_access if enabled via VkPhysicalDeviceFeatures2
- gitlab-ci: add lists of expected failures for RADV CI
- ac,radeonsi: fix compilations issues with LLVM 11
- radv: do not expose GTT as device local memory mostly for APUs
- radv: enable FMASK for color attachments only
- radv: remove unused radv_device_memory::map_size field
- radv: track memory heaps usage if overallocation is explicitly disallowed
- radv: advertise VK_AMD_memory_overallocation_behavior
- ac/llvm: fix nir_texop_texture_samples with NULL descriptors
- aco: fix nir_texop_texture_samples with NULL descriptors
- aco: fix adjusting the sample index with FMASK if value is negative
- radv: handle NULL descriptors

- radv: handle NULL vertex bindings
- radv: advertise VK_EXT_robustness2
- gitlab-ci: add a list of expected failures for FIJI with ACO
- ci: fix reporting the number of unexpected/flakes
- radv: report INITIALIZATION_FAILED when the amdgpu winsys init failed
- radv: don't report error with other vendor DRM devices
- aco: fix 64-bit trunc with negative exponents on GFX6
- radv: limit the Vulkan version to 1.1 for Android
- radv: handle different Vulkan API versions correctly
- radv: update the list of allowed Android extensions

Satyajit Sahu (1):

- st/va: GetConfigAttributes: check profile and entrypoint combination

Simon Ser (1):

- mesa: add support for NV_pixel_buffer_object

Simon Zeni (1):

- mesa: enable GL_EXT_draw_instanced for gles2

Sonny Jiang (1):

- radeonsi: enable EXT_texture_shadow_lod

Szymon Andrzejuk (1):

- virgl: Use align_free for align_malloc allocated buffer

Tapani Pälli (27):

- intel/vec4: fix valgrind errors with vf_values array
- glsl: fix a memory leak with resource_set
- iris: fix aux buf map failure in 32bits app on Android
- mesa: introduce boolean toggle for EXT_texture_norm16
- i965: toggle on EXT_texture_norm16
- mesa/st: toggle EXT_texture_norm16 based on format support
- mesa/st: fix formats required for EXT_texture_norm16
- nir: fix compilation warning on glsl_get_internal_ifc_packing
- iris: toggle on PIPE_CAP_MIXED_COLOR_DEPTH_BITS
- nir/glsl: gather bitmask of images used by program
- iris: use the images_used mask in resolve pass
- intel/compiler: detect if atomic load store operations are used
- iris: provide dummy iris_image_view_aux_usage
- iris: move existing image format fallback as a helper function
- iris: determine aux usage during predraw and state setup

- isl: allow compression for storage images on gen12+
- iris: allow compression conditionally for images on gen12
- glsl: set error_emitted true if type not ok for assignment
- mesa/st: unbind shader state before deleting it
- mesa/st: release variants for active programs before unref
- mesa: remove redudant check
- mesa: remove redudant assignment
- glsl: remove redudant assignment
- glsl: stop processing function parameters if error happened
- mesa/st: initialize all winsys_handle fields for memory objects
- anv: remove assert from GetImageMemoryRequirements[2]
- st/mesa: destroy only own program variants when program is released

Thomas Hellstrom (5):

- svga: Fix banded DMA upload
- svga, winsys/svga: Fix persistent memory discard maps
- svga: Treat forced coherent maps as maps of persistent memory
- gallium/pipebuffer: Use persistent maps for slabs
- winsys/svga: Optionally avoid caching buffer maps

Thong Thai (7):

- Revert "st/va: Convert interlaced NV12 to progressive"
- gallium/auxiliary/vl: fix bob compute shaders for deint yuv
- st/va: remove unneeded code
- st/va/postproc: reallocate interlaced destination buffer
- radeonsi: add 10-bit HEVC encode support for VCN2.0 devices
- radeon: add support for 10-bit HEVC encoding to VCN 2.0
- st/va: add check for P010 and P016 encode/decode support

Timothy Arceri (51):

- glsl: fix gl_nir_set_uniform_initializers() for image arrays
- glsl: fix possible memory leak in nir uniform linker
- glsl: set the correct number of samplers in a shader
- glsl: set the correct number of images in a shader
- glsl: fix resizing of the uniform remap table
- glsl: reset next_image_index count for each shader stage
- glsl: fix sampler index calculation in nir linker
- glsl: add some error checks to the nir uniform linker
- glsl: move nir link uniforms struct defs earlier

- glsl: move add_parameter() earlier in nir link uniforms
- glsl: move get_next_index() earlier in nir link uniforms
- glsl: add name support to nir uniform linker
- glsl: correctly find block index when linking glsl with nir linker
- nir: add glsl_get_internal_ifc_packing() helper
- nir: add glsl_get_std140_base_alignment() helper
- nir: add glsl_get_std140_size() helper
- nir: add glsl_get_std430_base_alignment() helper
- nir: add glsl_get_std430_size() helper
- glsl: add std140 and std430 layouts to nir uniform linker
- glsl: correctly set explicit offsets for struct members
- glsl: find the base offset for block members from unnamed blocks
- glsl: nir linker fix setting of ssbo top level array
- glsl: set ShaderStorageBlocksWriteAccess in the nir linker
- glsl: add support for builtins to the nir uniform linker
- glsl: dont try to assign uniform storage for uniform blocks
- glsl: add subroutine support to nir linker
- glsl: fix varying packing for 64bit integers
- nir: fix packing of TCS varyings not read by the TES
- nir: fix crash in varying packing on interface mismatch
- glsl_to_nir: remove dead code
- radeonsi: don't lower constant arrays to uniforms in GLSL IR
- nir: make opt_if_loop_terminator() less strict
- nir: add matrix_layout to nir_variable data
- glsl: fix struct offsets in the nir uniform linker
- glsl: tidy up uniform storage value count code in NIR linker
- Revert "glsl: fix resizing of the uniform remap table"
- glsl: fix explicit locations for the glsl linker
- glsl: error check max user assignable uniform locations
- glsl: fix block index in NIR uniform linker
- glsl: pull mark_array_elements_referenced() out into common helper
- glsl: only set stage ref when uniforms referenced in stage
- nir/gcm: allow derivative dependent intrinisics to be moved earlier
- nir/gcm: be more conservative about moving instructions from loops
- nir/gcm: dont move movs unless we can replace them later with their src
- glsl: add bindless support to nir uniform linker

- glsl: fix gl_nir_set_uniform_initializers() for bindless textures
- st/glsl_to_nir: make use of nir linker for linking uniforms
- glsl: some nir uniform linker fixes
- glsl: remove some duplicate code from the nir uniform linker
- glsl: stop cascading errors if process_parameters() fails
- glsl: fix slow linking of uniforms in the nir linker

Timur Kristóf (90):

- aco/optimizer: Don't combine uniform bool s_and to s_andn2.
- radv: Move some helper functions to the radv_shader.h header file.
- aco: Extract setup_gs_variables into a separate function.
- aco: Setup tessellation control shader variables.
- aco: Implement load_tess_coord.
- aco: Implement load_primitive_id for tessellation shaders.
- aco: Implement load_patch_vertices_in.
- aco: Implement load_invocation_id for tessellation control shaders.
- aco: Implement control_barrier for tessellation control shaders.
- aco: Implement memory_barrier_tcs_patch.
- aco: Implement load_view_index for TCS and TES.
- aco: Setup correct HW stages when tessellation is used.
- aco: Use mesa shader stage when loading inputs.
- aco: Remove vertex_geometry_gs assertion from merged shaders.
- aco: Extract LDS alignment calculation to a separate function.
- aco: Remove esgs_itemsize from LDS alignment calculation.
- aco: Introduce new VMEM load/store helpers.
- aco: Introduce new helpers for calculating address offsets.
- aco: Refactor load_per_vertex_input in preparation for tessellation.
- aco: Refactor VS output stores in preparation for tessellation.
- aco: Slight fix to lds_store and lds_load.
- aco: Fix combining DS additions in the optimizer.
- aco: Implement tessellation control shader input/output.
- aco: Store VS outputs correctly when tessellation is used.
- aco: Fix LS VGPR init bug on affected hardware.
- radv: Enable ACO for tessellation control shaders.
- aco: Setup tessellation evaluation shader variables.
- aco: Use TES output info when TES runs on the VS stage.
- aco: Store TES outputs when TES runs on the HW VS stage.

- aco: Enable streamout when TES runs on the HW VS stage.
- aco: Implement loading TES inputs.
- radv: Enable ACO for TES when there is no GS.
- aco: Enable running TES as ES, including merged TES+GS.
- radv: Enable ACO on all stages.
- aco: Don't generate an if when the first part of a merged HS or GS is empty.
- aco: Store tess factors in VMEM only at the end of the shader.
- aco: Only write TCS outputs to LDS when they are read by the TCS.
- aco: Don't store TCS outputs to LDS when we're sure that none are read.
- nir: Add ability to lower non-const quad broadcasts to const ones.
- radv: Enable lowering dynamic quad broadcasts.
- radv: Enable subgroup shuffle on GFX10 when ACO is used.
- aco: Create null exports in instruction selection instead of assembler.
- aco: Extract tcs_driver_location_matches_api_mask to separate function.
- aco: Fix handling of tess factors.
- aco: Allow combining TCS output VMEM stores.
- aco: Allow combining LDS loads when loading tess factors.
- aco: Skip 2nd read of merged wave info when TCS in/out vertices are equal.
- aco: Use more optimal sequence at the beginning of merged shaders.
- nir: Collect if shader uses cross-invocation or indirect I/O.
- aco: Treat outputs of the previous stage as inputs of the next stage.
- aco: Change isel inputs/outputs to a flat array.
- aco: Zero-fill undefined elements in create_vec_from_array.
- aco: Extract setup_tcs_info to a separate function.
- aco: Fix workgroup size calculation.
- aco: Extract store_output_to_temps into a separate function.
- aco: When LS and HS invocations are the same, pass LS outputs in temps.
- aco: Don't store LS VS outputs to LDS when TCS doesn't need them.
- aco: Fix crash in insert_wait_states.
- aco: Extract uniform if handling to separate functions.
- aco: Print block_kind_export_end.
- aco: Extract merged_wave_info_to_mask to its own function.
- aco: Treat s_setprio as a scheduling barrier.
- aco/ngg: Add new stage for hw_ngg_gs.
- aco/ngg: Initialize exec mask for NGG VS and TES.
- aco/ngg: Fix exports for NGG VS and TES.

- aco/ngg: Setup NGG VS and TES stages.
- aco/ngg: Implement NGG VS and TES.
- aco/ngg: Schedule position exports of NGG VS/TES.
- aco/ngg: Run GS_ALLOC_REQ on priority 3 for NGG VS and TES.
- radv: Enable ACO for NGG VS/TES, but disable NGG for ACO GS.
- aco: Print shader stage in aco_print_program.
- radv: Print shader stage before disassembly.
- radv: Add inputs read by TES to radv_shader_info.
- aco: Only store TCS outputs to VMEM when they are read by TES.
- aco: Increase barrier_count to 7 to include barrier_barrier.
- aco: Abort when RA can't find a register.
- aco: Const correctness for get_barrier_interaction.
- aco: Const correctness for aco_print_ir.
- aco: Use 24-bit multiplication in TCS I/O
- aco: Use 24-bit multiplication for NGG wave id and thread id.
- aco: Move s_setprio to correct place after the gs_alloc_req.
- radv: Refactor calculate_tess_lds_size and get_tcs_num_patches.
- aco: Use context variables instead of calculating TCS inputs/outputs.
- aco: Remember VS/TCS output driver locations.
- aco: Calculate workgroup size of legacy GS.
- aco: Set config->lds_size when TES or VS is running on HW ESGS.
- nir: Add new linking helper to set linked driver locations.
- radv: Use new linking helper to set default driver locations.
- aco: Use new default driver locations.
- radv: Use smaller esgs_itemsize for ACO.

Tobias Jakobi (1):

- meson: Link Gallium Nine with ld_args_build_id

Tomasz Pyra (1):

- gallium/swr: spin-lock performance improvement

Tomeu Vizoso (34):

- panfrost: Print intended field when decoding
- panfrost: Add more info to some assertions
- pan/midgard: Handle nir_intrinsic_load_barycentric_centroid
- panfrost: Use DBG macro to avoid noise in the console
- panfrost: Fix decoding of tiled 3D textures
- panfrost: Only clamp the LOD to disable mipmapping when needed

- gitlab-ci: Switch kernel for LAVA jobs to 5.5
- gitlab-ci: Disable the lima job for now
- gitlab-ci: Run GLES3 tests in dEQP on Panfrost
- panfrost: Remove some more prints to stdout
- gitlab-ci: Move to 5.5 kernel plus fixes for Panfrost
- gitlab-ci: Use PAN_MESA_DEBUG=gles3 for Panfrost
- gitlab-ci: Remove GLES3 test from Panfrost fails list
- gitlab-ci: Skip dEQP-GLES3.functional.shaders.derivate.*
- gallium: Add forgotten docs for new CAPs related to transform feedback
- gitlab-ci: Update renderdoc
- gitlab-ci: Use surfaceless platform also for apitrace
- gitlab-ci: Place files from the Mesa repo into the build tarball
- gitlab-ci: Serve files for LAVA via separate service
- gitlab-ci: Disable jobs for Collabora's LAVA lab
- Revert "gitlab-ci: Disable jobs for Collabora's LAVA lab"
- panfrost: Remove most usage of midgard_payload_vertex_tiler
- panfrost: Pass IS_BIFROST to pandecode_jc
- panfrost: Don't emit write_value jobs on Bifrost
- panfrost: On Bifrost, set the right tiler descriptor
- gitlab-ci: Test virgl driver
- panfrost: Clean up a bit the tiler structs for Bifrost
- panfrost: Emit sampler descriptor on bifrost
- panfrost: Emit texture descriptor on bifrost
- gitlab-ci: Update virglrenderer in the x86_test-gl image
- gitlab-ci: Allow test jobs to add options to the dEQP invocation
- gitlab-ci: Test OpenGL ES 3.1 on virgl
- gitlab-ci: Test Virgl with traces
- panfrost: Add Bifrost texture trampoline BO to batch

Uros Bizjak (1):

- doc: Update features.txt for r600 with misc supported features

Vasily Khoruzhick (19):

- lima: handle early-z and pixel kill better
- lima: implement PLB PP stream cache
- lima: add RGBA5551 and RGBA4444 formats
- lima: don't disable tiling if there's linear modifier in list
- lima: gpir: enforce instruction limit earlier

- panfrost: split index cache into shared part
- lima: enable minmax cache for index buffers
- lima: print gp uniforms if gp debug is enabled
- lima/gpir: improve disassembler output
- lima/gpir: print acc ops even if we have only one source
- lima/gpir: kill dead writes to regs in DCE
- lima/gpir: add better lowering for ftrunc
- lima/gpir: fix crash in schedule_insert_ready_list()
- lima: disable Z16 format
- lima: decode depth/stencil write bits in RSW
- lima: split pixel and texel format tables
- lima: add support for R and RG formats
- lima: Implement lima_texture_subdata
- lima: avoid situations when scissor minx > maxx or miny > maxy

Veerabadhran (1):

- radeon/vce: Move global function pointer si_get_pic_param to local encoder structure Multi gpu use case broken when the function was global

Vilya Harvey (1):

- zink. Don't set incorrect sType in VkImportMemoryFdInfoKHR struct

Vinson Lee (16):

- swr: Fix build with GCC 10.
- lima: Fix build with GCC 10.
- swr: Fix GCC 4.9 checks.
- panfrost: Remove unused anonymous enum variables.
- meson: Enable -Wno-deprecated only for bison > 2.3.
- swr: Fix non-pod-varargs error.
- st/nine: Fix incompatible-pointer-types-discards-qualifiers errors.
- panfrost: Fix gnu-empty-initializer error.
- util/u_process: Add util_get_process_exec_path for macOS.
- mesa: Change _mesa_exec_malloc argument type.
- gallivm: Add missing header for powf.
- swr/rasterizer: Use private functions for min/max to avoid namespace issues.
- swr: Remove Byte Order Mark.
- r600/sfn: Initialize VertexStageExportForGS m_num_clip_dist member variable.
- r600/sfn: Use correct setter method.
- freedreno: Add missing va_end.

Yevhenii Kolesnikov (1):

- intel/compiler: fix cmod propagation optimisations

Zhang, Boyuan (1):

- radeonsi: Add support for midstream bitrate change in encoder

luc (1):

- zink: confused compilation macro usage for zink in target helpers.

4.15 Mesa 20.0.7 Release Notes / 2020-05-14

Mesa 20.0.7 is a bug fix release which fixes bugs found since the 20.0.6 release.

Mesa 20.0.7 implements the OpenGL 4.6 API, but the version reported by glGetString(GL_VERSION) or glGetIntegerv(GL_MAJOR_VERSION) / glGetIntegerv(GL_MINOR_VERSION) depends on the particular driver being used. Some drivers don't support all the features required in OpenGL 4.6. OpenGL 4.6 is **only** available if requested at context creation. Compatibility contexts may report a lower version depending on each driver.

Mesa 20.0.7 implements the Vulkan 1.2 API, but the version reported by the apiVersion property of the VkPhysicalDeviceProperties struct depends on the particular driver being used.

4.15.1 SHA256 checksum

```
fe6e258fe772c3cd2ac01741bf7408058c3ac02d66acff9a6e669bd72e3ea178  mesa-20.0.7.tar.xz
```

4.15.2 New features

4.15.3 Bug fixes

- radv regression on android
- heavy glitches on amd ryzen 5 since version 20.x
- [bisected] [iris] mpv under wayland: failed to import supplied dmabufs: Unsupported buffer format 808669784
- iris: Crash when trying to capture window in OBS Studio
- mesa 20.0.5 causing kitty to crash
- radeonsi: "Torchlight II" trace showing regression on mesa-20.0.6 [bisected]
- [RADV/LLVM/ACO/Regression] After mesa commit a3dc7fffbb7be0f1b2ac478b16d3acc5662dff66 all games stucks at start
- intel/compiler: Register coalesce doesn't move conditional modifiers

4.15.4 Changes

Axel Davy (1):

- gallium/util: Fix leak in the live shader cache

Bas Nieuwenhuizen (2):

- radv: Extend tiling flags to 64-bit.
- winsys/amdgpu: Retrieve WC flags from imported buffers.

Blaž Tomažič (1):

- radeonsi: Fix omitted flush when moving suballocated texture

Christopher James Halse Rogers (1):

- egl/wayland: Fix zwp_linux_dmabuf usage

D Scott Phillips (2):

- intel/fs: Update location of Render Target Array Index for gen12
- anv,iris: Fix input vertex max for tcs on gen12

Danylo Piliaiev (1):

- i965: Fix out-of-bounds access to brw_stage_state::surf_offset

Dave Airlie (1):

- llvmpipo/nir: free compute shader NIR

Dylan Baker (16):

- docs: Add SHA256 sums for 20.0.6
- .pick_status.json: Update to 2efa76f795cb2b2bf00b317c580aeeeddd1e9bc2
- .pick_status.json: Update to 3fac55ce0d066d767d6c6c8308f79d0c3e566ec0
- .pick_status.json: Mark 3fac55ce0d066d767d6c6c8308f79d0c3e566ec0 as denominated
- .pick_status.json: Update to b97cc41aa203fd9fb9f5cf5f5aa7fd40f567917d
- radeonsi: Retab si_get.c
- .pick_status.json: Mark bdd2f284d90b7f07ac5e878490be8d216d0d23c6 as denominated
- .pick_status.json: Update to 6292059662dccd3e151c731a3b108fd0b9e4c606
- .pick_status.json: Mark d80fb024302aa6058945826a79ba0caf9611fcc1 as backported
- .pick_status.json: Mark 9392ddab4399d796fdf37602f586965ec17f2b2a as backported
- .pick_status.json: Update to 6d513eb0db25a272da65822f35907456b544f172
- radeonsi: retab si_shader_llvm_ps.c
- .pick_status.json: Update to d11e4738a86ecac6bb4cfaf5cad5c1d32169b18f
- radeonsi: retab
- .pick_status.json: Update to 0bea2a13212be10982e14617002a3ff851b84717
- .pick_status.json: Update to d76e722ed63607ecead2c66ef9f3a37a12b62bab

Ian Romanick (1):

- nir/algebraic: Optimize ushr of pack_half, not ishr

Ivan Molodetskikh (1):

- egl: allow INVALID format for linux_dmabuf

Jason Ekstrand (3):

- nir/copy_prop_vars: Report progress when deleting self-copies

- intel/fs: Don't delete coalesced MOVs if they have a cmod
- vulkan: Allow destroying NULL debug report callbacks

Jose Maria Casanova Crespo (2):

- v3d: Fix swizzle in DXT3 and DXT5 formats
- v3d: Include supported DXT formats to enable s3tc/dxt extensions

Lionel Landwerlin (3):

- iris: don't assert on unfinished aux import in copy paths
- intel/perf: store the probed i915-perf version
- anv: don't expose VK_INTEL_performance_query without kernel support

Marek Olšák (3):

- mesa: report GL_INVALID_OPERATION for invalid glTextureBuffer target
- radeonsi: unify and align down the max SSBO/TBO/UBO buffer binding size
- radeonsi: fix compilation of monolithic PS

Neil Armstrong (1):

- ci: disable t820/mali4xx tests

Pierre Moreau (1):

- clover/nir: Check the result of spirv_to_nir

Pierre-Eric Pelloux-Prayer (1):

- radeonsi: fix export count

Qiang Yu (1):

- panfrost: don't always build bifrost_compiler

Rhys Perry (2):

- nir: add missing group_memory_barrier handling
- aco: consider blocks unreachable if they are in the logical cfg

Samuel Pitoiset (4):

- radv: report INITIALIZATION_FAILED when the amdgpu winsys init failed
- radv: don't report error with other vendor DRM devices
- aco: fix 64-bit trunc with negative exponents on GFX6
- radv: limit the Vulkan version to 1.1 for Android

Tapani Pälli (1):

- st/mesa: destroy only own program variants when program is released

4.16 Mesa 20.0.6 Release Notes / 2020-04-29

Mesa 20.0.6 is a bug fix release which fixes bugs found since the 20.0.5 release.

Mesa 20.0.6 implements the OpenGL 4.6 API, but the version reported by glGetString(GL_VERSION) or glGetIntegerv(GL_MAJOR_VERSION) / glGetIntegerv(GL_MINOR_VERSION) depends on the particular driver being used.

Some drivers don't support all the features required in OpenGL 4.6. OpenGL 4.6 is **only** available if requested at context creation. Compatibility contexts may report a lower version depending on each driver.

Mesa 20.0.6 implements the Vulkan 1.2 API, but the version reported by the apiVersion property of the VkPhysicalDeviceProperties struct depends on the particular driver being used.

4.16.1 SHA256 checksum

```
30b5d8e9201a01a0e88e18bb79850e67b1d28443b34c4c5cacad4bd10f668b96  mesa-20.0.6.tar.xz
```

4.16.2 New features

4.16.3 Bug fixes

- dEQP-VK.subgroups.size_control.compute.* crashes on HSW and TGL
- piglit spec.!opengl 1.0.gl-1.0-fpexceptions crash on Iris
- SPIR-V: OpConvertUToPtr from spec constant fails to compile
- radv: Reading ViewportIndex in fragment shader returns garbage
- radeonsi: GL_LINES rendering is affected by GL_POINT_SPRITE
- [ANV] gfxbench Aztec Ruins misrenders on gen11+
- glxinfo cmd crashed

4.16.4 Changes

Abhishek Kumar (1):

- anv/android: fix assert in anv_import_ahw_memory

Bas Nieuwenhuizen (1):

- radv: Use actual memory type count for setting app-visible bitset.

Danylo Piliaiev (3):

- st/mesa: Re-assign vs in locations after updating nir info for ffvp/ARB_vp
- spirv: Expand workaround for OpControlBarrier on old GLSLang
- st/mesa: Treat vertex inputs absent in inputMapping as zero in mesa_to_tgsi

Dylan Baker (9):

- docs: Add sha256 sums for 20.0.5
- .pick_status.json: Update to c552b5fd1d106adc04f62fcbe71d650a9a17f7e0
- meson: update llvm dependency logic for meson 0.54.0
- .pick_status.json: Mark 0123b8f63415d3d320929e6112da2be2d837b262 as denominated
- .pick_status.json: Update to 51c1c4d95a05b6eb6fce74e8d624615e4a1b38ab
- .pick_status.json: Update to 51c1c4d95a05b6eb6fce74e8d624615e4a1b38ab
- .pick_status.json: Update to efdb7fa9a83b0a216b1837a5912b71669bf3f984

- .pick_status.json: Update to 42b1696ef627a5bfee29911a780fa0a4dbf04610
- .pick_status.json: Update to 6b551d9f360e45ba4e74867dbe79ae212e4766c5

Eric Anholt (1):

- freedreno: Fix calculation of the const buffer cmdstream size.

Erik Faye-Lund (2):

- mesa/gallium: do not use enum for bit-allocated member
- meson: correct windows-version define

Jason Ekstrand (12):

- anv: Move vb_emit setup closer to where it's used in flush_state
- anv: Apply any needed PIPE_CONTROLs before emitting state
- spirv: Allow constants and NULLs in SpvOpConvertUToPtr
- anv: Properly handle all sizes of specialization constants
- radv: Properly handle all sizes of specialization constants
- turnip: Properly handle all sizes of specialization constants
- nir/opt_deref: Remove certain sampler type casts
- spirv: Fix passing combined image/samplers through function calls
- anv: Drop an assert
- nir/lower_subgroups: Mask off unused bits in ballot ops
- intel/devinfo: Compute the correct L3$ size for Gen12
- anv: Expose CS workgroup sizes based on a maximum of 64 threads

Joshua Ashton (1):

- radv: Use TRUNC_COORD on samplers

Lionel Landwerlin (5):

- iris: fail screen creation when kernel support is not there
- intel/perf: move register definition to special file
- intel/perf: break GL query stuff away
- intel/perf: move mdapi query definitions to their own file
- intel/perf: Enable MDAPI queries for Gen12

Pierre-Eric Pelloux-Prayer (1):

- radeonsi: skip vs output optimizations for some outputs

Quentin Glidic (1):

- meson: Use dependency.partial_dependency()

Samuel Pitoiset (1):

- radv: make sure to export the viewport index if FS needs it

4.17 Mesa 20.0.5 Release Notes / 2020-04-22

Mesa 20.0.5 is a bug fix release which fixes bugs found since the 20.0.4 release.

Mesa 20.0.5 implements the OpenGL 4.6 API, but the version reported by glGetString(GL_VERSION) or glGetIntegerv(GL_MAJOR_VERSION) / glGetIntegerv(GL_MINOR_VERSION) depends on the particular driver being used. Some drivers don't support all the features required in OpenGL 4.6. OpenGL 4.6 is **only** available if requested at context creation. Compatibility contexts may report a lower version depending on each driver.

Mesa 20.0.5 implements the Vulkan 1.2 API, but the version reported by the apiVersion property of the VkPhysicalDeviceProperties struct depends on the particular driver being used.

4.17.1 SHA256 checksum

```
2c56a82a28cc924e40ea49752abdf1d701c9952481f53cbc7a080271597f572e  mesa-20.0.5.tar.xz
```

4.17.2 New features

4.17.3 Bug fixes

- nir: nir_lower_returns can't handle nested loops
- Graphic artifacts with Mesa 20.0.4 on intel HD 510 GPU
- Mesa 20 regression makes Lightsprint demos crash
- Build Fails with Clang Shared Library
- dri_common.h:58:8: error: unknown type name '__GLXDRIdrawable'
- Graphical glitches on Intel Graphics when Xorg started on Iris driver
- SIGSEGV src/compiler/glsl/ast_function.cpp:53
- manywin aborts with "i965: Failed to submit batchbuffer: Invalid argument"
- manywin aborts with "i965: Failed to submit batchbuffer: Invalid argument"
- manywin aborts with "i965: Failed to submit batchbuffer: Invalid argument"
- manywin aborts with "i965: Failed to submit batchbuffer: Invalid argument"
- v3d: transform feedback issue
- radv: dEQP-VK.binding_model.descriptorset_random.sets4.noarray.ubolimitlow.sbolimitlow.imglimitlow.noiub.comp.noia.0 segfault
- radv: RAVEN fails dEQP-VK.pipeline.timestamp.misc_tests.reset_query_before_copy
- https://gitlab.freedesktop.org/mesa/mesa/-/issues/2727
- enable storageBuffer16BitAccess feature in radv for SI and CIK
- Weston drm-backend.so seems to fail with Mesa master and LIBGL_ALWAYS_SOFTWARE=1
- vaapi bob deinterlacer produces wrong output height on AMD

4.17.4 Changes

Arcady Goldmints-Orlov (1):

- nir: Lower returns correctly inside nested loops

Bas Nieuwenhuizen (3):

- radv: Store 64-bit availability bools if requested.
- radv: Consider maximum sample distances for entire grid.
- radv: Use correct buffer count with variable descriptor set sizes.

D Scott Phillips (1):

- util/sparse_array: don't stomp head's counter on pop operations

Daniel Stone (1):

- EGL: Add eglSetDamageRegionKHR to GLVND dispatch list

Danylo Piliaiev (1):

- st/mesa: Update shader info of ffvp/ARB_vp after translation to NIR

Dave Airlie (2):

- draw: free the NIR IR.
- llvmpipe/nir: free the nir shader

Dylan Baker (6):

- .pick_status.json: Update to 089e1fb287eb9b70c191091128ed5ba7edd2960a
- .pick_status.json: Update to 65e2eaa4d3a7095ac438fafb09d1e36a4210966e
- .pick_status.json: Update to 28d36d26c2212276e1238fad8f0b12caab97fee8
- .pick_status.json: Update to acf7e73be54c7f1cc52fcc9be38a9df26849200e
- .pick_status.json: Update to 13ce637f1b28381e72470763ff5e39dd3c562476
- .pick_status.json: Update to c3c1f4d6bcc210408f8b180727d269838b38193b

Emil Velikov (4):

- glx: set the loader_logger early and for everyone
- egl/drm: reinstate (kms_)swrast support
- Revert "egl/dri2: Don't dlclose() the driver on dri2_load_driver_common failure"
- glx: omit loader_loader() for macOS

Eric Anholt (1):

- ci: Remove LLVM from ARM test drivers.

Eric Engestrom (1):

- docs/relnotes: add sha256sum for 20.0.4

Hyunjun Ko (1):

- nir: fix wrong assignment to buffer in xfb_varyings_info

Ilia Mirkin (1):

- nv50: don't try to upload MSAA settings for BUFFER textures

Jason Ekstrand (5):

- anv/image: Use align_u64 for image offsets
- nir/load_store_vectorize: Fix shared atomic info
- spirv: Handle OOB vector extract operations
- intel: Add _const versions of prog_data cast helpers
- anv: Report correct SLM size

Jose Maria Casanova Crespo (1):

- v3d: Primitive Counts Feedback needs an extra 32-bit padding.

Juan A. Suarez Romero (2):

- intel/compiler: store the FS inputs in WM prog data
- anv/pipeline: allow more than 16 FS inputs

Karol Herbst (2):

- clover: fix build with single library clang build
- Revert "nvc0: fix line width on GM20x+"

Lionel Landwerlin (7):

- iris: properly free resources on BO allocation failure
- iris: share buffer managers accross screens
- iris: make resources take a ref on the screen object
- i965: store DRM fd on intel_screen
- i965: share buffer managers across screens
- iris: drop cache coherent cpu mapping for external BO
- util/sparse_free_list: manipulate node pointers using atomic primitives

Marek Olšák (1):

- st/mesa: fix a crash due to passing a draw vertex shader into the driver

Mathias Fröhlich (1):

- i965: Move down genX_upload_sbe in profiles.

Matt Turner (1):

- meson: Specify the maximum required libdrm in dri.pc

Neil Armstrong (3):

- gitlab-ci/lava: fix handling of lava tags
- gitlab-ci: add FILES_HOST_URL and move FILES_HOST_NAME into jobs
- gitlab-ci: re-enable mali400/450 and t820 jobs

Rhys Perry (1):

- aco: fix 1D textureGrad() on GFX9

Rob Clark (1):

- nir: fix definition of imadsh_mix16 for vectors

Rohan Garg (1):

- ci: Split out radv build-testing on arm64

Samuel Pitoiset (9):

- ac/nir: split 8-bit load/store to global memory on GFX6
- ac/nir: split 8-bit SSBO stores on GFX6
- radv/llvm: enable 8-bit storage features on GFX6-GFX7
- ac/nir: split 16-bit load/store to global memory on GFX6
- ac/nir: split 16-bit SSBO stores on GFX6
- radv/llvm: enable 16-bit storage features on GFX6-GFX7
- radv: do not abort with unknown/unimplemented descriptor types
- radv/llvm: fix exporting the viewport index if the fragment shader needs it
- aco: fix exporting the viewport index if the fragment shader needs it

Tapani Pälli (4):

- mesa/st: unbind shader state before deleting it
- mesa/st: release variants for active programs before unref
- glsl: stop processing function parameters if error happened
- mesa/st: initialize all winsys_handle fields for memory objects

Thong Thai (1):

- gallium/auxiliary/vl: fix bob compute shaders for deint yuv

Timothy Arceri (1):

- radeonsi: don't lower constant arrays to uniforms in GLSL IR

Tobias Jakobi (1):

- meson: Link Gallium Nine with ld_args_build_id

Tomeu Vizoso (2):

- gitlab-ci: Place files from the Mesa repo into the build tarball
- gitlab-ci: Serve files for LAVA via separate service

Vinson Lee (2):

- swr/rasterizer: Use private functions for min/max to avoid namespace issues.
- swr: Remove Byte Order Mark.

pal1000 (1):

- scons/windows: Support build with LLVM 10.

4.18 Mesa 20.0.4 Release Notes / 2020-04-03

Mesa 20.0.4 is an emergency release which reverts a serious SPIR-V regression in the 20.0.3 release.

Mesa 20.0.4 implements the OpenGL 4.6 API, but the version reported by glGetString(GL_VERSION) or glGetIntegerv(GL_MAJOR_VERSION) / glGetIntegerv(GL_MINOR_VERSION) depends on the particular driver being used.

Some drivers don't support all the features required in OpenGL 4.6. OpenGL 4.6 is **only** available if requested at context creation. Compatibility contexts may report a lower version depending on each driver.

Mesa 20.0.4 implements the Vulkan 1.2 API, but the version reported by the apiVersion property of the VkPhysicalDeviceProperties struct depends on the particular driver being used.

4.18.1 SHA256 checksum

```
c4ed491517a94118a7a611810eeb92645d42ffd82280dcd51be8cc2ba1aabba5  mesa-20.0.4.tar.xz
```

4.18.2 New features

4.18.3 Bug fixes

4.18.4 Changes

Eric Engestrom (2):

- docs/relnotes: add sha256sum for 20.0.3
- .pick_status.json: Update to c71c1f44b055c680f073a2608a3bf560b55f8974

Jason Ekstrand (1):

- Revert "spirv: Implement OpCopyObject and OpCopyLogical as blind copies"

4.19 Mesa 20.0.3 Release Notes / 2020-04-01

Mesa 20.0.3 is a bug fix release which fixes bugs found since the 20.0.2 release.

Mesa 20.0.3 implements the OpenGL 4.6 API, but the version reported by glGetString(GL_VERSION) or glGetIntegerv(GL_MAJOR_VERSION) / glGetIntegerv(GL_MINOR_VERSION) depends on the particular driver being used. Some drivers don't support all the features required in OpenGL 4.6. OpenGL 4.6 is **only** available if requested at context creation. Compatibility contexts may report a lower version depending on each driver.

Mesa 20.0.3 implements the Vulkan 1.2 API, but the version reported by the apiVersion property of the VkPhysicalDeviceProperties struct depends on the particular driver being used.

4.19.1 SHA256 checksum

```
d63aaf2c27143eded2f4f376f18f7a766ad997f8eeb96c357e8ade84e8a237af  mesa-20.0.3.tar.xz
```

4.19.2 New features

4.19.3 Bug fixes

- RADV: flickering textures in Q.U.B.E. 2 through Proton
- src/compiler/glsl/ast_to_hir.cpp:2134: ir_rvalue* ast_expression::do_hir(exec_list*, _mesa_glsl_parse_state*, bool): Assertion 'result != NULL || !needs_rvalue' failed.

- [ACO] Reliable crash with RPCS3 that is not present with LLVM
- [RADV] vkCmdBindTransformFeedbackBuffersEXT pSizes optional parameter not handled
- soft-fp64: __fsat64 incorrectly returns NaN for a NaN input. It should return zero.
- Hang when using glWaitSync with multithreaded shared GL contexts

4.19.4 Changes

Caio Marcelo de Oliveira Filho (1):

- mesa/main: Fix overflow in validation of DispatchComputeGroupSizeARB

Dylan Baker (6):

- docs/relnotes: Add sha256 sums for 20.0.2
- .pick_status.json: Update to cf62c2b2ac69637785f55b790fdd601c17e7e9d5
- .pick_status.json: Mark 672d10619980687acec329742f055f7f3796c1b8 as backported
- .pick_status.json: Mark c923de68dd0ab10a5a5fb3196f539707d046d897 as backported
- .pick_status.json: Mark 56de6f698e3f164d97f132203e8159ef0b8e9bb8 as denominated
- .pick_status.json: Update to aee004a7c8900938d1c17f0ac299d40001b383b0

Eric Engestrom (6):

- .pick_status.json: Update to 3252041a7872c49e53bb02ffe8b079b5fc43f15e
- .pick_status.json: Update to 12711939320e4fcd3a0d86af22da1042ad92035f
- .pick_status.json: Update to 05069e1f0794aadd40ce9269f858e50c64254388
- .pick_status.json: Update to 8970b7839aebefa7207c9535ac34ab4e8cc0ae25
- .pick_status.json: Update to 5f4d9b419a1c931ad468b8b22b8a95b1216891e4
- .pick_status.json: Update to 70ac7f5b0c46370075a35067c9f7dfe78e84b16d

Erik Faye-Lund (3):

- rbug: do not return void-value
- pipebuffer: clean up cast-warnings
- vtn/opencl: fully enable OpenCLstd_Clz

Francisco Jerez (1):

- intel/fs/gen12: Fix interaction of SWSB dependency combination with EU fusion workaround.

Greg V (1):

- amd/addrlib: fix build on non-x86 platforms

Ian Romanick (2):

- soft-fp64/fsat: Correctly handle NaN
- soft-fp64: Split a block that was missing a cast on a comparison

Jason Ekstrand (5):

- intel/blorp: Add support for swizzling fast-clear colors
- anv: Swizzle fast-clear values

- nir/lower_int64: Lower 8 and 16-bit downcasts with nir_lower_mov64
- anv: Account for the header in anv_state_stream_alloc
- spirv: Implement OpCopyObject and OpCopyLogical as blind copies

John Stultz (2):

- gallium: hud_context: Fix scalar initializer warning.
- vc4_bufmgr: Remove duplicative VC definition

Jordan Justen (2):

- intel: Update TGL PCI strings
- intel: Add TGL PCI ID

Lionel Landwerlin (5):

- isl: implement linear tiling row pitch requirement for display
- isl: properly filter supported display modifiers on Gen9+
- isl: only apply main surface ccs pitch constraint with CCS
- isl: drop min row pitch alignment when set by the driver
- intel: add new TGL pci ids

Marek Olšák (3):

- nir: fix clip/cull_distance_array_size in nir_lower_clip_cull_distance_arrays
- ac: fix fast division
- st/mesa: fix use of uninitialized memory due to st_nir_lower_builtin

Marek Vasut (1):

- etnaviv: Emit PE.ALPHA_COLOR_EXT* on GPUs with half-float support

Neil Armstrong (1):

- Revert "ci: Remove T820 from CI temporarily"

Pierre-Eric Pelloux-Prayer (1):

- st/mesa: disallow deferred flush if there are multiple contexts

Rhys Perry (11):

- nir/gather_info: handle emit_vertex_with_counter
- aco: set has_divergent_branch for discards in loops
- aco: handle missing second predecessors at merge block phis
- aco: skip NIR in unreachable merge blocks
- aco: improve check for unreachable loop continue blocks
- aco: emit IR in IF's merge block instead if the other side ends in a jump
- aco: fix boolean undef regclass
- nir/gather_info: fix per-vertex handling in try_mask_partial_io
- aco: implement 64-bit VGPR constant copies in handle_operands()
- glsl: fix race in instance getters

- util/u_queue: fix race in total_jobs_size access

Rob Clark (2):

- freedreno/ir3/ra: fix array liveranges
- util: fix u_fifo_pop()

Samuel Pitoiset (7):

- radv/gfx10: fix required subgroup size with VK_EXT_subgroup_size_control
- radv/gfx10: fix required ballot size with VK_EXT_subgroup_size_control
- radv: fix optional pSizes parameter when binding streamout buffers
- radv: enable VK_KHR_8bit_storage on GFX6-GFX7
- ac/nir: use llvm.amdgcn.rcp for nir_op_frcp
- ac/nir: use llvm.amdgcn.rsq for nir_op_frsq
- ac/nir: use llvm.amdgcn.rcp in ac_build_fdiv()

Tapani Pälli (1):

- glsl: set error_emitted true if type not ok for assignment

Thomas Hellstrom (1):

- svga, winsys/svga: Fix persistent memory discard maps

Timothy Arceri (3):

- glsl: fix varying packing for 64bit integers
- nir: fix packing of TCS varyings not read by the TES
- nir: fix crash in varying packing on interface mismatch

Timur Kristóf (1):

- radv/llvm: fix subgroup shuffle for chips without bpermute

4.20 Mesa 20.0.2 Release Notes / 2020-03-18

Mesa 20.0.2 is a bug fix release which fixes bugs found since the 20.0.1 release.

Mesa 20.0.2 implements the OpenGL 4.6 API, but the version reported by glGetString(GL_VERSION) or glGetIntegerv(GL_MAJOR_VERSION) / glGetIntegerv(GL_MINOR_VERSION) depends on the particular driver being used. Some drivers don't support all the features required in OpenGL 4.6. OpenGL 4.6 is **only** available if requested at context creation. Compatibility contexts may report a lower version depending on each driver.

Mesa 20.0.2 implements the Vulkan 1.2 API, but the version reported by the apiVersion property of the VkPhysicalDeviceProperties struct depends on the particular driver being used.

4.20.1 SHA256 checksum

```
aa54f1cb669550606aab8ceb475105d15aeb814fca5a778ce70d0fd10e98e86f  mesa-20.0.2.tar.xz
```

4.20.2 New features

4.20.3 Bug fixes

- RPCS3 / Persona 5 - Performance regression [RADV / Navi]
- [CTS] dEQP-VK.descriptor_indexing.* fails on RADV/LLVM
- [RadeonSI][gfx10/navi] Kerbal Space Program crash: si_draw_vbo: Assertion '0' failed
- src/compiler/glsl/glcpp/glcpp-parse.y:1297: _token_print: Assertion '!"Error: Don't know how to print token."' failed.
- Budget Cuts hits VK_AMD_shader_fragment_mask assert

4.20.4 Changes

Andreas Baierl (1):

- gitlab-ci: Add add a set of lima flakes

Bas Nieuwenhuizen (2):

- amd/llvm: Fix divergent descriptor indexing. (v3)
- amd/llvm: Fix divergent descriptor regressions with radeonsi.

Danylo Piliaiev (2):

- glsl: do not crash if string literal is used outside of #include/#line
- st/mesa: Fix signed integer overflow when using util_throttle_memory_usage

Dave Airlie (1):

- gallium: fix build with latest meson and gcc10

Dylan Baker (8):

- docs: Add sha256sums for 20.0.1
- .pick_status.json: Update to 07f1ef5656e0721282d01a8421eaca056348137d
- .pick_status.json: Update to 70341d7746c177a4cd7377ef633e9f85afd11d54
- .pick_status.json: Update to 625d8705f02e211e2733c3fe12845505725c37d4
- .pick_status.json: Mark b83c9aca4a5fd02d920c90c1799137fed52dc1d9 as backported
- .pick_status.json: Update to ee9e0d1ecae307fa48200d2604d3114070253299
- .pick_status.json: Update to 3dd0d12aa5fefa94123269a541c94cdf57599e34
- .pick_status.json: Update to 94e37859a96cc56cf0c5418a5af00a3e9f5a1bf5

Eric Anholt (1):

- glsl/tests: Fix waiting for disk_cache_put() to finish.

Eric Engestrom (7):

- bin/gen_release_notes.py: fix commit list command
- .pick_status.json: Update to 24db276d11976905b2e8a44965c684bb48c3d49f
- gen_release_notes: fix vulkan version reported

- docs/relnotes/20.0: fix vulkan version reported
- .pick_status.json: Update to ba03e308b66b0b88f60b99d9d47851a5e1522e6e
- vulkan/wsi: fix cleanup when dup() fails
- gen_release_notes: fix version in "you should wait" message

Francisco Jerez (1):

- intel/fs: Fix workaround for VxH indirect addressing bug under control flow.

Jason Ekstrand (9):

- isl: Set 3DSTATE_DEPTH_BUFFER::Depth correctly for 3D surfaces
- iris: Don't skip fast depth clears if the color changed
- anv: Parse VkPhysicalDeviceFeatures2 in CreateDevice
- vulkan/wsi: Don't leak the FD when GetImageDrmFormatModifierProperties fails
- vulkan/wsi: Return an error if dup() fails
- anv: Use the PIPE_CONTROL instead of bits for the CS stall W/A
- anv: Use a proper end-of-pipe sync instead of just CS stall
- anv: Do end-of-pipe sync around MCS/CCS ops instead of CS stall
- anv: Do an end-of-pipe sync before updating AUX table entries

José Fonseca (1):

- meson: Avoid duplicate symbols.

Kristian Høgsberg (2):

- Revert "glsl: Use a simpler formula for tanh"
- Revert "spirv: Use a simpler and more correct implementaiton of tanh()"

Marek Olšák (4):

- Revert "mesa: check for z=0 in _mesa_Vertex3dv()"
- radeonsi: add a bug workaround for NGG - LATE_ALLOC_GS
- ac: add a bug workaround for the 100% NGG culling case
- gallium/cso_context: remove cso_delete_xxx_shader helpers to fix the live cache

Martin Fuzzey (3):

- freedreno: android: fix build failure on android due to python version
- freedreno: android: add a6xx-pack.xml.h generation to android build
- freedreno: android: fix build of perfcounters.

Michel Dänzer (1):

- llvmpipe: Use uintptr_t for pointer values

Rafael Antognolli (3):

- anv: Wait for the GPU to be idle before invalidating the aux table.
- iris: Split aux map initialization from invalidation.
- iris: Wait for the GPU to be idle before invalidating the aux table.

Rob Clark (1):

- freedreno: fix FD_MESA_DEBUG=inorder

Samuel Pitoiset (5):

- aco: fix image load/store with lod and 1D images
- nir/lower_input_attachments: remove bogus assert in try_lower_input_texop()
- ac/llvm: add missing optimization barrier for 64-bit readlanes
- radv: only inject implicit subpass dependencies if necessary
- radv: fix random depth range unrestricted failures due to a cache issue

Timur Kristóf (2):

- nir: Add ability to lower non-const quad broadcasts to const ones.
- radv: Enable lowering dynamic quad broadcasts.

Vinson Lee (1):

- st/nine: Fix incompatible-pointer-types-discards-qualifiers errors.

4.21 Mesa 19.3.5 Release Notes / 2020-03-09

Mesa 19.3.5 is a bug fix release which fixes bugs found since the 19.3.4 release.

Mesa 19.3.5 implements the OpenGL 4.6 API, but the version reported by glGetString(GL_VERSION) or glGetIntegerv(GL_MAJOR_VERSION) / glGetIntegerv(GL_MINOR_VERSION) depends on the particular driver being used. Some drivers don't support all the features required in OpenGL 4.6. OpenGL 4.6 is **only** available if requested at context creation. Compatibility contexts may report a lower version depending on each driver.

Mesa 19.3.5 implements the Vulkan 1.1 API, but the version reported by the apiVersion property of the VkPhysicalDeviceProperties struct depends on the particular driver being used.

4.21.1 SHA256 checksum

```
009895b055b971c82d7a0cd57227d414d323282605946e94e9b308a9cb49c6b6  mesa-19.3.5.tar.xz
```

4.21.2 New features

- None

4.21.3 Bug fixes

- V3D/Broadcom (Raspberry Pi 4) - GLES 3.1 - GL_EXT_texture_norm16 advertised, but not usable
- i965 assertion failure in fallback_rgbx_to_rgba
- drisw crashes on calling NULL putImage on EGL surfaceless platform (pbuffer EGLSurface)

4.21.4 Changes

Andrii Simiklit (1):

- Revert "glx: convert glx_config_create_list to one big calloc"

Arcady Goldmints-Orlov (1):

- spirv: Remove outdated SPIR-V decoration warnings

Caio Marcelo de Oliveira Filho (1):

- intel/gen12: Take into account opcode when decoding SWSB

Danylo Piliaiev (1):

- i965: Do not generate D16 B5G6R5_UNORM configs on gen < 8

Dave Airlie (1):

- gallivm/tgsi: fix stream id regression

Dylan Baker (7):

- docs: Add SHA256 sum for 19.3.4
- .pick_status.json: Update to 2a98cf3b2ecea43cea148df7f77d2abadfd1c9db
- .pick_status.json: Update to 946eacbafb47c8b94d47e7c9d2a8b02fff5a22fa
- .pick_status.json: Update to bee5c9b0dc13dbae0ccf124124eaccebf7f2a435
- .pick_status.json: Update to 8291d728dc997e87b4d2e4e451692643a1dba881
- .pick_status.json: Update to e4baff90812d799d586296fcad992ddcc553c359
- .pick_status.json: Update to 01496e3d1ea0370af03e6645dbd2b864c2ace94c

Eric Engestrom (10):

- .pick_status.json: Update to 74e4cda64b9d114321216eefe536f80644b0f0fd
- .pick_status.json: Mark dba71de5c63617677fe44558f995d35fad643413 as denominated
- .pick_status.json: Mark 5ea23ba659adc05ff75ca7a4c9d1bd01db889ddd as denominated
- .pick_status.json: Mark 34fd894e42ae1ec9d35bf9c4f05364b03dd4a223 as denominated
- .pick_status.json: Mark ddd767387f336ed1578f171a2af4ca33c564d7f3 as denominated
- .pick_status.json: Mark b9773631d3e79e2310ed0eb274b4dd9426205066 as denominated
- .pick_status.json: Mark 9fea90ad5170dd64376d22a14ac88c392813c96c as denominated
- bin/gen_release_notes.py: fix commit list command
- .pick_status.json: Update to 0103f02acb10dcdea23461ba214307a6827a7772
- gitlab-ci: update template to fix container build issues

Erik Faye-Lund (2):

- util: promote u_debug_memory.c to src/util
- .pick_status.json: Update to 74e4cda64b9d114321216eefe536f80644b0f0fd

Francisco Jerez (1):

- intel/fs/gen12: Fixup/simplify SWSB annotations of SIMD32 scratch writes.

Ian Romanick (1):

- intel/fs: Correctly handle multiply of fsign with a source modifier

Jason Ekstrand (3):

- isl: Set 3DSTATE_DEPTH_BUFFER::Depth correctly for 3D surfaces
- iris: Don't skip fast depth clears if the color changed
- anv: Parse VkPhysicalDeviceFeatures2 in CreateDevice

Jordan Justen (1):

- intel/compiler: Restrict cs_threads to 64

Jose Maria Casanova Crespo (1):

- v3d: Sync on last CS when non-compute stage uses resource written by CS

Kristian Høgsberg (2):

- Revert "glsl: Use a simpler formula for tanh"
- Revert "spirv: Use a simpler and more correct implementaiton of tanh()"

Krzysztof Raszkowski (1):

- gallium/swr: simplify environmental variabled expansion code

Marek Olšák (3):

- radeonsi: don't wait for shader compilation to finish when destroying a context
- mesa: fix immediate mode with tessellation and varying patch vertices
- Revert "mesa: check for z=0 in _mesa_Vertex3dv()"

Mathias Fröhlich (3):

- egl: Implement getImage/putImage on pbuffer swrast.
- egl: Fix A2RGB10 platform_{device,surfaceless} PBuffer configs.
- mesa: Flush vertices before changing the OpenGL state.

Michel Dänzer (1):

- st/vdpau: Only call is_video_format_supported hook if needed

Paulo Zanoni (1):

- intel/device: bdw_gt1 actually has 6 eus per subslice

Peng Huang (1):

- radeonsi: make si_fence_server_signal flush pipe without work

Rafael Antognolli (1):

- intel/gen12+: Disable mid thread preemption.

Samuel Pitoiset (3):

- ac/llvm: fix 64-bit fmed3
- ac/llvm: fix 16-bit fmed3 on GFX8 and older gens
- ac/llvm: flush denorms for nir_op_fmed3 on GFX8 and older gens

Tapani Pälli (5):

- iris: fix aux buf map failure in 32bits app on Android

- mesa: introduce boolean toggle for EXT_texture_norm16
- i965: toggle on EXT_texture_norm16
- mesa/st: toggle EXT_texture_norm16 based on format support
- mesa/st: fix formats required for EXT_texture_norm16

Timothy Arceri (1):

- glsl: fix gl_nir_set_uniform_initializers() for image arrays

luc (1):

- zink: confused compilation macro usage for zink in target helpers.

4.22 Mesa 20.0.1 Release Notes / 2020-03-05

Mesa 20.0.1 is a bug fix release which fixes bugs found since the 20.0.0 release.

Mesa 20.0.1 implements the OpenGL 4.6 API, but the version reported by glGetString(GL_VERSION) or glGetIntegerv(GL_MAJOR_VERSION) / glGetIntegerv(GL_MINOR_VERSION) depends on the particular driver being used. Some drivers don't support all the features required in OpenGL 4.6. OpenGL 4.6 is **only** available if requested at context creation. Compatibility contexts may report a lower version depending on each driver.

Mesa 20.0.1 implements the Vulkan 1.2 API, but the version reported by the apiVersion property of the VkPhysicalDeviceProperties struct depends on the particular driver being used.

4.22.1 SHA256 checksum

```
6153ba3f8cb0524bbfc08e4db76b408126b2d1be8f789dffe28d1a0461eedde4  mesa-20.0.1.tar.xz
```

4.22.2 New features

4.22.3 Bug fixes

- V3D/Broadcom (Raspberry Pi 4) - GLES 3.1 - GL_EXT_texture_norm16 advertised, but not usable
- i965 assertion failure in fallback_rgbx_to_rgba
- Compute copies do not handle SUBSAMPLED formats

4.22.4 Changes

Andreas Baierl (1):

- gitlab-ci: lima: Add flaky tests to the skips list

Andrii Simiklit (1):

- Revert "glx: convert glx_config_create_list to one big calloc"

Arcady Goldmints-Orlov (1):

- spirv: Remove outdated SPIR-V decoration warnings

Bas Nieuwenhuizen (1):

- radeonsi: Fix compute copies for subsampled formats.

Caio Marcelo de Oliveira Filho (1):

- intel/gen12: Take into account opcode when decoding SWSB

Chris Wilson (1):

- iris: Fix import sync-file into syncobj

Danylo Piliaiev (1):

- i965: Do not generate D16 B5G6R5_UNORM configs on gen < 8

Dave Airlie (7):

- dri: add another get shm variant.
- glx/drisw: add getImageShm2 path
- glx/drisw: return false if shmid == -1
- glx/drisw: fix shm put image fallback
- gallivm/tgsi: fix stream id regression
- gallivm/nir: fix integer divide SIGFPE
- gallivm/nir: handle mod 0 better.

Dylan Baker (7):

- docs: Add release notes for 20.0.0
- .pick_status.json: Update to 8291d728dc997e87b4d2e4e451692643a1dba881
- .pick_status.json: Update to e4baff90812d799d586296fcad992ddcc553c359
- .pick_status.json: Update to 01496e3d1ea0370af03e6645dbd2b864c2ace94c
- .pick_status.json: Update to 09323634898ab3efc0150dc7d756bf36b1b89b76
- .pick_status.json: Update to 3503cb4c28e01b34f3a25546c058150709c22348
- .pick_status.json: Update to 0ac731b1ff96de46998948aa06081efa5140d50e

Eric Anholt (3):

- llvmpipe: Fix real uninitialized use of “atype” for SEMANTIC_FACE
- turnip: Fix compiler warning about casting a nondispatchable handle.
- aco: Fix signed-vs-unsigned warning.

Erik Faye-Lund (1):

- util: promote u_debug_memory.c to src/util

Ian Romanick (2):

- nir/search: Use larger type to hold linearized index
- intel/fs: Correctly handle multiply of fsign with a source modifier

James Xiong (1):

- iris: handle the failure of converting unsupported yuv formats to isl

Jason Ekstrand (1):

- anv: Always enable the data cache

Jonathan Marek (1):

- turnip: fix srgb MRT

Jordan Justen (1):

- intel/compiler: Restrict cs_threads to 64

Jose Maria Casanova Crespo (1):

- v3d: Sync on last CS when non-compute stage uses resource written by CS

Kenneth Graunke (2):

- iris: Make mocs an inline helper in iris_resource.h
- iris: Fix BLORP vertex buffers to respect ISL MOCS settings

Marek Olšák (5):

- mesa: fix immediate mode with tessellation and varying patch vertices
- util: remove the dependency on kcmp.h
- tgsi_to_nir: set num_images and num_samplers with holes correctly
- mesa: call FLUSH_VERTICES before updating CoordReplace
- mesa: fix incorrect prim.begin/end for glMultiDrawElements

Mathias Fröhlich (2):

- egl: Fix A2RGB10 platform_{device,surfaceless} PBuffer configs.
- mesa: Flush vertices before changing the OpenGL state.

Michel Dänzer (1):

- st/vdpau: Only call is_video_format_supported hook if needed

Paulo Zanoni (3):

- intel: fix the gen 11 compute shader scratch IDs
- intel: fix the gen 12 compute shader scratch IDs
- intel/device: bdw_gt1 actually has 6 eus per subslice

Rafael Antognolli (2):

- iris: Apply the flushes when switching pipelines.
- intel/gen12+: Disable mid thread preemption.

Rhys Perry (2):

- aco: keep track of which events are used in a barrier
- aco: fix carry-out size for wave32 v_add_co_u32_e64

Samuel Pitoiset (3):

- ac/llvm: fix 64-bit fmed3
- ac/llvm: fix 16-bit fmed3 on GFX8 and older gens
- ac/llvm: flush denorms for nir_op_fmed3 on GFX8 and older gens

Tapani Pälli (4):

- mesa: introduce boolean toggle for EXT_texture_norm16

- i965: toggle on EXT_texture_norm16
- mesa/st: toggle EXT_texture_norm16 based on format support
- mesa/st: fix formats required for EXT_texture_norm16

4.23 Mesa 20.0.0 Release Notes / 2020-02-19

Mesa 20.0.0 is a new development release. People who are concerned with stability and reliability should stick with a previous release or wait for Mesa 20.0.1.

Mesa 20.0.0 implements the OpenGL 4.6 API, but the version reported by glGetString(GL_VERSION) or glGetIntegerv(GL_MAJOR_VERSION) / glGetIntegerv(GL_MINOR_VERSION) depends on the particular driver being used. Some drivers don't support all the features required in OpenGL 4.6. OpenGL 4.6 is **only** available if requested at context creation. Compatibility contexts may report a lower version depending on each driver.

Mesa 20.0.0 implements the Vulkan 1.2 API, but the version reported by the apiVersion property of the VkPhysicalDeviceProperties struct depends on the particular driver being used.

4.23.1 SHA256 checksum

```
bb6db3e54b608d2536d4000b3de7dd3ae115fc114e8acbb5afff4b3bbed04b34  mesa-20.0.0.tar.xz
```

4.23.2 New features

- OpenGL 4.6 on radeonsi.
- GL_ARB_gl_spirv on radeonsi.
- GL_ARB_spirv_extensions on radeonsi.
- GL_EXT_direct_state_access for compatibility profile.
- VK_AMD_device_coherent_memory on RADV.
- VK_AMD_mixed_attachment_samples on RADV.
- VK_AMD_shader_explicit_vertex_parameter on RADV.
- VK_AMD_shader_image_load_store_lod on RADV.
- VK_AMD_shader_fragment_mask on RADV.
- VK_EXT_subgroup_size_control on RADV/LLVM.
- VK_KHR_separate_depth_stencil_layouts on Intel, RADV.
- VK_KHR_shader_subgroup_extended_types on RADV.
- VK_KHR_swapchain_mutable_format on RADV.
- VK_KHR_shader_float_controls on RADV/ACO.
- GFX6 (Southern Islands) and GFX7 (Sea Islands) support on RADV/ACO.
- Wave32 support for GFX10 (Navi) on RADV/ACO.
- Compilation of Geometry Shaders on RADV/ACO.
- Vulkan 1.2 on Intel, RADV.

- GL_INTEL_shader_integer_functions2 and VK_INTEL_shader_integer_functions2 on Intel.

4.23.3 Bug fixes

- drisw crashes on calling NULL putImage on EGL surfaceless platform (pbuffer EGLSurface)
- [radeonsi][vaapi][bisected] invalid VASurfaceID when playing interlaced DVB stream in Kodi
- [RADV] GPU hangs while the cutscene plays in the game Assassin's Creed Origins
- ACO: The Elder Scrolls Online crashes on startup (Navi)
- Broken rendering of glxgears on S/390 architecture (64bit, BigEndian)
- aco: sun flickering with Assassins Creeds Origins
- !1896 broke ext_image_dma_buf_import piglit tests with radeonsi
- aco: wrong geometry with Assassins Creed Origins on GFX6
- valgrind errors since commit a8ec4082a41
- OSMesa osmesa_choose_format returns a format not supported by st_new_renderbuffer_fb
- Build error with VS on WIN
- Using EGL_KHR_surfaceless_context causes spurious "libEGL warning: FIXME: egl/x11 doesn't support front buffer rendering."
- !3460 broke texsubimage test with piglit on zink+anv
- The screen is black when using ACO
- [Regression] JavaFX unbounded VRAM+RAM usage
- radv: implement VK_AMD_shader_explicit_vertex_parameter
- Civilization VI crashes when loading game (AMD Vega Mobile)
- [radeonsi] X-Server crashes when trying to start Guild Wars 2 with the commits from !3421
- aco: implement GFX6 support
- Add support for VK_KHR_swapchain_mutable_format
- radv: The Surge 2 crashes in ac_get_elem_bits()
- [Regression] JavaFX unbounded VRAM+RAM usage
- Use the OpenCL dispatch defnitions from OpenCL_Headers
- [regression][ilk,g965,g45] various dEQP-GLES2.functional.shaders.* failures
- aco: Dead Rising 4 crashes in lower_to_hw_instr() on GFX6-GFX7
- libvulkan_radeon.so crash with 'free(): double free detected in tcache 2'
- Commit be08e6a causes crash in com.android.launcher3 (Launcher)
- anv: Regression causing issues for radv when there are no Intel devices
- Mesa no longer compiles with GCC 10
- [Navi/aco] Guild Wars 2 - ring gfx timeout with commit 3bca0af2
- [radv/aco] Regression is causing a soft crash in The Witcher 3
- [bisected] [radeonsi] GPU hangs/resets while playing interlaced content on Kodi with VAAPI

- [radeonsi] MSAA image not copied properly after image store through texture view
- T-Rex and Manhattan onscreen performance issue on Android
- VkSamplerCreateInfo compareEnable not respected
- VkSamplerCreateInfo compareEnable not respected
- Freedreno drm softpin driver implementation leaks memory
- [POLARIS10] VRAM leak involving glTexImage2D with non-NULL data argument
- [regression][bisected][ivb/byt] crucible test func.push-constants.basic.q0 causes gpu hang
- MR 3096 broke lots of piglit ext_framebuffer_object tests on Raven
- Rise of the Tomb Raider benchmark crash on Dell XPS 7390 2-in-1 w/ Iris Plus Graphics (Ice Lake 8x8 GT2)
- Raven Ridge (2400G): Resident Evil 2 crashes my machine
- Common practice of glGetActiveUniform leads to O(N^2) behavior in Mesa
- Rocket League ingame artifacts
- [radv] SteamVR direct mode no longer works
- [ANV] unused create parameters not properly ignored
- [Bisected] Mesa fails to start alacritty with the wayland backend (AMD Vega).
- [iris] piglit test clip-distance-vs-gs-out fails due to VUE map mismatch between VS <-> GS stages
- [radv] SteamVR direct mode no longer works
- Blocky corruption in The Surge 2
- radeonsi: Floating point exception on R9 270 gpu for a set of traces
- [RADV] [Navi] LOD artifacting in Halo - The Master Chief Collection (Halo Reach)
- [CTS] dEQP-VK.api.image_clearing.core.clear_color_image.2d.linear.single_layer.r32g32b32_* fail on GFX6-GFX8
- Vulkan: Please consider adding another sample count to sampledImageIntegerSampleCounts
- Navi10: Bitrate based encoding with VAAPI/RadeonSI unusable
- [RADV] create parameters not properly ignored
- [regression][bdw,gen9,hsw,icl][iris] gltcs failures on mesa=8172b1fa03f
- Bugs in RadeonSI VAAPI implementation
- [GFX10] Glitch rendering Custom Avatars in Beat Saber
- intel/fs: Check for 16-bit immediates in fs_visitor::lower_mul_dword_inst is too strict
- i965/iris: assert when destroy GL context with active query
- Visuals without alpha bits are not sRGB-capable
- swapchain throttling: wait for fence has 1ns timeout
- radeonsi: OpenGL app always produces page fault in gfxhub on Navi 10
- [regression] KHR-GLES31.core.geometry_shader.api.program_pipeline_vs_gs_capture fails for various drivers
- [CTS] dEQP-VK.spirv_assembly.instruction.spirv1p4.entrypoint.tess_con_pc_entry_point hangs on GFX10
- [RADV] SPIR-V warning when compiling shader using storage multisampled image array

- [RADV] The Dead Rising 4 is causing a GPU hang with LLVM backend
- macOS u_thread.h:156:4: error: implicit declaration of function 'pthread_getcpuclockid'
- [Wine / Vulkan] Doom 2016 Hangs on Main Menu
- NULL resource when playing VP9 video through VDPAU on RX 570
- radeonsi: mpv –vo=vaapi incorrect rendering on gfx9+
- [BSW/BDW] skia lcdblendmode & lcdoverlap test failure
- Create a way to prefer iris vs i965 via driconf
- [Bisected] i965: CS:GO crashes in emit_deref_copy_load_store with debug Mesa
- radv/aco Jedi Fallen Order hair rendering buggy
- Inaccurate information on https://www.mesa3d.org/repository.html about how to get git write access.
- [RADV] VK_KHR_timeline_semaphore balloons in runtime
- Shadow of Mordor has randomly dancing black shadows on Talion's face
- gen7 crucible failures func.push-constants.basic.q0 and func.shader-subgroup-vote.basic.q0
- GL_EXT_disjoint_timer_query failing with GL_INVALID_ENUM
- Unreal 4 Elemental and MatineeFightScene demos misrender
- gputest gimark has unwanted black liquorice flakes
- triangle strip clipping with GL_FIRST_VERTEX_CONVENTION causes wrong vertex's attribute to be broadcasted for flat interpolation
- [bisected][regression][g45,g965,ilk] piglit arb_fragment_program kil failures
- glcts crashes since the enablement of ARB_shading_language_include
- Android build broken
- ld.lld: error: duplicate symbol (mesa-19.3.0-rc1)
- Divinity: Original Sin Enhanced Edition(Native) crash on start
- HSW. Tropico 6 and SuperTuxKart have shadows flickering
- GL_EXT_disjoint_timer_query failing with GL_INVALID_ENUM
- glxgears segfaults on POWER / Xvnc
- [regression][bdw,gen9,icl][iris] piglit failures on mesa f9fd04aca15fd00889caa666ba38007268e67f5c
- Redundant builds of libmesa_classic and libmesa_gallium
- [IVB,BYT] [Regression] [Bisected] Core dump at launching arb_compute_shader/linker/bug-93840.shader_test
- Vulkan drivers need access to format utils of gallium
- Disabling lower_fragdata_array causes shader-db to crash for some drivers
- GL_EXT_disjoint_timer_query failing with GL_INVALID_ENUM
- Android build broken by commit 9020f51 "util/u_endian: Add error checks"
- radv secure compile feature breaks compilation of RADV on armhf EABI (19.3-rc1)
- radv_debug.c warnings when compiling on 32 bits : cast to pointer from integer of different size
- Meson: Mesa3D build failure with standalone Mingw-w64 multilib

- [regression][bisected] KHR46 VertexArrayAttribFormat has unexpectedly generated GL_INVALID_OPERATION
- textureSize(samplerExternalOES, int) missing in desktop mesa 19.1.7 implementation
- zink: implicly casting integers to pointers, warnings on 32-bit compile
- Objects leaving trails in Firefox with antialias and preserveDrawingBuffer in three.js WebGLRednerer with mesa 19.2

4.23.4 Changes

Aaron Watry (1):

- clover/llvm: fix build after llvm 10 commit 1dfede3122ee

Adam Jackson (1):

- drisw: Cache the depth of the X drawable

Afonso Bordado (4):

- pan/midgard: Optimize comparisions with similar operations
- pan/midgard: Move midgard_is_branch_unit to helpers
- pan/midgard: Optimize branches with inverted arguments
- pan/midgard: Fix midgard_compile.h includes

Alan Coopersmith (1):

- intel/perf: adapt to platforms like Solaris without d_type in struct dirent

Alejandro Piñeiro (4):

- v3d: adds an extra MOV for any sig.ld*
- mesa/main/util: moving gallium u_mm to util, remove main/mm
- nir/opt_peephole_select: remove unused variables
- turnip: remove unused descriptor state dirty

Alexander van der Grinten (1):

- egl: Fix _eglPointerIsDereferencable w/o mincore()

Alexander von Gluck IV (1):

- haiku/hgl: Fix build via header reordering

Alyssa Rosenzweig (223):

- pipe-loader: Build kmsro loader for with all kmsro targets
- pan/midgard: Remove OP_IS_STORE_VARY
- pan/midgard: Add a dummy source for loads
- pan/midgard: Refactor swizzles
- pan/midgard: Eliminate blank_alu_src
- pan/midgard: Use fp32 blend shaders
- pan/midgard: Validate tags when branching
- pan/midgard: Fix quadword_count handling

- pan/midgard: Compute bundle interference
- pan/midgard: Add bizarre corner case
- pan/midgard: offset_swizzle doesn't need dstsize
- pan/midgard: Extend offset_swizzle to non-32-bit
- pan/midgard: Extend swizzle packing for vec4/16-bit
- pan/midgard: Extend default_phys_reg to !32-bit
- panfrost/ci: Update T760 expectations
- pan/midgard: Fix printing of half-registers in texture ops
- pan/midgard: Disassemble half-steps correctly
- pan/midgard: Pass shader stage to disassembler
- pan/midgard: Switch base for vertex texturing on T720
- nir: Add load_output_u8_as_fp16_pan intrinsic
- pan/midgard: Identify ld_color_buffer_u8_as_fp16*
- pan/midgard: Implement nir_intrinsic_load_output_u8_as_fp16_pan
- pan/midgard: Pack load/store masks
- panfrost: Select format-specific blending intrinsics
- pan/midgard: Add blend shader selection bits for MRT
- pan/midgard: Implement linearly-constrained register allocation
- pan/midgard: Integrate LCRA
- pan/midgard: Remove util/ra support
- pan/midgard: Compute spill costs
- pan/lcra: Use Chaitin's spilling heuristic
- pan/midgard: Copypropagate vector creation
- pan/midgard: Fix copypropagation for textures
- pan/midgard: Generalize texture registers across GPUs
- pan/midgard: Fix vertex texturing on early Midgard
- pan/midgard: Use texture, not textureLod, on early Midgard
- pan/midgard: Disassemble with old pipeline always on T720
- pan/midgard: Prioritize texture registers
- pan/midgard: Expand 64-bit writemasks
- pan/midgard: Implement i2i64 and u2u64
- pan/midgard: Fix mir_round_bytemask_down for !32b
- pan/midgard: Pack 64-bit swizzles
- pan/midgard: Use generic constant packing for 8/64-bit
- pan/midgard: Implement non-aligned UBOs
- pan/midgard: Expose more typesize helpers

- pan/midgard: Fix masks/alignment for 64-bit loads
- pan/midgard: Represent ld/st offset unpacked
- pan/midgard: Use shader stage in mir_op_computes_derivative
- panfrost: Stub out clover callbacks
- panfrost: Pass kernel inputs as uniforms
- panfrost: Disable tiling for GLOBAL resources
- panfrost: Set PIPE_COMPUTE_CAP_ADDRESS_BITS to 64
- pan/midgard: Introduce quirks checks
- panfrost: Add the lod_bias field
- nir: Add load_sampler_lod_paramaters_pan intrinsic
- pan/midgard: Implement load_sampler_lod_paramaters_pan
- pan/midgard: Add LOD bias/clamp lowering
- pan/midgard: Describe quirk MIDGARD_BROKEN_LOD
- pan/midgard: Enable LOD lowering only on buggy chips
- panfrost: Add lcra.c to Android.mk
- pan/midgard: Use lower_tex_without_implicit_lod
- panfrost: Add information about T720 tiling
- panfrost: Implement pan_tiler for non-hierarchy GPUs
- panfrost: Simplify draw_flags
- pan/midgard: Splatter on fragment out
- gitlab-ci: Remove non-default skips from Panfrost
- panfrost: Remove blend shader hack
- panfrost: Update SET_VALUE with information from igt
- panfrost: Rename SET_VALUE to WRITE_VALUE
- gallium/util: Support POLYGON in u_stream_outputs_for_vertices
- pan/midgard: Move spilling code out of scheduler
- pan/midgard: Split spill node selection/spilling
- pan/midgard: Simplify spillability test
- pan/midgard: Remove spill cost heuristic
- pan/midgard: Move bounds checking into LCRA
- pan/midgard: Remove consecutive_skip code
- pan/midgard: Remove code marked “TODO: remove me”
- pan/midgard: Dynamically allocate r26/27 for spills
- pan/midgard: Use no_spill bitmask
- pan/midgard: Don’t use no_spill for memory spill src
- pan/midgard: Force alignment for csel_v

- pan/midgard: Don't try to free NULL in LCRA
- pan/midgard: Simplify and fix vector copyprop
- pan/midgard: Fix shift for TLS access
- panfrost: Describe thread local storage sizing rules
- panfrost: Rename unknown_address_0 -> scratchpad
- panfrost: Split stack_shift nibble from unk0
- panfrost: Add routines to calculate stack size/shift
- panfrost: Factor out panfrost_query_raw
- panfrost: Query core count and thread tls alloc
- panfrost: Route stack_size from compiler
- panfrost: Emit SFBD/MFBD after a batch, instead of before
- panfrost: Handle minor cppcheck issues
- pan/midgard: Remove unused ld/st packing hepers
- pan/midgard: Handle misc. cppcheck warnings
- panfrost: Calculate maximum stack_size per batch
- panfrost: Pass size to panfrost_batch_get_scratchpad
- pandecode: Add cast
- panfrost: Move nir_undef_to_zero to Midgard compiler
- panfrost: Move property queries to _encoder
- panfrost: Add panfrost_model_name helper
- panfrost: Report GPU name in es2_info
- ci: Remove T760/T860 from CI temporarily
- panfrost: Pass blend RT number through
- pan/midgard: Add schedule barrier after fragment writeout
- pan/midgard: Writeout per render target
- pan/midgard: Fix liveness analysis with multiple epilogues
- pan/midgard: Set r1.w magic
- panfrost: Fix FBD issue
- ci: Reinstate Panfrost CI
- panfrost: Remove fbd_type enum
- panfrost: Pack invocation_shifts manually instead of a bit field
- panfrost: Remove asserts in panfrost_pack_work_groups_compute
- panfrost: Simplify sampler upload condition
- panfrost: Don't double-create scratchpad
- panfrost: Add PAN_MESA_DEBUG=precompile for shader-db
- panfrost: Let precompile imply shaderdb

- panfrost: Handle empty shaders
- pan/midgard: Use a reg temporary for mutiple writes
- pan/midgard: Hoist temporary coordinate for cubemaps
- pan/midgard: Set .shadow for shadow samplers
- pan/midgard: Set Z to shadow comparator for 2D
- pan/midgard: Add uniform/work heuristic
- pan/midgard: Implement textureOffset for 2D textures
- pan/midgard: Fix crash with txs
- pan/midgard: Lower txd with lower_tex
- panfrost: Decode shader types in pantrace shader-db
- pan/decode: Skip COMPUTE in blobber-db
- pan/decode: Prefix blobberdb with MESA_SHADER_*
- pan/decode: Append 0:0 spills:fills to blobber-db
- pan/midgard: Fix disassembler cycle/quadword counting
- pan/midgard: Bounds check lcra_restrict_range
- pan/midgard: Extend IS_VEC4_ONLY to arguments
- pan/midgard: Clamp LOD register swizzle
- pan/midgard: Expand swizzle for texelFetch
- pan/midgard: Fix fallthrough from offset to comparator
- pan/midgard: Do witchcraft on texture offsets
- pan/midgard: Generalize temp coordinate to non-2D
- pan/midgard: Implement shadow cubemaps
- pan/midgard: Enable lower_(un)pack_* lowering
- pan/midgard: Support loads from R11G11B10 in a blend shader
- pan/midgard: Add mir_upper_override helper
- pan/midgard: Compute destination override
- panfrost: Rename pan_instancing.c -> pan_attributes.c
- panfrost: Factor batch/resource out of instancing routines
- panfrost: Move instancing routines to encoder/
- panfrost: Factor out panfrost_compute_magic_divisor
- panfrost: Fix off-by-one in pan_invocation.c
- pan/decode: Fix reference computation for invocations
- panfrost: Slight cleanup of Gallium's pan_attribute.c
- panfrost: Remove pan_shift_odd
- pan/decode: Handle gl_VertexID/gl_InstanceID
- panfrost: Unset vertex_id_zero_based

- pan/midgard: Factor out emit_attr_read
- pan/midgard: Lower gl_VertexID/gl_InstanceID to attributes
- panfrost: Extend attribute_count for vertex builtins
- panfrost: Route gl_VertexID through cmdstream
- pan/midgard: Fix minor typo
- panfrost: Remove MALI_SPECIAL_ATTRIBUTE_BASE defines
- panfrost: Update information on fixed attributes/varyings
- panfrost: Remove MALI_ATTR_INTERNAL
- panfrost: Inline away MALI_NEGATIVE
- panfrost: Implement remaining texture wrap modes
- panfrost: Add pan_attributes.c to Android.mk
- panfrost: Add missing #include in common header
- panfrost: Remove mali_alt_func
- panfrost; Update comment about work/uniform_count
- panfrost: Remove 32-bit next_job path
- glsl: Set .flat for gl_FrontFacing
- pan/midgard: Promote tilebuffer reads to 32-bit
- pan/midgard: Use type-appropriate st_vary
- pan/midgard: Implement flat shading
- panfrost: Identify glProvokingVertex flag
- panfrost: Disable some CAPs we want lowered
- panfrost: Implement integer varyings
- panfrost: Remove MRT indirection in blend shaders
- panfrost: Respect glPointSize()
- pan/midgard: Convert fragment writeout to proper branches
- pan/midgard: Remove prepacked_branch
- panfrost: Handle RGB16F colour clear
- panfrost: Pack MRT blend shaders into a single BO
- pan/midgard: Fix memory corruption in constant combining
- pan/midgard: Use better heuristic for shader termination
- pan/midgard: Generalize IS_ALU and quadword_size
- pan/midgard: Generate MRT writeout loops
- pan/midgard: Remove old comment
- pan/midgard: Identity ld_color_buffer as 32-bit
- pan/midgard: Use upper ALU tags for MFBD writeout
- panfrost: Texture from Z32F_S8 as R32F

- panfrost: Support rendering to non-zero Z/S layers
- panfrost: Implement sRGB blend shaders
- panfrost: Cleanup tiling selection logic
- panfrost: Report MSAA 4x supported for dEQP
- panfrost: Handle PIPE_FORMAT_R10G10B10A2_USCALED
- panfrost: Respect constant buffer_offset
- panfrost: Adjust for mismatch between hardware/Gallium in arrays/cube
- pan/midgard: Account for z/w flip in texelFetch
- panfrost: Don't double-flip Z/W for 2D arrays
- pan/midgard: Support indirect UBO offsets
- panfrost: Fix linear depth textures
- pan/midgard: Bytemasks should round up, not round down
- panfrost: Identify un/pack colour opcodes
- pan/midgard: Fix recursive csel scheduling
- panfrost: Expose some functionality with dEQP flag
- panfrost: Compile tiling routines with -O3
- panfrost,lima: De-Galliumize tiling routines
- panfrost: Rework linear<—>tiled conversions
- panfrost: Add pandecode entries for ASTC/ETC formats
- panfrost: Fix crash in compute variant allocation
- panfrost: Drop mysterious zero=0xFFFF field
- panfrost: Don't use implicit mali_exception_status enum
- pan/decode: Remove last_size
- pan/midgard: Remove pack_color define
- pan/decode: Remove SHORT_SLIDE indirection
- panfrost: Fix 32-bit warning for 'indices'
- pan/decode: Drop MFBD compute shader stuff
- pan/midgard: Record TEXTURE_OP_BARRIER
- pan/midgard: Disassemble barrier instructions
- pan/midgard: Validate barriers use a barrier tag
- pan/midgard: Handle tag 0x4 as texture
- pan/midgard: Remove float_bitcast
- pan/midgard: Fix missing prefixes
- pan/midgard: Don't crash with constants on unknown ops
- pan/midgard: Use fprintf instead of printf for constants

Andreas Baierl (14):

- lima: Beautify stream dumps
- lima: Parse VS and PLBU command stream while making a dump
- lima/streamparser: Fix typo in vs semaphore parser
- lima/streamparser: Add findings introduced with gl_PointSize
- lima/parser: Some fixes and cleanups
- lima/parser: Add RSW parsing
- lima/parser: Add texture descriptor parser
- lima: Rotate dump files after each finished pp frame
- lima: Fix dump file creation
- lima/parser: Fix rsw parser
- lima/parser: Fix VS cmd stream parser
- lima/parser: Make rsw alpha blend parsing more readable
- lima: Add stencil support
- lima: Fix alpha blending

Andres Rodriguez (1):

- vulkan/wsi: disable the hardware cursor

Andrii Simiklit (5):

- main: fix several 'may be used uninitialized' warnings
- glsl: fix an incorrect max_array_access after optimization of ssbo/ubo
- glsl: fix a binding points assignment for ssbo/ubo arrays
- glsl/nir: do not change an element index to have correct block name
- mesa/st: fix a memory leak in get_version

Anthony Pesch (5):

- util: import xxhash
- util: move fnv1a hash implementation into its own header
- util/hash_table: replace _mesa_hash_data's fnv1a hash function with xxhash
- util/hash_table: added hash functions for integer types
- util/hash_table: update users to use new optimal integer hash functions

Anuj Phogat (2):

- intel: Add device info for 1x4x6 Jasper Lake
- intel: Add pci-ids for Jasper Lake

Arno Messiaen (5):

- lima: fix stride in texture descriptor
- lima: add layer_stride field to lima_resource struct
- lima: introduce ppir_op_load_coords_reg to differentiate between loading texture coordinates straight from a varying vs loading them from a register

- lima: add cubemap support
- lima/ppir: add lod-bias support

Bas Nieuwenhuizen (33):

- radv: Fix timeout handling in syncobj wait.
- radv: Remove _mesa_locale_init/fini calls.
- turnip: Remove _mesa_locale_init/fini calls.
- anv: Remove _mesa_locale_init/fini calls.
- radv: Fix disk_cache_get size argument.
- radv: Close all unnecessary fds in secure compile.
- radv: Do not change scratch settings while shaders are active.
- radv: Allocate cmdbuffer space for buffer marker write.
- radv: Enable VK_KHR_buffer_device_address.
- amd/llvm: Refactor ac_build_scan.
- radv: Unify max_descriptor_set_size.
- radv: Fix timeline semaphore refcounting.
- radv: Fix RGBX Android<->Vulkan format correspondence.
- amd/common: Fix tcCompatible degradation on Stoney.
- amd/common: Always use addrlib for HTILE tc-compat.
- radv: Limit workgroup size to 1024.
- radv: Expose all sample counts for integer formats as well.
- amd/common: Handle alignment of 96-bit formats.
- nir: Add clone/hash/serialize support for non-uniform tex instructions.
- nir: print non-uniform tex fields.
- amd/common: Always initialize gfx9 mipmap offset/pitch.
- turnip: Use VK_NULL_HANDLE instead of NULL.
- meson: Enable -Werror=int-conversion.
- Revert "amd/common: Always initialize gfx9 mipmap offset/pitch."
- radv: Only use the gfx mipmap level offset/pitch for linear textures.
- spirv: Fix glsl type assert in spir2nir.
- radv: Emit a BATCH_BREAK when changing pixel shaders or CB_TARGET_MASK.
- radv: Use new scanout gfx9 metadata flag.
- radv: Disable VK_EXT_sample_locations on GFX10.
- radv: Remove syncobj_handle variable in header.
- radv: Expose VK_KHR_swapchain_mutable_format.
- radv: Allow DCC & TC-compat HTILE with VK_IMAGE_CREATE_EXTENDED_USAGE_BIT.
- radv: Do not set SX DISABLE bits for RB+ with unused surfaces.

Ben Crocker (1):

- llvmpipe: use ppc64le/ppc64 Large code model for JIT-compiled shaders

Bernd Kuhls (1):

- util/os_socket: Include unistd.h to fix build error

Boris Brezillon (21):

- panfrost: MALI_DEPTH_TEST is actually MALI_DEPTH_WRITEMASK
- panfrost: Destroy the upload manager allocated in panfrost_create_context()
- panfrost: Release the ctx->pipe_framebuffer ref
- panfrost: Move BO cache related fields to a sub-struct
- panfrost: Try to evict unused BOs from the cache
- gallium: Fix the ->set_damage_region() implementation
- panfrost: Make sure we reset the damage region of RTs at flush time
- panfrost: Remove unneeded phi nodes
- panfrost/midgard: Fix swizzle for store instructions
- panfrost/midgard: Print the actual source register for store operations
- panfrost/midgard: Use a union to manipulate embedded constants
- panfrost/midgard: Rework mir_adjust_constants() to make it type/size agnostic
- panfrost/midgard: Make sure promote_fmov() only promotes 32-bit imovs
- panfrost/midgard: Factorize f2f and u2u handling
- panfrost/midgard: Add f2f64 support
- panfrost/midgard: Fix mir_print_instruction() for branch instructions
- panfrost/midgard: Add 64 bits float <-> int converters
- panfrost/midgard: Add missing lowering passes for type/size conversion ops
- panfrost/midgard: Add a condense_writemask() helper
- panfrost/midgard: Prettify embedded constant prints
- panfrost: Fix the damage box clamping logic

Brian Ho (14):

- turnip: Update tu_query_pool with turnip-specific fields
- turnip: Implement vkCreateQueryPool for occlusion queries
- turnip: Implement vkCmdBeginQuery for occlusion queries
- turnip: Implement vkCmdEndQuery for occlusion queries
- turnip: Update query availability on render pass end
- turnip: Implement vkGetQueryPoolResults for occlusion queries
- turnip: Implement vkCmdResetQueryPool
- turnip: Implement vkCmdCopyQueryPoolResults for occlusion queries
- anv: Properly fetch partial results in vkGetQueryPoolResults

- anv: Handle unavailable queries in vkCmdCopyQueryPoolResults
- turnip: Enable occlusionQueryPrecise
- turnip: Free event->bo on vkDestroyEvent
- turnip: Fix vkGetQueryPoolResults with available flag
- turnip: Fix vkCmdCopyQueryPoolResults with available flag

Brian Paul (4):

- s/APIENTRY/GLAPIENTRY/ in teximage.c
- nir: fix a couple signed/unsigned comparison warnings in nir_builder.h
- Call shmget() with permission 0600 instead of 0777
- nir: no-op C99 _Pragma() with MSVC

C Stout (1):

- util/vector: Fix u_vector_foreach when head rolls over

Caio Marcelo de Oliveira Filho (24):

- spirv: Don't leak GS initialization to other stages
- glsl: Check earlier for MaxShaderStorageBlocks and MaxUniformBlocks
- glsl: Check earlier for MaxTextureImageUnits and MaxImageUniforms
- anv: Initialize depth_bounds_test_enable when not explicitly set
- spirv: Consider the sampled_image case in wa_glslang_179 workaround
- intel/fs: Lower 64-bit MOVs after lower_load_payload()
- intel/fs: Fix lowering of dword multiplication by 16-bit constant
- intel/vec4: Fix lowering of multiplication by 16-bit constant
- anv/gen12: Temporarily disable VK_KHR_buffer_device_address (and EXT)
- spirv: Implement SPV_KHR_non_semantic_info
- panfrost: Fix Makefile.sources
- anv: Drop unused function parameter
- anv: Ignore some CreateInfo structs when rasterization is disabled
- intel/fs: Only use SLM fence in compute shaders
- spirv: Drop EXT for PhysicalStorageBuffer symbols
- spirv: Handle PhysicalStorageBuffer in memory barriers
- nir: Add missing nir_var_mem_global to various passes
- intel/fs: Add FS_OPCODE_SCHEDULING_FENCE
- intel/fs: Add workgroup_size() helper
- intel/fs: Don't emit fence for shared memory if only one thread is used
- intel/fs: Don't emit control barrier if only one thread is used
- anv: Always initialize target_stencil_layout
- intel/compiler: Add names for SHADER_OPCODE_[IU]SUB_SAT

- nir: Make nir_deref_path_init skip trivial casts

Chris Wilson (1):

- egl: Mention if swrast is being forced

Christian Gmeiner (24):

- drm-shim: fix EOF case
- etnaviv: rs: upsampling is not supported
- etnaviv: add drm-shim
- etnaviv: drop not used config_out function param
- etnaviv: use a more self-explanatory param name
- etnaviv: handle 8 byte block in tiling
- etnaviv: add support for extended pe formats
- etnaviv: fix integer vertex formats
- etnaviv: use NORMALIZE_SIGN_EXTEND
- etnaviv: fix R10G10B10A2 vertex format entries
- etnaviv: handle integer case for GENERIC_ATTRIB_SCALE
- etnaviv: remove dead code
- etnaviv: remove not used etna_bits_ones(..)
- etnaviv: drop compiled_rs_state forward declaration
- etnaviv: update resource status after flushing
- gallium: add PIPE_CAP_MAX_VERTEX_BUFFERS
- etnaviv: check if MSAA is supported
- etnaviv: gc400 does not support any vertex sampler
- etnaviv: use a better name for FE_VERTEX_STREAM_UNK14680
- etnaviv: move state based texture structs
- etnaviv: move descriptor based texture structs
- etnaviv: add deqp debug option
- etnaviv: drop default state for PE_STENCIL_CONFIG_EXT2
- etnaviv: drm-shim: add GC400

Connor Abbott (19):

- nir: Fix non-determinism in lower_global_vars_to_local
- radv: Rename ac_arg_regfile
- ac: Add a shared interface between radv, radeonsi, LLVM and ACO
- ac/nir, radv, radeonsi: Switch to using ac_shader_args
- radv: Move argument declaration out of nir_to_llvm
- aco: Constify radv_nir_compiler_options in isel
- aco: Use radv_shader_args in aco_compile_shader()

- aco: Split vector arguments at the beginning
- aco: Make num_workgroups and local_invocation_ids one argument each
- radv: Replace supports_spill with explict_scratch_args
- aco: Use common argument handling
- aco: Make unused workgroup id's 0
- nir: Maintain the algebraic automaton's state as we work.
- a6xx: Add more CP packets
- freedreno: Use new macros for CP_WAIT_REG_MEM and CP_WAIT_MEM_GTE
- freedreno: Fix CP_MEM_TO_REG flag definitions
- freedreno: Document CP_COND_REG_EXEC more
- freedreno: Document CP_UNK_A6XX_55
- freedreno: Document CP_INDIRECT_BUFFER_CHAIN

Daniel Ogorchock (2):

- panfrost: Fix panfrost_bo_access memory leak
- panfrost: Fix headers and gpu_headers memory leak

Daniel Schürmann (58):

- aco: fix immediate offset for spills if scratch is used
- aco: only use single-dword loads/stores for spilling
- aco: fix accidential reordering of instructions when scheduling
- aco: workaround Tonga/Iceland hardware bug
- aco: fix invalid access on Pseudo_instructions
- aco: preserve kill flag on moved operands during RA
- aco: rematerialize s_movk instructions
- aco: check if SALU instructions are predeceeded by exec when calculating WQM needs
- aco: value number instructions using the execution mask
- aco: use s_and_b64 exec to reduce uniform booleans to one bit
- amd/llvm: Add Subgroup Scan functions for SI
- radv: Enable Subgroup Arithmetic and Clustered for SI
- aco: don't value-number instructions from within a loop with ones after the loop.
- aco: don't split live-ranges of linear VGPRs
- aco: fix a couple of value numbering issues
- aco: refactor visit_store_fs_output() to use the Builder
- aco: Initial GFX7 Support
- aco: SI/CI - fix sampler aniso
- aco: fix SMEM offsets for SI/CI
- aco: implement nir_op_fquantize2f16 for SI/CI

- aco: only use scalar loads for readonly buffers on SI/CI
- aco: implement nir_op_isign on SI/CI
- aco: move buffer_store data to VGPR if needed
- aco: implement quad swizzles for SI/CI
- aco: recognize SI/CI SMRD hazards
- aco: fix disassembly of writelane instructions.
- aco: split read/writelane opcode into VOP2/VOP3 version for SI/CI
- aco: implement 64bit VGPR shifts for SI/CI
- aco: make 1/2*PI a literal constant on SI/CI
- aco: implement 64bit i2b for SI /CI
- aco: implement 64bit ine/ieq for SI/CI
- aco: disable disassembly for SI/CI due to lack of support by LLVM
- radv: only flush scalar cache for SSBO writes with ACO on GFX8+
- aco: flush denorms after fmin/fmax on pre-GFX9
- aco: don't use a scalar temporary for reductions on GFX10
- aco: implement (clustered) reductions for SI/CI
- aco: implement inclusive_scan for SI/CI
- aco: implement exclusive scan for SI/CI
- radv: disable Youngblood app profile if ACO is used
- aco: return to loop_active mask at continue_or_break blocks
- radv: Enable ACO on GFX7 (Sea Islands)
- aco: use soffset for MUBUF instructions on SI/CI
- aco: improve readfirstlane after uniform ssbo loads on GFX7
- aco: propagate temporaries into expanded vectors
- nir: fix printing of var_decl with more than 4 components.
- aco: compact various Instruction classes
- aco: compact aco::span<T> to use uint16_t offset and size instead of pointer and size_t.
- aco: fix unconditional demote_to_helper
- aco: rework lower_to_cssa()
- aco: handle phi affinities transitively through parallelcopies
- aco: ignore parallelcopies to the same register on jump threading
- aco: fix combine_salu_not_bitwise() when SCC is used
- aco: reorder VMEM operands in ACO IR
- aco: fix register allocation with multiple live-range splits
- aco: simplify adjust_sample_index_using_fmask() & get_image_coords()
- aco: simplify gathering of MIMG address components

- docs: add new features for RADV/ACO.
- aco: fix image_atomic_cmp_swap

Daniel Stone (2):

- Revert "st/dri: do FLUSH_VERTICES before calling flush_resource"
- Revert "gallium: add st_context_iface::flush_resource to call FLUSH_VERTICES"

Danylo Piliaiev (12):

- intel/blorp: Fix usage of uninitialized memory in key hashing
- i965/program_cache: Lift restriction on shader key size
- intel/blorp: Fix usage of uninitialized memory in key hashing
- intel/fs: Do not lower large local arrays to scratch on gen7
- i965: Unify CC_STATE and BLEND_STATE atoms on Haswell as a workaround
- glsl: Add varyings to "zero-init of uninitialized vars" workaround
- drirc: Add glsl_zero_init workaround for GpuTest
- iris/query: Implement PIPE_QUERY_GPU_FINISHED
- iris: Fix value of out-of-bounds accesses for vertex attributes
- i965: Do not set front_buffer_dirty if there is no front buffer
- st/mesa: Handle the rest renderbuffer formats from OSMesa
- st/nir: Unify inputs_read/outputs_written before serializing NIR

Dave Airlie (74):

- nir/serialize: pack function has name and entry point into flags.
- nir/serialize: fix serializing functions with no implementations.
- spirv: don't store 0 to cs.ptr_size for non kernel stages.
- spirv: get the correct type for function returns.
- spirv/nir/opencl: handle some multiply instructions.
- nir: add 64-bit ufind_msb lowering support. (v2)
- nouveau: request ufind_msb64 lowering in the frontend.
- vtn/opencl: add clz support
- nir: fix deref offset builder
- llvmpipe: initial query buffer object support. (v2)
- docs: add llvmpipe to ARB_query_buffer_object.
- gallivm: split out the flow control ir to a common file.
- gallivm: nir->tgsi info convertor (v2)
- gallivm: add popcount intrinsic wrapper
- gallivm: add cttz wrapper
- gallivm: add selection for non-32 bit types
- gallivm: add nir->llvm translation (v2)

- draw: add nir info gathering and building support
- gallium: add nir lowering passes for the draw pipe stages. (v2)
- gallivm: add swizzle support where one channel isn't defined.
- llvmpipe: add initial nir support
- nir/samplers: don't zero samplers_used/txf.
- llvmpipe/images: handle undefined atomic without crashing
- gallivm/llvmpipe: add support for front facing in sysval.
- llvmpipe: enable texcoord semantics
- gallium/scons: fix graw-xlib build on OSX.
- llvmpipe: add queries disabled flag
- llvmpipe: disable occlusion queries when requested by state tracker
- draw: add support for collecting primitives generated outside streamout
- llvmpipe: enable support for primitives generated outside streamout
- aco: handle gfx7 int8/10 clamping on exports
- gallivm: add bitfield reverse and ufind_msb
- llvmpipe/nir: handle texcoord requirements
- gallivm: fix transpose for when first channel isn't created
- gallivm: fix perspective enable if usage_mask doesn't have 0 bit set
- gallivm/nir: cleanup code and call cmp wrapper
- gallivm/nir: copy compare ordering code from tgsi
- gallivm: add base instance sysval support
- gallivm/draw: add support for draw_id system value.
- gallivm: fixup base_vertex support
- llvmpipe: enable ARB_shader_draw_parameters.
- vtn: convert vload/store to single value loops
- vtn/opencl: add shuffle/shuffle support
- gallivm/nir: wrap idiv to avoid divide by 0 (v2)
- llvmpipe: switch to NIR by default
- nir: sanitize work group intrinsics to always be 32-bit.
- gallivm: add 64-bit const int creator.
- llvmpipe/gallivm: add kernel inputs
- gallivm: add support for 8-bit/16-bit integer builders
- gallivm: pick integer builders for alu instructions.
- gallivm/nir: allow 8/16-bit conversion and comparison.
- tgsi/mesa: handle KERNEL case
- gallivm/llvmpipe: add support for work dimension intrinsic.

- gallivm/llvmpipe: add support for block size intrinsic
- gallivm/llvmpipe: add support for global operations.
- llvmpipe: handle serialized nir as a shader type.
- llvmpipe: add support for compute shader params
- llvmpipe/nir: use nir_max_vec_components in more places
- gallivm: handle non-32 bit undefined
- llvmpipe: lower hadd/add_sat
- gallivm/nir: lower packing
- gallivm/nir: add vec8/16 support
- llvmpipe: add debug option to enable OpenCL support.
- gallivm: fixup const int64 builder.
- llvmpipe: enable ARB_shader_group_vote.
- gallium/util: add multi_draw_indirect to util_draw_indirect.
- llvmpipe: enable driver side multi draw indirect
- llvmpipe: add support for ARB_indirect_parameters.
- llvmpipe: add ARB_derivative_control support
- gallivm: fix gather component handling.
- llvmpipe: fix some integer instruction lowering.
- galllivm: fix gather offset casting
- gallivm: fix find lsb
- gallivm/nir: add missing break for isub.

David Heidelberg (1):

- .mailmap: use correct email address

David Stevens (1):

- virgl: support emulating planar image sampling

Denis Pauk (2):

- gallium/swr: Enable support bptc format.
- docs/features: mark GL_ARB_texture_compression_bptc as done for llvmpipe, softpipe, swr

Dongwon Kim (3):

- gallium: enable INTEL_PERFORMANCE_QUERY
- iris: INTEL performance query implementation
- gallium: check all planes' pipe formats in case of multi-samplers

Drew Davenport (1):

- radeonsi: Clear uninitialized variable

Drew DeVault (1):

- st_get_external_sampler_key: improve error message

Duncan Hopkins (1):

- zink: make sure src image is transfer-src-optimal

Dylan Baker (69):

- Bump VERSION to 20.0.0-devel
- docs/new_features: Empty the feature list for the 20.0 cycle
- nir: correct use of identity check in python
- r200: use preprocessor for big vs little endian checks
- r100: Use preprocessor to select big vs little endian paths
- dri/osmesa: use preprocessor for selecting endian code paths
- util/u_endian: Use _WIN32 instead of _MSC_VER
- util/u_endian: set PIPE_ARCH_*_ENDIAN to 1
- mesa/main: replace uses of _mesa_little_endian with preprocessor
- mesa/swrast: replace instances of _mesa_little_endian with preprocessor
- mesa/main: delete now unused _mesa_little_endian
- gallium/osmesa: Use PIPE_ARCH_*_ENDIAN instead of little_endian function
- util: rename PIPE_ARCH_*_ENDIAN to UTIL_ARCH_*_ENDIAN
- util/u_endian: Add error checks
- meson: Add dep_glvnd to egl deps when building with glvnd
- docs: add release notes for 19.2.3
- docs: add sha256 sum to 19.2.3 release notes
- docs: update calendar, add news item and link release notes for 19.2.2
- meson: gtest needs pthreads
- gallium/osmesa: Convert osmesa test to gtest
- osmesa/tests: Extend render test to cover other working cases
- util: Use ZSTD for shader cache if possible
- docs: Add release notes for 19.2.4
- docs: Add SHA256 sum for for 19.2.4
- docs: update calendar, add news item and link release notes for 19.2.4
- docs: Add relnotes for 19.2.5
- docs/relnotes/19.2.5: Add SHA256 sum
- docs: update calendar, add news item and link release notes for 19.2.5
- docs/release-calendar: Update for extended 19.3 rc period
- docs: Add release notes for 19.2.6
- docs: Add SHA256 sum for 19.2.6
- docs: update calendar, add news item and link release notes for 19.2.6
- gallium/auxiliary: Fix uses of gnu struct = { } extension

- meson: Add -Werror=gnu-empty-initializer to MSVC compat args
- docs: Add release notes for 19.2.7
- docs: Add SHA256 sums for 19.2.7
- docs: update calendar, add news item and link release notes for 19.2.7
- docs: Update mesa 19.3 release calendar
- meson/broadcom: libbroadcom_cle needs expat headers
- meson/broadcom: libbroadcom_cle also needs zlib
- docs: add release notes for 19.3.0
- docs/19.3.0: Add SHA256 sums
- docs: Update release notes, index, and calendar for 19.3.0
- dcos: add releanse notes for 19.3.1
- docs: Add release notes, update calendar, and add news for 19.3.1
- docs: add relnotes for 19.2.8
- docs/relnotes/19.2.8: Add SHA256 sum
- docs: Add release notes, news, and update calendar for 19.2.8
- docs: Add release notes for 19.3.2
- docs: add SHA256 sums for 19.3.2
- docs: Add release notes for 19.3.2, update calendar and home page
- docs: Update release calendar for 20.0
- docs: Add relnotes for 19.3.3 release
- docs: Add SHA 256 sums for 19.3.3
- docs: update news, calendar, and link release notes for 19.3.3
- VERSION: bump to 20.0.0-rc1
- bin/pick-ui: Add a new maintainer script for picking patches
- .pick_status.json: Update to 0d14f41625fa00187f690f283c1eb6a22e354a71
- .pick_status.json: Update to b550b7ef3b8d12f533b67b1a03159a127a3ff34a
- .pick_status.json: Update to 9afdcd64f2c96f3fcc1a28912987f2e8066aa995
- .pick_status.json: Update to 7eaf21cb6f67adbe0e79b80b4feb8c816a98a720
- VERSION: bump to 20.0-rc2
- .pick_status.json: Update to d8bae10bfe0f487dcaec721743cd51441bcc12f5
- .pick_status.json: Update to 689817c9dfde9a0852f2b2489cb0fa93ffbcb215
- .pick_status.json: Update to 23037627359e739c42b194dec54875aefbb9d00b
- VERSION: bump for 20.0.0-rc3
- .pick_status.json: Update to 2a98cf3b2ecea43cea148df7f77d2abadfd1c9db
- .pick_status.json: Update to 946eacbafb47c8b94d47e7c9d2a8b02fff5a22fa
- .pick_status.json: Update to bee5c9b0dc13dbae0ccf124124eaccebf7f2a435

Eduardo Lima Mitev (2):

- turnip: Remove failed command buffer from pool
- turnip: Fix issues in tu_compute_pipeline_create() that may lead to crash

Elie Tournier (4):

- Docs: remove duplicate meson docs for windows
- docs: fix ascii html representation
- nir/algebraic: i2f(f2i()) -> trunc()
- nir/algebraic: sqrt(x)*sqrt(x) -> fabs(x)

Emmanuel Gil Peyrot (1):

- intel/compiler: Return early if read() failed

Eric Anholt (102):

- ci: Make lava inherit the ccache setup of the .build script.
- ci: Switch over to an autoscaling GKE cluster for builds.
- Revert "ci: Switch over to an autoscaling GKE cluster for builds."
- mesa/st: Add mapping of MESA_FORMAT_RGB_SNORM16 to gallium.
- gallium: Add defines for FXT1 texture compression.
- gallium: Add some more channel orderings of packed formats.
- gallium: Add an equivalent of MESA_FORMAT_BGR_UNORM8.
- gallium: Add equivalents of packed MESA_FORMAT_*UINT formats.
- mesa: Stop defining a full separate format for RGBA_UINT8.
- mesa/st: Test round-tripping of all compressed formats.
- mesa: Prepare for the MESA_FORMAT_* enum to be sparse.
- mesa: Redefine MESA_FORMAT_* in terms of PIPE_FORMAT_*.
- mesa/st: Gut most of st_mesa_format_to_pipe_format().
- mesa/st: Make st_pipe_format_to_mesa_format an effective no-op.
- u_format: Fix swizzle of A1R5G5B5.
- ci: Use several debian buster packages instead of hand-building.
- ci: Make the skip list regexes match the full test name.
- ci: Use cts_runner for our dEQP runs.
- ci: Enable all of GLES3/3.1 testing for softpipe.
- ci: Remove old commented copy of freedreno artifacts.
- ci: Disable flappy blit tests on a630.
- ci: Expand the freedreno blit skip regex to cover more cases.
- util: Move gallium's PIPE_FORMAT utils to /util/format/
- mesa: Move compile of common Mesa core files to a static lib.
- mesa/st: Simplify st_choose_matching_format().

- mesa: Don't put sRGB formats in the array format table.
- mesa/st: Reuse st_choose_matching_format from st_choose_format().
- util: Add a mapping from VkFormat to PIPE_FORMAT.
- turnip: Drop the copy of the formats table.
- ci: Move freedreno's parallelism to the runner instead of gitlab-ci jobs.
- ci: Use a tag from the parallel-deqp-runner repo.
- nir: Add a scheduler pass to reduce maximum register pressure.
- nir: Refactor algebraic's block walk
- nir: Make algebraic backtrack and reprocess after a replacement.
- freedreno: Introduce a fd_resource_layer_stride() helper.
- freedreno: Introduce a fd_resource_tile_mode() helper.
- freedreno: Introduce a resource layout header.
- freedreno: Convert the slice struct to the new resource header.
- freedreno/a6xx: Log the tiling mode in resource layout debug.
- turnip: Disable timestamp queries for now.
- turnip: Fix unused variable warnings.
- turnip: Drop redefinition of VALIDREG now that it's in ir3.h.
- turnip: Reuse tu6_stage2opcode() more.
- turnip: Add basic SSBO support.
- turnip: Refactor the graphics pipeline create implementation.
- turnip: Add a helper function for getting tu_buffer iovas.
- turnip: Sanity check that we're adding valid BOs to the list.
- turnip: Move pipeline BO list adding to BindPipeline.
- turnip: Add support for compute shaders.
- ci: Disable egl_ext_device_drm tests in piglit.
- freedreno: Enable texture upload memory throttling.
- freedreno: Stop forcing ALLOW_MAPPED_BUFFERS_DURING_EXEC off.
- freedreno: Track the set of UBOs to be uploaded in UBO analysis.
- freedreno: Drop the extra offset field for mipmap slices.
- freedreno: Refactor the UBWC flags registers emission.
- freedreno: Move UBWC layout into a slices array like the non-UBWC slices.
- tu: Move our image layout into a freedreno_layout struct.
- freedreno: Move a6xx's setup_slices() to a shareable helper function.
- freedreno: Switch the 16-bit workaround to match what turnip does.
- tu: Move UBWC layout into fdl6_layout() and use that function.
- turnip: Lower usub_borrow.

- turnip: Drop unused variable.
- turnip: Add support for descriptor arrays.
- turnip: Fix support for immutable samplers.
- ci: Fix caselist results archiving after parallel-deqp-runner rename.
- mesa: Fix detection of invalidating both depth and stencil.
- mesa/st: Deduplicate the NIR uniform lowering code.
- mesa/st: Move the vec4 type size function into core GLSL types.
- mesa/prog: Reuse count_vec4_slots() from ir_to_mesa.
- mesa/st: Move the dword slot counting function to glsl_types as well.
- i965: Reuse the new core glsl_count_dword_slots().
- nir: Fix printing of ~0 .locations.
- turnip: Refactor linkage state setup.
- mesa: Make atomic lowering put atomics above SSBOs.
- gallium: Pack the atomic counters just above the SSBOs.
- nir: Drop the ssbo_offset to atomic lowering.
- compiler: Add a note about how num_ssbos works in the program info.
- freedreno: Stop scattered remapping of SSBOs/images to IBOs.
- radeonsi: Remove a bunch of default handling of pipe caps.
- r600: Remove a bunch of default handling of pipe caps.
- r300: Remove a bunch of default handling of pipe caps.
- radeonsi: Drop PIPE_CAP_TGSI_ANY_REG_AS_ADDRESS.
- turnip: Fix some whitespace around binary operators.
- turnip: Refactor the intrinsic lowering.
- turnip: Add limited support for storage images.
- turnip: Disable UBWC on images used as storage images.
- turnip: Add support for non-zero (still constant) UBO buffer indices.
- turnip: Add support for uniform texel buffers.
- freedreno/ir3: Plumb the ir3_shader_variant into legalize.
- turnip: Add support for fine derivatives.
- turnip: Fix execution of secondary cmd bufs with nothing in primary.
- freedreno: Add some missing a6xx address declarations.
- freedreno: Fix OUT_REG() on address regs without a .bo supplied.
- turnip: Port krh's packing macros from freedreno to tu.
- turnip: Convert renderpass setup to the new register packing macros.
- turnip: Convert the rest of tu_cmd_buffer.c over to the new pack macros.
- vulkan/wsi: Fix compiler warning when no WSI platforms are enabled.

- iris: Silence warning about AUX_USAGE_MC.
- mesa/st: Fix compiler warnings from INTEL_shader_integer_functions.
- ci: Enable -Werror on the meson-i386 build.
- tu: Fix binning address setup after pack macros change.
- Revert "gallium: Fix big-endian addressing of non-bitmask array formats."

Eric Engestrom (58):

- meson: split out idep_xmlconfig_headers from idep_xmlconfig
- anv: add missing xmlconfig headers dependency
- radv: drop unnecessary xmlpool_options_h
- pipe-loader: drop unnecessary xmlpool_options_h
- loader: replace xmlpool_options_h with idep_xmlconfig_headers
- targets/omx: replace xmlpool_options_h with idep_xmlconfig_headers
- targets/va: replace xmlpool_options_h with idep_xmlconfig_headers
- targets/vdpau: replace xmlpool_options_h with idep_xmlconfig_headers
- targets/xa: replace xmlpool_options_h with idep_xmlconfig_headers
- targets/xvmc: replace xmlpool_options_h with idep_xmlconfig_headers
- dri: replace xmlpool_options_h with idep_xmlconfig_headers
- i915: replace xmlpool_options_h with idep_xmlconfig_headers
- nouveau: replace xmlpool_options_h with idep_xmlconfig_headers
- r200: replace xmlpool_options_h with idep_xmlconfig_headers
- radeon: replace xmlpool_options_h with idep_xmlconfig_headers
- meson: move idep_xmlconfig_headers to xmlpool/
- gitlab-ci: build a recent enough version of GLVND (ie. 1.2.0)
- meson: require glvnd 1.2.0
- meson: revert glvnd workaround
- meson: add variable to control the symbols checks
- meson: move the generic symbols check arguments to a common variable
- meson: add windows support to symbols checks
- meson: require 'nm' again on Unix systems
- mesa/imports: let the build system detect strtok_r()
- egl: fix _EGL_NATIVE_PLATFORM fallback
- egl: move #include of local headers out of Khronos headers
- gitlab-ci: build libdrm using meson instead of autotools
- gitlab-ci: auto-cancel CI runs when a newer commit is pushed to the same branch
- CL: sync C headers with Khronos
- CL: sync C++ headers with Khronos

- vulkan: delete typo'd header
- egl: use EGL_CAST() macro in eglmesaext.h
- anv: add missing "fall-through" annotation
- vk_util: drop duplicate formats in vk_format_map[]
- meson: drop duplicate 'lib' prefix on libiris_gen*
- meson: drop 'intel_' prefix on imgui_core
- docs: reword a bit and list HTTPS before FTP
- intel: add mi_builder_test for gen12
- intel/compiler: add ASSERTED annotation to avoid "unused variable" warning
- intel/compiler: replace '0' pointer with 'NULL'
- util/simple_mtx: don't set the canary when it can't be checked
- anv: drop unused #include
- travis: autodetect python version instead of hard-coding it
- util/format: remove left-over util_format_description_table declaration
- util/format: add PIPE_FORMAT_ASTC_*x*x*_SRGB to util_format_{srgb,linear}()
- util/format: add trivial srgb<->linear conversion test
- u_format: move format tests to util/tests/
- amd: fix empty-body issues
- nine: fix empty-body-issues
- meson: simplify install_megadrivers.py invocation
- mesa: avoid returning a value in a void function
- meson: use github URL for wraps instead of completely unreliable wrapdb
- egl: drop confusing mincore() error message
- llvmpipe: drop LLVM < 3.4 support
- util/atomic: fix return type of p_atomic_add_return() fallback
- util/os_socket: fix header unavailable on windows
- freedreno/perfcntrs: fix fd leak
- util/disk_cache: check for write() failure in the zstd path

Erico Nunes (17):

- lima: fix nir shader memory leak
- lima: fix bo submit memory leak
- lima/ppir: enable lower_fdph
- gallium/util: add alignment parameter to util_upload_index_buffer
- lima: allocate separate bo to store varyings
- lima: refactor indexed draw indices upload
- vc4: move the draw splitting routine to shared code

- lima: split draw calls on 64k vertices
- lima/ppir: fix lod bias src
- lima/ppir: remove assert on ppir_emit_tex unsupported feature
- lima: set shader caps to optimize control flow
- lima/ppir: remove orphan load node after cloning
- lima/ppir: implement full liveness analysis for regalloc
- lima/ppir: handle write to dead registers in ppir
- lima/ppir: fix ssa undef emit
- lima/ppir: split ppir_op_undef into undef and dummy again
- lima/ppir: fix src read mask swizzling

Erik Faye-Lund (82):

- zink: heap-allocate samplers objects
- zink: emit line-width when using polygon line-mode
- anv: remove incorrect polygonMode=point early-out
- zink: use actual format for render-pass
- zink: always allow mutating the format
- zink: do not advertize coherent mapping
- zink: disable fragment-shader texture-lod
- zink: transition resources before resolving
- zink: always allow sampling of images
- zink: use u_blitter when format-reinterpreting
- zink/spirv: drop temp-array for component-count
- zink/spirv: support loading bool constants
- zink/spirv: implement bany_fnequal[2-4]
- zink/spirv: implement bany_inequal[2-4]
- zink/spirv: implement ball_iequal[2-4]
- zink/spirv: implement ball_fequal[2-4]
- zink: do advertize integer support in shaders
- zink/spirv: add support for nir_op_flrp
- zink: correct depth-stencil format
- nir: patch up deref-vars when lowering clip-planes
- zink: always allow transfer to/from buffers
- zink: implement buffer-to-buffer copies
- zink: remove no-longer-needed hack
- zink: move format-checking to separate source
- zink: move filter-helper to separate helper-header

- zink: move blitting to separate source
- zink: move drawing separate source
- st/mesa: unmap pbo after updating cache
- zink: use true/false instead of TRUE/FALSE
- zink: reject invalid sample-counts
- zink: fix crash when restoring sampler-states
- zink: delete query rather than allocating a new one
- zink: do not try to destroy NULL-fence
- zink: handle calloc-failure
- zink: avoid NULL-deref
- zink: avoid NULL-deref
- zink: avoid NULL-deref
- zink: error-check right variable
- zink: silence coverity error
- zink: enable PIPE_CAP_MIXED_COLORBUFFER_FORMATS
- zink: implement nir_texop_txd
- zink: implement txf
- zink: implement some more trivial opcodes
- zink: simplify front-face type
- zink: factor out builtin-var creation
- zink: implement load_vertex_id
- zink: use nir_fmul_imm
- zink: remove unused code-path in lower_pos_write
- nir/zink: move clip_halfz-lowering to common code
- etnaviv: use nir_lower_clip_halfz instead of open-coding
- st/mesa: use uint-samplers for sampling stencil buffers
- zink: fixup initialization of operand_mask / num_extra_operands
- util: initialize float-array with float-literals
- st/wgl: eliminate implicit cast warning
- gallium: fix a warning
- mesa/st: use float literals
- docs: fix typo in html tag name
- docs: fix paragraphs
- docs: open paragraph before closing it
- docs: use code-tag instead of pre-tag
- docs: use code-tags instead of pre-tags

- docs: use code-tags instead of pre-tags
- docs: move paragraph closing tag
- docs: remove double-closed definition-list
- docs: do not double-close link tag
- docs: do not use definition-list for sub-topics
- docs: use figure/figcaption instead of tables
- docs: remove trailing header
- docs: remove leading spaces
- docs: remove trailing newlines
- docs: use [1] instead of asterisk for footnote
- docs: remove pointless, stray newline
- docs: fixup indentation
- zink: implement nir_texop_txs
- zink: support offset-variants of texturing
- zink: avoid incorrect vector-construction
- zink: store image-type per texture
- zink: support sampling non-float textures
- zink: support arrays of samplers
- zink: set compareEnable when setting compareOp
- st/mesa: use uint-result for sampling stencil buffers
- Revert "nir: Add a couple trivial abs optimizations"

Florian Will (1):

- radv/winsys: set IB flags prior to submit in the sysmem path

Francisco Jerez (26):

- glsl: Fix software 64-bit integer to 32-bit float conversions.
- intel/fs/gen11+: Handle ROR/ROL in lower_simd_width().
- intel/fs/gen8+: Fix r127 dst/src overlap RA workaround for EOT message payload.
- intel/fs: Fix nir_intrinsic_load_barycentric_at_sample for SIMD32.
- intel/fs/cse: Fix non-deterministic behavior due to inaccurate liveness calculation.
- intel/fs: Make implied_mrf_writes() an fs_inst method.
- intel/fs: Try to vectorize header setup in lower_load_payload().
- intel/fs: Generalize fs_reg::is_contiguous() to register files other than VGRF.
- intel/fs: Rework fs_inst::is_copy_payload() into multiple classification helpers.
- intel/fs: Extend copy propagation dataflow analysis to copies with FIXED_GRF source.
- intel/fs: Add partial support for copy-propagating FIXED_GRFs.
- intel/fs: Add support for copy-propagating a block of multiple FIXED_GRFs.

- intel/fs: Allow limited copy propagation of a LOAD_PAYLOAD into another.
- intel/fs/gen4-6: Allocate registers from aligned_pairs_class based on LINTERP use.
- intel/fs/gen6: Constrain barycentric source of LINTERP during bank conflict mitigation.
- intel/fs/gen6: Generalize aligned_pairs_class to SIMD16 aligned barycentrics.
- intel/fs/gen6: Use SEL instead of bashing thread payload for unlit centroid workaround.
- intel/fs: Split fetch_payload_reg() into separate helper for barycentrics.
- intel/fs: Introduce barycentric layout lowering pass.
- intel/fs: Switch to standard vector layout for barycentrics at optimization time.
- intel/fs/cse: Make HALT instruction act as CSE barrier.
- intel/fs/gen7: Fix fs_inst::flags_written() for SHADER_OPCODE_FIND_LIVE_CHANNEL.
- intel/fs: Add virtual instruction to load mask of live channels into flag register.
- intel/fs/gen12: Workaround unwanted SEND execution due to broken NoMask control flow.
- intel/fs/gen12: Fixup/simplify SWSB annotations of SIMD32 scratch writes.
- intel/fs/gen12: Workaround data coherency issues due to broken NoMask control flow.

Fritz Koenig (1):

- freedreno: reorder format check

Georg Lehmann (3):

- Correctly wait in the fragment stage until all semaphores are signaled
- Vulkan Overlay: Don't try to change the image layout to present twice
- Vulkan overlay: use the corresponding image index for each swapchain

Gert Wollny (12):

- r600: Disable eight bit three channel formats
- virgl: Increase the shader transfer buffer by doubling the size
- gallium/tgsi_from_mesa: Add 'extern "C"' to be able to include from C++
- nir: make nir_get_texture_size/lod available outside nir_lower_tex
- gallium: tgsi_from_mesa - handle VARYING_SLOT_FACE
- r600: Add functions to dump the shader info
- r600: Make it possible to include r600_asm.h in a C++ file
- r600/sb: Correct SB disassambler for better debugging
- r600: Fix maximum line width
- r600: Make SID and unsigned value
- r600: Delete vertex buffer only if there is actually a shader state
- mesa/st: glsl_to_nir: don't lower atomics to SSBOs if driver supports HW atomics

Guido Günther (2):

- etnaviv: drm: Don't miscalculate timeout
- freedreno/drm: Don't miscalculate timeout

Gurchetan Singh (11):

- drirc: set allow_higher_compat_version for Faster Than Light
- virgl/drm: update UAPI
- teximage: split out helper from EGLImageTargetTexture2DOES
- glapi / teximage: implement EGLImageTargetTexStorageEXT
- dri_util: add driImageFormatToSizedInternalGLFormat function
- i965: track if image is created by a dmabuf
- i965: refactor intel_image_target_texture_2d
- i965: support EXT_EGL_image_storage
- st/dri: track if image is created by a dmabuf
- st/mesa: refactor egl image binding a bit
- st/mesa: implement EGLImageTargetTexStorage

Hyunjun Ko (7):

- freedreno/ir3: cleanup by removing repeated code
- freedreno: support 16b for the sampler opcode
- freedreno/ir3: fix printing output registers of FS.
- freedreno/ir3: fixup when changing to mad.f16
- freedreno/ir3: enable half precision for pre-fs texture fetch
- turnip: fix invalid VK_ERROR_OUT_OF_POOL_MEMORY
- freedreno/ir3: put the conversion back for half const to the right place.

Iago Toral Quiroga (32):

- v3d: rename vertex shader key (num)_fs_inputs fields
- mesa/st: make sure we remove dead IO variables before handing NIR to backends
- glsl: add missing initialization of the location path field
- v3d: fix indirect BO allocation for uniforms
- v3d: actually root the first BO in a command list in the job
- v3d: add missing plumbing for VPM load instructions
- v3d: add debug assert
- v3d: enable debug options for geometry shader dumps
- v3d: remove unused variable
- v3d: add initial compiler plumbing for geometry shaders
- v3d: fix packet descriptions for geometry and tessellation shaders
- v3d: emit geometry shader state commands
- v3d: implement geometry shader instancing
- v3d: add 1-way SIMD packing definition
- v3d: compute appropriate VPM memory configuration for geometry shader workloads

- v3d: we always have at least one output segment
- v3d: add support for adjacency primitives
- v3d: don't try to render if shaders failed to compile
- v3d: predicate geometry shader outputs inside non-uniform control flow
- v3d: save geometry shader state for blitting
- v3d: support transform feedback with geometry shaders
- v3d: remove obsolete assertion
- v3d: do not limit new CL space allocations with branch to 4096 bytes
- v3d: support rendering to multi-layered framebuffers
- v3d: move layer rendering to a separate helper
- v3d: handle writes to gl_Layer from geometry shaders
- v3d: fix primitive queries for geometry shaders
- v3d: disable lowering of indirect inputs
- v3d: support precompiling geometry shaders
- v3d: expose OES_geometry_shader
- u_vbuf: don't try to delete NULL driver CSO
- v3d: fix bug when checking result of syncobj fence import

Ian Romanick (39):

- intel/compiler: Report the number of non-spill/fill SEND messages on vec4 too
- nir/algebraic: Add the ability to mark a replacement as exact
- nir/algebraic: Mark other comparison exact when removing a == a
- intel/fs: Disable conditional discard optimization on Gen4 and Gen5
- nir/range-analysis: Add pragmas to help loop unrolling
- nir/range_analysis: Make sure the table validation only occurs once
- nir/opt_peephole_select: Don't count some unary operations
- intel/compiler: Increase nir_opt_peephole_select threshold
- nir/algebraic: Simplify some Inf and NaN avoidance code
- nir/algebraic: Rearrange bcsel sequences generated by nir_opt_peephole_select
- intel/compiler: Fix 'comparison is always true' warning
- mesa: Silence 'left shift of negative value' warning in BPTC compression code
- mesa: Silence unused parameter warning
- anv: Fix error message format string
- mesa: Extension boilerplate for INTEL_shader_integer_functions2
- glsl: Add new expressions for INTEL_shader_integer_functions2
- glsl_types: Add function to get an unsigned base type from a signed type
- glsl: Add built-in functions for INTEL_shader_integer_functions2

- nir: Add new instructions for INTEL_shader_integer_functions2
- nir/algebraic: Add lowering for uabs_usub and uabs_isub
- nir/algebraic: Add lowering for 64-bit hadd and rhadd
- nir/algebraic: Add lowering for 64-bit usub_sat
- nir/algebraic: Add lowering for 64-bit uadd_sat
- nir/algebraic: Add lowering for 64-bit iadd_sat and isub_sat
- compiler: Translate GLSL IR to NIR for new INTEL_shader_integer_functions2 expressions
- intel/fs: Don't lower integer multiplies that don't need lowering
- intel/fs: Add SHADER_OPCODE_[IU]SUB_SAT pseudo-ops
- intel/fs: Implement support for NIR opcodes for INTEL_shader_integer_functions2
- nir/spirv: Translate SPIR-V to NIR for new INTEL_shader_integer_functions2 opcodes
- spirv: Silence a bunch of unused parameter warnings
- spirv: Add support for IntegerFunctions2INTEL capability
- i965: Enable INTEL_shader_integer_functions2 on Gen8+
- gallium: Add a cap bit for OpenCL-style extended integer functions
- gallium: Add a cap bit for integer multiplication between 32-bit and 16-bit
- iris: Enable INTEL_shader_integer_functions2
- anv: Enable SPV_INTEL_shader_integer_functions2 and VK_INTEL_shader_integer_functions2
- nir/algebraic: Optimize some 64-bit integer comparisons involving zero
- relnotes: Add GL_INTEL_shader_integer_functions2 and VK_INTEL_shader_integer_functions2
- intel/fs: Don't count integer instructions as being possibly coissue

Icecream95 (16):

- gallium/auxiliary: Reduce conversions in u_vbuf_get_minmax_index_mapped
- gallium/auxiliary: Handle count == 0 in u_vbuf_get_minmax_index_mapped
- panfrost: Add negative lod bias support
- panfrost: Compact the bo_access readers array
- panfrost: Dynamically allocate shader variants
- panfrost: Add ETC1/ETC2 texture formats
- panfrost: Add ASTC texture formats
- pan/midgard: Fix bundle dynarray leak
- pan/midgard: Fix a memory leak in the disassembler
- pan/midgard: Support disassembling to a file
- pan/bifrost: Support disassembling to a file
- pan/decode: Support dumping to a file
- pan/decode: Dump to a file
- pan/decode: Rotate trace files

- panfrost: Don't copy uniforms when the size is zero
- pan/midgard: Fix a liveness info leak

Icenowy Zheng (2):

- lima: support indexed draw with bias
- lima: fix lima_set_vertex_buffers()

Ilia Mirkin (7):

- gm107/ir: fix loading z offset for layered 3d image bindings
- nv50/ir: mark STORE destination inputs as used
- nv50,nvc0: fix destination coordinates of blit
- nvc0: add dummy reset status support
- gm107/ir: avoid combining geometry shader stores at 0x60
- nvc0: treat all draws without color0 broadcast as MRT
- nvc0: disable xfb's which don't have a stride

Italo Nicola (1):

- intel/compiler: remove old comment

Iván Briano (4):

- intel/compiler: Don't change hstride if not needed
- anv: Export filter_minmax support only when it's really supported
- anv: Export VK_KHR_buffer_device_address only when really supported
- anv: Enable Vulkan 1.2 support

James Xiong (3):

- iris: try to set the specified tiling when importing a dmabuf
- gallium: dmabuf support for yuv formats that are not natively supported
- gallium: let the pipe drivers decide the supported modifiers

Jan Vesely (2):

- clover: Initialize Asm Parsers
- clover: Use explicit conversion from llvm::StringRef to std::string

Jan Zielinski (8):

- gallium/swr: Fix depth values for blit scenario
- swr/rasterizer: Add tessellator implementation to the rasterizer
- gallium/swr: Fix Windows build
- gallium/gallivm/tgsi: enable tessellation shaders
- gallium/gallivm: enable linking lp_bld_printf function with C++ code
- gallium/swr: implementation of tessellation shaders compilation
- gallium/swr: fix tessellation state save/restore
- docs: Update SWR tessellation support

Jason Ekstrand (212):

- util: Add a util_sparse_array data structure
- anv: Move refcount to anv_bo
- anv: Use a util_sparse_array for the GEM handle -> BO map
- anv: Fix a relocation race condition
- anv: Stop storing the GEM handle in anv_reloc_list_add
- anv: Declare the bo in the anv_block_pool_foreach_bo loop
- anv: Inline anv_block_pool_get_bo
- anv: Replace ANV_BO_EXTERNAL with anv_bo::is_external
- anv: Handle state pool relocations using "wrapper" BOs
- anv: Fix a potential BO handle leak
- anv: Rework anv_block_pool_expand_range
- anv: Use anv_block_pool_foreach_bo in get_bo_from_pool
- anv: Rework the internal BO allocation API
- anv: Choose BO flags internally in anv_block_pool
- anv/tests: Zero-initialize instances
- anv/tests: Initialize the BO cache and device mutex
- anv: Allocate block pool BOs from the cache
- anv: Use the query_slot helper in vkResetQueryPoolEXT
- anv: Allocate query pool BOs from the cache
- anv: Set more flags on descriptor pool buffers
- anv: Allocate descriptor buffers from the BO cache
- util: Add a free list structure for use with util_sparse_array
- anv: Allocate batch and fence buffers from the cache
- anv: Allocate scratch BOs from the cache
- anv: Allocate misc BOs from the cache
- anv: Drop anv_bo_init and anv_bo_init_new
- anv: Add a device parameter to anv_execbuf_add_bo
- anv: Set the batch allocator for compute pipelines
- anv: Use a bitset for tracking residency
- anv: Zero released anv_bo structs
- anv: Use the new BO alloc API for Android
- anv: Don't delete fragment shaders that write sample mask
- anv: Don't claim the null RT as a valid color target
- anv: Stop compacting render targets in the binding table
- anv: Move the RT BTI flush workaround to begin_subpass

- spirv: Remove the type from sampled_image
- spirv: Add a vtn_decorate_pointer helper
- spirv: Sort out the mess that is sampled image
- nir/builder: Add a nir_extract_bits helper
- nir: Add tests for nir_extract_bits
- intel/nir: Use nir_extract_bits in lower_mem_access_bit_sizes
- intel/fs: Add DWord scattered read/write opcodes
- intel/fs: refactor surface header setup
- intel/nir: Plumb devinfo through lower_mem_access_bit_sizes
- intel/fs: Implement the new load/store_scratch intrinsics
- intel/fs: Lower large local arrays to scratch
- anv: Lock around fetching sync file FDs from semaphores
- anv: Plumb timeline semaphore signal/wait values through from the API
- spirv: Fix the MSVC build
- anv/pipeline: Assume layout != NULL
- genxml: Mark everything in genX_pack.h always_inline
- anv: Input attachments are always single-plane
- anv: Flatten descriptor bindings in anv_nir_apply_pipeline_layout
- anv: Delete dead shader constant pushing code
- anv: Stop bounds-checking pushed UBOs
- anv: Pre-compute push ranges for graphics pipelines
- intel/compiler: Add a flag to avoid compacting push constants
- anv: Re-arrange push constant data a bit
- anv: Rework push constant handling
- anv: Use a switch statement for binding table setup
- anv: More carefully dirty state in BindDescriptorSets
- anv: More carefully dirty state in BindPipeline
- anv: Use an anv_state for the next binding table
- anv: Emit a NULL vertex for zero base_vertex/instance
- nir: Validate that variables are in the right lists
- iris: Re-enable param compaction
- Revert "i965/fs: Merge CMP and SEL into CSEL on Gen8+"
- vulkan/enum_to_str: Handle out-of-order aliases
- anv/entrypoints: Better handle promoted extensions
- vulkan: Update the XML and headers to 1.1.129
- anv: Push constants are relative to dynamic state on IVB

- anv: Set up SBE_SWIZ properly for gl_Viewport
- anv: Respect the always_flush_cache driconf option
- iris: Stop setting up fake params
- anv: Drop bo_flags from anv_bo_pool
- anv: Add a has_softpin boolean
- blorp: Pass the VB size to the VF cache workaround
- anv: Always invalidate the VF cache in BeginCommandBuffer
- anv: Apply cache flushes after setting index/draw VBs
- anv: Use PIPE_CONTROL flushes to implement the gen8 VF cache WA
- anv: Don't leak when set_tiling fails
- util/atomic: Add a _return variant of p_atomic_add
- anv: Disallow allocating above heap sizes
- anv: Stop tracking VMA allocations
- anv: Set up VMA heaps independently from memory heaps
- anv: Stop advertising two heaps just for the VF cache WA
- anv: Add an explicit_address parameter to anv_device_alloc_bo
- util/vma: Factor out the hole splitting part of util_vma_heap_alloc
- util/vma: Add a function to allocate a particular address range
- anv: Add allocator support for client-visible addresses
- anv: Use a pNext loop in AllocateMemory
- anv: Implement VK_KHR_buffer_device_address
- util/atomic: Add p_atomic_add_return for the unlocked path
- vulkan/wsi: Provide the implicitly synchronized BO to vkQueueSubmit
- vulkan/wsi: Add a hooks for signaling semaphores and fences
- anv: Always add in EXEC_OBJECT_WRITE when specified in extra_flags
- anv: Use submit-time implicit sync instead of allocate-time
- anv: Add a fence_reset_reset_temporary helper
- anv: Use BO fences/semaphores for AcquireNextImage
- anv: Return VK_ERROR_OUT_OF_DEVICE_MEMORY for too-large buffers
- anv: Re-capture all batch and state buffers
- anv: Re-emit all compute state on pipeline switch
- ANV: Stop advertising smoothLines support on gen10+
- anv: Flush the queue on DeviceWaitIdle
- anv: Unconditionally advertise Vulkan 1.1
- anv: Bump the advertised patch version to 129
- i965: Enable GL_EXT_gpu_shader4 on Gen6+

- anv: Properly advertise sampledImageIntegerSampleCounts
- anv: Drop unneeded struct keywords
- blorp: Stop whacking Z24 depth to BGRA8
- blorp: Allow reading with HiZ
- i965/blorp: Don't resolve HiZ unless we're reinterpreting
- intel/blorp: Use the source format when using blorp_copy with HiZ
- anv: Allow HiZ in TRANSFER_SRC_OPTIMAL on Gen8-9
- i965: Allow HiZ for glCopyImageSubData sources
- intel/nir: Add a memory barrier before barrier()
- intel/disasm: Fix decoding of src0 of SENDS
- genxml: Remove a non-existant HW bit
- anv: Don't add dynamic state base address to push constants on Gen7
- anv: Flag descriptors dirty when gl_NumWorkgroups is used
- anv: Re-use flush_descriptor_sets in flush_compute_state
- intel/vec4: Support scoped_memory_barrier
- nir: Handle more barriers in dead_write and copy_prop
- nir: Handle barriers with more granularity in combine_stores
- llmvpipe: No-op implement more barriers
- nir: Add a new memory_barrier_tcs_patch intrinsic
- spirv: Add a workaround for OpControlBarrier on old GLSLang
- spirv: Add output memory semantics to OpControlBarrier in TCS
- nir/glsl: Emit memory barriers as part of barrier()
- intel/nir: Stop adding redundant barriers
- nir: Rename nir_intrinsic_barrier to control_barrier
- nir/lower_atomics_to_ssbo: Also lower barriers
- anv: Drop an unused variable
- intel/blorp: Fill out all the dwords of MI_ATOMIC
- anv: Don't over-advertise descriptor indexing features
- anv: Memset array properties
- vulkan/wsi: Add a driconf option to force WSI to advertise BGRA8_UNORM first
- vulkan: Update the XML and headers to 1.2.131
- turnip: Pretend to support Vulkan 1.2
- anv: Bump the patch version to 131
- anv,nir: Lower quad_broadcast with dynamic index in NIR
- anv: Implement the new core version feature queries
- anv: Implement the new core version property queries

- relnotes: Add Vulkan 1.2
- anv: Drop some VK_IMAGE_TILING_OPTIMAL checks
- anv: Support modifiers in GetImageFormatProperties2
- vulkan/wsi: Move the ImageCreateInfo higher up
- vulkan/wsi: Use the interface from the real modifiers extension
- vulkan/wsi: Filter modifiers with ImageFormatProperties
- vulkan/wsi: Implement VK_KHR_swapchain_mutable_format
- anv/blorp: Rename buffer image stride parameters
- anv: Canonicalize buffer formats for image/buffer copies
- anv: Add an anv_physical_device field to anv_device
- anv: Take an anv_device in vk_errorf
- anv: Take a device in anv_perf_warn
- anv: Stop allocating WSI event fences off the instance
- anv: Drop the instance pointer from anv_device
- anv: Move the physical device dispatch table to anv_instance
- anv: Drop separate chipset_id fields
- anv: Re-arrange physical_device_init
- anv: Allow enumerating multiple physical devices
- anv/apply_pipeline_layout: Initialize the nir_builder before use
- intel/blorp: resize src and dst surfaces separately
- anv: Use TRANSFER_SRC_OPTIMAL for depth/stencil MSAA resolves
- anv: Add a layout_to_aux_state helper
- anv: Use isl_aux_state for HiZ resolves
- anv: Add a usage parameter to anv_layout_to_aux_usage
- anv: Allow HiZ in read-only depth layouts
- anv: Improve BTI change cache flushing
- intel/fs: Don't unnecessarily fall back to indirect sends on Gen12
- intel/disasm: Properly disassemble indirect SENDs
- intel/isl: Plumb devinfo into isl_genX(buffer_fill_state_s)
- intel/isl: Add a hack for the Gen12 A0 texture buffer bug
- anv: Rework the meaning of anv_image::planes[]::aux_usage
- anv: Replace aux_surface.isl.size_B checks with aux_usage checks
- intel/aux-map: Add some #defines
- intel/aux-map: Factor out some useful helpers
- anv: Delete a redundant calculation
- isl: Add a helper for calculating subimage memory ranges

- anv: Add another align_down helper
- anv: Make AUX table invalidate a PIPE_* bit
- anv: Make anv_vma_alloc/free a lot dumber
- anv: Rework CCS memory handling on TGL-LP
- intel/blorp: Add support for CCS_E copies with UNORM formats
- intel/isl: Allow CCS_E on more formats
- intel/genxml: Make SO_DECL::"Hole Flag" a Boolean
- anv: Insert holes for non-existant XFB varyings
- intel/blorp: Handle bit-casting UNORM and BGRA formats
- anv: Replace one more aux_surface.isl.size_B check
- intel/mi_builder: Force write completion on Gen12+
- anv: Set actual state pool sizes when we have softpin
- anv: Re-use one old BT block in reset_batch_bo_chain
- anv/block_pool: Ensure allocations have contiguous maps
- anv: Rename a variable
- genxml: Add a new 3DSTATE_SF field on gen12
- anv,iris: Set 3DSTATE_SF::DerefBlockSize to per-poly on Gen12+
- intel/genxml: Drop SLMEnable from L3CNTLREG on Gen11
- iris: Set SLMEnable based on the L3$ config
- iris: Store the L3$ configs in the screen
- iris: Use the URB size from the L3$ config
- i965: Re-emit l3 state before BLORP executes
- intel: Take a gen_l3_config in gen_get_urb_config
- intel/blorp: Always emit URB config on Gen7+
- iris: Consolodate URB emit
- anv: Emit URB setup earlier
- intel/common: Return the block size from get_urb_config
- intel/blorp: Plumb deref block size through to 3DSTATE_SF
- anv: Plumb deref block size through to 3DSTATE_SF
- iris: Plumb deref block size through to 3DSTATE_SF
- anv: Always fill out the AUX table even if CCS is disabled
- intel/fs: Write the address register with NoMask for MOV_INDIRECT
- anv/blorp: Use the correct size for vkCmdCopyBufferToImage

Jonathan Gray (4):

- winsys/amdgpu: avoid double simple_mtx_unlock()
- i965: update Makefile.sources for perf changes

- util/futex: use futex syscall on OpenBSD
- util/u_thread: don't restrict u_thread_get_time_nano() to __linux__

Jonathan Marek (98):

- freedreno: add Adreno 640 ID
- freedreno/ir3: disable texture prefetch for 1d array textures
- freedreno/registers: fix a6xx_2d_blit_cntl ROTATE
- etnaviv: blt: use only for tiling, and add missing formats
- etnaviv: separate PE and RS formats, use only RS only for tiling
- etnaviv: blt: set TS dirty after clear
- turnip: add display wsi
- turnip: add x11 wsi
- turnip: implement CmdClearColorImage/CmdClearDepthStencilImage
- turnip: fix sRGB GMEM clear
- util: add missing R8G8B8A8_SRGB format to vk_format_map
- freedreno/regs: update UBWC related bits
- turnip: implement UBWC
- etnaviv: avoid using RS for 64bpp formats
- etnaviv: implement 64bpp clear
- etnaviv: blt: fix partial ZS clears with TS
- etnaviv: support 3d/array/integer formats in texture descriptors
- turnip: fix integer render targets
- freedreno/registers: add missing MH perfcounter enum for a2xx
- freedreno/perfcntrs: add a2xx MH counters
- freedreno/perfcntrs/fdperf: fix u64 print on 32-bit builds
- freedreno/perfcntrs/fdperf: add missing a20x compatible
- freedreno/perfcntrs/fdperf: add missing a2xx case in select_counter
- turnip: fix display wsi fence timing out
- turnip: don't skip unused attachments when setting up tiling config
- turnip: implement CmdClearAttachments
- turnip: don't set unused BLIT_DST_INFO bits for GMEM clear
- turnip: MSAA resolve directly from GMEM
- turnip: allow writes to draw_cs outside of render pass
- turnip: add function to allocate aligned memory in a substream cs
- turnip: improve emit_textures
- turnip: implement border color
- turnip: add hw binning

- turnip: fix incorrectly failing assert
- freedreno/ir3: add GLSL_SAMPLER_DIM_SUBPASS to tex_info
- freedreno/registers: add a6xx texture format for stencil sampler
- turnip: fix hw binning render area
- turnip: fix tile layout logic
- turnip: update tile_align_w/tile_align_h
- turnip: set load_layer_id to zero
- turnip: set FRAG_WRITES_SAMPMASK bit
- turnip: fix VK_IMAGE_ASPECT_STENCIL_BIT image view
- turnip: no 8x msaa on 128bpp formats
- turnip: add dirty bit for push constants
- turnip: subpass rework
- turnip: CmdClearAttachments fixes
- turnip: implement subpass input attachments
- etnaviv: remove sRGB formats from format table
- etnaviv: sRGB render target support
- etnaviv: set output mode and saturate bits
- etnaviv: update INT_FILTER choice for GLES3 formats
- etnaviv: disable integer vertex formats on pre-HALTI2 hardware
- etnaviv: remove swizzle from format table
- etnaviv: add missing formats
- etnaviv: add missing vs_needs_z_div handling to NIR backend
- turnip: use single substream cs
- turnip: use common blit path for buffer copy
- turnip: don't require src image to be set for clear blits
- turnip: implement CmdFillBuffer/CmdUpdateBuffer
- freedreno/ir3: lower mul_2x32_64
- turnip: fix emit_textures for compute shaders
- turnip: remove compute emit_border_color
- turnip: fix emit_ibo
- turnip: change emit_ibo to be like emit_textures
- turnip: remove duplicate A6XX_SP_CS_CONFIG_NIBO
- nir: add option to lower half packing opcodes
- freedreno/ir3: lower pack/unpack ops
- turnip: don't set LRZ enable at end of renderpass
- freedreno/ir3: update prefetch input_offset when packing inlocs

- turnip: add cache invalidate to fix input attachment cases
- turnip: don't set SP_FS_CTRL_REG0_VARYING if only fragcoord is used
- freedreno/ir3: fix vertex shader sysvals with pre_assign_inputs
- freedreno/registers: document vertex/instance id offset bits
- freedreno/ir3: support load_base_instance
- turnip: emit base instance vs driver param
- turnip: emit_compute_driver_params fixes
- turnip: compute gmem offsets at renderpass creation time
- turnip: implement secondary command buffers
- nir: fix assign_io_var_locations for vertex inputs
- turnip: minor warning fixes
- util/format: add missing vulkan formats
- turnip: disable B8G8R8 vertex formats
- etnaviv: fix incorrectly failing vertex size assert
- etnaviv: update headers from rnndb
- etnaviv: HALTI2+ instanced draw
- etnaviv: implement gl_VertexID/gl_InstanceID
- etnaviv: remove unnecessary vertex_elements_state_create error checking
- st/mesa: don't lower YUV when driver supports it natively
- st/mesa: run st_nir_lower_tex_src_plane for lowered xyuv/ayuv
- freedreno/ir3: allow inputs with the same location
- turnip: remove tu_sort_variables_by_location
- turnip: fix array/matrix varyings
- turnip: hook up GetImageDrmFormatModifierPropertiesEXT
- turnip: set linear tiling for scanout images
- vulkan/wsi: remove unused image_get_modifier
- turnip: simplify tu_physical_device_get_format_properties
- etnaviv: implement UBOs
- turnip: hook up cmdbuffer event set/wait

Jordan Justen (7):

- iris: Add IRIS_DIRTY_RENDER_BUFFER state flag
- iris/gen11+: Move flush for render target change
- iris: Allow max dynamic pool size of 2GB for gen12
- intel: Remove unused Tigerlake PCI ID
- iris: Fix some indentation in iris_init_render_context
- iris: Emit CS Stall before Instruction Cache flush for gen12 WA

- anv: Emit CS Stall before Instruction Cache flush for gen12 WA

Jose Maria Casanova Crespo (1):

- v3d: Fix predication with atomic image operations

Juan A. Suarez Romero (3):

- nir/lower_double_ops: relax lower mod()
- Revert "nir/lower_double_ops: relax lower mod()"
- nir/spirv: skip unreachable blocks in Phi second pass

Kai Wasserbäch (4):

- nir: fix unused variable warning in nir_lower_vars_to_explicit_types
- nir: fix unused variable warning in find_and_update_previous_uniform_storage
- nir: fix unused function warning in src/compiler/nir/nir.c
- intel/gen_decoder: Fix unused-but-set-variable warning

Karol Herbst (14):

- nv50/ir: fix crash in isUniform for undefined values
- nir/validate: validate num_components on registers and intrinsics
- nir/serialize: fix vec8 and vec16
- nir/tests: add serializer tests
- nir/tests: MSVC build fix
- spirv: handle UniformConstant for OpenCL kernels
- clover/nir: treat UniformConstant as global memory
- clover/nir: set spirv environment to OpenCL
- clover/spirv: allow Int64 Atomics for supported devices
- nir: handle nir_deref_type_ptr_as_array in rematerialize_deref_in_block
- nv50/ir: implement global atomics and handle it for nir
- nir/serialize: cast swizzle before shifting
- aco: use NIR_MAX_VEC_COMPONENTS instead of 4
- nv50ir/nir: support vec8 and vec16

Kenneth Graunke (57):

- iris: Fix "Force Zero RTA Index Enable" setting again
- nir: Handle image arrays when setting variable data
- Revert "intel/blorp: Fix usage of uninitialized memory in key hashing"
- iris: Properly move edgeflag_out from output list to global list
- iris: Wrap iris_fix_edge_flags in NIR_PASS
- mesa: Handle GL_COLOR_INDEX in _mesa_format_from_format_and_type().
- iris: Change keybox parenting
- iris: Stop mutating the resource in get_rt_read_isl_surf().

- iris: Drop 'old_address' parameter from iris_rebind_buffer
- iris: Create an "iris_surface_state" wrapper struct
- iris: Maintain CPU-side SURFACE_STATE copies for views and surfaces.
- iris: Update SURFACE_STATE addresses when setting sampler views
- iris: Disable VF cache partial address workaround on Gen11+
- driconf, glsl: Add a vs_position_always_invariant option
- drirc: Set vs_position_always_invariant for Shadow of Mordor on Intel
- st/mesa: Add GL_TDFX_texture_compression_FXT1 support
- iris: Map FXT1 texture formats
- meson: Add a "prefer_iris" build option
- main: Change u_mmAllocMem align2 from bytes (old API) to bits (new API)
- meson: Include iris in default gallium-drivers for x86/x86_64
- util: Detect use-after-destroy in simple_mtx
- intel/genxml: Add a partial TCCNTLREG definition
- iris: Enable Gen11 Color/Z write merging optimization
- anv: Enable Gen11 Color/Z write merging optimization
- intel/decoder: Make get_state_size take a full 64-bit address and a base
- iris: Create smaller program keys without legacy features
- iris: Default to X-tiling for scanout buffers without modifiers
- iris: Alphabetize source files after iris_perf.c was added
- drirc: Final Fantasy VIII: Remastered needs allow_higher_compat_version
- iris: Make helper functions to turn iris shader keys into brw keys.
- iris: Fix shader recompile debug printing
- iris: Avoid replacing backing storage for buffers with no contents
- intel: Drop Gen11 WaBTPPrefetchDisable workaround
- st/nir: Optionally unify inputs_read/outputs_written when linking.
- iris: Set nir_shader_compiler_options::unify_interfaces.
- st/mesa: Allow ASTC5x5 fallbacks separately from other ASTC LDR formats.
- iris: Disable ASTC 5x5 support on Gen9 for now.
- iris: Delete remnants of the unimplemented ASTC 5x5 workaround
- iris: Allow HiZ for copy_region sources
- anv: Only enable EWA LOD algorithm when doing anisotropic filtering.
- Revert "nir: assert that nir_lower_tex runs after lowering derefs"
- i965: Simplify brw_get_renderer_string()
- iris: Simplify iris_get_renderer_string()
- intel: Use similar brand strings to the Windows drivers

- intel/compiler: Fix illegal mutation in get_nir_image_intrinsic_image
- iris: Fix export of fences that have already completed.
- st/mesa: Allocate full miplevels if MaxLevel is explicitly set
- iris: Drop some workarounds which are no longer necessary
- anv: Drop some workarounds that are no longer necessary
- intel: Fix aux map alignments on 32-bit builds.
- meson: Prefer 'iris' by default over 'i965'.
- loader: Check if the kernel driver is i915 before loading iris
- iris: Drop 'engine' from iris_batch.
- iris: Make iris_emit_default_l3_config pull devinfo from the batch
- iris: Support multiple chained batches.
- i965: Use brw_batch_references in tex_busy check
- loader: Fix leak of kernel driver name

Kristian Høgsberg (62):

- freedreno/registers: Fix typo
- freedreno/registers: Move SP_PRIMITIVE_CNTL and SP_VS_VPC_DST
- freedreno/registers: Add comments about primitive counters
- freedreno/a6xx: Fix primitive counters again
- freedreno/a6xx: Clear sysmem with CP_BLIT
- freedreno: Add nogmem debug option to force bypass rendering
- freedreno/a6xx: Fix layered texture type enum
- freedreno/a6x: Rename z/s formats
- freedreno/a6xx: Add register offset for STG/LDG
- freedreno/ir3: Emit link map as byte or dwords offsets as needed
- freedreno/ir3: Add load and store intrinsics for global io
- freedreno: Don't count primitives for patches
- freedreno/ir3: Add ir3 intrinsics for tessellation
- freedreno/ir3: Use imul24 in offset calculations
- freedreno/ir3: Add tessellation field to shader key
- freedreno/ir3: Extend geometry lowering pass to handle tessellation
- freedreno/ir3: Add new synchronization opcodes
- freedreno/ir3: End TES with chsh when using GS
- freedreno/ir3: Implement tess coord intrinsic
- freedreno/ir3: Implement TCS synchronization intrinsics
- freedreno/ir3: Setup inputs and outputs for tessellation stages
- freedreno/ir3: Don't assume binning shader is always VS

- freedreno/ir3: Pre-color TCS header and primitive ID inputs
- freedreno/ir3: Allocate const space for tessellation parameters
- freedreno/a6xx: Build the right draw command for tessellation
- freedreno/a6xx: Allocate and program tessellation buffer
- freedreno/a6xx: Emit constant parameters for tessellation stages
- freedreno/a6xx: Program state for tessellation stages
- freedreno: Use bypass rendering for tessellation
- freedreno/a6xx: Only set emit.hs/ds when we're drawing patches
- freedreno/blitter: Save tessellation state
- freedreno/a6xx: Only use merged regs and four quads for VS+FS
- freedreno/a6xx: Turn on tessellation shaders
- freedreno/ir3: Use regid() helper when setting up precolor regs
- freedreno/registers: Remove duplicate register definitions
- freedreno: New struct packing macros
- freedreno/registers: Add 64 bit address registers
- freedreno/a6xx: Drop stale include
- freedreno/a6xx: Include fd6_pack.h in a few files
- freedreno/a6xx: Convert emit_mrt() to OUT_REG()
- freedreno/a6xx: Convert emit_zs() to OUT_REG()
- freedreno/a6xx: Convert VSC pipe setup to OUT_REG()
- freedreno/a6xx: Convert gmem blits to OUT_REG()
- freedreno/a6xx: Convert some tile setup to OUT_REG()
- freedreno/a6xx: Silence warning for unused perf counters
- freedreno/a6xx: Document the CP_SET_DRAW_STATE enable bits
- freedreno/a6xx: Make DEBUG_BLIT_FALLBACK only dump fallbacks
- freedreno: Add debug flag for forcing linear layouts
- freedreno/a6xx: Program sampler swap based on resource tiling
- freedreno/a6xx: Pick blitter swap based on resource tiling
- freedreno/a6xx: Add fd_resource_swap() helper
- freedreno/a6xx: Use blitter for resolve blits
- freedreno/a6xx: RB6_R8G8B8 is actually 32 bit RGBX
- freedreno/a6xx: Use A6XX_SP_2D_SRC_FORMAT_MASK macro
- freedreno/a6xx: Handle srgb blits on the blitter
- freedreno/a6xx: Move handle_rgba_blit() up
- freedreno/a6xx: Rewrite compressed blits in a helper function
- freedreno/a6xx: Set up multisample sysmem MRTs correctly

- st/mesa: Lower vars to ssa and constant prop before gl_nir_lower_buffers
- ir3: Set up full/half register conflicts correctly
- iris: Advertise PIPE_CAP_NATIVE_FENCE_FD
- iris: Print warning and return *out = NULL when fd to syncobj fails

Krzysztof Raszkowski (10):

- gallium/swr: Fix GS invocation issues - Fixed proper setting gl_InvocationID. - Fixed GS vertices output memory overflow.
- gallium/swr: Enable some ARB_gpu_shader5 extensions Enable / add to features.txt: - Enhanced textureGather. - Geometry shader instancing. - Geometry shader multiple streams.
- gallium/swr: Fix crash when use GL_TDFX_texture_compression_FXT1 format.
- gallivm: add TGSI bit arithmetic opcodes support
- gallium/swr: Fix glVertexPointer race condition.
- gallium/swr: Disable showing detected arch message.
- docs/GL4: update gallium/swr features
- gallium/swr: add option for static link
- gallium/swr: Fix gcc 4.8.5 compile error
- gallium/swr: simplify environmental variabled expansion code

Lasse Lopperi (1):

- freedreno/drm: Fix memory leak in softpin implementation

Laurent Carlier (1):

- egl: avoid local modifications for eglext.h Khronos standard header file

Leo Liu (1):

- ac: add missing Arcturus to the info of pc lines

Lepton Wu (2):

- gallium: dri2: Use index as plane number.
- android: mesa: Revert “android: mesa: revert “Enable asm unconditionally”“

Lionel Landwerlin (60):

- intel/dev: set default num_eu_per_subslice on gen12
- intel/perf: add TGL support
- intel/perf: fix Android build
- mesa: check draw buffer completeness on glClearBufferfi/glClearBufferiv
- vulkan: bump headers/registry to 1.1.127
- anv: Properly handle host query reset of performance queries
- anv: implement VK_KHR_separate_depth_stencil_layouts
- mesa: check framebuffer completeness only after state update
- anv: invalidate file descriptor of semaphore sync fd at vkQueueSubmit
- anv: remove list items on batch fini

- anv: detach batch emission allocation from device
- anv: expose timeout helpers outside of anv_queue.c
- anv: move queue init/finish to anv_queue.c
- anv: allow NULL batch parameter to anv_queue_submit_simple_batch
- anv: prepare driver to report submission error through queues
- anv: refcount semaphores
- anv: prepare the driver for delayed submissions
- anv/wsi: signal the semaphore in the acquireNextImage
- anv: implement VK_KHR_timeline_semaphore
- intel/dev: flag the Elkhart Lake platform
- intel/perf: add EHL performance query support
- intel/perf: fix invalid hw_id in query results
- intel/perf: set read buffer len to 0 to identify empty buffer
- intel/perf: take into account that reports read can be fairly old
- intel/perf: simplify the processing of OA reports
- intel/perf: fix improper pointer access
- anv: fix missing gen12 handling
- anv: fix incorrect VMA alignment for CCS main surfaces
- anv: fix fence underlying primitive checks
- anv: fix assumptions about temporary fence payload
- intel/perf: drop batchbuffer flushing at query begin
- i965/iris: perf-queries: don't invalidate/flush 3d pipeline
- anv: constify pipeline layout in nir passes
- anv: drop unused parameter from apply layout pass
- vulkan/wsi: error out when image fence doesn't signal
- mesa: avoid triggering assert in implementation
- i965/iris/perf: factor out frequency register capture
- loader: fix close on uninitialized file descriptor value
- anv: don't close invalid syncfd semaphore
- anv: fix intel perf queries availability writes
- anv: set stencil layout for input attachments
- iris: Implement Gen12 workaround for non pipelined state
- anv: Implement Gen12 workaround for non pipelined state
- anv: only use VkSamplerCreateInfo::compareOp if enabled
- anv: fix pipeline switch back for non pipelined states
- genxml: add new Gen11+ PIPE_CONTROL field

- iris: handle new PIPE_CONTROL field
- iris: implement another workaround for non pipelined states
- anv: implement another workaround for non pipelined states
- intel/perf: expose timestamp begin for mdapi
- intel/perf: report query split for mdapi
- anv: enable VK_KHR_swapchain_mutable_format
- anv: don't report error with other vendor DRM devices
- anv: ensure prog params are initialized with 0s
- anv/iris: warn gen12 3DSTATE_HS restriction
- intel: Implement Gen12 workaround for array textures of size 1
- isl: drop CCS row pitch requirement for linear surfaces
- isl: add gen12 comment about CCS for linear tiling
- anv: implement gen9 post sync pipe control workaround
- anv: set MOCS on push constants

Luis Mendes (1):

- radv: fix radv secure compile feature breaks compilation on armhf EABI and aarch64

Marco Felsch (1):

- etnaviv: Fix assert when try to accumulate an invalid fd

Marek Olšák (245):

- glsl: encode/decode types using a union with bitfields for readability
- glsl: encode vector_elements and matrix_columns better
- glsl: encode explicit_stride for basic types better
- glsl: encode array types better
- glsl: encode struct/interface types better
- st/mesa: call nir_opt_access only once
- st/mesa: call nir_lower_flrp only once per shader
- compiler: make variable::data::binding unsigned
- nir: pack nir_variable::data::stream
- nir: pack nir_variable::data::xfb_*
- radeonsi: use IR SHA1 as the cache key for the in-memory shader cache
- radeonsi: don't keep compute shader IR after compilation
- radeonsi: keep serialized NIR instead of nir_shader in si_shader_selector
- nir: pack the rest of nir_variable::data
- nir/serialize: don't expand 16-bit variable state slots to 32 bits
- nir/serialize: store 32-bit object IDs instead of 64-bit
- nir/serialize: pack nir_variable flags

- mesa: expose SPIR-V extensions in the Compatibility profile too
- util: add blob_finish_get_buffer
- radeonsi/nir: call nir_serialize only once per shader
- radeonsi/nir: fix compute shader crash due to nir_binary == NULL
- glsl/linker: pass shader_info to analyze_clip_cull_usage directly
- compiler: pack shader_info from 160 bytes to 96 bytes
- st/mesa: fix Sanctuary and Tropics by disabling ARB_gpu_shader5 for them
- st/mesa: rename DEBUG_TGSI -> DEBUG_PRINT_IR
- st/mesa: remove \n being only printed in debug builds after printed TGSI
- st/mesa: print TCS/TES/GS/CS TGSI in the right place & keep disk cache enabled
- st/mesa: add ST_DEBUG=nir to print NIR shaders
- st/mesa: remove unused TGSI-only debug printing functions
- gallium/noop: call finalize_nir
- radeonsi/nir: remove dead function temps
- radeonsi/nir: call nir_lower_flrp only once per shader
- radeonsi/nir: don't lower fma, instead, fuse fma
- mesa: enable glthread for 7 Days To Die
- st/mesa: rename delete_basic_variant -> delete_common_variant
- st/mesa: decrease the size of st_fp_variant_key from 48 to 40 bytes
- st/mesa: start deduplicating some program code
- st/mesa: initialize affected_states and uniform storage earlier in deserialize
- st/mesa: consolidate and simplify code flagging program::affected_states
- st/mesa: trivially merge st_vertex_program into st_common_program
- st/mesa: rename st_common_program to st_program
- st/mesa: cleanups after unification of st_vertex/common program
- st/mesa: rename occurences of stcp to stp to correspond to st_program
- st/mesa: more cleanups after unification of st_vertex/common_program
- st/mesa: subclass st_vertex_program for VP-specific members
- st/mesa: call nir_sweep in st_finalize_nir
- st/mesa: keep serialized NIR instead of nir_shader in st_program
- st/mesa: call nir_serialize only once per shader
- nir: move data.image.access to data.access
- nir/print: only print image.format for image variables
- glsl_to_nir: rename image_access to mem_access
- nir: move data.descriptor_set above data.index for better packing
- nir: don't use GLenum16 in nir.h

- ac: add radeon_info::num_rings and move ring_type to amd_family.h
- ac: fill num_rings for remaining IPs
- winsys/amdgpu: detect noop dependencies on the same ring correctly
- nir: strip as we serialize to remove the nir_shader_clone call
- nir/serialize: do ctx = {0} instead of manual initializations
- util/blob: add 8-bit and 16-bit reads and writes
- nir/serialize: pack instructions better
- nir/serialize: pack src better and limit the object count to 1M from 1G
- nir/serialize: don't serialize var->data for temporaries
- nir/serialize: deduplicate serialized var types by reusing the last unique one
- nir/serialize: try to store a diff in var data locations instead of var data
- nir/serialize: pack load_const with non-64-bit constants better
- nir/serialize: pack 1-component constants into 20 bits if possible
- nir/serialize: pack nir_intrinsic_instr::const_index[] better
- nir/serialize: try to pack two alu srcs into 1 uint32
- nir/serialize: don't store deref types if not needed
- nir/serialize: don't serialize mode for deref non-cast instructions
- nir/serialize: try to put deref->var index into the unused bits of the header
- nir/serialize: cleanup - fold nir_deref_type_var cases into switches
- nir/serialize: try to pack both deref array src into 32 bits
- nir/serialize: remove up to 3 consecutive equal ALU instruction headers
- nir/serialize: reuse the writemask field for 2 src X swizzles of SSA ALU
- nir/serialize: serialize swizzles for vec8 and vec16
- nir/serialize: serialize writemask for vec8 and vec16
- nir/serialize: don't serialize redundant nir_intrinsic_instr::num_components
- nir/serialize: use 3 unused bits in intrinsic for packed_const_indices
- nir/serialize: support any num_components for remaining instructions
- ac: set swizzled bit in cache policy as a hint not to merge loads/stores
- radeonsi: initialize the per-context compiler on demand
- radeonsi/nir: don't run si_nir_opts again if there is no change
- st/mesa: don't serialize all streamout state if there are no SO outputs
- st/mesa: don't use redundant stp->state.ir.nir
- st/mesa: don't call ProgramStringNotify in glsl_to_nir
- st/mesa: propagate gl_PatchVerticesIn from TCS to TES before linking for NIR
- st/mesa: simplify looping over linked shaders when linking NIR
- st/mesa: don't use ** in the st_nir_link_shaders signature

- st/mesa: add st_variant base class to simplify code for shader variants
- ac/nir: don't rely on data.patch for tess factors
- radeonsi/nir: implement subgroup system values for SPIR-V
- radeonsi: simplify the interface of get_dw_address_from_generic_indices
- radeonsi: simplify get_tcs_tes_buffer_address_from_generic_indices
- radeonsi/nir: validate is_patch because SPIR-V doesn't set it for tess factors
- radeonsi/nir: don't rely on data.patch for tess factors
- radeonsi/nir: fix location_frac handling for TCS outputs
- radeonsi/nir: support interface output types to fix SPIR-V xfb piglits
- radeonsi: enable SPIR-V and GL 4.6 for NIR
- util/driconfig: print ATTENTION if MESA_DEBUG=silent is not set
- radeonsi/gfx10: simplify some duplicated NGG GS code
- radeonsi/gfx10: fix the vertex order for triangle strips emitted by a GS
- llvmpipe: implement TEX_LZ and TXF_LZ opcodes
- gallivm: implement LOAD with CONSTBUF but don't enable it for llvmpipe
- st/mesa: support UBOs for Selection/Feedback/RasterPos
- st/mesa: save currently bound vertex samplers and sampler views in st_context
- st/mesa: support samplers for Selection/Feedback/RasterPos
- st/mesa: support SSBOs for Selection/Feedback/RasterPos
- st/mesa: support shader images for Selection/Feedback/RasterPos
- st/mesa: use a separate VS variant for the draw module
- st/mesa: remove st_vp_variant::num_inputs
- st/mesa: remove struct st_vp_variant in favor of st_common_variant
- st/mesa: don't generate VS TGSI if NIR is enabled
- draw, st/mesa: generate TGSI for ffvp/ARB_vp if draw lacks LLVM
- st/mesa: release the draw shader properly to fix driver crashes (iris)
- st/dri: assume external consumers of back buffers can write to the buffers
- radeonsi: enable NIR by default and document GL 4.6 support
- radeonsi/gfx10: disable vertex grouping
- radeonsi/gfx10: simplify the tess_turns_off_ngg condition
- radeonsi: don't rely on CLEAR_STATE to set PA_SC_GENERIC_SCISSOR_*
- ac: fix ac_get_i1_sgpr_mask for Wave32
- ac: fix the return value in cull_bbox when bbox culling is disabled
- radeonsi: deduplicate ES and GS thread enablement code
- radeonsi: disallow compute-based culling if polygon mode is enabled
- radeonsi: set is_monolithic for VS prologs when the shader is really monolithic

- radeonsi: don't wrap the VS prolog in if (ES thread) .. endif
- radeonsi/gfx10: don't insert NGG streamout atomics if they are never used
- radeonsi: allow generating VS prologs with 0 inputs
- radeonsi: fix determining whether the VS prolog is needed
- radeonsi: reset more fields in si_llvm_context_set_ir to fix reusing ctx
- radeonsi/gfx10: fix ngg_get_ordered_id
- amd/addrlib: update to the latest version
- ac/surface: fix an assertion failure on gfx9 in CMASK computation
- radeonsi/gfx10: don't declare any LDS for NGG if it's not used
- radeonsi/gfx10: enable NGG passthrough for eligible shaders
- radeonsi/gfx10: improve performance for TES using PrimID but not exporting it
- Revert "u_vbuf: Regard non-constant vbufs with non-instance elements as free"
- winsys/radeon: initialize pte_fragment_size
- radeonsi: preserve the scanout flag for shared resources on gfx9 and gfx10
- radeonsi: ignore PIPE_BIND_SCANOUT for imported textures
- radeonsi: remove the "display_dcc_offset == 0" assertion
- radeonsi: rename SDMA debug flags
- radeonsi: remove broken and unused SI SDMA image copy code
- radeonsi: add AMD_DEBUG=nodmaclear for debugging
- radeonsi: add AMD_DEBUG=nodmacopyimage for debugging
- radeonsi: rename dma_cs -> sdma_cs
- radeonsi: move SI and CIK+ SDMA code into 1 common function for cleanups
- radeonsi: disable SDMA on gfx8 to fix corruption on RX 580
- radeonsi: remove TGSI
- gallium: put u_vbuf_get_caps return values into u_vbuf_caps
- gallium/cso_context: move non-vbuf vertex buffer and element code into helpers
- gallium: bypass u_vbuf if it's not needed (no fallbacks and no user VBOs)
- ac/gpu_info: always use distributed tessellation on gfx10
- radeonsi: fix monolithic pixel shaders with two-sided colors and SampleMaskIn
- radeonsi: fix context roll tracking in si_emit_shader_vs
- radeonsi: test polygon mode enablement accurately
- radeonsi: determine accurately if line stippling is enabled for performance
- radeonsi: clean up messy si_emit_rasterizer_prim_state
- ac: unify build_sendmsg_gs_alloc_req
- ac: unify primitive export code
- ac/gpu_info: add pc_lines and use it in radeonsi

- ac: add 128-bit bitcount
- ac: add ac_build_s_endpgm
- radeonsi/gfx9: force the micro tile mode for MSAA resolve correctly on gfx9
- radeonsi: rename desc_list_byte_size -> vb_desc_list_alloc_size
- radeonsi: add si_context::num_vertex_elements
- radeonsi: don't allow draw calls with uninitialized VS inputs
- radeonsi: simplify si_set_vertex_buffers
- ac,radeonsi: increase the maximum number of shader args and return values
- radeonsi: put up to 5 VBO descriptors into user SGPRs
- radeonsi: don't enable VBOs in user SGPRs if compute-based culling can be used
- radeonsi: fix assertion and other failures in si_emit_graphics_shader_pointers
- radeonsi: actually enable VBOs in user SGPRs
- radeonsi: don't adjust depth and stencil PS output locations
- radeonsi: rename DBG_NO_TGSI -> DBG_NO_NIR
- radeonsi: remove TGSI from comments
- radeonsi: rename si_shader_info -> si_shader_binary_info
- radeonsi: fork tgsi_shader_info and tgsi_tessctrl_info
- radeonsi: merge si_tessctrl_info into si_shader_info
- radeonsi: clean up si_shader_info
- radeonsi: rename si_compile_tgsi_main -> si_build_main_function
- radeonsi: rename si_shader_create -> si_create_shader_variant for clarity
- radeonsi: fold si_create_function into si_llvm_create_func
- radeonsi: remove always constant ballot_mask_bits from si_llvm_context_init
- radeonsi: move PS LLVM code into si_shader_llvm_ps.c
- radeonsi: separate code computing info for small primitive culling
- ac/cull: don't read Position.Z if it's not needed for culling
- radeonsi: make si_insert_input_* functions non-static
- radeonsi: move VS_STATE.LS_OUT_PATCH_SIZE a few bits higher to make space there
- radeonsi/gfx10: separate code for getting edgeflags from the gs_invocation_id VGPR
- radeonsi/gfx10: separate code for determining the number of vertices for NGG
- radeonsi: fix si_build_wrapper_function for compute-based primitive culling
- radeonsi: work around an LLVM crash when using llvm.amdgcn.icmp.i64.i1
- radeonsi: move si_insert_input_* functions
- radeonsi: move tessellation shader code into si_shader_llvm_tess.c
- radeonsi: remove llvm_type_is_64bit
- radeonsi: move geometry shader code into si_shader_llvm_gs.c

- radeonsi: move code for shader resources into si_shader_llvm_resources.c
- radeonsi: remove useless #includes
- radeonsi: merge si_compile_llvm and si_llvm_compile functions
- gallium: add st_context_iface::flush_resource to call FLUSH_VERTICES
- st/dri: do FLUSH_VERTICES before calling flush_resource
- Revert "radeonsi: unbind image before compute clear"
- radeonsi: clean up how internal compute dispatches are handled
- radeonsi: don't invoke decompression inside internal launch_grid
- radeonsi: fix doubles and int64
- radeonsi: turn an assertion into return in si_nir_store_output_tcs
- ac: add prefix bitcount functions
- ac: add ac_build_readlane without optimization barrier
- radeonsi/gfx10: update comments and remove invalid TODOs
- radeonsi/gfx10: correct VS PrimitiveID implementation for NGG
- radeonsi/gfx10: move s_sendmsg gs_alloc_req to the beginning of shaders
- radeonsi/gfx10: export primitives at the beginning of VS/TES
- radeonsi/gfx10: merge main and pos/param export IF blocks into one if possible
- radeonsi/gfx10: don't initialize VGPRs not used by NGG passthrough
- radeonsi/gfx10: move GE_PC_ALLOC setting to shader states
- radeonsi/gfx10: implement NGG culling for 4x wave32 subgroups
- ac: add helper ac_build_triangle_strip_indices_to_triangle
- radeonsi/gfx10: rewrite late alloc computation
- radeonsi/gfx10: enable GS fast launch for triangles and strips with NGG culling
- radeonsi: use ctx->ac. for types and integer constants
- radeonsi: move non-LLVM code out of si_shader_llvm.c
- radeonsi: move VS shader code into si_shader_llvm_vs.c
- radeonsi: move si_shader_llvm_build.c content into si_shader_llvm.c
- radeonsi: minor cleanup in si_shader_internal.h
- radeonsi: move si_nir_build_llvm into si_shader_llvm.c
- radeonsi: fold si_shader_context_set_ir into si_build_main_function
- radeonsi: move more LLVM functions into si_shader_llvm.c
- radeonsi: make si_compile_llvm return bool
- radeonsi: make si_compile_shader return bool
- radeonsi: change prototypes of si_is_multi_part_shader & si_is_merged_shader
- radeonsi: separate LLVM compilation from non-LLVM code
- util/simple_mtx: add a missing include to get ASSERTED

- gallium/util: add a cache of live shaders for shader CSO deduplication
- radeonsi: use the live shader cache
- radeonsi: restructure si_shader_cache_load_shader
- radeonsi: print shader cache stats with AMD_DEBUG=cache_stats
- radeonsi: expose shader cache stats to the HUD
- radeonsi: make screen available to shader part compilation
- radeonsi: fix a regression since the addition of si_shader_llvm_vs.c
- Revert "winsys/amdgpu: Close KMS handles for other DRM file descriptions"
- Revert "winsys/amdgpu: Re-use amdgpu_screen_winsys when possible"
- radeonsi: don't report that multi-plane formats are supported
- radeonsi: fix the DCC MSAA bug workaround
- radeonsi: don't wait for shader compilation to finish when destroying a context

Marek Vasut (5):

- etnaviv: Replace bitwise OR with logical OR
- etnaviv: tgsi: Fix gl_FrontFacing support
- etnaviv: Report correct number of vertex buffers
- etnaviv: Do not filter out PIPE_FORMAT_S8_UINT_Z24_UNORM on pre-HALTI2
- etnaviv: Destroy rsc->pending_ctx set in etna_resource_destroy()

Mark Janes (3):

- Revert "st/mesa: call nir_serialize only once per shader"
- Revert "st/mesa: keep serialized NIR instead of nir_shader in st_program"
- iris: separating out common perf code

Markus Wick (3):

- mapi/glapi: Generate sizeof() helpers instead of fixed sizes.
- mesa/glthread: Implement ARB_multi_bind.
- drirc: Enable glthread for dolphin/citra/yuzu.

Martin Fuzzey (1):

- etnaviv: update Android build files

Mathias Fröhlich (1):

- egl: Implement getImage/putImage on pbuffer swrast.

Matt Turner (19):

- intel/compiler: Use ARRAY_SIZE()
- intel/compiler: Extract GEN_* macros into separate file
- intel/compiler: Split has_64bit_types into float/int
- intel/compiler: Don't disassemble align1 3-src operands on Gen < 10
- intel/compiler: Limit compaction unit tests to specific gens

- intel/compiler: Add NF some more places
- intel/compiler: Add a INVALID_{,HW_}REG_TYPE macros
- intel/compiler: Split hw_type tables
- intel/compiler: Handle invalid inputs to brw_reg_type_to_*()
- intel/compiler: Handle invalid compacted immediates
- intel/compiler: Factor out brw_validate_instruction()
- intel/compiler: Validate some instruction word encodings
- intel/compiler: Add unit tests for new EU validation checks
- intel/compiler: Validate fuzzed instructions
- intel/compiler: Test compaction on Gen <= 12
- gitlab-ci: Skip ext_timer_query/time-elapsed
- intel/compiler: Move Gen4/5 rounding to visitor
- util: Explain BITSET_FOREACH_SET params
- util: Remove tmp argument from BITSET_FOREACH_SET macro

Mauro Rossi (9):

- android: aco: fix Lower to CSSA
- android: radeonsi: fix build error due to wrong u_format.csv file path
- android: util/format: fix include path list
- android: radeonsi: fix build after vl refactoring (v2)
- android: nir: add a load/store vectorization pass
- android: util: Add a mapping from VkFormat to PIPE_FORMAT.
- android: radv: fix vk_format_table.c generated source build
- android: radeonsi,ac: fix building error due to ac changes
- android: radv: build radv_shader_args.c

Michel Dänzer (36):

- gitlab-ci: Set arm job CCACHE_DIR properly
- gitlab-ci: Use separate arm64 build/test docker images
- gitlab-ci: Don't build libdrm for ARM
- gitlab-ci: Use ninja -j4 for building dEQP
- gitlab-ci: Move artifact preparation to separate script
- gitlab-ci: Share dEQP build process between x86 & ARM test image scripts
- gitlab-ci: Sort packages in debian-install.sh
- gitlab-ci: Run piglit tests with llvmpipe
- gitlab-ci: Use separate docker images for x86 build/test jobs
- gitlab-ci: Delete install/bin from artifacts as well
- gitlab-ci: Document that ci-templates refs must be in sync

- gitlab-ci: Use functional container job names
- gitlab-ci: Rename container install scripts to match job names (better)
- gitlab-ci: Organize images using new REPO_SUFFIX templates feature
- gitlab-ci: Directly use host-mapped directory for ccache
- gitlab-ci: Stop reporting piglit test results via JUnit
- gitlab-ci: Stop storing piglit test results as JUnit
- gitlab-ci: Put HTML summary in artifacts for failed piglit jobs
- gitlab-ci: Update to current ci-templates master
- gitlab-ci: Run piglit glslparser & quick_shader tests separately
- glsl/tests: Use splitlines() instead of strip()
- gitlab-ci: Use the common run policy for LAVA jobs as well again
- gitlab-ci: Overhaul job run policy
- gitlab-ci: Don't exclude any piglit quick_shader tests
- gitlab-ci: Test against LLVM / clang 9 on x86
- gitlab-ci: Stop using manual jobs for merge requests
- gitlab-ci: Set GIT_STRATEGY to none for the dummy job
- gitlab-ci: Use single if for manual job rules entry
- winsys/amdgpu: Keep a list of amdgpu_screen_winsyses in amdgpu_winsys
- winsys/amdgpu: Keep track of retrieved KMS handles using hash tables
- winsys/amdgpu: Only re-export KMS handles for different DRM FDs
- util: Add os_same_file_description helper
- winsys/amdgpu: Re-use amdgpu_screen_winsys when possible
- winsys/amdgpu: Close KMS handles for other DRM file descriptions
- winsys/amdgpu: Re-use amdgpu_screen_winsys when possible
- winsys/amdgpu: Close KMS handles for other DRM file descriptions

Michel Zou (3):

- Meson: Check for dladdr with MinGW
- disk_cache_get_function_timestamp: check for dladdr
- Meson: Add llvm>=9 modules

Miguel Casas-Sanchez (1):

- i965: Ensure that all 2101010 image imports can pass framebuffer completeness.

Nanley Chery (3):

- gallium/dri2: Fix creation of multi-planar modifier images
- gallium: Store the image format in winsys_handle
- iris: Fix import of multi-planar surfaces with modifiers

Nataraj Deshpande (1):

- egl/android: Restrict minimum triple buffering for android color_buffers

Nathan Kidd (1):

- llvmpipe: Check thread creation errors

Neha Bhende (3):

- st/mesa: release tgsi tokens for shader states
- svga: fix size of format_conversion_table[]
- svga: Use pipe_shader_state_from_tgsi to set shader state

Neil Armstrong (3):

- Add support for T820 CI Jobs
- ci: Remove T820 from CI temporarily
- gitlab-ci/lava: add pipeline information in the lava job name

Neil Roberts (9):

- nir/opcodes: Add a helper function to generate the comparison binops
- nir/opcodes: Add a helper function to generate reduce opcodes
- nir: Add a 16-bit bool type
- nir: Add a 8-bit bool type
- nir/lower_alu_to_scalar: Support lowering 8- and 16-bit reduce ops
- freedreno/ir3: Support 16-bit comparison instructions
- freedreno/ir3: Add implementation of nir_op_b16csel
- freedreno/ir3: Implement f2b16 and i2b16
- freedreno/ir3: Enabling lowering 16-bit flrp

Paul Cercueil (5):

- kmsro: Extend to include ingenic-drm
- u_vbuf: Mark vbufs incompatible if more were requested than HW supports
- u_vbuf: Only create driver CSO if no incompatible elements
- u_vbuf: Regard non-constant vbufs with non-instance elements as free
- u_vbuf: Return true in u_vbuf_get_caps if nb of vbufs is below minimum

Paul Gofman (1):

- state_tracker: Handle texture view min level in st_generate_mipmap()

Paulo Zanoni (2):

- intel/compiler: remove the operand restriction for src1 on GLK
- intel/compiler: fix nir_op_{i,u}*32 on ICL

Peng Huang (1):

- radeonsi: make si_fence_server_signal flush pipe without work

Philipp Sieweck (1):

- svga: check return value of define_query_vgpu{9,10}

Pierre Moreau (4):

- compiler/spirv: Fix uses of gnu struct = { } extension
- include/CL: Update OpenCL headers to latest
- clover: Use the dispatch table type from the OpenCL headers
- clover/meson: Define OpenCL header macros

Pierre-Eric Pelloux-Prayer (54):

- radeonsi: tell the shader disk cache what IR is used
- mesa: enable msaa in clear_with_quad if needed
- mesa: pass vao as a function paramter
- mesa: add EXT_dsa glVertexArray* functions declarations
- mesa: rework _mesa_lookup_vao_err to allow usage from EXT_dsa
- mesa: add vao/vbo lookup helper for EXT_dsa
- mesa: add EXT_dsa glVertexArray* functions implementation
- mesa: add gl_vertex_array_object parameter to client state helpers
- mesa: add EXT_dsa glEnableVertexArrayEXT / glDisableVertexArrayEXT
- mesa: add EXT_dsa EnableVertexArrayAttribEXT / DisableVertexArrayAttribEXT
- mesa: extract helper function from _mesa_GetPointerv
- mesa: add EXT_dsa glGetVertexArray* 4 functions
- mesa: fix call to _mesa_lookup_vao_err
- radeonsi: fix shader disk cache key
- radeonsi: enable mesa_glthread for GfxBench
- mesa: update features.txt to reflect EXT_dsa status
- mesa: add ARB_framebuffer_no_attachments named functions
- mesa: add ARB_vertex_attrib_64bit VertexArrayVertexAttribLOffsetEXT
- mesa: add ARB_clear_buffer_object named functions
- mesa: add ARB_gpu_shader_fp64 selector-less functions
- mesa: add ARB_instanced_arrays EXT_dsa function
- mesa: add ARB_texture_buffer_range glTextureBufferRangeEXT function
- mesa: implement ARB_texture_storage_multisample + EXT_dsa functions
- mesa: extend vertex_array_attrib_format to support EXT_dsa
- mesa: add ARB_vertex_attrib_binding glVertexArray* functions
- mesa: add ARB_sparse_buffer NamedBufferPageCommitmentEXT function
- mesa: enable EXT_direct_state_access
- mesa: fix warning in 32 bits build
- radeonsi: implement sdma for GFX9
- radeonsi: display cs blit count for AMD_DEBUG=testdma

- radeonsi: use gfx9.surf_offset to compute texture offset
- radeonsi: fix multi plane buffers creation
- radeonsi: dcc dirty flag
- st/mesa: add a notify_before_flush callback param to flush
- st/dri: use st->flush callback to flush the backbuffer
- radeonsi: disable dcc for 2x MSAA surface and bpe < 4
- gallium: refuse to create buffers larger than UINT32_MAX
- radeon/vcn2: enable rate control for hevc encoding
- radeonsi: check ctx->sdma_cs before using it
- radeonsi: release saved resources in si_retile_dcc
- radeonsi: release saved resources in si_compute_expand_fmask
- radeonsi: release saved resources in si_compute_clear_render_target
- radeonsi: release saved resources in si_compute_copy_image
- radeonsi: release saved resources in si_compute_clear_12bytes_buffer
- radeonsi: release saved resources in si_compute_do_clear_or_copy
- radeonsi: fix fmask expand compute shader
- radeonsi: make sure fmask expand is done if needed
- radeonsi: unbind image before compute clear
- radeonsi: drop the negation from fmask_is_not_identity
- util: call bind_sampler_states before setting sampler_views
- radeonsi: move AMD_DEBUG tests to AMD_TEST
- docs: document AMD_DEBUG variable
- radeonsi: stop using the VM_ALWAYS_VALID flag
- radeonsi/ngg: add VGT_FLUSH when enabling fast launch

Prodea Alexandru-Liviu (2):

- Meson: Remove lib prefix from graw and osmesa when building with Mingw. Also remove version sufix from osmesa swrast on Windows.
- Appveyor: Quickly fix meson build. As this required use of Python 3.8, mako module also had to be updated.

Qiang Yu (3):

- lima: sync lima_drm.h with kernel
- lima: create heap buffer with new interface if available
- lima: add noheap debug option

Rafael Antognolli (23):

- intel/isl: Add MOCS settings to isl_device.
- anv: Use mocs settings from isl_dev.
- iris: Use mocs from isl_dev.

- intel: Add workaround for stencil state.
- intel/genxml: Add 3DSTATE_CONSTANT_ALL packet.
- intel/aubinator: Decode 3DSTATE_CONSTANT_ALL.
- intel/blorp: Use 3DSTATE_CONSTANT_ALL to setup push constants.
- iris: Rework push constants emitting code.
- iris: Use 3DSTATE_CONSTANT_ALL when possible.
- anv: Move gen8+ push constant packet workaround.
- anv: Add get_push_range_address() helper.
- anv: Move code for emitting push constants into its own function.
- anv: Use 3DSTATE_CONSTANT_ALL when possible.
- iris: Add restriction to 3DSTATE_CONSTANT_ packets.
- util/os_socket: Add socket related functions.
- vulkan/overlay: Add a control socket.
- vulkan/overlay: Add support for a control socket.
- vulkan/overlay: Add a command to start capturing data to a file.
- vulkan/overlay: Add basic overlay control script.
- vulkan/overlay: Update docs.
- iris: Implement WA for push constants.
- utils/os_socket: Define ssize_t on windows.
- intel: Load the driver even if I915_PARAM_REVISION is not found.

Rhys Perry (131):

- radv: adjust loop unrolling heuristics for int64
- aco: add Instruction::usesModifiers() and add more checks in the optimizer
- radv: fix radv_nir_get_max_workgroup_size when nir=NULL
- aco: use DPP instead of exec modification when lowering GFX10 shuffles
- aco: fix shuffle with uniform operands
- nir/divergence: improve DA of shuffle
- aco: fix read_invocation with VGPR lane index
- aco: don't propagate vgprs into v_readlane/v_writelane
- aco: combine read_invocation and shuffle implementations
- radv: enable FP16/FP64 denormals earlier and only for LLVM
- aco: don't combine literals into v_cndmask_b32/v_subb/v_addc
- aco: fix 64-bit fsign with 0
- aco: implement VK_KHR_shader_float_controls
- aco: refactor reduction lowering helpers
- aco: implement 64-bit integer reductions

- radv/aco: enable VK_KHR_shader_subgroup_extended_types
- nir: make nir_variable::{num_members,num_state_slots} a uint16_t
- nir: add nir_variable::index and nir_index_vars
- nir/large_constants: use nir_index_vars and nir_variable::index
- docs: update features.txt for RADV
- aco: improve waitcnt insertion around loops
- aco: fix copy+paste error
- aco: fix waitcnts for barriers at block ends
- nir: add nir_num_variable_modes and nir_var_mem_push_const
- radv: set alignment for load_ssbo/store_ssbo in meta shaders
- nir: add a load/store vectorization pass
- nir: add load/store vectorizer tests
- aco: enable load/store vectorizer
- aco: allow constant offsets for global/scratch instructions on GFX10
- aco: set dlc/glc correctly for image loads
- aco: propagate p_wqm on an image_sample's coordinate p_create_vector
- aco: fix i2i64
- aco: fix incorrect cast in parse_wait_instr()
- aco: add v_nop inbetween exec write and VMEM/DS/FLAT
- aco: improve WAR hazard workaround with >64bit stores
- aco: fix GFX10 opcodes for some global/flat atomics
- aco: fix assembly of FLAT/GLOBAL atomics
- aco: fix SADDR with FLAT on GFX10
- aco: don't enable store_global for helper invocations
- aco: improve FLAT/GLOBAL scheduling
- aco: implement global atomics
- ac/llvm: fix pointer type for global atomics
- ac/llvm: improve sync scope for global atomics
- radv: set writes_memory for global memory stores/atomics
- aco: validate the CFG
- aco: handle loop exit and IF merge phis with break/discard
- aco: fix block_kind_discard s_andn2 definition to exec
- nir/lower_io_to_vector: don't create arrays when not needed
- nir/load_store_vectorize: fix combining stores with aliasing loads between
- aco/wave32: fix comparison optimizations
- aco: improve jump threading with wave32

- aco: fix vgpr alloc granule with wave32
- aco: limit register usage for large work groups
- aco: set vm for pos0 exports on GFX10
- aco: fix imageSize()/textureSize() with large buffers on GFX8
- aco: fix uninitialized data in the binary
- aco: handle VOP3 modifiers when combining a constant comparison's NaN test
- aco: handle omod successors with the constant in the first operand
- aco: check usesModifiers() when identifying a neg/abs
- aco: better handle neg/abs of sgprs
- aco: set exec_potentially_empty for demotes
- aco: don't DCE atomics with return values
- aco: disable add combining for ds_swizzle_b32
- aco: check if multiplication/clamp is live when applying output modifier
- nir/divergence: handle load_primitive_id in GS
- nir/lower_gs_intrinsics: add option for per-stream counts
- aco: update IR validator
- aco: apply literals to split mads
- aco: combine two sgprs into a VALU if they're the same
- aco: improve can_use_VOP3()
- aco: rewrite literal combining
- aco: rewrite apply_sgprs()
- aco: add check_vop3_operands()
- aco: be more careful with literals in combine_salu_{n2,lshl_add}
- aco: follow through temporary when merging tests into constant comparisons
- aco: allow applying two sgprs to an instruction
- aco: allow an extra SGPR with multiple uses to be applied to VOP3
- aco: take advantage of GFX10's constant bus limit and VOP3 literals
- aco: improve creation of v_madmk_f32/v_madak_f32
- aco: fix clamp optimization
- aco: improve clamp optimization
- aco: add min(-max(),) and max(-min(),) optimization
- aco: don't move literal to reg when making an instruction VOP3 on GFX10
- aco: allow input modifiers on v_cndmask_b32
- aco: replace extract_vector with copies
- aco: improve readfirstlane after uniform LDS loads
- aco: add integer min/max to can_swap_operands

- nir/sink,nir/move: move/sink load_per_vertex_input
- nir/sink,nir/move: move/sink nir_op_mov
- nir/algebraic: a & ~(a >> 31) -> imax(a, 0)
- aco: fix stack buffer overflow in apply_sgprs()
- aco: fix fall-through test in try_remove_simple_block() with back-edges
- aco: fix operand kill flags when a temporary is used more than once
- aco: fix off-by-one error when initializing sgpr_live_in
- radv: move gs copy shader creation before other variants
- aco: improve support for s_sendmsg
- radv/aco,aco: implement GS on GFX9+
- aco: implement GS on GFX7-8
- radv/aco: allow ACO for GS
- aco: explicitly mark end blocks for exports
- aco: remove needs_instance_id
- aco: implement GS copy shaders
- radv/aco: use ACO for GS copy shaders
- aco: use nir_move_copies
- aco: fix WaR check for >64-bit FLAT/GLOBAL instructions
- aco: fix operand to scc when selecting SGPR ufind_msb/ifind_msb
- aco: always add sgprs to sgpr_ids when choosing literals
- aco: fix literal application with v_cndmask_b32/v_addc_co_u32/etc
- amd/common,radv: move vertex_format_table to ac_shader_util.{h,c}
- aco: rework vertex fetching a bit
- aco: skip unused channels at the start when fetching vertices
- aco: handle unaligned vertex fetch on GFX10
- aco: value-number MUBUF instructions
- aco: use MUBUF in some situations instead of splitting vertex fetches
- aco: fix rebase error from GS copy shader support
- aco: ensure predecessors' p_logical_end is in WQM when a p_phi is in WQM
- aco: run p_wqm instructions in WQM
- nir/algebraic: add patterns for a >> #b << #b
- nir/algebraic: add some half packing optimizations
- aco: fix target calculation when vgpr spilling introduces sgpr spilling
- aco: don't consider loop header blocks branch blocks in add_coupling_code
- aco: don't update demand in add_coupling_code() for loop headers
- aco: only create parallelcopy to restore exec at loop exit if needed

- aco: don't always add logical edges from continue_break blocks to headers
- aco: error when block has no logical preds but VGPRs are live at the start
- aco: set exec_potentially_empty after continues/breaks in nested IFs
- aco: improve assertion at the end of spiller
- aco: fill reg_demand with sensible information in add_coupling_code()
- aco: parallelcopy exec mask before s_wqm
- aco: fix exec mask consistency issues
- aco: fix gfx10_wave64_bpermute

Ricardo Garcia (1):

- anv: Unify GetDeviceQueue and GetDeviceQueue2

Rob Clark (89):

- freedreno/ir3: split pre-coloring to it's own function
- freedreno/ir3: use SSA flag on dest register too
- freedreno/ir3: ir3_print tweaks
- freedreno/ir3/ra: move regs_count==0 check
- freedreno/ir3/ra: remove ir print after livein/out
- freedreno/ir3: remove obsolete comment
- freedreno/a3xx: fix SP_FS_MRT_REG.HALF_PRECISION
- freedreno/a4xx: fix SP_FS_MRT_REG.HALF_PRECISION
- freedreno/ir3: sync disasm changes from envytools
- freedreno/ir3: also track # of nops for shader-db
- freedreno: fix eglDupNativeFenceFD error
- freedreno/ir3: fix valgrind complaint with STLW
- freedreno/ir3: remove half-precision output
- freedreno/ir3: rename fanin/fanout to collect/split
- freedreno/ir3: remove impossible condition
- freedreno/ir3: add input/output iterators
- freedreno/ir3: show input/output wrmask's in disasm
- freedreno/ir3: helper to print ir if debug enabled
- freedreno/ir3: remove first-vertex sysval
- freedreno/ir3: simplify creating sysval inputs
- freedreno/ir3: re-work shader inputs/outputs
- freedreno/ir3: only tex instructions have wrmask
- freedreno/ir3: fix gpu hang with pre-fs-tex-fetch
- freedreno/ir3: legalize cleanups
- freedreno/ir3: remove unused parameter

- freedreno/perfcntrs: small cleanup
- freedreno/perfcntrs: remove gallium dependencies
- freedreno/perfcntrs: move to shared location
- freedreno/perfcntrs: add accessor to get per-gen tables
- freedreno/perfctrs/a2xx: move CP to be first group
- freedreno/perfcntrs/a6xx: remove RBBM counters
- freedreno/perfcntrs: add fdperf
- freedreno/perfctrs/fdperf: periodically restore counters
- gitlab-ci: update deqp build so we can generate xml
- gitlab-ci/deqp: preserve full list of unexpected results
- gitlab-ci/deqp: preserve caselists for blocks with fails
- gitlab-ci/deqp: detect and report flakes
- gitlab-ci: bump arm test container
- gitlab-ci/deqp: generate xml results for fails/flakes
- gitlab-ci/deqp: generate junit results
- gitlab-ci/freedreno/a6xx: remove most of the flakes
- freedreno: use rsc->slice accessor everywhere
- freedreno: switch to layout helper
- gitlab-ci: disable junit results for deqp
- freedreno/ir3: remove store_output lowered to store_shared_ir3
- freedreno/ir3: fix neverball assert in case of unused VS inputs
- nir/lower_clip: Fix incorrect driver loc for clipdist outputs
- freedreno/fdperf: use drmOpen()
- freedreno/a6xx: disable LRZ when blending
- freedreno/a5xx+a6xx: split LRZ layout to per-gen
- freedreno/a6xx: fix LRZ layout
- freedreno/a6xx: fix LRZ logic
- freedreno/a6xx: enable LRZ by default
- spirv: add OpLifetime*
- freedreno/ir3: add last-baryf shaderdb stat
- freedreno/ir3: add scheduler traces
- freedreno/ir3: add iterator macros
- freedreno/a6xx: fix OUT_REG() vs growable cmdstream
- nir+vtn: vec8+vec16 support
- freedreno/ir3: fix flat shading again
- nir: assert that nir_lower_tex runs after lowering derefs

- mesa/st: lower samplers before nir_lower_tex
- freedreno/ir3: rename instructions
- gitlab-ci: fix missing caselist.css/xsl
- freedreno/a6xx: limit scratch/debug markers to debug builds
- freedreno/a6xx: cleanup rasterizer state
- freedreno/a6xx: separate rast stateobj for prim restart
- freedreno/a6xx: drop a few more per-draw registers
- freedreno/a6xx: move dynamic program state to streaming stateobj
- freedreno/a6xx: add PROG_FB_RAST stateobj
- freedreno/drm: fix invalid-cmdstream-size with older kernels
- freedreno: use PIPE_CAP_RGB_OVERRIDE_DST_ALPHA_BLEND
- mesa/st: random whitespace cleanup
- freedreno/a6xx: remove special handling based on MRT format
- freedreno/a6xx: convert blend state to stateobj
- freedreno: extract vsc pipe bo from GMEM state
- freedreno: consolidate GMEM state
- freedreno: constify fd_tile
- freedreno: constify fd_vsc_pipe
- freedreno/a6xx: constify gmem state
- freedreno/a5xx: constify gmem state
- freedreno/a4xx: constify gmem state
- freedreno/a3xx: constify gmem state
- freedreno/a2xx: constify gmem state
- freedreno: get GMEM state from batch
- freedreno: add gmem state cache
- freedreno: add gmem_lock
- freedreno: remove flush-queue
- freedreno: allow ctx->batch to be NULL

Robert Foss (5):

- nir: Build nir_lower_point_size.c in libmesa_nir
- android: Add panfrost support to build scripts
- android: Fix u_format_table.c being generated twice
- panfrost: Prefix schedule_program to prevent collision
- android: Fix whitespace issue

Rohan Garg (1):

- gitlab-ci: Use lavacli from packages

Roland Scheidegger (3):

- gallium/scons: fix graw_gdi build
- util/atomic: Fix p_atomic_add for unlocked and msvc paths
- winsys/svga: use new ioctl for logging

Roman Stratiienko (2):

- Android: Fix build issue without LLVM
- panfrost: Fix Android build

Ross Zwisler (1):

- intel: limit shader geometry on BDW GT1

Sagar Ghuge (1):

- intel/compiler: Clear accumulator register before EOT

Samuel Iglesias Gonsálvez (1):

- main: fix coverity error in _mesa_program_resource_find_name()

Samuel Pitoiset (202):

- radv: declare NGG scratch for VS or TES and only on GFX10
- radv: fix compute pipeline keys when optimizations are disabled
- docs: document all RADV environment variables
- radv: add a note about perftest/debug options
- radv: fix 32-bit compiler warnings
- nir: fix packing of nir_variable
- radv/gfx10: enable wave32 for compute based on shader's wavesize
- radv: hardcode the number of waves for the GFX6 LS-HS bug
- radv: determine shaders wavesize at pipeline level
- radv: rely on shader's wavesize when computing NGG info
- radv: implement VK_EXT_subgroup_size_control
- radv/gfx10: fix primitive indices orientation for NGG GS
- ac: handle pointer types to LDS in ac_get_elem_bits()
- gitlab-ci: build a specific libdrm version for ARM64
- gitlab-ci: build RADV on ARM64
- ac: fix build with recent LLVM
- radv: remove useless RADV_DEBUG=unsafemath debug option
- radv: make sure to not clear the ds attachment after resolves
- ac: add radeon_info::has_l2_uncached
- radv: implement VK_AMD_device_coherent_memory
- spirv: fix lowering of OpGroupNonUniformAllEqual
- ac: remove useless cast in ac_build_set_inactive()

- ac: add 8-bit and 16-bit supports to ac_build_shuffle()
- ac: add 8-bit and 16-bit supports to ac_build_readlane()
- ac: add 8-bit and 16-bit supports to ac_build_set_inactive()
- ac: add 8-bit and 16-bit supports to ac_build_dpp()
- ac: add 8-bit and 16-bit supports to ac_build_swizzle()
- ac: add 8-bit and 16-bit supports to get_reduction_identity()
- ac: add 8-bit and 16-bit supports to ac_build_wwm()
- ac: add 8-bit and 16-bit supports to ac_build_optimization_barrier()
- ac: add 16-bit float support to ac_build_alu_op()
- radv: advertise VK_KHR_shader_subgroup_extended_types on GFX8-GFX9
- radv: enable VK_KHR_shader_subgroup_extended_types on GFX6-GFX7
- docs: add missing new features for RADV
- pipe-loader: check that the pointer to driconf_xml isn't NULL
- gitlab-ci: move building piglit into a separate script
- gitlab-ci: fix ldd check for Vulkan drivers
- gitlab-ci: add a job that only build things needed for testing
- gitlab-ci: do not build with debugoptimized for meson-main
- gitlab-ci: build swr in meson-main
- gitlab-ci: build GLVND in meson-clang
- gitlab-ci: remove now useless meson-swr-glvnd build job
- gitlab-ci: reduce the number of scons build
- radv: disable subgroup shuffle operations on GFX10
- ac/llvm: fix the local invocation index for wave32
- meson: only build imgui when needed
- radv: set the image view aspect mask during subpass transitions
- radv: set the image view aspect mask before resolves
- radv: rework creation of decompress/resummarize meta pipelines
- radv: create decompress pipelines for separate depth/stencil layouts
- radv: select the depth decompress path based on the aspect mask
- ac/llvm: fix warning in ac_build_canonicalize()
- radv: fix reporting subgroup size with VK_KHR_pipeline_executable_properties
- radv: fix enabling sample shading with SampleID/SamplePosition
- radv/gfx10: fix implementation of exclusive scans
- ac: add 8-bit and 16-bit supports to ac_build_permlane16()
- radv: enable VK_KHR_shader_subgroup_extended_types on GFX10
- ac/llvm: convert src operands to pointers if necessary

- radv: add more constants to avoid using magic numbers
- radv,ac/nir: lower deref operations for shared memory
- aco: drop useless lowering of deref operations for shared memory
- ac/llvm: fix atomic var operations if source isn't a deref
- radv: remove dead shader input/output variables
- radv: simplify a check in radv_fixup_vertex_input_fetches()
- radv/gfx10: fix the vertex order for triangle strips emitted by a GS
- gitlab-ci: rename build-deqp.sh to build-deqp-gl.sh
- gitlab-ci: add a gl suffix to the x86 test image and all test jobs
- gitlab-ci: add a new job that builds a base test image for VK
- gitlab-ci: build cts_runner in the x86 test image for VK
- gitlab-ci: build dEQP VK 1.1.6 in the x86 test image for VK
- gitlab-ci: add a new base test job for VK
- gitlab-ci: allow to run dEQP Vulkan with DEQP_VER
- gitlab-ci: configure the Vulkan ICD export with VK_DRIVER
- gitlab-ci: build RADV in meson-testing
- gitlab-ci: add a job that runs Vulkan CTS with RADV conditionally
- radv: do not use VK_TRUE/VK_FALSE
- radv: move emission of two PA_SC_* registers to the pipeline CS
- radv: fix possibly wrong PA_SC_AA_CONFIG value for conservative rast
- radv: synchronize after performing a separate depth/stencil fast clears
- radv: do not init HTILE as compressed state when dst layout allows it
- radv: initialize HTILE for separate depth/stencil aspects
- radv: implement VK_KHR_separate_depth_stencil_layouts
- gitlab-ci: set RADV_DEBUG=checkir for RADV test jobs
- ac/nir: fix out-of-bound access when loading constants from global
- radv: enable SpvCapabilityImageMSArray
- radv: handle unaligned vertex fetches on GFX6/GFX10
- radv/gfx10: fix ngg_get_ordered_id
- radv/gfx10: fix the out-of-bounds check for vertex descriptors
- ac: declare an enum for the OOB select field on GFX10
- radv: init a default multisample state for the resolve FS path
- radv: ignore pMultisampleState if rasterization is disabled
- radv: ignore pTessellationState if the pipeline doesn't use tess
- radv: ignore pDepthStencilState if rasterization is disabled
- radv: tidy up radv_pipeline_init_blend_state()

- radv: ignore pColorBlendState if rasterization is disabled
- radv: rely on pipeline layout when creating push descriptors with template
- radv: return the correct pitch for linear mipmaps on GFX10
- radv: record number of color/depth samples for each subpass
- radv: implement VK_AMD_mixed_attachment_samples
- ac/surface: use uint16_t for mipmap level pitches
- radv: do not fill keys from fragment shader twice
- spirv: add SpvCapabilityImageReadWriteLodAMD
- spirv,nir: add new lod parameter to image_{load,store} intrinsics
- amd/llvm: handle nir_intrinsic_image_deref_{load,store} with lod
- aco: handle nir_intrinsic_image_deref_{load,store} with lod
- radv: advertise VK_AMD_shader_image_load_store_lod
- radv/gfx10: disable vertex grouping
- radv/gfx10: determine if a pipeline is eligible for NGG passthrough
- radv/gfx10: do not declare LDS for NGG if useless
- radv/gfx10: add support for NGG passthrough mode
- radv/gfx10: improve performance for TES using PrimID but not exporting it
- radv: only use VkSamplerCreateInfo::compareOp if enabled
- radv/gfx10: enable all CUs if NGG is never used
- radv/gfx10: simplify some duplicated NGG GS code
- vulkan/overlay: Fix for Vulkan 1.2
- radv: update VK_EXT_descriptor_indexing for Vulkan 1.2
- radv: update VK_EXT_host_query_reset for Vulkan 1.2
- radv: update VK_EXT_sampler_filter_minmax for Vulkan 1.2
- radv: update VK_EXT_scalar_block_layout for Vulkan 1.2
- radv: update VK_KHR_8bit_storage for Vulkan 1.2
- radv: update VK_KHR_buffer_device_address for Vulkan 1.2
- radv: update VK_KHR_create_renderpass2 for Vulkan 1.2
- radv: update VK_KHR_depth_stencil_resolve for Vulkan 1.2
- radv: update VK_KHR_draw_indirect_count for Vulkan 1.2
- radv: update VK_KHR_driver_properties for Vulkan 1.2
- radv: update VK_KHR_image_format_list for Vulkan 1.2
- radv: update VK_KHR_imageless_framebuffer for Vulkan 1.2
- radv: update VK_KHR_shader_atomic_int64 for Vulkan 1.2
- radv: update VK_KHR_shader_float16_int8 for Vulkan 1.2
- radv: update VK_KHR_shader_float_controls for Vulkan 1.2

- radv: update VK_KHR_shader_subgroup_extended_types for Vulkan 1.2
- radv: update VK_KHR_uniform_buffer_standard_layout for Vulkan 1.2
- radv: update VK_KHR_timeline_semaphore for Vulkan 1.2
- radv: implement Vulkan 1.1 features and properties
- radv: implement Vulkan 1.2 features and properties
- radv: enable Vulkan 1.2
- aco: fix emitting SMEM instructions with no operands on GFX6-GFX7
- aco: do not select 96-bit/128-bit variants for ds_read/ds_write on GFX6
- aco: do not combine additions of DS instructions on GFX6
- aco: implement stream output with vec3 on GFX6
- aco: fix emitting slc for MUBUF instructions on GFX6-GFX7
- aco: print assembly with CLRXdisasm for GFX6-GFX7 if found on the system
- aco: fix constant folding of SMRD instructions on GFX6
- aco: do not use the vec3 variant for stores on GFX6
- aco: do not use the vec3 variant for loads on GFX6
- aco: add new addr64 bit to MUBUF instructions on GFX6-GFX7
- aco: implement nir_intrinsic_load_barycentric_at_sample on GFX6
- radv: fix double free corruption in radv_alloc_memory()
- radv: add explicit external subpass dependencies to meta operations
- radv: handle missing implicit subpass dependencies
- spirv: add SpvCapabilityFragmentMaskAMD
- nir: add two new texture ops for multisample fragment color/mask fetches
- spirv: add support for SpvOpFragment{Mask}FetchAMD operations
- nir/lower_input_attachments: lower nir_texop_fragment_{mask}_fetch
- ac/nir: add support for nir_texop_fragment_{mask}_fetch
- aco: add support for nir_texop_fragment_{mask}_fetch
- radv: advertise VK_AMD_shader_fragment_mask
- aco: fix printing assembly with CLRXdisasm on GFX6
- aco: fix wrong IR in nir_intrinsic_load_barycentric_at_sample
- aco: implement nir_intrinsic_store_global on GFX6
- aco: implement nir_intrinsic_load_global on GFX6
- aco: implement nir_intrinsic_global_atomic_* on GFX6
- aco: implement 64-bit nir_op_ftrunc on GFX6
- aco: implement 64-bit nir_op_fceil on GFX6
- aco: implement 64-bit nir_op_fround_even on GFX6
- aco: implement 64-bit nir_op_ffloor on GFX6

- aco: implement nir_op_f2i64/nir_op_f2u64 on GFX6
- ac/llvm: fix missing casts in ac_build_readlane()
- aco: combine MRTZ (depth, stencil, sample mask) exports
- aco: fix a hardware bug for MRTZ exports on GFX6
- aco: fix a hazard with v_interp_* and v_{read,readfirst}lane_* on GFX6
- aco: copy the literal offset of SMEM instructions to a temporary
- radv: enable ACO support for GFX6
- radv: print NIR shaders after lowering FS inputs/outputs
- radv: do not allow sparse resources with multi-planar formats
- radv: enable VK_AMD_shader_fragment_mask on GFX6-GFX7
- compiler: add a new explicit interpolation mode
- spirv: add support for SpvDecorationExplicitInterpAMD
- compiler: add PERSP to the existing barycentric system values
- compiler: add new SYSTEM_VALUE_BARYCENTRIC_*
- spirv: add support for SpvBuiltInBaryCoord*
- nir: add nir_intrinsic_load_barycentric_model
- nir: lower SYSTEM_VALUE_BARYCENTRIC_* to nir_load_barycentric()
- nir: add nir_intrinsic_interp_deref_at_vertex
- nir: lower interp_deref_at_vertex to load_input_vertex
- spirv: implement SPV_AMD_shader_explicit_vertex_parameter
- ac/llvm: implement VK_AMD_shader_explicit_vertex_parameter
- aco: implement VK_AMD_shader_explicit_vertex_parameter
- radv: gather which input PS variables use an explicit interpolation mode
- radv: implement VK_AMD_shader_explicit_vertex_parameter
- radv: bump conformance version to 1.2.0.0
- radv: remove the non conformant VK implementation warning on GFX10
- aco: fix VS input loads with MUBUF on GFX6
- radv/gfx10: add a separate flag for creating a GDS OA buffer
- radv/gfx10: implement NGG GS queries
- radv/gfx10: re-enable NGG GS
- radv: refactor physical device properties
- aco: fix MUBUF VS input loads when expanding vec3 to vec4 on GFX6
- aco: do not use ds_{read,write}2 on GFX6
- aco: fix waiting for scalar stores before “writing back” data on GFX8-GFX9
- aco: fix creating v_madak if v_mad_f32 has two sgpr literals
- nir: do not use De Morgan’s Law rules for flt and fge

Samuel Thibault (3):

- loader: #define PATH_MAX when undefined (eg. Hurd)
- util: Do not fail to build on unknown pthread_setname_np
- meson: Do not require libdrm for DRI2 on hurd

Satyajit Sahu (1):

- radeon/vcn: Handle crop parameters for encoder

Sonny Jiang (1):

- radeonsi: use compute shader for clear 12-byte buffer

Stephan Gerhold (1):

- kmsro: Add "mcde" entry point

Tapani Pälli (33):

- nir: fix couple of compile warnings
- util/android: fix android build errors
- Revert "egl: implement new functions from EGL_EXT_image_flush_external"
- Revert "egl: handle EGL_IMAGE_EXTERNAL_FLUSH_EXT"
- Revert "st/dri: add support for EGL_EXT_image_flush_external"
- Revert "st/dri: assume external consumers of back buffers can write to the buffers"
- Revert "dri_interface: add interface for EGL_EXT_image_flush_external"
- mesa: allow bit queries for EXT_disjoint_timer_query
- Revert "mesa: allow bit queries for EXT_disjoint_timer_query"
- mesa: allow bit queries for EXT_disjoint_timer_query
- gitlab-ci: update Piglit commit, update skips
- mapi: add GetInteger64vEXT with EXT_disjoint_timer_query
- glsl: handle max uniform limits with lower_const_arrays_to_uniforms
- gitlab-ci: bump piglit checkout commit
- glsl: additional interface redeclaration check for SSO programs
- intel/compiler: add newline to limit_dispatch_width message
- intel/compiler: force simd8 when dual src blending on gen8
- dri: add __DRI_IMAGE_FORMAT_SXRGB8
- i965: expose MESA_FORMAT_B8G8R8X8_SRGB visual
- mesa/st/i965: add a ProgramResourceHash for quicker resource lookup
- mesa: create program resource hash in a single place
- iris: set depth stall enabled when depth flush enabled on gen12
- anv: set depth stall enabled when depth flush enabled on gen12
- isl/gen12: add reminder comment about missing WA with 3D surfaces
- anv: fix assert in GetImageDrmFormatModifierPropertiesEXT

- anv: add assert for isl_mod_info in choose_isl_tiling_flags
- anv: initialize clear_color_is_zero_one
- egl/android: fix buffer_count for applications setting max count
- anv/android: setup gralloc1 usage from gralloc0 usage manually
- anv/android: make format_supported_with_usage static
- intel/vec4: fix valgrind errors with vf_values array
- glsl: fix a memory leak with resource_set
- iris: fix aux buf map failure in 32bits app on Android

Thomas Hellstrom (4):

- winsys/svga: Enable transhuge pages for buffer objects
- svga: Avoid discard DMA uploads
- gallium/util: Increase the debug_flush map depth
- svga: Fix banded DMA upload

Thong Thai (8):

- st/va: Convert interlaced NV12 to progressive
- util/format: Add the P010 format used for 10-bit videos
- gallium: Add PIPE_FORMAT_P010 support
- st/va: Add support for P010, used for 10-bit videos
- radeon: Use P010 for decoding of 10-bit videos
- r600: Remove HEVC related code since HEVC is not supported
- mesa: Prevent _MaxLevel from being less than zero
- Revert "st/va: Convert interlaced NV12 to progressive"

Timothy Arceri (66):

- glsl: just use NIR to lower outputs when driver can't read outputs
- glsl: disable lower_fragdata_array() for NIR drivers
- mesa: add ARB_shading_language_include stubs
- glsl: add infrastructure for ARB_shading_language_include
- mesa: add ARB_shading_language_include infrastructure to gl_shared_state
- mesa: add helper to validate tokenise shader include path
- mesa: add _mesa_lookup_shader_include() helper
- mesa: add copy_string() helper
- mesa: add glNamedStringARB() support
- mesa: implement glGetNamedStringARB()
- mesa: make error checking optional in _mesa_lookup_shader_include()
- mesa: implement glIsNamedStringARB()
- mesa: implement glGetNamedStringivARB()

- mesa: split _mesa_lookup_shader_include() in two
- mesa: implement glDeleteNamedStringARB()
- glsl: add ARB_shading_language_include support to #line
- glsl: pass gl_context to glcpp_parser_create()
- glsl: add preprocessor #include support
- glsl: error if #include used while extension is disabled
- glsl: add can_skip_compile() helper
- glsl: delay compilation skip if shader contains an include
- mesa: add support cursor support for relative path shader includes
- mesa: add shader include lookup support for relative paths
- mesa: implement glCompileShaderIncludeARB()
- mesa: enable ARB_shading_language_include
- gitlab-ci: bump piglit checkout commit
- gitlab-ci: update for arb_shading_language_include
- compiler: move build definition of pp_standalone_scaffolding.c
- radv: add some infrastructure for fresh forks for each secure compile
- radv: add a secure_compile_open_fifo_fds() helper
- radv: create a fresh fork for each pipeline compile
- docs: update source code repository documentation
- glsl: move calculate_array_size_and_stride() to link_uniforms.cpp
- glsl: don't set uniform block as used when its not
- glsl: make use of active_shader_mask when building resource list
- glsl/nir: iterate the system values list when adding varyings
- docs: remove mailing list as way of submitting patches
- glsl: move nir_remap_dual_slot_attributes() call out of glsl_to_nir()
- glsl: copy the how_declared field when converting to nir
- nir: add some fields to nir_variable_data
- glsl: copy the new data fields when converting to nir
- glsl: add support for named varyings in nir_build_program_resource_list()
- glsl: add subroutine support to nir_build_program_resource_list()
- st/glsl_to_nir: call gl_nir_lower_buffers() a little later
- st/glsl_to_nir: use nir based program resource list builder
- st/glsl_to_nir: fix SSO validation regression
- glsl: rename gl_nir_link() to gl_nir_link_spirv()
- glsl: add gl_nir_link_check_atomic_counter_resources()
- glsl: add new gl_nir_link_glsl() helper

- glsl: reorder link_and_validate_uniforms() calls
- mesa: add new UseNIRGLSLLinker constant
- glsl: use nir linker to link atomics
- glsl: add check_image_resources() for the nir linker
- glsl: use nir version of check_image_resources() for nir linker
- glsl: move check_subroutine_resources() into the shared util code
- glsl: call check_subroutine_resources() from the nir linker
- glsl: move uniform resource checks into the common linker code
- glsl: call uniform resource checks from the nir linker
- glsl: move calculate_subroutine_compat() to shared linker code
- glsl: call calculate_subroutine_compat() from the nir linker
- glsl: fix potential bug in nir uniform linker
- glsl: remove bogus assert in nir uniform linking
- glsl: fix check for matrices in blocks when using nir uniform linker
- glsl: count uniform components and storage better in nir linking
- glsl_to_nir: update interface type properly
- glsl: fix gl_nir_set_uniform_initializers() for image arrays

Timur Kristóf (39):

- ac: Handle invalid GFX10 format correctly in ac_get_tbuffer_format.
- aco: Make sure not to mistakenly propagate 64-bit constants.
- aco: Treat all booleans as per-lane.
- aco: Optimize out trivial code from uniform bools.
- aco: Fix operand of s_bcnt1_i32_b64 in emit_boolean_reduce.
- aco: Remove superfluous argument from emit_boolean_logic.
- aco: Remove lower_linear_bool_phi, it is not needed anymore.
- aco: Optimize load_subgroup_id to one bit field extract instruction.
- aco/wave32: Change uniform bool optimization to work with wave32.
- aco/wave32: Replace hardcoded numbers in spiller with wave size.
- aco/wave32: Introduce emit_mbcnt which takes wave size into account.
- aco/wave32: Add wave size specific opcodes to aco_builder.
- aco/wave32: Use lane mask regclass for exec/vcc.
- aco/wave32: Fix load_local_invocation_index to support wave32.
- aco/wave32: Use wave_size for barrier intrinsic.
- aco/wave32: Allow setting the subgroup ballot size to 64-bit.
- aco/wave32: Fix reductions.
- aco: Fix uniform i2i64.

- ac/llvm: Fix ac_build_reduce in wave32 mode.
- aco/wave32: Set the definitions of v_cmp instructions to the lane mask.
- aco: Implement 64-bit constant propagation.
- aco: Allow optimizing vote_all and nir_op_iand.
- aco: Don't skip combine_instruction when definitions[1] is used.
- aco: Optimize out s_and with exec, when used on uniform bitwise values.
- aco: Flip s_cbranch / s_cselect to optimize out an s_not if possible.
- nouveau/nvc0: add extern keyword to nvc0_miptree_vtbl.
- intel/compiler: Fix array bounds warning on GCC 10.
- radeon: Move si_get_pic_param to radeon_vce.c
- r600: Move get_pic_param to radeon_vce.c
- gallium: Fix a couple of multiple definition warnings.
- radeon: Fix multiple definition error with radeon_debug
- aco: Fix -Wstringop-overflow warnings in aco_span.
- aco: Fix maybe-uninitialized warnings.
- aco: Fix signedness compare warning.
- aco: Make a better guess at which instructions need the VCC hint.
- aco: Transform uniform bitwise instructions to 32-bit if possible.
- aco/gfx10: Fix VcmpxExecWARHazard mitigation.
- aco: Fix the meaning of is_atomic.
- aco/optimizer: Don't combine uniform bool s_and to s_andn2.

Tomasz Pyra (1):

- gallium/swr: Fix arb_transform_feedback2

Tomeu Vizoso (38):

- gitlab-ci: Disable lima jobs
- gitlab-ci: Run only LAVA jobs in special-named branches
- panfrost: Add checksum fields to SFBD descriptor
- panfrost: Set 0x10 bit on mali_shader_meta.unknown2_4 on T720
- panfrost: Rework format encoding on SFBD
- panfrost: Take into account texture layers in SFBD
- panfrost: Decode blend shaders for SFBD
- panfrost: Generate polygon list manually for SFBD
- panfrost: Print the right zero field
- panfrost: Pipe the GPU ID into compiler and disassembler
- panfrost: Set depth and stencil for SFBD based on the format
- panfrost: Multiply offset_units by 2

- panfrost: Make sure the shader descriptor is in sync with the GL state
- gitlab-ci: Remove limit on kernel logging
- panfrost: Just print tiler fields as-is for Tx20
- panfrost: Rework buffers in SFBD
- gitlab-ci: Fix dir name for VK-GL-CTS sources
- panfrost: Don't print the midgard_blend_rt structs on SFBD
- panfrost: Add quirks system to cmdstream
- panfrost: Simplify shader patching
- panfrost: White list the Mali T720
- gitlab-ci: Test Panfrost on T720 GPUs
- panfrost: Add PAN_MESA_DEBUG=sync
- panfrost: Hold a reference to sampler views
- pan/midgard: Remove undefined behavior
- nir: Don't copy empty array
- util: Don't access members of NULL pointers
- panfrost: Don't lose bits!
- st/mesa: Don't access members of NULL pointers
- panfrost: Handle Z24_UNORM_S8_UINT as MALI_Z32_UNORM
- panfrost: Increase PIPE_SHADER_CAP_MAX_OUTPUTS to 16
- panfrost: Dynamically allocate array of texture pointers
- panfrost: Map with size of first layer for 3D textures
- panfrost: Store internal format
- gitlab-ci: Update kernel for LAVA to 5.5-rc1 plus fixes
- gitlab-ci: Switch LAVA jobs to use shared dEQP runner
- gitlab-ci: Upgrade kernel for LAVA jobs to v5.5-rc5
- gitlab-ci: Consolidate container and build stages for LAVA

Urja Rannikko (4):

- panfrost: free last_read/write tables in mir_create_dependency_graph
- panfrost: free allocations in schedule_block
- panfrost: add lcra_free() to free lcra state
- panfrost: free spill cost table in mir_spill_register

Vasily Khoruzhick (31):

- lima: add debug prints for BO cache
- lima: align size before trying to fetch BO from cache
- lima: ignore flags while looking for BO in cache
- lima: set dithering flag when necessary

- lima: add support for gl_PointSize
- lima: enable tiling
- lima: handle DRM_FORMAT_MOD_INVALID in resource_from_handle()
- lima: expose tiled format modifier in query_dmabuf_modifiers()
- lima: use single BO for GP outputs
- lima: drop suballocator
- lima: fix allocation of GP outputs storage for indexed draw
- lima: postpone PP stream generation
- lima: don't reload and redraw tiles that were not updated
- lima: fix PP stream terminator size
- lima: use linear layout for shared buffers if modifier is not specified
- lima: add debug flag to disable tiling
- lima: drop support for R8G8B8 format
- lima: fix PLBU_CMD_PRIMITIVE_SETUP command
- lima: fix viewport clipping
- lima: implement polygon offset
- lima: fix PIPE_CAP_* to mark features that aren't supported yet
- lima: add new findings to texture descriptor
- lima: fix handling of reverse depth range
- ci: lava: pass CI_NODE_INDEX and CI_NODE_TOTAL to lava jobs
- ci: Re-enable CI for lima on mali450
- lima: implement invalidate_resource()
- nir: don't emit ishl in _nir_mul_imm() if backend doesn't support bitops
- lima: use imul for calculations with intrinsic src
- lima: ppir: don't delete root ld_tex nodes without successors in current block
- lima: ppir: always create move and update ld_tex successors for all blocks
- lima: disable early-z if fragment shader uses discard

Vinson Lee (9):

- swr: Fix build with llvm-10.0.
- panfrost: Fix gnu-empty-initializer build errors.
- scons: Bump C standard to gnu11 on macOS 10.15.
- util/u_thread: Restrict u_thread_get_time_nano on macOS.
- swr: Fix build with llvm-10.0.
- swr: Fix build with llvm-10.0.
- lima: Fix build with GCC 10.
- swr: Fix GCC 4.9 checks.

- panfrost: Remove unused anonymous enum variables.

Wladimir J. van der Laan (2):

- u_vbuf: add logic to use a limited number of vbufs
- u_vbuf: use single vertex buffer if it's not possible to have multiple

X512 (1):

- util/u_thread: Fix build under Haiku

Yevhenii Kolesnikov (5):

- glsl: Enable textureSize for samplerExternalOES
- meson: Fix linkage of libgallium_nine with libgalliumvl
- meta: Cleanup function for DrawTex
- main: allow external textures for BindImageTexture
- meta: Add cleanup function for Bitmap

Zebediah Figura (1):

- Revert "draw: revert using correct order for prim decomposition."

luc (1):

- zink: confused compilation macro usage for zink in target helpers.

4.24 Mesa 19.3.4 Release Notes / 2020-02-13

Mesa 19.3.4 is a bug fix release which fixes bugs found since the 19.3.3 release.

Mesa 19.3.4 implements the OpenGL 4.6 API, but the version reported by glGetString(GL_VERSION) or glGetIntegerv(GL_MAJOR_VERSION) / glGetIntegerv(GL_MINOR_VERSION) depends on the particular driver being used. Some drivers don't support all the features required in OpenGL 4.6. OpenGL 4.6 is **only** available if requested at context creation. Compatibility contexts may report a lower version depending on each driver.

Mesa 19.3.4 implements the Vulkan 1.1 API, but the version reported by the apiVersion property of the VkPhysicalDeviceProperties struct depends on the particular driver being used.

4.24.1 SHA256 checksum

```
1da467e6ae2799a517e242462331eafd29ae77d9872f3a845df81f7c308e8fe4  mesa-19.3.4.tar.xz
```

4.24.2 New features

- None

4.24.3 Bug fixes

- [RADV] GPU hangs while the cutscene plays in the game Assassin's Creed Origins
- Broken rendering of glxgears on S/390 architecture (64bit, BigEndian)

- GL_EXT_disjoint_timer_query failing with GL_INVALID_ENUM
- GL_EXT_disjoint_timer_query failing with GL_INVALID_ENUM
- OSMesa osmesa_choose_format returns a format not supported by st_new_renderbuffer_fb
- Using EGL_KHR_surfaceless_context causes spurious "libEGL warning: FIXME: egl/x11 doesn't support front buffer rendering."
- [Regression] JavaFX unbounded VRAM+RAM usage
- !3460 broke texsubimage test with piglit on zink+anv

4.24.4 Changes

Bas Nieuwenhuizen (1):

- radv: Do not set SX DISABLE bits for RB+ with unused surfaces.

Boris Brezillon (1):

- panfrost: Fix the damage box clamping logic

Brian Ho (2):

- anv: Properly fetch partial results in vkGetQueryPoolResults
- anv: Handle unavailable queries in vkCmdCopyQueryPoolResults

Danylo Piliaiev (2):

- i965: Do not set front_buffer_dirty if there is no front buffer
- st/mesa: Handle the rest renderbuffer formats from OSMesa

Drew Davenport (1):

- radeonsi: Clear uninitialized variable

Dylan Baker (15):

- docs: Add SHA 256 sums for 19.3.3
- .pick_status.json: Mark 58c929be0ddbbd9291d0dadbf11538170178e791 as backported
- .pick_status.json: Mark df34fa14bb872447fed9076e06ffc504d85e2d1c as backported
- .pick_status.json: Update to 997040e4b8353fe9b71a5e9fde2f933eae09c7a3
- .pick_status.json: Update to ca6a22305b275b49fbc88b8f4cba2fefb24c2a5d
- .pick_status.json: Mark 552028c013cc1d49a2b61ebe0fc3a3781a9ba826 as denominated
- .pick_status.json: Update to f09c466732e4a5b648d7503787777c926dd93c29
- bin/pick-ui: Add a new maintainer script for picking patches
- .pick_status.json: Update to b550b7ef3b8d12f533b67b1a03159a127a3ff34a
- .pick_status.json: Update to 9afdcd64f2c96f3fcc1a28912987f2e8066aa995
- .pick_status.json: Update to 7eaf21cb6f67adbe0e79b80b4feb8c816a98a720
- .pick_status.json: Mark ca6a22305b275b49fbc88b8f4cba2fefb24c2a5d as backported
- .pick_status.json: Update to d8bae10bfe0f487dcaec721743cd51441bcc12f5
- .pick_status.json: Update to 689817c9dfde9a0852f2b2489cb0fa93ffbcb215

- .pick_status.json: Update to 23037627359e739c42b194dec54875aefbb9d00b

Eric Anholt (1):

- Revert "gallium: Fix big-endian addressing of non-bitmask array formats."

Florian Will (1):

- radv/winsys: set IB flags prior to submit in the sysmem path

Georg Lehmann (3):

- Correctly wait in the fragment stage until all semaphores are signaled
- Vulkan Overlay: Don't try to change the image layout to present twice
- Vulkan overlay: use the corresponding image index for each swapchain

Hyunjun Ko (1):

- freedreno/ir3: put the conversion back for half const to the right place.

Ian Romanick (1):

- intel/fs: Don't count integer instructions as being possibly coissue

Jan Vesely (1):

- clover: Use explicit conversion from llvm::StringRef to std::string

Jason Ekstrand (6):

- anv: Insert holes for non-existant XFB varyings
- anv: Improve BTI change cache flushing
- anv,iris: Set 3DSTATE_SF::DerefBlockSize to per-poly on Gen12+
- genxml: Add a new 3DSTATE_SF field on gen12
- intel/fs: Write the address register with NoMask for MOV_INDIRECT
- anv/blorp: Use the correct size for vkCmdCopyBufferToImage

Kenneth Graunke (1):

- i965: Use brw_batch_references in tex_busy check

Lionel Landwerlin (1):

- isl: drop CCS row pitch requirement for linear surfaces

Marek Olšák (1):

- radeonsi: fix the DCC MSAA bug workaround

Marek Vasut (1):

- etnaviv: Destroy rsc->pending_ctx set in etna_resource_destroy()

Michel Dänzer (6):

- winsys/amdgpu: Keep a list of amdgpu_screen_winsyses in amdgpu_winsys
- winsys/amdgpu: Keep track of retrieved KMS handles using hash tables
- winsys/amdgpu: Only re-export KMS handles for different DRM FDs
- util: Add os_same_file_description helper
- winsys/amdgpu: Re-use amdgpu_screen_winsys when possible

- winsys/amdgpu: Close KMS handles for other DRM file descriptions

Neha Bhende (1):

- svga: fix size of format_conversion_table[]

Pierre-Eric Pelloux-Prayer (2):

- radeonsi: disable display DCC
- radeonsi: stop using the VM_ALWAYS_VALID flag

Rafael Antognolli (1):

- intel: Load the driver even if I915_PARAM_REVISION is not found.

Rhys Perry (6):

- aco: fix operand to scc when selecting SGPR ufind_msb/ifind_msb
- aco: ensure predecessors' p_logical_end is in WQM when a p_phi is in WQM
- aco: run p_wqm instructions in WQM
- aco: don't consider loop header blocks branch blocks in add_coupling_code
- aco: don't always add logical edges from continue_break blocks to headers
- aco: fix target calculation when vgpr spilling introduces sgpr spilling

Samuel Pitoiset (2):

- radv: do not allow sparse resources with multi-planar formats
- nir: do not use De Morgan's Law rules for flt and fge

Tapani Pälli (2):

- mapi: add GetInteger64vEXT with EXT_disjoint_timer_query
- mesa: allow bit queries for EXT_disjoint_timer_query

Thomas Hellstrom (1):

- svga: Fix banded DMA upload

Vasily Khoruzhick (1):

- lima: ppir: don't delete root ld_tex nodes without successors in current block

Vinson Lee (1):

- swr: Fix GCC 4.9 checks.

4.25 Mesa 19.3.3 Release Notes / 2020-01-28

Mesa 19.3.3 is a bug fix release which fixes bugs found since the 19.3.2 release.

Mesa 19.3.3 implements the OpenGL 4.6 API, but the version reported by glGetString(GL_VERSION) or glGetIntegerv(GL_MAJOR_VERSION) / glGetIntegerv(GL_MINOR_VERSION) depends on the particular driver being used. Some drivers don't support all the features required in OpenGL 4.6. OpenGL 4.6 is **only** available if requested at context creation. Compatibility contexts may report a lower version depending on each driver.

Mesa 19.3.3 implements the Vulkan 1.1 API, but the version reported by the apiVersion property of the VkPhysicalDeviceProperties struct depends on the particular driver being used.

4.25.1 SHA256 checksum

```
81ce4810bb25d61300f8104856461f4d49cf7cb794aa70cb572312e370c39f09  mesa-19.3.3.tar.xz
```

4.25.2 New features

- None

4.25.3 Bug fixes

- aco: Dead Rising 4 crashes in lower_to_hw_instr() on GFX6-GFX7
- libvulkan_radeon.so crash with 'free(): double free detected in tcache 2'
- Commit be08e6a causes crash in com.android.launcher3 (Launcher)
- Mesa no longer compiles with GCC 10
- [bisected] [radeonsi] GPU hangs/resets while playing interlaced content on Kodi with VAAPI
- [radeonsi] MSAA image not copied properly after image store through texture view
- T-Rex and Manhattan onscreen performance issue on Android
- VkSamplerCreateInfo compareEnable not respected
- VkSamplerCreateInfo compareEnable not respected
- Freedreno drm softpin driver implementation leaks memory
- [POLARIS10] VRAM leak involving glTexImage2D with non-NULL data argument

4.25.4 Changes

Adam Jackson (1):

- drisw: Cache the depth of the X drawable

Andrii Simiklit (1):

- mesa/st: fix a memory leak in get_version

Bas Nieuwenhuizen (2):

- radv: Disable VK_EXT_sample_locations on GFX10.
- radv: Remove syncobj_handle variable in header.

Caio Marcelo de Oliveira Filho (1):

- intel/fs: Only use SLM fence in compute shaders

Daniel Schürmann (2):

- aco: fix unconditional demote_to_helper
- aco: rework lower_to_cssa()

Dylan Baker (3):

- docs: add SHA256 sums for 19.3.2
- cherry-ignore: Update for 19.3.3

- .pick_status.json: Update to c787b8d2a16d5e2950f209b1fcbec6e6c0388845

Eric Anholt (1):

- mesa: Fix detection of invalidating both depth and stencil.

Eric Engestrom (1):

- meson: use github URL for wraps instead of completely unreliable wrapdb

Erik Faye-Lund (8):

- docs: fix typo in html tag name
- docs: fix paragraphs
- docs: open paragraph before closing it
- docs: use code-tag instead of pre-tag
- docs: use code-tags instead of pre-tags
- docs: use code-tags instead of pre-tags
- docs: move paragraph closing tag
- docs: remove double-closed definition-list

Francisco Jerez (3):

- glsl: Fix software 64-bit integer to 32-bit float conversions.
- intel/fs/gen11+: Handle ROR/ROL in lower_simd_width().
- intel/fs/gen8+: Fix r127 dst/src overlap RA workaround for EOT message payload.

Hyunjun Ko (1):

- turnip: fix invalid VK_ERROR_OUT_OF_POOL_MEMORY

Jan Vesely (1):

- clover: Initialize Asm Parsers

Jason Ekstrand (8):

- anv: Flag descriptors dirty when gl_NumWorkgroups is used
- intel/vec4: Support scoped_memory_barrier
- intel/blorp: Fill out all the dwords of MI_ATOMIC
- anv: Don't over-advertise descriptor indexing features
- anv: Memset array properties
- anv/blorp: Rename buffer image stride parameters
- anv: Canonicalize buffer formats for image/buffer copies
- anv: Stop allocating WSI event fences off the instance

Jonathan Marek (1):

- st/mesa: don't lower YUV when driver supports it natively

Kenneth Graunke (2):

- intel/compiler: Fix illegal mutation in get_nir_image_intrinsic_image
- intel: Fix aux map alignments on 32-bit builds.

Lasse Lopperi (1):

- freedreno/drm: Fix memory leak in softpin implementation

Lionel Landwerlin (4):

- anv: fix intel perf queries availability writes
- anv: only use VkSamplerCreateInfo::compareOp if enabled
- intel/perf: expose timestamp begin for mdapi
- intel/perf: report query split for mdapi

Marek Olšák (4):

- ac/gpu_info: always use distributed tessellation on gfx10
- radeonsi: work around an LLVM crash when using llvm.amdgcn.icmp.i64.i1
- radeonsi: clean up how internal compute dispatches are handled
- radeonsi: don't invoke decompression inside internal launch_grid

Nataraj Deshpande (1):

- egl/android: Restrict minimum triple buffering for android color_buffers

Pierre-Eric Pelloux-Prayer (8):

- radeonsi: release saved resources in si_retile_dcc
- radeonsi: release saved resources in si_compute_expand_fmask
- radeonsi: release saved resources in si_compute_clear_render_target
- radeonsi: release saved resources in si_compute_copy_image
- radeonsi: release saved resources in si_compute_do_clear_or_copy
- radeonsi: fix fmask expand compute shader
- radeonsi: make sure fmask expand is done if needed
- util: call bind_sampler_states before setting sampler_views

Rhys Perry (8):

- aco: set vm for pos0 exports on GFX10
- aco: fix imageSize()/textureSize() with large buffers on GFX8
- aco: fix uninitialized data in the binary
- aco: set exec_potentially_empty for demotes
- aco: disable add combining for ds_swizzle_b32
- aco: don't DCE atomics with return values
- aco: check if multiplication/clamp is live when applying output modifier
- aco: fix off-by-one error when initializing sgpr_live_in

Samuel Pitoiset (2):

- radv: only use VkSamplerCreateInfo::compareOp if enabled
- radv: fix double free corruption in radv_alloc_memory()

Samuel Thibault (1):

- meson: Do not require libdrm for DRI2 on hurd

Tapani Pälli (1):

- egl/android: fix buffer_count for applications setting max count

Thong Thai (1):

- mesa: Prevent _MaxLevel from being less than zero

Timur Kristóf (1):

- aco/gfx10: Fix VcmpxExecWARHazard mitigation.

4.26 Mesa 19.3.2 Release Notes / 2020-01-09

Mesa 19.3.2 is a bug fix release which fixes bugs found since the 19.3.1 release.

Mesa 19.3.2 implements the OpenGL 4.6 API, but the version reported by glGetString(GL_VERSION) or glGetIntegerv(GL_MAJOR_VERSION) / glGetIntegerv(GL_MINOR_VERSION) depends on the particular driver being used. Some drivers don't support all the features required in OpenGL 4.6. OpenGL 4.6 is **only** available if requested at context creation. Compatibility contexts may report a lower version depending on each driver.

Mesa 19.3.2 implements the Vulkan 1.1 API, but the version reported by the apiVersion property of the VkPhysicalDeviceProperties struct depends on the particular driver being used.

4.26.1 SHA256 checksum

```
4e3aee324616352bbc7f58d47ab573e10f68cc7719fd045bd6d3abcdd97ee1c1  mesa-19.3.2.tar.xz
```

4.26.2 New features

- None

4.26.3 Bug fixes

- Rise of the Tomb Raider benchmark crash on Dell XPS 7390 2-in-1 w/ Iris Plus Graphics (Ice Lake 8x8 GT2)
- Raven Ridge (2400G): Resident Evil 2 crashes my machine
- Rocket League ingame artifacts
- [radv] SteamVR direct mode no longer works
- [RADV] [Navi] LOD artifacting in Halo - The Master Chief Collection (Halo Reach)
- [ANV] unused create parameters not properly ignored
- Blocky corruption in The Surge 2
- radeonsi: Floating point exception on R9 270 gpu for a set of traces
- [CTS] dEQP-VK.api.image_clearing.core.clear_color_image.2d.linear.single_layer.r32g32b32_* fail on GFX6-GFX8
- Vulkan: Please consider adding another sample count to sampledImageIntegerSampleCounts
- Navi10: Bitrate based encoding with VAAPI/RadeonSI unusable

- [GFX10] Glitch rendering Custom Avatars in Beat Saber
- intel/fs: Check for 16-bit immediates in fs_visitor::lower_mul_dword_inst is too strict

4.26.4 Changes

Andrii Simiklit (3):

- glsl: fix an incorrect max_array_access after optimization of ssbo/ubo
- glsl: fix a binding points assignment for ssbo/ubo arrays
- glsl/nir: do not change an element index to have correct block name

Bas Nieuwenhuizen (7):

- radv: Limit workgroup size to 1024.
- radv: Expose all sample counts for integer formats as well.
- amd/common: Handle alignment of 96-bit formats.
- nir: Add clone/hash/serialize support for non-uniform tex instructions.
- spirv: Fix glsl type assert in spir2nir.
- radv: Only use the gfx mipmap level offset/pitch for linear textures.
- radv: Emit a BATCH_BREAK when changing pixel shaders or CB_TARGET_MASK.

Caio Marcelo de Oliveira Filho (4):

- intel/fs: Lower 64-bit MOVs after lower_load_payload()
- intel/fs: Fix lowering of dword multiplication by 16-bit constant
- intel/vec4: Fix lowering of multiplication by 16-bit constant
- anv: Ignore some CreateInfo structs when rasterization is disabled

Christian Gmeiner (1):

- etnaviv: update resource status after flushing

Dylan Baker (2):

- dcos: add releanse notes for 19.3.1
- cherry-ignore: update for 19.3.2

Eric Engestrom (4):

- util/format: remove left-over util_format_description_table declaration
- amd: fix empty-body issues
- nine: fix empty-body-issues
- mesa: avoid returning a value in a void function

Gert Wollny (1):

- r600: Fix maximum line width

Jason Ekstrand (2):

- anv: Properly advertise sampledImageIntegerSampleCounts
- intel/nir: Add a memory barrier before barrier()

Lionel Landwerlin (2):

- loader: fix close on uninitialized file descriptor value
- anv: don't close invalid syncfd semaphore

Marek Olšák (2):

- winsys/radeon: initialize pte_fragment_size
- radeonsi: disable SDMA on gfx8 to fix corruption on RX 580

Pierre-Eric Pelloux-Prayer (2):

- radeon/vcn2: enable rate control for hevc encoding
- radeonsi: check ctx->sdma_cs before using it

Samuel Pitoiset (2):

- radv/gfx10: fix the out-of-bounds check for vertex descriptors
- radv: return the correct pitch for linear mipmaps on GFX10

Timur Kristóf (1):

- aco: Fix uniform i2i64.

Yevhenii Kolesnikov (2):

- meta: Cleanup function for DrawTex
- main: allow external textures for BindImageTexture

4.27 Mesa 19.2.8 Release Notes / 2019-12-18

Mesa 19.2.8 is a bug fix release which fixes bugs found since the 19.2.7 release.

Mesa 19.2.8 implements the OpenGL 4.5 API, but the version reported by glGetString(GL_VERSION) or glGetIntegerv(GL_MAJOR_VERSION) / glGetIntegerv(GL_MINOR_VERSION) depends on the particular driver being used. Some drivers don't support all the features required in OpenGL 4.5. OpenGL 4.5 is **only** available if requested at context creation. Compatibility contexts may report a lower version depending on each driver.

Mesa 19.2.8 implements the Vulkan 1.1 API, but the version reported by the apiVersion property of the VkPhysicalDeviceProperties struct depends on the particular driver being used.

4.27.1 SHA256 checksum

```
cffa8fa755c7422ce014c39ca0b770a092d9e0bbae537ceb2609c106916e5a57  mesa-19.2.8.tar.xz
```

4.27.2 New features

- None

4.27.3 Bug fixes

- i965/iris: assert when destroy GL context with active query

4.27.4 Changes

Alyssa Rosenzweig (1):

- gallium/util: Support POLYGON in u_stream_outputs_for_vertices

Bas Nieuwenhuizen (2):

- amd/common: Always use addrlib for HTILE tc-compat.
- amd/common: Fix tcCompatible degradation on Stoney.

Dylan Baker (4):

- docs: Add SHA256 sums for 19.2.7
- meson/broadcom: libbroadcom_cle needs expat headers
- meson/broadcom: libbroadcom_cle also needs zlib
- cherry-ignore: Update for 19.2.8

Gert Wollny (1):

- virgl: Increase the shader transfer buffer by doubling the size

Iván Briano (1):

- anv: Export filter_minmax support only when it's really supported

Jason Ekstrand (2):

- anv: Re-emit all compute state on pipeline switch
- anv: Don't leak when set_tiling fails

Kenneth Graunke (1):

- iris: Default to X-tiling for scanout buffers without modifiers

Lionel Landwerlin (7):

- intel/perf: fix invalid hw_id in query results
- intel/perf: set read buffer len to 0 to identify empty buffer
- intel/perf: take into account that reports read can be fairly old
- intel/perf: simplify the processing of OA reports
- intel/perf: fix improper pointer access
- anv: fix fence underlying primitive checks
- mesa: avoid triggering assert in implementation

Nanley Chery (2):

- gallium/dri2: Fix creation of multi-planar modifier images
- gallium: Store the image format in winsys_handle

Rob Clark (1):

- nir/lower_clip: Fix incorrect driver loc for clipdist outputs

Timothy Arceri (1):

- glsl/nir: iterate the system values list when adding varyings

4.28 Mesa 19.3.1 Release Notes / 2019-12-18

Mesa 19.3.1 is a bug fix release which fixes bugs found since the 19.3.0 release.

Mesa 19.3.1 implements the OpenGL 4.6 API, but the version reported by glGetString(GL_VERSION) or glGetIntegerv(GL_MAJOR_VERSION) / glGetIntegerv(GL_MINOR_VERSION) depends on the particular driver being used. Some drivers don't support all the features required in OpenGL 4.6. OpenGL 4.6 is **only** available if requested at context creation. Compatibility contexts may report a lower version depending on each driver.

Mesa 19.3.1 implements the Vulkan 1.1 API, but the version reported by the apiVersion property of the VkPhysicalDeviceProperties struct depends on the particular driver being used.

4.28.1 SHA256 checksum

```
cd951db69c56a97ff0570a7ab2c0e39e6c5323f4cd8f4eb8274723e033beae59  mesa-19.3.1.tar.xz
```

4.28.2 New features

- None

4.28.3 Bug fixes

- i965/iris: assert when destroy GL context with active query
- Visuals without alpha bits are not sRGB-capable
- radv secure compile feature breaks compilation of RADV on armhf EABI (19.3-rc1)

4.28.4 Changes

Bas Nieuwenhuizen (2):

- amd/common: Fix tcCompatible degradation on Stoney.
- amd/common: Always use addrlib for HTILE tc-compat.

Dylan Baker (3):

- docs/19.3.0: Add SHA256 sums
- cherry-ignore: update for the 19.3.1 cycle
- docs: remove new_features.txt from stable branch

Gert Wollny (1):

- virgl: Increase the shader transfer buffer by doubling the size

Iván Briano (1):

- anv: Export filter_minmax support only when it's really supported

Kenneth Graunke (1):

- iris: Default to X-tiling for scanout buffers without modifiers

Lionel Landwerlin (2):

- anv: fix fence underlying primitive checks
- mesa: avoid triggering assert in implementation

Luis Mendes (1):

- radv: fix radv secure compile feature breaks compilation on armhf EABI and aarch64

Tapani Pälli (2):

- dri: add __DRI_IMAGE_FORMAT_SXRGB8
- i965: expose MESA_FORMAT_B8G8R8X8_SRGB visual

4.29 Mesa 19.3.0 Release Notes / 2019-12-12

Mesa 19.3.0 is a new development release. People who are concerned with stability and reliability should stick with a previous release or wait for Mesa 19.3.1.

Mesa 19.3.0 implements the OpenGL 4.6 API, but the version reported by glGetString(GL_VERSION) or glGetIntegerv(GL_MAJOR_VERSION) / glGetIntegerv(GL_MINOR_VERSION) depends on the particular driver being used. Some drivers don't support all the features required in OpenGL 4.6. OpenGL 4.6 is **only** available if requested at context creation. Compatibility contexts may report a lower version depending on each driver.

Mesa 19.3.0 implements the Vulkan 1.1 API, but the version reported by the apiVersion property of the VkPhysicalDeviceProperties struct depends on the particular driver being used.

4.29.1 SHA256 checksum

```
5fa0e4e9dca79560f6882e362f9db36d81cf96da16cf6a84e0ada7466a99a5d7  mesa-19.3.0.tar.xz
```

4.29.2 New features

- GL_ARB_gl_spirv on i965, iris.
- GL_ARB_spirv_extensions on i965, iris.
- GL_EXT_demote_to_helper_invocation on iris, i965.
- OpenGL 4.6 on i965, iris.
- EGL_EXT_image_flush_external
- VK_ANDROID_external_memory_android_hardware_buffer on RADV.
- VK_KHR_shader_clock on Intel, RADV.
- VK_KHR_shader_float_controls on Intel, RADV.
- VK_KHR_spirv_1_4 on Intel, RADV.
- VK_KHR_timeline_semaphore on RADV.
- VK_KHR_vulkan_memory_model on Intel.
- VK_EXT_shader_subgroup_ballot on Intel.
- VK_EXT_shader_subgroup_vote on Intel.
- VK_EXT_texel_buffer_alignment on RADV.

- VK_INTEL_performance_query on Intel.
- Meson support for windows using MSVC and MinGW
- scons has been deprecated for non windows
- Initial Intel gen12 (Tigerlake) support on anvil and iris
- New compiler backend "ACO" for RADV (RADV_PERFTEST=aco)
- VK_EXT_shader_demote_to_helper_invocation on RADV/ACO.

4.29.3 Bug fixes

- [RADV] The Dead Rising 4 is causing a GPU hang with LLVM backend
- radeonsi: mpv –vo=vaapi incorrect rendering on gfx9+
- NULL resource when playing VP9 video through VDPAU on RX 570
- gnome-shell overview mode crash in recent mesa
- radv/aco Jedi Fallen Order hair rendering buggy
- [RADV] VK_KHR_timeline_semaphore balloons in runtime
- Shadow of Mordor has randomly dancing black shadows on Talion's face
- ld.lld: error: duplicate symbol (mesa-19.3.0-rc1)
- triangle strip clipping with GL_FIRST_VERTEX_CONVENTION causes wrong vertex's attribute to be broadcasted for flat interpolation
- [bisected][regression][g45,g965,ilk] piglit arb_fragment_program kil failures
- textureSize(samplerExternalOES, int) missing in desktop mesa 19.1.7 implementation
- HSW. Tropico 6 and SuperTuxKart have shadows flickering
- glxgears segfaults on POWER / Xvnc
- Objects leaving trails in Firefox with antialias and preserveDrawingBuffer in three.js WebGLRednerer with mesa 19.2
- radv regression after 84d9551b232bdcead017b212cbb3e291486e698c: vk: error: failed to submit CS
- Rename ACO README to README.md
- Steam crash due to commit e137b3a9b71a2711c1f68c8a8b9c0a7407fbcc4b (bisected)
- [Anv regression] SPIR-V abort in Aztec Ruins
- FreeBSD does not have _GNU_SOURCE in util/strtod.c
- glLinkProgram crash when using gcc-9 -O3 -flto due to use of uninitialised value
- KeyError: 'force_scons':
- link_shader and deserialize_glsl_program suddenly consume huge amount of RAM
- build errors after "meson: add -Werror=empty-body to disallow 'if(x);'"
- performance regression in Heroes of the Storm with Mesa 19.1.1 & Polaris
- Vulkan version of "Middle-earth: Shadow of Mordor" has graphics glitches on RADV driver (part 2)
- swr/rasterizer/core/format_types.h:1183: undefined reference to '_mm256_cvtps_ph'
- Meson: Building osmesa gallium and tests at the same time results in osmesa gallium build failure

- Vulkan version of "Middle-earth: Shadow of Mordor" has graphics glitches on RADV driver
- [amdgpu][Navi][llvm] Minimap problem in Nier Automata
- [bisected] anon_inode:sync_file file descriptor leak
- Cache meson packagecach in appveyor
- Piglit tests regression in gallium drivers
- Black ground in Dirt 4
- Superbibles examples crashing Mesa drivers (radeonsi) and causing gpu reset
- [CTS] dEQP-VK.graphicsfuzz.write-red-in-loop-nest crashes
- mesa and libglvnd install the same headers
- Multiple EGL displays with multiple window systems leads to a crash
- Regression: Doom (2016) crashes on Mesa 19.2 and above and Radeon 380 with Vulkan (worked on Mesa 19.1)
- Rocket League displays corruption when the game starts
- drm.h:50:9: error: unknown type name 'uint8_t'
- Mesa build breaks when only building radeonsi due to missing llvm coroutines symbols
- radeonsi aborting in LLVM validation test in si_compile_tgsi_shader()
- meson.build:1447:6: ERROR: Problem encountered: libdrm required for gallium video statetrackers when using x11
- Mesa doesn't build with current Scons version (3.1.0)
- libXvMC-1.0.12 breaks mesa build
- Meson can't find 32-bit libXvMCW in non-standard path
- Mesa installs gl.pc and egl.pc even with libglvnd >= 1.2.0

4.29.4 Changes

Adam Jackson (44):

- glx: Whitespace cleanups
- glx: Sync <GL/glxext.h> with Khronos
- glx: Make __glXGetDrawableAttribute return true sometimes
- glx: Unset the direct_support bit for GLX_EXT_import_context
- Revert "glx: Unset the direct_support bit for GLX_EXT_import_context"
- egl: Enable 10bpc EGLConfigs for platform_{device,surfaceless}
- gallium/xlib: Fix an obvious thinko
- mesa: Remove unused gl_config::indexBits
- mesa: Eliminate gl_config::have{Accum,Depth,Stencil}Buffer
- mesa: Eliminate gl_config::rgbMode
- gallium: Require LLVM >= 3.4
- gallium: Require LLVM >= 3.5

- gallium: Require LLVM >= 3.6
- gallium: Require LLVM >= 3.7
- gallium: Require LLVM >= 3.8
- gallium: Require LLVM >= 3.9
- egl/dri2: Refuse to add EGLConfigs with no supported surface types
- glx: Remove unused indirection for glx_context->fillImage
- gallium: Restore VSX for llvm >= 4
- ci: Run tests on i386 cross builds
- gallium/xlib: Remove drawable caching from the MakeCurrent path
- gallium/xlib: Remove MakeCurrent_PrevContext
- gallium/xlib: Fix glXMakeCurrent(dpy, None, None, ctx)
- docs: Update bug report URLs for the gitlab migration
- glx: Avoid atof() when computing the server's GLX version
- glx: Fix drawable lookup bugs in glXUseXFont
- egl/wayland: Reindent the format table
- egl/wayland: Add FP16 format support
- egl/wayland: Implement getCapability for the dri2 and image loaders
- egl/surfaceless: Add FP16 format support
- libgbm: Wire up getCapability for the image loader
- glx: Move vertex array protocol state into the indirect backend
- glx: Lift sending the MakeCurrent request to top-level code
- glx: Implement GLX_EXT_no_config_context
- Revert "glx: Implement GLX_EXT_no_config_context"
- Revert "glx: Lift sending the MakeCurrent request to top-level code"
- drisw: Simplify GC setup
- drisw: Fix and simplify drawable setup
- glx: Log the filename of the drm device if we fail to open it
- egl/dri2: Don't dlclose() the driver on dri2_load_driver_common failure
- surfaceless: Support EGL_WL_bind_wayland_display
- egl: Make native display detection work more than once
- gallium/xlib: Fix xmesa drawable creation

Alan Coopersmith (6):

- gallium: Fix a bunch of undefined left-shifts in u_format_*
- c99_compat.h: Don't try to use 'restrict' in C++ code
- util: Make Solaris implemention of p_atomic_add work with gcc
- util: Workaround lack of flock on Solaris

- util: Solaris has linux-style pthread_setname_np
- meson: recognize "sunos" as the system name for Solaris
- intel/common: include unistd.h for ioctl() prototype on Solaris

Alejandro Piñeiro (5):

- i965: enable ARB_gl_spirv extension and ARB_spirv_extensions for gen7+
- mesa/version: uncomment SPIR-V extensions
- i965: Enable OpenGL 4.6 for Gen8+
- v3d: take into account prim_counts_offset
- v3d: adds an extra MOV for any sig.ld*

Alex Smith (1):

- radv: Change memory type order for GPUs without dedicated VRAM

Alexandros Frantzis (1):

- gitlab-ci: Update required libdrm version

Alyssa Rosenzweig (220):

- pan/decode: Eliminate DYN_MEMORY_PROP
- pan/decode: Don't print MALI_DRAW_NONE
- panfrost: Move pan_invocation to shared panfrost/
- panfrost: Set workgroups z to 32 for non-instanced graphics
- pan/decode: Don't print canonical workgroup encoding
- panfrost: Implement workgroups_x_shift_2 quirk
- pan/decode: Silence workgroups_x_shift_2
- pan/decode: Fix missing NULL terminator
- pan/decode: Don't print zero exception_status
- pan/decode: Express tiler structures as offsets
- pan/decode: Allow updating mmaps
- pan/decode: Bounds check polygon list and tiler heap
- panfrost: Move pan_tiler.c outside of Gallium
- pan/decode: Verify and omit polygon size
- pan/decode: Print "just right" count of texture pointers
- panfrost: Remove DRY_RUN
- panfrost: Correct polygon size computations
- pan/decode: Check for a number of potential issues
- pan/decode: Don't print unreferenced attribute memory
- pan/decode: Add static bounds checking utility
- pan/decode: Do not print uniform/buffers explicitly
- pan/decode: Validate AFBC fields are zero when AFBC is disabled

- pan/decode: Check for MFBD preload chicken bit
- pan/decode: Mark tripped zeroes with XXX
- pan/decode: Normalize case matching XXX format
- pan/decode: Normalize final instances of XXX
- panfrost: Fix scoreboarding with dependency on job #0
- panfrost: Do not expose PIPE_CAP_TEXTURE_MIRROR_CLAMP
- panfrost: Don't crash on GL_CLAMP
- pan/decode: Guard attribute unknowns
- panfrost: Don't trip the prefix magic field
- pan/decode: Handle VARYING_DISCARD
- pan/decode: Treat RESERVED swizzles as errors
- pan/decode: Validate swizzles against format
- pan/decode: Don't print the default swizzle
- pan/decode: Use GLSL style formats/swizzles
- pan/decode: Guard texture unknowns as zero trips
- pan/decode: Break out pandecode_texture function
- pan/decode: Validate texture dimensionality
- panfrost: nr_mipmap_levels -> levels
- panfrost: Remove ancient TODO
- pan/decode: Pretty-print sRGB format
- panfrost: Break up usage2 field
- pan/decode: Use concise texture printing
- pan/decode: Include address in union mali_attr
- pan/decode: Validate attribute/varying buffer pointer
- pan/decode: Cleanup mali_attr printing
- pan/midgard: Free liveness info
- pan/midgard: Allocate 'dependencies' on stack
- pan/decode: Don't leak FBD pointer
- pan/decode: Remove all_zero
- pan/bifrost: Avoid buffer overflow in disassembler
- pan/midgard: Represent unused nodes by ~0
- pan/midgard: Reorder bits check to fix 8-bit masks
- pan/midgard: Simplify contradictory check.
- panfrost: Don't check reads_point_coord
- pan/midgard: Mark fallthrough explicitly
- panfrost: Pay attention to framebuffer dimension sign

- panfrost: Clarify intention with PIPE_SWIZZLE_X check
- panfrost: Prevent potential integer overflow in instancing
- panfrost: Hoist job != NULL check
- panfrost: Hoist bo != NULL check before dereference
- panfrost: Fix missing ret assignment in DRM code
- pan/bifrost: Correct file size signedness
- panfrost: Guard against NULL rasterizer explicitly
- panfrost: Pass stream_output_info by reference
- pan/midgard: Breakout texture reg select printer
- pan/midgard: Identify and disassemble indirect texture/sampler
- panfrost: Don't bail on PIPE_BUFFER
- panfrost: Implement depth range clipping
- panfrost: Fix PIPE_BUFFER spacing
- pan/midgard,bifrost: Expand nir_const_load_to_arr
- nir: Remove nir_const_load_to_arr
- pan/decode: Hoist shader-db stats to shared decode
- pan/midgard: Sketch static analysis to uniform count
- pan/midgard: Compute work_count via writes
- pan/midgard: Analyze simple loads/store
- pan/midgard: Explain ffma
- pan/midgard: Disassemble integer constants in hex
- pan/decode: Remove mali_attr(_meta) framing
- pan/decode: Removing uniform buffer framing
- pan/decode: Eliminate non-FBD dumped case
- pan/decode: Validate MFBD tags
- pan/decode: Validate and simplify FRAGMENT payloads
- pan/decode: Validate blend shaders don't access I/O
- pan/decode: Fix uniform printing
- pan/decode: Promote <no shader> to an error
- pan/decode: Disassemble before printing shader descriptor
- pan/decode: Validate mali_shader_meta stats
- pan/decode: Validate, but do not print, index buffer
- pan/decode: Downgrade shader property mismatch to warning
- pan/decode: Decode actual varying_meta address
- pan/decode: Print stub for uniforms
- pan/decode: Decouple attribute/meta printing

- pan/decode: Remove size/stride divisibility check
- pan/decode: Handle special varyings
- panfrost: Remove vertex buffer offset from its size
- panfrost: Implement gl_FragCoord correctly
- pan/midgard: Fix writeout combining
- pan/midgard: Analyze helper invocations
- pan/decode: Validate and quiet helper invocation flag
- pan/midgard, bifrost: Set lower_fdph = true
- pan/midgard: Switch constants to uint32
- pan/midgard: Add imov->fmov optimization
- pan/midgard: Fold ssa_args into midgard_instruction
- pan/midgard: Fix invert fusing with r26
- freedreno/ir3: Link directly to Sethi-Ullman paper
- pan/midgard: Count shader-db stats by bundled instructions
- pan/midgard: Factor out mir_is_scalar
- pan/midgard: Extract instruction sizing helper
- pan/midgard: Expose mir_get/set_swizzle
- pan/midgard: Add OP_IS_CSEL_V helper
- pan/midgard: Fix corner case in RA
- pan/midgard: Add post-schedule iteration helpers
- pan/midgard: Include condition in branch->src[0]
- pan/midgard: Document Midgard scheduling requirements
- pan/midgard: Ensure fragment writeout is in the final block
- pan/midgard: Track csel swizzle
- pan/midgard: Add mir_insert_instruction*scheduled helpers
- pan/midgard: csel_swizzle with mir get swizzle
- pan/midgard: Extend mir_special_index to writeout
- pan/midgard: Improve mir_mask_of_read_components
- pan/midgard: Allow NULL argument in mir_has_arg
- pan/midgard: Track shader quadword count while scheduling
- pan/midgard: Add scheduling barriers
- pan/midgard: Cleanup fragment writeout branch
- pan/midgard: Remove texture_index
- pan/midgard: Print branches in MIR
- pan/midgard: Print MIR by the bundle
- pan/midgard: Fix misc. RA issues

- pan/midgard: Do not propagate swizzles into writeout
- pan/midgard: Handle fragment writeout in RA
- pan/midgard: Schedule before RA
- pan/midgard: Remove mir_opt_post_move_eliminate
- pan/midgard: Use shared psiz clamp pass
- pan/decode: Fix uninitialized variables
- pan/decode: Use %zu instead of %d
- pan/decode: Use portable format specifier for 64-bit
- pan/decode: Add missing format specifier
- pan/midgard: Correct issues in disassemble.c
- pan/midgard: Fix cppcheck issues
- pan/midgard: Remove cppwrap.cpp
- pan/midgard: Remove mir_print_bundle
- pan/midgard: Remove mir_rewrite_index_*_tag
- panfrost: Mark (1 << 31) as unsigned
- panfrost: Fix misc. issues flagged by cppcheck
- panfrost: Remove panfrost_upload
- pan/midgard: Add missing parans in SWIZZLE definition
- pan/midgard: Fix component count handling for ldst
- pan/midgard: Squeeze indices before scheduling
- pan/midgard: Add flatten_mir helper
- pan/midgard: Calculate dependency graph
- pan/midgard: Initialize worklist
- pan/midgard: Add mir_choose_instruction stub
- pan/midgard: Add mir_update_worklist helper
- pan/midgard: Add mir_choose_bundle helper
- pan/midgard: Add mir_schedule_texture/ldst/alu helpers
- pan/midgard: Remove csel constant unit force
- pan/midgard: Add constant intersection filters
- pan/midgard: Add predicate->exclude
- pan/midgard: Implement predicate->unit
- pan/midgard: Add helpers for scheduling conditionals
- pan/midgard: Extend csel_swizzle to branches
- pan/midgard: Implement load/store pairing
- pan/midgard: Add mir_choose_alu helper
- pan/midgard: Add distance metric to choose_instruction

- pan/midgard: Use new scheduler
- pan/midgard: Don't double check SCALAR units
- pan/midgard: Extend choose_instruction for scalar units
- pan/midgard: Schedule to smul/sadd
- pan/midgard: Only one conditional per bundle allowed
- pan/midgard: Allow 6 instructions per bundle
- pan/midgard: Allow writeout to see into the future
- pan/midgard: Tightly pack 32-bit constants
- pan/midgard: Add mir_flip helper
- pan/midgard: Add csel invert optimization
- pan/midgard: Allow scheduling conditions with constants
- pan/midgard: Remove mir_has_multiple_writes
- pan/midgard: Add mir_calculate_temp_count helper
- pan/midgard: Move RA's liveness analysis into midgard_liveness.c
- pan/midgard: Don't try to OR live_in of successors
- pan/midgard: Begin tracking liveness metadata
- pan/midgard: Invalidate liveness for mir_is_live_after
- pan/midgard: Calculate temp_count for liveness
- pan/midgard: Replace mir_is_live_after with new pass
- pan/midgard: Report read mask for branch arguments
- pan/midgard: Allow non-contiguous masks in UBO lowering
- pan/midgard: Don't try to propagate swizzles to branches
- pan/midgard: Add perspective ops to mir_get_swizzle
- pan/midgard: Fix mir_mask_of_read_components with dot products
- panfrost: Disable frame throttling
- pan/midgard: Use 16-bit liveness masks
- pan/midgard: Allow COMPUTE jobs in panfrost_bo_access_for_stage
- pan/midgard: Fix memory corruption in register spilling
- pan/midgard: Do not repeatedly spill same value
- pan/midgard: Debug mir_insert_instruction_after_scheduled
- pan/midgard: Identify 64-bit atomic opcodes
- pan/midgard/disasm: Fix printing 8-bit/16-bit masks
- pan/midgard: Factor out mir_get_alu_src
- pan/midgard: Tableize load/store ops
- pan/midgard: Implement OP_IS_STORE with table
- pan/midgard: Add helpers for manipulating byte masks

- pan/midgard: Report byte masks for read components
- pan/midgard: Simplify mir_bytemask_of_read_components
- pan/midgard: Implement per-byte liveness tracking
- pan/midgard: Handle nontrivial masks in texture RA
- pan/midgard: Create dependency graph bytewise
- pan/midgard: Implement SIMD-aware dead code elimination
- panfrost/ci: Update expectations list
- pan/midgard: Add mir_set_bytemask helper
- pan/midgard: Expose more typesize manipulation routines
- pan/midgard: Express allocated registers as offsets
- pipe-loader: Add kmsro pipe_loader target
- pipe-loader: Default to kmsro if probe fails
- panfrost: Expose serialized NIR support
- pan/midgard: Disable precise occlusion queries
- panfrost: Cleanup _shader_upper -> shader
- panfrost: Remove unused definitions in mali-job.h
- pipe-loader: Build kmsro loader for with all kmsro targets
- gallium/util: Support POLYGON in u_stream_outputs_for_vertices

Andreas Baierl (5):

- lima/ppir: Rename ppir_op_dummy to ppir_op_undef
- lima/ppir: Add undef handling
- lima/ppir: Add various varying fetch sources to disassembler
- lima: Fix compiler warning in standalone compiler
- lima: Fix crash when there are no vertex shader attributes

Andreas Gottschling (1):

- drisw: Fix shared memory leak on drawable resize

Andres Gomez (12):

- nir/algebraic: mark float optimizations returning one parameter as inexact
- docs: Update to OpenGL 4.6 in the release notes
- nir/opcodes: Clear variable names confusion
- docs: Add the maximum implemented Vulkan API version in 19.1 rel notes
- docs: Add the maximum implemented Vulkan API version in 19.2 rel notes
- docs: Add the maximum implemented Vulkan API version in 19.3 rel notes
- docs/features: Update status list of Vulkan extensions
- docs/features: Update VK_KHR_display_swapchain status
- i965/fs: add a comment about how the rounding mode in fmul is set

- i965/fs: set rounding mode when emitting the flrp instruction
- docs/relnotes: add support for GL_ARB_gl_spirv, GL_ARB_spirv_extensions and OpenGL 4.6 on i965 and iris
- egl: Remove the 565 pbuffer-only EGL config under X11.

Andres Rodriguez (2):

- radv: add RADV_DEBUG=allentrypoints
- radv: additional query fixes

Andrii Simiklit (1):

- glsl: disallow incompatible matrices multiplication

Anuj Phogat (5):

- intel/gen12: Add L3 configurations
- intel: Add few Ice Lake brand strings
- genxml/gen11+: Add COMMON_SLICE_CHICKEN4 register
- intel/gen11+: Enable Hardware filtering of Semi-Pipelined State in WM
- intel/isl/icl: Use halign 8 instead of 4 hw workaround

Arcady Goldmints-Orlov (1):

- anv: fix descriptor limits on gen8

Bas Nieuwenhuizen (63):

- radv: Use correct vgpr_comp_cnt for VS if both prim_id and instance_id are needed.
- radv: Emit VGT_GS_ONCHIP_CNTL for tess on GFX10.
- radv: Disable NGG for geometry shaders.
- tu: Set up glsl types.
- radv: Only break batch on framebuffer change with dfsm.
- radv: Disable dfsm by default even on Raven.
- radv: Add DFSM support.
- glx: Remove redundant null check.
- amd: Build aco only if radv is enabled
- radv: Add workaround for hang in The Surge 2.
- turnip: Add image->image blitting.
- turnip: Always use UINT formats for copies.
- turnip: Disallow NPoT formats.
- turnip: Add todo for d24_s8 copies
- radv: Fix condition for skipping the continue CS.
- radv: Fix warning in 32-bit build.
- meson: Always add LLVM coroutines module.
- amd/llvm: Fix warning due to asserted-only variable.
- radv: Implement & enable VK_EXT_texel_buffer_alignment.

- radv: Cleanup buffer_from_fd.
- radv: Handle device memory alloc failure with normal free.
- radv: Split out layout code from image creation.
- radv: Delay patching for imported images until layout time.
- radv: Handle slightly different image dimensions.
- radv: Unset vk_info in radv_image_create_layout.
- radv: Add VK_ANDROID_external_memory_android_hardware_buffer.
- radv/android: Add android hardware buffer field to device memory.
- radv/android: Add android hardware buffer queries.
- radv: Disallow sparse shared images.
- radv: Derive android usage from create flags.
- radv: Deal with Android external formats.
- radv/android: Add android hardware buffer import/export.
- radv: Allow Android image binding.
- radv: Expose image handle compat types for Android handles.
- radv: Check the size of the imported buffer.
- radv: Enable VK_ANDROID_external_memory_android_hardware_buffer.
- nir/dead_cf: Remove dead control flow after infinite loops.
- radv: Fix single stage constant flush with merged shaders.
- radv: Compute hashes in secure process for secure compilation.
- radv: Add an early exit in the secure compile if we already have the cache entries.
- radv: Clean up unused variable.
- radv: Split out commandbuffer submission.
- radv: Do sparse binding in queue submission.
- radv: Improve fence signalling in QueueSubmit.
- radv: Always enable syncobj when supported for all fences/semaphores.
- radv: Split semaphore into two parts as enum+union.
- radv: Add temporary datastructure for submissions.
- radv: Add timelines with a VK_KHR_timeline_semaphore impl.
- radv: Add wait-before-submit support for timelines.
- radv: Enable VK_KHR_timeline_semaphore.
- radv: Start signalling semaphores in WSI acquire.
- radv: Allocate space for temp. semaphore parts.
- radv: Fix timeout handling in syncobj wait.
- radv: Remove _mesa_locale_init/fini calls.
- turnip: Remove _mesa_locale_init/fini calls.

- anv: Remove _mesa_locale_init/fini calls.
- radv: Fix disk_cache_get size argument.
- radv: Close all unnecessary fds in secure compile.
- radv: Do not change scratch settings while shaders are active.
- radv: Allocate cmdbuffer space for buffer marker write.
- radv: Unify max_descriptor_set_size.
- radv: Fix timeline semaphore refcounting.
- radv: Fix RGBX Android<->Vulkan format correspondence.

Ben Crocker (1):

- llvmpipe: use ppc64le/ppc64 Large code model for JIT-compiled shaders

Boris Brezillon (73):

- panfrost: Free the instruction object in mir_remove_instruction()
- panfrost: Free all block/instruction objects before leaving midgard_compile_shader_nir()
- panfrost: Make sure bundle.instructions[] contains valid instructions
- Revert "panfrost: Free all block/instruction objects before leaving midgard_compile_shader_nir()"
- panfrost: Use ralloc() to allocate instructions to avoid leaking those objs
- panfrost: Reset the damage area on imported resources
- panfrost: Add transient BOs to job batches
- panfrost: s/job/batch/
- panfrost: Pass a batch to panfrost_drm_submit_vs_fs_batch()
- panfrost: Stop passing a ctx to functions being passed a batch
- panfrost: Make transient allocation rely on the BO cache
- panfrost: Convert ctx->{scratchpad, tiler_heap, tiler_dummy} to plain BOs
- panfrost: Get rid of unused panfrost_context fields
- panfrost: Get rid of the now unused SLAB allocator
- panfrost: Rename pan_bo_cache.c into pan_bo.c
- panfrost: Fix a list_assert() in schedule_block()
- panfrost: Rework midgard_pair_load_store() to kill the nested foreach loop
- panfrost: Use a pipe_framebuffer_state as the batch key
- panfrost: Get rid of the unused 'flush jobs accessing res' infra
- panfrost: Allow testing if a specific batch is targeting a scanout FB
- panfrost: Pass a batch to panfrost_{allocate,upload}_transient()
- panfrost: Pass a batch to functions emitting FB descs
- panfrost: Use ctx->wallpaper_batch in panfrost_blit_wallpaper()
- panfrost: Pass a batch to panfrost_set_value_job()
- panfrost: Prepare things to avoid flushes on FB switch

- panfrost: Delay payloads[].offset_start initialization
- panfrost: Move the fence creation in panfrost_flush()
- panfrost: Move the batch submission logic to panfrost_batch_submit()
- panfrost: Stop exposing internal panfrost_*_batch() functions
- panfrost: Use the correct type for the bo_handle array
- panfrost: Add missing panfrost_batch_add_bo() calls
- panfrost: Add polygon_list to the batch BO set at allocation time
- panfrost: Kill a useless memset(0) in panfrost_create_context()
- panfrost: Stop passing has_draws to panfrost_drm_submit_vs_fs_batch()
- panfrost: Get rid of pan_drm.c
- panfrost: Move panfrost_bo_{reference,unreference}() to pan_bo.c
- panfrost: s/PAN_ALLOCATE_/PAN_BO_/
- panfrost: Move the BO API to its own header
- panfrost: Stop exposing panfrost_bo_cache_{fetch,put}()
- panfrost: Don't check if BO is mmaped before calling panfrost_bo_mmap()
- panfrost: Stop passing screen around for BO operations
- panfrost: Stop using panfrost_bo_release() outside of pan_bo.c
- panfrost: Add panfrost_bo_{alloc,free}()
- panfrost: Don't return imported/exported BOs to the cache
- panfrost: Add the panfrost_batch_create_bo() helper
- panfrost: Add FBO BOs to batch->bos earlier
- panfrost: Allocate tiler and scratchpad BOs per-batch
- Revert "panfrost: Rework midgard_pair_load_store() to kill the nested foreach loop"
- panfrost: Fix indexed draws
- dEQP-GLES2.functional.buffer.write.use.index_array.* are passing now.
- panfrost: Add the shader BO to the batch in patch_shader_state()
- panfrost: Extend the panfrost_batch_add_bo() API to pass access flags
- panfrost: Make panfrost_batch->bos a hash table
- panfrost: Add a batch fence
- panfrost: Use the per-batch fences to wait on the last submitted batch
- panfrost: Add a panfrost_freeze_batch() helper
- panfrost: Start tracking inter-batch dependencies
- panfrost: Prepare panfrost_fence for batch pipelining
- panfrost: Add a panfrost_flush_all_batches() helper
- panfrost: Add a panfrost_flush_batches_accessing_bo() helper
- panfrost: Add flags to reflect the BO imported/exported state

- panfrost: Make sure the BO is 'ready' when picked from the cache
- panfrost: Do fine-grained flushing when preparing BO for CPU accesses
- panfrost: Kill the explicit serialization in panfrost_batch_submit()
- panfrost: Get rid of the flush in panfrost_set_framebuffer_state()
- Revert "st/dri2: Implement DRI2bufferDamageExtension"
- Revert "Revert "st/dri2: Implement DRI2bufferDamageExtension""
- panfrost: Make sure a clear does not re-use a pre-existing batch
- panfrost: Draw the wallpaper when only depth/stencil bufs are cleared
- panfrost: Fix support for packed 24-bit formats
- panfrost: Fix the DISCARD_WHOLE_RES case in transfer_map()
- gallium: Fix the ->set_damage_region() implementation
- panfrost: Make sure we reset the damage region of RTs at flush time

Brian Paul (3):

- st/nir: fix illegal designated initializer in st_glsl_to_nir.cpp
- REVIEWERS: add VMware reviewers
- Call shmget() with permission 0600 instead of 0777

Caio Marcelo de Oliveira Filho (66):

- intel/compiler: Silence maybe-uninitialized warning in GCC 9.1.1
- anv: Drop unused local variable
- compiler/glsl: Fix warning about unused function
- intel/decoders: Avoid uninitialized variable warnings
- iris: Guard GEN9-only function in Iris state to avoid warning
- tgsi: Remove unused local
- i965: Silence brw_blorp uninitialized warning
- nir/lower_explicit_io: Handle 1 bit loads and stores
- glsl/nir: Avoid overflow when setting max_uniform_location
- mesa/st: Do not rely on name to identify special uniforms
- compiler: Add glsl_contains_opaque() helper
- mesa: Pack gl_program_parameter struct
- glsl/nir: Fill in the Parameters in NIR linker
- mesa: Fill Parameter storage indices even when not using SPIR-V
- mesa/program: Associate uniform storage without using names
- mesa/st: Lookup parameters without using names
- mesa/st: Extract preprocessing NIR steps
- mesa/st: Add support for SPIR-V shaders
- mesa/st: Don't expect prog->nir to already exist

- mesa/spirv: Set a few more extensions
- gallium: Add ARB_gl_spirv support
- glsl/nir: Add and use a gl_nir_link() function
- iris: Enable ARB_gl_spirv and ARB_spirv_extensions
- mesa/st: Fallback to name lookup when the variable have no Parameter
- spirv: Update JSON and headers to 1.5
- spirv: Handle ShaderLayer and ShaderViewportIndex capabilities
- spirv: Add missing break for capability handling
- intel/fs: Add Fall-through comment
- mesa: Extension boilerplate for EXT_demote_to_helper_invocation
- glsl: Add ir_demote
- glsl: Parse ‘demote‘ statement
- glsl: Add helperInvocationEXT() builtin
- gallium: Add PIPE_CAP_DEMOTE_TO_HELPER_INVOCATION
- iris: Enable EXT_demote_to_helper_invocation
- i965: Enable EXT_demote_to_helper_invocation
- docs/relnotes: Add EXT_demote_to_helper_invocation support on iris, i965
- docs: Fix GL_EXT_demote_to_helper_invocation name
- vulkan: Update the XML and headers to 1.1.124
- spirv: Implement SPV_KHR_shader_clock
- anv: Implement VK_KHR_shader_clock
- anv: Enable VK_EXT_shader_subgroup_{ballot,vote}
- docs: Update recently enabled VK extensions on Intel
- intel: Add INTEL_DEBUG=nofc for disabling fast clears
- anv: Disable fast clears when running with INTEL_DEBUG=nofc
- iris: Disable fast clears when running with INTEL_DEBUG=nofc
- i965: Disable fast clears when running with INTEL_DEBUG=nofc
- vulkan: Update the XML and headers to 1.1.125
- anv: Advertise VK_KHR_spirv_1_4
- intel/fs/gen12: Add tests for scoreboard pass
- nir: Add scoped_memory_barrier intrinsic
- nir/tests: Add copy propagation tests with scoped_memory_barrier
- intel/fs: Implement scoped_memory_barrier
- spirv: Parse memory semantics for atomic operations
- spirv: Emit memory barriers for atomic operations
- spirv: Add SpvMemoryModelVulkan and related capabilities

- spirv: Add option to emit scoped memory barriers
- spirv: Handle MakeTexelAvailable/Visible
- spirv: Handle MakePointerAvailable/Visible
- anv: Implement VK_KHR_vulkan_memory_model
- spirv: Add imageoperands_to_string helper
- spirv: Check that only one offset is defined as Image Operand
- spirv: Add helper to find args of Image Operands
- anv: Fix output of INTEL_DEBUG=bat for chained batches
- spirv: Don't fail if multiple ordering semantics bits are set
- spirv: Don't leak GS initialization to other stages
- anv: Initialize depth_bounds_test_enable when not explicitly set

Chris Wilson (2):

- iris: Allow packed RGB pbo uploads
- st/mesa: Map MESA_FORMAT_RGB_UNORM8 <-> PIPE_FORMAT_R8G8B8_UNORM

Christian Gmeiner (13):

- gallium: util_set_vertex_buffers_mask(..): make use of u_bit_consecutive(..)
- etnaviv: a bit of micro-optimization
- Revert "gallium: remove PIPE_CAP_TEXTURE_SHADOW_MAP"
- etnaviv: disable ARB_shadow
- etnaviv: etna_resource_copy_region(..): drop assert
- etnaviv: support ARB_framebuffer_object
- etnaviv: nir: start to make use of compile_error(..)
- etnaviv: output the same shader-db format as freedreno, v3d and intel
- etnaviv: fix compile warnings
- etnaviv: fix code style
- etnaviv: store updated usage in pipe_transfer object
- etnaviv: keep track of buffer valid ranges for PIPE_BUFFER
- etnaviv: remove dead code

Clément Guérin (1):

- radeonsi: enable zerovram for Rocket League

Connor Abbott (40):

- st/nir: Fix num_inputs for VS inputs
- radeonsi/nir: Don't recompute num_inputs and num_outputs
- ac/nir: Handle const array offsets in get_deref_offset()
- ac/nir: Assert GS input index is constant
- radeonsi/nir: Don't add const offset to indirect

- radeonsi/nir: Add const_index when loading GS inputs
- radeonsi/nir: Rewrite store intrinsic gathering
- radeonsi/nir: Rewrite output scanning
- ac/nir: add a workaround for viewing a slice of 3D as a 2D image
- ac/nir: Remove gfx9_stride_size_workaround_for_atomic
- ac/nir: Rewrite gather4 integer workaround based on radeonsi
- ac/nir: Fix gather4 integer wa with unnormalized coordinates
- nir: Fix num_ssbos when lowering atomic counters
- ttn: Fill out more info fields
- radeonsi/nir: Remove uniform variable scanning
- radv/radeonsi: Don't count read-only data when reporting code size
- ac/nir: Support load_constant intrinsics
- ac/nir: Enable nir_opt_large_constants
- st/nir: Call nir_remove_unused_variables() in the opt loop
- st/nir: Don't lower indirects when linking
- gallium: Plumb through a way to disable GLSL const lowering
- radeonsi/nir: Don't lower constant arrays to uniforms
- radv: Call nir_propagate_invariant()
- lima/gpir: Do all lowerings before rsched
- lima/gpir: Ignore unscheduled successors in can_use_complex()
- lima/gpir: Fix schedule_first insertion logic
- lima/gpir: Fix fake dep handling for schedule_first nodes
- lima/gpir: Disallow moves for schedule_first nodes
- nir/opt_if: Fix undef handling in opt_split_alu_of_phi()
- lima/gpir: Fix compiler warning
- lima/gpir: Only try to place actual children
- lima/gpir: Support branch instructions
- lima/gpir: Use registers for values live in multiple blocks
- lima/gpir: Fix postlog2 fixup handling
- lima/gpir: Don't emit movs when translating from NIR
- lima/gpir: Fix 64-bit shift in scheduler spilling
- nir/opt_large_constants: Handle store writemasks
- nir: Fix overlapping vars in nir_assign_io_var_locations()
- nir/sink: Rewrite loop handling logic
- nir/sink: Don't sink load_ubo to outside of its defining loop

Daniel Kolesa (1):

- util: add auxv based PowerPC AltiVec/VSX detection

Daniel Schürmann (44):

- nir/algebraic: some subtraction optimizations
- aco: Initial commit of independent AMD compiler
- radv/aco: Setup alternate path in RADV to support the experimental ACO compiler
- radv: enable clustered reductions
- radv/aco: enable VK_EXT_shader_demote_to_helper_invocation
- radv: remove dead shared variables
- aco: only emit waitcnt on loop continues if we there was some load or export
- freedreno: Enable the nir_opt_algebraic_late() pass.
- nir: recombine nir_op_*sub when lower_sub = false
- nir: Remove unnecessary subtraction optimizations
- radv/aco: Don't lower subtractions
- aco: call nir_opt_algebraic_late() exhaustively
- nouveau: set lower_sub = true
- aco: re-use existing phi instruction when lowering boolean phis
- aco: don't reorder instructions in order to lower boolean phis
- aco: don't combine minmax3 if there is a neg or abs modifier in between
- aco: ensure that uniform booleans are computed in WQM if their uses happen in WQM
- aco: refactor value numbering
- aco: restrict scheduling depending on max_waves
- aco: only skip RAR dependencies if the variable is killed somewhere
- aco: add can_reorder flags to load_ubo and load_constant
- aco: don't schedule instructions through depending VMEM instructions
- aco: Lower to CSSA
- aco: improve live variable analysis
- aco: remove potential critical edge on loops.
- aco: fix live-range splits of phis
- aco: fix transitive affinities of spilled variables
- aco: don't insert the exec mask into set of live-out variables when spilling
- aco: consider loop_exit blocks like merge blocks, even if they have only one predecessor
- aco: don't add interferences between spilled phi operands
- aco: simplify calculation of target register pressure when spilling
- aco: ensure that spilled VGPR reloads are done after p_logical_start
- aco: omit linear VGPRs as spill variables
- aco: always set scratch_offset in startpgm

- aco: implement VGPR spilling
- docs/relnotes/new_features.txt: Add note about ACO
- aco: fix immediate offset for spills if scratch is used
- aco: only use single-dword loads/stores for spilling
- aco: fix accidential reordering of instructions when scheduling
- aco: workaround Tonga/Iceland hardware bug
- aco: fix invalid access on Pseudo_instructions
- aco: preserve kill flag on moved operands during RA
- aco: don't split live-ranges of linear VGPRs
- aco: fix a couple of value numbering issues

Daniel Stone (1):

- panfrost: Respect offset for imported resources

Danilo Spinella (1):

- egl: Include stddef.h in generated source

Danylo Piliaiev (10):

- nir/loop_unroll: Update the comments for loop_prepare_for_unroll
- nir/loop_unroll: Prepare loop for unrolling in wrapper_unroll
- nir/loop_analyze: Treat do{}while(false) loops as 0 iterations
- glsl: Fix unroll of do{} while(false) like loops
- tgsi_to_nir: Translate TGSI_INTERPOLATE_COLOR as INTERP_MODE_NONE
- iris: Fix fence leak in iris_fence_flush
- st/nine: Ignore D3DSIO_RET if it is the last instruction in a shader
- intel/compiler: Fix C++ one definition rule violations
- glsl: Initialize all fields of ir_variable in constructor
- i965: Unify CC_STATE and BLEND_STATE atoms on Haswell as a workaround

Dave Airlie (75):

- virgl: drop unused format field
- virgl: fix format conversion for recent gallium changes.
- gallivm: fix atomic compare-and-swap
- llvmpipe: refactor jit type creation
- gallivm: make lp_build_float_to_r11g11b10 take a const src
- gallivm: handle helper invocation (v2)
- gallivm: move first/last level jit texture members.
- llvmpipe: handle early test property.
- gallivm: add a basic image limit
- llvmpipe: move the fragment shader variant key to dynamic length.

- draw: add jit image type for vs/gs images.
- llvmpipe: introduce image jit type to fragment shader jit.
- gallivm/tgsi: add image interface to tgsi builder
- gallivm: add image load/store/atomic support
- draw: add vs/gs images support
- llvmpipe: add fragment shader image support
- llvmpipe: bind vertex/geometry shader images
- gallivm: add support for fences api on older llvm
- gallivm: add memory barrier support
- llvmpipe: flush on api memorybarrier.
- llvmpipe: enable ARB_shader_image_load_store
- docs: add shader image extensions for llvmpipe
- gallivm: fix appveyor build after images changes
- gallivm: disable accurate cube corner for integer textures.
- llvmpipe: enable fb no attach
- gallivm/flow: add counter reset for loops
- gallivm: add coroutine support files to gallivm.
- gallivm: add coroutine pass manager support
- llvmpipe: reogranise jit pointer ordering
- gallivm: add new compute related intrinsics
- gallivm: add support for compute shared memory
- llvmpipe: add compute threadpool + mutex
- gallivm: add barrier support for compute shaders.
- llvmpipe: introduce compute shader context
- llvmpipe: add initial compute state structs
- gallivm: add compute jit interface.
- llvmpipe: add compute debug option
- llvmpipe: add initial shader create/bind/destroy variants framework.
- llvmpipe: introduce new state dirty tracking for compute.
- llvmpipe: introduce variant building infrastrucutre.
- llvmpipe: add compute shader generation.
- llvmpipe: add grid launch
- llvmpipe: add compute pipeline statistics support.
- llvmpipe: add support for compute constant buffers.
- llvmpipe: add compute sampler + sampler view support.
- llvmpipe: add ssbo support to compute shaders

- llvmpipe: add compute shader images support
- llvmpipe: add compute shader parameter fetching support
- llvmpipe: add local memory allocation path
- llvmpipe: enable compute shaders if LLVM has coroutines
- docs: add llvmpipe features for fb_no_attach and compute shaders
- st/mesa: Prefer R8 for bitmap textures
- st/mesa: fix R8 bitmap texture for TGSI paths.
- llvmpipe: make texture buffer offset alignment == 16
- llvmpipe/draw: fix image sizes for vertex/geometry shaders.
- llvmpipe/draw: handle UBOs that are < 16 bytes.
- gallivm/sample: add gather component selection to the key.
- gallium: add a a new cap for changing the TGSI TG4 instruction encoding
- st/glsl: add support for alternate TG4 encoding.
- llvmpipe: add support for tg4 component selection.
- gallivm: fix coroutines on aarch64 with llvm 8
- gallivm/draw/swr: make the gs_iface not depend on tgsi.
- nir: add a pass to lower flat shading.
- gallium: add flatshade lowering capability
- st/mesa: handling lower flatshading for NIR drivers.
- llvmpipe: handle compute shader launch with 0 threads
- zink: ask for flatshade lowering
- zink: add dri loader
- zink: query support (v2)
- zink/spirv: store all values as uint.
- zink: add support for compressed formats
- zink: add sample mask support
- zink: add samples to rasterizer
- zink: attempt to get multisample resource creation right
- llvmpipe/ppc: fix if/ifdef confusion in backport.

Dave Stevenson (1):

- broadcom/v3d: Allow importing linear BOs with arbitrary offset/stride.

Duncan Hopkins (7):

- zink: clamped limits to INT_MAX when stored as uint32_t.
- zink: fix line-width calculation
- zink: respect ubo buffer alignment requirement
- zink: limited uniform buffer size so the limits is not exceeded.

- zink: pass line width from rast_state to gfx_pipeline_state.
- zink: Use optimal layout instead of general. Reduces valid layer warnings. Fixes RADV image noise.
- zink: make sure src image is transfer-src-optimal

Dylan Baker (120):

- docs: Mark 19.2.0-rc2 as done and push back rc3 and rc4/final
- glsl/tests: Handle windows \rn new lines
- meson: don't try to generate i18n translations on windows
- meson: Make shared-glapi a combo
- meson: don't build glapi_static_check_table on windows
- add a git ignore for subprojects
- meson: add a zlib subproject
- meson: add a expat subproject
- glapi: export glapi_destroy_multithread when building shared-glapi on windows
- meson: fix dl detection on non cygwin windows
- meson: build getopt when using msvc
- meson: Add a platform for windows
- meson: don't build glx or dri by default on windows
- meson: don't allow glvnd on windows
- meson: don't generate file into subdirs
- Docs: mark that 19.2.0-rc3 has been released
- scons: Make scons and meson agree about path to glapi generated headers
- docs: Add release notes for 19.2.0
- docs: add SHA256 sum for 19.2.0
- docs: update calendar, add news item, and link release notes for 19.2.0
- release: Push 19.3 back two weeks
- bin/get-pick-list: use –oneline=pretty instead of –oneline
- meson: fix logic for generating .pc files with old glvnd
- meson: Try finding libxvmcw via pkg-config before using find_library
- meson: Link xvmc with libxv
- meson: gallium media state trackers require libdrm with x11
- docs: update install docs for meson
- docs: use https for mesonbuild.com
- docs: remove stray newline
- meson: remove -DGALLIUM_SOFTPIPE from st/osmesa
- docs: Add use of Closes: tag for closing gitlab issues
- docs: add a new_features.text file and remove 19.3.0 release notes

- scripts: Add a gen_release_notes.py script
- release: Add an update_release_calendar.py script
- bin: delete unused releasing scripts
- meson: Only error building gallium video without libdrm when the platform is drm
- docs: Add relnotes for 19.2.1
- docs: Add SHA256 sum for 19.2.1
- docs: update calendar, add news item, and link release notes for 19.2.1
- util: use _WIN32 instead of WIN32
- meson: add windows compiler checks and libraries
- meson: Add windows defines to glapi
- meson: Add necessary defines for mesa_gallium on windows
- meson: build gallium gdi winsys
- meson: build wgl state tracker
- meson: build libgl-gdi target
- meson: build graw-gdi target
- meson: fix gallium-osmesa to build for windows
- meson: Don't check for posix_memalign on windows
- util/xmlconfig: include strndup.h for windows
- meson: fix pipe-loader compilation for windows
- meson: don't look for rt on windows
- meson: Add support for using win_flex and win_bison on windows
- meson: force inclusion of inttypes.h for glcpp with msvc
- meson: disable sse4.1 optimizations with msvc
- meson: add switches for SWR with MSVC
- meson: don't define USE_ELF_TLS for windows
- meson: Add idep_getopt for tests
- meson: Add msvc compat args to util/tests
- meson: Set visibility and compat args for graw
- meson: don't build gallium trivial tests on windows
- meson: disable graw tests on mingw
- meson: don't build or run mesa-sha1 test on windows
- meson: maintain names of shared API libraries
- meson: add msvc compat args to swr
- meson: don't error on formaters with mingw
- meson: only build timspec test if timespec is available
- meson: glcpp tests are expected to fail on windows

- meson/util: Don't run string_buffer tests on mingw
- glsl/tests: Handle no-exec errors
- docs: update meson docs for windows
- appveyor: Add support for meson as well as scons on windows
- gitlab-ci: Add a mingw x86_64 job
- meson: Don't use expat on windows
- gitlab-ci: Add a pkg-config for mingw
- Revert "gitlab-ci: Disable meson-mingw32-x86_64 job again for now"
- gitlab-ci: Set the meson wrapmode to disabled
- appveyor: Cache meson's wrap downloads
- meson/llvmpipe: Add dep_llvm to driver_swrast
- meson: Add support for wrapping llvm
- meson: Use cmake to find LLVM when building for windows
- docs: update meson docs for windows
- appveyor: Add support for building llvmpipe with meson
- appveyor: Move appveyor script into .appveyor directory
- docs: Add new feature for compiling for windows with meson
- meson: Require meson >= 0.49.1 when using icc or icl
- scons: Use print_function ins SConstruct
- scons: Print a deprecation warning about using scons on not windows
- scons: Also print a deprecation warning on windows
- docs: Add release not about scons deprecation
- docs: Add release notes for 19.2.2
- docs: Add sha256 sum for 19.2.2
- docs: update calendar, add news item and link release notes for 19.2.2
- bin/gen_release_notes.py: fix conditional of bugfix
- bin/gen_release_notes.py: strip '#' from gitlab bugs
- bin/gen_release_notes.py: Return "None" if there are no new features
- bin/post_version.py: Pass version as an argument
- bin/post_version.py: white space fixes
- bin/post_release.py: Add .html to hrefs
- bin/gen_release_notes.py: html escape all external data
- bin/gen_release_notes.py: Add a warning if new features are introduced in a point release
- docs: update releasing process to use new scripts and gitlab
- nir: Fix invalid code for MSVC
- gitlab-ci: refactor out some common stuff for Windows and Linux

- gitlab-ci: Add a job for meson on windows
- VERSION: bump to rc1
- nir: correct use of identity check in python
- meson: Add dep_glvnd to egl deps when building with glvnd
- Bump VERSION to 19.3.0-rc2
- cherry-ignore: Update for 19.3-rc3 cycle
- Bump version for -rc3
- cherry-ignore: update for 19.3.0-rc4 cycle
- VERSION: bump for 19.3.0-rc4
- VERSION: Bump version for -rc5
- VERSION: bump version for 19.3-rc6
- cherry-ignore: update for 19.3-rc7
- meson/broadcom: libbroadcom_cle needs expat headers
- meson/broadcom: libbroadcom_cle also needs zlib
- Revert "egl: avoid local modifications for eglext.h Khronos standard header file"
- Revert "egl: move #include of local headers out of Khronos headers"

Eduardo Lima Mitev (4):

- nir: Add new texop nir_texop_tex_prefetch
- freedreno/ir3: Add a NIR pass to select tex instructions eligible for pre-fetch
- nir: Add a new ALU nir_op_imad24_ir3
- freedreno/ir3: Handle newly added opcode nir_op_imad24_ir3

Emil Velikov (3):

- mesa: bump version to 19.3.0-devel
- docs: add 19.3.0-devel release notes template
- docs: update calendar for 19.2.x

Eric Anholt (57):

- gallium: Add a block depth field to the u_formats table.
- gallium: Add block depth to the format utils.
- gallium: Add the ASTC 3D formats.
- gallium: Fix mesa format name in unit test failure path.
- gallium: Skip generating the pack/unpack union if we don't use it.
- gallium: Drop the useless union wrapper on pack/unpack.
- gallium: Drop a bit of dead code from the pack/unpack python.
- gallium: Fix big-endian addressing of non-bitmask array formats.
- gallium: Don't emit identical endian-dependent pack/unpack code.
- freedreno/a6xx: Fix non-mipmap filtering selection.

- freedreno: Fix the type of single-component scaled vertex attrs.
- gallium/osmesa: Introduce a test.
- gallium/osmesa: Fix a race in creating the stmgr.
- gallium/osmesa: Move 565 format selection checks where the rest are.
- uapi: Update drm_fourcc.h
- dri: Use DRM_FORMAT_* instead of defining our own copy.
- gitlab-ci: Disable dEQP's watchdog timer.
- gitlab-ci: Log the driver version that got tested.
- freedreno: Introduce gitlab-based CI.
- gitlab-ci/a630: Disable flappy layout_binding.ssbo.fragment_binding_array
- egl/android: Fix build since the DRI fourcc removal.
- gitlab-ci/a630: Drop remaining dEQP-GLES3.functional.draw.random.* xfails.
- gitlab-ci/a630: Drop the MSAA expected failure.
- gitlab-ci: Make the test job fail when bugs are unexpectedly fixed.
- freedreno: Fix invalid read when a block has no instructions.
- freedreno/a3xx: Mostly fix min-vs-mag filtering decisions on non-mipmap tex.
- shader_enums: Move MAX_DRAW_BUFFERS to this file.
- turnip: Add a .editorconfig and .dir-locals.el
- turnip: Silence compiler warning about uninit pipeline.
- turnip: Fix failure behavior of vkCreateGraphicsPipelines.
- vc4: Enable the nir_opt_algebraic_late() pass.
- v3d: Enable the late algebraic optimizations to get real subs.
- nir: Make nir_search's dumping go to stderr.
- nir: Skip emitting no-op movs from the builder.
- nir: Keep the range analysis HT around intra-pass until we make a change.
- nir: Factor out most of the algebraic passes C code to .c/.h.
- nir: Fix some wonky whitespace in nir_search.h.
- turnip: Drop unused tu_pack_clear_value() return.
- turnip: Fill in clear color packing for r10g11b11 and rgb9e5.
- turnip: Tell spirv_to_nir that we want fragcoord as a sysval.
- turnip: Set up the correct tiling mode for small attachments.
- turnip: Emit clears of gmem using linear.
- freedreno/ci: Ban texsubimage2d_pbo.r16ui_2d, due to two flakes reported.
- mesa: Add debug info to _mesa_format_from_format_and_type() error path.
- mesa: Fix depth/stencil ordering in _mesa_format_from_format_and_type().
- mesa: Add format/type matching for DEPTH/UINT_24_8.

- mesa: Add support for array formats of depth and stencil.
- mesa: Refactor the entirety of _mesa_format_matches_format_and_type().
- v3d: Add Compute Shader support
- r100/r200: factor out txformat/txfilter setup from the TFP path.
- radeon: Fill in the TXOFFSET field containing the tile bits in our relocs.
- radeon: Drop the unused first arg of OUT_BATCH_RELOC.
- mesa: Replace the LA16_UNORM packed formats with one array format.
- mesa: Replace MESA_FORMAT_L8A8/A8L8 UNORM/SNORM/SRGB with an array format.
- gallium: Drop the unused PIPE_FORMAT_A*L* formats.
- mesa: Redefine the RG formats as array formats.
- ci: Disable lima until its farm can get fixed.

Eric Engestrom (104):

- scons: define MESA_LLVM_VERSION_STRING like the other build systems do
- llvmpipe: use LLVM version string instead of re-computing it
- swr: use LLVM version string instead of re-computing it
- scons: add support for MAJOR_IN_{MKDEV,SYSMACROS}
- egl: warn user if they set an invalid EGL_PLATFORM
- ttn: fix 64-bit shift on 32-bit ‘1‘
- egl: fix deadlock in malloc error path
- util/os_file: fix double-close()
- anv: fix format string in error message
- freedreno/drm-shim: fix mem leak
- nir: fix memleak in error path
- gallivm: replace ‘0x‘ version print with actual version string
- meson/scons/android: add LLVM_AVAILABLE binary flag
- aux/draw: replace binary HAVE_LLVM checks with LLVM_AVAILABLE
- r600: replace binary HAVE_LLVM checks with LLVM_AVAILABLE
- svga: replace binary HAVE_LLVM checks with LLVM_AVAILABLE
- amd: replace major llvm version checks with LLVM_VERSION_MAJOR
- swr: replace major llvm version checks with LLVM_VERSION_MAJOR
- gallivm: replace major llvm version checks with LLVM_VERSION_MAJOR
- clover: replace major llvm version checks with LLVM_VERSION_MAJOR
- gallivm: replace more complex 3.x version check with LLVM_VERSION_MAJOR/MINOR
- clover: replace more complex 3.x version check with LLVM_VERSION_MAJOR/MINOR
- llvmpipe: replace more complex 3.x version check with LLVM_VERSION_MAJOR/MINOR
- meson/scons/android: drop now-unused HAVE_LLVM

- gallivm: drop LLVM<3.3 code paths as no build system allows that
- anv: add support for driconf
- wsi: add minImageCount override
- anv: add support for vk_x11_override_min_image_count
- amd: move adaptive sync to performance section, as it is defined in xmlpool
- radv: add support for vk_x11_override_min_image_count
- drirc: override minImageCount=2 for gfxbench
- meson/iris: replace partial list of nir dep files with idep_nir_headers
- meson/v3d: replace partial list of nir dep files with idep_nir_headers
- gitlab-ci: rename stages to something simpler
- gl: drop incorrect pkg-config file for glvnd
- anv: split instance dispatch table
- anv: implement ICD interface v4
- meson: split compiler warnings one per line
- radv: fix s/load/store/ copy-paste typo
- meson: drop -Wno-foo bug workaround for Meson < 0.46
- meson: split more compiler options to their own line
- meson: re-add incorrect pkg-config files with GLVND for backward compatibility
- docs/release-calendar: fix bugfix release numbers
- docs/release-calendar: add missing <td> and </td>
- glsl: turn runtime asserts of compile-time value into compile-time asserts
- etnaviv: fix bitmask typo
- docs/install: drop autotools references
- git: delete .gitattributes
- egl: replace MESA_EGL_NO_X11_HEADERS hack with upstream EGL_NO_X11
- loader: replace int/1/0 with bool/true/false
- loader: s/int/bool/ for predicate result
- loader: use ARRAY_SIZE instead of NULL sentinel
- meson/loader: drop unneeded *.h file
- script: drop get_reviewer.pl
- meson: add missing idep_nir_headers in iris_gen_libs
- meson: use idep_nir instead of libnir in libnouveau
- meson: use idep_nir instead of libnir in libclnir
- meson: use idep_nir instead of libnir in gallium nine
- meson: use idep_nir instead of libnir in haiku softpipe
- meson: use idep_nir instead of libnir in pipe-loader

- meson: rename libnir to _libnir to make it clear it's not meant to be used anywhere else
- meson: drop duplicate inc_nir from libiris
- meson: drop duplicate inc_nir from libglsl
- meson: drop duplicate inc_nir from spirv2nir
- meson: drop unused inc_nir
- include: update drm-uapi
- meson: fix sys/mkdev.h detection on Solaris
- GL: drop symbols mangling support
- meson: rename 'glvnd_missing_pc_files' to 'not glvnd_has_headers_and_pc_files'
- meson: move a couple of include installs around
- meson: split headers one per line
- meson: split Mesa headers as a separate installation
- meson: skip installation of GLVND-provided headers
- symbols-check: ignore exported C++ symbols
- anv: add exported symbols check
- radv: add exported symbols check
- gbm: turn 0/-1 bool into true/false
- gbm: replace 1/0 bool with true/false
- gbm: replace NULL sentinel with explicit ARRAY_SIZE()
- gbm: use size_t for array indexes
- gitlab-ci: set a common job parent for container stage
- gitlab-ci: set a common job parent for build stage
- gitlab-ci: set a common job parent for test stage
- mesa/math: delete leftover... from 18 years ago (!)
- mesa/math: delete duplicate extern symbol
- util/u_atomic: fix return type of p_atomic_{inc,dec}_return() and p_atomic_{cmp,}xchg()
- travis: don't (re)install python
- travis: test meson install as well
- osmesa: add missing #include <stdint.h>
- llvmpipe: avoid compiling no-op block on release builds
- llvmpipe: avoid generating empty-body blocks
- meson: add -Werror=empty-body to disallow 'if(x);'
- anv: fix error message
- anv: fix empty-body instruction
- radv: fix empty-body instruction
- v3d: fix empty-body instruction

- tu: fix empty-body instruction
- anv: add a couple printflike() annotations
- loader: default to iris for all future PCI IDs
- travis: fix scons build after deprecation warning
- meson: define _GNU_SOURCE on FreeBSD
- egl: fix _EGL_NATIVE_PLATFORM fallback
- egl: move #include of local headers out of Khronos headers
- vulkan: delete typo'd header

Erico Nunes (7):

- lima: fix ppir spill stack allocation
- lima/ppir: lower selects to scalars
- lima/ppir: enable vectorize optimization
- lima/ppir: mark regalloc created ssa unspillable
- lima/ppir: optimizations in regalloc spilling code
- lima/ppir: improve regalloc spill cost calculation
- lima: remove partial clear support from pipe->clear()

Erik Faye-Lund (210):

- gallium/auxiliary/indices: consistently apply start only to input
- mesa/main: remove unused include
- util: fix SSE-version needed for double opcodes
- util: do not assume MSVC implies SSE
- mesa/x86: improve SSE-checks for MSVC
- util: only allow _BitScanReverse64 on 64-bit cpus
- gallium/gdi: use GALLIUM_FOO rather than HAVE_FOO
- st/mesa: remove always-true expression
- .mailmap: add an alias for Michel Dänzer
- .mailmap: add an alias for Eric Engestrom
- .mailmap: add an alias for Bas Nieuwenhuizen
- .mailmap: add an alias for Frank Binns
- glsl: correct bitcast-helpers
- loader/dri3: do not blit outside old/new buffers
- .mailmap: specify spelling for Elie Tournier
- .mailmap: add an alias for Alexandros Frantzis
- .mailmap: add an alias for Gert Wollny
- .mailmap: add an alias for Tomeu Vizoso
- .mailmap: add a couple of aliases for Jakob Bornecrantz

- nir: initialize uses_discard to false
- nir: initialize needs_helper_invocations as well
- mesa/main: prefer R8-textures instead of A8 for glBitmap in display lists
- gallium/u_blitter: set a more sane viewport-state
- mesa: expose alpha-ref as a state-variable
- nir: allow passing alpha-ref state to lowering-code
- mesa/gallium: automatically lower alpha-testing
- st/mesa: move point_size_per_vertex-logic to helper
- nir: add lowering-pass for point-size mov
- mesa/gallium: automatically lower point-size
- nir: support derefs in two-sided lighting lowering
- mesa/gallium: automatically lower two-sided lighting
- nir: support lowering clipdist to arrays
- nir: support feeding state to nir_lower_clip_[vg]s
- mesa/program: support referencing the clip-space clip-plane state
- mesa/st: support lowering user-clip-planes automatically
- panfrost: do not report alpha-test as supported
- vc4: do not report alpha-test as supported
- v3d: do not report alpha-test as supported
- nir: drop support for using load_alpha_ref_float
- nir: drop unused alpha_ref_float
- mesa/st: assert that lowering is supported
- Revert "nir: drop unused alpha_ref_float"
- Revert "nir: drop support for using load_alpha_ref_float"
- Revert "v3d: do not report alpha-test as supported"
- Revert "vc4: do not report alpha-test as supported"
- zink: introduce opengl over vulkan
- zink: detect presence of VK_KHR_maintenance1
- zink/spirv: implement point-sprites
- zink: transform z-range
- zink: remove discard_if
- zink/spirv: implement some integer ops
- zink/spirv: handle reading registers
- zink/spirv: prepare for control-flow
- zink/spirv: implement if-statements
- zink/spirv: implement discard

- zink/spirv: implement loops
- zink: prepare for caching of renderpases/framebuffers
- zink: move render-pass begin to helper
- zink: do not leak image-views
- zink: move cmdbuf-resetting into a helper
- zink: prepare for multiple cmdbufs
- zink: pass zink_render_pass to pipeline-creation
- zink: cache programs
- zink: move renderpass inside gfx pipeline state
- zink: cache those pipelines
- zink: reference renderpass and framebuffer from cmdbuf
- zink: return old fence from zink_flush
- zink: reference vertex and index buffers
- zink: reference ubos and textures
- zink: wait for idle on context-destroy
- zink: whitespace cleanup
- zink: reference blit/copy-region resources
- zink: add curr_cmdbuf-helper
- zink: delete samplers after the current cmdbuf
- zink: texture-rects?
- zink: store shader_info in zink_shader
- zink: implement fmod
- zink: track used resources
- zink: do not destroy staging-resource, deref it
- zink: use uvec for undefs
- zink: emit dedicated block for variables
- zink: ensure non-fragment shaders use lod-versions of texture
- zink: ensure textures are transitioned properly
- zink: assign increasing locations to varyings
- zink: move primitive-topology stuff into program
- zink: tweak state handling
- zink: remove unusual alignment
- zink: return after blitting
- zink: implement batching
- zink: simplify renderpass/framebuffer logic a tad
- zink: cache render-passes

- zink: cache framebuffers
- zink: more batch-ism
- zink: use helper
- zink: fixup parameter name
- zink: ensure sampler-views survive a batch
- zink: remove hack-comment
- zink: clean up render-pass management
- zink: rename sampler-view destroy function
- zink: pass screen instead of device to program-functions
- zink: keep a reference to used render-passes
- zink: prepare for shadow-samplers
- zink: kill dead code
- zink: clamp scissors
- zink: do not use hash-table for regs
- zink: squashme: forward declare hash_table
- zink: squashme: trade cplusplus wrapper for header-guard
- zink: fix off-by-one in assert
- zink: reuse constants
- zink: pool descriptors per batch
- zink: request alpha-test lowering
- zink/spirv: var -> regs
- zink/spirv: rename vec_type
- zink: do not lower io
- zink: request ucp-lowering
- zink: cleanup zink_end_batch
- zink: drop unused argument
- zink: refactor fence destruction
- zink: only consider format-desc if checking details
- zink: document end-of-frame hack
- zink: use pipe_stencil_ref instead of uint32_t-array
- zink: store sampler and image_view counts
- zink: save original scissor and viewport
- zink: save all supported util_blitter states
- zink: process one aspect-mask bit at the time
- zink: clean up opcode-emitting a bit
- zink: add some opcodes

- zink: add division ops
- zink: add shift ops
- zink: implement ineg
- zink: more comparison-ops
- zink: more converts
- zink: add more compares
- zink: crash hard on unknown queries
- zink: abort on submit-failure
- zink: stub resource_from_handle
- zink: make sure imageExtent.depth is 1 for arrays
- zink/spirv: correct opcode
- zink: support more texturing
- zink: wait for transfer when reading
- zink/spirv: be a bit more strict with fragment-results
- zink/spirv: debug-print unknown varying slots
- zink: ensure layout is reasonable before copying
- zink: fixup: save rasterizer
- zink: set ExecutionModeDepthReplacing when depth is written
- zink: avoid texelFetch until it's implemented
- zink: remove insecure comment
- zink: don't crash when setting rast-state to NULL
- zink: add note about enabling PIPE_CAP_CLIP_HALFZ
- zink/spirv: always enable Sampled1D for fragment shaders
- zink: do not use both depth and stencil aspects for sampler-views
- zink/spirv: support vec1 coordinates
- zink: fixup boolean queries
- zink: disable timestamp-queries
- zink: move set_active_query_state-stub to zink_query.c
- HACK: zink: suspend / resume queries on batch-boundaries
- zink: also accept txl
- zink: use primconvert to get rid of 8-bit indices
- zink: initialize nr_samples for pipe_surface
- zink: fix rendering to 3D-textures
- zink: support shadow-samplers
- zink: disable PIPE_CAP_QUERY_TIME_ELAPSED for now
- zink: add missing sRGB DXT-formats

- zink: lower point-size
- zink/spirv: use ordered compares
- zink/spirv: implement f2b1
- zink/spirv: assert bit-size
- zink/spirv: implement bcsel
- zink/spirv: implement bitwise ops
- zink/spirv: implement b2i32
- zink/spirv: implement emit_select helper
- zink/spirv: implement emit_float_const helper
- zink/spirv: use bit_size instead of hard-coding
- zink/spirv: add emit_bitcast-helper
- zink/spirv: add emit_uint_const-helper
- zink/spirv: inline get_uvec_constant into emit_load_const
- zink/spirv: clean up get_[fu]vec_constant
- zink/spirv: fixup b2i32 and implement b2f32
- zink/spirv: prepare for 1-bit booleans
- zink: do not lower bools to float
- zink/spirv: fixup b2i32
- zink/spirv: implement load_front_face
- zink/spirv: alias generic varyings on non-generic ones
- zink: lower two-sided coloring
- zink/spirv: alias var0 on tex0 etc instead
- zink: do not set VK_IMAGE_CREATE_2D_ARRAY_COMPATIBLE_BIT for non-3D textures
- zink: use VK_FORMAT_B8G8R8A8_UNORM for PIPE_FORMAT_B8G8R8X8_UNORM
- zink: implement resource_from_handle
- zink: refactor blitting
- zink: fixup return-value
- zink: pass screen to zink_create_gfx_pipeline
- zink: do not set lineWidth to invalid value
- zink: fixup scissoring
- zink/spirv: more complete sampler-dim handling
- zink: simplify gl-to-vulkan lowering
- gitlab-ci: also build Zink on CI
- gitlab-ci: fixup debian tags
- zink: error if VK_KHR_maintenance1 isn't supported
- zink: emulate optional depth-formats

- st/mesa: lower global vars to local after lowering clip
- zink: use dynamic state for line-width
- zink: use bitfield for dirty flagging
- zink: drop nop descriptor-updates
- zink: only enable KHR_external_memory_fd if supported
- zink: emit line-width when using polygon line-mode
- zink: use actual format for render-pass
- zink: always allow mutating the format
- zink: do not advertize coherent mapping
- zink: disable fragment-shader texture-lod
- zink: correct depth-stencil format

Francisco Jerez (56):

- intel/fs: Teach fs_inst::is_send_from_grf() about some missing send-like instructions.
- intel/fs: Define is_payload() method of the IR instruction class.
- intel/fs: Define is_send() convenience IR helper.
- intel/fs: Fix constness of implied_mrf_writes() argument.
- intel/eu: Split brw_inst ex_desc accessors for SEND(C) vs. SENDS(C).
- intel/eu: Fix up various type conversions in brw_eu.c that are illegal C++.
- intel/eu: Rework opcode description tables to allow efficient look-up by either HW or IR opcode.
- intel/eu: Encode and decode native instruction opcodes from/to IR opcodes.
- intel/ir: Drop hard-coded correspondence between IR and HW opcodes.
- intel/ir: Represent physical and logical subsets of the CFG.
- intel/ir: Add helper function to push block onto CFG analysis stack.
- intel/ir: Represent logical edge of BREAK instruction.
- intel/ir: Represent physical edge of ELSE instruction.
- intel/ir: Represent physical edge of unconditional CONTINUE instruction.
- intel/eu/gen12: Extend brw_inst.h macros for Gen12 support.
- intel/eu/gen12: Add sanity-check asserts to brw_inst_bits() and brw_inst_set_bits().
- intel/eu/gen12: Implement basic instruction binary encoding.
- intel/eu/gen12: Implement three-source instruction binary encoding.
- intel/eu/gen12: Implement control flow instruction binary encoding.
- intel/eu/gen12: Implement SEND instruction binary encoding.
- intel/eu/gen12: Implement indirect region binary encoding.
- intel/eu/gen12: Implement compact instruction binary encoding.
- intel/eu/gen12: Implement datatype binary encoding.
- intel/eu/gen11+: Mark dot product opcodes as unsupported on opcode_descs table.

- intel/eu/gen12: Add Gen12 opcode descriptions to the table.
- intel/eu/gen12: Fix codegen of immediate source regions.
- intel/eu/gen12: Codegen three-source instruction source and destination regions.
- intel/eu/gen12: Codegen control flow instructions correctly.
- intel/eu/gen12: Codegen pathological SEND source and destination regions.
- intel/eu/gen12: Codegen SEND descriptor regions correctly.
- intel/eu/gen12: Use SEND instruction for split sends.
- intel/eu/gen12: Don't set DD control, it's gone.
- intel/eu/gen12: Don't set thread control, it's gone.
- intel/ir/gen12: Add SYNC hardware instruction.
- intel/fs/gen12: Add codegen support for the SYNC instruction.
- intel/eu/gen12: Add auxiliary type to represent SWSB information during codegen.
- intel/eu/gen12: Add tracking of default SWSB state to the current brw_codegen instruction.
- intel/eu/gen12: Set SWSB annotations in hand-crafted assembly.
- intel/fs/gen12: Add scheduling information to the IR.
- intel/fs/gen12: Introduce software scoreboard lowering pass.
- intel/fs/gen12: Demodernize software scoreboard lowering pass.
- intel/disasm/gen12: Disassemble software scoreboard information.
- intel/disasm/gen12: Fix disassembly of some common instruction controls.
- intel/disasm/gen12: Disassemble three-source instruction source and destination regions.
- intel/disasm/gen12: Disassemble Gen12 SYNC instruction.
- intel/disasm/gen12: Disassemble Gen12 SEND instructions.
- intel/disasm: Don't disassemble saturate control on SEND instructions.
- intel/disasm: Disassemble register file of split SEND sources.
- intel/fs/gen12: Don't support source mods for 32x16 integer multiply.
- intel/eu/validate/gen12: Implement integer multiply restrictions in EU validator.
- intel/eu/validate/gen12: Fix validation of SYNC instruction.
- intel/eu/validate/gen12: Validation fixes for SEND instruction.
- intel/ir/gen12: Update assert in brw_stage_has_packed_dispatch().
- intel/eu: Don't set notify descriptor field of gateway barrier message.
- intel/fs/gen12: Fix barrier codegen.
- intel/fs/gen11+: Fix CS_OPCODE_CS_TERMINATE codegen.

Fritz Koenig (5):

- include/GLES2: Sync GLES2 headers with Khronos
- mesa: GetFramebufferParameteriv spelling
- mesa: Allow MESA_framebuffer_flip_y for GLES 3

- gallium: Enable MESA_framebuffer_flip_y
- freedreno: reorder format check

Gert Wollny (4):

- radeonsi: Release storage for smda_uploads when the context is destroyed
- etnaviv: enable triangle strips only when the hardware supports it
- r600: Fix interpolateAtCentroid
- r600: Disable eight bit three channel formats

Greg V (1):

- clover: use iterator_range in get_kernel_nodes

Gurchetan Singh (4):

- virgl: remove stride from virgl_hw_res
- virgl: modify resource_create_from_handle(..) callback
- virgl: modify internal structures to track winsys-supplied data
- virgl: honor winsys supplied metadata

Haihao Xiang (1):

- i965: support AYUV/XYUV for external import only

Hal Gentz (11):

- glx: Fix SEGV due to dereferencing a NULL ptr from XCB-GLX.
- clover: Fix build after clang r370122.
- gallium/osmesa: Fix the inability to set no context as current.
- egl: Add EGL_CONFIG_SELECT_GROUP_MESA ext.
- egl: Fixes transparency with EGL and X11.
- egl: Puts RGBA visuals in the second config selection group.
- egl: Configs w/o double buffering support have no 'EGL_WINDOW_BIT'.
- Revert "egl: Configs w/o double buffering support have no 'EGL_WINDOW_BIT'."
- Revert "egl: Puts RGBA visuals in the second config selection group."
- Revert "egl: Fixes transparency with EGL and X11."
- Revert "egl: Add EGL_CONFIG_SELECT_GROUP_MESA ext."

Heinrich Fink (8):

- include: sync GL headers with registry
- specs: Sync framebuffer_flip_y text with GL registry
- headers: remove redundant GL token from GL wrapper
- specs: Add GL_MESA_EGL_sync
- registry: update gl.xml with GL_MESA_EGL_sync token
- headers: Add GL_MESA_EGL_sync token to GL
- egl: Add GL_MESA_EGL_sync support

- mesa/gl: Sync with Khronos registry

Hyunjun Ko (3):

- freedreno/ir3: Add data structures to support texture pre-fetch
- freedreno/ir3: Add support for texture sampling pre-dispatch
- freedreno/ir3: fix printing output registers of FS.

Iago Toral (1):

- v3d: drop unused shader_rec_count member from context

Iago Toral Quiroga (13):

- prog_to_nir: VARYING_SLOT_PSIZ is a scalar
- gallium/ttn: VARYING_SLOT_PSIZ and VARYING_SLOT_FOGC are scalar
- nir/lower_point_size: assume scalar PSIZ
- v3d: add missing line break for performance debug message
- v3d: make sure we have enough space in the CL for the primitive counts packet
- v3d: remove redundant update of queued draw calls
- v3d: fix TF primitive counts for resume without draw
- mesa/main: GL_GEOMETRY_SHADER_INVOCATIONS exists in GL_OES_geometry_shader
- v3d: trivial update to obsolete comment
- v3d: add new flag dirty TMU cache at v3d_compiler
- broadcom: document known hardware issues for L2T flush command
- v3d: request the kernel to flush caches when TMU is dirty
- st/mesa: only require ESSL 3.1 for geometry shaders

Ian Romanick (22):

- nir/algrbraic: Don't optimize open-coded bitfield reverse when lowering is enabled
- intel/compiler: Request bitfield_reverse lowering on pre-Gen7 hardware
- nir/algebraic: Mark some value range analysis-based optimizations imprecise
- nir/algebraic: Clean up value range analysis-based optimizations
- nir/range-analysis: Adjust result range of exp2 to account for flush-to-zero
- nir/range-analysis: Adjust result range of multiplication to account for flush-to-zero
- nir/range-analysis: Fix incorrect fadd range result for (ne_zero, ne_zero)
- nir/range-analysis: Handle constants in nir_op_mov just like nir_op_bcsel
- nir/range-analysis: Range tracking for fpow
- nir/range-analysis: Add a lot more assertions about the contents of tables
- nir/algebraic: Do not apply late DPH optimization in vertex processing stages
- nir/algebraic: Additional D3D Boolean optimization
- nir/range-analysis: Bail if the types don't match
- nir/range-analysis: Use types in the hash key

- nir/range-analysis: Use types to provide better ranges from bcsel and mov
- nir/search: Fix possible NULL dereference in is_fsign
- intel/vec4: Don't try both sources as immediates for DPH
- intel/compiler: Report the number of non-spill/fill SEND messages on vec4 too
- nir/algebraic: Add the ability to mark a replacement as exact
- nir/algebraic: Mark other comparison exact when removing a == a
- intel/fs: Disable conditional discard optimization on Gen4 and Gen5
- intel/compiler: Fix 'comparison is always true' warning

Icenowy Zheng (4):

- lima: reset scissor state if scissor test is disabled
- lima: fix PLBU viewport configuration
- lima: support rectangle texture
- lima: do not set the PP uniforms address lowest bits

Ilia Mirkin (6):

- gallium/vl: use compute preference for all multimedia, not just blit
- teximage: ensure that Tex*SubImage* checks format
- gallium/tgsi: add support for DEMOTE and READ_HELPER opcodes
- nvc0: add support for GL_EXT_demote_to_helper_invocation
- gm107/ir: fix loading z offset for layered 3d image bindings
- nv50/ir: mark STORE destination inputs as used

Illia Iorin (2):

- Revert "mesa/main: Fix multisample texture initialize"
- mesa/main: Ignore filter state for MS texture completeness

Indrajit Das (1):

- radeon/vcn: exclude raven2 from vcn 2.0 encode initialization

James Xiong (5):

- gallium: simplify throttle implementation
- gallium: rename PIPE_CAP_MAX_FRAMES_IN_FLIGHT to PIPE_CAP_THROTTLE
- iris: finish aux import on get_param
- gallium: do not increase ref count of the new throttle fence
- iris: try to set the specified tiling when importing a dmabuf

Jan Beich (6):

- gallium/hud: add CPU usage support for DragonFly/NetBSD/OpenBSD
- util: skip NEON detection if built with -mfpu=neon
- util: detect NEON at runtime on FreeBSD
- util: skip AltiVec detection if built with -maltivec

- util: detect AltiVec at runtime on BSDs
- util: simplify BSD includes

Jan Zielinski (3):

- swr/rasterizer: Enable ARB_fragment_layer_viewport
- swr/rasterizer: Fix GS attributes processing
- gallium/swr: Fix depth values for blit scenario

Jason Ekstrand (57):

- nir: Add explicit signs to image min/max intrinsics
- intel/nir: Add a helper for getting BRW_AOP from an intrinsic
- v3d: Use the correct opcodes for signed image min/max
- intel/fs: Drop the gl_program from fs_visitor
- intel/fs: Fix FB write inst groups
- Revert "intel/fs: Move the scalar-region conversion to the generator."
- anv: Bump maxComputeWorkgroupSize
- intel/tools: Decode 3DSTATE_BINDING_TABLE_POINTERS on SNB
- intel/tools: Decode PS kernels on SNB
- blorp: Memset surface info to zero when initializing it
- intel/blorp: Expose surf_retile_w_to_y internally
- intel/blorp: Expose surf_fake_interleaved_msaa internally
- intel/blorp: Use wide formats for nicely aligned stencil clears
- nir: Handle complex derefs in nir_split_array_vars
- nir: Don't infinitely recurse in lower_ssa_defs_to_regs_block
- nir: Add a block_is_unreachable helper
- nir/repair_ssa: Repair dominance for unreachable blocks
- nir/repair_ssa: Insert deref casts when needed
- nir/dead_cf: Repair SSA if the pass makes progress
- intel/fs: Handle UNDEF in split_virtual_grfs
- vulkan: Update the XML and headers to 1.1.123
- Move blob from compiler/ to util/
- util/rb_tree: Add the unit tests
- util/rb_tree: Reverse the order of comparison functions
- intel/fs: Allow UB, B, and HF types in brw_nir_reduction_op_identity
- intel/fs: Allow CLUSTER_BROADCAST to do type conversion
- intel/fs: Do 8-bit subgroup scan operations in 16 bits
- anv: Advertise VK_KHR_shader_subgroup_extended_types
- nir/repair_ssa: Replace the unreachable check with the phi builder

- util/rb_tree: Replace useless ifs with asserts
- util/rb_tree: Also test _safe iterators
- util/rb_tree: Stop relying on &iter->field != NULL
- intel/fs: Fix fs_inst::flags_read for ANY/ALL predicates
- anv/pipeline: Capture serialized NIR
- intel/eu/validate/gen12: Don't blow up on indirect src0.
- intel/fs/gen12: Implement gl_FrontFacing on gen12+.
- intel/genxml: Remove W-tiling on gen12
- intel/isl: Select Y-tiling for stencil on gen12
- intel/isl: Add isl_aux_usage_has_ccs
- spirv/info: Add a memorymodel_to_string helper
- Revert "mapi: Inline call x86_current_tls."
- intel/blorp: Use surf instead of aux_surf for image dimensions
- intel/isl: Add new aux modes available on gen12
- intel/isl/fill_state: Separate aux_mode handling from aux_surf
- intel/isl: Update surf_fill_state for gen12
- intel/isl: Support HIZ_CCS in emit_depth_stencil_hiz
- anv: Delay allocation of relocation lists
- anv: Reduce the minimum number of relocations
- intel/vec4: Set brw_stage_prog_data::has_ubo_pull
- anv: Avoid emitting UBO surface states that won't be used
- anv: Fix a potential BO handle leak
- anv/tests: Zero-initialize instances
- anv: Set the batch allocator for compute pipelines
- anv: Stop bounds-checking pushed UBOs
- anv: Set up SBE_SWIZ properly for gl_Viewport
- anv: Re-emit all compute state on pipeline switch
- anv: Don't leak when set_tiling fails

Jean Hertel (1):

- Fix missing dri2_load_driver on platform_drm

Jiadong Zhu (1):

- mesa: fix texStore for FORMAT_Z32_FLOAT_S8X24_UINT

Jiang, Sonny (1):

- loader: always map the "amdgpu" kernel driver name to radeonsi (v2)

John Stultz (1):

- Android.mk: Fix missing \ from recent llvm change

Jon Turney (2):

- Fix timespec_from_nsec test for 32-bit time_t
- rbug: Fix use of alloca() without #include "c99_alloca.h"

Jonathan Gray (3):

- mapi: Adapted libglvnd x86 tsd changes
- winsys/amdgpu: avoid double simple_mtx_unlock()
- i965: update Makefile.sources for perf changes

Jonathan Marek (90):

- freedreno/a2xx: ir2: fix lowering of instructions after float lowering
- freedreno/a2xx: ir2: remove pointcoord y invert
- freedreno/a2xx: ir2: set lower_fdph
- freedreno/a2xx: ir2: fix saturate in cp
- freedreno/a2xx: ir2: check opcode on the right instruction in export cp
- freedreno/a2xx: ir2: fix incorrect instruction reordering
- freedreno/a2xx: ir2: update register state in scalar insert
- freedreno/a2xx: fix SRC_ALPHA_SATURATE for alpha blend function
- freedreno/a2xx: implement polygon offset
- freedreno/a2xx: fix depth gmem restore
- freedreno/a2xx: formats update
- u_format: add ETC2 to util_format_srgb/util_format_linear
- u_format: float type for R11G11B10_FLOAT/R9G9B9E5_FLOAT
- etnaviv: fix two-sided stencil
- turnip: fix binning shader compilation
- turnip: use image tile_mode for gmem configuration
- turnip: emit shader immediates
- turnip: fix vertex_id
- turnip: implement sampler state
- turnip: implement image view descriptor
- turnip: use linear tiling for scanout image
- turnip: align layer_size
- turnip: enable linear filtering
- turnip: basic descriptor sets (uniform buffer and samplers)
- turnip: lower samplers and uniform buffer indices
- turnip: use nir_opt_copy_prop_vars
- turnip: add some shader information in pipeline state
- turnip: emit texture and uniform state

- etnaviv: nir: fix gl_FrontFacing
- etnaviv: nir: allocate contiguous components for LOAD destination
- etnaviv: nir: set num_components for inputs/outputs
- qetnaviv: nir: use new immediates when possible
- etnaviv: nir: add native integers (HALTI2+)
- etnaviv: nir: use store_deref instead of store_output
- etnaviv: nir: remove "options" struct
- etnaviv: remove extra allocation for shader code
- etnaviv: nir: make lower_alu easier to follow
- etnaviv: disable earlyZ when shader writes fragment depth
- etnaviv: nir: fix gl_FragDepth
- etnaviv: update headers from rnndb
- etnaviv: implement texture comparator
- etnaviv: set texture INT_FILTER bit
- etnaviv: clear texture cache and flush ts when texture is modified
- etnaviv: get addressing mode from tiling layout
- etnaviv: rework compatible render base
- etnaviv: rework etna_resource_create tiling choice
- freedreno/ir3: remove input ncomp field
- freedreno/ir3: increase size of inputs/outputs arrays
- freedreno/ir3: implement fdd{x,y}_coarse opcodes
- freedreno/ir3: fix GETLOD for negative LODs
- freedreno/ir3: implement texop_texture_samples
- freedreno/ir3: implement fquantize2f16
- freedreno/regs: update a6xx 2d blit bits
- turnip: fix triangle strip
- turnip: fix 32 vertex attributes case
- turnip: fix segmentation fault in events
- turnip: fix segmentation fault with compute pipeline
- turnip: fix assert failing for 0 color attachments
- turnip: add astc format layout
- turnip: add format_is_uint/format_is_sint
- turnip: format table fixes
- turnip: add more 2d_ifmt translations
- turnip: improve view descriptor
- turnip: improve sampler descriptor

- turnip: add black border color
- turnip: add VK_KHR_sampler_mirror_clamp_to_edge
- turnip: update setup_slices
- turnip: disable tiling as necessary
- turnip: add anisotropy and compressed formats to device features
- turnip: update some shader state bits from GL driver
- turnip: fixup consts
- turnip: add code to lower indirect samplers
- turnip: add missing nir passes
- turnip: use nir_assign_io_var_locations instead of nir_assign_var_locations
- turnip: improve CmdCopyImage and implement CmdBlitImage
- turnip: basic msaa working
- turnip: depth/stencil
- turnip: push constants
- turnip: more descriptor sets
- spirv: set correct dest_type for texture query ops
- etnaviv: fix linear_nearest / nearest_linear filters on GC7000Lite
- etnaviv: fix TS samplers on GC7000L
- etnaviv: check NO_ASTC feature bit
- freedreno/a2xx: use sysval for pointcoord
- freedreno/a2xx: add missing vertex formats (SSCALE/USCALE/FIXED)
- etnaviv: fix depth bias
- etnaviv: stencil fix
- etnaviv: fix non-pointsprite points on GC7000L
- freedreno/ir3: disable texture prefetch for 1d array textures
- freedreno/registers: fix a6xx_2d_blit_cntl ROTATE

Jordan Justen (42):

- intel/genxml: Handle field names with different spacing/hyphen
- intel/genxml/gen11: Add spaces in EnableUnormPathInColorPipe
- intel/genxml: Run sort_xml.sh to tidy gen9.xml and gen11.xml
- intel/genxml: Add gen12.xml as a copy of gen11.xml
- intel/genxml: Build gen12 genxml
- intel/isl: Build gen12 using gen11 code paths
- intel/compiler: Disable compaction on gen12 for now
- intel/l3: Don't assert on gen12 (use gen11 config temporarily)
- iris: Build for gen12

- anv: Build for gen12
- i965: Exit with error if gen12+ is detected
- pci_id_driver_map: Support preferring iris over i965
- anv,iris: L3ALLOC register replaces L3CNTLREG for gen12
- iris/state: Move reg/mem load/store functions earlier in file
- intel/ir: Lower fpow on Gen12.
- intel/genxml,isl: Add gen12 render surface state changes
- intel/genxml,isl: Add gen12 depth buffer changes
- intel/genxml,isl: Add gen12 stencil buffer changes
- intel/isl: Add gen12 depth/stencil surface alignments
- iris: Let isl decide the supported tiling in more situations
- intel/isl: Add R10G10B10_FLOAT_A2_UNORM format
- iris/resource: Use isl surface alignment during bo allocation
- intel/common: Add interface to allocate device buffers
- anv: Implement aux-map allocator interface
- intel/common: Add surface to aux map translation table support
- anv/gen12: Initialize aux map context
- genxml/gen12: Add AUX MAP register definitions
- anv/gen12: Write GFX_AUX_TABLE base address register
- iris/bufmgr: Initialize aux map context for gen12
- isl/gen12: 64k surface alignment
- iris: Map each surf to it's aux-surf in the aux-map tables
- iris/gen12: Write GFX_AUX_TABLE base address register
- iris: Mark aux-map BO as used by all batches
- intel: Update alignment restrictions for HiZ surfaces.
- iris: Set MOCS for external surfaces to uncached
- intel/genxml: Add gen12 tile cache flush bit
- intel/dev: Add preliminary device info for Tigerlake
- intel/eu/validate/gen12: Add TGL to eu_validate tests.
- docs/relnotes/new_features.txt: Add note about gen12 support
- iris: Add IRIS_DIRTY_RENDER_BUFFER state flag
- iris/gen11+: Move flush for render target change
- iris: Allow max dynamic pool size of 2GB for gen12

Jose Maria Casanova Crespo (5):

- mesa: recover target_check before get_current_tex_objects
- v3d: writes to magic registers aren't RF writes after THREND

- v3d: flag dirty state when binding compute states
- v3d: Explicitly expose OpenGL ES Shading Language 3.1
- v3d: Fix predication with atomic image operations

José Fonseca (5):

- glx: Fix incompatible function pointer types.
- util: Prevent implicit declaration of function getenv.
- util: Prevent strcasecmp macro redefinion.
- scons: Make GCC builds stricter.
- scons: Fix force_scons parsing.

Juan A. Suarez Romero (14):

- docs: add release notes for 19.1.5
- docs: add sha256 checksums for 19.1.5
- docs: update calendar, add news item and link release notes for 19.1.5
- docs: add release notes for 19.1.6
- docs: add sha256 checksums for 19.1.6
- docs: update calendar, add news item and link release notes for 19.1.6
- docs: extend 19.1.x releases
- docs: add release notes for 19.1.7
- docs: add sha256 checksums for 19.1.7
- docs: update calendar, add news item and link release notes for 19.1.7
- bin/get-pick-list.sh: sha1 commits can be smaller than 8 chars
- docs: add release notes for 19.1.8
- docs: add release notes for 19.1.8
- docs: update calendar, add news item and link release notes for 19.1.8

Karol Herbst (15):

- gallium: add blob field to pipe_llvm_program_header
- rename pipe_llvm_program_header to pipe_binary_program_header
- clover/functional: add id_equals helper
- clover: add support for drivers having no proper binary format
- clover: prepare supporting multiple IRs
- clover: add support for passing kernels as nir to the driver
- nvc0: expose spirv support
- clover/nir: fix compilation with g++-5.5 and maybe earlier
- nv50/ir: fix unnecessary parentheses warning
- nv50/ir/nir: comparison of integer expressions of different signedness warning
- clover/llvm: remove harmful std::move call

- clover/codegen: remove unused get_symbol_offsets function
- clover: eliminate "ignoring attributes on template argument" warning
- st/mesa: fix crash for drivers supporting nir defaulting to tgsi
- nv50/ir: remove DUMMY edge type

Ken Mays (1):

- haiku: fix Mesa build

Kenneth Graunke (86):

- gallium/ddebug: Wrap resource_get_param if available
- gallium/trace: Wrap resource_get_param if available
- gallium/rbug: Wrap resource_get_param if available
- gallium/noop: Implement resource_get_param
- iris: Replace devinfo->gen with GEN_GEN
- iris: Fix broken aux.possible/sampler_usages bitmask handling
- iris: Update fast clear colors on Gen9 with direct immediate writes.
- iris: Drop copy format hacks from copy region based transfer path.
- iris: Avoid unnecessary resolves on transfer maps
- iris: Set MOCS in all STATE_BASE_ADDRESS commands
- iris: Fix large timeout handling in rel2abs()
- isl: Drop UnormPathInColorPipe for buffer surfaces.
- isl: Don't set UnormPathInColorPipe for integer surfaces.
- iris: Delete dead prototype
- intel/compiler: Fix src0/desc setter ordering
- intel/compiler: Handle bits 15:12 in brw_send_indirect_split_message()
- intel/compiler: Refactor FB write message control setup into a helper.
- intel/compiler: Use generic SEND for Gen7+ FB writes
- intel/compiler: Use new Gen11 headerless RT writes for MRT cases
- util: Add a _mesa_i64roundevenf() helper.
- mesa: Fix _mesa_float_to_unorm() on 32-bit systems.
- iris: Drop swizzling parameter from s8_offset.
- iris: Don't auto-flush/dirty on transfer unmap for coherent buffers
- iris: Actually describe bo_reuse driconf option
- iris: Fix partial fast clear checks to account for miplevel.
- iris: Lessen texture cache hack flush for blits/copies on Icelake.
- iris: Report correct number of planes for planar images
- iris: Invalidate state/texture/constant caches after STATE_BASE_ADDRESS
- intel: Stop redirecting state cache to command streamer cache section

- iris: Support the disable_throttling=true driconf option.
- iris: Ignore line stipple information if it's disabled
- iris: Add support for the always_flush_cache=true debug option.
- iris: Optimize out redundant sampler state binds
- iris: Avoid flushing for cache history on transfer range flushes
- iris: Fix constant buffer sizes for non-UBOs
- gallium: Fix util_format_get_depth_only
- iris: Finish initializing the BO before stuffing it in the hash table
- iris: Set bo->reusable = false in iris_bo_make_external_locked
- st/mesa: Only pause queries if there are any active queries to pause.
- iris: trivial whitespace fixes
- iris: Initialize ice->state.prim_mode to an invalid value
- st/mesa: Prefer 5551 formats for GL_UNSIGNED_SHORT_5_5_5_1.
- st/mesa: Increase GL_POINT_SIZE_RANGE minimum to 1.0
- intel/compiler: Set "Null Render Target" ex_desc bit on Gen11
- iris: Skip allocating a null surface when there are 0 color regions.
- iris: Flag IRIS_DIRTY_BINDINGS_XS on constant buffer rebinds
- iris: Explicitly emit 3DSTATE_BTP_XS on Gen9 with DIRTY_CONSTANTS_XS
- iris: Don't flag IRIS_DIRTY_BINDINGS for constant usage history
- iris: Track per-stage bind history, reduce work accordingly
- intel/compiler: Record whether any pull constant loads occur
- iris: Avoid uploading SURFACE_STATE descriptors for UBOs if possible
- iris: Use state_refs for draw parameters.
- iris: Rework iris_update_draw_parameters to be more efficient
- iris: Skip double-disabling TCS/TES/GS after BLORP operations
- isl: Drop WaDisableSamplerL2BypassForTextureCompressedFormats on Gen11
- st/mesa: Bail on incomplete attachments in discard_framebuffer
- intel/genxml: Stop manually scrubbing 'α' -> "alpha"
- broadcom/genxml: Stop manually scrubbing 'α' -> "alpha"
- Revert "intel/gen11+: Enable Hardware filtering of Semi-Pipelined State in WM"
- intel: Increase Gen11 compute shader scratch IDs to 64.
- iris: Only resolve for image levels/layers which are actually in use.
- iris: Disable CCS_E for 32-bit floating point textures.
- iris: Fix iris_rebind_buffer() for VBOs with non-zero offsets.
- st/dri: Perform MSAA downsampling for __DRI2_THROTTLE_COPYSUBBUFFER
- dri: Avoid swapbuffer throttling in glXCopySubBufferMESA

- iris: Refactor push constant allocation so we can reuse it
- iris: Hack up a SKL/Gen9LP PS push constant fifo depth workaround
- Revert "iris: Hack up a SKL/Gen9LP PS push constant fifo depth workaround"
- iris: Drop bonus parameters from iris_init_*_context()
- iris: Drop vtbl usage for some load_register calls
- iris: Update comment about 3-component formats and buffer textures
- iris: Properly unreference extra VBOs for draw parameters
- st/mesa: Fix inverted polygon stipple condition
- iris: Implement the Broadwell NP Z PMA Stall Fix
- intel/fs/gen12: Use TCS 8_PATCH mode.
- iris: Implement the Gen < 9 tessellation quads workaround
- mesa: Use ctx->ReadBuffer in glReadBuffer back-to-front tests
- mesa: Make back_to_front_if_single_buffered non-static
- mesa: Handle pbuffers in desktop GL framebuffer attachment queries
- intel/compiler: Report the number of non-spill/fill SEND messages
- st/mesa: Silence chatty debug printf
- iris: Rework edgeflag handling
- nir: Use VARYING_SLOT_TESS_MAX to size indirect bitmasks
- iris: Fix "Force Zero RTA Index Enable" setting again
- driconf, glsl: Add a vs_position_always_invariant option
- drirc: Set vs_position_always_invariant for Shadow of Mordor on Intel

Kevin Strasser (14):

- drm-uapi: Update headers for fp16 formats
- i965: Add helper function for allowed config formats
- gallium: Use consistent approach for config format filtering
- dri: Add config attributes for color channel shift
- util: move bitcount to bitscan.h
- egl: Convert configs to use shifts and sizes instead of masks
- glx: Add fields for color shifts
- dri: Handle configs with floating point pixel data
- egl: Handle dri configs with floating point pixel data
- dri: Add fp16 formats
- gbm: Add buffer handling and visuals for fp16 formats
- i965: Add handling for fp16 configs
- gallium: Add buffer and configs handling or fp16 formats
- egl: Fix implicit declaration of ffs

Khaled Emara (2):

- freedreno/a3xx: fix texture tiling parameters
- freedreno/a3xx: fix sysmem <-> gmem tiles transfer

Kristian Høgsberg (40):

- freedreno/a6xx: Let the GPU track streamout offsets
- freedreno/a6xx: Implement primitive count queries on GPU
- freedreno/a6xx: Track location of gl_Position out as we link it
- freedreno/a6xx: Share shader state constructor and destructor
- freedreno/a6xx: Turn on vectorize_io
- freedreno/a6xx: Write multiple regs for SP_VS_OUT_REG and SP_VS_VPC_DST_REG
- freedreno/regs: Fix CP_DRAW_INDX_OFFSET command
- freedreno/regs: A couple of tess updates
- freedreno/a6xx: Factor out const state setup
- freedreno: Rename vp and fp to vs and fs in fd_program_stateobj
- freedreno: Add state binding functions for HS/DS/GS
- freedreno: Move fs functions after geometry pipeline stages
- freedreno/a6xx: Add generic program stateobj support for HS/DS/GS
- freedreno/ir3: Add HS/DS/GS to shader key and cache
- freedreno/a6xx: Emit const and texture state for HS/DS/GS
- freedreno/a6xx: Move instrlen and obj_start writes to fd6_emit_shader
- freedreno/registers: Update with GS, HS and DS registers
- freedreno/a6xx: Trim a few regs from fd6_emit_restore()
- freedreno/ir3: Add support for CHSH and CHMASK instructions
- freedreno/ir3: Use third register for offset for LDL and LDLV
- freedreno/ir3: Extend RA with mechanism for pre-coloring registers
- freedreno/ir3: Add new LDLW/STLW instructions
- freedreno/ir3: Add intrinsics that map to LDLW/STLW
- freedreno/a6xx: Add missing adjacency primitives to table
- freedreno/ir3: Add has_gs flag to shader key
- freedreno/ir3: Implement lowering passes for VS and GS
- freedreno/ir3: Implement primitive layout intrinsics
- freedreno/ir3: Setup ir3 inputs and outputs for GS
- freedreno/ir3: Pre-color GS header and primitive ID
- freedreno/ir3: Start GS with (ss) and (sy)
- freedreno/ir3: End VS with CHMASK and CHSH in GS pipelines
- freedreno/a6xx: Emit program state for GS

- freedreno/a6xx: Support layered render targets
- st/mesa: Also enable GS when ESSLVersion > 320
- freedreno/blitter: Save GS state
- freedreno/a6xx: Implement PIPE_QUERY_PRIMITIVES_GENERATED for GS
- freedreno/ci: Add failing tests to skip list
- freedreno/a6xx: Turn on geometry shaders
- nir: Use BITSET for tracking varyings in lower_io_arrays
- freedreno/a6xx: Disable geometry shaders for release

Krzysztof Raszkowski (2):

- util: Add unreachable() definition for clang compiler.
- gallium/swr: Enable GL_ARB_gpu_shader5: multiple streams

Laurent Carlier (1):

- egl: avoid local modifications for eglext.h Khronos standard header file

Leo Liu (3):

- radeon/vcn: add RENOIR VCN decode support
- radeon/vcn: Add VP9 8K decode support
- radeonsi: enable 8K video decode support for HEVC and VP9

Lepton Wu (14):

- st/mesa: Allow zero as [level|layer]_override
- virgl: Fix pipe_resource leaks under multi-sample.
- egl/android: Only keep BGRA EGL configs as fallback
- virgl: replace fprintf with _debug_printf
- virgl: Remove wrong EAGAIN handling for drmIoctl
- gbm: Add GBM_MAX_PLANES definition
- egl/android: Remove our own reference to buffers.
- virgl: Remove formats with unusual sample count.
- mapi: Inline call x86_current_tls.
- mapi: split entry_generate_or_patch for x86 tls
- mapi: Clean up entry_patch_public for x86 tls
- mapi: Inline call x86_current_tls.
- mapi: Improve the x86 tsd stubs performance.
- gallium: dri2: Use index as plane number.

Lionel Landwerlin (59):

- glsl/tests: take refs on glsl types
- nir/tests: take reference on glsl types
- compiler: ensure glsl types are not created without a reference

- mesa/compiler: rework tear down of builtin/types
- radeonsi: take reference glsl types for compile threads
- i965: honor scanout requirement from DRI
- util/timespec: use unsigned 64 bit integers for nsec values
- util: fix compilation on macos
- egl: fix platform selection
- vulkan/overlay: bounce image back to present layout
- intel: update product names for WHL
- radv: store engine name
- driconfig: add a new engine name/version parameter
- vulkan: add vk_x11_strict_image_count option
- util/xmlconfig: fix regexp compile failure check
- drirc: include unreal engine version 0 to 23
- anv: gem-stubs: return a valid fd got anv_gem_userptr()
- intel: use proper label for Comet Lake skus
- intel: Add new Comet Lake PCI-ids
- mesa: don't forget to clear _Layer field on texture unit
- intel: fix topology query
- intel/error2aub: add support for platforms without PPGTT
- intel: fix subslice computation from topology data
- intel/isl: Set null surface format to R32_UINT
- intel/isl: set surface array appropriately
- intel/isl: set vertical surface alignment on null surfaces
- etnaviv: remove variable from global namespace
- anv: fix vkUpdateDescriptorSets with inline uniform blocks
- anv: fix memory leak on device destroy
- anv: fix unwind of vkCreateDevice fail
- intel/perf: add mdapi maker helper
- intel/perf: expose some utility functions
- intel/perf: extract register configuration
- intel/perf: move registers to their own header
- drm-uapi: Update headers from drm-next
- intel/perf: add support for querying kernel loaded configurations
- intel/genxml: add generic perf counters registers
- intel/genxml: add RPSTAT register for core frequency
- intel/perf: add mdapi writes for register perf counters

- anv: implement VK_INTEL_performance_query
- docs: Add new Intel extension
- intel/dev: store whether the device uses an aux map tables on devinfo
- anv: Add aux-map translation for gen12+
- intel/perf: update ICL configurations
- intel/dump_gpu: handle context create extended ioctl
- intel/dev: set default num_eu_per_subslice on gen12
- mesa: check draw buffer completeness on glClearBufferfi/glClearBufferiv
- anv: Properly handle host query reset of performance queries
- mesa: check framebuffer completeness only after state update
- anv: invalidate file descriptor of semaphore sync fd at vkQueueSubmit
- anv: remove list items on batch fini
- anv/wsi: signal the semaphore in the acquireNextImage
- intel/perf: fix invalid hw_id in query results
- intel/perf: set read buffer len to 0 to identify empty buffer
- intel/perf: take into account that reports read can be fairly old
- intel/perf: simplify the processing of OA reports
- intel/perf: fix improper pointer access
- anv: fix missing gen12 handling
- anv: fix incorrect VMA alignment for CCS main surfaces

Lucas Stach (17):

- etnaviv: fix vertex buffer state emission for single stream GPUs
- gallium/util: don't depend on implementation defined behavior in listen()
- rbug: fix transmitted texture sizes
- rbug: unwrap index buffer resource
- rbug: move flush_resource initialization
- rbug: implement missing explicit sync related fence functions
- rbug: forward texture_barrier to pipe driver
- rbug: forward can_create_resource to pipe driver
- rbug: implement resource creation with modifier
- rbug: remove superfluous NULL check
- etnaviv: keep references to pending resources
- etnaviv: drm: remove unused etna_cmd_stream_finish
- etnaviv: rework the stream flush to always go through the context flush
- etnaviv: drm: add softpin interface
- etnaviv: check for softpin availability on Halti5 devices

- etnaviv: add linear texture support on GC7000
- etnaviv: GC7000: flush TX descriptor and instruction cache

Marek Olšák (161):

- radeonsi/gfx10: fix the legacy pipeline by storing as_ngg in the shader cache
- radeonsi: move some global shader cache flags to per-binary flags
- radeonsi/gfx10: fix tessellation for the legacy pipeline
- radeonsi/gfx10: fix the PRIMITIVES_GENERATED query if using legacy streamout
- radeonsi/gfx10: create the GS copy shader if using legacy streamout
- radeonsi/gfx10: add as_ngg variant for VS as ES to select Wave32/64
- radeonsi/gfx10: fix InstanceID for legacy VS+GS
- radeonsi/gfx10: don't initialize VGT_INSTANCE_STEP_RATE_0
- radeonsi/gfx10: always use the legacy pipeline for streamout
- radeonsi/gfx10: finish up Navi14, add PCI ID
- radeonsi/gfx10: add AMD_DEBUG=nongg
- winsys/amdgpu+radeon: process AMD_DEBUG in addition to R600_DEBUG
- radeonsi: add PKT3_CONTEXT_REG_RMW
- radeonsi/gfx10: remove incorrect ngg/pos_writes_edgeflag variables
- radeonsi/gfx10: set PA_CL_VS_OUT_CNTL with CONTEXT_REG_RMW to fix edge flags
- radeonsi: consolidate determining VGPR_COMP_CNT for API VS
- radeonsi: align scratch and ring buffer allocations for faster memory access
- radeonsi: unbind blend/DSA/rasterizer state correctly in delete functions
- radeonsi: fix scratch buffer WAVESIZE setting leading to corruption
- ac: enable LLVM atomic optimizations
- ac: use fma on gfx10
- radeonsi/gfx10: use fma for TGSI_OPCODE_FMA
- radeonsi/gfx10: don't call gfx10_destroy_query with compute-only contexts
- radeonsi: disable DCC when importing a texture from an incompatible driver
- radeonsi: only support at most 1024 threads per block
- radeonsi/gfx10: fix wave occupancy computations
- r300,r600,radeonsi: read winsys_handle::stride,offset in drivers, not winsyses
- r300,r600,radeonsi: set winsys_handle::stride,offset in drivers, not winsyses
- ac/surface: add RADEON_SURF_NO_FMASK
- radeonsi: handle NO_DCC early
- radeonsi: move HTILE allocation outside of radeonsi
- radeonsi: move texture storage allocation outside of radeonsi
- radeonsi: remove redundant si_texture offset and size fields

- ac: replace HAVE_LLVM with LLVM_VERSION_MAJOR for atomic-optimizations
- prog_to_nir, tgsi_to_nir: make sure kill doesn't discard NaNs
- radeonsi/gfx9: honor user stride for imported buffers
- radeonsi: add Navi12 PCI ID
- ac: move PBB MAX_ALLOC_COUNT into radeon_info
- ac: move num_sdp_interfaces into radeon_info
- ac: move ac_get_max_wave64_per_simd into radeon_info
- ac: move ac_get_num_physical_sgprs into radeon_info
- ac: move ac_get_num_physical_vgprs into radeon_info
- gallium: extend resource_get_param to be as capable as resource_get_handle
- radeonsi: implement pipe_screen::resource_get_param
- radeonsi: include drm_fourcc.h to fix the build
- amd: add more PCI IDs for Navi14
- ac/addrlib: fix chip identification for Vega10, Arcturus, Raven2, Renoir
- ac: stop using PCI IDs for chip identification
- amd: remove all PCI IDs supported by amdgpu
- nir: don't add bindless variables to num_textures and num_images
- nir: define 8-byte size and alignment for bindless variables
- tgsi_to_nir: fix masked out image loads
- tgsi_to_nir: fix 2-component system values like tess_level_inner_default
- ac/nir: port Z compare value clamping from radeonsi
- ac/nir: force unnormalized coordinates for RECT
- radeonsi: initialize displayable DCC using the retile blit to prevent hangs
- gallium/vl: don't set PIPE_HANDLE_USAGE_EXPLICIT_FLUSH
- radeonsi/gfx10: fix L2 cache rinse programming
- ac: fix incorrect vram_size reported by the kernel
- ac: add radeon_info::tcc_harvested
- radeonsi/gfx10: fix corruption for chips with harvested TCCs
- ac: fix num_good_cu_per_sh for harvested chips
- ac: set the number of SDPs same as the number of TCCs
- ac: reorder and print all radeon_info fields
- tgsi_to_nir: handle PIPE_FORMAT_NONE in image opcodes
- ac/surface: don't allocate FMASK if there is no graphics
- ac: add ac_build_image_get_sample_count from radeonsi
- ac/nir: fix GLSL imageSamples()
- winsys/radeon: initialize SIMD properties in radeon_info

- util: use simple_mtx_t for util_range
- gallium: add PIPE_RESOURCE_FLAG_SINGLE_THREAD_USE to skip util_range lock
- st/mesa: use simple_mtx_t instead of mtx_t
- radeonsi: use simple_mtx_t instead of mtx_t
- amd: don't use AMD_FAMILY definitions from amdgpu_drm.h
- gallium/util: remove enum numbering from util_format_layout
- gallium/util: add planar format layouts and helpers
- gallium/u_tests: test NV12 allocation and export
- vl: use u_format in vl_video_buffer_formats
- radeonsi: allocate planar multimedia formats in 1 buffer
- radeonsi: remove si_vid_join_surfaces and use combined planar allocations
- radeonsi: ignore metadata for non-zero planes
- radeonsi: don't set BO metadata for non-zero planes
- nir: add shader_info::last_msaa_image
- tgsi/scan: add tgsi_shader_info::msaa_images_declared
- radeonsi: fix GLSL imageSamples()
- radeonsi: set the sample index for shader images correctly
- radeonsi: add FMASK slots for shader images (for MSAA images)
- radeonsi: clean up image_fetch_rsrc
- radeonsi: apply FMASK to MSAA image loads
- radeonsi: expand FMASK before MSAA image stores are used
- radeonsi: enable MSAA shader images
- nir: add a strip parameter to nir_serialize
- nir: move gl_nir_opt_access from glsl directory
- nir/drawpixels: handle load_color0, load_input, load_interpolated_input
- nir/drawpixels: fix what appears to be a copy-paste bug in get_texcoord_const
- tgsi_to_nir: add #ifdef header guards
- nir: add nir_shader_compiler_options::lower_to_scalar
- st/mesa: use nir_shader_compiler_options::lower_to_scalar
- tgsi_to_nir: use nir_shader_compiler_options::lower_to_scalar
- gallium: remove PIPE_SHADER_CAP_SCALAR_ISA
- ac/nir: add back nir_op_fmod
- clover: fix the nir_serialize build failure
- st/mesa: always allocate pack/unpack buffers as staging
- radeonsi/nir: simplify si_lower_nir signature
- st/mesa: use *prog at the end of st_link_nir

- st/mesa: deduplicate code for ATI fs in st_program_string_notify
- st/mesa: simplify the signature of st_release_basic_variants
- st/mesa: don't store stream output info to shader cache for tess ctrl shaders
- st/mesa: remove st_compute_program in favor of st_common_program
- st/mesa: deduplicate cases in st_deserialise_ir_program
- st/mesa: sink TCS/TES/GS/CS translate code into st_translate_common_program
- st/mesa: deduplicate st_common_program code in st_program_string_notify
- st/mesa: clean up more after the removal of st_compute_program
- st/mesa: move vertex program preparation code into st_prepare_vertex_program
- st/mesa: unify transform feedback info translation code
- st/mesa: finalize NIR after shader variant passes for TCS/TES/GS/CS
- st/mesa: don't call translate_*_program functions for NIR
- st/mesa: call prog_to_nir sooner for ARB_fp
- st/mesa: reorder and document code in st_translate_vertex_program
- st/mesa: call the reset callback if glGetGraphicsResetStatus returns a failure
- radeonsi: call the reset callback if get_device_reset_status returns a failure
- radeonsi: recreate aux_context after a GPU reset
- gallium/u_blitter: remove an unused variable
- st/mesa: silence a warning in st_nir_lower_tex_src_plane
- st/mesa: call st_nir_opts for linked shaders only once
- st/mesa: lower doubles for NIR after linking
- st/mesa: rename st_xxx_program::tgsi to state
- st/mesa: rename basic -> common for st_common_program
- st/mesa: remove num_tgsi_tokens from st_xx_program
- st/mesa: remove st_vp_variant_key in favor of st_common_variant_key
- st/mesa: remove unused st_xxx_program::sha1
- st/mesa: remove redundant function st_reference_compprog
- st/mesa: merge st_fragment_program into st_common_program
- st/mesa: don't call variables "tgsi" when they can reference NIR
- nir: allow nir_lower_uniforms_to_ubo to be run repeatedly
- st/mesa: replace pipe_shader_state with tgsi_token* in st_vp_variant
- gallium/noop: implement get_disk_shader_cache and get_compiler_options
- util/disk_cache: finish all queue jobs in destroy instead of killing them
- util/u_queue: skip util_queue_finish if num_threads is 0
- st/mesa: move some NIR lowering before shader caching
- st/mesa: don't lower_global_vars_to_local for VS if there are no dead inputs

- st/mesa: assign driver locations for VS inputs for NIR before caching
- st/mesa: update VS shader_info for NIR after lowering passes
- gallium: add pipe_screen::finalize_nir
- tgsi_to_nir: use pipe_screen::finalize_nir
- st/mesa: use pipe_screen::finalize_nir
- radeonsi/nir: implement pipe_screen::finalize_nir
- glsl/serialize: restructure remap table code
- glsl/serialize: optimize for equal offsets in uniform remap tables
- include: add the definition of EGL_EXT_image_flush_external
- dri_interface: add interface for EGL_EXT_image_flush_external
- st/dri: assume external consumers of back buffers can write to the buffers
- st/dri: add support for EGL_EXT_image_flush_external
- egl: handle EGL_IMAGE_EXTERNAL_FLUSH_EXT
- egl: implement new functions from EGL_EXT_image_flush_external
- docs: document new feature EGL_EXT_image_flush_external
- radeonsi: don't print diagnostic LLVM remarks and notes
- radeonsi: initialize shader compilers in threads on demand
- ac: get tcc_harvested from the kernel
- winsys/amdgpu: use the new GPU reset query
- st/mesa: fix Sanctuary and Tropics by disabling ARB_gpu_shader5 for them

Marek Vasut (4):

- etnaviv: Make contexts track resources
- etnaviv: Rework resource status tracking
- etnaviv: Command buffer realloc
- etnaviv: Rework locking

Marijn Suijten (2):

- freedreno/a5xx: enable a510
- freedreno/ir3: Add missing ir3_nir_lower_tex_prefetch.c to Android.mk

Matt Turner (6):

- clover: Remove unused code
- intel/compiler: Remove unreachable() from brw_reg_type.c
- intel/compiler: Restructure instruction compaction in preparation for Gen12
- intel/compiler: Inline get_src_index()
- intel/compiler: Make separate src0/src1 index tables
- intel/compiler: Add instruction compaction support on Gen12

Mauro Rossi (8):

- android: mesa: revert "Enable asm unconditionally"
- android: anv: libmesa_vulkan_common: add libmesa_util static dependency
- android: aco: fix undefined template 'std::__1::array' build errors
- android: compiler/nir: build nir_divergence_analysis.c
- android: aco: add support for libmesa_aco
- android: amd/common: export amd/llvm headers
- android: aco: fix Lower to CSSA
- android: radeonsi: fix build after vl refactoring (v2)

Maya Rashish (3):

- intel/compiler: avoid truncating int64_t to int
- meson: Test for -Wl,–build-id=sha1
- llvmpipe: avoid left-shifting a negative number.

Michael Schellenberger Costa (1):

- aco: Cleanup insert_before_logical_end

Michel Dänzer (48):

- gitlab-ci: Move up meson-main job definition
- gitlab-ci: Use new needs: keyword
- gitlab-ci: Explicitly install linux-libc-dev for foreign architectures
- gitlab-ci: Keep g++ from stretch when installing foreign toolchains
- gitlab-ci: Add needs stanza to arm64_a306_gles2 job definition
- gitlab-ci: Use multiple inheritance instead of YAML references
- gitlab-ci: Simplify some job definitions by extending more similar jobs
- gitlab-ci: Move dependencies/needs for meson-main job to .deqp-test
- gitlab-ci: Move up meson-arm64 job definition
- gallivm: Limit DEBUG workaround to LLVM < 7
- swr: Limit DEBUG workaround to LLVM < 7
- ac: Remove DEBUG workaround
- gitlab-ci: Reference full ci-templates commit hash
- gitlab-ci: Pass –no-remove to apt-get where possible
- gitlab-ci: Create separate docker images for Debian stretch & buster
- gitlab-ci: Use newer packages from backports by default
- gitlab-ci: Use crossbuild-essential-* packages
- gitlab-ci: Move scons build/test commands to a separate shell script
- gitlab-ci: Test scons with all LLVM versions
- gitlab-ci: Merge scons-nollvm and scons-llvm jobs
- radeonsi: fix VAAPI segfault due to various bugs

- loader: Avoid use-after-free / use of uninitialized local variables
- gitlab-ci: Declare needs: for stretch docker image
- gitlab-ci: Add needs: for x86 buster docker image
- gitlab-ci: Add test-container:arm64 to needs: for arm64 test jobs
- gitlab-ci: Set ccache path for cross compilers in meson cross file
- gitlab-ci: Use per-job ccache
- dri3: Pass __DRI2_THROTTLE_COPYSUBBUFFER from loader_dri3_copy_drawable
- loader: Simplify handling of the radeonsi driver
- gitlab-ci/lava: Add needs: for container image to test jobs
- gitlab-ci: Remove redundant .meson-cross template script
- gitlab-ci: Add .use-debian-10 template
- gitlab-ci: Disable meson-mingw32-x86_64 job again for now
- gitlab-ci: Sort ARM docker image packages in alphabetical order
- gitlab-ci: Bring ARM docker image install script in line with x86_64
- gitlab-ci: Explicitly list debian-10 in needs: for .deqp-test template
- gitlab-ci: Use native aarch64 runner for ARM build jobs
- gitlab-ci: Update the meson cross file for LLVM_VERSION as well
- gitlab-ci: Enable llvmpipe in ARM build jobs
- intel/compiler: Don't left-shift by >= the number of bits of the type
- intel/compiler: Cast to target type before shifting left
- intel/fs: Check for NULL key in fs_visitor constructor
- gallium/util: Cast to target type before shifting left
- util: Use uint64_t for shifting left in sign_extend and strunc
- util/tests: Avoid int64_t overflow issues in fast_idiv_by_const test
- gitlab-ci: Enable UBSan for the meson-vulkan job
- gitlab-ci: Only run the pipeline if any files affecting it have changed
- gitlab-ci: Disable meson-windows job for the time being

Michel Zou (1):

- scons: add py3 support

Nanley Chery (47):

- anv/blorp: Use BLORP_BATCH_NO_UPDATE_CLEAR_COLOR
- anv: Properly allocate aux-tracking space for CCS_E
- anv/formats: Disable I915_FORMAT_MOD_Y_TILED_CCS on TGL+
- iris: Drop support for I915_FORMAT_MOD_Y_TILED_CCS on TGL+
- isl: Disable CCS_D on Gen12+
- anv/image: Disable CCS_D on Gen12+

- anv/cmd_buffer: Don't assume CCS_E includes CCS_D
- iris: Don't assume CCS_E includes CCS_D
- isl: Round up some pitches to 512B for Gen12's CCS
- intel/blorp: Halve the Gen12 fast-clear/resolve rectangle
- intel/blorp: Don't assert aux slices match main slices
- anv/private: Modify aux slice helpers for Gen12 CCS
- i965/miptree: Avoid -Wswitch for the Gen12 aux modes
- isl/drm: Map HiZ and CCS tilings to Y
- iris: Allow for non-Y-tiled aux allocation
- isl: Add and use isl_tiling_flag_to_enum()
- isl: Redefine the CCS layout for Gen12
- intel: Enable CCS_E for some formats on Gen12
- intel/blorp: Disable depth testing for slow depth clears
- iris: Clear ::has_hiz when disabling aux
- intel: Use RENDER_SURFACE_STATE::DepthStencilResource
- intel: Use 3DSTATE_DEPTH_BUFFER::ControlSurfaceEnable
- intel: Enable CCS_E for R24_UNORM_X8_TYPELESS on TGL+
- isl: Reduce assertions during aux surf creation
- intel: Support HIZ_CCS in isl_surf_get_ccs_surf
- intel/blorp: Assert against HiZ in surface states
- intel/blorp: Treat HIZ_CCS like HiZ
- iris: Don't guess the aux_usage
- iris: Create an unusable secondary aux surface
- iris: Define initial HIZ_CCS state and transitions
- iris: Enable HIZ_CCS in depth buffer instructions
- isl: Add isl_surf_supports_hiz_ccs_wt()
- intel: Refactor blorp_can_hiz_clear_depth()
- intel/blorp: Satisfy HIZ_CCS fast-clear alignments
- iris: Start using blorp_can_hiz_clear_depth()
- intel: Fix and use HIZ_CCS write through mode
- intel/blorp: Satisfy clear color rules for HIZ_CCS
- iris: Enable HIZ_CCS sampling
- iris: Don't leak the resource for unsupported modifier
- iris: Disallow incomplete resource creation
- iris: Drop iris_resource::aux::extra_aux::bo
- iris: Bail resource creation upon aux creation error

- iris: Determine aux offsets within configure_aux
- iris: Allocate main and aux surfaces together
- gallium/dri2: Fix creation of multi-planar modifier images
- gallium: Store the image format in winsys_handle
- iris: Fix import of multi-planar surfaces with modifiers

Nataraj Deshpande (1):

- egl/android: Enable HAL_PIXEL_FORMAT_RGBA_FP16 format

Neil Armstrong (1):

- Revert "ci: Disable lima until its farm can get fixed."

Neil Roberts (6):

- glsl: Store the precision for a function return type
- nir/builder: Move nir_atan and nir_atan2 from SPIR-V translator
- nir/builtin: Add #include u_math.h to the header
- nir/builtin: Add extern "C" guards to nir_builtin_builder.h
- glsl: Add opcodes for atan and atan2
- glsl/builtin: Add alternate versions of atan using new ops

OBATA Akio (1):

- util: fix to detect NetBSD properly

Paulo Zanoni (8):

- intel/fs: grab fail_msg from v32 instead of v16 when v32->run_cs fails
- intel/fs: make scan/reduce work with SIMD32 when it fits 2 registers
- intel/fs: roll the loop with the <0,1,0> additions in emit_scan()
- intel/fs: the maximum supported stride width is 16
- intel/fs: fix SHADER_OPCODE_CLUSTER_BROADCAST for SIMD32
- intel/fs: don't forget the stride at generate_shuffle
- intel/compiler: remove the operand restriction for src1 on GLK
- intel/compiler: fix nir_op_{i,u}*32 on ICL

Pierre Moreau (5):

- meson: Check for SPIRV-Tools and llvm-spirv
- clover/spirv: Add functions for validating SPIR-V binaries
- clover/spirv: Add functions for parsing arguments, linking programs, etc.
- clover/llvm: Add options for dumping SPIR-V binaries
- clover/llvm: Add functions for compiling from source to SPIR-V

Pierre-Eric Pelloux Prayer (1):

- mesa: implement glTextureStorageNDEXT functions

Pierre-Eric Pelloux-Prayer (23):

- glsl: replace 'x + (-x)' with constant 0
- mesa: fix invalid target error handling for teximage
- mesa: add EXT_dsa glNamedRenderbufferStorageEXT and glGetNamedRenderbufferParameterivEXT
- mesa: add EXT_dsa glClientAttribDefaultEXT / glPushClientAttribDefaultEXT
- mesa: add EXT_dsa NamedProgram functions
- mesa: add EXT_dsa glProgramUniform*EXT functions
- mesa: add EXT_dsa + EXT_texture_buffer_object functions
- mesa: add EXT_dsa + EXT_texture_integer functions
- mesa: add EXT_dsa + EXT_gpu_shader4 functions
- mesa: add EXT_dsa + EXT_gpu_program_parameters functions
- mesa: add EXT_dsa glGetFloati_vEXT/glGetDoublei_vEXT
- mesa: refactor GenerateTextureMipmap handling
- mesa: add EXT_dsa Generate*MipmapEXT functions
- mesa: add EXT_dsa NamedRenderbufferStorageMultisampleEXT function
- mesa: add EXT_dsa NamedCopyBufferSubDataEXT function
- radeonsi: align sdma byte count to dw
- radeonsi: sdma misc fixes
- radeonsi: disable sdma for gfx10
- radeonsi: tell the shader disk cache what IR is used
- mesa: enable msaa in clear_with_quad if needed
- radeonsi: fix shader disk cache key
- radeonsi: fix multi plane buffers creation
- radeonsi: use gfx9.surf_offset to compute texture offset

Plamena Manolova (8):

- genxml: Add 3DSTATE_DEPTH_BOUNDS instruction.
- iris: Add support for depth bounds testing.
- anv: Add support for depth bounds testing.
- genxml: Change 3DSTATE_DEPTH_BOUNDS bias.
- anv: Set depthBounds to true in anv_GetPhysicalDeviceFeatures.
- genxml: Add 3DSTATE_SO_BUFFER_INDEX_* instructions
- iris: Implement new way for setting streamout buffers.
- anv: Implement new way for setting streamout buffers.

Prodea Alexandru-Liviu (4):

- scons/windows: Fix build with LLVM>=8
- scons/MSYS2-MinGW-W64: Fix build options defaults Signed-off-by: Prodea Alexandru-Liviu <liviuprodea@yahoo.com> Reviewed-by: Jose Fonseca <jfonseca@vmware.com> Cc: <mesa-stable@lists.freedesktop.org>

- Appveyor/Meson: Add build test of osmesa gallium Signed-off-by: Prodea Alexandru-Liviu <liviuprodea@yahoo.com> Acked-by: Eric Engestrom <eric@engestrom.ch> Reviewed-by: Dylan Baker <dylan@pnwbakers.com>
- Meson: Remove lib prefix from graw and osmesa when building with Mingw. Also remove version sufix from osmesa swrast on Windows.

Qiang Yu (4):

- lima: move format handling to unified place
- lima: implement EGL_KHR_partial_update
- lima: don't use damage system when full damage
- lima: move damage bound build to resource

Rafael Antognolli (13):

- anv: Only re-emit non-dynamic state that has changed.
- intel/tools: Fix aubinator usage of rb_tree.
- anv/block_pool: Align anv_block_pool state to 64 bits.
- intel/tools: Factor out GGTT allocation.
- intel/tools: Use common code for GGTT address allocation.
- intel/tools: Add basic aub_context code and helpers.
- intel/tools: Support multiple contexts in intel_dump_gpu.
- intel/blorp/gen12: Set FWCC when storing the clear color.
- anv: Align fast clear color state buffer to a page.
- iris: Align fast clear color state buffer to a page.
- iris: Add Tile Cache Flush for Unified Cache.
- blorp: Add Tile Cache Flush for Unified Cache.
- anv: Add Tile Cache Flush for Unified Cache.

Rhys Perry (84):

- nir/lower_io_to_vector: allow FS outputs to be vectorized
- nir/lower_io_to_vector: add flat mode
- util: include u_endian.h in u_math.h
- nir/lower_io_to_vector: don't merge compact varyings
- radv: keep GS threads with excessive emissions which could write to memory
- radv: always emit a position export in gs copy shaders
- radv: never kill a NGG GS shader
- nir/opt_remove_phis: handle phis with no sources
- aco: run nir_lower_int64() before nir_lower_idiv()
- aco: implement 64-bit ineg
- aco: fix GFX9 opcode for v_xad_u32
- aco: fix v_subrev_co_u32_e64 opcode

- aco: fix opcode for s_mul_hi_i32
- aco: check for duplicate opcode numbers
- radv/aco: actually disable ACO when unsupported
- aco,radv/aco: get dissassembly for release builds if requested
- aco: store printed backend IR in binary
- radv/aco: return a correct name and description for the backend IR
- aco,radv: rename record_llvm_ir/llvm_ir_string to record_ir/ir_string
- aco: don't CSE v_readlane_b32/v_readfirstlane_b32
- aco: CSE readlane/readfirstlane/permute/reduce with the same exec mask
- aco: set loop_info::has_discard for demotes
- aco: don't remove the loop exec mask in transition_to_Exact()
- radv/aco,aco: set lower_fmod
- nir/print: always use the right FILE *
- aco: fix load_constant with multiple arrays
- nir/constant_folding: add back and use constant_fold_state
- nir/constant_folding: fold load_constant intrinsics
- aco: move s_andn2_b64 instructions out of the p_discard_if
- aco: enable nir_opt_sink
- aco: Allow literals on VOP3 instructions.
- aco: Assemble opsel in VOP3 instructions.
- aco: workaround GFX10 0x3f branch bug
- aco: pad code with s_code_end on GFX10
- aco: Initial work to avoid GFX10 hazards.
- aco: Use the VOP3-only add/sub GFX10 instructions if needed.
- aco: Have s_waitcnt_vscnt write to NULL.
- radv/aco: disable NGG when ACO is used
- aco/gfx10: fix inline uniform blocks
- aco/gfx10: disable GFX9 1D texture workarounds
- aco: rework scratch resource code
- aco: update print_ir
- nir/lower_non_uniform: lower image/texture instructions taking derefs
- nir/lower_input_attachments: pass on non-uniform access flag
- aco: don't apply sgprs/constants to read/write lane instructions
- aco: use can_accept_constant in valu_can_accept_literal
- aco: readfirstlane vgpr pointers in convert_pointer_to_64_bit()
- aco: implement divergent vulkan_resource_index

- aco: don't use p_as_uniform for vgpr sampler/image indices
- aco: fix scheduling with s_memtime/s_memrealtime
- aco: don't CSE s_memtime
- aco: emit_split_vector() s_memtime results
- nir/lower_idiv: add new llvm-based path
- aco: use nir_lower_idiv_precise
- aco: run opt_algebraic in a loop
- aco: small stage corrections
- aco: fix 64-bit p_extract_vector on 32-bit p_create_vector
- aco: create load_lds/store_lds helpers
- aco: fix sparse store_lds()
- aco: properly combine additions into ds_write2_b64/ds_read2_b64
- aco: use ds_read2_b64/ds_write2_b64
- aco: add a few missing checks in value numbering
- aco: keep can_reorder/barrier when combining addition into SMEM
- aco: add missing bld.scc()
- Revert "aco: only emit waitcnt on loop continues if we there was some load or export"
- radv: round vgprs/sgprs before calculating max_waves
- aco: increase accuracy of SGPR limits
- aco: take LDS into account when calculating num_waves
- aco: Fix reductions on GFX10.
- aco: Remove dead code in reduction lowering.
- aco: try to group together VMEM loads of the same resource
- aco: a couple loop handling fixes for GFX10 hazard pass
- aco: rename README to README.md
- aco: fix new_demand calculation for first instructions
- aco: fix shuffle with uniform operands
- aco: fix read_invocation with VGPR lane index
- aco: don't propagate vgprs into v_readlane/v_writelane
- aco: don't combine literals into v_cndmask_b32/v_subb/v_addc
- aco: fix 64-bit fsign with 0
- aco: propagate p_wqm on an image_sample's coordinate p_create_vector
- aco: fix i2i64
- aco: add v_nop inbetween exec write and VMEM/DS/FLAT
- radv: set writes_memory for global memory stores/atomics
- nir/lower_io_to_vector: don't create arrays when not needed

Rob Clark (60):

- freedreno/ir3: convert block->predecessors to set
- freedreno/ir3: maintain predecessors/successors
- freedreno/ir3: do better job of marking convergence points
- nir: remove unused constant_fold_state
- freedreno/drm: fix 64b iova shifts
- freedreno/ir3: use uniform base
- freedreno/ir3: cleanup "partially const" ubo srcs
- freedreno/ir3: fix addr/pred spilling
- freedreno/ir3: fix mad copy propagation special case
- freedreno/ir3: assert that only single address
- freedreno/ir3: fix cp cmps.s opt
- freedreno/ir3: allow copy propagation for relative
- util: android logging support
- freedreno/a6xx: don't tile things that are too small
- freedreno/a6xx: fix 3d tex layout
- freedreno: fix compiler warning
- freedreno/a6xx: pre-calculate userconst stateobj size
- gitlab-ci/a630: skip dEQP-GLES3.functional.fbo.msaa.2_samples.stencil_index8
- freedreno/a6xx: un-open-code PC_PRIMITIVE_CNTL_1.PSIZE
- freedreno/a6xx: fix binning pass vs. xfb
- freedreno/a6xx: do streamout only in binning pass
- freedreno/ir3: drop unused param
- freedreno/ir3: handle multi component alu src when propagating shifts
- freedreno: update registers
- freedreno/ir3: remove unused ir3_instruction::inout
- freedreno/ir3: track sysval slot for inputs
- freedreno/ir3: don't DCE ij_pix if used for pre-fs-texture-fetch
- freedreno/ir3: add meta instruction for pre-fs texture fetch
- freedreno/ir3: fixup register footprint to account for prefetch
- freedreno/ir3: add dummy bary.f(ei) for pre-fs-fetch
- freedreno/ir3: add pre-dispatch tex fetch to disasm
- freedreno/ir3: force i/j pixel to r0.x
- freedreno/a6xx: add support for pre-fs texture fetch
- turnip: add support for pre-fs texture fetch
- freedreno/ir3: enable pre-fs texture fetch for a6xx

- nir/search: fix the PoT helpers
- freedreno/ir3: rename mul.s/mul.u
- nir: Add a new ALU nir_op_imul24
- nir: add amul instruction
- nir: add address calc related opt rules
- nir: add nir_lower_amul pass
- freedreno/ir3: add rule to generate imad24
- freedreno/ir3: optimize immed 2nd src to mad
- freedreno/ir3: add imul24 opcode
- freedreno/ir3: handle imad24_ir3 case in UBO lowering
- freedreno/ir3: handle scalarized varying inputs
- freedreno/ir3: fixup register footprint fixup
- freedreno/ir3: debug cleanup
- freedreno/ir3: make high regs easier to see in IR dumps
- freedreno/ir3: propagate dest flags for collect/fanin
- freedreno/ir3: treat high vs low reg as conversion
- freedreno/ir3: allow copy-propagate out of fanout
- freedreno/ir3: remove restrictions on const + (abs)/(neg)
- freedreno/ir3: handle the progress case
- freedreno/a6xx: remove some left over dead code
- freedreno/a6xx: cleanup magic registers
- freedreno/a6xx: add a618 support
- freedreno/ir3: fix gpu hang with pre-fs-tex-fetch
- Revert "freedreno/ir3: enable pre-fs texture fetch for a6xx"
- nir/lower_clip: Fix incorrect driver loc for clipdist outputs

Robin Murphy (1):

- egl/gbm: Fix config validation

Rohan Garg (3):

- panfrost: Remove unused argument from panfrost_drm_submit_vs_fs_job()
- panfrost: Jobs must be per context, not per screen
- panfrost: protect access to shared bo cache and transient pool

Roland Scheidegger (4):

- gallivm: use fallback code for mul_hi with llvm >= 7.0
- llvmpipe: fix CALLOC vs. free mismatches
- llvmpipe: increase max texture size to 2GB
- gallivm: Fix saturated signed psub/padd intrinsics on llvm 8

Roman Stratiienko (1):

- lima: Return fence unconditionally

Sagar Ghuge (26):

- intel/eu/gen12: Implement immediate 64 bit constant encoding.
- nir: Add alpha_to_coverage lowering pass
- intel/compiler: Remove emit_alpha_to_coverage workaround from backend
- intel: Add missing entry for brw_nir_lower_alpha_to_coverage in Makefile
- intel/compiler: Add Immediate support for 3 source instruction
- intel/compiler: Set bits according to source file
- intel/compiler: Don't move immediate in register
- intel/compiler: Refactor disassembly of sources in 3src instruction
- intel/isl: Don't reconfigure aux surfaces for MCS
- iris: Initialize CCS to fast clear while using with MCS
- iris: Define MCS_CCS state transitions and usages
- intel/blorp: Use isl_aux_usage_has_mcs instead of comparing
- iris: Get correct resource aux usage for copy
- intel/isl: Support lossless compression with multisamples
- iris: Create resource with aux_usage MCS_CCS
- genxml/gen12: Add Stencil Buffer Resolve Enable bit
- intel/blorp: Assign correct view while clearing depth stencil
- intel/blorp: Add helper function for stencil buffer resolve
- intel: Track stencil aux usage on Gen12+
- intel/blorp: Set stencil resolve enable bit
- iris: Resolve stencil buffer lossless compression with WM_HZ_OP packet
- iris: Prepare stencil resource before clear depth stencil
- iris: Prepare depth resource if clear_depth enable
- iris: Prepare resources before stencil blit operation
- iris: Resolve stencil resource prior to copy or used by CPU
- intel/isl: Allow stencil buffer to support compression on Gen12+

Samuel Iglesias Gonsálvez (26):

- spirv: check support for SPV_KHR_float_controls capabilities
- spirv/nir: keep track of SPV_KHR_float_controls execution modes
- nir: add auxiliary functions to detect if a mode is enabled
- nir: add support for flushing to zero denorm constants
- util: add softfloat functions to operate with doubles and floats
- util: add float to float16 conversions with RTZ and RTNE

- util: add fp64 -> fp32 conversion support for RTNE and RTZ rounding modes
- nir: add support for round to zero rounding mode to nir_op_f2f32
- nir: mind rounding mode on fadd, fsub, fmul and fma opcodes
- nir/opcodes: make sure f2f16_rtz and f2f16_rtne behavior is not overriden by the float controls execution mode
- nir/constant_expressions: mind rounding mode converting from float to float16 destinations
- nir/algebraic: disable inexact optimizations depending on float controls execution mode
- nir: fix denorms in unpack_half_1x16()
- nir: fix denorm flush-to-zero in sqrt's lowering at nir_lower_double_ops
- nir: fix fmin/fmax support for doubles
- intel/nir: do not apply the fsin and fcos trig workarounds for consts
- i965/fs/nir: add nir_op_unpack_half_2x16_split_*_flush_to_zero
- i965/fs/generator: refactor rounding mode helper in preparation for float controls
- i965/fs/generator: add new opcode to set float controls modes in control register
- i965/fs: add emit_shader_float_controls_execution_mode() and aux functions
- i965/fs: set rounding mode when emitting fadd, fmul and ffma instructions
- i965/fs: set rounding mode when emitting nir_op_f2f32 or nir_op_f2f16
- i965/fs: add support for shader float control to remove_extra_rounding_modes()
- anv: enable VK_KHR_shader_float_controls and SPV_KHR_float_controls
- docs/relnotes: add support for VK_KHR_shader_float_controls on Intel
- nir/algebraic: refactor inexact opcode restrictions

Samuel Pitoiset (136):

- radv/gfx10: tidy up gfx10_format_table.py
- radv/gfx10: hardcode some depth+stencil formats in the format table
- radv: allow to enable VK_AMD_shader_ballot only on GFX8+
- radv: add a new debug option called RADV_DEBUG=noshaderballot
- radv: force enable VK_AMD_shader_ballot for Wolfenstein Youngblood
- radv: implement VK_AMD_shader_core_properties2
- ac: fix exclusive scans on GFX8-GFX9
- ac,radv,radeonsi: remove LLVM 7 support
- gitlab-ci: bump LLVM to 8 for meson-vulkan and meson-clover
- radv/gfx10: don't initialize VGT_INSTANCE_STEP_RATE_0
- radv/gfx10: do not use NGG with NAVI14
- radv: fix getting the index type size for uint8_t
- radv: add radv_process_depth_image_layer() helper
- radv: add mipmaps support for decompress/resummarize
- radv: decompress mipmapped depth/stencil images during transitions

- radv: allocate metadata space for mipmapped depth/stencil images
- radv: add mipmap support for the TC-compat zrange bug
- radv: add mipmap support for the clear depth/stencil values
- ac: drop llvm8 from some load/store helpers
- ac: add has_clear_state to ac_gpu_info
- ac: add has_distributed_tess to ac_gpu_info
- ac: add has_dcc_constant_encode to ac_gpu_info
- ac: add has_rbplus to ac_gpu_info
- ac: add has_load_ctx_reg_pkt to ac_gpu_info
- ac: add has_out_of_order_rast to ac_gpu_info
- ac: add cpdma_prefetch_writes_memory to ac_gpu_info
- ac: add has_gfx9_scissor_bug to ac_gpu_info
- ac: add has_tc_compat_zrange_bug to ac_gpu_info
- ac: add rbplus_allowed to ac_gpu_info
- ac: add has_msaa_sample_loc_bug to ac_gpu_info
- ac: add has_ls_vgpr_init_bug to ac_gpu_info
- radv: make use of has_ls_vgpr_init_bug
- radv/gfx10: compute the LDS size for exporting PrimID for VS
- ac: import linear/perspective PS input parameters from radv/radeonsi
- ac: drop now useless lookup_interp_param from ABI
- radv: gather info about PS inputs in the shader info pass
- radv: move lowering PS inputs/outputs at the right place
- radv: remove some unused fields from radv_shader_context
- radv: remove unused shader_info parameter in ac_compile_llvm_module()
- radv: remove useless ac_llvm_util.h include from the WSI code
- radv: remove radv_init_llvm_target() helper
- radv: replace ac_nir_build_if by ac_build_ifcc
- radv: move setting can_discard to ac_fill_shader_info()
- radv: keep a pointer to a NIR shader into radv_shader_context
- nir: do not assume that the result of fexp2(a) is always an integral
- radv/gfx10: always set ballot_mask_bits to 64
- radv: merge radv_shader_variant_info into radv_shader_info
- radv: move ac_fill_shader_info() to radv_nir_shader_info_pass()
- radv: gather clip/cull distances in the shader info pass
- radv: gather pointsize in the shader info pass
- radv: gather viewport in the shader info pass

- radv: gather layer in the shader info pass
- radv: gather primitive ID in the shader info pass
- radv: calculate the GSVS vertex size in the shader info pass
- radv: calculate esgs_itemsize in the shader info pass
- radv/gfx10: account for the subpass view for the NGG GS storage
- radv/gfx10: make use the output usage mask when exporting NGG GS params
- radv/gfx10: determine the number of vertices per primitive for TES
- radv: do not pass all compiler options to the shader info pass
- radv: fill shader info for all stages in the pipeline
- radv: store GFX9 GS state as part of the shader info
- radv: store GFX10 NGG state as part of the shader info
- radv: store the ESGS ring size as part of gfx10_ngg_info
- radv: calculate GFX9 GS and GFX10 NGG states before compiling shader variants
- radv/gfx10: declare a LDS symbol for the NGG emit space
- radv: fix allocating number of user sgprs if streamout is used
- radv/winsys: add support for GS and OA domains
- radv/gfx10: add an option to switch from legacy to NGG streamout
- radv/gfx10: implement NGG streamout begin/end functions
- radv/gfx10: allocate GDS/OA buffer objects for NGG streamout
- radv/gfx10: adjust the GS NGG scratch size for streamout
- radv/gfx10: unconditionally declare scratch space for NGG streamout without GS
- radv/gfx10: adjust the LDS size for VS/TES NGG streamout
- radv/gfx10: fix unnecessary LDS overallocation for NGG GS
- radv/gfx10: compute the correct buffer size for NGG streamout
- radv/gfx10: gather GS output for VS as NGG
- radv/gfx10: enable NGG_WAVE_ID_EN for NGG streamout
- radv/gfx10: make GDS idle when leaving the IB
- radv/gfx10: make sure to wait for idle before clearing GDS
- radv/gfx10: implement NGG streamout
- radv/gfx10: disable unsupported transform feedback features for NGG
- radv: fix writing depth/stencil clear values to image
- radv: fix loading 64-bit GS inputs
- radv/gfx10: fix VK_KHR_pipeline_executable_properties with NGG GS
- radv/gfx10: add radv_device::use_ngg
- radv/gfx10: add missing counter buffer to the BO list
- radv/gfx10: fix storing/loading NGG stream outputs for VS and TES

- radv/gfx10: use the component mask when storing/loading NGG stream outputs
- radv/gfx10: fix storing/loading NGG stream outputs for GS
- radv/gfx10: fix NGG streamout with triangle strips for VS
- radv: rework the slow depthstencil clear to write depth from PS
- Revert "radv: disable viewport clamping even if FS doesn't write Z"
- radv: fix build
- radv/gfx10: fix the ESGS ring size symbol
- radv: enable lower_fmod for the LLVM path
- ac/nir: remove unused code for nir_op_{fmod,frem}
- radv: implement VK_KHR_shader_clock
- drirc: enable vk_x11_override_min_image_count for DOOM
- radv: bump minTexelBufferOffsetAlignment to 4
- radv: get the device name from radeon_info::name
- radv: sync before resetting query pools if timestamps have been written
- radv: use a compute shader for copying timestamp query results
- radv: fix DCC fast clear code for intensity formats
- radv: rename VK_KHR_shader_float16_int8 structs/constants
- Revert "radv: do not emit PKT3_CONTEXT_CONTROL with AMDGPU 3.6.0+"
- radv: fix DCC fast clear code for intensity formats (correctly)
- ac/llvm: add ac_build_canonicalize() helper
- ac/llvm: add AC_FLOAT_MODE_ROUND_TO_ZERO
- ac/llvm: force fneg/fabs to flush denorms to zero if requested
- radv: implement VK_KHR_shader_float_controls
- radv: enable VK_KHR_shader_float_controls on GFX6-GFX7
- radv: do not print useless descriptors info in hang reports
- radv: print which ring is dumped in hang reports
- radv: dump trace files earlier if a GPU hang is detected
- radv: do not dump descriptors twice in hang reports
- radv: advertise VK_KHR_spirv_1_4
- ac/llvm: fix ac_to_integer_type() for 32-bit const addr space pointers
- radv: fix updating bound fast ds clear values with different aspects
- radv: do not create meta pipelines with 16 samples
- radv: add an assertion in radv_gfx10_compute_bin_size()
- radv: do not emit rbplus if attachments are undefined
- radv/gfx10: re-enable fast depth/stencil clears with separate aspects
- radv/gfx10: fix 3D images

- radv: fix vkUpdateDescriptorSets with inline uniform blocks
- radv: fix a performance regression with graphics depth/stencil clears
- radv: compute the number of records correctly for vertex buffers
- radv: fix VK_KHR_shader_float_controls dependency on GFX6-7
- radv: enable fast depth/stencil clears with separate aspects on GFX8
- radv: fix OpQuantizeToF16 for NaN on GFX6-7
- radv: fix dumping SPIR-V into hang reports
- radv: move nomemorycache debug option at the right palce
- radv: fix perftest options
- radv: fix compute pipeline keys when optimizations are disabled
- radv: fix enabling sample shading with SampleID/SamplePosition
- radv/gfx10: fix implementation of exclusive scans
- ac/nir: fix out-of-bound access when loading constants from global

Sergii Romantsov (4):

- intel/dri: finish proper glthread
- nir/large_constants: more careful data copying
- nir/large_constants: pass after lowering copy_deref
- meta: leak of shader program when decompressing tex-images

Stephen Barber (1):

- nouveau: add idep_nir_headers as dep for libnouveau

Tapani Pälli (23):

- util: fix os_create_anonymous_file on android
- iris/android: fix build and link with libmesa_intel_perf
- egl: reset blob cache set/get functions on terminate
- intel/genxml: generate pack files for gen12 on android builds
- intel/isl: build android libmesa_isl for gen12
- iris: build android libmesa_iris for gen12
- anv: build libanv for gen12 in android build
- i965: initialize bo_reuse when creating brw_bufmgr
- iris: use driconf for 'bo_reuse' parameter
- android: fix linking issues with liblog
- iris: close screen fd on iris_destroy_screen
- egl: check for NULL value like eglGetSyncAttribKHR does
- iris: disable aux on first get_param if not created with aux
- mesa/st: calculate texture size based on EGLImage miplevel
- anv/android: fix images created with external format support

- i965: setup sized internalformat for MESA_FORMAT_R10G10B10A2_UNORM
- mesa: add [Program]Uniform*64ARB display list support
- mesa: enable ARB_gpu_shader_int64 in compat profile
- Revert "egl: implement new functions from EGL_EXT_image_flush_external"
- Revert "egl: handle EGL_IMAGE_EXTERNAL_FLUSH_EXT"
- Revert "st/dri: add support for EGL_EXT_image_flush_external"
- Revert "st/dri: assume external consumers of back buffers can write to the buffers"
- Revert "dri_interface: add interface for EGL_EXT_image_flush_external"

Thomas Hellstrom (2):

- svga: Fix banded DMA upload unmap
- winsys/svga: Limit the maximum DMA hardware buffer size

Thong Thai (2):

- Revert "radeonsi: don't emit PKT3_CONTEXT_CONTROL on amdgpu"
- radeonsi: add JPEG decode support for VCN 2.0 devices

Timothy Arceri (35):

- radeonsi/nir: fix number of used samplers
- util/disk_cache: bump thread count assigned to disk cache queue
- util/u_queue: track job size and limit the size of queue growth
- util/disk_cache: make use of the total job size limiting feature
- radeonsi/nir: lower load constants to scalar
- glsl: fix crash compiling bindless samplers inside unnamed UBOs
- nir: fix nir_variable_data packing
- nir: improve nir_variable packing
- glsl: remove propagate_invariance() call from the linker
- radv: get topology from pipeline key rather than VkGraphicsPipelineCreateInfo
- radv: add debug option to turn off in memory cache
- radv: add radv_create_shaders() to radv_shader.h
- radv: add radv_secure_compile_type enum
- radv: add some new members to radv device and instance for secure compile
- radv: add radv_device_use_secure_compile() helper
- radv: allow the secure process to read and write from disk cache
- radv: for secure compile exit early from radv_shader_variant_create()
- radv: add radv_secure_compile()
- radv: a support for a secure compile fork at device creation
- radv: enable secure compile support
- util: remove LIST_INITHEAD macro

- util: remove LIST_ADDTAIL macro
- util: remove LIST_ADD macro
- util: remove LIST_REPLACE macro
- util: remove LIST_DELINIT macro
- util: remove LIST_DEL macro
- util: rename list_empty() to list_is_empty()
- util: remove LIST_IS_EMPTY macro
- radv: allow select() calls in secure compile
- radv: add radv_sc_read() helper
- radv: make use of radv_sc_read()
- radv: add some infrastructure for fresh forks for each secure compile
- radv: add a secure_compile_open_fifo_fds() helper
- radv: create a fresh fork for each pipeline compile
- glsl/nir: iterate the system values list when adding varyings

Timur Kristóf (48):

- st/nine: Properly initialize GLSL types for NIR shaders.
- nir: Carve out nir_lower_samplers from GLSL code.
- tgsi_to_nir: Remove dependency on libglsl.
- amd/common: Move ac_export_mrt_z to ac_llvm_build.
- amd/common: Extract some helper functions to ac_shader_util.
- amd/common: Add num_shared_vgprs to ac_shader_config for GFX10.
- radv: Set shared VGPR count in radv_postprocess_config.
- amd/common: Introduce ac_get_fs_input_vgpr_cnt.
- radv: Add debug option to dump meta shaders.
- radv: Fix L2 cache rinse programming.
- amd: Move all amd/common code that depends on LLVM to amd/llvm.
- aco: Set +wavefrontsize64 for LLVM disassembler in GFX10 wave64 mode.
- aco: Add missing GFX10 specific fields and some README notes.
- aco: Support GFX10 SMEM in aco_assembler.
- aco: Support GFX10 VINTRP in aco_assembler.
- aco: Support GFX10 DS in aco_assembler.
- aco: Support GFX10 MUBUF in aco_assembler.
- amd/common: Add extern “C” to some headers that were missing it.
- aco: Link ACO with amd/common.
- aco: Support GFX10 MTBUF in aco_assembler.
- aco: Support GFX10 MIMG and GFX9 D16 in aco_assembler.

- aco: Fix GFX9 FLAT, SCRATCH, GLOBAL instructions, add GFX10 support.
- aco: Support GFX10 EXP in aco_assembler.
- aco: Support GFX10 VOP3 and VOP1 as VOP3 in aco_assembler.
- aco: Set GFX10 DLC bit properly.
- aco: Use ac_get_sampler_dim, delete duplicate code.
- aco: Set GFX10 dimensionality on the instructions that need it.
- aco: Support subvector loops in aco_assembler.
- aco: Fix VS input VGPRs on GFX10.
- aco: Fix s_dcache_wb on GFX10.
- aco: Add extra assertion for number of FS input VGPRs.
- aco: Clean up usages of PhysReg::reg from aco_assembler.
- aco/gfx10: Wait for pending SMEM stores before loads
- aco/gfx10: Fix PS exports for SPI_SHADER_32_AR.
- aco/gfx10: Update constant addresses in fix_branches_gfx10.
- aco/gfx10: Add notes about some GFX10 hazards.
- aco/gfx10: Mitigate VcmpxPermlaneHazard.
- aco/gfx10: Mitigate VcmpxExecWARHazard.
- aco/gfx10: Mitigate SMEMtoVectorWriteHazard.
- aco/gfx10: Mitigate LdsBranchVmemWARHazard.
- aco/gfx10: Fix mitigation of VMEMtoScalarWriteHazard.
- aco: Refactor hazard mitigations, separate pass for GFX10.
- st/nine: Fix build with -Werror=empty-body
- st/nine: Fix unused variable warnings in release build.
- aco: Implement subgroup shuffle in GFX10 wave64 mode.
- aco: Introduce vgpr_limit to keep track of available VGPRs.
- radv: Enable ACO on Navi.
- ac: Handle invalid GFX10 format correctly in ac_get_tbuffer_format.

Tomeu Vizoso (19):

- panfrost/ci: Use Volt-based runner for dEQP tests
- panfrost/ci: Print bootstrap log
- panfrost/ci: Build kernel with CONFIG_DETECT_HUNG_TASK
- panfrost/ci: Install qemu-arm-static into chroot
- panfrost/ci: Print load stats
- panfrost/ci: Print only regressions
- panfrost/ci: Re-add support for armhf
- panfrost/ci: Use special runner for LAVA jobs

- panfrost/ci: Increase timeouts
- panfrost/ci: Run dEQP with the surfaceless platform
- panfrost/ci: Update kernel to 5.3-rc8
- panfrost/ci: Use releases for Volt dEQP
- gitlab-ci: Run dEQP on devices with Panfrost
- gitlab-ci: Move LAVA-related files into top-level ci dir
- gitlab-ci/lava: Fix image to use in test jobs
- gitlab-ci/lava: Use files to list tests to skip
- gitlab-ci/lava: Test Lima driver with dEQP
- panfrost: Keep track of active BOs
- gitlab-ci: Update kernel for LAVA jobs to 5.4-rc4

Urja Rannikko (1):

- panfrost: allocate bo for occlusion query results

Vasily Khoruzhick (35):

- lima/ppir: refactor const lowering
- lima/ppir: clone ld_{uni,tex,var} into each block
- lima/ppir: add support for unconditional branches and condition negation
- lima/ppir: set write mask for texture loads if dest is reg
- lima/ppir: fix ordering deps
- lima/ppir: add write after read deps for registers
- lima/ppir: add dummy op
- lima/ppir: create ppir block for each corresponding NIR block
- lima/ppir: turn store_color into ALU node
- lima/ppir: validate shader outputs
- lima/ppir: add better liveness analysis
- lima/ppir: add control flow support
- lima/ppir: print register index and components number for spilled register
- lima: fix texture descriptor issues
- lima/ppir: add common helper for creating movs
- lima/ppir: don't assume that load coords gets value from register
- lima/ppir: clone uniforms and load_coords into each successor
- nir: allow specifying filter callback in lower_alu_to_scalar
- lima/ppir: don't lower vector {b,f}csel to scalar if condition is scalar
- lima/ppir: don't lower phis to scalar
- lima/gpir: lower fceil
- lima/gpir: fix warning in gpir disassembler

- lima: run opt_algebraic between int_to_float and boot_to_float for vs
- lima/ppir: drop fge/flt/feq/fne options
- lima: set .out_sync field of req in lima_submit_start()
- lima: add standalone disassembler with primitive MBS parser
- lima: use 0 to poll if BO is busy in lima_bo_wait()
- lima: implement BO cache
- lima/ppir: don't attempt to clone tex coords if it's not varying
- lima/ppir: add node dependency types
- lima/ppir: add support for indirect load of uniforms and varyings
- lima/ppir: add NIR pass to split varying loads
- lima: set uniforms_address lower bits properly
- lima/ppir: don't clone texture loads
- lima: fix PP stack size

Vinson Lee (7):

- glx: Fix up glXQueryGLXPbufferSGIX on macOS.
- swr: Fix build with llvm-9.0 again.
- travis: Fail build if any command in if statement fails.
- util: Define strchrnul on macOS.
- swr: Fix make_unique build error.
- scons: Add coroutines component to build.
- meson: Add coroutines component to llvmpipe build.

Wladimir J. van der Laan (1):

- etnaviv: GC7000: Texture descriptors

Yevhenii Kolesnikov (2):

- glsl: Enable textureSize for samplerExternalOES
- meson: Fix linkage of libgallium_nine with libgalliumvl

Zebediah Figura (1):

- Revert "draw: revert using correct order for prim decomposition."

Zhaowei Yuan (1):

- broadcom/vc4: Expand width of dst surface

Zhu, James (1):

- radeon: Fix mjpeg issue for ARCTURUS

nia (1):

- loader: include limits.h for PATH_MAX

pal1000 (3):

- scons/windows: Support build with LLVM 9.

- scons: Fix MSYS2 Mingw-w64 build.
- scons/windows: Enable compute shaders when possible.

renchenglei (1):

- egl/android: Enable HAL_PIXEL_FORMAT_RGBA_1010102 format

4.30 Mesa 19.2.7 Release Notes / 2019-12-04

Mesa 19.2.7 is a bug fix release which fixes bugs found since the 19.2.6 release.

Mesa 19.2.7 implements the OpenGL 4.5 API, but the version reported by glGetString(GL_VERSION) or glGetIntegerv(GL_MAJOR_VERSION) / glGetIntegerv(GL_MINOR_VERSION) depends on the particular driver being used. Some drivers don't support all the features required in OpenGL 4.5. OpenGL 4.5 is **only** available if requested at context creation. Compatibility contexts may report a lower version depending on each driver.

Mesa 19.2.7 implements the Vulkan 1.1 API, but the version reported by the apiVersion property of the VkPhysicalDeviceProperties struct depends on the particular driver being used.

4.30.1 SHA256 checksum

```
e3799fb7896fd9ed2f90f651fb907b95cdebfbd494968ff116e6bf1be143579e  mesa-19.2.7.tar.xz
```

4.30.2 New features

- None

4.30.3 Bug fixes

- ld.lld: error: duplicate symbol (mesa-19.3.0-rc1)
- triangle strip clipping with GL_FIRST_VERTEX_CONVENTION causes wrong vertex's attribute to be broadcasted for flat interpolation
- [bisected][regression][g45,g965,ilk] piglit arb_fragment_program kil failures

4.30.4 Changes

Bas Nieuwenhuizen (2):

- radv: Allocate cmdbuffer space for buffer marker write.
- radv: Unify max_descriptor_set_size.

Boris Brezillon (1):

- gallium: Fix the ->set_damage_region() implementation

Ian Romanick (1):

- intel/fs: Disable conditional discard optimization on Gen4 and Gen5

Jason Ekstrand (1):

- anv: Set up SBE_SWIZ properly for gl_Viewport

Jonathan Gray (2):

- winsys/amdgpu: avoid double simple_mtx_unlock()
- i965: update Makefile.sources for perf changes

Rhys Perry (1):

- radv: set writes_memory for global memory stores/atomics

Samuel Pitoiset (3):

- radv: fix enabling sample shading with SampleID/SamplePosition
- radv/gfx10: fix implementation of exclusive scans
- radv: fix compute pipeline keys when optimizations are disabled

Yevhenii Kolesnikov (1):

- meson: Fix linkage of libgallium_nine with libgalliumvl

Zebediah Figura (1):

- Revert "draw: revert using correct order for prim decomposition."

4.31 Mesa 19.2.6 Release Notes / 2019-11-21

Mesa 19.2.6 is a bug fix release which fixes bugs found since the 19.2.5 release.

Mesa 19.2.6 implements the OpenGL 4.5 API, but the version reported by glGetString(GL_VERSION) or glGetIntegerv(GL_MAJOR_VERSION) / glGetIntegerv(GL_MINOR_VERSION) depends on the particular driver being used. Some drivers don't support all the features required in OpenGL 4.5. OpenGL 4.5 is **only** available if requested at context creation. Compatibility contexts may report a lower version depending on each driver.

Mesa 19.2.6 implements the Vulkan 1.1 API, but the version reported by the apiVersion property of the VkPhysicalDeviceProperties struct depends on the particular driver being used.

4.31.1 SHA256 checksum

```
9d7b24fa60c82db34788196450042a55ce6cb2d70c7a8d5c31401619b6907797  mesa-19.2.6.tar.xz
```

4.31.2 New features

- None

4.31.3 Bug fixes

- glesv2.pc is not built since fafd20f67dec9f589
- textureSize(samplerExternalOES, int) missing in desktop mesa 19.1.7 implementation
- [19.2.5] lp_bld_misc: broken #if PIPE_ARCH_LITTLE_ENDIAN on ppc64l

4.31.4 Changes

Alejandro Piñeiro (1):

- v3d: adds an extra MOV for any sig.ld*

Dave Airlie (1):

- llvmpipe/ppc: fix if/ifdef confusion in backport.

Dylan Baker (2):

- docs/relnotes/19.2.5: Add SHA256 sum
- meson: generate .pc files for gles and gles2 with old glvnd

Eric Engestrom (1):

- vulkan: delete typo'd header

Hyunjun Ko (1):

- freedreno/ir3: fix printing output registers of FS.

Jose Maria Casanova Crespo (1):

- v3d: Fix predication with atomic image operations

Yevhenii Kolesnikov (1):

- glsl: Enable textureSize for samplerExternalOES

4.32 Mesa 19.2.5 Release Notes / 2019-11-20

Mesa 19.2.5 is a bug fix release which fixes bugs found since the 19.2.4 release.

Mesa 19.2.5 implements the OpenGL 4.5 API, but the version reported by glGetString(GL_VERSION) or glGetIntegerv(GL_MAJOR_VERSION) / glGetIntegerv(GL_MINOR_VERSION) depends on the particular driver being used. Some drivers don't support all the features required in OpenGL 4.5. OpenGL 4.5 is **only** available if requested at context creation. Compatibility contexts may report a lower version depending on each driver.

Mesa 19.2.5 implements the Vulkan 1.1 API, but the version reported by the apiVersion property of the VkPhysicalDeviceProperties struct depends on the particular driver being used.

4.32.1 SHA256 checksum

```
3d010a366b28d10bdd71e32091d8684baf1522e6466c5c5703667091b2108c8b  mesa-19.2.5.tar.xz
```

4.32.2 New features

- None

4.32.3 Bug fixes

- HSW. Tropico 6 and SuperTuxKart have shadows flickering
- glxgears segfaults on POWER / Xvnc

- Cannot start Civ6 with AMD GPU on Linux

4.32.4 Changes

Ben Crocker (1):

- llvmpipe: use ppc64le/ppc64 Large code model for JIT-compiled shaders

Brian Paul (1):

- Call shmget() with permission 0600 instead of 0777

Caio Marcelo de Oliveira Filho (1):

- spirv: Don't leak GS initialization to other stages

Danylo Piliaiev (1):

- i965: Unify CC_STATE and BLEND_STATE atoms on Haswell as a workaround

Dylan Baker (2):

- docs: Add SHA256 sum for for 19.2.4
- cherry-ignore: Update for 19.2.4 cycle

Eric Engestrom (1):

- egl: fix _EGL_NATIVE_PLATFORM fallback

Ian Romanick (2):

- nir/algebraic: Add the ability to mark a replacement as exact
- nir/algebraic: Mark other comparison exact when removing a == a

Illia Iorin (1):

- mesa/main: Ignore filter state for MS texture completeness

Jason Ekstrand (1):

- anv: Stop bounds-checking pushed UBOs

Lepton Wu (1):

- gallium: dri2: Use index as plane number.

Lionel Landwerlin (3):

- anv: invalidate file descriptor of semaphore sync fd at vkQueueSubmit
- anv: remove list items on batch fini
- anv/wsi: signal the semaphore in the acquireNextImage

Marek Olšák (3):

- st/mesa: fix Sanctuary and Tropics by disabling ARB_gpu_shader5 for them
- tgsi_to_nir: fix masked out image loads
- tgsi_to_nir: handle PIPE_FORMAT_NONE in image opcodes

Paulo Zanoni (1):

- intel/compiler: fix nir_op_{i,u}*32 on ICL

Pierre-Eric Pelloux-Prayer (3):

- radeonsi: disable sdma for gfx10
- radeonsi: tell the shader disk cache what IR is used
- radeonsi: fix shader disk cache key

4.33 Mesa 19.2.4 Release Notes / 2019-11-13

Mesa 19.2.4 is an emergency bug fix release to fix on ciritcal bug in 19.2.3.

Mesa 19.2.4 implements the OpenGL 4.5 API, but the version reported by glGetString(GL_VERSION) or glGetIntegerv(GL_MAJOR_VERSION) / glGetIntegerv(GL_MINOR_VERSION) depends on the particular driver being used. Some drivers don't support all the features required in OpenGL 4.5. OpenGL 4.5 is **only** available if requested at context creation. Compatibility contexts may report a lower version depending on each driver.

Mesa 19.2.4 implements the Vulkan 1.1 API, but the version reported by the apiVersion property of the VkPhysicalDeviceProperties struct depends on the particular driver being used.

4.33.1 SHA256 checksum

```
09000a0f7dbbd82e193b81a8f1bf0c118eab7ca975c0329181968596e548e30f  mesa-19.2.4.tar.xz
```

4.33.2 New features

- None

4.33.3 Bug fixes

- Dirt Rally: Menu system doesn't show up with Mesa 19.2.3

4.33.4 Changes

Lionel Landwerlin (1):

- mesa: check framebuffer completeness only after state update

4.34 Mesa 19.2.3 Release Notes / 2019-11-06

Mesa 19.2.3 is a bug fix release which fixes bugs found since the 19.2.2 release.

Mesa 19.2.3 implements the OpenGL 4.5 API, but the version reported by glGetString(GL_VERSION) or glGetIntegerv(GL_MAJOR_VERSION) / glGetIntegerv(GL_MINOR_VERSION) depends on the particular driver being used. Some drivers don't support all the features required in OpenGL 4.5. OpenGL 4.5 is **only** available if requested at context creation. Compatibility contexts may report a lower version depending on each driver.

Mesa 19.2.3 implements the Vulkan 1.1 API, but the version reported by the apiVersion property of the VkPhysicalDeviceProperties struct depends on the particular driver being used.

4.34.1 SHA256 checksum

```
5ee6e42504fe41dcc9a6eba26982656a675b2550a640946f463927ed7f1c5047  mesa-19.2.3.tar.xz
```

4.34.2 New features

- None

4.34.3 Bug fixes

- 19.2.2 fails mesa:util / timespec test on x86
- Objects leaving trails in Firefox with antialias and preserveDrawingBuffer in three.js WebGLRednerer with mesa 19.2
- glLinkProgram crash when using gcc-9 -O3 -flto due to use of uninitialised value

4.34.4 Changes

Bas Nieuwenhuizen (4):

- radv: Fix timeout handling in syncobj wait.
- radv: Remove _mesa_locale_init/fini calls.
- turnip: Remove _mesa_locale_init/fini calls.
- anv: Remove _mesa_locale_init/fini calls.

Caio Marcelo de Oliveira Filho (1):

- anv: Fix output of INTEL_DEBUG=bat for chained batches

Danylo Piliaiev (1):

- glsl: Initialize all fields of ir_variable in constructor

Dylan Baker (11):

- bin/gen_release_notes.py: fix conditional of bugfix
- bin/gen_release_notes.py: strip '#' from gitlab bugs
- bin/gen_release_notes.py: Return "None" if there are no new features
- bin/post_version.py: Pass version as an argument
- bin/post_version.py: white space fixes
- bin/post_release.py: Add .html to hrefs
- bin/gen_release_notes.py: html escape all external data
- bin/gen_release_notes.py: Add a warning if new features are introduced in a point release
- cherry-ignore: update for 19.2.3 cycle
- nir: correct use of identity check in python
- meson: Add dep_glvnd to egl deps when building with glvnd

Ilia Mirkin (1):

- nv50/ir: mark STORE destination inputs as used

Illia Iorin (1):

- Revert "mesa/main: Fix multisample texture initialize"

Jason Ekstrand (2):

- anv: Fix a potential BO handle leak
- anv/tests: Zero-initialize instances

Jon Turney (2):

- rbug: Fix use of alloca() without #include "c99_alloca.h"
- Fix timespec_from_nsec test for 32-bit time_t

Jonathan Marek (1):

- etnaviv: fix depth bias

Kenneth Graunke (1):

- iris: Fix "Force Zero RTA Index Enable" setting again

Lionel Landwerlin (2):

- anv: fix unwind of vkCreateDevice fail
- mesa: check draw buffer completeness on glClearBufferfi/glClearBufferiv

Marek Olšák (1):

- util/u_queue: skip util_queue_finish if num_threads is 0

Nanley Chery (5):

- anv: Properly allocate aux-tracking space for CCS_E
- intel/blorp: Disable depth testing for slow depth clears
- iris: Clear ::has_hiz when disabling aux
- iris: Don't leak the resource for unsupported modifier
- iris: Disallow incomplete resource creation

Paulo Zanoni (1):

- intel/compiler: remove the operand restriction for src1 on GLK

Pierre-Eric Pelloux-Prayer (1):

- mesa: enable msaa in clear_with_quad if needed

Sagar Ghuge (1):

- intel/blorp: Assign correct view while clearing depth stencil

Samuel Pitoiset (4):

- radv: do not create meta pipelines with 16 samples
- radv: do not emit rbplus if attachments are undefined
- radv/gfx10: fix 3D images
- radv: fix vkUpdateDescriptorSets with inline uniform blocks

Tapani Pälli (1):

- i965: setup sized internalformat for MESA_FORMAT_R10G10B10A2_UNORM

Thomas Hellstrom (2):

- svga: Fix banded DMA upload unmap
- winsys/svga: Limit the maximum DMA hardware buffer size

4.35 Mesa 19.2.2 Release Notes / 2019-10-23

Mesa 19.2.2 is a bug fix release which fixes bugs found since the 19.2.1 release.

Mesa 19.2.2 implements the OpenGL 4.5 API, but the version reported by glGetString(GL_VERSION) or glGetIntegerv(GL_MAJOR_VERSION) / glGetIntegerv(GL_MINOR_VERSION) depends on the particular driver being used. Some drivers don't support all the features required in OpenGL 4.5. OpenGL 4.5 is **only** available if requested at context creation. Compatibility contexts may report a lower version depending on each driver.

Mesa 19.2.2 implements the Vulkan 1.1 API, but the version reported by the apiVersion property of the VkPhysicalDeviceProperties struct depends on the particular driver being used.

4.35.1 SHA256 checksum

```
7e4f0e2678bfcf3b94f533078b514f37943378a4a8604e477c888ec8a2904394  mesa-19.2.2.tar.xz
```

4.35.2 New features

- None

4.35.3 Bug fixes

- Vulkan version of "Middle-earth: Shadow of Mordor" has graphics glitches on RADV driver (part 2)
- Vulkan version of "Middle-earth: Shadow of Mordor" has graphics glitches on RADV driver
- [amdgpu][Navi][llvm] Minimap problem in Nier Automata
- Black ground in Dirt 4
- Superbibles examples crashing Mesa drivers (radeonsi) and causing gpu reset
- [CTS] dEQP-VK.graphicsfuzz.write-red-in-loop-nest crashes
- mesa and libglvnd install the same headers
- Regression: Doom (2016) crashes on Mesa 19.2 and above and Radeon 380 with Vulkan (worked on Mesa 19.1)
- Rocket League displays corruption when the game starts

4.35.4 Changes

Alan Coopersmith (6):

- c99_compat.h: Don't try to use 'restrict' in C++ code
- util: Make Solaris implemention of p_atomic_add work with gcc

- util: Workaround lack of flock on Solaris
- util: Solaris has linux-style pthread_setname_np
- meson: recognize "sunos" as the system name for Solaris
- intel/common: include unistd.h for ioctl() prototype on Solaris

Alejandro Piñeiro (1):

- v3d: take into account prim_counts_offset

Bas Nieuwenhuizen (3):

- radv: Disallow sparse shared images.
- nir/dead_cf: Remove dead control flow after infinite loops.
- radv: Fix single stage constant flush with merged shaders.

Clément Guérin (1):

- radeonsi: enable zerovram for Rocket League

Connor Abbott (2):

- nir/sink: Rewrite loop handling logic
- nir/sink: Don't sink load_ubo to outside of its defining loop

Dylan Baker (1):

- docs: Add SHA256 sum for 19.2.1

Eric Engestrom (7):

- GL: drop symbols mangling support
- meson: rename 'glvnd_missing_pc_files' to 'not glvnd_has_headers_and_pc_files'
- meson: move a couple of include installs around
- meson: split headers one per line
- meson: split Mesa headers as a separate installation
- meson: skip installation of GLVND-provided headers
- util/u_atomic: fix return type of p_atomic_{inc,dec}_return() and p_atomic_{cmp,}xchg()

Ian Romanick (2):

- nir/search: Fix possible NULL dereference in is_fsign
- intel/vec4: Don't try both sources as immediates for DPH

James Xiong (1):

- iris: finish aux import on get_param

Kenneth Graunke (2):

- iris: Properly unreference extra VBOs for draw parameters
- iris: Implement the Gen < 9 tessellation quads workaround

Lepton Wu (1):

- egl/android: Remove our own reference to buffers.

Lionel Landwerlin (3):

- etnaviv: remove variable from global namespace
- anv: fix vkUpdateDescriptorSets with inline uniform blocks
- anv: fix memory leak on device destroy

Lucas Stach (3):

- etnaviv: fix vertex buffer state emission for single stream GPUs
- rbug: fix transmitted texture sizes
- rbug: unwrap index buffer resource

Pierre-Eric Pelloux-Prayer (1):

- mesa: fix invalid target error handling for teximage

Roland Scheidegger (1):

- gallivm: Fix saturated signed psub/padd intrinsics on llvm 8

Samuel Pitoiset (6):

- drirc: enable vk_x11_override_min_image_count for DOOM
- radv: bump minTexelBufferOffsetAlignment to 4
- radv: fix DCC fast clear code for intensity formats
- Revert "radv: do not emit PKT3_CONTEXT_CONTROL with AMDGPU 3.6.0+"
- radv: fix DCC fast clear code for intensity formats (correctly)
- radv: fix updating bound fast ds clear values with different aspects

Timothy Arceri (1):

- glsl: fix crash compiling bindless samplers inside unnamed UBOs

4.36 Mesa 19.1.8 Release Notes / October 21, 2019

Mesa 19.1.8 is a bug fix release which fixes bugs found since the 19.1.7 release.

Mesa 19.1.8 implements the OpenGL 4.5 API, but the version reported by glGetString(GL_VERSION) or glGetIntegerv(GL_MAJOR_VERSION) / glGetIntegerv(GL_MINOR_VERSION) depends on the particular driver being used. Some drivers don't support all the features required in OpenGL 4.5. OpenGL 4.5 is **only** available if requested at context creation. Compatibility contexts may report a lower version depending on each driver.

Mesa 19.1.8 implements the Vulkan 1.1 API, but the version reported by the apiVersion property of the VkPhysicalDeviceProperties struct depends on the particular driver being used.

4.36.1 SHA256 checksums

```
f0fe8289b7d147943bf2fc2147833254881577e8f9ed3d94ddb39e430e711725  mesa-19.1.8.tar.xz
```

4.36.2 New features

None

4.36.3 Bug fixes

- Bug 111236 - VA-API radeonsi SIGSEGV __memmove_avx_unaligned
- Bug 111664 - [Bisected] Segmentation fault on FS shader compilation (mat4x3 * mat4x3)
- Issue #121 - Shared Memeory leakage in XCreateDrawable
- Issue #795 - Xorg does not render with mesa 19.1.7
- Issue #939 - Meson can't find 32-bit libXvMCW in non-standard path
- Issue #944 - Mesa doesn't build with current Scons version (3.1.0)
- Issue #1838 - Mesa installs gl.pc and egl.pc even with libglvnd >= 1.2.0
- Issue #1844 - libXvMC-1.0.12 breaks mesa build
- Issue #1869 - X server does not start with Mesa 19.2.0
- Issue #1872 - [bisected] piglit spec.arb_texture_view.bug-layers-image causes gpu hangs on IVB
- Issue #1878 - meson.build:1447:6: ERROR: Problem encountered: libdrm required for gallium video state-trackers when using x11

4.36.4 Changes

Adam Jackson (1):

- docs: Update bug report URLs for the gitlab migration

Alan Coopersmith (5):

- c99_compat.h: Don't try to use 'restrict' in C++ code
- util: Make Solaris implemention of p_atomic_add work with gcc
- util: Workaround lack of flock on Solaris
- meson: recognize "sunos" as the system name for Solaris
- intel/common: include unistd.h for ioctl() prototype on Solaris

Andreas Gottschling (1):

- drisw: Fix shared memory leak on drawable resize

Andres Gomez (3):

- docs: Add the maximum implemented Vulkan API version in 19.1 rel notes
- docs/features: Update VK_KHR_display_swapchain status
- egl: Remove the 565 pbuffer-only EGL config under X11.

Andrii Simiklit (1):

- glsl: disallow incompatible matrices multiplication

Arcady Goldmints-Orlov (1):

- anv: fix descriptor limits on gen8

Bas Nieuwenhuizen (2):

- tu: Set up glsl types.
- radv: Add workaround for hang in The Surge 2.

Danylo Piliaiev (1):

- st/nine: Ignore D3DSIO_RET if it is the last instruction in a shader

Dylan Baker (5):

- meson: fix logic for generating .pc files with old glvnd
- meson: Try finding libxvmcw via pkg-config before using find_library
- meson: Link xvmc with libxv
- meson: gallium media state trackers require libdrm with x11
- meson: Only error building gallium video without libdrm when the platform is drm

Eric Engestrom (4):

- gl: drop incorrect pkg-config file for glvnd
- meson: re-add incorrect pkg-config files with GLVND for backward compatibility
- util/anon_file: add missing #include
- util/anon_file: const string param

Erik Faye-Lund (1):

- glsl: correct bitcast-helpers

Greg V (1):

- util: add anon_file.h for all memfd/temp file usage

Haihao Xiang (1):

- i965: support AYUV/XYUV for external import only

Hal Gentz (1):

- gallium/osmesa: Fix the inability to set no context as current.

Jason Ekstrand (2):

- nir/repair_ssa: Replace the unreachable check with the phi builder
- intel/fs: Fix fs_inst::flags_read for ANY/ALL predicates

Juan A. Suarez Romero (11):

- docs: add sha256 checksums for 19.1.7
- cherry-ignore: add explicit 19.2 only nominations
- cherry-ignore: add explicit 19.3 only nominations
- Revert "Revert "intel/fs: Move the scalar-region conversion to the generator.""
- cherry-ignore: Revert "gallium: remove PIPE_CAP_TEXTURE_SHADOW_MAP"
- bin/get-pick-list.sh: sha1 commits can be smaller than 8 chars
- cherry-ignore: nir/opt_large_constants: Handle store writemasks
- cherry-ignore: util: added missing headers in anon-file
- cherry-ignore: radv: Fix condition for skipping the continue CS.
- cherry-ignore: Revert "radv: disable viewport clamping even if FS doesn't write Z"
- Update version to 19.1.8

Ken Mays (1):

- haiku: fix Mesa build

Kenneth Graunke (4):

- iris: Initialize ice->state.prim_mode to an invalid value
- intel: Increase Gen11 compute shader scratch IDs to 64.
- iris: Disable CCS_E for 32-bit floating point textures.
- iris: Fix iris_rebind_buffer() for VBOs with non-zero offsets.

Lionel Landwerlin (5):

- anv: gem-stubs: return a valid fd got anv_gem_userptr()
- intel: use proper label for Comet Lake skus
- mesa: don't forget to clear _Layer field on texture unit
- intel: fix subslice computation from topology data
- intel/isl: Set null surface format to R32_UINT

Marek Olšák (1):

- gallium/vl: don't set PIPE_HANDLE_USAGE_EXPLICIT_FLUSH

Matt Turner (1):

- util: Drop preprocessor guards for glibc-2.12

Michel Dänzer (1):

- radeonsi: fix VAAPI segfault due to various bugs

Michel Zou (2):

- scons: add py3 support
- scons: For MinGW use -posix flag.

Paulo Zanoni (1):

- intel/fs: fix SHADER_OPCODE_CLUSTER_BROADCAST for SIMD32

Prodea Alexandru-Liviu (1):

- scons/MSYS2-MinGW-W64: Fix build options defaults Signed-off-by: Prodea Alexandru-Liviu <liviuprodea@yahoo.com> Reviewed-by: Jose Fonseca <jfonseca@vmware.com> Cc: <mesa-stable@lists.freedesktop.org>

Rhys Perry (2):

- radv: always emit a position export in gs copy shaders
- nir/opt_remove_phis: handle phis with no sources

Samuel Iglesias Gonsálvez (1):

- intel/nir: do not apply the fsin and fcos trig workarounds for consts

Stephen Barber (1):

- nouveau: add idep_nir_headers as dep for libnouveau

Tapani Pälli (3):

- iris: close screen fd on iris_destroy_screen

- egl: check for NULL value like eglGetSyncAttribKHR does
- util: fix os_create_anonymous_file on android

pal1000 (2):

- scons/windows: Support build with LLVM 9.
- scons: Fix MSYS2 Mingw-w64 build.

4.37 Mesa 19.2.1 Release Notes / 2019-10-09

Mesa 19.2.1 is a bug fix release which fixes bugs found since the 19.2.0 release.

Mesa 19.2.1 implements the OpenGL 4.5 API, but the version reported by glGetString(GL_VERSION) or glGetIntegerv(GL_MAJOR_VERSION) / glGetIntegerv(GL_MINOR_VERSION) depends on the particular driver being used. Some drivers don't support all the features required in OpenGL 4.5. OpenGL 4.5 is **only** available if requested at context creation. Compatibility contexts may report a lower version depending on each driver.

Mesa 19.2.1 implements the Vulkan 1.1 API, but the version reported by the apiVersion property of the VkPhysicalDeviceProperties struct depends on the particular driver being used.

4.37.1 SHA256 checksum

```
4cc53ca1a8d12c6ff0e5ea44a5213c05c88447ab50d7e28bb350cd29199f01e9  mesa-19.2.1.tar.xz
```

4.37.2 New features

- None

4.37.3 Bug fixes

- meson.build:1447:6: ERROR: Problem encountered: libdrm required for gallium video statetrackers when using x11
- Mesa doesn't build with current Scons version (3.1.0)
- libXvMC-1.0.12 breaks mesa build
- Meson can't find 32-bit libXvMCW in non-standard path
- Mesa installs gl.pc and egl.pc even with libglvnd >= 1.2.0

4.37.4 Changes

Andreas Gottschling (1):

- drisw: Fix shared memory leak on drawable resize

Andres Gomez (1):

- egl: Remove the 565 pbuffer-only EGL config under X11.

Andrii Simiklit (1):

- glsl: disallow incompatible matrices multiplication

Bas Nieuwenhuizen (1):

- radv: Fix condition for skipping the continue CS.

Connor Abbott (1):

- nir/opt_large_constants: Handle store writemasks

Danylo Piliaiev (1):

- st/nine: Ignore D3DSIO_RET if it is the last instruction in a shader

Dylan Baker (9):

- meson: fix logic for generating .pc files with old glvnd
- meson: Try finding libxvmcw via pkg-config before using find_library
- meson: Link xvmc with libxv
- meson: gallium media state trackers require libdrm with x11
- .cherry-ignore: Update for 19.2.1 cycle
- meson: Only error building gallium video without libdrm when the platform is drm
- scripts: Add a gen_release_notes.py script
- release: Add an update_release_calendar.py script
- bin: delete unused releasing scripts

Eric Engestrom (3):

- radv: fix s/load/store/ copy-paste typo
- meson: drop -Wno-foo bug workaround for Meson < 0.46
- meson: add missing idep_nir_headers in iris_gen_libs

Erik Faye-Lund (1):

- glsl: correct bitcast-helpers

Ian Romanick (1):

- nir/range-analysis: Bail if the types don't match

Jason Ekstrand (1):

- intel/fs: Fix fs_inst::flags_read for ANY/ALL predicates

Ken Mays (1):

- haiku: fix Mesa build

Kenneth Graunke (2):

- iris: Disable CCS_E for 32-bit floating point textures.
- iris: Fix iris_rebind_buffer() for VBOs with non-zero offsets.

Lionel Landwerlin (6):

- anv: gem-stubs: return a valid fd got anv_gem_userptr()
- intel: use proper label for Comet Lake skus
- mesa: don't forget to clear _Layer field on texture unit
- intel: fix topology query

- intel: fix subslice computation from topology data
- intel/isl: Set null surface format to R32_UINT

Marek Olšák (7):

- gallium/vl: don't set PIPE_HANDLE_USAGE_EXPLICIT_FLUSH
- gallium: extend resource_get_param to be as capable as resource_get_handle
- radeonsi/gfx10: fix L2 cache rinse programming
- ac: fix incorrect vram_size reported by the kernel
- ac: fix num_good_cu_per_sh for harvested chips
- ac: add radeon_info::tcc_harvested
- radeonsi/gfx10: fix corruption for chips with harvested TCCs

Mauro Rossi (1):

- android: compiler/nir: build nir_divergence_analysis.c

Michel Dänzer (1):

- radeonsi: fix VAAPI segfault due to various bugs

Michel Zou (1):

- scons: add py3 support

Prodea Alexandru-Liviu (1):

- scons/MSYS2-MinGW-W64: Fix build options defaults

Rhys Perry (1):

- nir/opt_remove_phis: handle phis with no sources

Stephen Barber (1):

- nouveau: add idep_nir_headers as dep for libnouveau

Tapani Pälli (2):

- iris: disable aux on first get_param if not created with aux
- anv/android: fix images created with external format support

pal1000 (2):

- scons: Fix MSYS2 Mingw-w64 build.
- scons/windows: Support build with LLVM 9.

4.38 Mesa 19.2.0 Release Notes / 2019.09.25

Mesa 19.2.0 is a new development release. People who are concerned with stability and reliability should stick with a previous release or wait for Mesa 19.2.1.

Mesa 19.2.0 implements the OpenGL 4.5 API, but the version reported by glGetString(GL_VERSION) or glGetIntegerv(GL_MAJOR_VERSION) / glGetIntegerv(GL_MINOR_VERSION) depends on the particular driver being used. Some drivers don't support all the features required in OpenGL 4.5. OpenGL 4.5 is **only** available if requested at context creation. Compatibility contexts may report a lower version depending on each driver.

Mesa 19.2.0 implements the Vulkan 1.1 API, but the version reported by the apiVersion property of the VkPhysicalDeviceProperties struct depends on the particular driver being used.

4.38.1 SHA256 checksums

```
b060caa2a00f856431160ff7377d0e8f58f2aa48c16ee5a9e265ebdccb10852a  mesa-19.2.0.tar.xz
```

4.38.2 New features

- GL_ARB_post_depth_coverage on radeonsi (Navi)
- GL_ARB_seamless_cubemap_per_texture on etnaviv (if GPU supports SEAMLESS_CUBE_MAP)
- GL_EXT_shader_image_load_store on radeonsi (with LLVM >= 10)
- GL_EXT_shader_samples_identical on iris and radeonsi (if using NIR)
- GL_EXT_texture_shadow_lod on i965, iris
- EGL_EXT_platform_device
- VK_AMD_buffer_marker on radv
- VK_EXT_index_type_uint8 on radv
- VK_EXT_post_depth_coverage on radv
- VK_EXT_queue_family_foreign on radv
- VK_EXT_sample_locations on radv
- VK_EXT_shader_demote_to_helper_invocation on Intel.
- VK_KHR_depth_stencil_resolve on radv
- VK_KHR_imageless_framebuffer on radv
- VK_KHR_shader_atomic_int64 on radv
- VK_KHR_uniform_buffer_standard_layout on radv

4.38.3 Bug fixes

- Bug 103674 - u_queue.c:173:7: error: implicit declaration of function 'timespec_get' is invalid in C99
- Bug 104395 - [CTS] GTF-GL46.gtf32.GL3Tests.packed_pixels.packed_pixels tests fail on 32bit Mesa
- Bug 110765 - ANV regression: Assertion 'pass->attachment_count == framebuffer->attachment_count' failed
- Bug 110814 - KWin compositor crashes on launch
- Bug 111069 - Assertion fails in nir_opt_remove_phis.c during compilation of SPIR-V shader
- Bug 111213 - VA-API nouveau SIGSEGV and asserts
- Bug 111241 - Shadertoy shader causing hang
- Bug 111248 - Navi10 Font rendering issue in Overwatch
- Bug 111271 - Crash in eglMakeCurrent
- Bug 111308 - [Regression, NIR, bisected] Black squares in Unigine Heaven via DXVK

- Bug 111401 - Vulkan overlay layer - async compute not supported, making overlay disappear in Doom
- Bug 111405 - Some infinite 'do{}while' loops lead mesa to an infinite compilation
- Bug 111411 - SPIR-V shader leads to GPU hang, sometimes making machine unstable
- Bug 111414 - [REGRESSION] [BISECTED] Segmentation fault in si_bind_blend_state after removal of the blend state NULL check
- Bug 111467 - WOLF RPG Editor + Gallium Nine Standalone: Rendering issue when using Iris driver
- Bug 111490 - [REGRESSION] [BISECTED] Shadow Tactics: Blades of the Shogun - problems rendering water
- Bug 111493 - In the game The Surge (378540) - textures disappear then appear again when I change the camera angle view
- Bug 111509 - [regression][bisected] piglit.spec.ext_image_dma_buf_import.ext_image_dma_buf_import-export fails on iris
- Bug 111522 - [bisected] Supraland no longer start
- Bug 111529 - EGL_PLATFORM=drm doesn't expose MESA_query_driver extension
- Bug 111552 - Geekbench 5.0 Vulkan compute benchmark fails on Anvil
- Bug 111566 - [REGRESSION] [BISECTED] Large CS workgroup sizes broken in combination with FP64 on Intel.
- Bug 111576 - [bisected] Performance regression in X4:Foundations in 19.2
- Bug 111676 - Tropico 6 apitrace throws error into logs
- Bug 111734 - Geometry shader with double interpolators fails in LLVM

4.38.4 Changes

Adam Jackson (1):

- docs: Update bug report URLs for the gitlab migration

Alex Smith (1):

- radv: Change memory type order for GPUs without dedicated VRAM

Alyssa Rosenzweig (1):

- pan/midgard: Fix writeout combining

Andres Gomez (1):

- docs: Add the maximum implemented Vulkan API version in 19.2 rel notes

Andres Rodriguez (1):

- radv: additional query fixes

Arcady Goldmints-Orlov (1):

- anv: fix descriptor limits on gen8

Bas Nieuwenhuizen (6):

- radv: Use correct vgpr_comp_cnt for VS if both prim_id and instance_id are needed.
- radv: Emit VGT_GS_ONCHIP_CNTL for tess on GFX10.
- radv: Disable NGG for geometry shaders.

- Revert "ac/nir: Lower large indirect variables to scratch"
- tu: Set up glsl types.
- radv: Add workaround for hang in The Surge 2.

Caio Marcelo de Oliveira Filho (2):

- nir/lower_explicit_io: Handle 1 bit loads and stores
- glsl/nir: Avoid overflow when setting max_uniform_location

Connor Abbott (1):

- radv: Call nir_propagate_invariant()

Danylo Piliaiev (3):

- nir/loop_unroll: Prepare loop for unrolling in wrapper_unroll
- nir/loop_analyze: Treat do{}while(false) loops as 0 iterations
- tgsi_to_nir: Translate TGSI_INTERPOLATE_COLOR as INTERP_MODE_NONE

Dave Airlie (2):

- virgl: fix format conversion for recent gallium changes.
- gallivm: fix atomic compare-and-swap

Dave Stevenson (1):

- broadcom/v3d: Allow importing linear BOs with arbitrary offset/stride.

Dylan Baker (9):

- bump version to 19.2-rc2
- nir: Add is_not_negative helper function
- Bump version for rc3
- meson: don't generate file into subdirs
- add patches to be ignored
- Bump version for 19.2.0-rc4
- cherry-ignore: Add patches
- rehardcode from origin/master to upstream/master
- bin/get-pick-list: use –oneline=pretty instead of –oneline

Emil Velikov (1):

- Update version to 19.2.0-rc1

Eric Engestrom (14):

- ttn: fix 64-bit shift on 32-bit '1'
- egl: fix deadlock in malloc error path
- util/os_file: fix double-close()
- anv: fix format string in error message
- freedreno/drm-shim: fix mem leak
- nir: fix memleak in error path

- anv: add support for driconf
- wsi: add minImageCount override
- anv: add support for vk_x11_override_min_image_count
- amd: move adaptive sync to performance section, as it is defined in xmlpool
- radv: add support for vk_x11_override_min_image_count
- drirc: override minImageCount=2 for gfxbench
- gl: drop incorrect pkg-config file for glvnd
- meson: re-add incorrect pkg-config files with GLVND for backward compatibility

Erik Faye-Lund (2):

- gallium/auxiliary/indices: consistently apply start only to input
- util: fix SSE-version needed for double opcodes

Haihao Xiang (1):

- i965: support AYUV/XYUV for external import only

Hal Gentz (2):

- glx: Fix SEGV due to dereferencing a NULL ptr from XCB-GLX.
- gallium/osmesa: Fix the inability to set no context as current.

Iago Toral Quiroga (1):

- v3d: make sure we have enough space in the CL for the primitive counts packet

Ian Romanick (8):

- nir/algrbraic: Don't optimize open-coded bitfield reverse when lowering is enabled
- intel/compiler: Request bitfield_reverse lowering on pre-Gen7 hardware
- nir/algebraic: Mark some value range analysis-based optimizations imprecise
- nir/range-analysis: Adjust result range of exp2 to account for flush-to-zero
- nir/range-analysis: Adjust result range of multiplication to account for flush-to-zero
- nir/range-analysis: Fix incorrect fadd range result for (ne_zero, ne_zero)
- nir/range-analysis: Handle constants in nir_op_mov just like nir_op_bcsel
- nir/algebraic: Do not apply late DPH optimization in vertex processing stages

Ilia Mirkin (1):

- gallium/vl: use compute preference for all multimedia, not just blit

Jason Ekstrand (9):

- anv: Bump maxComputeWorkgroupSize
- nir: Handle complex derefs in nir_split_array_vars
- nir: Don't infinitely recurse in lower_ssa_defs_to_regs_block
- nir: Add a block_is_unreachable helper
- nir/repair_ssa: Repair dominance for unreachable blocks
- nir/repair_ssa: Insert deref casts when needed

- nir/dead_cf: Repair SSA if the pass makes progress
- intel/fs: Handle UNDEF in split_virtual_grfs
- nir/repair_ssa: Replace the unreachable check with the phi builder

Jonathan Marek (1):

- freedreno/a2xx: ir2: fix lowering of instructions after float lowering

Jose Maria Casanova Crespo (1):

- mesa: recover target_check before get_current_tex_objects

Juan A. Suarez Romero (1):

- bin/get-pick-list.sh: sha1 commits can be smaller than 8 chars

Kenneth Graunke (20):

- gallium/ddebug: Wrap resource_get_param if available
- gallium/trace: Wrap resource_get_param if available
- gallium/rbug: Wrap resource_get_param if available
- gallium/noop: Implement resource_get_param
- iris: Replace devinfo->gen with GEN_GEN
- iris: Fix broken aux.possible/sampler_usages bitmask handling
- iris: Update fast clear colors on Gen9 with direct immediate writes.
- iris: Drop copy format hacks from copy region based transfer path.
- iris: Avoid unnecessary resolves on transfer maps
- iris: Fix large timeout handling in rel2abs()
- isl: Drop UnormPathInColorPipe for buffer surfaces.
- isl: Don't set UnormPathInColorPipe for integer surfaces.
- util: Add a _mesa_i64roundevenf() helper.
- mesa: Fix _mesa_float_to_unorm() on 32-bit systems.
- iris: Fix partial fast clear checks to account for miplevel.
- iris: Report correct number of planes for planar images
- iris: Fix constant buffer sizes for non-UBOs
- gallium: Fix util_format_get_depth_only
- iris: Initialize ice->state.prim_mode to an invalid value
- intel: Increase Gen11 compute shader scratch IDs to 64.

Lepton Wu (1):

- virgl: Fix pipe_resource leaks under multi-sample.

Lionel Landwerlin (9):

- util/timespec: use unsigned 64 bit integers for nsec values
- util: fix compilation on macos
- egl: fix platform selection

- vulkan/overlay: bounce image back to present layout
- radv: store engine name
- driconfig: add a new engine name/version parameter
- vulkan: add vk_x11_strict_image_count option
- util/xmlconfig: fix regexp compile failure check
- drirc: include unreal engine version 0 to 23

Marek Olšák (23):

- radeonsi/gfx10: fix the legacy pipeline by storing as_ngg in the shader cache
- radeonsi: move some global shader cache flags to per-binary flags
- radeonsi/gfx10: fix tessellation for the legacy pipeline
- radeonsi/gfx10: fix the PRIMITIVES_GENERATED query if using legacy streamout
- radeonsi/gfx10: create the GS copy shader if using legacy streamout
- radeonsi/gfx10: add as_ngg variant for VS as ES to select Wave32/64
- radeonsi/gfx10: fix InstanceID for legacy VS+GS
- radeonsi/gfx10: don't initialize VGT_INSTANCE_STEP_RATE_0
- radeonsi/gfx10: always use the legacy pipeline for streamout
- radeonsi/gfx10: finish up Navi14, add PCI ID
- radeonsi/gfx10: add AMD_DEBUG=nongg
- winsys/amdgpu+radeon: process AMD_DEBUG in addition to R600_DEBUG
- radeonsi: add PKT3_CONTEXT_REG_RMW
- radeonsi/gfx10: remove incorrect ngg/pos_writes_edgeflag variables
- radeonsi/gfx10: set PA_CL_VS_OUT_CNTL with CONTEXT_REG_RMW to fix edge flags
- radeonsi: consolidate determining VGPR_COMP_CNT for API VS
- radeonsi: unbind blend/DSA/rasterizer state correctly in delete functions
- radeonsi: fix scratch buffer WAVESIZE setting leading to corruption
- radeonsi/gfx10: don't call gfx10_destroy_query with compute-only contexts
- radeonsi/gfx10: fix wave occupancy computations
- radeonsi: add Navi12 PCI ID
- amd: add more PCI IDs for Navi14
- ac/addrlib: fix chip identification for Vega10, Arcturus, Raven2, Renoir

Mauro Rossi (2):

- android: mesa: revert "Enable asm unconditionally"
- android: anv: libmesa_vulkan_common: add libmesa_util static dependency

Paulo Zanoni (2):

- intel/fs: grab fail_msg from v32 instead of v16 when v32->run_cs fails
- intel/fs: fix SHADER_OPCODE_CLUSTER_BROADCAST for SIMD32

Pierre-Eric Pelloux-Prayer (1):

- glsl: replace 'x + (-x)' with constant 0

Rafael Antognolli (1):

- anv: Only re-emit non-dynamic state that has changed.

Rhys Perry (1):

- radv: always emit a position export in gs copy shaders

Samuel Iglesias Gonsálvez (1):

- intel/nir: do not apply the fsin and fcos trig workarounds for consts

Samuel Pitoiset (11):

- radv: allow to enable VK_AMD_shader_ballot only on GFX8+
- radv: add a new debug option called RADV_DEBUG=noshaderballot
- radv: force enable VK_AMD_shader_ballot for Wolfenstein Youngblood
- ac: fix exclusive scans on GFX8-GFX9
- radv/gfx10: don't initialize VGT_INSTANCE_STEP_RATE_0
- radv/gfx10: do not use NGG with NAVI14
- radv: fix getting the index type size for uint8_t
- nir: do not assume that the result of fexp2(a) is always an integral
- radv: fix allocating number of user sgprs if streamout is used
- radv: fix loading 64-bit GS inputs
- radv/gfx10: fix VK_KHR_pipeline_executable_properties with NGG GS

Sergii Romantsov (2):

- intel/dri: finish proper glthread
- nir/large_constants: more careful data copying

Tapani Pälli (5):

- util: fix os_create_anonymous_file on android
- iris/android: fix build and link with libmesa_intel_perf
- egl: reset blob cache set/get functions on terminate
- iris: close screen fd on iris_destroy_screen
- egl: check for NULL value like eglGetSyncAttribKHR does

Thong Thai (1):

- Revert "radeonsi: don't emit PKT3_CONTEXT_CONTROL on amdgpu"

Timur Kristóf (1):

- st/nine: Properly initialize GLSL types for NIR shaders.

Vinson Lee (2):

- swr: Fix build with llvm-9.0 again.
- travis: Fail build if any command in if statement fails.

4.39 Mesa 19.1.7 Release Notes / September 17, 2019

Mesa 19.1.7 is a bug fix release which fixes bugs found since the 19.1.6 release.

Mesa 19.1.7 implements the OpenGL 4.5 API, but the version reported by glGetString(GL_VERSION) or glGetIntegerv(GL_MAJOR_VERSION) / glGetIntegerv(GL_MINOR_VERSION) depends on the particular driver being used. Some drivers don't support all the features required in OpenGL 4.5. OpenGL 4.5 is **only** available if requested at context creation. Compatibility contexts may report a lower version depending on each driver.

Mesa 19.1.7 implements the Vulkan 1.1 API, but the version reported by the apiVersion property of the VkPhysicalDeviceProperties struct depends on the particular driver being used.

4.39.1 SHA256 checksums

```
e287920fdb38712a9fed448dc90b3ca95048c7face5db52e58361f8b6e0f3cd5  mesa-19.1.7.tar.xz
```

4.39.2 New features

None

4.39.3 Bug fixes

- Bug 110814 - KWin compositor crashes on launch
- Bug 111069 - Assertion fails in nir_opt_remove_phis.c during compilation of SPIR-V shader
- Bug 111271 - Crash in eglMakeCurrent
- Bug 111401 - Vulkan overlay layer - async compute not supported, making overlay disappear in Doom
- Bug 111405 - Some infinite 'do{}while' loops lead mesa to an infinite compilation
- Bug 111467 - WOLF RPG Editor + Gallium Nine Standalone: Rendering issue when using Iris driver
- Bug 111552 - Geekbench 5.0 Vulkan compute benchmark fails on Anvil

4.39.4 Changes

Caio Marcelo de Oliveira Filho (1):

- glsl/nir: Avoid overflow when setting max_uniform_location

Connor Abbott (1):

- radv: Call nir_propagate_invariant()

Danylo Piliaiev (1):

- tgsi_to_nir: Translate TGSI_INTERPOLATE_COLOR as INTERP_MODE_NONE

Eric Engestrom (10):

- ttn: fix 64-bit shift on 32-bit '1'
- egl: fix deadlock in malloc error path
- util/os_file: fix double-close()

- anv: fix format string in error message
- nir: fix memleak in error path
- anv: add support for driconf
- wsi: add minImageCount override
- anv: add support for vk_x11_override_min_image_count
- amd: move adaptive sync to performance section, as it is defined in xmlpool
- radv: add support for vk_x11_override_min_image_count

Erik Faye-Lund (2):

- gallium/auxiliary/indices: consistently apply start only to input
- util: fix SSE-version needed for double opcodes

Hal Gentz (1):

- glx: Fix SEGV due to dereferencing a NULL ptr from XCB-GLX.

Jason Ekstrand (7):

- Revert "intel/fs: Move the scalar-region conversion to the generator."
- anv: Bump maxComputeWorkgroupSize
- nir: Don't infinitely recurse in lower_ssa_defs_to_regs_block
- nir: Add a block_is_unreachable helper
- nir/repair_ssa: Repair dominance for unreachable blocks
- nir/repair_ssa: Insert deref casts when needed
- nir/dead_cf: Repair SSA if the pass makes progress

Juan A. Suarez Romero (3):

- docs: add sha256 checksums for 19.1.6
- cherry-ignore: add explicit 19.2 only nominations
- Update version to 19.1.7

Kenneth Graunke (1):

- gallium: Fix util_format_get_depth_only

Lionel Landwerlin (1):

- vulkan/overlay: bounce image back to present layout

Mauro Rossi (3):

- android: radv: fix necessary dependecies
- android: amd/common: fix missing include path
- android: anv: libmesa_vulkan_common: add libmesa_util static dependency

Samuel Pitoiset (1):

- radv: fix allocating number of user sgprs if streamout is used

Sergii Romantsov (1):

- intel/dri: finish proper glthread

4.40 Mesa 19.1.6 Release Notes / September 3, 2019

Mesa 19.1.6 is a bug fix release which fixes bugs found since the 19.1.5 release.

Mesa 19.1.6 implements the OpenGL 4.5 API, but the version reported by glGetString(GL_VERSION) or glGetIntegerv(GL_MAJOR_VERSION) / glGetIntegerv(GL_MINOR_VERSION) depends on the particular driver being used. Some drivers don't support all the features required in OpenGL 4.5. OpenGL 4.5 is **only** available if requested at context creation. Compatibility contexts may report a lower version depending on each driver.

4.40.1 SHA256 checksums

```
2a369b7b48545c6486e7e44913ad022daca097c8bd937bf30dcf3f17a94d3496  mesa-19.1.6.tar.xz
```

4.40.2 New features

None

4.40.3 Bug fixes

- Bug 104395 - [CTS] GTF-GL46.gtf32.GL3Tests.packed_pixels.packed_pixels tests fail on 32bit Mesa
- Bug 111213 - VA-API nouveau SIGSEGV and asserts
- Bug 111241 - Shadertoy shader causing hang
- Bug 111411 - SPIR-V shader leads to GPU hang, sometimes making machine unstable

4.40.4 Changes

Andres Rodriguez (1):

- radv: additional query fixes

Daniel Schürmann (1):

- nir/lcssa: handle deref instructions properly

Danylo Piliaiev (1):

- nir/loop_unroll: Prepare loop for unrolling in wrapper_unroll

Ian Romanick (2):

- nir/algrbraic: Don't optimize open-coded bitfield reverse when lowering is enabled
- intel/compiler: Request bitfield_reverse lowering on pre-Gen7 hardware

Ilia Mirkin (1):

- gallium/vl: use compute preference for all multimedia, not just blit

Jonas Ådahl (1):

- wayland/egl: Ensure correct buffer size when allocating

Juan A. Suarez Romero (6):

- docs: add sha256 checksums for 19.1.5

- cherry-ignore: add explicit 19.2 only nominations
- cherry-ignore: iris: Replace devinfo->gen with GEN_GEN
- cherry-ignore: iris: Update fast clear colors on Gen9 with direct immediate writes.
- cherry-ignore: iris: Avoid unnecessary resolves on transfer maps
- Update version to 19.1.6

Kenneth Graunke (6):

- iris: Fix broken aux.possible/sampler_usages bitmask handling
- iris: Drop copy format hacks from copy region based transfer path.
- iris: Fix large timeout handling in rel2abs()
- util: Add a _mesa_i64roundevenf() helper.
- mesa: Fix _mesa_float_to_unorm() on 32-bit systems.
- intel/compiler: Fix src0/desc setter ordering

Marek Olšák (1):

- radeonsi: fix scratch buffer WAVESIZE setting leading to corruption

Paulo Zanoni (1):

- intel/fs: grab fail_msg from v32 instead of v16 when v32->run_cs fails

Pierre-Eric Pelloux-Prayer (1):

- glsl: replace 'x + (-x)' with constant 0

Tapani Pälli (1):

- egl: reset blob cache set/get functions on terminate

4.41 Mesa 19.1.5 Release Notes / August 23, 2019

Mesa 19.1.5 is a bug fix release which fixes bugs found since the 19.1.4 release.

Mesa 19.1.5 implements the OpenGL 4.5 API, but the version reported by glGetString(GL_VERSION) or glGetIntegerv(GL_MAJOR_VERSION) / glGetIntegerv(GL_MINOR_VERSION) depends on the particular driver being used. Some drivers don't support all the features required in OpenGL 4.5. OpenGL 4.5 is **only** available if requested at context creation. Compatibility contexts may report a lower version depending on each driver.

4.41.1 SHA256 checksums

```
7b54e14e35c7251b171b4cf9d84cbc1d760eafe00132117db193454999cd6eb4  mesa-19.1.5.tar.xz
```

4.41.2 New features

None

4.41.3 Bug fixes

- Bug 109630 - vkQuake flickering geometry under Intel
- Bug 110395 - Shadows are flickering in SuperTuxKart
- Bug 111113 - ANGLE BlitFramebufferTest.MultisampleDepthClear/ES3_OpenGL fails on Intel Ubuntu19.04
- Bug 111267 - [CM246] Flickering with multiple draw calls within the same graphics pipeline if a compute pipeline is present

4.41.4 Changes

Bas Nieuwenhuizen (4):

- radv: Do non-uniform lowering before bool lowering.
- ac/nir: Use correct cast for readfirstlane and ptrs.
- radv: Avoid binning RAVEN hangs.
- radv: Avoid VEGA/RAVEN scissor bug in binning.

Danylo Piliaiev (1):

- i965: Emit a dummy MEDIA_VFE_STATE before switching from GPGPU to 3D

Eric Engestrom (1):

- util: fix mem leak of program path

Erik Faye-Lund (2):

- gallium/dump: add missing query-type to short-list
- gallium/dump: add missing query-type to short-list

Greg V (2):

- anv: remove unused Linux-specific include
- intel/perf: use MAJOR_IN_SYSMACROS/MAJOR_IN_MKDEV

Jason Ekstrand (1):

- anv: Emit a dummy MEDIA_VFE_STATE before switching from GPGPU to 3D

Juan A. Suarez Romero (3):

- docs: add sha256 checksums for 19.1.4
- cherry-ignore: panfrost: Make ctx->job useful
- Update version to 19.1.5

Marek Olšák (2):

- radeonsi: disable SDMA image copies on dGPUs to fix corruption in games
- radeonsi: fix an assertion failure: assert(!res->b.is_shared)

Matt Turner (1):

- meson: Test for program_invocation_name

Sergii Romantsov (1):

- i965/clear: clear_value better precision

4.42 Mesa 19.1.4 Release Notes / August 7, 2019

Mesa 19.1.4 is a bug fix release which fixes bugs found since the 19.1.3 release.

Mesa 19.1.4 implements the OpenGL 4.5 API, but the version reported by glGetString(GL_VERSION) or glGetIntegerv(GL_MAJOR_VERSION) / glGetIntegerv(GL_MINOR_VERSION) depends on the particular driver being used. Some drivers don't support all the features required in OpenGL 4.5. OpenGL 4.5 is **only** available if requested at context creation. Compatibility contexts may report a lower version depending on each driver.

4.42.1 SHA256 checksums

```
a6d268a7d9edcfd92b6da80f2e34e6e0a7baaa442efbeba2fc66c404943c6bfb  mesa-19.1.4.tar.xz
```

4.42.2 New features

None

4.42.3 Bug fixes

- Bug 109203 - [cfl dxvk] GPU Crash Launching Monopoly Plus (Iris Plus 655 / Wine + DXVK)
- Bug 109524 - "Invalid glsl version in shading_language_version()" when trying to run directX games using wine
- Bug 110309 - [icl][bisected] regression on piglit arb_gpu_shader_int 64.execution.fs-ishl-then-* tests
- Bug 110663 - threads_posix.h:96: undefined reference to 'pthread_once'
- Bug 110955 - Mesa 18.2.8 implementation error: Invalid GLSL version in shading_language_version()
- Bug 111010 - Cemu Shader Cache Corruption Displaying Solid Color After commit 11e16ca7ce0
- Bug 111071 - SPIR-V shader processing fails with message about "extra dangling SSA sources"
- Bug 111075 - Processing of SPIR-V shader causes device hang, sometimes leading to system reboot
- Bug 111097 - Can not detect VK_ERROR_OUT_OF_DATE_KHR or VK_SUBOPTIMAL_KHR when window resizing

4.42.4 Changes

Andres Rodriguez (1):

- radv: fix queries with WAIT_BIT returning VK_NOT_READY

Andrii Simiklit (2):

- intel/compiler: don't use a keyword struct for a class fs_reg
- meson: add a warning for meson < 0.46.0

Arcady Goldmints-Orlov (1):

- anv: report HOST_ALLOCATION as supported for images

Bas Nieuwenhuizen (3):

- radv: Set correct metadata size for GFX9+.

- radv: Take variable descriptor counts into account for buffer entries.
- radv: Fix descriptor set allocation failure.

Boyuan Zhang (4):

- radeon/uvd: fix poc for hevc encode
- radeon/vcn: fix poc for hevc encode
- radeon/uvd: enable rate control for hevc encoding
- radeon/vcn: enable rate control for hevc encoding

Caio Marcelo de Oliveira Filho (1):

- anv: Remove special allocation for anv_push_constants

Connor Abbott (1):

- nir: Allow qualifiers on copy_deref and image instructions

Daniel Schürmann (1):

- spirv: Fix order of barriers in SpvOpControlBarrier

Dave Airlie (1):

- st/nir: fix arb fragment stage conversion

Dylan Baker (1):

- meson: allow building all glx without any drivers

Emil Velikov (1):

- egl/drm: ensure the backing gbm is set before using it

Eric Anholt (1):

- freedreno: Fix data races with allocating/freeing struct ir3.

Eric Engestrom (5):

- nir: don't return void
- util: fix no-op macro (bad number of arguments)
- gallium+mesa: fix tgsi_semantic array type
- scons+meson: suppress spammy build warning on MacOS
- nir: remove explicit nir_intrinsic_index_flag values

Francisco Jerez (1):

- intel/ir: Fix CFG corruption in opt_predicated_break().

Ilia Mirkin (4):

- gallium/vl: fix compute tgsi shaders to not process undefined components
- nv50,nvc0: update sampler/view bind functions to accept NULL array
- nvc0: allow a non-user buffer to be bound at position 0
- nv50/ir: handle insn not being there for definition of CVT arg

Jason Ekstrand (6):

- intel/fs: Stop stack allocating large arrays

- anv: Disable transform feedback on gen7
- isl/formats: R8G8B8_UNORM_SRGB isn't supported on HSW
- anv: Don't claim support for 24 and 48-bit formats on IVB
- intel/fs: Use ALIGN16 instructions for all derivatives on gen <= 7
- intel/fs: Implement quad_swap_horizontal with a swizzle on gen7

Juan A. Suarez Romero (2):

- docs: add sha256 checksums for 19.1.3
- Update version to 19.1.4

Kenneth Graunke (4):

- mesa: Fix ReadBuffers with pbuffers
- egl: Quiet warning about front buffer rendering for pixmaps/pbuffers
- egl: Make the 565 pbuffer-only config single buffered.
- egl: Only expose 565 pbuffer configs if X can export them as DRI3 images

Lionel Landwerlin (5):

- anv: fix use of comma operator
- nir: add access to image_deref intrinsics
- spirv: wrap push ssa/pointer values
- spirv: propagate access qualifiers through ssa & pointer
- spirv: don't discard access set by vtn_pointer_dereference

Mark Menzynski (1):

- nvc0/ir: Fix assert accessing null pointer

Nataraj Deshpande (1):

- egl/android: Update color_buffers querying for buffer age

Nicolas Dufresne (1):

- egl: Also query modifiers when exporting DMABuf

Rhys Perry (1):

- ac/nir: fix txf_ms with an offset

Samuel Pitoiset (1):

- radv: fix crash in vkCmdClearAttachments with unused attachment

Tapani Pälli (1):

- mesa: add glsl_type ref to one_time_init and decref to atexit

Yevhenii Kolesnikov (1):

- main: Fix memleaks in mesa_use_program

4.43 Mesa 19.1.3 Release Notes / July 23, 2019

Mesa 19.1.3 is a bug fix release which fixes bugs found since the 19.1.2 release.

Mesa 19.1.3 implements the OpenGL 4.5 API, but the version reported by glGetString(GL_VERSION) or glGetIntegerv(GL_MAJOR_VERSION) / glGetIntegerv(GL_MINOR_VERSION) depends on the particular driver being used. Some drivers don't support all the features required in OpenGL 4.5. OpenGL 4.5 is **only** available if requested at context creation. Compatibility contexts may report a lower version depending on each driver.

4.43.1 SHA256 checksums

```
845460b2225d15c15d4a9743dec798ff0b7396b533011d43e774e67f7825b7e0  mesa-19.1.3.tar.xz
```

4.43.2 New features

None

4.43.3 Bug fixes

- Bug 109203 - [cfl dxvk] GPU Crash Launching Monopoly Plus (Iris Plus 655 / Wine + DXVK)
- Bug 109524 - "Invalid glsl version in shading_language_version()" when trying to run directX games using wine
- Bug 110309 - [icl][bisected] regression on piglit arb_gpu_shader_int 64.execution.fs-ishl-then-* tests
- Bug 110663 - threads_posix.h:96: undefined reference to 'pthread_once'
- Bug 110955 - Mesa 18.2.8 implementation error: Invalid GLSL version in shading_language_version()
- Bug 111010 - Cemu Shader Cache Corruption Displaying Solid Color After commit 11e16ca7ce0
- Bug 111071 - SPIR-V shader processing fails with message about "extra dangling SSA sources"
- Bug 111075 - Processing of SPIR-V shader causes device hang, sometimes leading to system reboot
- Bug 111097 - Can not detect VK_ERROR_OUT_OF_DATE_KHR or VK_SUBOPTIMAL_KHR when window resizing

4.43.4 Changes

Bas Nieuwenhuizen (3):

- radv: Handle cmask being disallowed by addrlib.
- anv: Add android dependencies on android.
- radv: Only save the descriptor set if we have one.

Caio Marcelo de Oliveira Filho (2):

- anv: Fix pool allocator when first alloc needs to grow
- spirv: Fix stride calculation when lowering Workgroup to offsets

Chia-I Wu (2):

- anv: fix VkExternalBufferProperties for unsupported handles

- anv: fix VkExternalBufferProperties for host allocation

Connor Abbott (1):

- nir: Add a helper to determine if an intrinsic can be reordered

Dave Airlie (1):

- radv: fix crash in shader tracing.

Eric Anholt (1):

- freedreno: Fix assertion failures in context setup in shader-db mode.

Gert Wollny (1):

- softpipe: Remove unused static function

Ian Romanick (4):

- intel/vec4: Reswizzle VF immediates too
- nir: Add unit tests for nir_opt_comparison_pre
- nir: Use nir_src_bit_size instead of alu1->dest.dest.ssa.bit_size
- mesa: Set minimum possible GLSL version

Jason Ekstrand (13):

- nir/instr_set: Expose nir_instrs_equal()
- nir/loop_analyze: Fix phi-of-identical-alu detection
- nir: Add more helpers for working with const values
- nir/loop_analyze: Handle bit sizes correctly in calculate_iterations
- nir/loop_analyze: Bail if we encounter swizzles
- anv: Set Stateless Data Port Access MOCS
- nir/opt_if: Clean up single-src phis in opt_if_loop_terminator
- nir,intel: Add support for lowering 64-bit nir_opt_extract_*
- anv: Account for dynamic stencil write disables in the PMA fix
- nir/regs_to_ssa: Handle regs in phi sources properly
- nir/loop_analyze: Refactor detection of limit vars
- nir: Add some helpers for chasing SSA values properly
- nir/loop_analyze: Properly handle swizzles in loop conditions

Juan A. Suarez Romero (2):

- docs: add sha256 checksums for 19.1.2
- Update version to 19.1.3

Lepton Wu (1):

- virgl: Set meta data for textures from handle.

Lionel Landwerlin (6):

- vulkan/overlay: fix command buffer stats
- vulkan/overlay: fix crash on freeing NULL command buffer

- anv: fix crash in vkCmdClearAttachments with unused attachment
- vulkan/wsi: update swapchain status on vkQueuePresent
- anv: report timestampComputeAndGraphics true
- anv: fix format mapping for depth/stencil formats

Marek Olšák (1):

- radeonsi: don't set READ_ONLY for const_uploader to fix bindless texture hangs

Samuel Iglesias Gonsálvez (1):

- anv: fix alphaToCoverage when there is no color attachment

Samuel Pitoiset (1):

- radv: fix VGT_GS_MODE if VS uses the primitive ID

Sergii Romantsov (1):

- meta: memory leak of CopyPixels usage

Timothy Arceri (1):

- mesa: save/restore SSO flag when using ARB_get_program_binary

Vinson Lee (1):

- meson: Add dep_thread dependency.

Yevhenii Kolesnikov (1):

- meta: leaking of BO with DrawPixels

4.44 Mesa 19.1.2 Release Notes / July 9, 2019

Mesa 19.1.2 is a bug fix release which fixes bugs found since the 19.1.1 release.

Mesa 19.1.2 implements the OpenGL 4.5 API, but the version reported by glGetString(GL_VERSION) or glGetIntegerv(GL_MAJOR_VERSION) / glGetIntegerv(GL_MINOR_VERSION) depends on the particular driver being used. Some drivers don't support all the features required in OpenGL 4.5. OpenGL 4.5 is **only** available if requested at context creation. Compatibility contexts may report a lower version depending on each driver.

4.44.1 SHA256 checksums

```
813a144ea8ebefb7b48b6733f3f603855b0f61268d86cc1cc26a6b4be908fcfd  mesa-19.1.2.tar.xz
```

4.44.2 New features

None

4.44.3 Bug fixes

- Bug 110702 - segfault in radeonsi HEVC hardware decoding with yuv420p10le
- Bug 110783 - Mesa 19.1 rc crashing MPV with VAAPI

- Bug 110944 - [Bisected] Blender 2.8 crashes when closing certain windows
- Bug 110953 - Adding a redundant single-iteration do-while loop causes different image to be rendered
- Bug 110999 - 19.1.0: assert in vkAllocateDescriptorSets using immutable samplers on Ivy Bridge
- Bug 111019 - radv doesn't handle variable descriptor count properly

4.44.4 Changes

Anuj Phogat (3):

- Revert "i965/icl: Add WA_2204188704 to disable pixel shader panic dispatch"
- Revert "anv/icl: Add WA_2204188704 to disable pixel shader panic dispatch"
- Revert "iris/icl: Add WA_2204188704 to disable pixel shader panic dispatch"

Arfrever Frehtes Taifersar Arahesis (1):

- meson: Improve detection of Python when using Meson >=0.50.

Bas Nieuwenhuizen (2):

- radv: Only allocate supplied number of descriptors when variable.
- radv: Fix interactions between variable descriptor count and inline uniform blocks.

Caio Marcelo de Oliveira Filho (1):

- spirv: Ignore ArrayStride in OpPtrAccessChain for Workgroup

Dylan Baker (2):

- meson: Add support for using cmake for finding LLVM
- Revert "meson: Add support for using cmake for finding LLVM"

Eric Anholt (2):

- freedreno: Fix UBO load range detection on booleans.
- freedreno: Fix up end range of unaligned UBO loads.

Eric Engestrom (1):

- meson: bump required libdrm version to 2.4.81

Gert Wollny (2):

- gallium: Add CAP for opcode DIV
- vl: Use CS composite shader only if TEX_LZ and DIV are supported

Ian Romanick (1):

- glsl: Don't increase the iteration count when there are no terminators

James Clarke (1):

- meson: GNU/kFreeBSD has DRM/KMS and requires -D_GNU_SOURCE

Jason Ekstrand (2):

- anv/descriptor_set: Only write texture swizzles if we have an image view
- iris: Use a uint16_t for key sizes

Jory Pratt (2):

- util: Heap-allocate 256K zlib buffer
- meson: Search for execinfo.h

Juan A. Suarez Romero (3):

- docs: add sha256 checksums for 19.1.1
- intel: fix wrong format usage
- Update version to 19.1.2

Kenneth Graunke (2):

- iris: Enable PIPE_CAP_SURFACE_REINTERPRET_BLOCKS
- gallium: Make util_copy_image_view handle shader_access

Lionel Landwerlin (2):

- intel/compiler: fix derivative on y axis implementation
- intel/compiler: don't use byte operands for src1 on ICL

Nanley Chery (2):

- intel: Add and use helpers for level0 extent
- isl: Don't align phys_level0_sa by block dimension

Nataraj Deshpande (1):

- anv: Add HAL_PIXEL_FORMAT_IMPLEMENTATION_DEFINED in vk_format

Pierre-Eric Pelloux-Prayer (2):

- mesa: delete framebuffer texture attachment sampler views
- radeon/uvd: fix calc_ctx_size_h265_main10

Rob Clark (1):

- freedreno/a5xx: fix batch leak in fd5 blitter path

Sagar Ghuge (1):

- glsl: Fix round64 conversion function

Samuel Pitoiset (1):

- radv: only enable VK_AMD_gpu_shader_{half_float,int16} on GFX9+

Sergii Romantsov (1):

- i965: leaking of upload-BO with push constants

Ville Syrjälä (1):

- anv/cmd_buffer: Reuse gen8 Cmd{Set, Reset}Event on gen7

4.45 Mesa 19.0.8 Release Notes / June 26, 2019

Mesa 19.0.8 is an emergency bug fix release which fixes a critical bug found in the 19.0.7 release.

Mesa 19.0.8 implements the OpenGL 4.5 API, but the version reported by glGetString(GL_VERSION) or glGetIntegerv(GL_MAJOR_VERSION) / glGetIntegerv(GL_MINOR_VERSION) depends on the particular driver being used.

Some drivers don't support all the features required in OpenGL 4.5. OpenGL 4.5 is **only** available if requested at context creation. Compatibility contexts may report a lower version depending on each driver.

4.45.1 SHA256 checksums

```
1a3dc3f2af853c76aadb4a1e03c9ba420361c04a742d457a702b781671a96a57  mesa-19.0.8.tar.gz
d017eb53a810c32dabeedf6ca2238ae1e897ce9090e470e9ce1d6c9e3f1b0862  mesa-19.0.8.tar.xz
```

4.45.2 New features

N/A

4.45.3 Bug fixes

None

4.45.4 Changes

Dylan Baker (2):

- docs: Add SHA256 sums for 19.0.7
- version: bump to 19.0.8

Kenneth Graunke (1):

- egl/x11: calloc dri2_surf so it's properly zeroed

4.46 Mesa 19.1.1 Release Notes / June 25, 2019

Mesa 19.1.1 is a bug fix release which fixes bugs found since the 19.1.0 release.

Mesa 19.1.1 implements the OpenGL 4.5 API, but the version reported by glGetString(GL_VERSION) or glGetIntegerv(GL_MAJOR_VERSION) / glGetIntegerv(GL_MINOR_VERSION) depends on the particular driver being used. Some drivers don't support all the features required in OpenGL 4.5. OpenGL 4.5 is **only** available if requested at context creation. Compatibility contexts may report a lower version depending on each driver.

4.46.1 SHA256 checksums

```
72114b16b4a84373b2acda060fe2bb1d45ea2598efab3ef2d44bdeda74f15581  mesa-19.1.1.tar.xz
```

4.46.2 New features

None

4.46.3 Bug fixes

- Bug 110709 - g_glxglvnddispatchfuncs.c and glxglvnd.c fail to build with clang 8.0
- Bug 110901 - mesa-19.1.0/src/util/futex.h:82: use of out of scope variable ?
- Bug 110902 - mesa-19.1.0/src/broadcom/compiler/vir_opt_redundant_flags.c:104]: (style) Same expression
- Bug 110921 - virgl on OpenGL 3.3 host regressed to OpenGL 2.1

4.46.4 Changes

Alejandro Piñeiro (1):

- v3d: fix checking twice auf flag

Bas Nieuwenhuizen (5):

- radv: Skip transitions coming from external queue.
- radv: Decompress DCC when the image format is not allowed for buffers.
- radv: Fix vulkan build in meson.
- anv: Fix vulkan build in meson.
- meson: Allow building radeonsi with just the android platform.

Dave Airlie (1):

- nouveau: fix frees in unsupported IR error paths.

Eduardo Lima Mitev (1):

- freedreno/a5xx: Fix indirect draw max_indices calculation

Eric Engestrom (3):

- util/futex: fix dangling pointer use
- glx: fix glvnd pointer types
- util/os_file: resize buffer to what was actually needed

Gert Wollny (1):

- virgl: Assume sRGB write control for older guest kernels or virglrenderer hosts

Haihao Xiang (1):

- i965: support UYVY for external import only

Jason Ekstrand (1):

- anv: Set STATE_BASE_ADDRESS upper bounds on gen7

Juan A. Suarez Romero (2):

- docs: Add SHA256 sums for 19.1.0
- Update version to 19.1.1

Kenneth Graunke (2):

- glsl: Fix out of bounds read in shader_cache_read_program_metadata
- iris: Fix iris_flush_and_dirty_history to actually dirty history.

Kevin Strasser (2):

- gallium/winsys/kms: Fix dumb buffer bpp
- st/mesa: Add rgbx handling for fp formats

Lionel Landwerlin (2):

- anv: do not parse genxml data without INTEL_DEBUG=bat
- intel/dump: fix segfault when the app hasn't accessed the device

Mathias Fröhlich (1):

- egl: Don't add hardware device if there is no render node v2.

Richard Thier (1):

- r300g: restore performance after RADEON_FLAG_NO_INTERPROCESS_SHARING was added

Rob Clark (1):

- freedreno/a6xx: un-swap X24S8_UINT

Samuel Pitoiset (4):

- radv: fix occlusion queries on VegaM
- radv: fix VK_EXT_memory_budget if one heap isn't available
- radv: fix FMASK expand with SRGB formats
- radv: disable viewport clamping even if FS doesn't write Z

4.47 Mesa 19.0.7 Release Notes / June 24, 2019

Mesa 19.0.7 is a bug fix release which fixes bugs found since the 19.0.6 release.

Mesa 19.0.7 implements the OpenGL 4.5 API, but the version reported by glGetString(GL_VERSION) or glGetIntegerv(GL_MAJOR_VERSION) / glGetIntegerv(GL_MINOR_VERSION) depends on the particular driver being used. Some drivers don't support all the features required in OpenGL 4.5. OpenGL 4.5 is **only** available if requested at context creation. Compatibility contexts may report a lower version depending on each driver.

4.47.1 SHA256 checksums

```
81119f0cbbd1fbe7c0574e1e2690e0dae8868124d24c875f5fb76f165db3a54d  mesa-19.0.7.tar.gz
d7bf3db2e442fe5eeb96144f8508d94f04aededdf37af477e644638d366b2b28  mesa-19.0.7.tar.xz
```

4.47.2 New features

N/A

4.47.3 Bug fixes

- Bug 110302 - [bisected][regression] piglit egl-create-pbuffer-surface and egl-gl-colorspace regressions
- Bug 110921 - virgl on OpenGL 3.3 host regressed to OpenGL 2.1

4.47.4 Changes

Bas Nieuwenhuizen (5):

- radv: Prevent out of bound shift on 32-bit builds.
- radv: Decompress DCC when the image format is not allowed for buffers.
- radv: Fix vulkan build in meson.
- anv: Fix vulkan build in meson.
- meson: Allow building radeonsi with just the android platform.

Charmaine Lee (1):

- svga: Remove unnecessary check for the pre flush bit for setting vertex buffers

Deepak Rawat (1):

- winsys/svga/drm: Fix 32-bit RPCI send message

Dylan Baker (3):

- docs: Add SHA256 sums for 19.0.6
- cherry-ignore: add additional 19.1 only patches
- Bump version for 19.0.7 release

Emil Velikov (1):

- mapi: correctly handle the full offset table

Gert Wollny (2):

- virgl: Add a caps feature check version
- virgl: Assume sRGB write control for older guest kernels or virglrenderer hosts

Haihao Xiang (1):

- i965: support UYVY for external import only

Jason Ekstrand (2):

- nir/propagate_invariant: Don't add NULL vars to the hash table
- anv: Set STATE_BASE_ADDRESS upper bounds on gen7

Kenneth Graunke (1):

- glsl: Fix out of bounds read in shader_cache_read_program_metadata

Kevin Strasser (2):

- gallium/winsys/kms: Fix dumb buffer bpp
- st/mesa: Add rgbx handling for fp formats

Lionel Landwerlin (2):

- intel/perf: fix EuThreadsCount value in performance equations
- intel/perf: improve dynamic loading config detection

Mathias Fröhlich (1):

- egl: Don't add hardware device if there is no render node v2.

Nanley Chery (1):

- anv/cmd_buffer: Initalize the clear color struct for CNL+

Nataraj Deshpande (1):

- anv: Fix check for isl_fmt in assert

Samuel Pitoiset (5):

- radv: fix alpha-to-coverage when there is unused color attachments
- radv: fix setting CB_SHADER_MASK for dual source blending
- radv: fix occlusion queries on VegaM
- radv: fix VK_EXT_memory_budget if one heap isn't available
- radv: fix FMASK expand with SRGB formats

4.48 Mesa 19.1.0 Release Notes / June 11, 2019

Mesa 19.1.0 is a new development release. People who are concerned with stability and reliability should stick with a previous release or wait for Mesa 19.1.1.

Mesa 19.1.0 implements the OpenGL 4.5 API, but the version reported by glGetString(GL_VERSION) or glGetIntegerv(GL_MAJOR_VERSION) / glGetIntegerv(GL_MINOR_VERSION) depends on the particular driver being used. Some drivers don't support all the features required in OpenGL 4.5. OpenGL 4.5 is **only** available if requested at context creation. Compatibility contexts may report a lower version depending on each driver.

4.48.1 SHA256 checksums

```
2a6c3af3a803389183168e449c536304cf03e0f82c4c9333077933543b9d02f3  mesa-19.1.0.tar.xz
```

4.48.2 New features

- GL_ARB_parallel_shader_compile on all drivers.
- GL_EXT_gpu_shader4 on all GL 3.1 drivers.
- GL_EXT_shader_image_load_formatted on radeonsi.
- GL_EXT_texture_buffer_object on all GL 3.1 drivers.
- GL_EXT_texture_compression_s3tc_srgb on Gallium drivers and i965 (ES extension).
- GL_NV_compute_shader_derivatives on iris and i965.
- GL_KHR_parallel_shader_compile on all drivers.
- VK_EXT_buffer_device_address on Intel and RADV.
- VK_EXT_depth_clip_enable on Intel and RADV.
- VK_KHR_ycbcr_image_arrays on Intel.
- VK_EXT_inline_uniform_block on Intel and RADV.
- VK_EXT_external_memory_host on Intel.
- VK_EXT_host_query_reset on Intel and RADV.
- VK_KHR_surface_protected_capabilities on Intel and RADV.

- VK_EXT_pipeline_creation_feedback on Intel and RADV.
- VK_KHR_8bit_storage on RADV.
- VK_AMD_gpu_shader_int16 on RADV.
- VK_AMD_gpu_shader_half_float on RADV.
- VK_NV_compute_shader_derivatives on Intel.
- VK_KHR_shader_float16_int8 on Intel and RADV (RADV only supports int8).
- VK_KHR_shader_atomic_int64 on Intel.
- VK_EXT_descriptor_indexing on Intel.
- VK_KHR_shader_float16_int8 on Intel and RADV.
- GL_INTEL_conservative_rasterization on iris.
- VK_EXT_memory_budget on Intel.

4.48.3 Bug fixes

- Bug 81843 - [SNB IVB HSW] ETC2 textures are not returned as compressed images
- Bug 99781 - Some Unity games fail assertion on startup in glXCreateContextAttribsARB
- Bug 100239 - Incorrect rendering in CS:GO
- Bug 100316 - Linking GLSL 1.30 shaders with invariant and deprecated variables triggers an 'mismatching invariant qualifiers' error
- Bug 104272 - [OpenGL CTS] [HSW] KHR-GL46.direct_state_access.textures_compressed_subimage assert fails
- Bug 104355 - Ivy Bridge ignores component mappings in texture views
- Bug 104602 - [apitrace] Graphical artifacts in Civilization VI on RX Vega
- Bug 107052 - [Regression][bisected]. Crookz - The Big Heist Demo can't be launched despite the "true" flag in "drirc"
- Bug 107505 - [lars] dEQP-GLES31.functional.geometry_shading.layered#render_with_default_layer_3d failure
- Bug 107510 - [GEN8+] up to 10% perf drop on several 3D benchmarks
- Bug 107563 - [RADV] Broken rendering in Unity demos
- Bug 107987 - [Debug mesa only]. Crash happens when calling drawArrays
- Bug 108250 - [GLSL] layout-location-struct.shader_test fails to link
- Bug 108457 - [OpenGL CTS] KHR-GL46.tessellation_shader.single.xfb_captures_data_from_correct_stage fails
- Bug 108540 - vkAcquireNextImageKHR blocks when timeout=0 in Wayland
- Bug 108766 - Mesa built with meson has RPATH entries
- Bug 108824 - Invalid handling when GL buffer is bound on one context and invalidated on another
- Bug 108841 - [RADV] SPIRV's control flow attributes do not propagate to LLVM
- Bug 108879 - [CIK] [regression] All opencl apps hangs indefinitely in si_create_context

- Bug 108999 - Calculating the scissors fields when the y is flipped (0 on top) can generate negative numbers that will cause assertion failure later on.
- Bug 109057 - texelFetch from GL_TEXTURE_2D_MULTISAMPLE with integer format fails
- Bug 109107 - gallium/st/va: change va max_profiles when using Radeon VCN Hardware
- Bug 109216 - 4-27% performance drop in Vulkan benchmarks
- Bug 109326 - mesa: Meson configuration summary should be printed
- Bug 109328 - [BSW BXT GLK] dEQP-VK.subgroups.arithmetic.subgroup regressions
- Bug 109391 - LTO Build fails
- Bug 109401 - [DXVK] Project Cars rendering problems
- Bug 109404 - [ANV] The Witcher 3 shadows flickering
- Bug 109443 - Build failure with MSVC when using Scons >= 3.0.2
- Bug 109451 - [IVB,SNB] LINE_STRIPs following a TRIANGLE_FAN fail to use primitive restart
- Bug 109543 - After upgrade mesa to 19.0.0~rc1 all vulkan based application stop working ["vulkan-cube" received SIGSEGV in radv_pipeline_init_blend_state at ../src/amd/vulkan/radv_pipeline.c:699]
- Bug 109561 - [regression, bisected] code re-factor causing games to stutter or lock-up system
- Bug 109573 - dEQP-VK.spirv_assembly.instruction.graphics.module.same_module
- Bug 109575 - Mesa-19.0.0-rc1 : Computer Crashes trying to run anything Vulkan
- Bug 109581 - [BISECTED] Nothing is Rendered on Sascha Willem's "subpasses" demo
- Bug 109594 - totem assert failure: totem: src/intel/genxml/gen9_pack.h:72: __gen_uint: La declaración 'v <= max' no se cumple.
- Bug 109597 - wreckfest issues with transparent objects & skybox
- Bug 109601 - [Regression] RuneLite GPU rendering broken on 18.3.x
- Bug 109603 - nir_instr_as_deref: Assertion 'parent && parent->type == nir_instr_type_deref' failed.
- Bug 109645 - build error on arm64: tegra_screen.c:33: /usr/include/xf86drm.h:41:10: fatal error: drm.h: No such file or directory
- Bug 109646 - New video compositor compute shader render glitches mpv
- Bug 109647 - /usr/include/xf86drm.h:40:10: fatal error: drm.h: No such file or directory
- Bug 109648 - AMD Raven hang during va-api decoding
- Bug 109659 - Missing OpenGL symbols in OSMesa Gallium when building with meson
- Bug 109698 - dri.pc contents invalid when built with meson
- Bug 109717 - [regression] Cull distance tests asserting
- Bug 109735 - [Regression] broken font with mesa_vulkan_overlay
- Bug 109738 - Child of Light shows only a black screen
- Bug 109739 - Mesa build fails when vulkan-overlay-layer option is enabled
- Bug 109742 - vdpau state tracker on nv92 started to hit assert after vl compute work
- Bug 109743 - Test fails: piglit.spec.arb_sample_shading.arb_sample_shading-builtin-gl-sample-mask-mrt-alpha
- Bug 109747 - Add framerate to vulkan-overlay-layer

- Bug 109759 - [BISECTED][REGRESSION][IVB, HSW] Font rendering problem in OpenGL
- Bug 109788 - vulkan-overlay-layer: Only installs 64bit version
- Bug 109810 - nir_opt_copy_prop_vars.c:454: error: unknown field 'ssa' specified in initializer
- Bug 109929 - tgsi_to_nir.c:2111: undefined reference to 'gl_nir_lower_samplers_as_deref'
- Bug 109944 - [bisected] Android build test fails with: utils.c: error: use of undeclared identifier 'PACKAGE_VERSION'
- Bug 109945 - pan_assemble.c:51:46: error: passing argument 2 of 'tgsi_to_nir' from incompatible pointer type [-Werror=incompatible-pointer-types]
- Bug 109980 - [i915 CI][HSW] spec@arb_fragment_shader_interlock@arb_fragment_shader_interlock-image-load-store - fail
- Bug 109984 - unhandled VkStructureType VK_STRUCTURE_TYPE_RENDER_PASS_INPUT_ATTACHMENT_ASPECT_CR
- Bug 110134 - SIGSEGV while playing large hevc video in mpv
- Bug 110143 - Doom 3: BFG Edition - Steam and GOG.com - white flickering screen
- Bug 110201 - [ivb] mesa 19.0.0 breaks rendering in kitty
- Bug 110211 - If DESTDIR is set to an empty string, the dri drivers are not installed
- Bug 110216 - radv: Segfault when compiling compute shaders from Assassin's Creed Odyssey (regression, bisected)
- Bug 110221 - build error with meson
- Bug 110239 - Mesa SIGABRT: src/intel/genxml/gen9_pack.h:72: __gen_uint: Assertion 'v <= max' failed
- Bug 110257 - Major artifacts in mpeg2 vaapi hw decoding
- Bug 110259 - radv: Sampling depth-stencil image in GENERAL layout returns nothing but zero (regression, bisected)
- Bug 110291 - Vega 64 GPU hang running Space Engineers
- Bug 110302 - [bisected][regression] piglit egl-create-pbuffer-surface and egl-gl-colorspace regressions
- Bug 110305 - Iris driver fails ext_packed_depth_stencil-getteximage test
- Bug 110311 - [IVB HSW SNB][regression][bisected] regressions on vec4 deqp/gl{es}cts tests
- Bug 110349 - radv: Dragon Quest XI (DXVK) has a graphical glitch (regression, bisected)
- Bug 110353 - weird colors seen in valley
- Bug 110355 - radeonsi: GTK elements become invisible in some applications (GIMP, LibreOffice)
- Bug 110356 - install_megadrivers.py creates new dangling symlink [bisected]
- Bug 110404 - Iris fails piglit.spec.ext_transform_feedback.immediate-reuse test
- Bug 110422 - AMD_DEBUG=forcedma will crash OpenGL aps with SIGFAULT on VegaM 8706G
- Bug 110441 - [llvmpipe] complex-loop-analysis-bug regression
- Bug 110443 - vaapi/vpp: wrong output for non 64-bytes align width (ex: 1200)
- Bug 110454 - [llvmpipe] piglit arb_color_buffer_float-render GL_RGBA8_SNORM failure with llvm-9
- Bug 110462 - Epic Games Launcher renders nothing with "-opengl" option
- Bug 110474 - [bisected][regression] vk cts fp16 arithmetic failures

- Bug 110497 - [DXVK][Regression][Bisected][SKL] Project Cars 2 crashes with Bug Splat when loading finishes
- Bug 110526 - [CTS] dEQP-VK.ycbcr.{conversion,format}.* fail
- Bug 110530 - [CTS] dEQP-VK.ycbcr.format.g8_b8_r8_3plane_420* reports VM faults on Vega10
- Bug 110535 - [bisected] [icl] GPU hangs on crucible func.miptree.r8g8b8a8-unorm.aspect-color.view-2d.levels01.array01.extent-512x512.upload-copy-with-draw tests
- Bug 110540 - [AMD TAHITI XT] valve artifact broken
- Bug 110573 - Mesa vulkan-radeon 19.0.3 system freeze and visual artifacts (RADV)
- Bug 110590 - [Regression][Bisected] GTA under wine fails with GLXBadFBConfig
- Bug 110632 - "glx: Fix synthetic error generation in __glXSendError" broke wine games on 32-bit
- Bug 110648 - Dota2 will not open using vulkan since 19.0 series
- Bug 110655 - VK_LAYER_MESA_OVERLAY_CONFIG=draw,fps renders sporadically
- Bug 110698 - tu_device.c:900:4: error: initializer element is not constant
- Bug 110701 - GPU faults in in Unigine Valley 1.0
- Bug 110721 - graphics corruption on steam client with mesa 19.1.0 rc3 on polaris
- Bug 110761 - Huge problems between Mesa and Electron engine apps
- Bug 110784 - [regression][bisected] Reverting 'expose 0 shader binary formats for compat profiles for Qt' causes get_program_binary failures on Iris

4.48.4 Changes

Adam Jackson (1):

- drisw: Try harder to probe whether MIT-SHM works

Albert Pal (1):

- Fix link release notes for 19.0.0.

Alejandro Piñeiro (12):

- blorp: introduce helper method blorp_nir_init_shader
- nir, glsl: move pixel_center_integer/origin_upper_left to shader_info.fs
- nir/xfb: add component_offset at nir_xfb_info
- nir_types: add glsl_varying_count helper
- nir/xfb: adding varyings on nir_xfb_info and gather_info
- nir/xfb: sort varyings too
- nir_types: add glsl_type_is_struct helper
- nir/xfb: handle arrays and AoA of basic types
- nir/linker: use nir_gather_xfb_info
- nir/linker: fix ARRAY_SIZE query with xfb varyings
- nir/xfb: move varyings info out of nir_xfb_info
- docs: document MESA_GLSL=errors keyword

Alexander von Gluck IV (1):

- haiku: Fix hgl dispatch build. Tested under meson/scons.

Alexandros Frantzis (1):

- virgl: Fake MSAA when max samples is 1

Alok Hota (32):

- swr/rast: update SWR rasterizer shader stats
- gallium/swr: Param defaults for unhandled PIPE_CAPs
- gallium/aux: add PIPE_CAP_MAX_VARYINGS to u_screen
- swr/rast: Convert system memory pointers to gfxptr_t
- swr/rast: Disable use of __forceinline by default
- swr/rast: Correctly align 64-byte spills/fills
- swr/rast: Flip BitScanReverse index calculation
- swr/rast: Move knob defaults to generated cpp file
- swr/rast: FP consistency between POSH/RENDER pipes
- swr/rast: Refactor scratch space variable names
- swr/rast: convert DWORD->uint32_t, QWORD->uint64_t
- swr/rast: simdlib cleanup, clipper stack space fixes
- swr/rast: Add translation support to streamout
- swr/rast: bypass size limit for non-sampled textures
- swr/rast: Cleanup and generalize gen_archrast
- swr/rast: Add initial SWTag proto definitions
- swr/rast: Add string handling to AR event framework
- swr/rast: Add general SWTag statistics
- swr/rast: Fix autotools and scons codegen
- swr/rast: Remove deprecated 4x2 backend code
- swr/rast: AVX512 support compiled in by default
- swr/rast: enforce use of tile offsets
- swr/rast: add more llvm intrinsics
- swr/rast: update guardband rects at draw setup
- swr/rast: add SWR_STATIC_ASSERT() macro
- swr/rast: add flat shading
- swr/rast: add guards for cpuid on Linux
- swr/rast: early exit on empty triangle mask
- swr/rast: Cleanup and generalize gen_archrast
- swr/rast: Add initial SWTag proto definitions
- swr/rast: Add string handling to AR event framework

- swr/rast: Add general SWTag statistics

Alyssa Rosenzweig (192):

- panfrost: Initial stub for Panfrost driver
- panfrost: Implement Midgard shader toolchain
- meson: Remove panfrost from default driver list
- kmsro: Move DRM entrypoints to shared block
- panfrost: Use u_pipe_screen_get_param_defaults
- panfrost: Check in sources for command stream
- panfrost: Include glue for out-of-tree legacy code
- kmsro: Silence warning if missing
- panfrost: Clean-up one-argument passing quirk
- panfrost: Don't hardcode number of nir_ssa_defs
- panfrost: Add kernel-agnostic resource management
- panfrost: Remove if 0'd dead code
- panfrost: Remove speculative if 0'd format bit code
- panfrost: Elucidate texture op scheduling comment
- panfrost: Specify supported draw modes per-context
- panfrost: Fix build; depend on libdrm
- panfrost: Backport driver to Mali T600/T700
- panfrost: Identify MALI_OCCLUSION_PRECISE bit
- panfrost: Implement PIPE_QUERY_OCCLUSION_COUNTER
- panfrost: Don't align framebuffer dims
- panfrost: Improve logging and patch memory leaks
- panfrost: Fix various leaks unmapping resources
- panfrost: Free imported BOs
- panfrost: Swap order of tiled texture (de)alloc
- panfrost: Cleanup mali_viewport (clipping) code
- panfrost: Preserve w sign in perspective division
- panfrost: Fix clipping region
- panfrost: Stub out separate stencil functions
- panfrost: Add pandecode (command stream debugger)
- panfrost: Implement pantrace (command stream dump)
- panfrost/midgard: Refactor tag lookahead code
- panfrost/midgard: Fix nested/chained if-else
- panfrost: Rectify doubleplusungood extended branch
- panfrost/midgard: Emit extended branches

- panfrost: Dynamically set discard branch targets
- panfrost: Verify and print brx condition in disasm
- panfrost: Use tiler fast path (performance boost)
- panfrost/meson: Remove subdir for nondrm
- panfrost/nondrm: Flag CPU-invisible regions
- panfrost/nondrm: Make COHERENT_LOCAL explicit
- panfrost/nondrm: Split out dump_counters
- panfrost/midgard: Add fround(_even), ftrunc, ffma
- panfrost: Decode render target swizzle/channels
- panfrost: Add RGB565, RGB5A1 texture formats
- panfrost: Identify 4-bit channel texture formats
- panfrost: Expose perf counters in environment
- panfrost/midgard: Allow flt to run on most units
- panfrost: Import job data structures from v3d
- panfrost: Decouple Gallium clear from FBD clear
- panfrost: Cleanup cruft related to clears
- panfrost/midgard: Don't force constant on VLUT
- panfrost: Flush with offscreen rendering
- panfrost/midgard: Promote smul to vmul
- panfrost/midgard: Preview for data hazards
- panfrost: List primitive restart enable bit
- panfrost/drm: Cast pointer to u64 to fix warning
- panfrost: Cleanup needless if in create_bo
- panfrost: Combine has_afbc/tiled in layout enum
- panfrost: Delay color buffer setup
- panfrost: Determine framebuffer format bits late
- panfrost: Allocate dedicated slab for linear BOs
- panfrost: Support linear depth textures
- panfrost: Document "depth-buffer writeback" bit
- panfrost: Identify fragment_extra flags
- util: Add a drm_find_modifier helper
- v3d: Use shared drm_find_modifier util
- vc4: Use shared drm_find_modifier util
- freedreno: Use shared drm_find_modifier util
- panfrost: Break out fragment to SFBD/MFBD files
- panfrost: Remove staging SFBD for pan_context

- panfrost: Remove staging MFBD
- panfrost: Minor comment cleanup (version detection)
- panfrost/mfbd: Implement linear depth buffers
- panfrost/mfbd: Respect per-job depth write flag
- panfrost: Comment spelling fix
- panfrost: Allocate extra data for depth buffer
- panfrost; Disable AFBC for depth buffers
- panfrost: Compute viewport state on the fly
- panfrost/midgard: Implement fpow
- panfrost: Workaround buffer overrun with mip level
- panfrost: Fix primconvert check
- panfrost: Disable PIPE_CAP_TGSI_TEXCOORD
- panfrost/decode: Respect primitive size pointers
- panfrost: Replay more varying buffers
- panfrost: Rewrite varying assembly
- panfrost/midgard: Fix b2f32 swizzle for vectors
- panfrost: Fix viewports
- panfrost: Implement scissor test
- panfrost/midgard: Add fcsel_i opcode
- panfrost/midgard: Schedule ball/bany to vectors
- panfrost/midgard: Add more ball/bany, iabs ops
- panfrost/midgard: Map more bany/ball opcodes
- panfrost/midgard: Lower bool_to_int32
- panfrost/midgard: Lower f2b32 to fne
- panfrost/midgard: Lower i2b32
- panfrost/midgard: Implement b2i; improve b2f/f2b
- panfrost/midgard: Lower source modifiers for ints
- panfrost/midgard: Cleanup midgard_nir_algebraic.py
- panfrost: Stub out ES3 caps/callbacks
- panfrost/midgard: Add ult/ule ops
- panfrost/midgard: Expand fge lowering to more types
- panfrost/midgard: Handle i2b constant
- panfrost/midgard: fpow is a two-part operation
- panfrost: Preliminary work for mipmaps
- panfrost: Fix vertex buffer corruption
- panfrost/midgard: Disassemble ‘cube‘ texture op

- panfrost/midgard: Add L/S op for writing cubemap coordinates
- panfrost: Preliminary work for cubemaps
- panfrost/decode: Decode all cubemap faces
- panfrost: Include all cubemap faces in bitmap list
- panfrost/midgard: Emit cubemap coordinates
- panfrost: Implement command stream for linear cubemaps
- panfrost: Extend tiling for cubemaps
- panfrost: Implement missing texture formats
- panfrost/decode: Print negative_start
- panfrost: Clean index state between indexed draws
- panfrost: Fix index calculation types and asserts
- panfrost: Implement FIXED formats
- panfrost: Remove support for legacy kernels
- nir: Add "viewport vector" system values
- panfrost: Implement system values
- panfrost: Cleanup some indirection in pan_resource
- panfrost: Respect box->width in tiled stores
- panfrost: Size tiled temp buffers correctly
- panfrost/decode: Add flags for tilebuffer readback
- panfrost: Add tilebuffer load? branch
- panfrost/midgard: Add umin/umax opcodes
- panfrost/midgard: Add ilzcnt op
- panfrost/midgard: Add ibitcount8 op
- panfrost/midgard: Enable lower_find_lsb
- panfrost: Remove "mali_unknown6" nonsense
- panfrost/midgard: Drop dependence on mesa/st
- panfrost: Cleanup indexed draw handling
- nir: Add nir_lower_viewport_transform
- panfrost/midgard: Use shared nir_lower_viewport_transform
- panfrost: Track BO lifetime with jobs and reference counts
- panfrost: Fixup vertex offsets to prevent shadow copy
- panfrost/mdg: Use shared fsign lowering
- panfrost/mdg/disasm: Print raw varying_parameters
- panfrost/midgard: Pipe through varying arrays
- panfrost/midgard: Implement indirect loads of varyings/UBOs
- panfrost/midgard: Respect component of bcsel condition

- panfrost/midgard: Remove useless MIR dump
- panfrost: Respect backwards branches in RA
- panfrost/midgard: Don't try to inline constants on branches
- panfrost/midgard: imul can only run on *mul
- panfrost: Disable indirect outputs for now
- panfrost: Use actual imov instruction
- panfrost/midgard: Dead code eliminate MIR
- panfrost/midgard: Track loop depth
- panfrost/midgard: Fix off-by-one in successor analysis
- panfrost/midgard: Remove unused mir_next_block
- panfrost/midgard: Update integer op list
- panfrost/midgard: Document sign-extension/zero-extension bits (vector)
- panfrost/midgard: Set integer mods
- panfrost/midgard: Implement copy propagation
- panfrost/midgard: Optimize MIR in progress loop
- panfrost/midgard: Refactor opcode tables
- panfrost/midgard: Add "op commutes?" property
- panfrost/midgard: Remove assembler
- panfrost/midgard: Reduce fmax(a, 0.0) to fmov.pos
- panfrost/midgard: Extend copy propagation pass
- panfrost/midgard: Optimize csel involving 0
- panfrost/midgard: Copy prop for texture registers
- panfrost/midgard: Identify inand
- panfrost/midgard: Add new bitwise ops
- Revert "panfrost/midgard: Extend copy propagation pass"
- panfrost/midgard: Only copyprop without an outmod
- panfrost/midgard: Fix regressions in -bjellyfish
- panfrost/midgard: Fix tex propogation
- panfrost/midgard: imov workaround
- panfrost: Use fp32 (not fp16) varyings
- panfrost/midgard: Safety check immediate precision degradations
- panfrost: Workaround -bshadow regression
- panfrost: Remove shader dump
- panfrost/decode: Hit MRT blend shader enable bits
- panfrost: Fix blend shader upload
- panfrost/midgard: reg_mode_full -> reg_mode_32, etc

- panfrost/midgard/disasm: Catch mask errors
- panfrost/midgard/disasm: Extend print_reg to 8-bit
- panfrost/midgard/disasm: Fill in .int mod
- panfrost/midgard: Fix crash on unknown op
- panfrost/midgard: Rename ilzcnt8 -> iclz
- panfrost/midgard/disasm: Support 8-bit destination
- panfrost/midgard/disasm: Print 8-bit sources
- panfrost/midgard/disasm: Stub out 64-bit
- panfrost/midgard/disasm: Handle dest_override generalized
- panfrost: Support RGB565 FBOs
- panfrost/midgard: Fix integer selection
- panfrost/midgard: Fix RA when temp_count = 0
- panfrost/midgard: Lower mixed csel (NIR)
- panfrost/midgard: iabs cannot run on mul

Alyssa Ross (1):

- get_reviewer.pl: improve portability

Amit Pundir (1):

- mesa: android: freedreno: build libfreedreno_{drm,ir3} static libs

Andre Heider (5):

- iris: fix build with gallium nine
- iris: improve PIPE_CAP_VIDEO_MEMORY bogus value
- iris: add support for tgsi_to_nir
- st/nine: enable csmt per default on iris
- st/nine: skip position checks in SetCursorPosition()

Andreas Baierl (2):

- nir: add rcp(w) lowering for gl_FragCoord
- lima/ppir: Add gl_FragCoord handling

Andres Gomez (12):

- mesa: INVALID_VALUE for wrong type or format in Clear*Buffer*Data
- gitlab-ci: install distro's ninja
- glsl: correctly validate component layout qualifier for dvec{3,4}
- glsl/linker: always validate explicit location among inputs
- glsl/linker: don't fail non static used inputs without matching outputs
- glsl/linker: simplify xfb_offset vs xfb_stride overflow check
- Revert "glsl: relax input->output validation for SSO programs"
- glsl/linker: location aliasing requires types to have the same width

- docs: drop Andres Gomez from the release cycles
- glsl/linker: always validate explicit locations for first and last interfaces
- docs/relnotes: add support for VK_KHR_shader_float16_int8
- glsl/linker: check for xfb_offset aliasing

Andrii Simiklit (5):

- i965: consider a 'base level' when calculating width0, height0, depth0
- i965: re-emit index buffer state on a reset option change.
- util: clean the 24-bit unused field to avoid an issues
- iris: make the TFB result visible to others
- egl: return correct error code for a case req ver < 3 with forward-compatible

Antia Puentes (1):

- nir/linker: Fix TRANSFORM_FEEDBACK_BUFFER_INDEX

Anuj Phogat (7):

- i965/icl: Add WA_2204188704 to disable pixel shader panic dispatch
- anv/icl: Add WA_2204188704 to disable pixel shader panic dispatch
- intel: Add Elkhart Lake device info
- intel: Add Elkhart Lake PCI-IDs
- iris/icl: Set Enabled Texel Offset Precision Fix bit
- iris/icl: Add WA_2204188704 to disable pixel shader panic dispatch
- intel: Add support for Comet Lake

Axel Davy (49):

- st/nine: Ignore window size if error
- st/nine: Ignore multisample quality level if no ms
- st/nine: Disable depth write when nothing gets updated
- st/nine: Do not advertise support for D15S1 and D24X4S4
- st/nine: Do not advertise CANMANAGERESOURCE
- st/nine: Change a few advertised caps
- Revert "d3dadapter9: Support software renderer on any DRI device"
- st/nine: Fix D3DWindowBuffer_release for old wine nine support
- st/nine: Use FLT_MAX/2 for RCP clamping
- st/nine: Upload managed textures only at draw using them
- st/nine: Upload managed buffers only at draw using them
- st/nine: Fix buffer/texture unbinding in nine_state_clear
- st/nine: Finish if nooverwrite after normal mapping
- st/nine: Always return OK on SetSoftwareVertexProcessing
- st/nine: Enable modifiers on ps 1.X texcoords

- st/nine: Ignore nooverwrite for systemmem
- st/nine: Fix SINCOS input
- st/nine: Optimize surface upload with conversion
- st/nine: Optimize volume upload with conversion
- st/nine: rename *_conversion to *_internal
- st/nine: Refactor surface GetSystemMemPointer
- st/nine: Refactor volume GetSystemMemPointer
- st/nine: Support internal compressed format for surfaces
- st/nine: Support internal compressed format for volumes
- st/nine: Add drirc option to use data_internal for dynamic textures
- drirc: Add Gallium nine workaround for Rayman Legends
- st/nine: Recompile optimized shaders based on b/i consts
- st/nine: Control shader constant inlining with drirc
- st/nine: Regroup param->rel tests
- st/nine: Refactor param->rel
- st/nine: Compact nine_ff_get_projected_key
- st/nine: Compact pixel shader key
- st/nine: use helper ureg_DECL_sampler everywhere
- st/nine: Manually upload vs and ps constants
- st/nine: Refactor shader constants ureg_src computation
- st/nine: Make swvp_on imply IS_VS
- st/nine: Refactor ct_ctor
- st/nine: Track constant slots used
- st/nine: Refactor counting of constants
- st/nine: Prepare constant compaction in nine_shader
- st/nine: Propagate const_range to context
- st/nine: Cache constant buffer size
- st/nine: Handle const_ranges in nine_state
- st/nine: Enable computing const_ranges
- st/nine: Use TGSI_SEMANTIC_GENERIC for fog
- st/nine: Optimize a bit writeonly buffers
- st/nine: Throttle rendering similarly for thread_submit
- st/nine: Check discard_delayed_release is set before allocating more
- d3dadapter9: Revert to old throttling limit value

Bart Oldeman (1):

- gallium-xlib: query MIT-SHM before using it.

Bas Nieuwenhuizen (105):

- radv: Only look at pImmutableSamples if the descriptor has a sampler.
- amd/common: Add gep helper for pointer increment.
- amd/common: Implement ptr->int casts in ac_to_integer.
- radv: Fix the shader info pass for not having the variable.
- amd/common: Use correct writemask for shared memory stores.
- amd/common: Fix stores to derefs with unknown variable.
- amd/common: Handle nir_deref_type_ptr_as_array for shared memory.
- amd/common: handle nir_deref_cast for shared memory from integers.
- amd/common: Do not use 32-bit loads for shared memory.
- amd/common: Implement global memory accesses.
- radv: Do not use the bo list for local buffers.
- radv: Implement VK_EXT_buffer_device_address.
- radv: Use correct num formats to detect whether we should be use 1.0 or 1.
- radv: Sync ETC2 whitelisted devices.
- radv: Clean up a bunch of compiler warnings.
- radv: Handle clip+cull distances more generally as compact arrays.
- radv: Implement VK_EXT_depth_clip_enable.
- radv: Disable depth clamping even without EXT_depth_range_unrestricted.
- radv: Fix float16 interpolation set up.
- radv: Allow interpolation on non-float types.
- radv: Interpolate less aggressively.
- turnip: Add driver skeleton (v2)
- turnip: Fix up detection of device.
- turnip: Gather some device info.
- turnip: Remove abort.
- turnip: Fix newly introduced warning.
- turnip: Add buffer allocation & mapping support.
- turnip: Report a memory type and heap.
- turnip: Cargo cult the Intel heap size functionality.
- turnip: Initialize memory type in requirements.
- turnip: Disable more features.
- turnip: Add 630 to the list.
- turnip: Fix bo allocation after we stopped using libdrm_freedreno . . .
- turnip: Fix memory mapping.
- turnip: Add image layout calculations.

- turnip: Stop hardcoding the msm version check.
- turnip: move tu_gem.c to tu_drm.c
- turnip: Implement pipe-less param query.
- turnip: Implement some format properties for RGBA8.
- turnip: Remove some radv leftovers.
- turnip: clean up TODO.
- turnip: Implement some UUIDs.
- turnip: Implement a slow bo list
- turnip: Add a command stream.
- turnip: Add msm queue support.
- turnip: Make bo_list functions not static
- turnip: Implement submission.
- turnip: Fill command buffer
- turnip: Shorten primary_cmd_stream name.
- turnip: Add emit functions in a header.
- turnip: Move stream functions to tu_cs.c
- turnip: Add buffer memory binding.
- turnip: Make tu6_emit_event_write shared.
- turnip: Add tu6_rb_fmt_to_ifmt.
- turnip: Implement buffer->buffer DMA copies.
- turnip: Add image->buffer DMA copies.
- turnip: Add buffer->image DMA copies.
- turnip: Add todo for copies.
- turnip: Fix GCC compiles.
- turnip: Deconflict vk_format_table regeneration
- gitlab-ci: Build turnip.
- radeonsi: Remove implicit const cast.
- radv: Allow fast clears with concurrent queue mask for some layouts.
- vulkan/util: Handle enums that are in platform-specific headers.
- vulkan: Update the XML and headers to 1.1.104
- radv: Implement VK_EXT_host_query_reset.
- radv: Use correct image view comparison for fast clears.
- radv: Implement VK_EXT_pipeline_creation_feedback.
- ac/nir: Return frag_coord as integer.
- nir: Add access qualifiers on load_ubo intrinsic.
- radv: Add non-uniform indexing lowering.

- radv: Add bolist RADV_PERFTEST flag.
- ac: Move has_local_buffers disable to radeonsi.
- radv: Use local buffers for the global bo list.
- radv: Support VK_EXT_inline_uniform_block.
- radv: Add support for driconf.
- vulkan/wsi: Add X11 adaptive sync support based on dri options.
- radv: Add adaptive_sync driconfig option and enable it by default.
- radv: Add logic for subsampled format descriptions.
- radv: Add logic for multisample format descriptions.
- radv: Add multiple planes to images.
- radv: Add single plane image views & meta operations.
- radv: Support different source & dest aspects for planar images in blit2d.
- radv: Add ycbcr conversion structs.
- radv: Add support for image views with multiple planes.
- radv: Allow mixed src/dst aspects in copies.
- ac/nir: Add support for planes.
- radv: Add ycbcr samplers in descriptor set layouts.
- radv: Update descriptor sets for multiple planes.
- radv: Add ycbcr lowering pass.
- radv: Run the new ycbcr lowering pass.
- radv: Add hashing for the ycbcr samplers.
- radv: Add ycbcr format features.
- radv: Add ycbcr subsampled & multiplane formats to csv.
- radv: Enable YCBCR conversion feature.
- radv: Expose VK_EXT_ycbcr_image_arrays.
- radv: Expose Vulkan 1.1 for Android.
- radv: Fix hang width YCBCR array textures.
- radv: Set is_array in lowered ycbcr tex instructions.
- radv: Restrict YUVY formats to 1 layer.
- radv: Disable subsampled formats.
- radv: Implement cosited_even sampling.
- radv: Do not use extra descriptor space for the 3rd plane.
- nir: Actually propagate progress in nir_opt_move_load_ubo.
- radv: Prevent out of bound shift on 32-bit builds.

Benjamin Gordon (1):

- configure.ac/meson.build: Add options for library suffixes

Benjamin Tissoires (1):

- CI: use wayland ci-templates repo to create the base image

Boyan Ding (3):

- gk110/ir: Add rcp f64 implementation
- gk110/ir: Add rsq f64 implementation
- gk110/ir: Use the new rcp/rsq in library

Boyuan Zhang (1):

- st/va: reverse qt matrix back to its original order

Brian Paul (51):

- st/mesa: whitespace/formatting fixes in st_cb_texture.c
- svga: assorted whitespace and formatting fixes
- svga: fix dma.pending > 0 test
- mesa: fix display list corner case assertion
- st/mesa: whitespace fixes in st_sampler_view.c
- st/mesa: line wrapping, whitespace fixes in st_cb_texture.c
- st/mesa: whitespace fixes in st_texture.h
- svga: init fill variable to avoid compiler warning
- svga: silence array out of bounds warning
- st/wgl: init a variable to silence MinGW warning
- gallium/util: whitespace cleanups in u_bitmask.[ch]
- gallium/util: add some const qualifiers in u_bitmask.c
- pipebuffer: use new pb_usage_flags enum type
- pipebuffer: whitespace fixes in pb_buffer.h
- winsys/svga: use new pb_usage_flags enum type
- st/mesa: move, clean-up shader variant key decls/inits
- st/mesa: whitespace, formatting fixes in st_cb_flush.c
- svga: refactor draw_vgpu10() function
- svga: remove SVGA_RELOC_READ flag in SVGA3D_BindGBSurface()
- pipebuffer: s/PB_ALL_USAGE_FLAGS/PB_USAGE_ALL/
- st/mesa: init hash keys with memset(), not designated initializers
- intel/decoders: silence uninitialized variable warnings in gen_print_batch()
- intel/compiler: silence unitialized variable warning in opt_vector_float()
- st/mesa: move utility functions, macros into new st_util.h file
- st/mesa: move around some code in st_context.c
- st/mesa: add/improve sampler view comments
- st/mesa: rename st_texture_release_sampler_view()

- st/mesa: minor refactoring of texture/sampler delete code
- docs: try to improve the Meson documentation (v2)
- drisw: fix incomplete type compilation failure
- gallium/winsys/kms: fix incomplete type compilation failure
- nir: silence a couple new compiler warnings
- docs: separate information for compiler selection and compiler options
- docs: link to the meson_options.txt file gitlab.freedesktop.org
- st/mesa: implement "zombie" sampler views (v2)
- st/mesa: implement "zombie" shaders list
- st/mesa: stop using pipe_sampler_view_release()
- svga: stop using pipe_sampler_view_release()
- llvmpipe: stop using pipe_sampler_view_release()
- swr: remove call to pipe_sampler_view_release()
- i915g: remove calls to pipe_sampler_view_release()
- gallium/util: remove pipe_sampler_view_release()
- nir: fix a few signed/unsigned comparison warnings
- st/mesa: fix texture deletion context mix-up issues (v2)
- nir: use {0} initializer instead of {} to fix MSVC build
- util: no-op __builtin_types_compatible_p() for non-GCC compilers
- docs: s/Aptril/April/
- llvmpipe: init some vars to NULL to silence MinGW compiler warnings
- glsl: work around MinGW 7.x compiler bug
- svga: add SVGA_NO_LOGGING env var (v2)
- glsl: fix typo in #warning message

Caio Marcelo de Oliveira Filho (61):

- nir: keep the phi order when splitting blocks
- i965: skip bit6 swizzle detection in Gen8+
- anv: skip bit6 swizzle detection in Gen8+
- isl: assert that Gen8+ don't have bit6_swizzling
- intel/compiler: use 0 as sampler in emit_mcs_fetch
- nir: fix example in opt_peel_loop_initial_if description
- iris: Fix uses of gl_TessLevel*
- iris: Add support for TCS passthrough
- iris: always include an extra constbuf0 if using UBOs
- nir/copy_prop_vars: don't get confused by array_deref of vectors
- nir/copy_prop_vars: add debug helpers

- nir/copy_prop_vars: keep track of components in copy_entry
- nir/copy_prop_vars: change test helper to get intrinsics
- nir: nir_build_deref_follower accept array derefs of vectors
- nir/copy_prop_vars: add tests for load/store elements of vectors
- nir: fix MSVC build
- st/nir: count num_uniforms for FS bultin shader
- nir/copy_prop_vars: rename/refactor store_to_entry helper
- nir/copy_prop_vars: use NIR_MAX_VEC_COMPONENTS
- nir/copy_prop_vars: handle load/store of vector elements
- nir/copy_prop_vars: add tests for indirect array deref
- nir/copy_prop_vars: prefer using entries from equal derefs
- nir/copy_prop_vars: handle indirect vector elements
- anv: Implement VK_EXT_external_memory_host
- nir: Add a pass to combine store_derefs to same vector
- intel/nir: Combine store_derefs after vectorizing IO
- intel/nir: Combine store_derefs to improve code from SPIR-V
- nir: Handle array-deref-of-vector case in loop analysis
- spirv: Add an execution environment to the options
- intel/compiler: handle GLSL_TYPE_INTERFACE as GLSL_TYPE_STRUCT
- spirv: Use interface type for block and buffer block
- iris: Clean up compiler warnings about unused
- nir: Take if_uses into account when repairing SSA
- mesa: Extension boilerplate for NV_compute_shader_derivatives
- glsl: Remove redundant conditions when asserting in_qualifier
- glsl: Enable derivative builtins for NV_compute_shader_derivatives
- glsl: Enable texture builtins for NV_compute_shader_derivatives
- glsl: Parse and propagate derivative_group to shader_info
- nir/algebraic: Lower CS derivatives to zero when no group defined
- nir: Don't set LOD=0 for compute shader that has derivative group
- intel/fs: Use TEX_LOGICAL whenever implicit lod is supported
- intel/fs: Add support for CS to group invocations in quads
- intel/fs: Don't loop when lowering CS intrinsics
- intel/fs: Use NIR_PASS_V when lowering CS intrinsics
- i965: Advertise NV_compute_shader_derivatives
- gallium: Add PIPE_CAP_COMPUTE_SHADER_DERIVATIVES
- iris: Enable NV_compute_shader_derivatives

- spirv: Add support for DerivativeGroup capabilities
- anv: Implement VK_NV_compute_shader_derivatives
- docs: Add NV_compute_shader_derivatives to 19.1.0 relnotes
- spirv: Add more to_string helpers
- spirv: Tell which opcode or value is unhandled when failing
- spirv: Rename vtn_decoration literals to operands
- spirv: Handle SpvOpDecorateId
- nir: Add option to lower tex to txl when shader don't support implicit LOD
- intel/fs: Don't handle texop_tex for shaders without implicit LOD
- spirv: Properly handle SpvOpAtomicCompareExchangeWeak
- intel/fs: Assert when brw_fs_nir sees a nir_deref_instr
- anv: Fix limits when VK_EXT_descriptor_indexing is used
- nir: Fix nir_opt_idiv_const when negatives are involved
- nir: Fix clone of nir_variable state slots

Carlos Garnacho (1):

- wayland/egl: Ensure EGL surface is resized on DRI update_buffers()

Chad Versace (17):

- turnip: Drop Makefile.am and Android.mk
- turnip: Fix indentation in function signatures
- turnip: Fix result of vkEnumerate*LayerProperties
- turnip: Fix result of vkEnumerate*ExtensionProperties
- turnip: Use vk_outarray in all relevant public functions
- turnip: Fix a real -Wmaybe-uninitialized
- turnip: Fix indentation
- turnip: Require DRM device version >= 1.3
- turnip: Add TODO for Android logging
- turnip: Use vk_errorf() for initialization error messages
- turnip: Replace fd_bo with tu_bo
- turnip: Add TODO file
- turnip: Fix 'unused' warnings
- turnip: Don't return from tu_stub funcs
- turnip: Annotate vkGetImageSubresourceLayout with tu_stub
- turnip: Fix error behavior for VkPhysicalDeviceExternalImageFormatInfo
- turnip: Use Vulkan 1.1 names instead of KHR

Charmaine Lee (5):

- svga: add svga shader type in the shader variant

- svga: move host logging to winsys
- st/mesa: purge framebuffers with current context after unbinding winsys buffers
- mesa: unreference current winsys buffers when unbinding winsys buffers
- svga: Remove unnecessary check for the pre flush bit for setting vertex buffers

Chenglei Ren (1):

- anv/android: fix missing dependencies issue during parallel build

Chia-I Wu (78):

- egl: fix KHR_partial_update without EXT_buffer_age
- turnip: add .clang-format
- turnip: use msm_drm.h from inc_freedreno
- turnip: remove unnecessary libfreedreno_drm dep
- turnip: add wrappers around DRM_MSM_GET_PARAM
- turnip: add wrappers around DRM_MSM_SUBMITQUEUE_*
- turnip: constify tu_device in tu_gem_*
- turnip: preliminary support for tu_QueueWaitIdle
- turnip: run sed and clang-format on tu_cs
- turnip: document tu_cs
- turnip: add tu_cs_add_bo
- turnip: minor cleanup to tu_cs_end
- turnip: update cs->start in tu_cs_end
- turnip: inline tu_cs_check_space
- turnip: add more tu_cs helpers
- turnip: build drm_msm_gem_submit_bo array directly
- turnip: add tu_bo_list_merge
- turnip: add cmdbuf->bo_list to bo_list in queue submit
- turnip: preliminary support for tu_BindImageMemory2
- turnip: preliminary support for tu_image_view_init
- turnip: preliminary support for tu_CmdBeginRenderPass
- turnip: add tu_cs_reserve_space(_assert)
- turnip: emit HW init in tu_BeginCommandBuffer
- turnip: preliminary support for tu_GetRenderAreaGranularity
- turnip: add tu_tiling_config
- turnip: add internal helpers for tu_cs
- turnip: add tu_cs_{reserve,add}_entry
- turnip: specify initial size in tu_cs_init
- turnip: never fail tu_cs_begin/tu_cs_end

- turnip: add tu_cs_sanity_check
- turnip: provide both emit_ib and emit_call
- turnip: add tu_cs_mode
- turnip: add TU_CS_MODE_SUB_STREAM
- turnip: preliminary support for loadOp and storeOp
- turnip: add a more complete format table
- turnip: add functions to import/export prime fd
- turnip: advertise VK_KHR_external_memory_capabilities
- turnip: advertise VK_KHR_external_memory
- turnip: add support for VK_KHR_external_memory_{fd,dma_buf}
- turnip: fix VkClearValue packing
- turnip: preliminary support for fences
- turnip: respect color attachment formats
- turnip: mark IBs for dumping
- turnip: use 32-bit offset in tu_cs_entry
- turnip: more/better asserts for tu_cs
- turnip: add tu_cs_discard_entries
- turnip: tu_cs_emit_array
- turnip: fix tu_cs sub-streams
- turnip: simplify tu_cs sub-streams usage
- turnip: create a less dummy pipeline
- turnip: parse VkPipelineDynamicStateCreateInfo
- turnip: parse VkPipelineInputAssemblyStateCreateInfo
- turnip: parse VkPipelineViewportStateCreateInfo
- turnip: parse VkPipelineRasterizationStateCreateInfo
- turnip: parse VkPipelineDepthStencilStateCreateInfo
- turnip: parse VkPipeline{Multisample,ColorBlend}StateCreateInfo
- turnip: preliminary support for shader modules
- turnip: compile VkPipelineShaderStageCreateInfo
- turnip: parse VkPipelineShaderStageCreateInfo
- turnip: parse VkPipelineVertexInputStateCreateInfo
- turnip: add draw_cs to tu_cmd_buffer
- turnip: preliminary support for draw state binding
- turnip: preliminary support for tu_CmdDraw
- turnip: guard -Dvulkan-driver=freedreno
- turnip: preliminary support for tu_GetImageSubresourceLayout

- turnip: preliminary support for Wayland WSI
- vulkan/wsi: move modifier array into wsi_wl_swapchain
- vulkan/wsi: create wl_drm wrapper as needed
- vulkan/wsi: refactor drm_handle_format
- vulkan/wsi: add wsi_wl_display_drm
- vulkan/wsi: add wsi_wl_display_dmabuf
- vulkan/wsi: make wl_drm optional
- virgl: handle fence_server_sync in winsys
- virgl: hide fence internals from the driver
- virgl: introduce virgl_drm_fence
- virgl: fix fence fd version check
- virgl: clear vertex_array_dirty
- virgl: skip empty cmdbufs

Chris Forbes (3):

- glsl: add scaffolding for EXT_gpu_shader4
- glsl: enable noperspective|flat|centroid for EXT_gpu_shader4
- glsl: enable types for EXT_gpu_shader4

Chris Wilson (19):

- i965: Assert the execobject handles match for this device
- iris: fix import from dri2/3
- iris: IndexFormat = size/2
- iris: Set resource modifier on handle
- iris: Wrap userptr for creating bo
- iris: AMD_pinned_memory
- iris: Record reusability of bo on construction
- iris: fix memzone_for_address since multibinder changes
- iris: Tidy exporting the flink handle
- iris: Fix assigning the output handle for exporting for KMS
- iris: Merge two walks of the exec_bos list
- iris: Tag each submitted batch with a syncobj
- iris: Add fence support using drm_syncobj
- iris: Wire up EGL_IMG_context_priority
- iris: Use PIPE_BUFFER_STAGING for the query objects
- iris: Use coherent allocation for PIPE_RESOURCE_STAGING
- iris: Use streaming loads to read from tiled surfaces
- iris: Push heavy memchecker code to DEBUG

- iris: Adapt to variable ppGTT size

Christian Gmeiner (12):

- etnaviv: rs: mark used src resource as read from
- etnaviv: blt: mark used src resource as read from
- etnaviv: implement ETC2 block patching for HALTI0
- etnaviv: keep track of mapped bo address
- etnaviv: hook-up etc2 patching
- etnaviv: enable ETC2 texture compression support for HALTI0 GPUs
- etnaviv: fix resource usage tracking across different pipe_context's
- etnaviv: fix compile warnings
- st/dri: allow direct UYVY import
- etnaviv: shrink struct etna_3d_state
- nir: add lower_ftrunc
- etnaviv: use the correct uniform dirty bits

Chuck Atkins (1):

- meson: Fix missing glproto dependency for gallium-glx

Connor Abbott (6):

- nir/serialize: Prevent writing uninitialized state_slot data
- nir: Add a stripping pass for improved cacheability
- radeonsi/nir: Use nir stripping pass
- nir/search: Add automaton-based pre-searching
- nir/search: Add debugging code to dump the pattern matched
- nir/algebraic: Don't emit empty initializers for MSVC

Daniel Schürmann (2):

- nir: Define shifts according to SM5 specification.
- nir: Use SM5 properties to optimize shift(a@32, iand(31, b))

Daniel Stone (2):

- panfrost: Properly align stride
- vulkan/wsi/wayland: Respect non-blocking AcquireNextImage

Danylo Piliaiev (13):

- anv: Handle VK_ATTACHMENT_UNUSED in colorAttachment
- radv: Handle VK_ATTACHMENT_UNUSED in CmdClearAttachment
- anv: Fix VK_EXT_transform_feedback working with varyings packed in PSIZ
- anv: Fix destroying descriptor sets when pool gets reset
- anv: Treat zero size XFB buffer as disabled
- glsl: Cross validate variable's invariance by explicit invariance only

- i965,iris,anv: Make alpha to coverage work with sample mask
- intel/fs: Make alpha test work with MRT and sample mask
- st/mesa: Fix GL_MAP_COLOR with glDrawPixels GL_COLOR_INDEX
- iris: Fix assert when using vertex attrib without buffer binding
- intel/compiler: Do not reswizzle dst if instruction writes to flag register
- drirc: Add workaround for Epic Games Launcher
- anv: Do not emulate texture swizzle for INPUT_ATTACHMENT, STORAGE_IMAGE

Dave Airlie (63):

- virgl: enable elapsed time queries
- virgl: ARB_query_buffer_object support
- docs: update qbo support for virgl
- glsl: glsl to nir fix uninit class member.
- radv/llvm: initialise passes member.
- radv: remove alloc parameter from pipeline init
- iris: fix some hangs around null framebuffers
- iris: fix crash in sparse vertex array
- iris: add initial transform feedback overflow query paths (V3)
- iris: fix cube texture view
- iris: execute compute related query on compute batch.
- iris: iris add load register reg32/64
- iris: add conditional render support
- iris: fix gpu calcs for timestamp queries
- iris/WIP: add broadwell support
- iris: limit gen8 to 8 samples
- iris: setup gen8 caps
- iris: add fs invocations query workaround for broadwell
- iris: handle qbo fragment shader invocation workaround
- st/mesa: add support for lowering fp64/int64 for nir drivers
- softpipe: fix texture view crashes
- nir/spirv: don't use bare types, remove assert in split vars for testing
- nir/deref: remove casts of casts which are likely redundant (v3)
- softpipe: fix 32-bit bitfield extract
- softpipe: handle 32-bit bitfield inserts
- softpipe: remove shadow_ref assert.
- softpipe: fix integer texture swizzling for 1 vs 1.0f
- nir/split_vars: fixup some more explicit_stride related issues.

- draw: bail instead of assert on instance count (v2)
- draw/gs: fix point size outputs from geometry shader.
- draw/vs: partly fix basevertex/vertex id
- softpipe: fix clears to only clear specified color buffers.
- softpipe/draw: fix vertex id in soft paths.
- softpipe: add indirect store buffer/image unit
- nir/deref: fix struct wrapper casts. (v3)
- nir: use proper array sizing define for vectors
- intel/compiler: use defined size for vector components
- iris: avoid use after free in shader destruction
- ddebug: add compute functions to help hang detection
- draw: add stream member to stats callback
- tgsi: add support for geometry shader streams.
- softpipe: add support for indexed queries.
- draw: add support to tgsi paths for geometry streams. (v2)
- softpipe: add support for vertex streams (v2)
- virgl: add support for missing command buffer binding.
- virgl: add support for ARB_multi_draw_indirect
- virgl: add support for ARB_indirect_parameters
- draw: fix undefined shift of (1 << 31)
- swrast: fix undefined shift of 1 << 31
- llvmpipe: fix undefined shift 1 << 31.
- virgl/drm: cleanup buffer from handle creation (v2)
- virgl/drm: handle flink name better.
- virgl/drm: insert correct handles into the table. (v3)
- intel/compiler: fix uninit non-static variable. (v2)
- nir: fix bit_size in lower indirect derefs.
- r600: reset tex array override even when no view bound
- spirv: fix SpvOpBitSize return value.
- nir: fix lower vars to ssa for larger vector sizes.
- util/tests: add basic unit tests for bitset
- util/bitset: fix bitset range mask calculations.
- kmsro: add _dri.so to two of the kmsro drivers.
- glsl: init packed in more constructors.
- Revert "mesa: unreference current winsys buffers when unbinding winsys buffers"

David Riley (3):

- virgl: Store mapped hw resource with transfer object.
- virgl: Allow transfer queue entries to be found and extended.
- virgl: Re-use and extend queue transfers for intersecting buffer subdatas.

David Shao (1):

- meson: ensure that xmlpool_options.h is generated for gallium targets that need it

Deepak Rawat (2):

- winsys/drm: Fix out of scope variable usage
- winsys/svga/drm: Fix 32-bit RPCI send message

Dominik Drees (1):

- Add no_aos_sampling GALLIVM_PERF option

Drew Davenport (1):

- util: Don't block SIGSYS for new threads

Dylan Baker (40):

- bump version for 19.0 branch
- docs: Add relnotes stub for 19.1
- gallium: wrap u_screen in extern "C" for c++
- automake: Add –enable-autotools to distcheck flags
- android,autotools,i965: Fix location of float64_glsl.h
- meson: remove build_by_default : true
- meson: fix style in intel/tools
- meson: remove -std=c++11 from intel/tools
- get-pick-list: Add –pretty=medium to the arguments for Cc patches
- meson: Add dependency on genxml to anvil
- meson/iris: Use current coding style
- docs: Add release notes for 19.0.0
- docs: Add SHA256 sums for 19.0.0
- docs: update calendar, add news item, and link release notes for 19.0.0
- bin/install_megadrivers.py: Correctly handle DESTDIR=' '
- bin/install_megadrivers.py: Fix regression for set DESTDIR
- docs: Add release notes for 19.0.1
- docs: Add SHA256 sums for mesa 19.0.1
- docs: update calendar, add news item and link release notes for 19.0.1
- meson: Error if LLVM doesn't have rtti when building clover
- meson: Error if LLVM is turned off but clover it turned on
- docs: Add release notes for 19.0.2
- docs: Add sha256 sums for 19.0.2

- docs: update calendar, and news item and link release notes for 19.0.2
- Delete autotools
- docs: drop most autoconf references
- ci: Delete autotools build jobs
- docs: add relnotes for 19.0.3
- docs: Add SHA256 sums for mesa 19.0.3
- docs: update calendar, and news item and link release notes for 19.0.3
- meson: always define libglapi
- glsl: fix general_ir_test with mingw
- meson: switch gles1 and gles2 to auto options
- meson: Make shader-cache a trillean instead of boolean
- meson: make nm binary optional
- util/tests: Use define instead of VLA
- glsl/tests: define ssize_t on windows
- tests/vma: fix build with MSVC
- meson: Don't build glsl cache_test when shader cache is disabled
- meson: Force the use of config-tool for llvm

Eduardo Lima Mitev (5):

- freedreno/a6xx: Silence compiler warnings
- nir: Add ir3-specific version of most SSBO intrinsics
- ir3/nir: Add a new pass 'ir3_nir_lower_io_offsets'
- ir3/compiler: Enable lower_io_offsets pass and handle new SSBO intrinsics
- ir3/lower_io_offsets: Try propagate SSBO's SHR into a previous shift instruction

El Christianito (1):

- drirc: add Budgie WM to adaptive-sync blacklist

Eleni Maria Stea (6):

- i965: Faking the ETC2 compression on Gen < 8 GPUs using two miptrees.
- i965: Fixed the CopyImageSubData for ETC2 on Gen < 8
- i965: Enabled the OES_copy_image extension on Gen 7 GPUs
- i965: Removed the field etc_format from the struct intel_mipmap_tree
- i965: fixed clamping in set_scissor_bits when the y is flipped
- radv: consider MESA_VK_VERSION_OVERRIDE when setting the api version

Elie Tournier (3):

- virgl: Add a caps to advertise GLES backend
- virgl: Set PIPE_CAP_DOUBLES when running on GLES This is a lie but no known app use fp64.
- virgl: Return an error if we use fp64 on top of GLES

Emil Velikov (30):

- vc4: Declare the last cpu pointer as being modified in NEON asm.
- docs: add release notes for 18.3.3
- docs: add sha256 checksums for 18.3.3
- docs: update calendar, add news item and link release notes for 18.3.3
- anv: wire up the state_pool_padding test
- docs: add release notes for 18.3.4
- docs: add sha256 checksums for 18.3.4
- docs: update calendar, add news item and link release notes for 18.3.4
- egl/dri: de-duplicate dri2_load_driver*
- meson: egl: correctly manage loader/xmlconfig
- loader: use loader_open_device() to handle O_CLOEXEC
- egl/android: bump the number of drmDevices to 64
- docs: mention "Allow commits from members who can merge…"
- egl/sl: split out swrast probe into separate function
- egl/sl: use drmDevice API to enumerate available devices
- egl/sl: use kms_swrast with vgem instead of a random GPU
- docs: add release notes for 18.3.5
- docs: add sha256 checksums for 18.3.5
- docs: update calendar, add news item and link release notes for 18.3.5
- docs: add release notes for 18.3.6
- docs: add sha256 checksums for 18.3.6
- docs: update calendar, add news item and link release notes for 18.3.6
- turnip: drop dead close(master_fd)
- vulkan/wsi: check if the display_fd given is master
- vulkan/wsi: don't use DUMB_CLOSE for normal GEM handles
- llvmpipe: add lp_fence_timedwait() helper
- llvmpipe: correctly handle waiting in llvmpipe_fence_finish
- egl/dri: flesh out and use dri2_create_drawable()
- mapi: add static_date offset to MaxShaderCompilerThreadsKHR
- mapi: correctly handle the full offset table

Emmanuel Gil Peyrot (1):

- docs: make bugs.html easier to find

Eric Anholt (121):

- v3d: Always enable the NEON utile load/store code.
- v3d: Fix a release build set-but-unused compiler warning.

- mesa: Skip partial InvalidateFramebuffer of packed depth/stencil.
- v3d: Fix image_load_store clamping of signed integer stores.
- nir: Move V3D's "the shader was TGSI, ignore FS output types" flag to NIR.
- v3d: Fix precompile of FRAG_RESULT_DATA1 and higher outputs.
- v3d: Store the actual mask of color buffers present in the key.
- v3d: Fix dumping of shaders with alpha test.
- v3d: Fix pack/unpack of VFPACK operand unpacks.
- v3d: Fix input packing of .l for rounding/fdx/fdy.
- v3d: Fix copy-propagation of input unpacks.
- v3d: Whitespace consistency fix.
- nir: Move panfrost's isign lowering to nir_opt_algebraic.
- v3d: Use the NIR lowering for isign instead of rolling our own.
- intel: Use the NIR lowering for isign.
- freedreno: Use the NIR lowering for isign.
- v3d: Clear the GMP on initialization of the simulator.
- v3d: Sync indirect draws on the last rendering.
- v3d: Use the early_fragment_tests flag for the shader's disable-EZ field.
- v3d: Fix incorrect flagging of ldtmu as writing r4 on v3d 4.x.
- v3d: Drop a perf note about merging unpack_half_*, which has been implemented.
- v3d: Drop our hand-lowered nir_op_ffract.
- v3d: Add a helper function for getting a nop register.
- v3d: Refactor bcsel and if condition handling.
- v3d: Do bool-to-cond for discard_if as well.
- v3d: Kill off vir_PF(), which is hard to use right.
- v3d: Fix f2b32 behavior.
- v3d: Fix the check for "is the last thrsw inside control flow"
- v3d: Add a function to describe what the c->execute.file check means.
- v3d: Stop tracking num_inputs for VPM loads.
- v3d: Delay emitting ldvpm on V3D 4.x until it's actually used.
- v3d: Emit a simpler negate for the iabs implementation.
- v3d: Move i2b and f2b support into emit_comparison.
- kmsro: Add the rest of the current set of tinydrm drivers.
- nir: Just return when asked to rewrite uses of an SSA def to itself.
- v3d: Fix vir_is_raw_mov() for input unpacks.
- v3d: Dump the VIR after register spilling if we were forced to.
- v3d: Rematerialize MOVs of uniforms instead of spilling them.

- v3d: Fix build of NEON code with Mesa's cflags not targeting NEON.
- v3d: Restrict live intervals to the blocks reachable from any def.
- v3d: Stop treating exec masking specially.
- nir: Improve printing of load_input/store_output variable names.
- v3d: Translate f2i(fround_even) as FTOIN.
- v3d: Move the stores for fixed function VS output reads into NIR.
- v3d: Fix temporary leaks of temp_registers and when spilling.
- v3d: Do uniform rematerialization spilling before dropping threadcount
- v3d: Switch implicit uniforms over to being any qinst->uniform != ~0.
- v3d: Add support for vir-to-qpu of ldunif instructions to a temp.
- v3d: Drop the old class bits splitting up the accumulators.
- v3d: Add support for register-allocating a ldunif to a QFILE_TEMP.
- v3d: Use ldunif instructions for uniforms.
- v3d: Eliminate the TLB and TLBU files.
- v3d: Drop the V3D 3.x vpm read dead code elimination.
- v3d: Include a count of register pressure in the RA failure dumps.
- st/dri: Set the PIPE_BIND_SHARED flag on create_image_with_modifiers.
- util: Add a DAG datastructure.
- vc4: Switch over to using the DAG datastructure for QIR scheduling.
- v3d: Reuse list_for_each_entry_rev().
- vc4: Reuse list_for_each_entry_rev().
- v3d: Use the DAG datastructure for QPU instruction scheduling.
- vc4: Switch the post-RA scheduler over to the DAG datastructure.
- v3d: Disable PIPE_CAP_BLIT_BASED_TEXTURE_TRANSFER.
- v3d: Fix leak of the mem_ctx after the DAG refactor.
- v3d: Fix leak of the renderonly struct on screen destruction.
- mesa/st: Make sure that prog_to_nir NIR gets freed.
- mesa/st: Fix leaks of TGSI tokens in VP variants.
- v3d: Always lay out shared tiled buffers with UIF_TOP set.
- v3d: Allow the UIF modifier with renderonly.
- v3d: Expose the dma-buf modifiers query.
- v3d: Rename v3d_tmu_config_data to v3d_unit_data.
- v3d: Move constant offsets to UBO addresses into the main uniform stream.
- v3d: Upload all of UBO[0] if any indirect load occurs.
- v3d: Remove some dead members of struct v3d_compile.
- egl: Add a 565 pbuffer-only EGL config under X11.

- dri3: Return the current swap interval from glXGetSwapIntervalMESA().
- v3d: Add support for handling OOM signals from the simulator.
- v3d: Bump the maximum texture size to 4k for V3D 4.x.
- v3d: Don't try to use the TFU blit path if a scissor is enabled.
- v3d: Add some more new packets for V3D 4.x.
- st: Lower uniforms in st in the !PIPE_CAP_PACKED_UNIFORMS case as well.
- vc4: Don't forget to set the range when scalarizing our uniforms.
- vc4: Split UBO0 and UBO1 address uniform handling.
- vc4: Upload CS/VS UBO uniforms together.
- v3d: Add an optimization pass for redundant flags updates.
- nir: Drop comments about the constant_index slots for load/stores.
- nir: Drop remaining references to const_index in favor of the call to use.
- nir: Add a comment about how intrinsic definitions work.
- v3d: Add and use a define for the number of channels in a QPU invocation.
- v3d: Drop a note for the future about PIPE_CAP_PACKED_UNIFORMS.
- v3d: Include the number of max temps used in the shader-db output.
- v3d: Replace the old shader-db env var output with the ARB_debug_output.
- v3d: Add Compute Shader compilation support.
- v3d: Add missing base offset to CS shared memory accesses.
- v3d: Add missing dumping for the spill offset/size uniforms.
- v3d: Detect the correct number of QPUs and use it to fix the spill size.
- v3d: Use the new lower_to_scratch implementation for indirects on temps.
- v3d: Only look up the 3rd texture gather offset for non-arrays.
- v3d: Always set up the qregs for CSD payload.
- v3d: Fix an invalid reuse of flags generation from before a thrsw.
- v3d: Fix atomic cmpxchg in shaders on hardware.
- nir: Fix deref offset calculation for structs.
- nir: Use the nir_builder _imm helpers in setting up deref offsets.
- gallium: Remove the pool pipebuffer manager.
- gallium: Remove the ondemand pipebuffer manager.
- gallium: Remove the "alt" pipebuffer manager interface.
- gallium: Remove the malloc pipebuffer manager.
- st/mesa: Don't set atomic counter size != 0 if MAX_SHADER_BUFFERS == 0.
- v3d: Disable SSBOs and atomic counters on vertex shaders.
- v3d: Fill in the ignored segment size fields to appease new simulator.
- v3d: Apply the GFXH-930 workaround to the case where the VS loads attrs.

- v3d: Assert that we do request the normal texturing return data.
- v3d: Use _mesa_hash_table_remove_key() where appropriate.
- vc4: Use _mesa_hash_table_remove_key() where appropriate.
- v3d: Add a note about i/o indirection for future performance work.
- v3d: Don't try to update the shadow texture for separate stencil.
- Revert "v3d: Disable PIPE_CAP_BLIT_BASED_TEXTURE_TRANSFER."
- v3d: Re-add support for memory_barrier_shared.
- v3d: Fix detection of the last ldtmu before a new TMU op.
- v3d: Fix detection of TMU write sequences in register spilling.
- kmsro: Add support for V3D.
- vc4: Fall back to renderonly if the vc4 driver doesn't have v3d.

Eric Engestrom (142):

- wsi/display: add comment
- egl: use coherent variable names
- gitlab-ci: add ubuntu container
- gitlab-ci: add a meson vulkan build
- gitlab-ci: add a make vulkan build
- gitlab-ci: add a scons no-llvm build
- gitlab-ci: add scons llvm 3.5 build
- gitlab-ci: add scons SWR build
- gitlab-ci: add meson loader/classic DRI build
- gitlab-ci: add meson gallium SWR build
- gitlab-ci: add meson gallium RadeonSI build
- gitlab-ci: add meson gallium "other drivers" build
- gitlab-ci: add meson gallium ST Clover (LLVM 5.0) build
- gitlab-ci: add meson gallium ST Clover (LLVM 6.0) build
- gitlab-ci: add meson gallium ST Clover (LLVM 7.0) build
- gitlab-ci: add meson gallium ST "Other" build
- gitlab-ci: add make loaders/classic DRI build
- gitlab-ci: add make Gallium Drivers SWR build
- gitlab-ci: add make Gallium Drivers RadeonSI build
- gitlab-ci: add make Gallium Drivers "Other" build
- gitlab-ci: add make Gallium ST Clover LLVM-3.9 build
- gitlab-ci: add make Gallium ST Clover LLVM-4.0 build
- gitlab-ci: add make Gallium ST Clover LLVM-5.0 build
- gitlab-ci: add make Gallium ST Clover LLVM-6.0 build

- gitlab-ci: add make Gallium ST Clover LLVM-7 build
- gitlab-ci: add make Gallium ST Other build
- travis: remove unused linux code path
- travis: remove unused scons code path
- gitlab-ci: add meson glvnd build
- xvmc: fix string comparison
- xvmc: fix string comparison
- meson: add script to print the options before configuring a builddir
- driconf: drop unused macro
- travis: fix osx make build
- gitlab-ci: workaround docker bug for users with uppercase characters
- wsi: query the ICD's max dimensions instead of hard-coding them
- gitlab-ci: limit ninja to 4 threads max
- drm-uapi/README: remove explicit list of driver names
- drm-uapi: use local files, not system libdrm
- gbm: drop duplicate #defines
- st/dri: drop duplicate #define
- etnaviv: drop duplicate #define
- anv/tests: compile to something sensible in release builds
- util/tests: compile to something sensible in release builds
- gitlab-ci: use ccache to speed up builds
- tegra/meson: add missing dep_libdrm
- tegra/autotools: add missing libdrm cflags
- gitlab-ci: limit the automatic CI to master and MRs
- gitlab-ci: automatically run the CI on pushes to 'ci/*' branches
- anv: sort extensions alphabetically
- anv: sort vendors extensions after KHR and EXT
- anv: make sure the extensions stay sorted
- anv: drop unused imports
- anv: use anv_shader_bin_write_to_blob()'s return value
- gitlab-ci: always run the containers build
- dri_interface: add missing #include
- driinfo: add DTD to allow the xml to be validated
- meson/swr: replace hard-coded path with current_build_dir()
- egl/android: replace magic 0=CbCr,1=CrCb with simple enum
- vulkan: use VkBase{In,Out}Structure instead of a custom struct

- driconf: add DTD to allow the drirc xml (00-mesa-defaults.conf) to be validated
- gitlab-ci: install xmllint to validate 00-mesa-defaults.conf
- anv: simplify chained comparison
- anv: drop unused parameter
- anv: remove spaces around kwargs assignment
- anv: fix typo
- Revert "swr/rast: Archrast codegen updates"
- meson: avoid going back up the tree with include_directories()
- anv: use the platform defines in vk.xml instead of hard-coding them
- radv: use the platform defines in vk.xml instead of hard-coding them
- util: #define PATH_MAX when undefined (eg. Hurd)
- vulkan: import missing file from Khronos
- egl: fix libdrm-less builds
- vulkan: import vk_layer.h from Khronos
- gitlab-ci: drop job prefixes
- meson: fix with_dri2 definition for GNU Hurd
- meson: remove unused include_directories(vulkan)
- vulkan/util: use the platform defines in vk.xml instead of hard-coding them
- vulkan/overlay: fix missing var rename in previous commit
- meson: don't build libGLES*.so with GLVND
- autotools: don't build libGLES*.so with GLVND
- travis: fix meson build by letting 'auto' do its job
- travis: drop unused vars
- travis: clean up
- gitlab-ci: only build the default (=latest) and oldest llvm versions
- gitlab-ci: autotools needs to be told which llvm version to use
- r600: cast pointer to expected type
- build: make passing an incorrect pointer type a hard error
- gitlab-ci: fix llvm version (7 doesn't have a ".0")
- hgl/meson: drop unused include directory
- glx/meson: use full include path for dri_interface.h
- android: fix missing backspace for line continuation
- panfrost: fix tgsi_to_nir() call
- panfrost: move #include to fix compilation
- gitlab-ci: add panfrost to the gallium drivers build
- wsi: deduplicate get_current_time() functions between display and x11

- wsi/display: s/#if/#ifdef/ to fix -Wundef
- wsi/wayland: fix pointer casting warning on 32bit
- wsi/x11: use WSI_FROM_HANDLE() instead of pointer casts
- turnip: use the platform defines in vk.xml instead of hard-coding them
- travis: fix osx meson build
- nir: const ‘nir_call_instr::callee‘
- gitlab-ci: add clang build
- gitlab-ci: drop most autotools builds
- util/disk_cache: close fd in the fallback path
- egl: hide entrypoints that shouldn’t be exported when using glvnd
- meson: strip rpath from megadrivers
- gallium/hud: fix memory leaks
- gallium/hud: prevent buffer overflow
- gallium/hud: fix rounding error in nic bps computation
- simplify LLVM version string printing
- util/process: document memory leak
- vk/util: remove unneeded array index
- bin: drop unused import from install_megadrivers.py
- meson: remove meson-created megadrivers symlinks
- gitlab-ci: build gallium extra hud
- gitlab-ci: add lima to the build
- delete autotools .gitignore files
- delete autotools input files
- docs: remove unsupported GL function name mangling
- docs: drop autotools python information
- docs: replace autotools intructions with meson equivalent
- docs: use past tense when talking about autotools
- docs: haiku can be built using meson
- egl: fixup autotools-specific wording
- util: add os_read_file() helper
- anv: add support for VK_EXT_memory_budget
- radv: update to use the new features struct names
- turnip: update to use the new features struct names
- gitlab-ci: build vulkan drivers in clang build
- util: move #include out of #if linux
- wsi/wayland: document lack of vkAcquireNextImageKHR timeout support

- egl: hard-code destroy function instead of passing it around as a pointer
- gitlab-ci: add scons windows build using mingw
- gitlab-ci: merge several meson jobs
- gitlab-ci: meson-gallium-radeonsi was a subset of meson-gallium-clover-llvm
- gitlab-ci: simplify meson job names
- gitlab-ci: merge meson-glvnd into meson-swr
- travis: fix syntax, and drop unused stuff
- util/os_file: always use the 'grow' mechanism
- meson: expose glapi through osmesa
- util/os_file: actually return the error read() gave us

Erico Nunes (5):

- lima/ppir: support ppir_op_ceil
- nir/algebraic: add lowering for fsign
- lima: enable nir fsign lowering in ppir
- lima/gpir: add limit of max 512 instructions
- lima/ppir: support nir_op_ftrunc

Erik Faye-Lund (79):

- mesa: expose NV_conditional_render on GLES
- st/mesa: remove unused header-file
- swr/codegen: fix autotools build
- virgl: remove unused variables
- virgl: remove unused variable
- virgl: remove unused variable
- virgl: remove unused variable
- virgl: do not allow compressed formats for buffers
- virgl: stricter usage of compressed 3d textures
- virgl: also destroy all read-transfers
- virgl: use debug_printf instead of fprintf
- virgl: unsigned int -> unsigned
- virgl: only warn about unchecked flags
- virgl: do not warn about display-target binding
- virgl: use debug_printf instead of fprintf
- virgl: remove pointless transfer-counter
- virgl: tmp_resource -> templ
- virgl: track full virgl_resource instead of just virgl_hw_res
- virgl: simplify virgl_texture_transfer_unmap logic

- virgl: make unmap queuing a bit more straight-forward
- virgl: check for readback on correct resource
- virgl: wait for the right resource
- virgl: return error if allocating resolve_tmp fails
- virgl: rewrite core of virgl_texture_transfer_map
- virgl: use pipe_box for blit dst-rect
- virgl: support write-back with staged transfers
- virgl: make sure bind is set for non-buffers
- gallium/util: support translating between uint and sint formats
- virgl: get readback-formats from host
- virgl: only blit if resource is read
- virgl: do color-conversion during when mapping transfer
- virgl: document potentially failing blit
- mesa/st: remove impossible error-check
- gallium/u_vbuf: support NULL-resources
- i915: support NULL-resources
- nouveau: support NULL-resources
- swr: support NULL-resources
- mesa/st: accept NULL and empty buffer objects
- mesa/st: remove always-false state
- softpipe: setup pixel_offset for all primitive types
- docs: normaize css-indent style
- docs: remove non-existent css attribute
- docs: remove long commented out css
- docs: add missing semicolon
- docs: avoid repeating the font
- docs: avoid repeating the color
- docs: remove spurious newline
- docs: use multiple background-images for header
- docs: simplify css-centering
- docs: do not hard-code header-height
- docs: properly escape '>'
- docs: properly escape ampersand
- docs: remove stray paragraph-close
- docs: use h2 instead of b-tag for headings
- docs: use dl/dd instead of blockquote for freedesktop link

- docs: open list-item before closing it
- docs: close paragraphs before lists
- docs: close lists
- docs: remove stray paragraph-close
- docs: close paragraphs before preformatted text
- docs: start paragraph before closing it
- docs: drop paragraph around preformatted text
- docs: fix incorrectly closed paragraph
- docs: don't pointlessly close and re-start definition lists
- docs: remove stray list-start
- docs: fixup bad paragraphing
- docs: add missing lists
- docs: fix closing of paragraphs
- docs: fixup list-item tags
- docs: fix closing of list-items
- docs: replace empty list with a none-paragraph
- docs: turn faq-index into an ordered list
- docs: drop centered heading for faq
- docs: reorder heading and notice
- meson: lift driver-collection out into parent build-file
- meson: give dri- and gallium-drivers separate vars
- meson: add build-summary
- docs: fixup mistake in contents
- draw: flush when setting stream-out targets

Ernestas Kulik (2):

- vc4: Fix leak in HW queries error path
- v3d: Fix leak in resource setup error path

Francisco Jerez (6):

- intel/dump_gpu: Disambiguate between BOs from different GEM handle spaces.
- intel/fs: Exclude control sources from execution type and region alignment calculations.
- intel/fs: Lower integer multiply correctly when destination stride equals 4.
- intel/fs: Cap dst-aligned region stride to maximum representable hstride value.
- intel/fs: Implement extended strides greater than 4 for IR source regions.
- intel/fs: Rely on undocumented unrestricted regioning for 32x16-bit integer multiply.

Fritz Koenig (4):

- freedreno: pass count to query_dmabuf_modifiers

- freedreno/a6xx: UBWC support
- freedreno: UBWC allocator
- freedreno/a6xx: Enable UBWC modifier

Gert Wollny (35):

- mesa/core: Enable EXT_texture_sRGB_R8 also for desktop GL
- radeonsi: release tokens after creating the shader program
- mesa: release references to image textures when a context is destroyed
- virgl: Enable mixed color FBO attachemnets only when the host supports it
- mesa/core: Enable EXT_depth_clamp for GLES >= 2.0
- nir: Add posibility to not lower to source mod 'abs' for ops with three sources
- mesa: Expose EXT_texture_query_lod and add support for its use shaders
- softpipe: Enable PIPE_CAP_MIXED_COLORBUFFER_FORMATS It seems softpipe actually supports this. This change enables the following piglits as passing without regressions in the gpu test set:
- virgl: Add a caps feature check version
- softpipe: Implement ATOMFADD and enable cap TGSI_ATOMFADD
- virgl: define MAX_VERTEX_STREAMS based on availability of TF3
- softpipe: Use mag texture filter also for clamped lod == 0
- softpipe: Don't use mag filter for gather op
- softpipe: raise number of bits used for X coordinate texture lookup
- softpipe: Add an extra code path for the buffer texel lookup
- softpipe: Enable PIPE_CAP_TEXTURE_BUFFER_OFFSET_ALIGNMENT
- Gallium: Add new CAP that indicated whether IO array definitions can be shriked
- virgl: Enable passing arrays as input to fragment shaders
- doc/features: Add a few extensions to the feature matrix
- softpipe: Factor gradient evaluation out of the lambda evaluation
- softpipe: Prepare handling explicit gradients
- softpipe: Pipe gather_comp through from st_tgsi_get_samples
- softpipe: Move selection of shadow values up and clean parameter list
- softpipe: tie in new code path for lod evaluation
- softpipe: keep input lod for explicite derivatives
- softpipe: evaluate cube the faces on a per sample bases
- softpipe: Factor out evaluation of the source indices
- softpipe: Add an per-input array for interpolator correctors to machine
- softpipe: Add (fake) support for TGSI_OPCODE_INTERP_SAMPLE
- softpipe: Add support for TGSI_OPCODE_INTERP_OFFSET
- softpipe: Add support for TGSI_OPCODE_INTERP_CENTROID

- softpipe: Increase the GLSL feature level
- doc: Update feature matrix
- softpipe/buffer: load only as many components as the the buffer resource type provides
- Revert "softpipe/buffer: load only as many components as the the buffer resource type provides"

Greg V (3):

- util: emulate futex on FreeBSD using umtx
- gallium/hud: add CPU usage support for FreeBSD
- gallium: enable dmabuf on BSD as well

Grigori Goronzy (1):

- glx: add support for GLX_ARB_create_context_no_error (v3)

Guido Günther (4):

- docs: Fix 19.0.x version numbers
- gallium: ddebug: Add missing fence related wrappers
- gallium/u_dump: util_dump_sampler_view: Dump u.tex.first_level
- gallium: trace: Add missing fence related wrappers

Gurchetan Singh (44):

- mesa/main: Expose EXT_texture_compression_s3tc_srgb
- i965: Set flag for EXT_texture_compression_s3tc_srgb
- st/mesa: expose EXT_texture_compression_s3tc_srgb
- docs: add GL_EXT_texture_compression_s3tc_srgb to release notes
- virgl: add ability to do finer grain dirty tracking
- virgl: use virgl_resource_dirty helper
- virgl: don't mark unclean after a flush
- virgl: track level cleanliness rather than resource cleanliness
- virgl: make alignment smaller when uploading index user buffers
- virgl: unmap uploader at flush time
- virgl: when creating / freeing transfers, pass slab pool directly
- virgl: add protocol for resource transfers
- virgl: use virgl_transfer in inline write
- virgl: limit command length to 16 bits
- virgl: keep track of number of computations
- virgl: pass virgl transfer to virgl_res_needs_flush_wait
- virgl: add extra checks in virgl_res_needs_flush_wait
- virgl: make winsys modifications for encoded transfers
- virgl: add encoder functions for new protocol
- virgl: introduce transfer queue

- virgl: use transfer queue
- virgl: use virgl_transfer_inline_write even less
- virgl/vtest: deprecate protocol version 1
- egl/sl: also allow virtgpu to fallback to kms_swrast
- virgl: use uint16_t mask instead of separate booleans
- configure.ac / meson: depend on libnativewindow when appropriate
- anv: move anv_GetMemoryAndroidHardwareBufferANDROID up a bit
- anv: fix build on Nougat
- egl/android: move droid_image_loader_extension down a bit
- egl/android: move droid_open_device_drm_gralloc down a bit
- egl/android: droid_open_device_drm_gralloc –> droid_open_device
- egl/android: refactor droid_load_driver a bit
- egl/android: plumb swrast option
- egl/android: use swrast option in droid_load_driver
- egl/android: use software rendering when appropriate
- egl/android: chose node type based on swrast and preprocessor flags
- virgl: wait after a flush
- virgl/vtest: execute a transfer_get when flushing the front buffer
- virgl/vtest: add utilities for receiving fds
- virgl/vtest: plumb support for shared memory
- virgl/vtest: receive and handle shared memory fd
- virgl/vtest: modify sending and receiving data for shared memory
- virgl/vtest: wait after issuing a transfer get
- virgl/vtest: bump up protocol version + support encoded transfers

Guttula, Suresh (1):

- st/va:Add support for indirect manner by returning VA_STATUS_ERROR_OPERATION_FAILED

Hal Gentz (1):

- glx: Fix synthetic error generation in __glXSendError

Heinrich (1):

- gbm: Improve documentation of BO import

Iago Toral Quiroga (39):

- compiler/nir: add an is_conversion field to nir_op_info
- compiler/nir: add lowering option for 16-bit fmod
- compiler/nir: add lowering for 16-bit flrp
- compiler/nir: add lowering for 16-bit ldexp
- intel/compiler: add a NIR pass to lower conversions

- intel/compiler: split float to 64-bit opcodes from int to 64-bit
- intel/compiler: handle b2i/b2f with other integer conversion opcodes
- intel/compiler: assert restrictions on conversions to half-float
- intel/compiler: lower some 16-bit float operations to 32-bit
- intel/compiler: handle extended math restrictions for half-float
- intel/compiler: implement 16-bit fsign
- intel/compiler: drop unnecessary temporary from 32-bit fsign implementation
- intel/compiler: add instruction setters for Src1Type and Src2Type.
- intel/compiler: add new half-float register type for 3-src instructions
- intel/compiler: don't compact 3-src instructions with Src1Type or Src2Type bits
- intel/compiler: allow half-float on 3-source instructions since gen8
- intel/compiler: set correct precision fields for 3-source float instructions
- intel/compiler: fix ddx and ddy for 16-bit float
- intel/compiler: fix ddy for half-float in Broadwell
- intel/compiler: workaround for SIMD8 half-float MAD in gen8
- intel/compiler: split is_partial_write() into two variants
- intel/compiler: activate 16-bit bit-size lowerings also for 8-bit
- intel/compiler: rework conversion opcodes
- intel/compiler: ask for an integer type if requesting an 8-bit type
- intel/eu: force stride of 2 on NULL register for Byte instructions
- intel/compiler: generalize the combine constants pass
- intel/compiler: implement is_zero, is_one, is_negative_one for 8-bit/16-bit
- intel/compiler: add a brw_reg_type_is_integer helper
- intel/compiler: fix cmod propagation for non 32-bit types
- intel/compiler: remove inexact algebraic optimizations from the backend
- intel/compiler: skip MAD algebraic optimization for half-float or mixed mode
- intel/compiler: implement SIMD16 restrictions for mixed-float instructions
- intel/compiler: also set F execution type for mixed float mode in BDW
- intel/compiler: validate region restrictions for half-float conversions
- intel/compiler: validate conversions between 64-bit and 8-bit types
- intel/compiler: validate region restrictions for mixed float mode
- compiler/spirv: move the check for Int8 capability
- anv/pipeline: support Float16 and Int8 SPIR-V capabilities in gen8+
- anv/device: expose VK_KHR_shader_float16_int8 in gen8+

Ian Romanick (55):

- nir: Silence zillions of unused parameter warnings in release builds

- intel/compiler: Silence warning about value that may be used uninitialized
- nir: Document some fields of nir_loop_terminator
- nir: Refactor code that checks phi nodes in opt_peel_loop_initial_if
- nir: Select phi nodes using prev_block instead of continue_block
- nir: Split ALU instructions in loops that read phis
- nir: Convert a bcsel with only phi node sources to a phi node
- spirv: Add missing break
- nir/algebraic: Convert some f2u to f2i
- nir/algebraic: Simplify comparison with sequential integers starting with 0
- intel/vec4: Emit constants for some ALU sources as immediate values
- nir/algebraic: Replace i2b used by bcsel or if-statement with comparison
- intel/fs: Relax type matching rules in cmod propagation from MOV instructions
- intel/fs: Handle OR source modifiers in algebraic optimization
- intel/fs: Refactor ALU source and destination handling to a separate function
- intel/fs: Emit logical-not of operands on Gen8+
- intel/fs: Use De Morgan's laws to avoid logical-not of a logic result on Gen8+
- intel/fs: Emit better code for b2f(inot(a)) and b2i(inot(a))
- nir/algebraic: Replace a bcsel of a b2f sources with a b2f(!(a || b))
- intel/fs: Generate if instructions with inverted conditions
- nir/algebraic: Replace a-fract(a) with floor(a)
- intel/fs: Don't assert on b2f with a saturate modifier
- nir/algebraic: Optimize away an fsat of a b2f
- intel/compiler: Silence many unused parameter warnings in brw_eu.h
- intel/compiler: Silence unused parameter warning in brw_interpolation_map.c
- intel/fs: nir_op_extract_i8 extracts a byte, not a word
- intel/fs: Fix extract_u8 of an odd byte from a 64-bit integer
- nir/algebraic: Fix up extract_[iu]8 after loop unrolling
- nir/algebraic: Remove redundant extract_[iu]8 patterns
- nir/algebraic: Add missing 64-bit extract_[iu]8 patterns
- nir/algebraic: Add missing 16-bit extract_[iu]8 patterns
- nir/algebraic: Fix up extract_[iu]8 after loop unrolling
- nir/algebraic: Remove redundant extract_[iu]8 patterns
- nir/algebraic: Add missing 64-bit extract_[iu]8 patterns
- nir/algebraic: Add missing 16-bit extract_[iu]8 patterns
- nir: Add nir_const_value_negative_equal
- nir: Add nir_alu_srcs_negative_equal

- nir: Add partial redundancy elimination for compares
- intel/compiler: Use partial redundancy elimination for compares
- intel/fs: Eliminate dead code first
- intel/fs: Refactor code generation for nir_op_fsign to its own function
- intel/fs: Add a scale factor to emit_fsign
- intel/fs: Generate better code for fsign multiplied by a value
- nir/algebraic: Recognize open-coded copysign(1.0, a)
- nir/algebraic: Replace a pattern where iand with a Boolean is used as a bcsel
- nir/algebraic: Fix some 1-bit Boolean weirdness
- nir/algebraic: Strength reduce some compares of x and -x
- intel/fs: Add support for float16 to the fsign optimizations
- glsl: Silence may unused parameter warnings in glsl/ir.h
- intel/compiler: Don't have sepearate, per-Gen nir_options
- intel/compiler: Lower ffma on Gen4 and Gen5
- intel/fs: Fix D to W conversion in opt_combine_constants
- mesa: Add missing display list support for GL_FOG_COORDINATE_SOURCE
- nir: Saturating integer arithmetic is not associative
- Revert "nir: add late opt to turn inot/b2f combos back to bcsel"

Icenowy Zheng (5):

- lima: add dummy set_sample_mask function
- lima: make lima_context_framebuffer subtype of pipe_framebuffer_state
- lima: implement blit with util_blitter
- lima: lower bool to float when building shaders
- lima: add Android build

Ilia Mirkin (14):

- nv50,nvc0: add explicit settings for recent caps
- nvc0: add support for handling indirect draws with attrib conversion
- nvc0/ir: always use CG mode for loads from atomic-only buffers
- nvc0/ir: fix second tex argument after levelZero optimization
- nvc0: fix 3d images on kepler
- nv50,nvc0: use condition for occlusion queries when already complete
- nvc0: stick zero values for the compute invocation counts
- nvc0: we have 16k-sized framebuffers, fix default scissors
- swr: set PIPE_CAP_MAX_VARYINGS correctly
- mesa: add explicit enable for EXT_float_blend, and error condition
- st/mesa: enable GL_EXT_float_blend when possible

- i965: always enable EXT_float_blend
- nv50: disable compute
- glsl: fix recording of variables for XFB in TCS shaders

Illia Iorin (1):

- mesa/main: Fix multisample texture initialize

James Zhu (12):

- gallium/auxiliary/vl: Move dirty define to header file
- gallium/auxiliary/vl: Split vl_compositor graphic shaders from vl_compositor API
- gallium/auxiliary/vl: Rename csc_matrix and increase its size.
- gallium/auxiliary/vl: Add compute shader to support video compositor render
- gallium/auxiliary/vl: Add video compositor compute shader render
- gallium/auxiliary/vl: Fix transparent issue on compute shader with rgba
- gallium/auxiliary/vl: Increase shader_params size
- gallium/auxiliary/vl: Change grid setting
- gallium/auxiliary/vl: Change weave compute shader implementation
- gallium/auxiliary/vl: Fixed blur issue with weave compute shader
- gallium/auxiliary/vl: Fixed blank issue with compute shader
- gallium/auxiliary/vl: Add barrier/unbind after compute shader launch.

Jan Vesely (2):

- Partially revert "gallium: fix autotools build of pipe_msm.la"
- gallium/aux: Report error if loading of a pipe driver fails.

Jan Zielinski (1):

- swr/rast: fix 32-bit compilation on Linux

Jason Ekstrand (212):

- spirv: Replace vtn_constant_value with vtn_constant_uint
- spirv: Rework handling of spec constant workgroup size built-ins
- spirv: Handle constants and types before execution modes
- spirv: Handle OpExecutionModeId
- spirv: Support LocalSizeId and LocalSizeHintId execution modes
- intel/nir: Add global support to lower_mem_access_bit_sizes
- intel/fs/cse: Split create_copy_instr into three cases
- intel/fs: Properly handle 64-bit types in LOAD_PAYLOAD
- intel/fs: Do the grf127 hack on SIMD8 instructions in SIMD16 mode
- intel/fs: Implement load/store_global with A64 untyped messages
- intel/fs: Use SENDS for A64 writes on gen9+
- intel/fs: Implement nir_intrinsic_global_atomic_*

- anv: Implement VK_EXT_buffer_device_address
- relnotes: Add VK_EXT_buffer_device_address
- nir/deref: Drop zero ptr_as_array derefs
- README: Drop the badges from the readme
- intel/fs: Use enumerated array assignments in fb read TXF setup
- nir/deref: Rematerialize parents in rematerialize_derefs_in_use_blocks
- nir: Silence a couple of warnings in release builds
- anv/blorp: Delete a pointless assert
- anv: Silence some compiler warnings in release builds
- intel/fs: Silence a compiler warning
- intel/fs: Bail in optimize_extract_to_float if we have modifiers
- nir/dead_cf: Inline cf_node_has_side_effects
- nir/dead_cf: Stop relying on liveness analysis
- compiler/types: Add a contains_64bit helper
- nir/xfb: Properly align 64-bit values
- nir: Rewrite lower_clip_cull_distance_arrays to do a lot less lowering
- nir/xfb: Work in terms of components rather than slots
- nir/xfb: Handle compact arrays in gather_xfb_info
- nir: Fix a compile warning
- nir/lower_clip_cull: Fix an incorrect assert
- iris: Don't lower image formats for write-only images
- iris/compute: Don't increment the grid size offset
- iris/compute: Zero out the last grid size on indirect dispatches
- iris: Configure the L3$ on the compute context
- iris: Don't set constant read lengths at upload time
- iris: Allocate buffer resources separately
- iris: Copy anv's MI_MATH helpers for multiplication and division
- nir/split_vars: Don't compact vectors unnecessarily
- nir/builder: Don't emit no-op swizzles
- intel/eu: Add an EOT parameter to send_indirect_[split]_message
- intel/fs: Add an enum type for logical sampler inst sources
- intel/fs: Re-order logical surface arguments
- intel/fs: Drop the fs_surface_builder
- intel/vec4: Drop dead code for handling typed surface messages
- intel/fs: Get rid of the IMAGE_SIZE opcode
- intel/compiler: Drop unused surface opcodes

- intel/schedule_instructions: Move some comments
- intel/compiler: Re-prefix non-logical surface opcodes with VEC4
- anv: Count surfaces for non-YCbCr images in GetDescriptorSetLayoutSupport
- spirv: OpImageQueryLod requires a sampler
- intel,nir: Lower TXD with min_lod when the sampler index is not < 16
- anv: Use an actual binding for gl_NumWorkgroups
- anv/pipeline: Drop anv_fill_binding_table
- anv/descriptor_set: Refactor alloc/free of descriptor sets
- anv: Rework arguments to anv_descriptor_set_write_*
- anv: Stop allocating buffer views for dynamic buffers
- anv: Count image param entries rather than images
- anv: Clean up descriptor set layouts
- anv: drop add_var_binding from anv_nir_apply_pipeline_layout.c
- anv: Refactor descriptor pushing a bit
- anv: Take references to push descriptor set layouts
- anv: Add a concept of a descriptor buffer
- spirv: Pull offset/stride from the pointer for OpArrayLength
- spirv: Use the generic dereference function for OpArrayLength
- spirv: Use the same types for resource indices as pointers
- anv: Implement VK_EXT_inline_uniform_block
- nir: Expose double and int64 op_to_options_mask helpers
- nir: Teach loop unrolling about 64-bit instruction lowering
- i965: Compile the fp64 program based on nir options
- intel/debug: Add a debug flag to force software fp64
- intel/nir: Drop an unneeded lower_constant_initializers call
- glsl/nir: Add a shared helper for building float64 shaders
- glsl/nir: Inline functions in float64_funcs_to_nir
- nir/inline_functions: Break inlining into a builder helper
- nir/deref: Expose nir_opt_deref_impl
- nir/lower_doubles: Inline functions directly in lower_doubles
- intel/nir: Move 64-bit lowering later
- st/nir: Move 64-bit lowering later
- nir/builder: Emit better code for iadd/imul_imm
- nir/builder: Cast array indices in build_deref_follower
- nir/builder: Add a build_deref_array_imm helper
- intel/nir: Move lower_mem_access_bit_sizes to postprocess_nir

- anv/pipeline: Move lower_explicit_io much later
- nir: Add a pass for lowering IO back to vector when possible
- intel/nir: Vectorize all IO
- anv: Ignore VkRenderPassInputAttachementAspectCreateInfo
- nir/loop_unroll: Fix out-of-bounds access handling
- glsl/list: Add a list variant of insert_after
- glsl/lower_vector_derefs: Don't use a temporary for TCS outputs
- anv: Stop using VK_TRUE/FALSE
- anv/pass: Flag the need for a RT flush for resolve attachments
- anv: Only set 3DSTATE_PS::VectorMaskEnable on gen8+
- nir/algebraic: Add a couple optimizations for iabs and ishr
- nir/validate: Only require bare types to match for copy_deref
- nir/validate: Allow 32-bit boolean load/store intrinsics
- compiler/types: Add a new is_interface C wrapper
- compiler/types: Add a C wrapper to get full struct field data
- compiler/types: Add helpers to get explicit types for standard layouts
- nir/deref: Consider COHERENT decorated var derefs as aliasing
- nir: Rename nir_address_format_vk_index_offset to not be vk
- nir/lower_io: Add a new buffer_array_length intrinsic and lowering
- glsl: Don't lower vector derefs for SSBOs, UBOs, and shared
- glsl/nir: Set explicit types on UBO/SSBO variables
- glsl/nir: Handle unlowered SSBO atomic and array_length intrinsics
- glsl/nir: Add a pass to lower UBO and SSBO access
- i965: Stop setting LowerBuferInterfaceBlocks
- st/mesa: Let NIR lower UBO and SSBO access when we have it
- nir/builder: Add a vector extract helper
- nir: Add a new pass to lower array dereferences on vectors
- intel/nir: Lower array-deref-of-vector UBO and SSBO loads
- anv: Implement VK_EXT_host_query_reset
- anv,radv: Implement VK_KHR_surface_capability_protected
- Revert "nir: const 'nir_call_instr::callee'"
- anv: Bump maxComputeWorkgroupInvocations
- nir: Constant values are per-column not per-component
- anv,radv,turnip: Lower TG4 offsets with nir_lower_tex
- spirv: Drop inline tg4 lowering
- nir/lower_io: Add a bounds-checked 64-bit global address format

- nir: Add a lowering pass for non-uniform resource access
- nir: Add texture sources and intrinsics for bindless
- nir: Add access flags to deref and SSBO atomics
- spirv: Handle the NonUniformEXT decoration
- Revert "anv/radv: release memory allocated by glsl types during spirv_to_nir"
- nir: Lock around validation fail shader dumping
- nir/algebraic: Drop some @bool specifiers
- nir/algebraic: Add some logical OR and AND patterns
- vc4: Prefer nir_src_comp_as_uint over nir_src_as_const_value
- nir/search: Search for all combinations of commutative ops
- nir: Get rid of nir_register::is_packed
- nir: Get rid of global registers
- intel/common: Add a MI command builder
- intel/common: Add unit tests for gen_mi_builder
- anv: Use gen_mi_builder for CmdDrawIndirectByteCount
- anv: Use gen_mi_builder for computing resolve predicates
- anv: Use gen_mi_builder for indirect draw parameters
- anv: Use gen_mi_builder for indirect dispatch
- anv: Use gen_mi_builder for conditional rendering
- anv: Use gen_mi_builder for queries
- anv: Move mi_memcpy and mi_memset to gen_mi_builder
- anv/cmd_buffer: Use gen_mi_sub instead of gen_mi_add with a negative
- intel/common: Support bigger right-shifts with mi_builder
- anv/pipeline: Fix MEDIA_VFE_STATE::PerThreadScratchSpace on gen7
- nir: Add a pass for selectively lowering variables to scratch space
- intel/nir: Take a nir_tex_instr and src index in brw_texture_offset
- nir/builder: Add a nir_imm_zero helper
- nir/print: Use nir_src_as_int for array indices
- nir/constant_folding: Get rid of a bit size switch statement
- spirv: Drop some unneeded bit size switch statements
- nir/load_const_to_scalar: Get rid of a bit size switch statement
- nir/validate: Require unused bits of nir_const_value to be zero
- vulkan: Update the XML and headers to 1.1.106
- anv: Update to use the new features struct names
- nir/algebraic: Move the template closer to the render function
- nir/algebraic: Use a cache to avoid re-emitting structs

- intel/mi_builder: Re-order an initializer
- intel/mi_builder: Disable mem_mem tests on IVB
- nir: Drop "struct" from some nir_* declarations
- nir: Rework nir_src_as_alu_instr to not take a pointer
- nir: Add a nir_src_as_intrinsic() helper
- anv: Re-sort the GetPhysicalDeviceFeatures2 switch statement
- anv: Drop some unneeded ANV_FROM_HANDLE for physical devices
- intel/fs: Account for live range lengths in spill costs
- anv: Make all VkDeviceMemory BOs resident permanently
- anv: Put image params in the descriptor set buffer on gen8 and earlier
- anv: Add a #define for the max binding table size
- anv/pipeline: Sort bindings by most used first
- anv/pipeline: Add skeleton support for spilling to bindless
- nir/lower_io: Expose some explicit I/O lowering helpers
- intel/nir: Re-run int64 lowering in postprocess_nir
- anv: Add a has_a64_buffer_access to anv_physical_device
- anv: Lower some SSBO operations in apply_pipeline_layout
- anv: Implement SSBOs bindings with GPU addresses in the descriptor BO
- anv: Implement VK_KHR_shader_atomic_int64
- intel,nir: Lower TXD with a bindless sampler
- intel/fs: Add support for bindless texture ops
- anv: Count the number of planes in each descriptor binding
- anv: Use write_image_view to initialize immutable samplers
- anv: Pass the plane into lower_tex_deref
- anv: Use bindless textures and samplers
- intel/fs: Add support for bindless image load/store/atomic
- anv: Use bindless handles for images
- anv: Put binding flags in descriptor set layouts
- anv: Implement VK_EXT_descriptor_indexing
- nir: Add helpers for getting the type of an address format
- anv/nir: Add a central helper for figuring out SSBO address formats
- anv: Ignore descriptor binding flags if bindingCount == 0
- anv: Rework the descriptor set layout create loop
- anv,radv: Update release notes for newly implemented extensiosn
- nir: Use the NIR_SRC_AS_ macro to define nir_src_as_deref
- anv/descriptor_set: Unlink sets from the pool in set_destroy

- anv/descriptor_set: Destroy sets before pool finalization
- anv/descriptor_set: Only vma_heap_finish if we have a descriptor buffer
- anv/descriptor_set: Properly align descriptor buffer to a page
- anv: Better handle 32-byte alignment of descriptor set buffers
- anv/descriptor_set: Don't fully destroy sets in pool destroy/reset
- nir/algebraic: Optimize integer cast-of-cast
- util/bitset: Return an actual bool from test macros
- anv: Stop including POS in FS input limits
- anv,i965: Stop warning about incomplete gen11 support
- nir: Add a SSA type gathering pass
- intel/fs/ra: Only add dest interference to sources that exist
- intel/fs/ra: Stop adding RA interference to too many SENDS nodes
- anv: Emulate texture swizzle in the shader when needed
- anv: Stop forcing bindless for images
- anv: Only consider minSampleShading when sampleShadingEnable is set
- iris: Don't assume UBO indices are constant
- intel/fs,vec4: Use g0 as the header for MFENCE
- intel/fs: Do a stalling MFENCE in endInvocationInterlock()
- nir/dead_cf: Call instructions aren't dead
- nir/propagate_invariant: Don't add NULL vars to the hash table

Jian-Hong Pan (1):

- intel: Fix the description of Coffeelake pci-id 0x3E98

Jiang, Sonny (1):

- va: use a compute shader for the blit

John Stultz (3):

- mesa: android: freedreno: Fix build failure due to path change
- mesa: Makefile.sources: Add ir3_nir_lower_load_barycentric_at_sample/offset to Makefile.sources
- mesa: Makefile.sources: Add nir_lower_fb_read.c to Makefile.sources list

Jon Turney (1):

- meson: Force '.so' extension for DRI drivers

Jonathan Marek (22):

- nir: add missing vec opcodes in lower_bool_to_float
- freedreno: a2xx: fix fast clear
- freedreno: a2xx: don't write 4th vertex in mem2gmem
- freedreno: a2xx: add use_hw_binning function
- freedreno: a2xx: fix fast clear for some gmem configurations

- freedreno: a2xx: fix mipmapping for NPOT textures
- freedreno: use renderonly path for buffers allocated with modifiers
- freedreno: catch failing fd_blit and fallback to software blit
- mesa: add GL_AMD_compressed_ATC_texture support
- gallium: add ATC format support
- llvmpipe, softpipe: no support for ATC textures
- st/mesa: add ATC support
- freedreno: a3xx: add GL_AMD_compressed_ATC_texture support
- freedreno: a2xx: add GL_AMD_compressed_ATC_texture support
- svga: add new ATC formats to the format conversion table
- freedreno: a2xx: fix builtin blit program compilation
- freedreno: a2xx: disable PIPE_CAP_PACKED_UNIFORMS
- freedreno: a2xx: use nir_lower_io for TGSI shaders
- freedreno: a2xx: enable batch reordering
- freedreno: a2xx: same gmem2mem sequence for all tiles
- nir: improve convert_yuv_to_rgb
- freedreno/ir3: fix input ncomp for vertex shaders

Jordan Justen (22):

- iris: Set num_uniforms in bytes
- iris/compute: Set mask bits on PIPELINE_SELECT
- iris: Add IRIS_DIRTY_CONSTANTS_CS
- iris: Add iris_restore_compute_saved_bos
- iris/compute: Add MEDIA_STATE_FLUSH following WALKER
- iris/compute: Flush compute batches
- iris/compute: Get group counts from grid->grid
- iris/program: Don't try to push ubo ranges for compute
- iris/compute: Wait on compute batch when mapping
- iris/compute: Provide binding table entry for gl_NumWorkGroups
- iris/compute: Flush compute batch on memory-barriers
- iris/compute: Push subgroup-id
- iris/compute: Support indirect compute dispatch
- iris: Emit default L3 config for the render pipeline
- genxml/gen_bits_header.py: Use regex to strip no alphanum chars
- genxml: Remove extra space in gen4/45/5 field name
- iris: Add gitlab-ci build testing
- iris: Always use in-tree i915_drm.h

- nir: Add int64/doubles options into nir_shader_compiler_options
- intel/compiler: Move int64/doubles lowering options
- scons: Generate float64_glsl.h for glsl_to_nir fp64 lowering
- intel/genxml: Support base-16 in value & start fields in gen_sort_tags.py

Jose Maria Casanova Crespo (4):

- iris: Enable ARB_shader_draw_parameters support
- glsl: fix typos in comments "transfor" -> "transform"
- glsl: TCS outputs can not be transform feedback candidates on GLES
- iris: setup EdgeFlag Vertex Element when needed.

José Fonseca (1):

- scons: Workaround failures with MSVC when using SCons 3.0.[2-4].

Juan A. Suarez Romero (22):

- anv/cmd_buffer: check for NULL framebuffer
- nir: move ALU instruction before the jump instruction
- nir: remove jump from two merging jump-ending blocks
- genxml: add missing field values for 3DSTATE_SF
- anv: advertise 8 subpixel precision bits
- nir/spirv: return after emitting a branch in block
- anv: destroy descriptor sets when pool gets reset
- nir: deref only for OpTypePointer
- anv: advertise 8 subtexel/mipmap precision bits
- nir/xfb: do not use bare interface type
- meson: Add dependency on genxml to anvil genfiles
- Revert "intel/compiler: split is_partial_write() into two variants"
- spirv: add missing SPV_EXT_descriptor_indexing capabilities
- radv: enable descriptor indexing capabilities
- anv: enable descriptor indexing capabilities
- Update version to 19.1.0-rc1
- Update version to 19.1.0-rc2
- cherry-ignore: radeonsi: update buffer descriptors in all contexts after buffer invalidation
- Update version to 19.1.0-rc3
- Update version to 19.1.0-rc4
- Update version to 19.1.0-rc5
- Update version to 19.1.0

Julien Isorce (5):

- gallium: add resource_get_info to pipe_screen

- radeonsi: implement resource_get_info
- st/va: properly set stride and offset in vlVaDeriveImage
- r600: implement resource_get_info
- st/va: check resource_get_info nullity in vlVaDeriveImage

Józef Kucia (3):

- mesa: Fix GL_NUM_DEVICE_UUIDS_EXT
- radv: Fix driverUUID
- radv: clear vertex bindings while resetting command buffer

Karol Herbst (82):

- nvc0/ir: replace cvt instructions with add to improve shader performance
- gk104/ir: Use the new rcp/rsq in library
- gm107/ir: add fp64 rcp
- gm107/ir: add fp64 rsq
- gallium: add PIPE_CAP_MAX_VARYINGS
- st/mesa: require RGBA2, RGB4, and RGBA4 to be renderable
- glsl_type: initialize offset and location to -1 for glsl_struct_field
- nir/opt_if: don't mark progress if nothing changes
- clover: update ICD table to support everything up to 2.2
- nir: replace magic numbers with M_PI
- nir/spirv: improve parsing of the memory model
- nir: add support for address bit sized system values
- nir/vtn: add support for SpvBuiltInGlobalLinearId
- nir/spirv: initial handling of OpenCL.std extension opcodes
- prog_to_nir: fix write from vps to FOG
- nvc0: print the shader type when dumping headers
- nv50/ir: move common converter code in base class
- nv50/ir: add lowering helper
- nouveau: add support for nir
- nouveau: fix nir and TGSI shader cache collision
- nv50/ir/nir: run some passes to make the conversion easier
- nv50/ir/nir: track defs and provide easy access functions
- nv50/ir/nir: add nir type helper functions
- nv50/ir/nir: run assignSlots
- nv50/ir/nir: add loadFrom and storeTo helpler
- nv50/ir/nir: parse NIR shader info
- nv50/ir/nir: implement nir_load_const_instr

- nv50/ir/nir: add skeleton for nir_intrinsic_instr
- nv50/ir/nir: implement nir_alu_instr handling
- nv50/ir/nir: implement nir_intrinsic_load_uniform
- nv50/ir/nir: implement nir_intrinsic_store_(per_vertex_)output
- nv50/ir/nir: implement load_(interpolated_)input/output
- nv50/ir/nir: implement intrinsic_discard(_if)
- nv50/ir/nir: implement loading system values
- nv50/ir/nir: implement nir_ssa_undef_instr
- nv50/ir/nir: implement nir_instr_type_tex
- nv50/ir/nir: add skeleton getOperation for intrinsics
- nv50/ir/nir: implement vote and ballot
- nv50/ir/nir: implement variable indexing
- nv50/ir/nir: implement geometry shader nir_intrinsics
- nv50/ir/nir: implement nir_intrinsic_load_ubo
- nv50/ir/nir: implement ssbo intrinsics
- nv50/ir/nir: implement images
- nv50/ir/nir: add memory barriers
- nv50/ir/nir: implement load_per_vertex_output
- nv50/ir/nir: implement intrinsic shader_clock
- nv50/ir/nir: handle user clip planes for each emitted vertex
- nv50ir/nir: move immediates before use
- glsl: add packed for struct types
- glsl: add cl_size and cl_alignment
- nir/lower_locals_to_regs: cast array index to 32 bit
- nir/spirv: handle kernel function parameters
- nir/spirv: support physical pointers
- nir: add support for gather offsets
- nv50/ir/nir: support gather offsets
- nir/lower_tex: Add support for tg4 offsets lowering
- nir/print: fix printing the image_array intrinsic index
- nir/validate: validate that tex deref sources are actually derefs
- v3d: prefer using nir_src_comp_as_int over nir_src_as_const_value
- panfrost/midgard: use nir_src_is_const and nir_src_as_uint
- glsl/standalone: add GLES3.1 and GLES3.2 compatibility
- nir: move brw_nir_rewrite_image_intrinsic into common code
- glsl_to_nir: handle bindless textures

- glsl/nir: fetch the type for images from the deref instruction
- glsl/nir: add support for lowering bindless images_derefs
- nv50/ir/nir: handle bindless texture
- nv50/ir/nir: add support for bindless images
- nvc0/nir: enable bindless texture
- lima: add bool parameter to type_size function
- amd/nir: some cleanups
- radv: use nir constant helpers
- intel/nir: use nir_src_is_const and nir_src_as_uint
- freedreno/ir3: use nir_src_as_uint in a few places
- lima: use nir_src_as_float
- nir/builder: Move nir_imm_vec2 from blorp into the builder
- nir/loop_analyze: use nir_const_value.b for boolean results, not u32
- spirv: reduce array size in vtn_handle_constant
- nir: make nir_const_value scalar
- vtn: handle bitcast with pointer src/dest
- nir: Add a nir_builder_alu variant which takes an array of components
- nir: Add nir_op_vec helper
- spirv/cl: support vload/vstore

Kasireddy, Vivek (3):

- nir/lower_tex: Add support for XYUV lowering
- dri: Add XYUV8888 format
- i965: Add support for sampling from XYUV images

Kenneth Graunke (872):

- st/mesa: Set pipe_image_view::shader_access in PBO readpixels.
- st/nir: Move varying setup code to a helper function.
- st/nir: Make new helpers for constructing built-in NIR shaders.
- st/mesa: Add a NIR version of the drawpixels/bitmap VS copy shader.
- st/mesa: Add NIR versions of the drawpixels Z/stencil fragment shaders.
- st/mesa: Add NIR versions of the clear shaders.
- st/mesa: Add a NIR version of the OES_draw_texture built-in shaders.
- st/mesa: Add NIR versions of the PBO upload/download shaders.
- program: Use u_bit_scan64 in prog_to_nir.
- program: Extend prog_to_nir handle system values.
- nir: Record info->fs.pixel_center_integer in lower_system_values
- compiler: Mark clip/cull distance arrays as compact before lowering.

- nir: Bail on clip/cull distance lowering if GLSL IR already did it.
- nir: Avoid clip/cull distance lowering multiple times.
- nir: Avoid splitting compact arrays into per-element variables.
- st/nir: Call nir_lower_clip_cull_distance_arrays().
- gallium: Add a PIPE_CAP_NIR_COMPACT_ARRAYS capability bit.
- nouveau: Silence unhandled cap warnings
- st/mesa: Limit GL_MAX_[NATIVE_]PROGRAM_PARAMETERS_ARB to 2048
- glsl: Allow gl_nir_lower_samplers*() without a gl_shader_program
- glsl: Don't look at sampler uniform storage for internal vars
- i965: Call nir_lower_samplers for ARB programs.
- st/nir: Pull sampler lowering into a helper function.
- st/nir: Lower sampler derefs for builtin shaders.
- st/nir: Use sampler derefs in built-in shaders.
- program: Make prog_to_nir create texture/sampler derefs.
- nir: Use sampler derefs in drawpixels and bitmap lowering.
- nir: Gather texture bitmasks in gl_nir_lower_samplers_as_deref.
- i965: Drop unnecessary 'and' with prog->SamplerUnits
- i965: Use info->textures_used instead of prog->SamplersUsed.
- mesa: Advertise EXT_float_blend in ES 3.0+ contexts.
- anv: Put MOCS in the correct location
- spirv: Eliminate dead input/output variables after translation.
- nir: Don't reassociate add/mul chains containing only constants
- compiler: Make is_64bit(GL_*) helper more broadly available
- mesa: Align doubles to a 64-bit starting boundary, even if packing.
- radeonsi: Go back to using llvm.pow intrinsic for nir_op_fpow
- st/mesa: Copy VP TGSI tokens if they exist, even for NIR shaders.
- nir: Don't forget if-uses in new nir_opt_dead_cf liveness check
- iris: Initial commit of a new 'iris' driver for Intel Gen8+ GPUs.
- iris: viewport state, sort of
- iris: port over batchbuffer updates
- iris: initial render state upload
- iris: packing with valgrind.
- iris: merge pack
- iris: initial gpu state, merges
- iris: RASTER + SF + some CLIP, fix DIRTY vs. NEW
- iris: scissors

- iris: SF_CLIP_VIEWPORT
- iris: Surfaces!
- iris: sampler views
- iris: stipples and vertex elements
- iris: framebuffers
- iris: don't segfault on !old_cso
- iris: fix SF_CL length
- iris: a bit of depth
- iris: some draw info, vbs, sample mask
- iris: fix crash - CSO binding can be NULL (when destroying context)
- iris: COLOR_CALC_STATE
- iris: sampler states
- iris: emit 3DSTATE_SAMPLER_STATE_POINTERS
- iris: basic push constant alloc
- iris: some program code
- iris: linear resources
- iris: maps
- iris: shader debug log
- iris: drop unused field
- iris: make an ice->render_batch field
- iris: disable execbuf for now
- iris: delete iris_pipe.c, shuffle code around
- iris: init the batch!
- iris: fix/rework line stipple
- iris: actually save VBs
- iris: msaa sample count packing problems
- iris: fix prim type
- iris: fix bogus index buffer reference
- iris: draw->restart_index is uninitialized if PR is not enabled
- iris: parse INTEL_DEBUG
- iris: reworks, FS compile pieces
- iris: import program cache code
- iris: do the FS... asserts because we don't lower uniforms yet
- iris: lower io
- iris: make iris_batch target a particular ring
- iris: kill iris_new_batch

- iris: move MAX defines to iris_batch.h
- iris: bit of SBA code
- iris: flag SBA updates when instruction BO changes
- iris: try and have an iris address
- iris: so, sba then.
- iris: reference VB BOs
- iris: VB addresses
- iris: DEBUG=bat
- iris: VB fixes
- iris: actually APPEND commands, not stomp over the top and never incr
- iris: actually flush the commands
- iris: actually advance forward when emitting commands
- iris: initialize dirty bits to ~0ull
- iris: hack to stop crashing on samplers for now
- iris: fix indentation
- iris: fix assert
- iris: fix VBs
- iris: vertex packet fixes
- iris: fix VF instancing length so we don't get garbage in batch
- iris: 3DPRIMITIVE fields
- iris: bind_state -> compute state
- iris: scissor slots
- iris: some shader bits
- iris: promote iris_program_cache_item to iris_compiled_shader
- iris: actually save derived state
- iris: emit shader packets
- iris: convert IRIS_DIRTY_* to #defines
- iris: don't forget about TE
- iris: reorganize commands to match brw
- iris: initial gpu state
- iris: WM.
- iris: index buffer BO
- iris: more comes from bits filled in
- iris: drop const from prog data parameters
- iris: softpin some things
- iris: use vtbl to avoid multiple symbols, fix state base address

- iris: fix SBA
- iris: move key pop to state module
- iris: bits of WM key
- iris: shuffle comments
- iris: no NEW_SBA
- iris: rewrite program cache to use u_upload_mgr
- iris: actually destroy the cache
- iris: actually softpin at an address
- iris: actually set KSP offsets
- iris: URB configs.
- iris: dummy constants
- iris: blend state
- iris: alpha testing in PSB
- iris: basic SBE code
- iris: warning fixes
- iris: fix silly unused batch with addr macro
- iris: render targets!
- iris: don't do samplers for disabled stages
- iris: smaller blend state
- iris: actually pin the instruction cache buffers
- iris: compctrl
- iris: more sketchy SBE
- iris: fix dmabuf retval comparisons
- iris: more SF CL VPs
- iris: catastrophic state pointer mistake
- iris: fix extents
- iris: write DISABLES are not write ENABLES... whoops
- iris: sample mask... not 0.
- iris: uniform bits... badly
- iris: warn if execbuf fails
- iris: NOOP pad batches correctly
- iris: decode batches if they fail to submit
- iris: enable a few more formats
- iris: set strides on transfers
- iris: stop adding 9 to our varyings
- iris: bufmgr updates.

- iris: some thinking about binding tables
- iris: Soft-pin the universe
- iris: fix icache memzone
- iris: dump gtt offset in dump_validation_list
- iris: Also set SUPPORTS_48B? Not sure if necessary.
- iris: more uploaders
- iris: rewrite to use memzones and not relocs
- iris: set EXEC_OBJECT_WRITE
- iris: include p_defines.h in iris_bufmgr.h
- iris: binders
- iris: hook up batch decoder
- iris: binder fixes
- iris: decoder fixes
- iris: update vb BO handling now that we have softpin
- iris: validation dumping improvements
- iris: canonicalize addresses.
- iris: delete more trash
- iris: allocate SURFACE_STATEs up front and stop streaming them
- iris: same treatment for sampler views
- iris: assemble SAMPLER_STATE table at bind time
- iris: fix a scissor bug
- iris: SBA once at context creation, not per batch
- iris: TES stash
- iris: isv freeing fixes
- iris: set sampler views
- iris: decoder fixes
- iris: better BT asserts
- iris: increase allocator alignment
- iris: fix index
- iris: port bug fix from i965
- iris: fixes from i965
- iris: fixes
- iris: crazy pipe control code
- iris: bo reuse
- iris: vma fixes - don't free binder address
- iris: vma - fix assert

- iris: better SBE
- iris: fix texturing!
- iris: Move get_command_space to iris_batch.c
- iris: Defines for base addresses rather than numbers everywhere
- iris: pull in newer comments
- iris: copy over i965's cache tracking
- iris: move bo_offset_from_sba
- iris: bits of blorp code
- iris: more blitting code to make readpixels work
- iris: drop bogus binder free
- iris: fix sampler view crashes
- iris: more blorp
- iris: fix blorp prog data crashes
- iris: add INTEL_DEBUG=reemit
- iris: drop the 48b printout, we never use anything else
- iris: hacky flushing for now
- iris: linear staging buffers - fast CPU access…
- iris: make blorp pin the binder
- iris: blorp URB
- iris: no more drawing rectangle in blorp
- iris: assert surf init
- iris: some depth stuff :(
- iris: bump GL version to 4.2
- iris: uniforms for VS
- iris: proper length for VE packet?
- iris: proper # of uniforms
- iris: properly reject formats, fixes RGB32 rendering with texture float
- iris: blorp bug fixes
- iris: delete growing code and just die for now
- iris: just turn batch reset_and_clear_caches into reset
- iris: chaining not growing
- iris: caps
- iris: fix batch chaining…
- iris: fix decoding and undo testing code
- iris: Lower the max number of decoded VBO lines
- iris: fix whitespace

- iris: fix 3DSTATE_VERTEX_ELEMENTS length
- iris: more depth stuffs…
- iris: fix VF INSTANCING length
- iris: util_copy_framebuffer_state (ported from Rob's v3d patches)
- iris: transfers
- iris: flush always
- iris: maybe slightly less boats uniforms
- iris: fix constant packet length to match i965
- iris: better ubo handling
- iris: completely rewrite binder
- iris: have more than one const_offset
- iris: make surface states for cbufs
- iris: fill out pull constant buffers
- iris: fix pull bufs that aren't the first user upload
- iris: use u_transfer helpers for now
- iris: better VFI
- iris: fix release builds
- iris: drop assert for now
- iris: disable __gen_validate_value in release mode
- iris: allow mapped buffers during execution (faster)
- iris: comment about reemitting and flushing
- iris: state cleaning
- iris: untested index buffer upload
- iris: delete some pointless STATIC_ASSERTS
- iris: untested SAMPLER_STATE pin BO fix
- iris: put back the always flush - fixes some things :(
- iris: save pointers to streamed state resources
- iris: fix the validation list on new batches
- iris: flag DIRTY_WM properly
- iris: bindings dirty tracking
- iris: some dirty fixes
- iris: clear dirty
- iris: plug leaks
- iris: more leak fixes
- iris: pc fixes
- iris: remove 4 bytes of padding in iris_compiled_shader

- iris: rzalloc iris_compiled_shader so memcmp works even if padding creeps in
- iris: don't leak sampler state table resources
- iris: don't leak keyboxes when searching for an existing program
- iris: indentation
- iris: use pipe resources not direct BOs
- iris: clean up some warnings so I can see through the noise
- iris: print binder utilization in INTEL_DEBUG=submit
- iris: redo VB CSO a bit
- iris: print refcounts in INTEL_DEBUG=submit
- iris: support signed vertex buffer offsets
- iris: fix major refcounting bug with resources
- iris: fix caps so tests run again
- iris: avoid crashing on unbound constant resources
- iris: emit 3DSTATE_SBE_SWIZ
- iris: max VP index
- iris: fix viewport counts and settings
- iris: fix num viewports to be based on programs
- iris: fix VP iteration
- iris: scissor count fixes
- iris: actually init num_viewports
- iris: print second batch size separately
- iris: don't always flush
- iris: Handle batch submission failure "better"
- iris: bad inherited comments
- iris: colorize batchbuffer failures to make them stand out
- iris: iris - fix QWord aligned endings after batch chaining rework
- iris: tidy comments about mirroring modes
- iris: Disable unsupported mirror clamp modes
- iris: fix fragcoord ytransform
- iris: better boxing on maps
- iris: clears
- iris: rework DEBUG_REEMIT
- iris: shader dirty bits
- iris: clear fix
- iris: fall back to u_generate_mipmap
- iris: implement copy image

- iris: lightmodel flat
- iris: maybe-flush before blorp operations
- iris: fix provoking vertex ordering
- iris: larger polygon offset
- iris: TES uniform fixes
- iris: geometry shader support
- iris: don't emit garbage 3DSTATE_VERTEX_BUFFERS when there aren't any
- iris: fix 3DSTATE_VERTEX_ELEMENTS / VF_INSTANCING for 0 elements
- iris: fix GS dispatch mode
- iris: depth clears
- iris: null surface for unbound textures
- iris: state ref tuple
- iris: don't include binder in surface VMA range
- iris: border color memory zone :(
- iris: implement border color, fix other sampler nonsense
- iris: dead pointer
- iris: just malloc one iris_genx_state instead of a bunch of oddball pieces
- iris: SBE change stash
- iris: fix zoffset asserts with 2DArray/Cube
- iris: rename map->stride
- iris: actually set cube bit properly
- iris: keep DISCARD_RANGE
- iris: actually handle array layers in blits
- iris: comment out l/a/i/la
- iris: fix clip flagging on fb changes
- iris: fix depth bounds clamp enables
- iris: don't crash on shader perf logs
- iris: slab allocate transfers
- iris: rearrange iris_resource.h
- iris: Implement 3DSTATE_SO_DECL_LIST
- iris: SO buffers
- iris: streamout
- iris: set even if no outputs
- iris: bother setting program_string_id…
- iris: fix SO_DECL_LIST
- iris: actually pin the buffers

- iris: fix sample mask for MSAA-off
- iris: disable 6x MSAA support
- iris: multislice transfer maps
- iris: fix CC_VIEWPORT
- iris: draw indirect support?
- iris: save query type
- iris: bits of multisample program key
- iris: s/hwcso/state/g
- iris: bind state helper function
- iris: NOS mechanics
- iris: record FS NOS
- iris: fix crash
- iris: fix sampler views of TBOs
- iris: fix texture buffer stride
- iris: TES program key inputs
- iris: compile a TCS…don't bother with passthrough yet
- iris: don't emit SO_BUFFERS and SO_DECL_LIST unless streamout is enabled
- iris: vertex ID, instance ID
- iris: fix SGVS when there are no valid vertex elements
- iris: fill out MAX_PATCH_VERTICES
- iris: assert about passthrough shaders to make this easier to detect
- iris: fix EmitNoIndirect
- iris: fix Z24
- iris: reemit blend state for alpha test function changes
- iris: point sprite enables
- iris: hack around samples confusion
- iris: fix blorp filters
- iris: expose more things that we already support
- iris: fix msaa flipping filters
- iris: export get_shader_info
- iris: implement set_shader_buffers
- iris: emit binding table for atomic counters and SSBOs
- iris: shorten loop
- iris: unbind compiled shaders if none are present
- iris: fix TBO alignment to match 965
- iris: enable SSBOs

- iris: fix SSBO indexing
- iris: fix for disabling ssbos
- iris: update bindings when changing programs
- iris: drop unused bo parameter
- iris: implement texture/memory barriers
- iris: Don't reserve new binding table section unless things are dirty
- iris: update a todo comment
- iris: BIG OL' HACK for UBO updates
- iris: enable texture gather
- iris: Avoid croaking when trying to create FBO surfaces with bad formats
- iris: fix GS output component limit
- iris: drop pipe_shader_state
- iris: fix sample mask
- iris: cube arrays are cubes too
- iris: we don't support textureGatherOffsets, need it lowered
- iris: add minor comments
- iris: comment everything
- iris: sync bugfixes from brw_bufmgr
- iris: remember to set bo->userptr
- iris: rename ring to engine
- iris: simplify batch len qword alignment
- iris: get angry about execbuf failures
- iris: fill out more caps
- iris: depth or stencil fixes
- iris: clear stencil
- iris: actually emit stencil packets
- iris: allow S8 as a stencil format
- iris: WTF transfers
- iris: use u_transfer_helper for depth stencil packing/unpacking
- iris: drop stencil handling now that u_transfer_helper does it
- iris: refcounting, who needs it?
- iris: actually do stencil blits
- iris: say no to more formats
- iris: deal with Marek's new MSAA caps
- iris: we can do multisample Z resolves
- iris: Convert RGBX to RGBA for rendering.

- iris: disallow RGB32 formats too
- iris: Fix tiled memcpy for cubes. . . and for array slices
- iris: blorp blit multiple slices
- iris: assert depth is 1 in resource_copy_region
- iris: call maybe_flush for each blorp operation
- iris: implement ARB_clear_texture
- iris: last VUE map NOS, handle > 16 FS inputs
- iris: drop dead assignments
- iris: drop pwrite
- iris: port non-bucket alignment bugfix
- iris: don't emit SBE all the time
- iris: rename pipe to base
- iris: Drop bogus sampler state saving
- iris: move iris_shader_state from ice->shaders.state to ice->state.shaders
- iris: Move things to iris_shader_state
- iris: Move iris_sampler_view declaration to iris_resource.h
- iris: track depth/stencil writes enabled
- iris: use consistent copyright formatting
- iris: Move cache tracking to iris_resolve.c
- iris: proper cache tracking
- iris: precompute hashes for cache tracking
- iris: Reduce binder alignment from 64 to 32
- iris: reenable R32G32B32 texture buffers
- iris: z_res -> s_res
- iris: implement get_sample_position
- iris: fix line-aa-width
- iris: try to hack around binder issue
- iris: fix sampler state setting
- iris: big old hack for tex-miplevel-selection
- iris: use linear for 1D textures
- iris: handle level/layer in direct maps
- iris: fix crash when binding optional shader for the first time
- iris: Skip primitive ID overrides if the shader wrote a custom value
- iris: fix blend state memcpy
- iris: new caps
- iris: use Eric's new caps helper

- iris: Allow inlining of require/get_command_space
- iris: skip over whole function if dirty == 0
- iris: don't unconditionally emit 3DSTATE_VF / 3DSTATE_VF_TOPOLOGY
- iris: fix constant buffer 0 to be absolute
- iris: set EXEC_OBJECT_CAPTURE on all driver internal buffers
- iris: fix null FB and unbound tex surface state addresses
- iris: Support multiple binder BOs, update Surface State Base Address
- iris: fix SO offset writes for multiple streams
- iris: update comments for multibinder
- iris: move binder pinning outside the dirty == 0 check
- iris: re-pin binding table contents if we didn't re-emit them
- iris: enable ARB_enhanced_layouts
- iris: refactor LRIs in context setup
- iris: initialize "don't suck" bits, as Ben likes to call them
- iris: totally untested icelake support
- iris: refactor program CSO stuff
- iris: silence const warning
- iris: fix context restore of 3DSTATE_CONSTANT ranges
- iris: properly re-pin stencil buffers
- iris: delete bogus comment
- iris: inherit the index buffer properly
- iris: use 0 for TCS passthrough program string ID
- iris: rw_bo for pipe controls
- iris: LRM/SRM/SDI hooks
- iris: initial query code
- iris: gen10+ workarounds and break fix
- iris: results write
- iris: flush batch when asking for result via QBO
- iris: fix random failures via CS stall... but why?
- iris: gpr0 to bool
- iris: play chicken with timer queries for now
- iris: pipeline stats
- iris: primitives generated query support
- iris: drop explicit pinning
- iris: timestamps
- iris: ... and SO prims emitted queries

- iris: glGet timestamps, more correct timestamps
- iris: Need to | 1 when asking for timestamps
- iris: 36-bit overflow fixes
- iris: early return properly
- iris: better query file comment
- iris: magic number 36 -> #define
- iris: Enable ARB_shader_vote
- iris: just mark snapshots_landed from the CPU
- iris: drop a bunch of pipe_sampler_state stuff we don't need
- iris: vma_free bo->size, not bo_size
- iris: don't mark contains_draw = false when chaining batches
- iris: fix Z32_S8 depth sampling
- iris: stencil texturing
- iris: force persample interp cap
- iris: pipe to scs -> iris_pipe.h
- iris: inline stage_from_pipe to avoid unused warnings
- iris: add gen11 to genX_call
- iris: Allow PIPE_CONTROL with Stall at Scoreboard and RT flush
- iris: rework format translation apis
- iris: Use R/RG instead of I/L/A when sampling
- iris: enable I/L formats
- iris: X32_S8X24 :/
- iris: set the binding table size
- iris: lower storage image derefs
- iris: implement set_shader_images hook
- iris: bother with BTIs
- iris: set image access correctly
- iris: actually set image access
- iris: null for non-existent cbufs
- iris: move images next to textures in binding table
- iris: advertise GL_ARB_shader_texture_image_samples
- iris: Enable fb fetch
- iris: initial compute caps
- iris: yes
- iris: drop dead format //'s
- iris: drop XXX's about swizzling

- iris: little bits of compute basics
- iris: drop XXX that Jordan handled
- iris: drop unnecessary #ifdefs
- iris: leave XXX about unnecessary binding table uploads
- iris: bail if SLM is needed
- iris: fix whitespace
- iris: XXX for compute state tracking :/
- iris: rewrite grid surface handling
- iris: better dirty checking
- iris: don't let render/compute contexts stomp each other's dirty bits
- iris: hack to avoid memorybarriers out the wazoo
- iris: do PIPELINE_SELECT for render engine, add flushes, GLK hacks
- iris: fix SBA flushing by refactoring code
- iris: try and avoid pointless compute submissions
- iris: fix UBOs with bindings that have an offset
- iris: flag CC_VIEWPORT when changing num viewports
- iris: fix SF_CLIP_VIEWPORT array indexing with multiple VPs
- iris: Fix texture buffer / image buffer sizes.
- iris: Clamp UBO and SSBO access to the actual BO size, for safety
- iris: Move snapshots_landed to the front.
- iris: Fix off by one in scissoring, empty scissors, default scissors
- iris: Fall back to 1x1x1 null surface if no framebuffer supplied
- iris: SO_DECL_LIST fix
- iris: Fix refcounting of grid surface
- iris: delete dead code
- iris: fix overhead regression from "don't stomp each other's dirty bits"
- iris: allow binding a null vertex buffer
- iris: Flag constants dirty on program changes
- iris: Disable a PIPE_CONTROL workaround on Icelake
- iris: Enable ARB_shader_stencil_export
- iris: Enable A8/A16_UNORM in an inefficient manner
- iris: Drop B5G5R5X1 support
- iris: Use at least 1x1 size for null FB surface state.
- iris: Cross-link iris_batches so they can potentially flush each other
- iris: cross batch flushing
- iris: Don't leak the compute batch

- iris: Actually create/destroy HW contexts
- iris: Enable msaa_map transfer helpers
- iris: tidy more warnings
- iris: implement scratch space!
- iris: Fix MSAA smooth points
- iris: Fix TextureBarrier
- iris: Fix multiple RTs with non-independent blending
- iris: partial set_query_active_state
- iris: Print the batch name when decoding
- iris: Clone the NIR
- iris: Defer cbuf0 upload to draw time
- iris: drop unnecessary param[] setup from iris_setup_uniforms
- iris: add param domain defines
- iris: fill out params array with built-ins, like clip planes
- iris: only bother with params if there are any…
- iris: lower user clip planes
- iris: hook up key stuff for clip plane lowering
- iris: fix system value remapping
- iris: dodge backend UCP lowering
- iris: bypass params and do it ourselves
- iris: actually upload clip planes.
- iris: fix num clip plane consts
- iris: fix more uniform setup
- iris: drop iris_setup_push_uniform_range
- iris: enable push constants if we have sysvals but no uniforms
- iris: regather info so we get CLIP_DIST slots, not CLIP_VERTEX
- iris: don't support pull constants.
- iris: don't trip on param asserts
- iris: drop param stuffs
- iris: don't forget to upload CS consts
- iris: fix sysval only binding tables
- iris: only clip lower if there's something to clip against
- iris: leave another TODO
- iris: Fix SourceAlphaBlendFactor
- iris: "Fix" transfer maps of buffers
- iris: Fix independent alpha blending.

- iris: more TODO
- iris: scissored and mirrored blits
- iris: more todo notes
- iris: Fix TCS/TES slot unification
- iris: properly pin stencil buffers
- iris: Fix SLM
- iris: Use iris_use_pinned_bo rather than add_exec_bo directly
- iris: Combine iris_use_pinned_bo and add_exec_bo
- iris: Avoid cross-batch synchronization on read/reads
- iris: Avoid synchronizing due to the workaround BO
- iris: replace vestiges of fence fds with newer exec_fence API
- iris: Drop vestiges of throttling code
- iris: Hang on to the last batch's sync-point, so we can wait on it
- iris: Add wait fences to properly sync between render/compute
- iris: leave a TODO
- iris: flush the compute batch too if border pool is redone
- iris: put render batch first in fence code
- iris: Put batches in an array
- iris: PIPE_CONTROL workarounds for GPGPU mode
- iris: RT flush for memorybarrier with texture bit
- iris: update comment
- iris: Enable ctx->Const.UseSTD430AsDefaultPacking
- iris: Lie about indirects
- iris: Fix buffer -> buffer copy_region
- iris: Fix VIEWPORT/LAYER in stream output info
- iris: Do the 48-bit vertex buffer address invalidation workaround
- iris: drop long dead XXX comment
- iris: Track a binding history for buffer resources
- iris: add iris_flush_and_dirty_for_history
- iris: Flush for history at various moments
- iris: Re-pin even if nothing is dirty
- iris: fix prototype warning
- iris: export iris_upload_shader
- iris: fix comment location
- iris: Use wrappers for create_xs_state rather than a switch statement
- iris: rework program cache interface

- iris: Enable precompiles
- iris: Use program's num textures not the state tracker's bound
- iris: drop pull constant binding table entry
- iris: add assertions about binding table starts
- iris: add an extra BT assert from Chris Wilson
- iris: actually flush for storage images
- iris: fix some SO overflow query bugs and tidy the code a bit
- iris: drop key_size_for_cache
- iris: for BLORP, only use the predicate enable bit when USE_BIT
- iris: check query first
- iris: fix conditional compute, don't stomp predicate for pipelined queries
- iris: Rework tiling/modifiers handling
- iris: Fix failed to compile TCS message
- iris: Destroy transfer helper on screen teardown
- iris: Destroy the border color pool
- iris: Unref unbound_tex resource
- iris: Fix IRIS_MEMZONE_COUNT to exclude the border color pool
- iris: Destroy the bufmgr
- iris: Stop leaking iris_uncompiled_shaders like mad
- iris: move some non-buffer case code in a bit
- iris: Don't bother considering if the underlying surface is a cube
- iris: fix alpha channel for RGB BC1 formats
- iris: fix dma buf import strides
- iris: CS stall for stream out -> VB
- iris: make clipper statistics dynamic
- iris: reject all clipping when we can't use streamout render disabled
- iris: omask can kill
- iris: reemit SBE when sprite coord origin changes
- iris: re-pin inherited streamout buffers
- iris: Fix NOS mechanism
- iris: fix overhead regression from flushing for storage images
- iris: fix set_sampler_views to not unbind, be better about bounds
- iris: Fix set_sampler_views with start > 0
- iris: Replace num_textures etc with a bitmask we can scan
- iris: Drop continues in resolve
- iris: Fix clear dimensions

- iris: Clamp viewport extents to the framebuffer dimensions
- iris: Enable guardband clipping
- iris: Fix primitive generated query active flag
- iris: Always do rasterizer discard in clipper
- iris: override alpha to one src1 blend factors
- iris: handle PatchVerticesIn as a system value.
- iris: rewrite set_vertex_buffer and VB handling
- iris: Reorder LRR parameters to have dst first.
- iris: Add _MI_ALU helpers that don't paste
- iris: Don't bother packing 3DSTATE_SO_BUFFER at create time
- iris: Move iris_stream_output_target def to iris_context.h
- iris: only get space for one offset in stream output targets
- iris: Implement DrawTransformFeedback()
- iris: drop unnecessary genx->streamout field
- iris: Fix for PIPE_CAP_SIGNED_VERTEX_BUFFER_OFFSET
- iris: Fix the prototype for iris_bo_alloc_tiled
- iris: don't print the pointer in INTEL_DEBUG=submit
- iris: Use a surface state fill helper
- iris: Make a alloc_surface_state helper
- iris: whitespace fixes
- iris: Track blend enables, save outbound for resolve code
- iris: always pin the binder... in the compute context, too.
- iris: delete finished comments
- iris: pin and re-pin the scratch BO
- iris: more dead comments
- iris: only mark depth/stencil as writable if writes are actually enabled
- iris: better MOCS
- iris: Fix scratch space allocation on Icelake.
- iris: Only resolve inputs for actual shader stages
- iris: Add a more long term TODO about timebase scaling
- iris: Fix compute scratch pinning
- iris: Delete bogus comment about cube array counting.
- iris: Fix framebuffer layer count
- iris: Don't enable push constants just because there are system values
- iris: Don't make duplicate system values
- iris: Fill out brw_image_params for storage images on Broadwell

- iris: Fix surface states for Gen8 lowered-to-untype images
- iris: Leave a comment about why Broadwell images are broken
- iris: Implement multi-slice copy_region
- iris: Flush the render cache in flush_and_dirty_for_history
- iris: Handle PIPE_TRANSFER_DISCARD_WHOLE_RESOURCE somewhat
- iris: Don't check other batches for our batch BO
- iris: Drop a dead comment
- iris: Delete genx->bound_vertex_buffers
- iris: Fix Broadwell WaDividePSInvocationCountBy4
- iris: Use new PIPE_STAT_QUERY enums rather than hardcoded numbers.
- iris: Switch to the new PIPELINE_STATISTICS_QUERY_SINGLE capability
- iris: fail to create screen for older unsupported HW
- iris: Allow sample mask of 0
- iris: Don't enable smooth points when point sprites are enabled
- iris: Assert about blits with color masking
- iris: Pay attention to blit masks
- iris: CS stall on VF cache invalidate workarounds
- iris: Fix SO issue with INTEL_DEBUG=reemit, set fewer bits
- iris: Don't whack SO dirty bits when finishing a BLORP op
- iris: Fix memzone_for_address for the surface and binder zones
- iris: Do binder address allocations per-context, not globally.
- iris: Zero the compute predicate when changing the render condition
- iris: Remap stream output indexes back to VARYING_SLOT_*.
- iris: Enable PIPE_CAP_COMPACT_ARRAYS
- iris: Drop comment about ISP_DIS
- iris: Drop dead state_size hash table
- iris: Unreference some more things on state module teardown
- iris: minor tidying
- iris: Fix bug in bound vertex buffer tracking
- iris: Implement ALT mode for ARB_{vertex,fragment}_shader
- iris: Add a timeout_nsec parameter, rename check_syncpt to wait_syncpt
- iris: Fix accidental busy-looping in query waits
- iris: Use READ_ONCE and WRITE_ONCE for snapshots_landed
- iris: Make a iris_batch_reference_signal_syncpt helper function.
- iris: Add PIPE_CAP_MAX_VARYINGS
- iris: rework num textures to util_lastbit

- iris: Stop chopping off the first nine characters of the renderer string
- iris: Drop XXX about alpha testing
- iris: Set 3DSTATE_WM::ForceThreadDispatchEnable
- iris: Set HasWriteableRT correctly
- iris: Drop XXX about checking for swizzling
- iris: Move create and bind driver hooks to the end of iris_program.c
- iris: Make an IRIS_MAX_MIPLEVELS define
- iris: Simplify iris_get_depth_stencil_resources
- iris: Add missing depth cache flushes
- iris: Always emit at least one BLEND_STATE
- iris: Add iris_resource fields for aux surfaces
- iris: Fill out res->aux.possible_usages
- iris: Fill out SURFACE_STATE entries for each possible aux usage
- iris: create aux surface if needed
- iris: Initial import of resolve code
- iris: blorp using resolve hooks
- iris: add some draw resolve hooks
- iris: actually use the multiple surf states for aux modes
- iris: try to fix copyimage vs copybuffers
- iris: be sure to skip buffers in resolve code
- iris: resolve before transfer maps
- iris: pin the buffers
- iris: store modifier info in res
- iris: Make blit code use actual aux usages
- iris: consider framebuffer parameter for aux usages
- iris: Resolves for compute
- iris: disable aux for external things
- iris: some initial HiZ bits
- iris: don't use hiz for MSAA buffers
- iris: Set program key fields for MCS
- iris: make surface states for CCS_D too
- iris: do flush for buffers still
- iris: Allow disabling aux via INTEL_DEBUG options
- iris: Fix aux usage in render resolve code
- iris: Only resolve compute resources for compute shaders
- iris: Enable auxiliary buffer support

- iris: Enable -msse2 and -mstackrealign
- Revert "iris: Enable auxiliary buffer support"
- vulkan: Fix 32-bit build for the new overlay layer
- mesa: Fix RGBBuffers for renderbuffers with sized internal formats
- iris: Drop RGBX -> RGBA for storage image usages
- iris: Properly allow rendering to RGBX formats.
- i965: Implement threaded GL support.
- tgsi_to_nir: use sampler variables and derefs
- iris: Fix MOCS for blits and clears
- isl: Add a swizzle parameter to isl_buffer_fill_state()
- iris: Plumb through ISL_SWIZZLE_IDENTITY in buffer surface emitters
- iris: Defer uploading sampler state tables until draw time
- iris: Properly support alpha and luminance-alpha formats
- iris: Drop PIPE_CAP_BUFFER_SAMPLER_VIEW_RGBA_ONLY
- iris: Spruce up "are we using this engine?" checks for flushing
- iris: Export a copy_region helper that doesn't flush
- iris: Use copy_region and staging resources to avoid transfer stalls
- Revert MR 369 (Fix extract_i8 and extract_u8 for 64-bit integers)
- iris: Fix backface stencil write condition
- iris: Rework default tessellation level uploads
- iris: Fix TES gl_PatchVerticesIn handling.
- iris: Move depth/stencil flushes so they actually do something
- iris: Refactor depth/stencil buffer pinning into a helper.
- iris: Fix write enable in pinning of depth/stencil resources
- i965: Move some genX infrastructure to genX_boilerplate.h.
- i965: Rename ISP_DIS to INDIRECT_STATE_POINTERS_DISABLE.
- i965: Use genxml for emitting PIPE_CONTROL.
- i965: Reimplement all the PIPE_CONTROL rules.
- intel/fs: Fix opt_peephole_csel to not throw away saturates.
- iris: Don't mutate box in transfer map code
- iris: Don't flush the batch for unsynchronized mappings
- iris: Slightly better bounds on buffer sizes
- gallium: Add PIPE_BARRIER_UPDATE_BUFFER and UPDATE_TEXTURE bits.
- nvc0: Skip new update barrier bits
- nir: Record non-vector/scalar varyings as unmovable when compacting
- iris: Fix util_vma_heap_init size for IRIS_MEMZONE_SHADER

- iris: Skip input resolve handling if bindings haven't changed
- iris: Skip framebuffer resolve tracking if framebuffer isn't dirty
- iris: Skip resolves and flushes altogether if unnecessary
- iris: Fix batch chaining map_next increment.
- iris: Actually advertise some modifiers
- st/nir: Free the GLSL IR after linking.
- st/mesa: Fix blitting from GL_DEPTH_STENCIL to GL_STENCIL_INDEX
- iris: Fix blits with S8_UINT destination
- iris: Print the memzone name when allocating BOs with INTEL_DEBUG=buf
- iris: Save/restore MI_PREDICATE_RESULT, not MI_PREDICATE_DATA.
- iris: Silence unused variable warnings in release mode
- gallium/util: Add const to u_range_intersect
- iris: Actually pin the scratch BO.
- glsl: Set location on structure-split sampler uniform variables
- intel: Emit 3DSTATE_VF_STATISTICS dynamically
- iris: Actually mark blorp_copy_buffer destinations as written.
- iris: Preserve all PIPE_TRANSFER flags in xfer->usage
- iris: Fix FLUSH_EXPLICIT handling with staging buffers.
- iris: Make shader_perf_log print to stderr if INTEL_DEBUG=perf is set
- i965: Move program key debugging to the compiler.
- iris: Print the reason for shader recompiles.
- iris: Move iris_debug_recompile calls before uploading.
- iris: Change vendor and renderer strings
- iris: Add texture cache flushing hacks for blit and resource_copy_region
- iris: Be less aggressive at postdraw work skipping
- iris: Add mechanism for iris-specific driconf options
- iris: Enable the dual_color_blend_by_location driconf option.
- iris: Track bound and writable SSBOs
- Revert "glsl: Set location on structure-split sampler uniform variables"
- i965: Ignore uniform storage for samplers or images, use binding info
- i965: Tidy bogus indentation left by previous commit
- iris: Mark constants dirty on transfer unmap even if no flushes occur
- iris: Track bound constant buffers
- iris: Rework UBOs and SSBOs to use pipe_shader_buffer
- iris: Rework image views to store pipe_image_view.
- iris: Make a gl_shader_stage -> pipe_shader_stage helper function

- iris: Make memzone_for_address non-static
- iris: Replace buffer backing storage and rebind to update addresses.
- iris: Make a resource_is_busy() helper
- iris: Track valid data range and infer unsynchronized mappings.
- iris: Make some offset math helpers take a const isl_surf pointer
- iris: Fix DrawTransformFeedback math when there's a buffer offset
- iris: Prefer staging blits when destination supports CCS_E.
- iris: Actually put Mesa in GL_RENDERER string
- iris: Split iris_flush_and_dirty_for_history into two helpers.
- iris: Enable GL_AMD_depth_clamp_separate
- iris: Advertise EXT_texture_sRGB_R8 support
- iris: Some tidying for preemption support
- iris: Silence unused function warning
- iris: Fix zeroing of transform feedback offsets in strange cases.
- glsl/list: Add an exec_list_is_singular() helper.
- nir: Add a new nir_cf_list_is_empty_block() helper.
- intel/fs: Don't emit empty ELSE blocks.
- iris: Set XY Clipping correctly.
- iris: Only enable GL_AMD_depth_clamp_separate on Gen9+
- iris: Fix imageBuffer and PBO download.
- iris: Disable dual source blending when shader doesn't handle it
- iris: Resolve textures used by the program, not merely bound textures
- iris: Fix 4GB memory zone heap sizes.
- iris: leave the top 4Gb of the high heap VMA unused
- iris: Force VMA alignment to be a multiple of the page size.
- iris: Delete bucketing allocators
- i965: Fix BRW_MEMZONE_LOW_4G heap size.
- i965: Force VMA alignment to be a multiple of the page size.
- i965: leave the top 4Gb of the high heap VMA unused
- i965: Fix memory leaks in brw_upload_cs_work_groups_surface().
- iris: Use full ways for L3 cache setup on Icelake.
- egl/x11: calloc dri2_surf so it's properly zeroed

Kevin Strasser (1):

- egl/dri: Avoid out of bounds array access

Khaled Emara (1):

- freedreno: PIPE_CAP_SHADER_BUFFER_OFFSET_ALIGNMENT unreachable statement

Khem Raj (1):

- winsys/svga/drm: Include sys/types.h

Kishore Kadiyala (1):

- android: static link with libexpat with Android O+

Konstantin Kharlamov (1):

- mapi: work around GCC LTO dropping assembly-defined functions

Kristian Høgsberg (49):

- st/nir: Use src/ relative include path for autotools
- freedreno/a6xx: Emit blitter dst with OUT_RELOCW
- freedreno/a6xx: Use tiling for all resources
- freedreno/a6xx: regen headers
- freedreno/a6xx: Drop render condition check in blitter
- freedreno: Log number of draw for sysmem passes
- freedreno/a6xx: Use the right resource for separate stencil stride
- freedreno/a6xx: Combine emit_blit and fd6_blit
- freedreno: Consolidate u_blitter functions in freedreno_blitter.c
- freedreno: Don't tell the blitter what it can't do
- freedreno/a6xx: Move blit check so as to restore comment
- freedreno/a6xx: Support some depth/stencil blits on blitter
- freedreno/a6xx: Support y-inverted blits
- freedreno/a6xx: Add format argument to fd6_tex_swiz()
- freedreno/a6xx: Fall back to masked RGBA blits for depth/stencil
- freedreno/a6xx: Clean up mixed use of swap and swizzle for texture state
- freedreno/a6xx: Update headers
- freedreno/a6xx: Front facing needs UNK3 bit
- freedreno/a6xx: Fix point coord
- .mailmap: Add a few more alises for myself
- freedreno: Update headers
- freedreno/a6xx: Copy stencil as R8_UINT
- freedreno/a6xx: Support MSAA resolve blits on blitter
- freedreno/a6xx: Only output MRT control for used framebuffers
- freedreno/a6xx: Don't zero SO buffer addresses
- freedreno: Fix a couple of warnings
- turnip: Only get bo offset when we need to mmap
- freedreno: Use c_vis_args and no_override_init_args
- freedreno/a6xx: Remove extra parens

- freedreno/ir3: Track whether shader needs derivatives
- freedreno/ir3: Fix operand order for DSX/DSY
- st/glsl_to_nir: Calculate num_uniforms from NumParameterValues
- freedreno/ir3: Enable PIPE_CAP_PACKED_UNIFORMS
- freedreno/ir3: Push UBOs to constant file
- freedreno/ir3: Don't access beyond available regs
- freedreno/ir3: Add workaround for VS samgq
- freedreno/ir3: Mark ir3_context_error() as NORETURN
- freedreno/a2xx: Fix redundant if statement
- freedreno: Use enum values from matching enum
- freedreno/a6xx: Add helper for incrementing regid
- freedreno: Fix format string warning
- .gitignore: Remove autotool artifacts
- tgsi: Mark tgsi_strings_check() unused
- glsl_to_nir: Initialize debug variable
- nir_opcodes.py: Saturate to expression that doesn't overflow
- ralloc: Fully qualify non-virtual destructor call
- egl/dri2: Mark potentially unused 'display' variable with MAYBE_UNUSED
- gallium/auxiliary/vl: Fix a couple of warnings
- freedreno/drm: Quiet pointer to u64 conversion warning

Leo Liu (6):

- st/va: fix the incorrect max profiles report
- st/va/vp9: set max reference as default of VP9 reference number
- vl/dri3: remove the wait before getting back buffer
- radeon/vcn: add H.264 constrained baseline support
- radeon/vcn/vp9: search the render target from the whole list
- winsys/amdgpu: add VCN JPEG to no user fence group

Lepton Wu (2):

- virgl: close drm fd when destroying virgl screen.
- virgl: Set bind when creating temp resource.

Lionel Landwerlin (127):

- anv: assert that color attachment are valid
- radv: assert that colorAttachment is valid for CmdClearAttachment
- i965: scale factor changes should trigger recompile
- vulkan: Update the XML and headers to 1.1.101
- anv: implement VK_EXT_depth_clip_enable

- build: move imgui out of src/intel/tools to be reused
- imgui: bump copy
- imgui: make sure our copy of imgui doesn't clash with others in the same process
- vulkan: add an overlay layer
- intel: fix urb size for CFL GT1
- anv: add support for INTEL_DEBUG=bat
- Revert "anv: add support for INTEL_DEBUG=bat"
- intel/aub_viewer: printout 48bits addresses
- intel/aub_viewer: silence compiler warning
- intel/aub_viewer: silence more compiler warnings
- vulkan/overlay: fix missing installation of layer
- vulkan/overlay: fix includes
- imgui: update commit
- imgui: update memory editor
- vulkan/overlay: install layer binary in libdir
- intel/compiler: use correct swizzle for replacement
- vulkan/overlay: fix min/max computations
- vulkan/overlay: rework option parsing
- vulkan/overlay: add support for fps output in file
- anv: add support for INTEL_DEBUG=bat
- vulkan: update headers/registry to 1.1.102
- anv: update supported patch version
- radv: set num_components on vulkan_resource_index intrinsic
- vulkan/util: make header available from c++
- vulkan/util: generate instance/device dispatch tables
- vulkan/overlay: drop dependency on validation layer headers
- intel/decoders: add address space indicator to get BOs
- intel/decoders: handle decoding MI_BBS from ring
- intel/decoders: limit number of decoded batchbuffers
- intel/aub_read: reuse defines from gen_context
- intel/aub_write: split comment section from HW setup
- intel/aub_write: write header in init
- intel/aub_write: break execlist write in 2
- intel/aub_write: switch to use i915_drm engine classes
- intel/aub_write: log mmio writes
- intel/aub_write: store the physical page allocator in struct

- intel/aub_write: turn context images arrays into functions
- intel/aub_write: factorize context image/pphwsp/ring creation
- iris: fix decoder call
- iris: fix decode_get_bo callback
- intel/error2aub: build a list of BOs before writing them
- intel/error2aub: identify buffers by engine
- intel/error2aub: strenghten batchbuffer identifier marker
- intel/error2aub: parse other buffer types
- intel/error2aub: annotate buffer with their address space
- intel/error2aub: store engine last ring buffer head/tail pointers
- intel/error2aub: write GGTT buffers into the aub file
- intel/error2aub: add a verbose option
- intel/error2aub: deal with GuC log buffer
- intel/error2aub: support older style engine names
- vulkan: factor out wsi dependencies
- anv: implement VK_EXT_pipeline_creation_feedback
- vulkan/overlay: properly register layer object with loader
- vulkan/overlay: silence validation layer warnings
- vulkan/overlay: check return value of swapchain get images
- vulkan/overlay: improve error reporting
- i965: perf: sklgt2: update a priority for register programming
- i965: perf: sklgt2: update compute metrics config
- i965: perf: sklgt2: update memory write config
- i965: perf: add PMA stall metrics
- i965: perf: chv: fixup counters names
- i965: perf: hsw: drop register programming not needed on HSW
- i965: perf: sklgt2: drop programming of an unused NOA register
- i965: perf: add Icelake metrics
- i965: perf: enable Icelake metrics
- i965: perf: add ring busyness metric for cfl gt2
- i965: perf: update render basic configs for big core gen9/gen10
- anv: implement VK_KHR_swapchain revision 70
- intel: add dependency on genxml generated files
- genxml: add a sorting script
- genxml: sort xml files using new script
- anv: don't use default pipeline cache for hits for VK_EXT_pipeline_creation_feedback

- anv: store heap address bounds when initializing physical device
- anv: leave the top 4Gb of the high heap VMA unused
- i965: store device revision in gen_device_info
- i965: extract performance query metrics
- i965: move mdapi data structure to intel/perf
- i965: move OA accumulation code to intel/perf
- i965: move brw_timebase_scale to device info
- i965: move mdapi result data format to intel/perf
- i965: move mdapi guid into intel/perf
- intel/perf: stub gen10/11 missing definitions
- i965: perf: add mdapi pipeline statistics queries on gen10/11
- intel/perf: drop counter size field
- intel/perf: constify accumlator parameter
- iris: implement WaEnableStateCacheRedirectToCS
- i965: implement WaEnableStateCacheRedirectToCS
- anv: implement WaEnableStateCacheRedirectToCS
- anv: fix uninitialized pthread cond clock domain
- intel/devinfo: fix missing num_thread_per_eu on ICL
- intel/devinfo: add basic sanity tests on device database
- anv: limit URB reconfigurations when using blorp
- intel: workaround VS fixed function issue on Gen9 GT1 parts
- anv: fix argument name for vkCmdEndQuery
- i965: fix icelake performance query enabling
- Revert "anv: limit URB reconfigurations when using blorp"
- vulkan/util: generate a helper function to return pNext struct sizes
- vulkan/overlay: update help printout
- vulkan/overlay: record stats in command buffers and accumulate on exec/submit
- vulkan/overlay: add pipeline statistic & timestamps support
- vulkan/overlay: add no display option
- vulkan/overlay: add a margin to the size of the window
- vulkan/overlay: record all select metrics into output file
- vulkan/overlay: add a frame counter option
- vulkan/overlay: make overlay size configurable
- vulkan/overlay: make overriden functions static
- vulkan/overlay: add TODO list
- anv: fix crash when application does not provide push constants

- anv: rework queries writes to ensure ordering memory writes
- anv: fix use after free
- anv: Use corresponding type from the vector allocation
- vulkan/overlay: keep allocating draw data until it can be reused
- nir: fix lower_non_uniform_access pass
- vulkan/overlay-layer: fix cast errors
- vulkan/overlay: fix truncating error on 32bit platforms
- nir: lower_non_uniform_access: iterate over instructions safely
- vulkan/overlay: fix timestamp query emission with no pipeline stats
- vulkan: fix build dependency issue with generated files
- anv: fix apply_pipeline_layout pass for arrays of YCbCr descriptors
- nir/lower_non_uniform: safely iterate over blocks
- intel/perf: fix EuThreadsCount value in performance equations
- intel/perf: improve dynamic loading config detection

Lubomir Rintel (3):

- kmsro: Extend to include armada-drm
- gallivm: guess CPU features also on ARM
- gallivm: disable NEON instructions if they are not supported

Lucas Stach (3):

- etnaviv: don't flush own context when updating resource use
- etnaviv: flush all pending contexts when accessing a resource with the CPU
- etnaviv: only try to construct scanout resource when on KMS winsys

Marek Olšák (121):

- radeonsi: enable dithered alpha-to-coverage for better quality
- radeonsi: merge & rename texture BO metadata functions
- radeonsi: unify error paths in si_texture_create_object
- winsys/amdgpu: remove amdgpu_drm.h definitions
- r600: add -Wstrict-overflow=0 to meson to silence the warning
- radeonsi: fix a comment typo in si_fine_fence_set
- gallium: allow more PIPE_RESOURCE_ driver flags
- meson: drop the xcb-xrandr version requirement
- radeonsi: handle render_condition_enable in si_compute_clear_render_target
- radeonsi: fix crashing performance counters (division by zero)
- radeonsi: initialize textures using DCC to black when possible
- radeonsi: clear allocator_zeroed_memory with SDMA
- radeonsi: make allocator_zeroed_memory unmappable and use bigger buffers

- radeonsi: don't leak an index buffer if draw_vbo fails
- radeonsi: use local ws variable in si_need_dma_space
- gallium/u_threaded: fix EXPLICIT_FLUSH for flush offsets > 0
- radeonsi: fix EXPLICIT_FLUSH for flush offsets > 0
- winsys/amdgpu: don't drop manually added fence dependencies
- winsys/amdgpu: unify fence list code
- winsys/amdgpu: use a separate fence list for syncobjs
- winsys/amdgpu: remove occurence of INDIRECT_BUFFER_CONST
- winsys/amdgpu: clean up IB buffer size computation
- winsys/amdgpu: cs_check_space sets the minimum IB size for future IBs
- radeonsi: add AMD_DEBUG env var as an alternative to R600_DEBUG
- radeonsi: use MEM instead of MEM_GRBM in COPY_DATA.DST_SEL
- radeonsi: add driconf option radeonsi_enable_nir
- radeonsi: always enable NIR for Civilization 6 to fix corruption
- driconf: add Civ6Sub executable for Civilization 6
- st/mesa: always unmap the uploader in st_atom_array.c
- gallium/u_threaded: always unmap const_uploader
- gallium/u_upload_mgr: allow use of FLUSH_EXPLICIT with persistent mappings
- radeonsi: use SDMA for uploading data through const_uploader
- tgsi: don't set tgsi_info::uses_bindless_images for constbufs and hw atomics
- radeonsi: always use compute rings for clover on CI and newer (v2)
- gallium/u_tests: use a compute-only context to test GCN compute ring
- gallium: add pipe_grid_info::last_block
- omx: clean up enc_LoadImage_common
- omx: add a compute path in enc_LoadImage_common
- radeonsi: fix assertion failure by using the correct type
- mesa: implement ARB/KHR_parallel_shader_compile
- gallium: implement ARB/KHR_parallel_shader_compile
- util/queue: move thread creation into a separate function
- util/queue: add ability to kill a subset of threads
- util/queue: hold a lock when reading num_threads in util_queue_finish
- util/queue: add util_queue_adjust_num_threads
- radeonsi: implement ARB/KHR_parallel_shader_compile callbacks
- radeonsi: don't use PFP_SYNC_ME with compute-only contexts
- docs/relnotes: document parallel_shader_compile changes in 19.1.0, not 19.0.0
- amd/addrlib: fix uninitialized values for Addr2ComputeDccAddrFromCoord

- radeonsi/gfx9: add support for PIPE_ALIGNED=0
- radeonsi: add ability to bind images as image buffers
- radeonsi: add support for displayable DCC for 1 RB chips
- radeonsi: add support for displayable DCC for multi-RB chips
- radeonsi: enable displayable DCC on Ravens
- gallium: add writable_bitmask parameter into set_shader_buffers
- glsl: remember which SSBOs are not read-only and pass it to gallium
- radeonsi: set exact shader buffer read/write usage in CS
- tegra: fix the build after the set_shader_buffers change
- radeonsi: fix a crash when unbinding sampler states
- glsl: fix shader_storage_blocks_write_access for SSBO block arrays
- Revert "glsl: fix shader_storage_blocks_write_access for SSBO block arrays"
- glsl: allow the #extension directive within code blocks for the dri option
- mesa: don't overwrite existing shader files with MESA_SHADER_CAPTURE_PATH
- radeonsi: set AC_FUNC_ATTR_READNONE for image opcodes where it was missing
- ac: use the common helper ac_apply_fmask_to_sample
- ac: fix incorrect bindless atomic code in visit_image_atomic
- radeonsi: enable GL_EXT_shader_image_load_formatted
- nir: optimize gl_SampleMaskIn to gl_HelperInvocation for radeonsi when possible
- winsys/amdgpu: don't set GTT with GDS & OA placements on APUs
- radeonsi/gfx9: use the correct condition for the DPBB + QUANT_MODE workaround
- radeonsi: use CP DMA for the null const buffer clear on CIK
- tgsi/scan: add uses_drawid
- ac: add radeon_info::marketing_name, replacing the winsys callback
- ac: add radeon_info::is_pro_graphics
- ac: add ac_get_i1_sgpr_mask
- ac: add REWIND and GDS registers to register headers
- winsys/amdgpu: make IBs writable and expose their address
- winsys/amdgpu: reorder chunks, make BO_HANDLES first, IB and FENCE last
- winsys/amdgpu: enable chaining for compute IBs
- winsys/amdgpu: clean up and remove nonsensical assertion
- radeonsi: add si_cp_copy_data
- radeonsi: add helper si_get_minimum_num_gfx_cs_dwords
- radeonsi: delay adding BOs at the beginning of IBs until the first draw
- gallium: document conservative rasterization flags
- st/dri: simplify throttling code

- gallium: replace DRM_CONF_THROTTLE with PIPE_CAP_MAX_FRAMES_IN_FLIGHT
- gallium: replace DRM_CONF_SHARE_FD with PIPE_CAP_DMABUF
- gallium: replace drm_driver_descriptor::configuration with driconf_xml
- gallium: set PIPE_CAP_MAX_FRAMES_IN_FLIGHT to 2 for all drivers
- gallium: add PIPE_CAP_PREFER_COMPUTE_BLIT_FOR_MULTIMEDIA
- util: fix a compile failure in u_compute.c on windows
- mesa: enable glGet for EXT_gpu_shader4
- glsl: add 'unsigned int' type for EXT_GPU_shader4
- glsl: apply some 1.30 and other rules to EXT_gpu_shader4 as well
- glsl: add builtin variables for EXT_gpu_shader4
- glsl: add arithmetic builtin functions for EXT_gpu_shader4
- glsl: add texture builtin functions for EXT_gpu_shader4
- glsl: allow "varying out" for fragment shader outputs with EXT_gpu_shader4
- mesa: expose EXT_texture_buffer_object
- mesa: only allow EXT_gpu_shader4 in the compatibility profile
- st/mesa: expose EXT_gpu_shader4 if GLSL 1.40 is supported
- glsl: handle interactions between EXT_gpu_shader4 and texture extensions
- radeonsi: add BOs after need_cs_space
- radeonsi/gfx9: set that window_rectangles always roll the context
- radeonsi/gfx9: rework the gfx9 scissor bug workaround (v2)
- radeonsi: remove dirty slot masks from scissor and viewport states
- glsl: fix shader_storage_blocks_write_access for SSBO block arrays (v2)
- radeonsi: don't ignore PIPE_FLUSH_ASYNC
- mesa: rework error handling in glDrawBuffers
- mesa: fix pbuffers because internally they are front buffers
- st/mesa: don't flush the front buffer if it's a pbuffer
- radeonsi: use new atomic LLVM helpers
- radeonsi: set sampler state and view functions for compute-only contexts
- st/dri: decrease input lag by syncing sooner in SwapBuffers
- glsl: fix and clean up NV_compute_shader_derivatives support
- st/mesa: fix 2 crashes in st_tgsi_lower_yuv
- radeonsi: remove old_va parameter from si_rebind_buffer by remembering offsets
- radeonsi: update buffer descriptors in all contexts after buffer invalidation
- radeonsi: fix a regression in si_rebind_buffer
- u_blitter: don't fail mipmap generation for depth formats containing stencil
- ac: fix a typo in ac_build_wg_scan_bottom

Mario Kleiner (1):

- drirc: Add sddm-greeter to adaptive_sync blacklist.

Mark Janes (5):

- mesa: properly report the length of truncated log messages
- mesa: rename logging functions to reflect that they format strings
- mesa: add logging function for formatted string
- intel/common: move gen_debug to intel/dev
- intel/tools: Remove redundant definitions of INTEL_DEBUG

Mateusz Krzak (2):

- panfrost: cast bo_handles pointer to uintptr_t first
- panfrost: use os_mmap and os_munmap

Mathias Fröhlich (22):

- st/mesa: Reduce array updates due to current changes.
- mesa: Track buffer object use also for VAO usage.
- st/mesa: Invalidate the gallium array atom only if needed.
- mesa: Implement helper functions to map and unmap a VAO.
- mesa: Factor out _mesa_array_element.
- mesa: Use _mesa_array_element in dlist save.
- mesa: Replace _ae_{,un}map_vbos with _mesa_vao_{,un}map_arrays
- mesa: Remove _ae_{,un}map_vbos and dependencies.
- mesa: Use mapping tools in debug prints.
- vbo: Fix basevertex handling in display list compiles.
- vbo: Fix GL_PRIMITIVE_RESTART_FIXED_INDEX in display list compiles.
- mesa: Add assert to _mesa_primitive_restart_index.
- mesa: Factor out index function that will have multiple use.
- mesa: Use glVertexAttrib*NV functions for fixed function attribs.
- mesa: Implement _mesa_array_element by walking enabled arrays.
- mesa: Rip out now unused gl_context::aelt_context.
- mesa: Remove the now unused _NEW_ARRAY state change flag.
- mesa: Constify static const array in api_arrayelt.c
- mesa: Remove the _glapi_table argument from _mesa_array_element.
- mesa: Set CurrentSavePrimitive in vbo_save_NotifyBegin.
- mesa: Correct the is_vertex_position decision for dlists.
- mesa: Leave aliasing of vertex and generic0 attribute to the dlist code.

Matt Turner (7):

- intel/compiler/test: Set devinfo->gen = 7

- intel/compiler: Avoid propagating inequality cmods if types are different
- intel/compiler/test: Add unit test for mismatched signedness comparison
- intel/compiler: Add commas on final values of compaction table arrays
- intel/compiler: Use SIMD16 instructions in fs saturate prop unit test
- intel/compiler: Add unit tests for sat prop for different exec sizes
- intel/compiler: Improve fix_3src_operand()

Matthias Lorenz (1):

- vulkan/overlay: Add fps counter

Mauro Rossi (6):

- android: intel/isl: remove redundant building rules
- android: anv: fix generated files depedencies (v2)
- android: anv: fix libexpat shared dependency
- android: nouveau: add support for nir
- android: fix LLVM version string related building errors
- draw: fix building error in draw_gs_init()

Maya Rashish (1):

- configure: fix test portability

Michel Dänzer (19):

- loader/dri3: Use strlen instead of sizeof for creating VRR property atom
- gitlab-ci: Re-use docker image from the main repo in forked repos
- gitlab-ci: List some longer-running jobs before others of the same stage
- gitlab-ci: Use 8 CPU cores in autotools job
- gitlab-ci: Make sure clang job actually uses ccache
- gitlab-ci: Only pull/push cache contents in build+test stage jobs
- gitlab-ci: Automatically retry jobs after runner system failure
- gitlab-ci: Run CI pipeline for all branches in the main repository
- gitlab-ci: Use Debian stretch instead of Ubuntu bionic
- gitlab-ci: Use HTTPS for APT repositories
- gitlab-ci: Use Debian packages instead of pip ones for meson and scons
- gitlab-ci: Install most packages from Debian buster
- gitlab-ci: Remove unneded (stuff from) APT command lines
- gitlab-ci: Remove unused Debian packages from Docker image
- gitlab-ci: Use clang 8 instead of 7
- gitlab-ci: Drop unused clang 5/6 packages
- gitlab-ci: Do not use subshells for compiling dependencies
- gitlab-ci: Use LLVM 3.4 from Debian jessie for scons-llvm job

- gitlab-ci: Use meson buildtype debug instead of default debugoptimized

Mike Blumenkrantz (6):

- iris: support INTEL_NO_HW environment variable
- gallium: add pipe cap for inner_coverage conservative raster mode
- st/mesa: indicate intel extension support for inner_coverage based on cap
- iris: add support for INTEL_conservative_rasterization
- iris: add preemption support on gen9
- iris: enable preemption support for gen10

Nanley Chery (3):

- i965: Rename intel_mipmap_tree::r8stencil_* -> ::shadow_*
- anv: Fix some depth buffer sampling cases on ICL+
- anv/cmd_buffer: Initalize the clear color struct for CNL+

Nataraj Deshpande (1):

- anv: Fix check for isl_fmt in assert

Neha Bhende (2):

- st/mesa: Fix topogun-1.06-orc-84k-resize.trace crash
- draw: fix memory leak introduced 7720ce32a

Nicolai Hähnle (9):

- amd/surface: provide firstMipIdInTail for metadata surface calculations
- radeonsi: add si_debug_options for convenient adding/removing of options
- util/u_log: flush auto loggers before starting a new page
- ddebug: set thread name
- ddebug: log calls to pipe->flush
- ddebug: dump driver state into a separate file
- ddebug: expose some helper functions as non-inline
- radeonsi: add radeonsi_aux_debug option for aux context debug dumps
- radeonsi: add radeonsi_sync_compile option

Oscar Blumberg (3):

- intel/fs: Fix memory corruption when compiling a CS
- radeonsi: Fix guardband computation for large render targets
- glsl: Fix function return typechecking

Patrick Lerda (1):

- lima/ppir: fix pointer referenced after a free

Patrick Rudolph (1):

- d3dadapter9: Support software renderer on any DRI device

Philipp Zabel (1):

- etnaviv: fill missing offset in etna_resource_get_handle

Pierre Moreau (12):

- include/CL: Update to the latest OpenCL 2.2 headers
- clover: Avoid warnings from new OpenCL headers
- clover: Remove the TGSI backend as unused
- clover: Add an helper for checking if an IR is supported
- clover/api: Rework the validation of devices for building
- clover/api: Fail if trying to build a non-executable binary
- clover: Disallow creating libraries from other libraries
- clover: Validate program and library linking options
- clover: Move device extensions definitions to core/device.cpp
- clover: Move platform extensions definitions to clover/platform.cpp
- clover: Only use devices supporting IR_NATIVE
- clover: Fix indentation issues

Pierre-Eric Pelloux-Prayer (1):

- radeonsi: init sctx->dma_copy before using it

Plamena Manolova (3):

- i965: Disable ARB_fragment_shader_interlock for platforms prior to GEN9
- isl: Set ClearColorConversionEnable.
- i965: Re-enable fast color clears for GEN11.

Qiang Yu (9):

- u_math: add ushort_to_float/float_to_ushort
- u_dynarray: add util_dynarray_grow_cap
- gallium/u_vbuf: export u_vbuf_get_minmax_index
- drm-uapi: add lima_drm.h
- gallium: add lima driver
- lima/gpir: fix compile fail when two slot node
- lima/gpir: fix alu check miss last store slot
- lima: fix lima_blit with non-zero level source resource
- lima: fix render to non-zero level texture

Rafael Antognolli (45):

- iris: Store internal_format when getting resource from handle.
- iris: Skip msaa16 on gen < 9.
- iris: Flush before hiz_exec.
- iris: Pin HiZ buffers when rendering.
- iris: Avoid leaking if we fail to allocate the aux buffer.

- iris/clear: Pass on render_condition_enabled.
- iris: Skip resolve if there's no context.
- iris: Flag ALL_DIRTY_BINDINGS on aux state change.
- iris: Add resolve on iris_flush_resource.
- iris: Convert RGBX to RGBA always.
- iris: Enable auxiliary buffer support again
- iris: Enable HiZ for multisampled depth surfaces.
- iris: Make intel_hiz_exec public.
- iris: Allocate buffer space for the fast clear color.
- iris: Use the clear depth when emitting 3DSTATE_CLEAR_PARAMS.
- iris: Fast clear depth buffers.
- iris: Add helper to convert fast clear color.
- iris: Add function to update clear color in surface state.
- iris: Bring back check for srgb and fast clear color.
- intel/isl: Add isl_format_has_color_component() function.
- intel/blorp: Make swizzle_color_value public.
- iris: Implement fast clear color.
- iris: Add iris_resolve_conditional_render().
- iris: Stall on the CPU and resolve predication during fast clears.
- iris: Track fast clear color.
- iris: Let blorp update the clear color for us.
- i965/blorp: Remove unused parameter from blorp_surf_for_miptree.
- iris: Only update clear color for gens 8 and 9.
- iris/gen8: Re-emit the SURFACE_STATE if the clear color changed.
- iris: Manually apply fast clear color channel overrides.
- iris: Do not allocate clear_color_bo for gen8.
- iris: Add aux.sampler_usages.
- iris: Enable fast clears on gen8.
- intel/fs: Only propagate saturation if exec_size is the same.
- intel/fs: Move the scalar-region conversion to the generator.
- intel/fs: Add a lowering pass for linear interpolation.
- intel/fs: Remove fs_generator::generate_linterp from gen11+.
- intel/isl: Resize clear color buffer to full cacheline
- intel/genxml: Update MI_ATOMIC genxml definition.
- intel/blorp: Make blorp update the clear color in gen11.
- iris: Do not advertise multisampled image load/store.

- iris: Support sRGB fast clears even if the colorspaces differ.
- iris: Use the linear version of the surface format during fast clears.
- iris: Update the surface state clear color address when available.
- iris: Enable fast clear colors on gen11.

Ray Zhang (1):

- glx: fix shared memory leak in X11

Rhys Kidd (1):

- iris: Fix assertion in iris_resource_from_handle() tiling usage

Rhys Perry (28):

- nvc0: add compute invocation counter
- radv: bitcast 16-bit outputs to integers
- radv: ensure export arguments are always float
- ac/nir: implement 8-bit nir_load_const_instr
- ac/nir: fix 64-bit nir_op_f2f16_rtz
- ac/nir: make ac_build_clamp work on all bit sizes
- ac/nir: make ac_build_isign work on all bit sizes
- ac/nir: make ac_build_fdiv support 16-bit floats
- ac/nir: implement half-float nir_op_frcp
- ac/nir: implement half-float nir_op_frsq
- ac/nir: implement half-float nir_op_ldexp
- ac/nir: fix 16-bit ssbo stores
- ac/nir: implement 8-bit push constant, ssbo and ubo loads
- ac/nir: implement 8-bit ssbo stores
- ac/nir: add 8-bit types to glsl_base_to_llvm_type
- ac/nir: implement 8-bit conversions
- radv: enable VK_KHR_8bit_storage
- ac/nir: implement 16-bit pack/unpack opcodes
- radv: lower 16-bit flrp
- ac: add 16-bit support to ac_build_ddxy()
- nir,ac/nir: fix cube_face_coord
- gallium: add support for formatted image loads
- mesa, glsl: add support for EXT_shader_image_load_formatted
- st/mesa: add support for EXT_shader_image_load_formatted
- vc4: fix build
- ac,ac/nir: use a better sync scope for shared atomics
- radv: fix set_output_usage_mask() with composite and 64-bit types

- ac/nir: mark some texture intrinsics as convergent

Rob Clark (135):

- freedreno: fix release tarball
- freedreno: more fixing release tarball
- freedreno/a6xx: small compiler warning fix
- freedreno/ir3: fix varying packing vs. tex sharp edge
- freedreno/a6xx: move stream-out emit to helper
- freedreno/a6xx: clean up some open-coded bits
- freedreno/ir3: split out image helpers
- freedreno/ir3: split out a4xx+ instructions
- freedreno/ir3: fix ncomp for _store_image() src
- freedreno/ir3: add image/ssbo <-> ibo/tex mapping
- freedreno/ir3: add a6xx instruction encoding
- freedreno/ir3: add a6xx+ SSBO/image support
- freedreno/ir3: HIGH reg w/a for a6xx
- freedreno/a6xx: border-color offset helper
- freedreno/a6xx: image/ssbo state emit
- freedreno/a6xx: compute support
- freedreno/a6xx: cache flush harder
- freedreno/a6xx: fix helper_invocation (sampler mask/id)
- freedreno/ir3: handle quirky atomic dst for a6xx
- freedreno/ir3: fix legalize for vecN inputs
- freedreno/ir3: fix crash in compile fail case
- freedreno/a6xx: 3d and cube image fixes
- freedreno: fix crash w/ masked non-SSA dst
- freedreno/ir3: rename put_dst()
- freedreno/ir3/a6xx: fix load_ssbo barrier type.
- freedreno/ir3: sync instr/disasm and add ldib encoding
- freedreno/ir3/a6xx: use ldib for ssbo reads
- freedreno/a6xx: samplerBuffer fixes
- freedreno/a6xx: enable tiled images
- freedreno: fix race condition
- freedreno/ir3: don't hardcode wrmask
- freedreno/a6xx: fix border-color offset
- freedreno/a6xx: cube image fix
- freedreno/a6xx: fix hangs with large shaders

- freedreno/ir3: use nopN encoding when possible
- freedreno/a6xx: fix ssbo alignment
- freedreno/ir3/a6xx: fix non-ssa atomic dst
- freedreno/a6xx: fix DRAW_IDX_INDIRECT max_indicies
- freedreno/a6xx: vertex_id is not _zero_based
- freedreno/ir3/a6xx: fix atomic shader outputs
- freedreno/ir3: gsampler2DMSArray fixes
- freedreno/ir3: include nopN in expanded instruction count
- freedreno/ir3: add Sethi–Ullman numbering pass
- freedreno/ir3: track register pressure in sched
- freedreno: fix ir3_cmdline build
- freedreno/a6xx: remove astc_srgb workaround
- freedreno/a6xx: refactor fd6_tex_swiz()
- freedreno/a6xx: fix border-color swizzles
- freedreno/a6xx: perfcntrs
- freedreno/ir3: fix ir3_cmdline harder
- freedreno/ir3: turn on [iu]mul_high
- freedreno/a6xx: more bcolor fixes
- freedreno/ir3/cp: fix ldib bug
- freedreno/ir3/a6xx: fix ssbo comp_swap
- freedreno/ir3 better cat6 encoding detection
- freedreno/ir3/ra: fix half-class conflicts
- freedreno/ir3: fix sam.s2en decoding
- freedreno/ir3: fix sam.s2en encoding
- freedreno/ir3: fix regmask for merged regs
- nir: move gls_type_get_{sampler,image}_count()
- freedreno/ir3: find # of samplers from uniform vars
- freedreno/ir3: enable indirect tex/samp (sam.s2en)
- freedreno/ir3: optimize sam.s2en to sam
- freedreno/ir3: additional lowering
- freedreno/ir3: fix bit_count
- freedreno/ir3: dynamic UBO indexing vs 64b pointers
- freedreno/ir3: rename has_kill to no_earlyz
- freedreno/ir3: disable early-z for SSBO/image writes
- gallium: add PIPE_CAP_ESSL_FEATURE_LEVEL
- mesa/st: use ESSL cap top enable gpu_shader5

- freedreno: add ESSL cap
- docs: update freedreno status
- freedreno/a6xx: small cleanup
- freedreno/ir3: sched fix
- freedreno/ir3: reads/writes to unrelated arrays are not dependent
- freedreno/ir3: align const size to vec4
- nir: print var name for load_interpolated_input too
- nir: add lower_all_io_to_elements
- freedreno/ir3: re-indent comment
- freedreno/ir3: rework varying packing
- freedreno/ir3: add pass to move varying loads
- freedreno/ir3: convert to "new style" frag inputs
- gallium/docs: clarify set_sampler_views (v2)
- iris: fix set_sampler_view
- freedreno/ir3: fix const assert
- freedreno/drm: update for robustness
- freedreno: add robustness support
- compiler: rename SYSTEM_VALUE_VARYING_COORD
- freedreno/ir3: fix rgetpos decoding
- freedreno/ir3: more emit-cat5 fixes
- freedreno/ir3: cleanup instruction builder macros
- freedreno: update generated headers
- freedreno/ir3: lower load_barycentric_at_sample
- freedreno/ir3: lower load_barycentric_at_offset
- freedreno/ir3: remove bogus assert
- freedreno/ir3: rename frag_vcoord -> ij_pixel
- freedreno/a6xx: add VALIDREG/CONDREG helper macros
- freedreno/ir3: fix load_interpolated_input slot
- freedreno: wire up core sample-shading support
- freedreno/ir3: sample-shading support
- freedreno/a6xx: sample-shading support
- docs/features: update GL too
- freedreno/ir3: switch fragcoord to sysval
- freedreno/a6xx: small texture emit cleanup
- freedreno/a6xx: pre-bake UBWC flags in texture-view
- freedreno/ir3: fixes for half reg in/out

- freedreno/ir3: fix shader variants vs UBO analysis
- freedreno/ir3: fix lowered ubo region alignment
- freedreno/ir3: add IR3_SHADER_DEBUG flag to disable ubo lowering
- freedreno/ir3: add some ubo range related asserts
- nir: rework tex instruction printing
- nir: fix lower_wpos_ytransform in load_frag_coord case
- nir: add pass to lower fb reads
- freedreno/drm: expose GMEM_BASE address
- freedreno/ir3: fb read support
- freedreno/a6xx: KHR_blend_equation_advanced support
- freedreno/a6xx: smaller hammer for fb barrier
- docs: mark KHR_blend_equation_advanced done on a6xx
- nir: fix nir tex print harder
- freedreno/ir3: remove assert
- freedreno/a6xx: OUT_RELOC vs OUT_RELOCW fixes
- freedreno: update generated headers
- freedreno/a6xx: UBWC fixes
- freedreno/a6xx: UBWC support for images
- freedreno: mark imported resources as valid
- freedreno/a6xx: buffer resources cannot be compressed
- freedreno: move UBWC color offset to fd_resource_offset()
- freedreno: add ubwc_enabled helper
- freedreno/a6xx: deduplicate a few lines
- freedreno: remove unused forward struct declaration
- freedreno/ir3: fix rasterflat/glxgears
- freedreno/ir3: set more barrier bits
- freedreno/a6xx: fix GPU crash on small render targets
- freedreno/a6xx: fix issues with gallium HUD
- freedreno/a6xx: fix hangs with newer sqe fw

Rob Herring (2):

- kmsro: Add lima renderonly support
- kmsro: Add platform support for exynos and sun4i

Rodrigo Vivi (1):

- intel: Add more PCI Device IDs for Coffee Lake and Ice Lake.

Roland Scheidegger (2):

- gallivm: fix bogus assert in get_indirect_index

- gallivm: fix saturated signed add / sub with llvm 9

Romain Failliot (1):

- docs: changed "Done" to "DONE" in features.txt

Ross Burton (1):

- Revert "meson: drop GLESv1 .so version back to 1.0.0"

Ryan Houdek (1):

- panfrost: Adds Bifrost shader disassembler utility

Sagar Ghuge (10):

- iris: Don't allocate a BO per query object
- nir/glsl: Add another way of doing lower_imul64 for gen8+
- glsl: [u/i]mulExtended optimization for GLSL
- nir/algebraic: Optimize low 32 bit extraction
- spirv: Allow [i/u]mulExtended to use new nir opcode
- iris: Refactor code to share 3DSTATE_URB_* packet
- iris: Track last VS URB entry size
- iris: Flag fewer dirty bits in BLORP
- intel/fs: Remove unused condition from opt_algebraic case
- intel/compiler: Fix assertions in brw_alu3

Samuel Iglesias Gonsálvez (4):

- isl: remove the cache line size alignment requirement
- isl: the display engine requires 64B alignment for linear surfaces
- radv: don't overwrite results in VkGetQueryPoolResults() when queries are not available
- radv: write availability status vkGetQueryPoolResults() when the data is not available

Samuel Pitoiset (147):

- radv/winsys: fix hash when adding internal buffers
- radv: fix build
- radv: bail out when no image transitions will be performed
- radv: remove unused radv_render_pass_attachment::view_mask
- radv: remove useless MAYBE_UNUSED in CmdBeginRenderPass()
- radv: add radv_cmd_buffer_begin_subpass() helper
- radv: move subpass image transitions to radv_cmd_buffer_begin_subpass()
- radv: store the list of attachments for every subpass
- radv: use the new attachments array when starting subpasses
- radv: determine the last subpass id for every attachments
- radv: handle final layouts at end of every subpass and render pass
- radv: move some render pass things to radv_render_pass_compile()

- radv: add radv_render_pass_add_subpass_dep() helper
- radv: track if subpasses have color attachments
- radv: handle subpass dependencies correctly
- radv: accumulate all ingoing external dependencies to the first subpass
- radv: execute external subpass barriers after ending subpasses
- radv: drop useless checks when resolving subpass color attachments
- radv: do not set preserveAttachments for internal render passes
- radv: don't flush src stages when dstStageMask == BOTTOM_OF_PIPE
- radv: fix compiler issues with GCC 9
- radv: gather more info about push constants
- radv: gather if shaders load dynamic offsets separately
- radv: keep track of the number of remaining user SGPRs
- radv: add support for push constants inlining when possible
- radv: fix using LOAD_CONTEXT_REG with old GFX ME firmwares on GFX8
- radv/winsys: fix BO list creation when RADV_DEBUG=allbos is set
- radv: always export gl_SampleMask when the fragment shader uses it
- ac: make use of ac_build_expand_to_vec4() in visit_image_store()
- radv: use MAX_{VBS,VERTEX_ATTRIBS} when defining max vertex input limits
- radv: store vertex attribute formats as pipeline keys
- radv: reduce the number of loaded channels for vertex input fetches
- radv: fix radv_fixup_vertex_input_fetches()
- radv: fix invalid element type when filling vertex input default values
- ac: add ac_build_llvm8_tbuffer_load() helper
- ac: use new LLVM 8 intrinsic when loading 16-bit values
- radv: write the alpha channel of MRT0 when alpha coverage is enabled
- radv: remove unused variable in gather_push_constant_info()
- radv: fix writing the alpha channel of MRT0 when alpha coverage is enabled
- radv: fix clearing attachments in secondary command buffers
- radv: fix out-of-bounds access when copying descriptors BO list
- radv: don't copy buffer descriptors list for samplers
- rav: use 32_AR instead of 32_ABGR when alpha coverage is required
- radv: allocate enough space in cmdbuf when starting a subpass
- radv: properly align the fence and EOP bug VA on GFX9
- radv: enable lower_mul_2x32_64
- Revert "radv: execute external subpass barriers after ending subpasses"
- radv: fix pointSizeRange limits

- radv: set the maximum number of IBs per submit to 192
- ac: rework typed buffers loads for LLVM 7
- radv: store more vertex attribute infos as pipeline keys
- radv: use typed buffer loads for vertex input fetches
- ac: add ac_build_{struct,raw}_tbuffer_load() helpers
- ac: use the raw tbuffer version for 16-bit SSBO loads
- radv: always initialize HTILE when the src layout is UNDEFINED
- radv: always load 3 channels for formats that need to be shuffled
- ac: use llvm.amdgcn.fract intrinsic for nir_op_ffract
- radv: fix binding transform feedback buffers
- ac: make use of ac_get_store_intr_attribs() where possible
- ac/nir: set attrib flags for SSBO and image store operations
- ac: add ac_build_buffer_store_format() helper
- ac/nir: remove one useless check in visit_store_ssbo()
- ac/nir: use new LLVM 8 intrinsics for SSBO atomic operations
- ac/nir: use ac_build_buffer_load() for SSBO load operations
- ac/nir: use ac_build_buffer_store_dword() for SSBO store operations
- ac: use new LLVM 8 intrinsics in ac_build_buffer_load()
- ac: add ac_build_{struct,raw}_tbuffer_store() helpers
- ac: use new LLVM 8 intrinsic when storing 16-bit values
- ac: use new LLVM 8 intrinsics in ac_build_buffer_store_dword()
- ac: add various int8 definitions
- ac: add ac_build_tbuffer_load_byte() helper
- ac: add ac_build_tbuffer_store_byte() helper
- radv: add missing initializations since VK_EXT_pipeline_creation_feedback
- ac: add f16_0 and f16_1 constants
- ac: add 16-bit support fo fsign
- ac: add 16-bit support to fract
- ac: fix 16-bit shifts
- ac: fix incorrect argument type for tbuffer.{load,store} with LLVM 7
- nir: use generic float types for frexp_exp and frexp_sig
- spirv,nir: lower frexp_exp/frexp_sig inside a new NIR pass
- nir: add nir_{load,store}_deref_with_access() helpers
- spirv: propagate the access flag for store and load derefs
- ac: use llvm.amdgcn.fmed3 intrinsic for nir_op_fmed3
- ac: add ac_build_frexp_mant() helper and 16-bit/32-bit support

- ac: add ac_build_frex_exp() helper ans 16-bit/32-bit support
- radv: do not lower frexp_exp and frexp_sig
- radv: enable VK_AMD_gpu_shader_int16
- radv: skip updating depth/color metadata for conditional rendering
- radv: do not always initialize HTILE in compressed state
- ac: fix return type for llvm.amdgcn.frexp.exp.i32.64
- ac/nir: fix nir_op_b2i16
- ac: fix ac_build_bit_count() for 16-bit integer type
- ac: fix ac_build_bitfield_reverse() for 16-bit integer type
- ac: fix ac_find_lsb() for 16-bit integer type
- ac: fix ac_build_umsb() for 16-bit integer type
- ac/nir: add support for nir_op_b2i8
- ac: add 8-bit support to ac_build_bit_count()
- ac: add 8-bit support to ac_find_lsb()
- ac: add 8-bit support to ac_build_umsb()
- ac: add 8-bit and 64-bit support to ac_build_bitfield_reverse()
- radv: partially enable VK_KHR_shader_float16_int8
- nir: do not pack varying with different types
- ac/nir: fix intrinsic names for atomic operations with LLVM 9+
- radv: fix getting the vertex strides if the bindings aren't contiguous
- ac/nir: fix nir_op_b2f16
- radv: enable VK_AMD_gpu_shader_half_float
- wsi: allow to override the present mode with MESA_VK_WSI_PRESENT_MODE
- ac/nir: make use of ac_build_imax() where possible
- ac/nir: make use of ac_build_imin() where possible
- ac/nir: make use of ac_build_umin() where possible
- ac: add ac_build_umax() and use it where possible
- ac: add ac_build_ddxy_interp() helper
- ac: add ac_build_load_helper_invocation() helper
- ac/nir: remove useles LLVMGetUndef for nir_op_pack_64_2x32_split
- ac/nir: remove useless integer cast in adjust_sample_index_using_fmask()
- ac/nir: remove useless integer cast in visit_image_load()
- ac/nir: remove some useless integer casts for ALU operations
- spirv: add SpvCapabilityFloat16 support
- radv: enable VK_KHR_shader_float16_int8
- radv: set ACCESS_NON_READABLE on stores for copy/fill/clear meta shaders

- radv: enable shaderInt8 on SI and CIK
- radv: sort the shader capabilities alphabetically
- ac/nir: use new LLVM 8 intrinsics for SSBO atomics except cmpswap
- ac/nir: add 64-bit SSBO atomic operations support
- radv: add VK_KHR_shader_atomic_int64 but disable it for now
- ac: add support for more types with struct/raw LLVM intrinsics
- ac: use struct/raw load intrinsics for 8-bit/16-bit int with LLVM 9+
- ac: use struct/raw store intrinsics for 8-bit/16-bit int with LLVM 9+
- ac/nir: only use the new raw/struct image atomic intrinsics with LLVM 9+
- ac/nir: only use the new raw/struct SSBO atomic intrinsics with LLVM 9+
- ac/nir: use the new raw/struct SSBO atomic intrisics for comp_swap
- radv: add VK_NV_compute_shader_derivates support
- radv: add missing VEGA20 chip in radv_get_device_name()
- radv: do not need to force emit the TCS regs on Vega20
- radv: fix color conversions for normalized uint/sint formats
- radv: implement a workaround for VK_EXT_conditional_rendering
- ac: tidy up ac_build_llvm8_tbuffer_{load,store}
- radv: set WD_SWITCH_ON_EOP=1 when drawing primitives from a stream output buffer
- radv: only need to force emit the TCS regs on Vega10 and Raven1
- radv: fix radv_get_aspect_format() for D+S formats
- radv: apply the indexing workaround for atomic buffer operations on GFX9
- radv: fix setting the number of rectangles when it's dyanmic
- radv: add a workaround for Monster Hunter World and LLVM 7&8
- radv: allocate more space in the CS when emitting events
- radv: do not use gfx fast depth clears for layered depth/stencil images
- radv: fix alpha-to-coverage when there is unused color attachments
- radv: fix setting CB_SHADER_MASK for dual source blending

Sergii Romantsov (4):

- dri: meson: do not prefix user provided dri-drivers-path
- d3d: meson: do not prefix user provided d3d-drivers-path
- i965,iris/blorp: do not blit 0-sizes
- glsl: Fix input/output structure matching across shader stages

Sonny Jiang (1):

- radeonsi: use compute for clear_render_target when possible

Tapani Pälli (42):

- nir: add option to use scaling factor when sampling planes YUV lowering

- dri: add P010, P012, P016 for 10bit/12bit/16bit YUV420 formats
- intel/compiler: add scale_factors to sampler_prog_key_data
- i965: add P0x formats and propagate required scaling factors
- drirc/i965: add option to disable 565 configs and visuals
- mesa: return NULL if we exceed MaxColorAttachments in get_fb_attachment
- anv: anv: refactor error handling in anv_shader_bin_write_to_blob()
- iris: add Android build
- nir: initialize value in copy_prop_vars_block
- nir: use nir_variable_create instead of open-coding the logic
- android: add liblog to libmesa_intel_common build
- android: make libbacktrace optional on USE_LIBBACKTRACE
- iris: add libmesa_iris_gen8 library to the build
- util: fix a warning when building against clang7 headers
- anv: retain the is_array state in create_plane_tex_instr_implicit
- anv: toggle on support for VK_EXT_ycbcr_image_arrays
- anv: use anv_gem_munmap in block pool cleanup
- anv: call blob_finish when done with it
- nir: free dead_ctx in case of no progress
- anv: destroy descriptor sets when pool gets destroyed
- anv: release memory allocated by bo_heap when descriptor pool is destroyed
- anv: release memory allocated by glsl types during spirv_to_nir
- anv: revert "anv: release memory allocated by glsl types during spirv_to_nir"
- i965: remove scaling factors from P010, P012
- isl: fix automake build when sse41 is not supported
- android: Build fixes for OMR1
- iris: initialize num_cbufs
- iris: mark switch case fallthrough
- anv/radv: release memory allocated by glsl types during spirv_to_nir
- st/mesa: fix compilation warning on storage_flags_to_buffer_flags
- st/mesa: fix warnings about implicit conversion on enumeration type
- spirv: fix a compiler warning
- st/nir: run st_nir_opts after 64bit ops lowering
- iris: move variable to the scope where it is being used
- iris: move iris_flush_resource so we can call it from get_handle
- iris: handle aux properly in iris_resource_get_handle
- egl: setup fds array correctly when exporting dmabuf

- compiler/glsl: handle case where we have multiple users for types
- android/iris: fix driinfo header filename
- nir: use braces around subobject in initializer
- glsl: use empty brace initializer
- anv: expose VK_EXT_queue_family_foreign on Android

Thomas Hellstrom (5):

- winsys/svga: Add an environment variable to force host-backed operation
- winsys/svga: Enable the transfer_from_buffer GPU command for vgpu10
- svga: Avoid bouncing buffer data in malloced buffers
- winsys/svga: Update the drm interface file
- winsys/svga: Don't abort on EBUSY errors from execbuffer

Timo Aaltonen (1):

- util/os_misc: Add check for PIPE_OS_HURD

Timothy Arceri (72):

- st/glsl_to_nir: remove dead local variables
- ac/radv/radeonsi: add ac_get_num_physical_sgprs() helper
- radv: take LDS into account for compute shader occupancy stats
- util: move BITFIELD macros to util/macros.h
- st/glsl_to_nir: call nir_remove_dead_variables() after lowing local indirects
- nir: add support for marking used patches when packing varyings
- nir: add glsl_type_is_32bit() helper
- nir: add is_packing_supported_for_type() helper
- nir: rewrite varying component packing
- nir: prehash instruction in nir_instr_set_add_or_rewrite()
- nir: turn ssa check into an assert
- nir: turn an ssa check in nir_search into an assert
- nir: remove simple dead if detection from nir_opt_dead_cf()
- radeonsi/nir: set input_usage_mask properly
- radeonsi/nir: set colors_read properly
- radeonsi/nir: set shader_buffers_declared properly
- st/nir: use NIR for asm programs
- nir: remove non-ssa support from nir_copy_prop()
- nir: clone instruction set rather than removing individual entries
- nir: allow nir_lower_phis_to_scalar() on more src types
- radeonsi: fix query buffer allocation
- glsl: fix shader cache for packed param list

- radeonsi/nir: move si_lower_nir() call into compiler thread
- glsl: rename is_record() -> is_struct()
- glsl: rename get_record_instance() -> get_struct_instance()
- glsl: rename record_location_offset() -> struct_location_offset()
- glsl: rename record_types -> struct_types
- nir: rename glsl_type_is_struct() -> glsl_type_is_struct_or_ifc()
- glsl/freedreno/panfrost: pass gl_context to the standalone compiler
- glsl: use NIR function inlining for drivers that use glsl_to_nir()
- i965: stop calling nir_lower_returns()
- radeonsi/nir: stop calling nir_lower_returns()
- st/glsl: start spilling out common st glsl conversion code
- anv: add support for dumping shader info via VK_EXT_debug_report
- nir: add guess trip count support to loop analysis
- nir: add new partially_unrolled bool to nir_loop
- nir: add partial loop unrolling support
- nir: calculate trip count for more loops
- nir: unroll some loops with a variable limit
- nir: simplify the loop analysis trip count code a little
- nir: add helper to return inversion op of a comparison
- nir: add get_induction_and_limit_vars() helper to loop analysis
- nir: pass nir_op to calculate_iterations()
- nir: find induction/limit vars in iand instructions
- st/glsl_to_nir: fix incorrect arrary access
- radeonsi/nir: call some more var optimisation passes
- ac/nir_to_llvm: add assert to emit_bcsel()
- nir: only override previous alu during loop analysis if supported
- nir: fix opt_if_loop_last_continue()
- nir: add support for user defined loop control
- spirv: make use of the loop control support in nir
- nir: add support for user defined select control
- spirv: make use of the select control support in nir
- Revert “ac/nir: use new LLVM 8 intrinsics for SSBO atomic operations”
- nir: propagate known constant values into the if-then branch
- Revert “nir: propagate known constant values into the if-then branch”
- nir/radv: remove restrictions on opt_if_loop_last_continue()
- nir: initialise some variables in opt_if_loop_last_continue()

- nir/i965/freedreno/vc4: add a bindless bool to type size functions
- ac/nir_to_llvm: make get_sampler_desc() more generic and pass it the image intrinsic
- ac/nir_to_llvm: add image bindless support
- nir: fix packing components with arrays
- radeonsi/nir: fix scanning of bindless images
- st/mesa/radeonsi: fix race between destruction of types and shader compilation
- nir: fix nir_remove_unused_varyings()
- radeonsi/nir: create si_nir_opts() helper
- radeonsi/nir: call radeonsi nir opts before the scan pass
- util/drirc: add workarounds for bugs in Doom 3: BFG
- radeonsi: add config entry for Counter-Strike Global Offensive
- Revert "glx: Fix synthetic error generation in __glXSendError"
- Revert "st/mesa: expose 0 shader binary formats for compat profiles for Qt"
- st/glsl: make sure to propagate initialisers to driver storage

Timur Kristóf (19):

- radeonsi/nir: Use uniform location when calculating const_file_max.
- iris: implement clearing render target and depth stencil
- nir: Add ability for shaders to use window space coordinates.
- tgsi_to_nir: Fix the TGSI ARR translation by converting the result to int.
- tgsi_to_nir: Fix TGSI LIT translation by using flt.
- tgsi_to_nir: Make the TGSI IF translation code more readable.
- tgsi_to_nir: Split to smaller functions.
- nir: Move nir_lower_uniforms_to_ubo to compiler/nir.
- nir: Add multiplier argument to nir_lower_uniforms_to_ubo.
- freedreno: Plumb pipe_screen through to irX_tgsi_to_nir.
- tgsi_to_nir: Produce optimized NIR for a given pipe_screen.
- tgsi_to_nir: Restructure system value loads.
- tgsi_to_nir: Extract ttn_emulate_tgsi_front_face into its own function.
- tgsi_to_nir: Support FACE and POSITION properly.
- tgsi_to_nir: Improve interpolation modes.
- tgsi_to_nir: Set correct location for uniforms.
- radeonsi/nir: Only set window_space_position for vertex shaders.
- iris: Face should be a system value.
- gallium: fix autotools build of pipe_msm.la

Tobias Klausmann (1):

- vulkan/util: meson build - add wayland client include

Tomasz Figa (1):

- llvmpipe: Always return some fence in flush (v2)

Tomeu Vizoso (19):

- panfrost: Add gem_handle to panfrost_memory and panfrost_bo
- panfrost: Add backend targeting the DRM driver
- panfrost/midgard: Add support for MIDGARD_MESA_DEBUG
- panfrost: Add support for PAN_MESA_DEBUG
- panfrost: Set bo->size[0] in the DRM backend
- panfrost: Set bo->gem_handle when creating a linear BO
- panfrost: Adapt to uapi changes
- panfrost: Fix sscanf format options
- panfrost: Set the GEM handle for AFBC buffers
- panfrost: Also tell the kernel about the checksum_slab
- panfrost: Pass the context BOs to the kernel so they aren't unmapped while in use
- panfrost: Wait for last job to finish in force_flush_fragment
- panfrost: split asserts in pandecode
- panfrost: Guard against reading past end of buffer
- panfrost/ci: Initial commit
- panfrost/midgard: Skip register allocation if there's no work to do
- panfrost/midgard: Skip liveness analysis for instructions without dest
- panfrost: Fix two uninitialized accesses in compiler
- panfrost: Only take the fast paths on buffers aligned to block size

Toni Lönnberg (8):

- intel/genxml: Only handle instructions meant for render engine when generating headers
- intel/genxml: Media instructions and structures for gen6
- intel/genxml: Media instructions and structures for gen7
- intel/genxml: Media instructions and structures for gen7.5
- intel/genxml: Media instructions and structures for gen8
- intel/genxml: Media instructions and structures for gen9
- intel/genxml: Media instructions and structures for gen10
- intel/genxml: Media instructions and structures for gen11

Topi Pohjolainen (2):

- intel/compiler/icl: Use tcs barrier id bits 24:30 instead of 24:27
- intel/compiler/fs/icl: Use dummy masked urb write for tess eval

Vasily Khoruzhick (2):

- lima: use individual tile heap for each GP job.

- lima: add support for depth/stencil fbo attachments and textures

Vinson Lee (5):

- gallium/auxiliary/vl: Fix duplicate symbol build errors.
- nir: Fix anonymous union initialization with older GCC.
- swr: Fix build with llvm-9.0.
- gallium: Fix autotools build with libxatracker.la.
- freedreno: Fix GCC build error.

Vivek Kasireddy (1):

- drm-uapi: Update headers from drm-next

Xavier Bouchoux (1):

- nir/spirv: Fix assert when unsampled OpTypeImage has unknown 'Depth'

Yevhenii Kolesnikov (1):

- i965: Fix allow_higher_compat_version workaround limited by OpenGL 3.0

coypu (1):

- gbm: don't return void

davidbepo (1):

- drirc: add Waterfox to adaptive-sync blacklist

grmat (1):

- drirc: add Spectacle, Falkon to a-sync blacklist

pal1000 (1):

- scons: Compatibility with Scons development version string

suresh guttula (3):

- vl: Add cropping flags for H264
- radeon/vce:Add support for frame_cropping_flag of VAEncSequenceParameterBufferH264
- st/va/enc: Add support for frame_cropping_flag of VAEncSequenceParameterBufferH264

4.49 Mesa 19.0.6 Release Notes / May 21, 2019

Mesa 19.0.6 is a bug fix release which fixes bugs found since the 19.0.5 release.

Mesa 19.0.6 implements the OpenGL 4.5 API, but the version reported by glGetString(GL_VERSION) or glGetIntegerv(GL_MAJOR_VERSION) / glGetIntegerv(GL_MINOR_VERSION) depends on the particular driver being used. Some drivers don't support all the features required in OpenGL 4.5. OpenGL 4.5 is **only** available if requested at context creation. Compatibility contexts may report a lower version depending on each driver.

4.49.1 SHA256 checksums

```
SHA256: ac8e9ea388ec5c69f5a690190edf8ede602afdbaeea62d49e108057737430ac7  mesa-19.0.6.
↪tar.gz
SHA256: 2db2f2fcaa4048b16e066fad76b8a93944f7d06d329972b0f5fd5ce692ce3d24  mesa-19.0.6.
↪tar.xz
```

4.49.2 New features

N/A

4.49.3 Bug fixes

- Bug 110721 - graphics corruption on steam client with mesa 19.1.0 rc3 on polaris
- Bug 110761 - Huge problems between Mesa and Electron engine apps
- Bug 110784 - [regression][bisected] Reverting 'expose 0 shader binary formats for compat profiles for Qt' causes get_program_binary failures on Iris

4.49.4 Changes

Alok Hota (2):

- gallium/swr: Param defaults for unhandled PIPE_CAPs
- gallium/aux: add PIPE_CAP_MAX_VARYINGS to u_screen

Bas Nieuwenhuizen (1):

- nir: Actually propagate progress in nir_opt_move_load_ubo.

Chenglei Ren (1):

- anv/android: fix missing dependencies issue during parallel build

Christian Gmeiner (1):

- etnaviv: use the correct uniform dirty bits

Dave Airlie (1):

- Revert "mesa: unreference current winsys buffers when unbinding winsys buffers"

Deepak Rawat (1):

- winsys/drm: Fix out of scope variable usage

Dylan Baker (6):

- docs: Add Sha256 sums for 19.0.5
- cherry-ignore: Add a commit that was manually backported
- cherry-ignore: add another 19.1 only patch
- cherry-ignore: add another 19.1 only patch
- gallium: wrap u_screen in extern "C" for c++
- VERSION: bump to 19.0.6

Emil Velikov (1):

- egl/dri: flesh out and use dri2_create_drawable()

Jan Zielinski (1):

- swr/rast: fix 32-bit compilation on Linux

Lionel Landwerlin (1):

- vulkan: fix build dependency issue with generated files

Marek Olšák (2):

- u_blitter: don't fail mipmap generation for depth formats containing stencil
- ac: fix a typo in ac_build_wg_scan_bottom

Philipp Zabel (1):

- etnaviv: fill missing offset in etna_resource_get_handle

Rob Clark (3):

- freedreno/ir3: dynamic UBO indexing vs 64b pointers
- freedreno/ir3: set more barrier bits
- freedreno/a6xx: fix GPU crash on small render targets

Sagar Ghuge (1):

- intel/compiler: Fix assertions in brw_alu3

Samuel Pitoiset (2):

- radv: allocate more space in the CS when emitting events
- radv: do not use gfx fast depth clears for layered depth/stencil images

Timothy Arceri (2):

- Revert "st/mesa: expose 0 shader binary formats for compat profiles for Qt"
- st/glsl: make sure to propagate initialisers to driver storage

4.50 Mesa 19.0.5 Release Notes / May 21, 2019

Mesa 19.0.5 is a bug fix release which fixes bugs found since the 19.0.4 release.

Mesa 19.0.5 implements the OpenGL 4.5 API, but the version reported by glGetString(GL_VERSION) or glGetIntegerv(GL_MAJOR_VERSION) / glGetIntegerv(GL_MINOR_VERSION) depends on the particular driver being used. Some drivers don't support all the features required in OpenGL 4.5. OpenGL 4.5 is **only** available if requested at context creation. Compatibility contexts may report a lower version depending on each driver.

4.50.1 SHA256 checksums

```
b6e6b78c23bec15d1e7887c78b7ad00ce395ea1b20ad8aab6ce441f55f724e70  mesa-19.0.5.tar.gz
6aecb7f67c136768692fb3c33a54196186c6c4fcafab7973516a355e1a54f831  mesa-19.0.5.tar.xz
```

4.50.2 New features

N/A

4.50.3 Bug fixes

- Bug 109659 - Missing OpenGL symbols in OSMesa Gallium when building with meson
- Bug 110134 - SIGSEGV while playing large hevc video in mpv
- Bug 110648 - Dota2 will not open using vulkan since 19.0 series

4.50.4 Changes

Caio Marcelo de Oliveira Filho (2):

- nir: Fix nir_opt_idiv_const when negatives are involved
- nir: Fix clone of nir_variable state slots

Charmaine Lee (2):

- st/mesa: purge framebuffers with current context after unbinding winsys buffers
- mesa: unreference current winsys buffers when unbinding winsys buffers

Dylan Baker (4):

- docs: Add SHA256 sums for mesa 19.0.4
- cherry-ignore: add patches for panfrost
- cherry-ignore: Add more 19.1 patches
- bump version to 19.0.5

Eric Engestrom (1):

- meson: expose glapi through osmesa

Gert Wollny (2):

- softpipe/buffer: load only as many components as the the buffer resource type provides
- Revert "softpipe/buffer: load only as many components as the the buffer resource type provides"

Ian Romanick (1):

- Revert "nir: add late opt to turn inot/b2f combos back to bcsel"

Jason Ekstrand (3):

- intel/fs/ra: Only add dest interference to sources that exist
- intel/fs/ra: Stop adding RA interference to too many SENDS nodes
- anv: Only consider minSampleShading when sampleShadingEnable is set

Józef Kucia (1):

- radv: clear vertex bindings while resetting command buffer

Kenneth Graunke (1):

- i965: Fix memory leaks in brw_upload_cs_work_groups_surface().

Leo Liu (1):

- winsys/amdgpu: add VCN JPEG to no user fence group

Lionel Landwerlin (1):

- anv: Use corresponding type from the vector allocation

Marek Olšák (1):

- st/mesa: fix 2 crashes in st_tgsi_lower_yuv

Nanley Chery (1):

- anv: Fix some depth buffer sampling cases on ICL+

Samuel Pitoiset (1):

- radv: add a workaround for Monster Hunter World and LLVM 7&8

4.51 Mesa 19.0.4 Release Notes / May 9, 2019

Mesa 19.0.4 is a bug fix release which fixes bugs found since the 19.0.3 release.

Mesa 19.0.4 implements the OpenGL 4.5 API, but the version reported by glGetString(GL_VERSION) or glGetIntegerv(GL_MAJOR_VERSION) / glGetIntegerv(GL_MINOR_VERSION) depends on the particular driver being used. Some drivers don't support all the features required in OpenGL 4.5. OpenGL 4.5 is **only** available if requested at context creation. Compatibility contexts may report a lower version depending on each driver.

4.51.1 SHA256 checksums

```
de361c76bf7aae09219f571b9ae77a34864a1cd9f6ba24c845b18b3cd5e4b9a2  mesa-19.0.4.tar.gz
39f9f32f448d77388ef817c6098d50eb0c1595815ce7e895dec09dd68774ce47  mesa-19.0.4.tar.xz
```

4.51.2 New features

N/A

4.51.3 Bug fixes

- Bug 99781 - Some Unity games fail assertion on startup in glXCreateContextAttribsARB
- Bug 100239 - Incorrect rendering in CS:GO
- Bug 108540 - vkAcquireNextImageKHR blocks when timeout=0 in Wayland
- Bug 110143 - Doom 3: BFG Edition - Steam and GOG.com - white flickering screen
- Bug 110291 - Vega 64 GPU hang running Space Engineers
- Bug 110355 - radeonsi: GTK elements become invisible in some applications (GIMP, LibreOffice)
- Bug 110573 - Mesa vulkan-radeon 19.0.3 system freeze and visual artifacts (RADV)
- Bug 110590 - [Regression][Bisected] GTA under wine fails with GLXBadFBConfig
- Bug 110632 - "glx: Fix synthetic error generation in __glXSendError" broke wine games on 32-bit

4.51.4 Changes

Alejandro Piñeiro (1):

- docs: document MESA_GLSL=errors keyword

Andrii Simiklit (1):

- egl: return correct error code for a case req ver < 3 with forward-compatible

Axel Davy (1):

- st/nine: Fix D3DWindowBuffer_release for old wine nine support

Bas Nieuwenhuizen (1):

- radv: Disable VK_EXT_descriptor_indexing.

Brian Paul (1):

- svga: add SVGA_NO_LOGGING env var (v2)

Caio Marcelo de Oliveira Filho (1):

- spirv: Handle SpvOpDecorateId

Charmaine Lee (1):

- svga: move host logging to winsys

Chuck Atkins (1):

- meson: Fix missing glproto dependency for gallium-glx

Daniel Stone (1):

- vulkan/wsi/wayland: Respect non-blocking AcquireNextImage

Dave Airlie (2):

- r600: reset tex array override even when no view bound
- util/bitset: fix bitset range mask calculations.

Dylan Baker (7):

- docs: Add SHA256 sums for mesa 19.0.3
- cherry-ignore: Add a patch that was manually backported
- cherry-ignore: Add more backported patches
- cherry-ignore: Add another patch
- cherry-ignore: Add more patches
- meson: Force the use of config-tool for llvm
- VERSION: bump for 19.0.4 release

Emil Velikov (3):

- vulkan/wsi: check if the display_fd given is master
- vulkan/wsi: don't use DUMB_CLOSE for normal GEM handles
- configure.ac: check for libdrm when using VL with X11

Erik Faye-Lund (2):

- softpipe: setup pixel_offset for all primitive types

- draw: flush when setting stream-out targets

Francisco Jerez (2):

- intel/fs: Lower integer multiply correctly when destination stride equals 4.
- intel/fs: Cap dst-aligned region stride to maximum representable hstride value.

Hal Gentz (1):

- glx: Fix synthetic error generation in __glXSendError

Ian Romanick (2):

- glsl: Silence may unused parameter warnings in glsl/ir.h
- mesa: Add missing display list support for GL_FOG_COORDINATE_SOURCE

Jason Ekstrand (1):

- anv/descriptor_set: Destroy sets before pool finalization

Jon Turney (1):

- meson: Force '.so' extension for DRI drivers

Juan A. Suarez Romero (2):

- spirv: add missing SPV_EXT_descriptor_indexing capabilities
- radv: enable descriptor indexing capabilities

Kenneth Graunke (6):

- glsl: Allow gl_nir_lower_samplers*() without a gl_shader_program
- glsl: Don't look at sampler uniform storage for internal vars
- i965: Ignore uniform storage for samplers or images, use binding info
- i965: Fix BRW_MEMZONE_LOW_4G heap size.
- i965: Force VMA alignment to be a multiple of the page size.
- i965: leave the top 4Gb of the high heap VMA unused

Lionel Landwerlin (4):

- anv: store heap address bounds when initializing physical device
- anv: leave the top 4Gb of the high heap VMA unused
- anv: fix argument name for vkCmdEndQuery
- anv: rework queries writes to ensure ordering memory writes

Marek Olšák (2):

- radeonsi/gfx9: set that window_rectangles always roll the context
- radeonsi/gfx9: rework the gfx9 scissor bug workaround (v2)

Nicolai Hähnle (1):

- radeonsi: add si_debug_options for convenient adding/removing of options

Rhys Perry (1):

- radv: fix set_output_usage_mask() with composite and 64-bit types

Ross Burton (1):

- Revert "meson: drop GLESv1 .so version back to 1.0.0"

Samuel Pitoiset (8):

- radv: add missing VEGA20 chip in radv_get_device_name()
- radv: do not need to force emit the TCS regs on Vega20
- radv: fix color conversions for normalized uint/sint formats
- radv: implement a workaround for VK_EXT_conditional_rendering
- radv: set WD_SWITCH_ON_EOP=1 when drawing primitives from a stream output buffer
- radv: only need to force emit the TCS regs on Vega10 and Raven1
- radv: apply the indexing workaround for atomic buffer operations on GFX9
- radv: fix setting the number of rectangles when it's dyanmic

Tapani Pälli (1):

- anv: expose VK_EXT_queue_family_foreign on Android

Timothy Arceri (4):

- nir: fix nir_remove_unused_varyings()
- util/drirc: add workarounds for bugs in Doom 3: BFG
- radeonsi: add config entry for Counter-Strike Global Offensive
- Revert "glx: Fix synthetic error generation in __glXSendError"

4.52 Mesa 19.0.3 Release Notes / April 24, 2019

Mesa 19.0.3 is a bug fix release which fixes bugs found since the l9.0.2 release.

Mesa 19.0.3 implements the OpenGL 4.5 API, but the version reported by glGetString(GL_VERSION) or glGetIntegerv(GL_MAJOR_VERSION) / glGetIntegerv(GL_MINOR_VERSION) depends on the particular driver being used. Some drivers don't support all the features required in OpenGL 4.5. OpenGL 4.5 is **only** available if requested at context creation. Compatibility contexts may report a lower version depending on each driver.

4.52.1 SHA256 checksums

```
59543ec3c9f8c72990e77887f13d1678cb6739e5d5f56abc21ebf9e772389c5e  mesa-19.0.3.tar.gz
f027244e38dc309a4c12db45ef79be81ab62c797a50a88d566e4edb6159fc4d5  mesa-19.0.3.tar.xz
```

4.52.2 New features

N/A

4.52.3 Bug fixes

- Bug 108879 - [CIK] [regression] All opencl apps hangs indefinitely in si_create_context
- Bug 110201 - [ivb] mesa 19.0.0 breaks rendering in kitty
- Bug 110356 - install_megadrivers.py creates new dangling symlink [bisected]

- Bug 110441 - [llvmpipe] complex-loop-analysis-bug regression

4.52.4 Changes

Andres Gomez (1):

- glsl/linker: location aliasing requires types to have the same width

Bas Nieuwenhuizen (1):

- ac: Move has_local_buffers disable to radeonsi.

Chia-I Wu (1):

- virgl: fix fence fd version check

Danylo Piliaiev (1):

- intel/compiler: Do not reswizzle dst if instruction writes to flag register

Dylan Baker (2):

- docs: Add sha256 sums for 19.0.2
- Bump version for 19.0.3

Eric Anholt (1):

- nir: Fix deref offset calculation for structs.

Eric Engestrom (1):

- meson: remove meson-created megadrivers symlinks

Jason Ekstrand (2):

- anv/pipeline: Fix MEDIA_VFE_STATE::PerThreadScratchSpace on gen7
- anv: Add a #define for the max binding table size

Juan A. Suarez Romero (1):

- meson: Add dependency on genxml to anvil genfiles

Kenneth Graunke (2):

- glsl: Set location on structure-split sampler uniform variables
- Revert "glsl: Set location on structure-split sampler uniform variables"

Lionel Landwerlin (2):

- anv: fix uninitialized pthread cond clock domain
- intel/devinfo: fix missing num_thread_per_eu on ICL

Lubomir Rintel (2):

- gallivm: guess CPU features also on ARM
- gallivm: disable NEON instructions if they are not supported

Marek Olšák (1):

- radeonsi: use CP DMA for the null const buffer clear on CIK

Rhys Perry (1):

- nir,ac/nir: fix cube_face_coord

Roland Scheidegger (1):

- gallivm: fix bogus assert in get_indirect_index

Samuel Pitoiset (2):

- ac/nir: only use the new raw/struct image atomic intrinsics with LLVM 9+
- radv: do not load vertex attributes that are not provided by the pipeline

4.53 Mesa 19.0.2 Release Notes / April 10, 2019

Mesa 19.0.2 is a bug fix release which fixes bugs found since the 19.0.1 release.

Mesa 19.0.2 implements the OpenGL 4.5 API, but the version reported by glGetString(GL_VERSION) or glGetIntegerv(GL_MAJOR_VERSION) / glGetIntegerv(GL_MINOR_VERSION) depends on the particular driver being used. Some drivers don't support all the features required in OpenGL 4.5. OpenGL 4.5 is **only** available if requested at context creation. Compatibility contexts may report a lower version depending on each driver.

4.53.1 SHA256 checksums

```
SHA256: eb972fc11d4e1261d34ec0b91a701f158d4870c0428fb108353ae7eab64b1118  mesa-19.0.2.
↪tar.gz
SHA256: 1a2edc3ce56906a676c91e6851298db45903df1f5cb9827395a922c1452db802  mesa-19.0.2.
↪tar.xz
```

4.53.2 New features

4.53.3 Bug fixes

- Bug 108766 - Mesa built with meson has RPATH entries
- Bug 109648 - AMD Raven hang during va-api decoding
- Bug 110257 - Major artifacts in mpeg2 vaapi hw decoding
- Bug 110259 - radv: Sampling depth-stencil image in GENERAL layout returns nothing but zero (regression, bisected)

4.53.4 Changes

Boyuan Zhang (1):

- st/va: reverse qt matrix back to its original order

Caio Marcelo de Oliveira Filho (1):

- nir: Take if_uses into account when repairing SSA

Dylan Baker (2):

- docs: Add SHA256 sums for mesa 19.0.1
- VERSION: bump version for 19.0.2

Eric Anholt (3):

- dri3: Return the current swap interval from glXGetSwapIntervalMESA().
- v3d: Bump the maximum texture size to 4k for V3D 4.x.
- v3d: Don't try to use the TFU blit path if a scissor is enabled.

Eric Engestrom (1):

- meson: strip rpath from megadrivers

Jason Ekstrand (1):

- Revert "anv/radv: release memory allocated by glsl types during spirv_to_nir"

Karol Herbst (1):

- nir/print: fix printing the image_array intrinsic index

Leo Liu (2):

- radeon/vcn: add H.264 constrained baseline support
- radeon/vcn/vp9: search the render target from the whole list

Lionel Landwerlin (1):

- intel: add dependency on genxml generated files

Marek Olšák (1):

- radeonsi: fix assertion failure by using the correct type

Samuel Pitoiset (2):

- radv: skip updating depth/color metadata for conditional rendering
- radv: do not always initialize HTILE in compressed state

4.54 Mesa 18.3.6 Release Notes / April 5, 2019

Mesa 18.3.6 is a bug fix release which fixes bugs found since the 18.3.5 release.

Mesa 18.3.6 implements the OpenGL 4.5 API, but the version reported by glGetString(GL_VERSION) or glGetIntegerv(GL_MAJOR_VERSION) / glGetIntegerv(GL_MINOR_VERSION) depends on the particular driver being used. Some drivers don't support all the features required in OpenGL 4.5. OpenGL 4.5 is **only** available if requested at context creation. Compatibility contexts may report a lower version depending on each driver.

4.54.1 SHA256 checksums

```
4619d92afadf7072f7956599a2ccd0934fc45b4ddbc2eb865bdcb50ddf963f87  mesa-18.3.6.tar.gz
aaf17638dcf5a90b93b6389e152fdc9ef147768b09598f24d2c5cf482fcfc705  mesa-18.3.6.tar.xz
```

4.54.2 New features

None

4.54.3 Bug fixes

- Bug 100316 - Linking GLSL 1.30 shaders with invariant and deprecated variables triggers an 'mismatching invariant qualifiers' error
- Bug 108766 - Mesa built with meson has RPATH entries
- Bug 109648 - AMD Raven hang during va-api decoding
- Bug 109980 - [i915 CI][HSW] spec@arb_fragment_shader_interlock@arb_fragment_shader_interlock-image-load-store - fail
- Bug 110211 - If DESTDIR is set to an empty string, the dri drivers are not installed
- Bug 110221 - build error with meson
- Bug 110259 - radv: Sampling depth-stencil image in GENERAL layout returns nothing but zero (regression, bisected)

4.54.4 Changes

Andres Gomez (4):

- glsl: correctly validate component layout qualifier for dvec{3,4}
- glsl/linker: don't fail non static used inputs without matching outputs
- glsl/linker: simplify xfb_offset vs xfb_stride overflow check
- Revert "glsl: relax input->output validation for SSO programs"

Bas Nieuwenhuizen (2):

- radv: Use correct image view comparison for fast clears.
- ac/nir: Return frag_coord as integer.

Danylo Piliaiev (1):

- glsl: Cross validate variable's invariance by explicit invariance only

Dave Airlie (1):

- softpipe: fix texture view crashes

Dylan Baker (1):

- bin/install_megadrivers.py: Correctly handle DESTDIR=''

Emil Velikov (2):

- docs: add sha256 checksums for 18.3.5
- Update version to 18.3.6

Eric Anholt (1):

- dri3: Return the current swap interval from glXGetSwapIntervalMESA().

Eric Engestrom (1):

- meson: strip rpath from megadrivers

Jason Ekstrand (2):

- anv/pass: Flag the need for a RT flush for resolve attachments

- Revert "anv/radv: release memory allocated by glsl types during spirv_to_nir"

Józef Kucia (2):

- mesa: Fix GL_NUM_DEVICE_UUIDS_EXT
- radv: Fix driverUUID

Leo Liu (2):

- radeon/vcn: add H.264 constrained baseline support
- radeon/vcn/vp9: search the render target from the whole list

Marek Olšák (1):

- radeonsi: fix assertion failure by using the correct type

Mark Janes (1):

- mesa: properly report the length of truncated log messages

Plamena Manolova (1):

- i965: Disable ARB_fragment_shader_interlock for platforms prior to GEN9

Samuel Pitoiset (2):

- radv: fix binding transform feedback buffers
- radv: do not always initialize HTILE in compressed state

Tapani Pälli (1):

- anv/radv: release memory allocated by glsl types during spirv_to_nir

Timothy Arceri (1):

- st/glsl_to_nir: fix incorrect arrary access

Tobias Klausmann (1):

- vulkan/util: meson build - add wayland client include

4.55 Mesa 19.0.1 Release Notes / March 27, 2019

Mesa 19.0.1 is a bug fix release which fixes bugs found since the 19.0.0 release.

Mesa 19.0.1 implements the OpenGL 4.5 API, but the version reported by glGetString(GL_VERSION) or glGetIntegerv(GL_MAJOR_VERSION) / glGetIntegerv(GL_MINOR_VERSION) depends on the particular driver being used. Some drivers don't support all the features required in OpenGL 4.5. OpenGL 4.5 is **only** available if requested at context creation. Compatibility contexts may report a lower version depending on each driver.

4.55.1 SHA256 checksums

```
f1dd1980ed628edea3935eed7974fbc5d8353e9578c562728b880d63ac613dbd  mesa-19.0.1.tar.gz
6884163c0ea9e4c98378ab8fecd72fe7b5f437713a14471beda378df247999d4  mesa-19.0.1.tar.xz
```

4.55.2 New features

None

4.55.3 Bug fixes

- Bug 100316 - Linking GLSL 1.30 shaders with invariant and deprecated variables triggers an 'mismatching invariant qualifiers' error
- Bug 107563 - [RADV] Broken rendering in Unity demos
- Bug 109698 - dri.pc contents invalid when built with meson
- Bug 109980 - [i915 CI][HSW] spec@arb_fragment_shader_interlock@arb_fragment_shader_interlock-image-load-store - fail
- Bug 110211 - If DESTDIR is set to an empty string, the dri drivers are not installed
- Bug 110221 - build error with meson

4.55.4 Changes

Andres Gomez (4):

- glsl: correctly validate component layout qualifier for dvec{3,4}
- glsl/linker: don't fail non static used inputs without matching outputs
- glsl/linker: simplify xfb_offset vs xfb_stride overflow check
- Revert "glsl: relax input->output validation for SSO programs"

Bas Nieuwenhuizen (2):

- radv: Use correct image view comparison for fast clears.
- ac/nir: Return frag_coord as integer.

Danylo Piliaiev (2):

- anv: Treat zero size XFB buffer as disabled
- glsl: Cross validate variable's invariance by explicit invariance only

Dave Airlie (1):

- softpipe: fix texture view crashes

Dylan Baker (5):

- docs: Add SHA256 sums for 19.0.0
- cherry-ignore: Add commit that doesn't apply
- bin/install_megadrivers.py: Correctly handle DESTDIR=''
- bin/install_megadrivers.py: Fix regression for set DESTDIR
- bump version for 19.0.1

Eric Anholt (1):

- v3d: Fix leak of the renderonly struct on screen destruction.

Jason Ekstrand (6):

- glsl/lower_vector_derefs: Don't use a temporary for TCS outputs
- glsl/list: Add a list variant of insert_after
- anv/pass: Flag the need for a RT flush for resolve attachments

- nir/builder: Add a vector extract helper
- nir: Add a new pass to lower array dereferences on vectors
- intel/nir: Lower array-deref-of-vector UBO and SSBO loads

Józef Kucia (2):

- radv: Fix driverUUID
- mesa: Fix GL_NUM_DEVICE_UUIDS_EXT

Kenneth Graunke (1):

- intel/fs: Fix opt_peephole_csel to not throw away saturates.

Kevin Strasser (1):

- egl/dri: Avoid out of bounds array access

Mark Janes (1):

- mesa: properly report the length of truncated log messages

Plamena Manolova (1):

- i965: Disable ARB_fragment_shader_interlock for platforms prior to GEN9

Samuel Pitoiset (3):

- radv: set the maximum number of IBs per submit to 192
- radv: always initialize HTILE when the src layout is UNDEFINED
- radv: fix binding transform feedback buffers

Sergii Romantsov (1):

- d3d: meson: do not prefix user provided d3d-drivers-path

Tapani Pälli (2):

- isl: fix automake build when sse41 is not supported
- anv/radv: release memory allocated by glsl types during spirv_to_nir

4.56 Mesa 18.3.5 Release Notes / March 18, 2019

Mesa 18.3.5 is a bug fix release which fixes bugs found since the 18.3.4 release.

Mesa 18.3.5 implements the OpenGL 4.5 API, but the version reported by glGetString(GL_VERSION) or glGetIntegerv(GL_MAJOR_VERSION) / glGetIntegerv(GL_MINOR_VERSION) depends on the particular driver being used. Some drivers don't support all the features required in OpenGL 4.5. OpenGL 4.5 is **only** available if requested at context creation. Compatibility contexts may report a lower version depending on each driver.

4.56.1 SHA256 checksums

```
5f40a336cb2af9b1d66fa243bb03c2c8a3f9b3f067aab6aaaad4316d1bc0e58b  mesa-18.3.5.tar.gz
4027aea82cc63240b3fcf60eec9eea882955f098c989b29357b01d1695747953  mesa-18.3.5.tar.xz
```

4.56.2 New features

None

4.56.3 Bug fixes

- Bug 104297 - [i965] Downward causes GPU hangs and misrendering on Haswell
- Bug 104602 - [apitrace] Graphical artifacts in Civilization VI on RX Vega
- Bug 107052 - [Regression][bisected]. Crookz - The Big Heist Demo can't be launched despite the "true" flag in "drirc"
- Bug 107563 - [RADV] Broken rendering in Unity demos
- Bug 108457 - [OpenGL CTS] KHR-GL46.tessellation_shader.single.xfb_captures_data_from_correct_stage fails
- Bug 108999 - Calculating the scissors fields when the y is flipped (0 on top) can generate negative numbers that will cause assertion failure later on.
- Bug 109328 - [BSW BXT GLK] dEQP-VK.subgroups.arithmetic.subgroup regressions
- Bug 109443 - Build failure with MSVC when using Scons >= 3.0.2
- Bug 109451 - [IVB,SNB] LINE_STRIPs following a TRIANGLE_FAN fail to use primitive restart
- Bug 109594 - totem assert failure: totem: src/intel/genxml/gen9_pack.h:72: __gen_uint: La declaración 'v <= max' no se cumple.
- Bug 109597 - wreckfest issues with transparent objects & skybox
- Bug 109601 - [Regression] RuneLite GPU rendering broken on 18.3.x
- Bug 109698 - dri.pc contents invalid when built with meson
- Bug 109735 - [Regression] broken font with mesa_vulkan_overlay

4.56.4 Changes

Alok Hota (1):

- swr/rast: bypass size limit for non-sampled textures

Andrii Simiklit (1):

- i965: re-emit index buffer state on a reset option change.

Axel Davy (2):

- st/nine: Ignore window size if error
- st/nine: Ignore multisample quality level if no ms

Bas Nieuwenhuizen (4):

- radv: Sync ETC2 whitelisted devices.
- radv: Fix float16 interpolation set up.
- radv: Allow interpolation on non-float types.
- radv: Interpolate less aggressively.

Carlos Garnacho (1):

- wayland/egl: Ensure EGL surface is resized on DRI update_buffers()

Danylo Piliaiev (1):

- glsl/linker: Fix unmatched TCS outputs being reduced to local variable

David Shao (1):

- meson: ensure that xmlpool_options.h is generated for gallium targets that need it

Eleni Maria Stea (1):

- i965: fixed clamping in set_scissor_bits when the y is flipped

Emil Velikov (7):

- docs: add sha256 checksums for 18.3.4
- meson: egl: correctly manage loader/xmlconfig
- cherry-ignore: add 19.0 only anv/push buffer nominations
- cherry-ignore: add gitlab-ci fixup commit
- cherry-ignore: ignore glsl_types memory cleanup patch
- cherry-ignore: add explicit 19.0 performance optimisations
- Update version to 18.3.5

Eric Engestrom (1):

- egl: fix libdrm-less builds

Francisco Jerez (1):

- intel/fs: Implement extended strides greater than 4 for IR source regions.

Ian Romanick (2):

- intel/fs: nir_op_extract_i8 extracts a byte, not a word
- intel/fs: Fix extract_u8 of an odd byte from a 64-bit integer

Ilia Mirkin (1):

- glsl: fix recording of variables for XFB in TCS shaders

Jason Ekstrand (10):

- intel/fs: Bail in optimize_extract_to_float if we have modifiers
- compiler/types: Add a contains_64bit helper
- nir/xfb: Properly align 64-bit values
- nir/xfb: Work in terms of components rather than slots
- nir/xfb: Handle compact arrays in gather_xfb_info
- anv: Count surfaces for non-YCbCr images in GetDescriptorSetLayoutSupport
- spirv: OpImageQueryLod requires a sampler
- spirv: Pull offset/stride from the pointer for OpArrayLength
- glsl/list: Add a list variant of insert_after
- glsl/lower_vector_derefs: Don't use a temporary for TCS outputs

Jose Maria Casanova Crespo (1):

- glsl: TCS outputs can not be transform feedback candidates on GLES

José Fonseca (1):

- scons: Workaround failures with MSVC when using SCons 3.0.[2-4].

Juan A. Suarez Romero (3):

- genxml: add missing field values for 3DSTATE_SF
- anv: advertise 8 subpixel precision bits
- anv: destroy descriptor sets when pool gets reset

Kenneth Graunke (1):

- intel/fs: Fix opt_peephole_csel to not throw away saturates.

Kevin Strasser (1):

- egl/dri: Avoid out of bounds array access

Lionel Landwerlin (1):

- intel: fix urb size for CFL GT1

Marek Olšák (5):

- radeonsi: add driconf option radeonsi_enable_nir
- radeonsi: always enable NIR for Civilization 6 to fix corruption
- driconf: add Civ6Sub executable for Civilization 6
- tgsi: don't set tgsi_info::uses_bindless_images for constbufs and hw atomics
- radeonsi: compile clear and copy buffer compute shaders on demand

Mauro Rossi (2):

- android: anv: fix generated files depedencies (v2)
- android: anv: fix libexpat shared dependency

Ray Zhang (1):

- glx: fix shared memory leak in X11

Rhys Perry (2):

- radv: bitcast 16-bit outputs to integers
- radv: ensure export arguments are always float

Samuel Pitoiset (8):

- radv: write the alpha channel of MRT0 when alpha coverage is enabled
- radv: fix writing the alpha channel of MRT0 when alpha coverage is enabled
- radv: fix clearing attachments in secondary command buffers
- radv: fix out-of-bounds access when copying descriptors BO list
- radv: don't copy buffer descriptors list for samplers
- radv: properly align the fence and EOP bug VA on GFX9
- radv: fix pointSizeRange limits
- radv: always initialize HTILE when the src layout is UNDEFINED

Sergii Romantsov (2):

- dri: meson: do not prefix user provided dri-drivers-path
- d3d: meson: do not prefix user provided d3d-drivers-path

Tapani Pälli (3):

- nir: initialize value in copy_prop_vars_block
- anv: retain the is_array state in create_plane_tex_instr_implicit
- anv: destroy descriptor sets when pool gets destroyed

Timothy Arceri (1):

- glsl: fix shader cache for packed param list

Yevhenii Kolesnikov (1):

- i965: Fix allow_higher_compat_version workaround limited by OpenGL 3.0

pal1000 (1):

- scons: Compatibility with Scons development version string

4.57 Mesa 19.0.0 Release Notes / TBD

Mesa 19.0.0 is a new development release. People who are concerned with stability and reliability should stick with a previous release or wait for Mesa 19.0.1.

Mesa 19.0.0 implements the OpenGL 4.5 API, but the version reported by glGetString(GL_VERSION) or glGetIntegerv(GL_MAJOR_VERSION) / glGetIntegerv(GL_MINOR_VERSION) depends on the particular driver being used. Some drivers don't support all the features required in OpenGL 4.5. OpenGL 4.5 is **only** available if requested at context creation. Compatibility contexts may report a lower version depending on each driver.

4.57.1 SHA256 checksums

```
4c5b9c5227d37c1f6bdc786a6fa7ee7fbce40b2e8a87340c7d3234534ece3304  mesa-19.0.0.tar.gz
5a549dfb40ec31e5c36c47aadac04554cb2e2a8d144a046a378fc16da57e38f8  mesa-19.0.0.tar.xz
```

4.57.2 New features

- GL_AMD_texture_texture4 on all GL 4.0 drivers.
- GL_EXT_shader_implicit_conversions on all drivers (ES extension).
- GL_EXT_texture_compression_bptc on all GL 4.0 drivers (ES extension).
- GL_EXT_texture_compression_rgtc on all GL 3.0 drivers (ES extension).
- GL_EXT_render_snorm on gallium drivers (ES extension).
- GL_EXT_texture_view on drivers supporting texture views (ES extension).
- GL_OES_texture_view on drivers supporting texture views (ES extension).
- GL_NV_shader_atomic_float on nvc0 (Fermi/Kepler only).

- Shader-based software implementations of GL_ARB_gpu_shader_fp64, GL_ARB_gpu_shader_int64, GL_ARB_vertex_attrib_64bit, and GL_ARB_shader_ballot on i965.
- VK_ANDROID_external_memory_android_hardware_buffer on Intel
- Fixed and re-exposed VK_EXT_pci_bus_info on Intel and RADV
- VK_EXT_scalar_block_layout on Intel and RADV
- VK_KHR_depth_stencil_resolve on Intel
- VK_KHR_draw_indirect_count on Intel
- VK_EXT_conditional_rendering on Intel
- VK_EXT_memory_budget on RADV

4.57.3 Bug fixes

- Bug 32211 - [GLSL] lower_jumps with continue-statements in for-loops prevents loop unrolling
- Bug 102349 - nv4x crashing with plasmashell - gdb log included
- Bug 102597 - [Regression] mpv, high rendering times (two to three times higher)
- Bug 104297 - [i965] Downward causes GPU hangs and misrendering on Haswell
- Bug 104602 - [apitrace] Graphical artifacts in Civilization VI on RX Vega
- Bug 105301 - The big SKQP bug
- Bug 106577 - broken rendering with nine and nouveau (GM107)
- Bug 106595 - [RADV] Rendering distortions only when MSAA is enabled
- Bug 107052 - [Regression][bisected]. Crookz - The Big Heist Demo can't be launched despite the "true" flag in "drirc"
- Bug 107510 - [GEN8+] up to 10% perf drop on several 3D benchmarks
- Bug 107626 - [SNB] The graphical corruption and GPU hang occur sometimes on the piglit test "arb_texture_multisample-large-float-texture" with parameter –fp16
- Bug 107728 - Wrong background in Sascha Willem's Multisampling Demo
- Bug 107842 - "invariant" qualifier on outputs of GLSL ES fragment shader causes compilation error.
- Bug 107856 - i965 incorrectly calculates the number of layers for texture views (assert)
- Bug 108114 - [vulkancts] new VK_KHR_16bit_storage tests fail.
- Bug 108116 - [vulkancts] stencil partial clear tests fail.
- Bug 108245 - RADV/Vega: Low mip levels of large BCn textures get corrupted by vkCmdCopyBufferToImage
- Bug 108311 - Query buffer object support is broken on r600.
- Bug 108457 - [OpenGL CTS] KHR-GL46.tessellation_shader.single.xfb_captures_data_from_correct_stage fails
- Bug 108560 - Mesa 32 is built without sse
- Bug 108624 - [regression][bisected] "nir: Copy propagation between blocks" regression
- Bug 108630 - [G965] piglit.spec.!opengl 1_2.tex3d-maxsize spins forever
- Bug 108635 - Mesa master commit 68dc591af16ebb36814e4c187e4998948103c99c causes XWayland to segfault

- Bug 108636 - test_optpass has use after free bug, failing with memory testing tools like address sanitizer
- Bug 108713 - Gallium: use after free with transform feedback
- Bug 108734 - Regression: [bisected] dEQP-GLES31.functional.tessellation.invariance.* start failing on r600
- Bug 108805 - i965 regressions from EXT_texture_sRGB_R8
- Bug 108829 - [meson] libglapi exports internal API
- Bug 108868 - [BYT IVB] Tesselation test regressions
- Bug 108877 - OpenGL CTS gl43 test cases were interrupted due to segment fault
- Bug 108894 - [anv] vkCmdCopyBuffer() and vkCmdCopyQueryPoolResults() write-after-write hazard
- Bug 108909 - Vkd3d test failure test_resolve_non_issued_query_data()
- Bug 108910 - Vkd3d test failure test_multisample_array_texture()
- Bug 108911 - Vkd3d test failure test_clear_render_target_view()
- Bug 108914 - blocky shadow artifacts in The Forest with DXVK, RADV_DEBUG=nohiz fixes this
- Bug 108925 - vkCmdCopyQueryPoolResults(VK_QUERY_RESULT_WAIT_BIT) for timestamps with large query count hangs
- Bug 108936 - [ILK,G45,G965] Regressions from texture-format enums rework
- Bug 108943 - Build fails on ppc64le with meson
- Bug 108961 - make check test_replace_src_bitsize failure
- Bug 108974 - make check DispatchSanity_test regression
- Bug 108999 - Calculating the scissors fields when the y is flipped (0 on top) can generate negative numbers that will cause assertion failure later on.
- Bug 109023 - error: inlining failed in call to always_inline '__m512 _mm512_and_ps(__m512, __m512)': target specific option mismatch
- Bug 109072 - GPU hang in blender 2.80
- Bug 109075 - radv: New D3D boolean optimizations cause GPU hang in Witcher 3
- Bug 109081 - [bisected] [HSW] Regression in clipping.user_defined.clip_* vulkancts tests
- Bug 109086 - Crash software mesa with gl_select render mode
- Bug 109107 - gallium/st/va: change va max_profiles when using Radeon VCN Hardware
- Bug 109129 - format_types.h:1220: undefined reference to '_mm256_cvtps_ph'
- Bug 109151 - [KBL-G][vulkan] dEQP-VK.texture.explicit_lod.2d.sizes.31x55_nearest_linear_mipmap_nearest_repeat failed verification.
- Bug 109190 - virgl: buffer flushing error with some dEQP tests [bisected]
- Bug 109202 - nv50_ir.cpp:749:19: error: cannot use typeid with -fno-rtti
- Bug 109204 - [regression, bisected] retroarch's crt-royale shader crash radv
- Bug 109229 - glLinkProgram locks up for ~30 seconds
- Bug 109231 - [nir] src/compiler/nir/nir_loop_analyze.c uninitialized variable
- Bug 109242 - [RADV] The Witcher 3 system freeze
- Bug 109304 - GfxBench AztecRuins Vulkan version Segfault

- Bug 109325 - mesa: Need ability to retrieve command line of Meson configuration
- Bug 109328 - [BSW BXT GLK] dEQP-VK.subgroups.arithmetic.subgroup regressions
- Bug 109353 - [regression][bisected] "nir: Switch to using 1-bit Booleans for almost everything" regression with shared bools
- Bug 109401 - [DXVK] Project Cars rendering problems
- Bug 109404 - [ANV] The Witcher 3 shadows flickering
- Bug 109442 - "make check" test anv_block_pool_no_free fails intermittently
- Bug 109443 - Build failure with MSVC when using Scons >= 3.0.2
- Bug 109449 - [snb] quakespasm triggers a segmentation fault.
- Bug 109451 - [IVB,SNB] LINE_STRIPs following a TRIANGLE_FAN fail to use primitive restart
- Bug 109543 - After upgrade mesa to 19.0.0~rc1 all vulkan based application stop working ["vulkan-cube" received SIGSEGV in radv_pipeline_init_blend_state at ../src/amd/vulkan/radv_pipeline.c:699]
- Bug 109561 - [regression, bisected] code re-factor causing games to stutter or lock-up system
- Bug 109573 - dEQP-VK.spirv_assembly.instruction.graphics.module.same_module
- Bug 109575 - Mesa-19.0.0-rc1 : Computer Crashes trying to run anything Vulkan
- Bug 109581 - [BISECTED] Nothing is Rendered on Sascha Willem's "subpasses" demo
- Bug 109594 - totem assert failure: totem: src/intel/genxml/gen9_pack.h:72: __gen_uint: La declaración 'v <= max' no se cumple.
- Bug 109597 - wreckfest issues with transparent objects & skybox
- Bug 109601 - [Regression] RuneLite GPU rendering broken on 18.3.x
- Bug 109603 - nir_instr_as_deref: Assertion 'parent && parent->type == nir_instr_type_deref' failed.
- Bug 109698 - dri.pc contents invalid when built with meson
- Bug 109717 - [regression] Cull distance tests asserting
- Bug 109735 - [Regression] broken font with mesa_vulkan_overlay
- Bug 109759 - [BISECTED][REGRESSION][IVB, HSW] Font rendering problem in OpenGL

4.57.4 Changes

Adam Jackson (4):

- glx: Demand success from CreateContext requests (v2)
- specs: Remove GLES profile interaction text from GLX_MESA_query_renderer
- specs: Remove GLX_RENDERER_ID_MESA from GLX_MESA_query_renderer
- specs: Bump GLX_MESA_query_renderer to version 9

Aditya Swarup (1):

- i965: Lift restriction in external textures for EGLImage support

Alejandro Piñeiro (3):

- nir: remove unused variable
- nir/xfb: don't assert when xfb_buffer/stride is present but not xfb_offset

- nir/xfb: distinguish array of structs vs array of blocks

Alex Deucher (3):

- pci_ids: add new vega10 pci ids
- pci_ids: add new vega20 pci id
- pci_ids: add new VegaM pci id

Alex Smith (1):

- radv: Flush before vkCmdWriteTimestamp() if needed

Alexander von Gluck IV (1):

- egl/haiku: Fix reference to disp vs dpy

Alok Hota (8):

- swr/rast: Use gfxptr_t value in JitGatherVertices
- swr/rast: Add annotator to interleave isa text
- swr/rast: partial support for Tiled Resources
- swr/rast: Unaligned and translations in gathers
- swr/rast: Scope MEM_CLIENT enum for mem usages
- swr/rast: New execution engine per JIT
- swr/rast: Store cached files in multiple subdirs
- swr/rast: bypass size limit for non-sampled textures

Alyssa Rosenzweig (1):

- util: Fix warning in u_cpu_detect on non-x86

Andre Heider (4):

- st/nine: fix stack corruption due to ABI mismatch
- st/nine: plug thread related leaks
- st/nine: clean up thead shutdown sequence a bit
- d3dadapter9: use snprintf(. . . , "%s", . . .) instead of strncpy

Andres Gomez (8):

- glsl/linker: complete documentation for assign_attribute_or_color_locations
- docs: update 18.3 and add 19.x cycles for the release calendar
- glsl: correct typo in GLSL compilation error message
- editorconfig: Add max_line_length property
- glsl/linker: specify proper direction in location aliasing error
- docs: complete the calendar and release schedule documentation
- bin/get-pick-list.sh: fix the oneline printing
- bin/get-pick-list.sh: fix redirection in sh

Andrii Simiklit (9):

- intel/tools: avoid 'unused variable' warnings

- compiler: avoid 'unused variable' warnings
- i965: avoid 'unused variable' warnings
- i965/batch: avoid reverting batch buffer if saved state is an empty
- intel/tools: make sure the binary file is properly read
- anv/pipeline: remove unnecessary null-pointer check
- intel/batch-decoder: fix vertex buffer size calculation for gen<8
- intel/batch-decoder: fix a vb end address calculation
- i965: re-emit index buffer state on a reset option change.

Anuj Phogat (7):

- i965/icl: Set Error Detection Behavior Control Bit in L3CNTLREG
- anv/icl: Set Error Detection Behavior Control Bit in L3CNTLREG
- anv/icl: Disable prefetching of sampler state entries
- i965/icl: Fix L3 configurations
- i965/icl: Set use full ways in L3CNTLREG
- intel/icl: Set way_size_per_bank to 4
- anv/icl: Set use full ways in L3CNTLREG

Axel Davy (12):

- st/nine: Allow 'triple buffering' with thread_submit
- st/nine: Remove thread_submit warning
- st/nine: Use helper to release swapchain buffers later
- st/nine: Switch to presentation buffer if resize is detected
- st/nine: Fix volumetexture dtor on ctor failure
- st/nine: Bind src not dst in nine_context_box_upload
- st/nine: Add src reference to nine_context_range_upload
- st/nine: Increase the limit of cached ff shaders
- st/nine: Immediately upload user provided textures
- st/nine: Enable debug info if NDEBUG is not set
- st/nine: Ignore window size if error
- st/nine: Ignore multisample quality level if no ms

Bart Oldeman (1):

- gallium-xlib: query MIT-SHM before using it.

Bas Nieuwenhuizen (41):

- radv: Use structured intrinsics instead of indexing workaround for GFX9.
- vulkan: Allow storage images in the WSI.
- radv: Fix opaque metadata descriptor last layer.
- radv: Clamp gfx9 image view extents to the allocated image extents.

- radv: Align large buffers to the fragment size.
- radv/android: Mark android WSI image as shareable.
- radv/android: Use buffer metadata to determine scanout compat.
- radv: Check for shareable images in central place.
- radv: Remove redundant format check.
- radv: Fix multiview depth clears
- radv: Work around non-renderable 128bpp compressed 3d textures on GFX9.
- radv: Fix wrongly positioned paren.
- radv: Do a cache flush if needed before reading predicates.
- radv: Implement buffer stores with less than 4 components.
- anv/android: Do not reject storage images.
- radv: Remove device path.
- radv: Remove unused variable.
- amd/common: Add some parentheses to silence warning.
- radv: Fix rasterization precision bits.
- spirv: Fix matrix parameters in function calls.
- freedreno: Move register constant files to src/freedreno.
- radv: Only use 32 KiB per threadgroup on Stoney.
- radv: Set partial_vs_wave for pipelines with just GS, not tess.
- nir: Account for atomics in copy propagation.
- radv: Remove unused variable.
- radv/winsys: Set winsys bo priority on creation.
- radv/winsys: Add priority handling during submit.
- radv: Enable VK_EXT_memory_priority.
- radv: Fix the shader info pass for not having the variable.
- amd/common: Fix stores to derefs with unknown variable.
- amd/common: Add gep helper for pointer increment.
- amd/common: Handle nir_deref_type_ptr_as_array for shared memory.
- amd/common: handle nir_deref_cast for shared memory from integers.
- radv: Only look at pImmutableSamples if the descriptor has a sampler.
- amd/common: Use correct writemask for shared memory stores.
- radv: Sync ETC2 whitelisted devices.
- radv: Fix float16 interpolation set up.
- radv: Allow interpolation on non-float types.
- radv: Handle clip+cull distances more generally as compact arrays.
- radv: Fix rebase issue in 19.0 for float16 fix.

- radv: Interpolate less aggressively.

Boyan Ding (3):

- gk110/ir: Add rcp f64 implementation
- gk110/ir: Add rsq f64 implementation
- gk110/ir: Use the new rcp/rsq in library

Brian Paul (3):

- svga: add new gallium formats to the format conversion table
- mesa: fix display list corner case assertion
- svga: remove SVGA_RELOC_READ flag in SVGA3D_BindGBSurface()

Bruce Cherniak (1):

- gallium/swr: Fix multi-context sync fence deadlock.

Caio Marcelo de Oliveira Filho (10):

- nir: properly clear the entry sources in copy_prop_vars
- nir: properly find the entry to keep in copy_prop_vars
- nir: add a way to print the deref chain
- nir: remove dead code from copy_prop_vars
- nir: fix warning in nir_lower_io.c
- util: Helper to create sets and hashes with pointer keys
- src/compiler: use new hash table and set creation helpers
- src/intel: use new hash table and set creation helpers
- nir: check NIR_SKIP to skip passes by name
- gallium: Add PIPE_CAP_GLSL_TESS_LEVELS_AS_INPUTS

Carlos Garnacho (1):

- wayland/egl: Ensure EGL surface is resized on DRI update_buffers()

Carsten Haitzler (Rasterman) (2):

- vc4: Use named parameters for the NEON inline asm.
- vc4: Declare the cpu pointers as being modified in NEON asm.

Chad Versace (1):

- i965: Fix -Wswitch on INTEL_COPY_STREAMING_LOAD

Chia-I Wu (2):

- meson: fix EGL/X11 build without GLX
- freedreno/drm: sync uapi again

Christian Gmeiner (6):

- nir: add lowering for ffloor
- etnaviv: drop redundant ctx function parameter
- meson: add etnaviv to the tools option

- etnaviv: extend etna_resource with an addressing mode
- etnaviv: update headers from rnndb
- etnaviv: add linear sampling support

Connor Abbott (4):

- Revert "radv: disable VK_SUBGROUP_FEATURE_VOTE_BIT"
- nir/algebraic: Rewrite bit-size inference
- nir/algebraic: Add unit tests for bitsize validation
- nir: Fixup algebraic test for variable-sized conversions

Daniel Stone (1):

- gbm: Clarify acceptable formats for gbm_bo

Danylo Piliaiev (9):

- i965: Fix calculation of layers array length for isl_view
- nir: add if opt opt_if_loop_last_continue()
- glsl/linker: Fix unmatched TCS outputs being reduced to local variable
- glsl: Make invariant outputs in ES fragment shader not to cause error
- glsl: Fix copying function's out to temp if dereferenced by array
- anv: Implement VK_KHR_draw_indirect_count for gen 7+
- anv: Implement VK_EXT_conditional_rendering for gen 7.5+
- anv: Fix VK_EXT_transform_feedback working with varyings packed in PSIZ
- anv: Fix destroying descriptor sets when pool gets reset

Dave Airlie (19):

- radv: apply xfb buffer offset at buffer binding time not later. (v2)
- radv: fix begin/end transform feedback with 0 counter buffers.
- virgl: fix vtest regression since fencing changes.
- spirv/vtn: handle variable pointers without offset lowering
- nir: move getting deref from var after we check deref type.
- nir: handle shared pointers in lowering indirect derefs.
- ac: avoid casting pointers on bcsel and stores
- radv: handle loading from shared pointers
- ac: handle cast derefs
- r600: make suballocator 256-bytes align
- virgl: fix undefined shift to use unsigned.
- virgl: fix const warning on debug flags.
- radv: use 3d shader for gfx9 copies if dst is 3d
- radv/xfb: fix counter buffer bounds checks.
- virgl/vtest: fix front buffer flush with protocol version 0.

- virgl: use primconvert provoking vertex properly
- dri_interface: add put shm image2 (v2)
- glx: add support for putimageshm2 path (v2)
- gallium: use put image shm2 path (v2)

David Shao (1):

- meson: ensure that xmlpool_options.h is generated for gallium targets that need it

Dieter Nützel (1):

- docs/features: Delete double nv50 entry and wrong enumeration

Dylan Baker (48):

- meson: link gallium nine with pthreads
- meson: Don't set -Wall
- meson: fix libatomic tests
- meson: Add tests to suites
- util: promote u_memory to src/util
- meson: Add nir_algebraic_parser_test to suites
- meson: Fix ppc64 little endian detection
- meson: remove duplicate definition
- meson: Add support for gnu hurd
- meson: Add toggle for glx-direct
- docs/meson: Recommend not using CFLAGS and friends
- travis: meson: use native files to override llvm-config
- travis: Don't try to read libdrm out of configure.ac
- travis: meson: enable unit tests
- docs: add note about using backticks for rbs in gitlab
- docs/install: Add meson to the main install page
- docs/meson: Update LLVM section with information about native files
- docs/install: Update python dependency section
- docs/autoconf: Mark autoconf as being replaced
- meson: Override C++ standard to gnu++11 when building with altivec on ppc64
- meson: Error out if building nouveau and using LLVM without rtti
- autotools: Remove tegra vdpau driver
- meson: Add a script to extract the cmd line used for meson
- meson: allow building dri driver without window system if osmesa is classic
- bin/meson-cmd-extract: Also handle cross and native files
- meson: fix swr KNL build
- meson: Fix compiler checks for SWR with ICC

- meson: Add warnings and errors when using ICC
- automake: Fix path to generated source
- automake: Add float64.glsl to dist tarball
- automake: Add include dir for nir src directory
- configure: Bump SWR LLVM requirement to 7
- automake: Add –enable-autotools to distcheck flags
- android,autotools,i965: Fix location of float64_glsl.h
- VERSION: bump to 19.0.0-rc1
- Version: Bump for rc2
- cherry-ignore: Add some patches
- Revert "intel/compiler: More peephole_select for pre-Gen6"
- Revert "nir/opt_peephole_select: Don't peephole_select expensive math instructions"
- Revert "intel/compiler: More peephole select"
- Bump version for 19.0-rc3
- version: bump for 19.0-rc4
- get-pick-list: Add –pretty=medium to the arguments for Cc patches
- meson: Add dependency on genxml to anvil
- Version: update to 19.0-rc5
- Bump version for rc6
- VERSION: bump version for rc7
- cherry-ignore: Update the cherry-ignore file

Eduardo Lima Mitev (2):

- freedreno/ir3: Make imageStore use num components from image format
- freedreno/ir3: Handle GL_NONE in get_num_components_for_glformat()

Eleni Maria Stea (1):

- i965: fixed clamping in set_scissor_bits when the y is flipped

Elie Tournier (17):

- glsl: Add "built-in" function to do abs(fp64)
- glsl: Add "built-in" functions to do neg(fp64)
- glsl: Add "built-in" function to do sign(fp64)
- glsl: Add "built-in" functions to do eq/ne(fp64, fp64)
- glsl: Add utility function to extract 64-bit sign
- glsl: Add "built-in" functions to do lt(fp64, fp64)
- glsl: Add "built-in" functions to do add(fp64, fp64)
- glsl: Add "built-in" functions to do mul(fp64, fp64)
- glsl: Add "built-in" functions to do fp64_to_uint(fp64)

- glsl: Add “built-in” functions to do uint_to_fp64(uint)
- glsl: Add “built-in” functions to do fp64_to_int(fp64)
- glsl: Add “built-in” functions to do int_to_fp64(int)
- glsl: Add “built-in” functions to do fp64_to_fp32(fp64)
- glsl: Add “built-in” functions to do fp32_to_fp64(fp32)
- glsl: Add “built-in” functions to do sqrt(fp64)
- glsl: Add “built-in” functions to do trunc(fp64)
- glsl: Add “built-in” functions to do round(fp64)

Emil Velikov (81):

- mesa: bump version to 19.1.0-devel
- docs: add 19.0.0-devel release notes template
- docs: mention EXT_shader_implicit_conversions
- egl: add EGL_EXT_device_base entrypoints
- egl/glvnd: correctly report errors when vendor cannot be found
- docs/releasing.html: polish cherry-picking/testing text
- docs/submittingpatches.html: correctly handle the <p> tag
- docs: document the staging branch and add reference to it
- bin/get-pick-list.sh: simplify git oneline printing
- bin/get-pick-list.sh: prefix output with “[stable] “
- bin/get-pick-list.sh: handle “typod” usecase.
- bin/get-pick-list.sh: handle the fixes tag
- bin/get-pick-list.sh: tweak the commit sha matching pattern
- bin/get-pick-list.sh: flesh out is_sha_nomination
- bin/get-pick-list.sh: handle fixes tag with missing colon
- bin/get-pick-list.sh: handle unofficial “broken by” tag
- bin/get-pick-list.sh: use test instead of [ ]
- bin/get-pick-list.sh: handle reverts prior to the branchpoint
- travis: drop unneeded x11proto-xf86vidmode-dev
- glx: make xf86vidmode mandatory for direct rendering
- travis: adding missing x11-xcb for meson+vulkan
- egl/wayland: bail out when drmGetMagic fails
- egl/wayland: plug memory leak in drm_handle_device()
- docs: update 18.3.0 release notes
- docs: add sha256 checksums for 18.3.0
- docs: update calendar, add news item and link release notes for 18.3.0
- freedreno: drop duplicate MKDIR_GEN declaration

- freedreno: add the missing _la in libfreedreno_ir3_la
- amd/addrlib: drop si_ci_vi_merged_enum.h from the list
- docs: add release notes for 18.3.1
- docs: add sha256 checksums for 18.3.1
- docs: update calendar, add news item and link release notes for 18.3.1
- glx: mandate xf86vidmode only for "drm" dri platforms
- bin/get-pick-list.sh: rework handing of sha nominations
- bin/get-pick-list.sh: warn when commit lists invalid sha
- meson: don't require glx/egl/gbm with gallium drivers
- pipe-loader: meson: reference correct library
- TODO: glx: meson: build dri based glx tests, only with -Dglx=dri
- glx: meson: drop includes from a link-only library
- glx: meson: wire up the dispatch-index-check test
- glx/test: meson: assorted include fixes
- configure: add CXX11_CXXFLAGS to LLVM_CXXFLAGS
- travis: flip to distro xenial, drop sudo false
- travis: meson: print the configured state
- travis: printout llvm-config –version
- travis: meson: use FOO_DRIVERS directly
- travis: meson: add unwind handling
- travis: meson: explicitly control the DRI loaders
- travis: meson: add explicit handling to gallium ST
- travis: meson: port gallium build combinations over
- docs: add release notes for 18.3.2
- docs: add sha256 checksums for 18.3.2
- docs: update calendar, add news item and link release notes for 18.3.2
- freedreno: automake: ship ir3_nir_trig.py in the tarball
- mesa: correctly use os.path.join in our python scripts
- Revert "mesa/main: remove ARB suffix from glGetnTexImage"
- mapi: sort static entrypoints numerically
- mapi: add all _glapi_table entrypoints to static_data.py
- genCommon.py: Fix typo in _LIBRARY_FEATURE_NAMES.
- mapi: move genCommon.py to src/mapi/new
- mapi/new: import mapi scripts from glvnd
- mapi/new: sort by slot number
- mapi/new: use the static_data offsets in the new generator

- mapi/new: reinstate _NO_HIDDEN suffixes in the new generator
- mapi/new: split out public_entries handling
- mapi/new: don't print info we don't need for ES1/ES2
- mapi/new: fixup the GLDEBUGPROCKHR typedef to the non KHR one
- mapi/new: remove duplicate GLvoid/void substitution
- autotools: wire the new generator for es1 and es2
- meson: wire the new generator for es1 and es2
- scons: wire the new generator for es1 and es2
- Revert "mapi/new: sort by slot number"
- mapi/es*api: remove GL_OES_EGL_image entrypoints
- mapi/es*api: remove GL_EXT_multi_draw_arrays entrypoints
- mapi/es2api: remove no longer present entrypoints
- mapi: remove old, unused ES* generator code
- mapi: remove machinery handling CSV files
- mapi: print function declarations for shared glapi
- vc4: Declare the last cpu pointer as being modified in NEON asm.
- anv: wire up the state_pool_padding test
- meson: egl: correctly manage loader/xmlconfig

Eric Anholt (171):

- v3d: Fix a copy-and-paste comment in the simulator code.
- v3d: Fix a typo in a comment in job handling.
- v3d: Drop #if 0-ed out v3d_dump_to_file().
- v3d: Respect user-passed strides for BO imports.
- v3d: Take advantage of _mesa_hash_table_remove_key() in the simulator.
- v3d: Use the TLB R/B swapping instead of recompiles when available.
- v3d: Update the TLB config for depth writes on V3D 4.2.
- vc4: Drop the winsys_stride relayout in the simluator
- v3d: Maintain a mapping of the GEM buffer in the simulator.
- v3d: Remove the special path for simulaton of the submit ioctl.
- vc4: Take advantage of _mesa_hash_table_remove_key() in the simulator.
- vc4: Maintain a separate GEM mapping of BOs in the simulator.
- vc4: Use the normal simulator ioctl path for CL submit as well.
- gbm: Move gbm_format_canonicalize() to the core.
- gbm: Introduce a helper function for printing GBM format names.
- egl: Improve the debugging of gbm format matching in DRI configs.
- v3d: Fix double-swapping of R/B on V3D 4.1

- v3d: Don't try to set PF flags on a LDTMU operation
- vc4: Make sure we make ro scanout resources for create_with_modifiers.
- vc4: Don't return a vc4 BO handle on a renderonly screen.
- glx: Remove an old DEFAULT_DRIVER_DIR default.
- glx: Move DRI extensions pointer loading to driOpenDriver().
- egl: Move loader_set_logger() up to egl_dri2.c.
- loader: Stop using a local definition for an in-tree header
- loader: Factor out the common driver opening logic from each loader.
- egl: Print the actual message to the console from _eglError().
- gallium: Fix uninitialized variable warning in compute test.
- gallium: Remove unused variable in u_tests.
- v3d: Add renderonly support.
- v3d: Add support for RGBA_SRGB along with BGRA_SRGB.
- v3d: Add missing OES_half_float_linear support.
- v3d: Use combined input/output segments.
- v3d: Add the V3D TFU submit interface to the simulator.
- v3d: Use the TFU to do generatemipmap.
- v3d: Update simulator cache flushing code to match the kernel better.
- v3d: Create a state uploader for packing our shaders together.
- v3d: Put default vertex attribute values into the state uploader as well.
- v3d: Re-use the wrap mode uniform on V3D 3.3.
- v3d: Make an array for frag/vert texture state in the context.
- v3d: Don't forget to flush writes to UBOs.
- v3d: Convert to using nir_src_as_uint() from const_value derefs.
- v3d: Fix a comment typo
- v3d: Return the right gl_SampleMaskIn[] value.
- v3d: Fix handling of texture first_layer offsets for 3D textures.
- v3d: Avoid confusing auto-indenting in TEXTURE_SHADER_STATE packing
- v3d: Split most of TEXTURE_SHADER_STATE setup out of sampler views.
- v3d: Garbage collect unused uniforms code.
- v3d: Simplify VIR uniform dumping using a temporary.
- v3d: Add VIR dumping of TMU config p0/p1.
- v3d: Fix a leak of the transfer helper on screen destroy.
- vc4: Fix a leak of the transfer helper on screen destroy.
- v3d: Fix a leak of the disassembled instruction string during debug dumps.
- tfu

- shader-packing
- nir: Add some more consts to the nir_format_convert.h helpers.
- nir: Pull some of intel's image load/store format conversion to nir_format.h
- intel: Simplify the half-float packing in image load/store lowering.
- mesa/st: Expose compute shaders when NIR support is advertised.
- nir: Print the format of image variables.
- Revert "intel: Simplify the half-float packing in image load/store lowering."
- nir: Move intel's half-float image store lowering to to nir_format.h.
- v3d: Don't forget to wait for our TFU job before rendering from it.
- v3d: Set up the right stride for raster TFU.
- v3d: Don't forget to bump the number of writes when doing TFU ops.
- v3d: Add support for using the TFU to do some blits.
- v3d: Add support for texturing from linear.
- v3d: Add safety checks for resource_create().
- v3d: Make sure that a thrsw doesn't split a multop from its umul24.
- v3d: Add missing flagging of SYNCB as a TSY op.
- v3d: Add support for draw indirect for GLES3.1.
- v3d: Avoid assertion failures when removing end-of-shader instructions.
- v3d: Move uinfo->data[] dereference to the top of v3d_write_uniforms().
- v3d: Move uniform pretty-printing to its own helper function.
- v3d: Use the uniform pretty-printer in v3d_write_uniforms()'s debug code.
- v3d: Do uniform pretty-printing in the QPU dump.
- v3d: Drop in a bunch of notes about performance improvement opportunities.
- vc4: Use the original bit size when scalarizing uniform loads.
- v3d: Use the original bit size when scalarizing uniform loads.
- vc4: Reuse nir_format_convert.h in our blend lowering.
- v3d: Fix the argument type for vir_BRANCH().
- nir: Fix clamping of uints for image store lowering.
- v3d: Put the dst bo first in the list of BOs for TFU calls.
- v3d: Fix check for TFU job completion in the simulator.
- v3d: Don't try to create shadow tiled temporaries for 1D textures.
- v3d: Remove dead prototypes for load/store utile functions.
- v3d: Implement texture_subdata to reduce teximage upload copies.
- vc4: Move the utile load/store functions to a header for reuse by v3d.
- v3d: Add a fallthrough path for utile load/store of 32 byte lines.
- v3d: Load and store aligned utiles all at once.

- docs: Add a note that MRs should still include any r-b or a-b tags.
- docs: Add an encouraging note about providing reviews and acks.
- v3d: Fix simulator mode on i915 render nodes.
- v3d: Drop shadow comparison state from shader variant key.
- v3d: Hook up perf_debug() output to GL_ARB_debug output as well.
- vc4: Hook up perf_debug() output to GL_ARB_debug_output as well.
- gallium/ttn: Fix setup of outputs_written.
- v3d: Fix uniform pretty printing assertion failure with branches.
- v3d: Add a "precompile" debug flag for shader-db.
- v3d: Hook up some shader-db output to GL_ARB_debug_output.
- v3d: Drop unused count_nir_instrs() helper.
- v3d: Drop incorrect dependency for flpop.
- v3d: Move "does this instruction have flags" from sched to generic helpers.
- v3d: Don't generate temps for comparisons.
- v3d: Dead-code eliminate unused flags updates.
- v3d: Add a note for a potential performance win on multop/umul24.
- v3d: Force sampling from base level for tg4.
- v3d: Add support for non-constant texture offsets.
- v3d: Add support for requesting the sample offsets.
- v3d: Add support for textureSize() on MSAA textures.
- v3d: Add support for gl_HelperInvocation.
- v3d: Fix segfault when failing to compile a program.
- v3d: Don't forget to include RT writes in precompiles.
- v3d: Simplify the emission of comparisons for the bcsel optimization.
- v3d: Move the "Find the ALU instruction generating our bool" out of bcsel.
- v3d: Don't try to fold non-SSA-src comparisons into bcsels.
- v3d: Fold comparisons for IF conditions into the flags for the IF.
- v3d: Handle dynamically uniform IF statements with uniform control flow.
- v3d: Refactor compiler entrypoints.
- v3d: Reinstate the new shader-db output after v3d_compile() refactor.
- v3d: Fix up VS output setup during precompiles.
- v3d: Remove dead switch cases and comments from v3d_nir_lower_io.
- v3d: Do UBO loads a vector at a time.
- v3d: Stop scalarizing our uniform loads.
- nir: Allow nir_format_unpack_int/sint to unpack larger values.
- nir: Add nir_lower_tex options to lower sampler return formats.

- v3d: Use the core tex lowering.
- nir: Add nir_lower_tex support for Broadcom's swizzled TG4 results.
- v3d: Enable GL_ARB_texture_gather on V3D 4.x.
- nir: Make nir_deref_instr_build/get_const_offset actually use size_align.
- glsl: Fix buffer overflow with an atomic buffer binding out of range.
- v3d: Add support for flushing dirty TMU data at job end.
- v3d: Add support for the early_fragment_tests flag.
- v3d: Add support for GL_ARB_framebuffer_no_attachments.
- v3d: Fix txf_ms 2D_ARRAY array index.
- v3d: Add an isr to the simulator to catch GMP violations.
- v3d: Add support for matrix inputs to the FS.
- v3d: Drop the GLSL version level.
- v3d: Add SSBO/atomic counters support.
- v3d: Add support for shader_image_load_store.
- v3d: Add support for CS workgroup/invocation id intrinsics.
- v3d: Add support for CS shared variable load/store/atomics.
- v3d: Add support for CS barrier() intrinsics.
- v3d: SHARED but not necessarily SCANOUT buffers on RO must be linear.
- v3d: If the modifier is not known on BO import, default to linear for RO.
- v3d: Restructure RO allocations using resource_from_handle.
- v3d: Don't leak the GPU fd for renderonly usage.
- vc4: Don't leak the GPU fd for renderonly usage.
- gallium: Enable unit tests as actual meson unit tests.
- gallium: Fix comment about possible colorspaces.
- gallium: Make sure we return is_unorm/is_snorm for compressed formats.
- v3d: Rename gallium-local limits defines from VC5 to V3D.
- v3d: Fix overly-large vattr_sizes structs.
- v3d: Avoid duplicating limits defines between gallium and v3d core.
- v3d: Drop maximum number of texture units down to 16.
- v3d: Fix BO stats accounting for imported buffers.
- v3d: Flush blit jobs immediately after generating them.
- v3d: Fix release-build warning about utile_h.
- v3d: Fix stencil sampling from packed depth/stencil.
- v3d: Fix stencil sampling from a separate-stencil buffer.
- v3d: Use the symbolic names for wrap modes from the XML.
- v3d: Move the sampler state to the long-lived state uploader.

- v3d: Create separate sampler states for the various blend formats.
- pl111: Rename the pl111 driver to "kmsro".
- kmsro: Extend to include hx8357d.
- vc4: Enable NEON asm on meson cross-builds.
- v3d: Fix the autotools build.
- mesa: Skip partial InvalidateFramebuffer of packed depth/stencil.
- v3d: Fix image_load_store clamping of signed integer stores.
- v3d: Use the early_fragment_tests flag for the shader's disable-EZ field.
- v3d: Fix the check for "is the last thrsw inside control flow"
- st/dri: Set the PIPE_BIND_SHARED flag on create_image_with_modifiers.

Eric Engestrom (47):

- wsi/wayland: use proper VkResult type
- wsi/wayland: only finish() a successfully init()ed display
- REVIEWERS: add include path for EGL
- REVIEWERS: add Emil as EGL reviewer
- REVIEWERS: add Vulkan reviewer group
- xmlpool: update translation po files
- meson: only run vulkan's meson.build when building vulkan
- gbm: remove unnecessary meson include
- meson: fix wayland-less builds
- gbm: add new entrypoint to symbols check
- egl: add missing glvnd entrypoint for EGL_ANDROID_blob_cache
- egl: fix bad rebase
- gbm: add missing comma between strings
- glapi: add missing visibility args
- anv: correctly use vulkan 1.0 by default
- vulkan/utils: s/VERSION/PACKAGE_VERSION/
- build: stop defining unused VERSION
- wsi/display: fix mem leak when freeing swapchains
- vulkan/wsi: fix s/,/;/ typo
- meson: skip asm check when asm is disabled
- anv: add unreachable() for VK_EXT_fragment_density_map
- mesa: drop unused & deprecated lib
- loader: deduplicate logger function declaration
- docs: add meson cross compilation instructions
- docs: format code blocks a bit nicely

- docs: fix the meson aarch64 cross-file
- docs: advertise distro-provided meson cross-files
- anv: drop unneeded KHR suffix
- wsi: drop unneeded KHR suffix
- radv: remove a few more unnecessary KHR suffixes
- egl: add missing includes
- egl: remove unused include
- travis: avoid using unset llvm-config
- egl: fix python lib deprecation warning
- docs: explain how to see what meson options exist
- travis: fix autotools build after –enable-autotools switch addition
- configure: EGL requirements only apply if EGL is built
- egl: finalize EGL_MESA_query_driver
- egl: update headers from Khronos
- egl: add glvnd entrypoints for EGL_MESA_query_driver
- travis: bump libdrm to 2.4.97
- egl/glvnd: sync egl.xml from Khronos
- anv: drop always-successful VkResult
- meson/vdpau: add missing soversion
- xvmc: fix string comparison
- xvmc: fix string comparison
- egl: fix libdrm-less builds

Erik Faye-Lund (70):

- glsl: add has_implicit_conversions()-helper
- glsl: add has_implicit_uint_to_int_conversion()-helper
- glsl: fall back to inexact function-match
- mesa/glsl: add support for EXT_shader_implicit_conversions
- glsl: do not allow implicit casts of unsized array initializers
- mesa: expose NV_conditional_render on GLES
- mesa/main: fixup make check after NV_conditional_render for gles
- Revert “mesa/main: fixup make check after NV_conditional_render for gles”
- Revert “mesa: expose NV_conditional_render on GLES”
- mesa/main: correct requirement for EXT_occlusion_query_boolean
- mesa/main: correct year for EXT_occlusion_query_boolean
- mesa/main: use non-prefixed enums for consistency
- mesa/main: simplify pipeline-statistics query validation

- mesa/main: fix validation of GL_SAMPLES_PASSED
- mesa/main: fix validation of GL_ANY_SAMPLES_PASSED
- mesa/main: fix validation of GL_ANY_SAMPLES_PASSED_CONSERVATIVE
- mesa/main: fix validation of GL_TIME_ELAPSED
- mesa/main: fix validation of transform-feedback queries
- mesa/main: fix validation of transform-feedback overflow queries
- mesa/main: fix validation of ARB_query_buffer_object
- mesa/main: fix validation of GL_TIMESTAMP
- mesa/main: remove overly strict query-validation
- mesa/main: remove ARB suffix from glGetnTexImage
- mesa/main: remove bogus error for zero-sized images
- mesa/main: factor out tex-image error-checking
- mesa/main: factor out common error-checking
- mesa/main: check cube-completeness in common code
- mesa/main: fix incorrect depth-error
- mesa/main: fixup requirements for GL_PRIMITIVES_GENERATED
- mesa/main: make _mesa_has_tessellation return bool
- mesa/main: rename format-check function
- mesa/main: clean up S3_s3tc check
- mesa/main: clean up OES_texture_float_linear check
- mesa/main: clean up ES2_compatibility check
- mesa/main: clean up integer texture check
- mesa/main: use _mesa_has_FOO_bar for compressed format checks
- mesa/main: do not allow s3tc enums on gles1
- mesa/main: do not allow etc2 enums on gles1
- mesa/main: do not allow astc enums on gles1
- mesa/main: do not allow depth-texture enums on gles1
- mesa/main: do not allow stencil-texture enums on gles1
- mesa/main: do not allow ARB_texture_rgb10_a2ui enums before gles3
- mesa/main: do not allow integer-texture enums before gles3
- mesa/main: do not allow ARB_depth_buffer_float enums before gles3
- mesa/main: do not allow EXT_packed_float enums before gles3
- mesa/main: do not allow rg-textures enums before gles3
- mesa/main: do not allow EXT_texture_shared_exponent enums before gles3
- mesa/main: do not allow MESA_ycbcr_texture enums on gles
- mesa/main: do not allow type_2_10_10_10_REV enums before gles3

- mesa/main: do not allow floating-point texture enums on gles1
- mesa/main: do not allow snorm-texture enums before gles3
- mesa/main: do not allow sRGB texture enums before gles3
- mesa/main: do not allow EXT_texture_sRGB_R8 enums before gles3
- mesa/main: split float-texture support checking in two
- mesa/main: require EXT_texture_type_2_10_10_10_REV for gles3
- mesa/main: require EXT_texture_sRGB for gles3
- mesa/st: do not probe for the same texture-formats twice
- mesa/main: do not require float-texture filtering for es3
- mesa/main: correct validation for GL_RGB565
- mesa/main: fix up _mesa_has_rg_textures for gles2
- virgl: force linear texturing support
- virgl: simplify virgl_hw_set_vertex_buffers
- virgl: simplify virgl_hw_set_index_buffer
- virgl: wrap vertex element state in a struct
- virgl: work around bad assumptions in virglrenderer
- anv/meson: make sure tests link with -msse2
- anv/autotools: make sure tests link with -msse2
- docs: add note about sending merge-requests from forks
- mapi: drop unneeded gl_dispatch_stub declarations
- virgl: remove unused variable

Ernestas Kulik (2):

- vc4: Fix leak in HW queries error path
- v3d: Fix leak in resource setup error path

Francisco Jerez (14):

- intel/fs: Prevent emission of IR instructions not aligned to their own execution size.
- intel/fs: Handle source modifiers in lower_integer_multiplication().
- intel/fs: Implement quad swizzles on ICL+.
- intel/fs: Fix bug in lower_simd_width while splitting an instruction which was already split.
- intel/eu/gen7: Fix brw_MOV() with DF destination and strided source.
- intel/fs: Respect CHV/BXT regioning restrictions in copy propagation pass.
- intel/fs: Constify fs_inst::can_do_source_mods().
- intel/fs: Introduce regioning lowering pass.
- intel/fs: Remove existing lower_conversions pass.
- intel/fs: Remove nasty open-coded CHV/BXT 64-bit workarounds.
- intel/fs: Remove FS_OPCODE_UNPACK_HALF_2x16_SPLIT opcodes.

- intel/fs: Promote execution type to 32-bit when any half-float conversion is needed.
- intel/fs: Exclude control sources from execution type and region alignment calculations.
- intel/fs: Implement extended strides greater than 4 for IR source regions.

Fritz Koenig (2):

- freedreno: drm_fourcc.h header include
- freedreno: add query for dmabuf modifiers

Gert Wollny (30):

- mesa/core: Add definitions and translations for EXT_texture_sRGB_R8
- Gallium: Add format PIPE_FORMAT_R8_SRGB
- mesa/st: Add support for EXT_texture_sRGB_R8
- virgl/vtest-winsys: Use virgl version of bind flags
- r600: Add support for EXT_texture_sRGB_R8
- mesa: Reference count shaders that are used by transform feedback objects
- virgl: Add command and flags to initiate debugging on the host (v2)
- nir: Allow to skip integer ops in nir_lower_to_source_mods
- i965: Correct L8_UNORM_SRGB table entry
- i965: be more specific about FBO completeness errors
- i965: Force zero swizzles for unused components in GL_RED and GL_RG
- i965: Add support for and expose EXT_texture_sRGB_R8
- virgl: Use file descriptor instead of un-allocated object
- i965:use FRAMEBUFFER_UNSUPPORTED instead of FRAMEBUFFER_INCOMPLETE_DIMENSIONS
- r600: Only set context streamout strides info from the shader that has outputs
- r600: clean up the GS ring buffers when the context is destroyed
- glsl: free or reuse memory allocated for TF varying
- virgl,vtest: Initialize return value
- virgl: Don't try handling server fences when they are not supported
- i965: Explicitely handle swizzles for MESA_FORMAT_R_SRGB8
- i965: Set the FBO error state INCOMPLETE_ATTACHMENT only for SRGB_R8
- autotools: Deprecate the use of autotools
- Gallium: Add new CAPS to indicate whether a driver can switch SRGB write
- virgl: Set sRGB write control CAP based on host capabilities
- mesa:main: Add flag for EXT_sRGB to gl_extensions
- i965: Set flag for EXT_sRGB
- mesa/st: rework support for sRGB framebuffer attachements
- mesa/main: Use flag for EXT_sRGB instead of EXT_framebuffer_sRGB where possible
- mesa/main/version: Lower the requirements for GLES 3.0

- mesa/main: Expose EXT_sRGB_write_control

Guido Günther (2):

- etnaviv: Make sure rs alignment checks match
- etnaviv: fix typo in cflush_all description

Gurchetan Singh (18):

- egl: add missing #include <stddef.h> in egldevice.h
- virgl: quadruple command buffer size
- virgl: avoid large inline transfers
- virgl: don't mark buffers as unclean after a write
- virgl: texture_transfer_pool –> transfer_pool
- virgl: remove unnessecary code
- virgl: move texture metadata to common code
- virgl: move virgl_resource_layout to common code
- virgl: move vrend_get_tex_image_offset to common code
- virgl: store layer_stride in metadata
- virgl: consolidate transfer code
- virgl: make transfer code with PIPE_BUFFER targets
- virgl: make virgl_buffers use resource helpers
- virgl: modify how we handle GL_MAP_FLUSH_EXPLICIT_BIT
- virgl: move resource metadata into base resource
- virgl: move resource creation / import / destruction to common code
- virgl: don't flush an empty range
- virgl: remove empty file

Hanno Böck (1):

- glsl/test: Fix use after free in test_optpass.

Hyunjun Ko (1):

- freedreno: implements get_sample_position

Iago Toral Quiroga (22):

- intel/compiler: fix node interference of simd16 instructions
- nir/constant_folding: fix incorrect bit-size check
- nir/from_ssa: fix bit-size of temporary register
- Revert "nir/builder: Assert that intN_t immediates fit"
- intel/compiler: fix indentation style in opt_algebraic()
- intel/compiler: fix register allocation in opt_peephole_sel
- intel/compiler: do not copy-propagate strided regions to ddx/ddy arguments
- intel/compiler: move nir_lower_bool_to_int32 before nir_lower_locals_to_regs

- compiler/nir: add a nir_b2f() helper
- compiler/nir: add nir_fadd_imm() and nir_fmul_imm() helpers
- compiler/spirv: handle 16-bit float in radians() and degrees()
- compiler/spirv: implement 16-bit asin
- compiler/spirv: implement 16-bit acos
- compiler/spirv: implement 16-bit atan
- compiler/spirv: implement 16-bit atan2
- compiler/spirv: implement 16-bit exp and log
- compiler/spirv: implement 16-bit hyperbolic trigonometric functions
- compiler/spirv: implement 16-bit frexp
- compiler/spirv: use 32-bit polynomial approximation for 16-bit asin()
- anv/pipeline_cache: fix incorrect guards for NIR cache
- anv/pipeline_cache: free NIR shader cache
- anv/device: fix maximum number of images supported

Ian Romanick (28):

- glsl: Add warning tests for identifiers with __
- glsl: Add pragma to disable all warnings
- glsl: prevent qualifiers modification of predeclared variables
- glsl: Omit redundant qualifier checks on redeclarations
- glsl: Refactor type checking for redeclarations
- nir: Add a saturated unsigned integer add opcode
- i965/fs: Implement nir_op_uadd_sat
- nir/phi_builder: Internal users should use nir_phi_builder_value_set_block_def too
- util/slab: Rename slab_mempool typed parameters to mempool
- util/hash_table: Add _mesa_hash_table_init function
- nir/phi_builder: Use per-value hash table to store [block] -> def mapping
- nir: Fix holes in nir_instr
- nir: Release per-block metadata in nir_sweep
- i965/vec4: Silence unused parameter warnings in vec4 compiler tests
- i965/vec4/dce: Don't narrow the write mask if the flags are used
- i965/fs: Eliminate unary op on operand of compare-with-zero
- i965/vec4: Propagate conditional modifiers from more compares to other compares
- nir/opt_peephole_select: Don't try to remove flow control around indirect loads
- intel/compiler: More peephole select
- nir/opt_peephole_select: Don't peephole_select expensive math instructions
- intel/compiler: More peephole_select for pre-Gen6

- Revert "nir/lower_indirect: Bail early if modes == 0"
- nir/algebraic: Don't put quotes around floating point literals
- glsl: Add utility to convert text files to C strings
- nir: Silence zillions of unused parameter warnings in release builds
- spirv: Add missing break
- intel/fs: nir_op_extract_i8 extracts a byte, not a word
- intel/fs: Fix extract_u8 of an odd byte from a 64-bit integer

Ilia Mirkin (37):

- nv50/ir: delete MINMAX instruction that is no longer in the BB
- nv50/ir/ra: improve condition for short regs, unify with cond for 16-bit
- nv50/ir/ra: enforce max register requirement, and change spill order
- nv50/ir: remove dnz flag when converting MAD to ADD due to optimizations
- nv50: always keep TSC slot 0 bound
- nv50,nvc0: add explicit handling of PIPE_CAP_MAX_VERTEX_ELEMENT_SRC_OFFSET
- nouveau: set texture upload budget
- nvc0: replace use of explicit default_tsc with entry 0
- nvc0: always keep TSC slot 0 bound to fix TXF
- st/mesa: remove sampler associated with buffer texture in pbo logic
- st/mesa: allow glDrawElements to work with GL_SELECT feedback
- tgsi: add ATOMFADD operation
- gallium: add PIPE_CAP_TGSI_ATOMFADD to indicate support
- st/mesa: select ATOMFADD when source type is float
- st/mesa: expose GL_NV_shader_atomic_float when ATOMFADD is supported
- nv50/ir: add support for converting ATOMFADD to proper ir
- nvc0: enable GL_NV_shader_atomic_float on pre-Maxwell
- nv50,nvc0: add missing CAPs for unsupported features
- nv30: avoid setting user_priv without setting cur_ctx
- nv30: fix rare issue with fp unbinding not finding the bufctx
- nv30: add support for multi-layer transfers
- nv30: use correct helper to get blocks in y direction
- nv30: fix some s3tc layout issues
- nv30: disable rendering to 3D textures
- docs: fix gallium screen cap docs
- nv50,nvc0: mark textures dirty on fb update
- nvc0: don't put text segment into bufctx
- nvc0/ir: fix second tex argument after levelZero optimization

- nv50,nvc0: add explicit settings for recent caps
- nvc0: add support for handling indirect draws with attrib conversion
- nvc0/ir: always use CG mode for loads from atomic-only buffers
- nvc0: fix 3d images on kepler
- nv50,nvc0: use condition for occlusion queries when already complete
- nvc0: stick zero values for the compute invocation counts
- nvc0: we have 16k-sized framebuffers, fix default scissors
- swr: set PIPE_CAP_MAX_VARYINGS correctly
- glsl: fix recording of variables for XFB in TCS shaders

Indrajit Das (1):

- st/va: Return correct status from vlVaQuerySurfaceStatus

Jakob Bornecrantz (1):

- virgl/vtest: Use default socket name from protocol header

Jan Vesely (2):

- amd: Make vgpr-spilling depend on llvm version
- clover: Fix build after clang r348827

Jason Ekstrand (207):

- vulkan: Update the XML and headers to 1.1.91
- intel/fs,vec4: Clean up a repeated pattern with SSBOs
- intel/fs: Use the new nir_src_is_const and friends
- nir: Add a read_mask helper for ALU instructions
- intel/vec4: Use the new nir_src_is_const and friends
- intel/analyze_ubo_ranges: Use nir_src_is_const and friends
- anv: Use nir_src_is_const and friends in lowering code
- intel/fs: Add an assert to optimize_frontfacing_ternary
- nir/lower_alu_to_scalar: Don't try to lower unpack_32_2x16
- nir/builder: Assert that intN_t immediates fit
- nir/builder: Add iadd_imm and imul_imm helpers
- nir/builder: Add a nir_pack/unpack/bitcast helpers
- nir/spirv: Force 32-bit for UBO and SSBO Booleans
- nir/glsl: Force 32-bit for UBO and SSBO Booleans
- nir/lower_io: Add shared to get_io_offset_src
- nir: Add alignment parameters to SSBO, UBO, and shared access
- intel/compiler: Lower SSBO and shared loads/stores in NIR
- intel,nir: Move gl_LocalInvocationID lowering to nir_lower_system_values
- intel/fs,vec4: Fix a compiler warning

- vulkan: Update the XML and headers to 1.1.93
- anv: Expose VK_EXT_scalar_block_layout
- anv: Put robust buffer access in the pipeline hash
- anv/nir: Rework arguments to apply_pipeline_layout
- nir/derefs: Add a nir_derefs_do_not_alias enum value
- vulkan: Update the XML and headers to 1.1.95
- nir/opcodes: Pull in the type helpers from constant_expressions
- nir/opcodes: Rename tbool to tbool32
- nir/algebraic: Clean up some __str__ cruft
- nir/algebraic: Refactor codegen a bit
- nir/algebraic: Add support for unsized conversion opcodes
- nir/opt_algebraic: Simplify an optimization using the new search ops
- nir/opt_algebraic: Drop bit-size suffixes from conversions
- nir/opt_algebraic: Add 32-bit specifiers to a bunch of booleans
- nir: Make boolean conversions sized just like the others
- anv,radv: Disable VK_EXT_pci_bus_info
- intel/ir: Don't allow allocating zero registers
- spirv: Add support for MinLod
- nir/lower_tex: Simplify lower_gradient logic
- nir/lower_tex: Modify txd instructions instead of replacing them
- nir/lower_tex: Add lowering for some min_lod cases
- intel/fs: Support min_lod parameters on texture instructions
- anv: Advertise support for MinLod on Skylake+
- anv/pipeline: Set the correct binding count for compute shaders
- intel/blorp: Assert that we don't re-layout a compressed surface
- nir: Document the function inlining process
- nir: Allow [iu]mul_high on non-32-bit types
- nir/lower_int64: Add support for [iu]mul_high
- nir: Add a pass for lowering integer division by constants
- i965/vec4: Implement nir_op_uadd_sat
- i965: Enable nir_opt_idiv_const for 32 and 64-bit integers
- nir/lower_idiv: Use ilt instead of bit twiddling
- nir/tgsi: Use nir_bany in ttn_kill_if
- nir/constant_folding: Fix source bit size logic
- nir/algebraic: Optimize x2b(xneg(a)) -> a
- nir: Drop support for lower_b2f

- nir/algebraic: Make an optimization more specific
- nir: Rename Boolean-related opcodes to include 32 in the name
- nir/constant_expressions: Rework Boolean handling
- nir: Add support for 1-bit data types
- nir/large_constants: Properly handle 1-bit bools
- nir/algebraic: Generalize an optimization
- nir: Add 1-bit Boolean opcodes
- nir: Add a bool to int32 lowering pass
- nir: Switch to using 1-bit Booleans for almost everything
- nir/algebraic: Optimize 1-bit Booleans
- nir/algebraic: Add some optimizations for D3D-style Booleans
- radv: Fix a stupid if in gather_intrinsic_info
- st/nir: Use nir_src_as_uint for tokens
- vulkan: Update the XML and headers to 1.1.96
- anv,radv: Re-enable VK_EXT_pci_bus_info
- anv: Bump the patch version to 96
- nir/propagate_invariant: Skip unknown vars
- nir/linking_helpers: Look at derefs for modes
- nir/lower_io_arrays_to_elements: Look at derefs for modes
- nir/lower_io_to_scalar: Look at derefs for modes
- nir/lower_wpos_center: Look at derefs for modes
- nir/copy_prop_vars: Get modes directly from derefs
- nir/dead_write_vars: Get modes directly from derefs
- radv/query: Add a nir_test_flag helper
- radv/query: Use 1-bit booleans in query shaders
- intel/blorp: Be more conservative about copying clear colors
- vulkan: Update the XML and headers to 1.1.97
- glsl_type: Support serializing 8 and 16-bit types
- spirv: Handle any bit size in vector_insert/extract
- anv/apply_pipeline_layout: Set the cursor in lower_res_reindex_intrinsic
- spirv: Sign-extend array indices
- spirv: Emit NIR deref instructions on-the-fly
- nir/builder: Add nir_i2i and nir_u2u helpers which take a bit size
- spirv: Handle arbitrary bit sizes for deref array indices
- nir/validate: Require array indices to match the deref bit size
- nir: Allow storing to shader_storage

- nir: Distinguish between normal uniforms and UBOs
- glsl_type: Drop the glsl_get_array_instance C helper
- glsl_type: Add a C wrapper to get struct field offsets
- glsl_type: Simplify glsl_channel_type
- glsl_type: Add support for explicitly laid out matrices and arrays
- spirv: Propagate layout decorations to created glsl_types
- nir: Move propagation of cast derefs to a new nir_opt_deref pass
- nir: Add a ptr_as_array deref type
- nir/validate: Don't allow derefs in if conditions
- nir/opt_deref: Properly optimize ptr_as_array derefs
- nir/deref: Support casts and ptr_as_array in comparisons
- nir/deref: Skip over casts in fixup_deref_modes
- nir/remove_dead_variables: Properly handle deref casts
- nir/validate: Allow derefs in phi nodes
- nir/intrinsics: Allow deref sources to consume anything
- nir/intrinsics: Add access flags to load/store_deref
- nir/validate: Allow array derefs on vectors in more modes
- nir/lower_io: Add "explicit" IO lowering
- nir/vulkan: Add a descriptor type to vulkan resource intrinsics
- spirv: Add error checking for Block and BufferBlock decorations
- spirv: Choose atomic deref type with pointer_uses_ssa_offset
- spirv: Add explicit pointer types
- spirv: Make better use of vtn_pointer_uses_ssa_offset
- spirv: Add support for using derefs for UBO/SSBO access
- anv: Enable the new deref-based UBO/SSBO path
- spirv: Sort supported capabilities
- anv: Sort properties and features switch statements
- nir: Add some more int64 lowering helpers
- anv/pipeline: Constant fold after apply_pipeline_layout
- anv/pipeline: Move wpos and input attachment lowering to lower_nir
- compiler/types: Serialize/deserialize subpass input types correctly
- anv/pipeline: Hash shader modules and spec constants separately
- anv/pipeline_cache: Add support for caching NIR
- anv/pipeline: Cache the pre-lowered NIR
- intel/peephole_ffma: Fix swizzle propagation
- spirv: Whack sampler/image pointers to uniform

- spirv: Contain the GLSLang issue #179 workaround to old GLSLang
- intel/nir: Call nir_opt_deref in brw_nir_optimize
- nir/gcm: Support deref instructions
- spirv: Emit switch conditions on-the-fly
- intel/blorp: Add two more filter modes
- anv: Rename has_resolve to has_color_resolve
- anv/blorp: Refactor MSAA resolves into an exportable helper function
- anv: Move resolve_subpass to genX_cmd_buffer.c
- anv: Implement VK_KHR_depth_stencil_resolve
- nir: Add a bool to float32 lowering pass
- intel/eu: Stop overriding exec sizes in send_indirect_message
- intel/fs: Don't touch accumulator destination while applying regioning alignment rule
- anv: Re-sort the extensions list
- anv: Only parse pImmutableSamplers if the descriptor has samplers
- relnotes: Add newly added Vulkan extensions
- anv/pipeline: Add a pdevice helper variable
- nir: Mark deref UBO and SSBO access as non-scalar
- spirv: Update the JSON and headers from Khronos master
- anv: Always emit at least one vertex element
- spirv: Initialize struct member offsets to -1
- spirv: Only split blocks
- spirv: Only set interface_type on blocks
- nir: Preserve offsets in lower_io_to_scalar_early
- nir/xfb: Fix offset accounting for dvec3/4
- nir/xfb: Properly handle arrays of blocks
- anv: Add but do not enable VK_EXT_transform_feedback
- anv: Add pipeline cache support for xfb_info
- anv: Implement the basic form of VK_EXT_transform_feedback
- anv: Implement vkCmdDrawIndirectByteCountEXT
- anv: Implement CmdBegin/EndQueryIndexed
- genxml: Add SO_PRIM_STORAGE_NEEDED and SO_NUM_PRIMS_WRITTEN
- anv: Implement transform feedback queries
- nir: Add load/store/atomic global intrinsics
- nir/lower_io: Add a 32 and 64-bit global address formats
- nir/lower_io: Add support for nir_var_mem_global
- nir/validate: Allow array derefs of vectors for nir_var_mem_global

- nir: Allow SSBOs and global to alias
- spirv: Drop a bogus assert
- spirv: Handle OpTypeForwardPointer
- spirv: Implement OpConvertPtrToU and OpConvertUToPtr
- spirv: Add support for SPV_EXT_physical_storage_buffer
- intel/fs: Get rid of fs_inst::equals
- intel/defines: Explicitly cast to uint32_t in SET_FIELD and SET_BITS
- intel/fs: Handle IMAGE_SIZE in size_read() and is_send_from_grf()
- intel/fs: Take an explicit exec size in brw_surface_payload_size()
- intel/eu: Add has_simd4x2 bools to surface_write functions
- intel/eu: Rework surface descriptor helpers
- intel/fs: Add a generic SEND opcode
- intel/fs: Use SHADER_OPCODE_SEND for surface messages
- intel/fs: Use a logical opcode for IMAGE_SIZE
- intel/fs: Use SHADER_OPCODE_SEND for texturing on gen7+
- intel/fs: Use SHADER_OPCODE_SEND for varying UBO pulls on gen7+
- intel/eu: Use GET_BITS in brw_inst_set_send_ex_desc
- intel/eu/validate: SEND restrictions also apply to SENDC
- intel/eu: Add more message descriptor helpers
- intel/disasm: Rework SEND decoding to use descriptors
- intel/inst: Fix the ia16_addr_imm helpers
- intel/inst: Indent some code
- intel/eu: Add support for the SENDS[C] messages
- intel/disasm: Properly disassemble split sends
- intel/fs: Support SENDS in SHADER_OPCODE_SEND
- intel/fs: Add interference between SENDS sources
- intel/fs: Use split sends for surface writes on gen9+
- intel/fs: Do the grf127 hack on SIMD8 instructions in SIMD16 mode
- nir/deref: Rematerialize parents in rematerialize_derefs_in_use_blocks
- intel/fs: Bail in optimize_extract_to_float if we have modifiers
- compiler/types: Add a contains_64bit helper
- nir/xfb: Properly align 64-bit values
- nir: Rewrite lower_clip_cull_distance_arrays to do a lot less lowering
- nir/xfb: Work in terms of components rather than slots
- nir/xfb: Handle compact arrays in gather_xfb_info
- nir/lower_clip_cull: Fix an incorrect assert

- anv: Count surfaces for non-YCbCr images in GetDescriptorSetLayoutSupport
- spirv: OpImageQueryLod requires a sampler
- intel,nir: Lower TXD with min_lod when the sampler index is not < 16
- spirv: Pull offset/stride from the pointer for OpArrayLength
- anv: Refactor descriptor pushing a bit
- anv: Take references to push descriptor set layouts
- nir: Add a pass for lowering IO back to vector when possible
- intel/nir: Vectorize all IO

Jiang, Sonny (1):

- radeonsi: add compute_last_block to configure the partial block fields

Jon Turney (3):

- glx: Fix compilation with GLX_USE_WINDOWSGL
- appveyor: put build steps in a script, rather than inline in appveyor.yml
- appveyor: Add a Cygwin build script

Jonathan Marek (42):

- nir: add fceil lowering
- freedreno: a2xx: fd2_draw update
- freedreno/a2xx: fix POINT_MINMAX_MAX overflow
- freedreno: add missing a20x ids
- freedreno/a2xx: set VIZ_QUERY_ID on a20x
- freedreno/a2xx: Compute depth base in gmem correctly
- freedreno: a2xx texture update
- freedreno: use GENERIC instead of TEXCOORD for blit program
- freedreno: use MSM_BO_SCANOUT with scanout buffers
- glsl/nir: int constants as float for native_integers=false
- glsl/nir: ftrunc for native_integers=false float to int cast
- glsl/nir: keep bool types when native_integers=false
- freedreno: a2xx: cleanup init_shader_const
- freedreno: a2xx: cleanup REG_A2XX_PA_CL_VTE_CNTL
- freedreno: a2xx: fix gmem2mem viewport
- freedreno: a2xx: fix VERTEX_REUSE/DEALLOC on a20x
- freedreno: a2xx: fix non-zero texture base offsets
- freedreno: a2xx: sysmem rendering
- freedreno: a2xx: NIR backend
- freedreno: a2xx: insert scalar MOV to allow 2 source scalar
- freedreno: a2xx: add ir2 copy propagation

- freedreno: a2xx: add partial lower_scalar pass for ir2
- freedreno: add renderonly scanout
- freedreno: a2xx: ir2 cleanup
- freedreno: a2xx: enable early-Z testing
- freedreno: update a2xx registers
- freedreno: a2xx: a20x hw binning
- freedreno: a2xx: clear fixes and fast clear path
- freedreno: a2xx: minor solid_vertexbuf fixups
- freedreno: a2xx: add perfcntrs
- kmsro: Add freedreno renderonly support
- st/dri: invalidate_resource depth/stencil before flush_resource
- mesa/st: wire up DiscardFramebuffer
- freedreno: fix invalidate logic
- freedreno: fix depth usage logic
- freedreno: fix sysmem rendering being used when clear is used
- freedreno: a2xx: fix fast clear
- freedreno: a2xx: don't write 4th vertex in mem2gmem
- freedreno: a2xx: add use_hw_binning function
- freedreno: a2xx: fix fast clear for some gmem configurations
- freedreno: a2xx: fix mipmapping for NPOT textures
- freedreno: use renderonly path for buffers allocated with modifiers

Jordan Justen (3):

- docs: Document GitLab merge request process (email alternative)
- i965/genX_state: Add register access functions
- i965/compute: Emit GPGPU_WALKER in genX_state_upload

Jose Maria Casanova Crespo (1):

- glsl: TCS outputs can not be transform feedback candidates on GLES

José Fonseca (2):

- appveyor: Revert commits adding Cygwin support.
- scons: Workaround failures with MSVC when using SCons 3.0.[2-4].

Juan A. Suarez Romero (17):

- docs: add release notes for 18.2.5
- docs: add sha256 checksums for 18.2.5
- docs: update calendar, add news item and link release notes for 18.2.5
- docs: add release notes for 18.2.6
- docs: add sha256 checksums for 18.2.6

- docs: update calendar, add news item and link release notes for 18.2.6
- docs: extends 18.2 lifecycle
- docs: add release notes for 18.2.7
- docs: add sha256 checksums for 18.2.7
- docs: update calendar, add news item and link release notes for 18.2.7
- docs: add release notes for 18.2.8
- docs: add sha256 checksums for 18.2.8
- docs: update calendar, add news item and link release notes for 18.2.8
- anv/cmd_buffer: check for NULL framebuffer
- genxml: add missing field values for 3DSTATE_SF
- anv: advertise 8 subpixel precision bits
- anv: destroy descriptor sets when pool gets reset

Józef Kucia (1):

- nir: Fix assert in print_intrinsic_instr().

Karol Herbst (35):

- nv50/ir: print color masks of tex instructions
- nv50/ra: add condenseDef overloads for partial condenses
- nv50/ir: add scalar field to TexInstructions
- gm107/ir: use scalar tex instructions where possible
- gm107/ir: fix compile time warning in getTEXSMask
- nir: add const_index parameters to system value builder function
- nir: replace nir_load_system_value calls with appropiate builder functions
- nir/spirv: cast shift operand to u32
- nv50,nvc0: Fix gallium nine regression regarding sampler bindings
- nv50/ir: initialize relDegree staticly
- nouveau: use atomic operations for driver statistics
- nv50/ir: fix use-after-free in ConstantFolding::visit
- nir: rename global/local to private/function memory
- nv50/ir: disable tryCollapseChainedMULs in ConstantFolding for precise instructions
- gm107/ir: disable TEXS for tex with derivAll set
- nir: rename nir_var_private to nir_var_shader_temp
- nir: rename nir_var_function to nir_var_function_temp
- nir: rename nir_var_ubo to nir_var_mem_ubo
- nir: rename nir_var_ssbo to nir_var_mem_ssbo
- nir: rename nir_var_shared to nir_var_mem_shared
- nir/spirv: handle SpvStorageClassCrossWorkgroup

- glsl/lower_output_reads: set invariant and precise flags on temporaries
- nir: replace more nir_load_system_value calls with builder functions
- nir/validate: allow to check against a bitmask of bit_sizes
- nir: add legal bit_sizes to intrinsics
- nir: add bit_size parameter to system values with multiple allowed bit sizes
- mesa: add MESA_SHADER_KERNEL
- vtn: handle SpvExecutionModelKernel
- nir/spirv: handle ContractionOff execution mode
- gk104/ir: Use the new rcp/rsq in library
- gm107/ir: add fp64 rcp
- gm107/ir: add fp64 rsq
- gallium: add PIPE_CAP_MAX_VARYINGS
- st/mesa: require RGBA2, RGB4, and RGBA4 to be renderable
- nir/opt_if: don't mark progress if nothing changes

Kenneth Graunke (41):

- intel: Use a URB start offset of 0 for disabled stages.
- st/mesa: Pull nir_lower_wpos_ytransform work into a helper function.
- st/nir: Drop unused parameter from st_nir_assign_uniform_locations().
- st/mesa: Don't record garbage streamout information in the non-SSO case.
- i915: Delete swizzling detection logic.
- nir: Use nir_shader_get_entrypoint in nir_lower_clip_vs().
- nir: Inline lower_clip_vs() into nir_lower_clip_vs().
- nir: Save nir_variable pointers in nir_lower_clip_vs rather than locs.
- nir: Make nir_lower_clip_vs optionally work with variables.
- i965: Allow only one slot of clip distances to be set on Gen4-5.
- i965: Use a 'nir' temporary rather than poking at brw_program
- i965: Do NIR shader cloning in the caller.
- intel/compiler: Use nir's info when checking uses_streams.
- intel/blorp: Expand blorp_address::offset to be 64 bits.
- i965: Delete dead brw_meta_resolve_color prototype.
- i965: Flip arguments to load_register_reg helpers.
- genxml: Consistently use a numeric "MOCS" field
- i965: Don't override subslice count to 4 on Gen11.
- st/mesa: Drop dead 'passthrough_fs' field.
- st/mesa: Drop !passColor optimization in drawpixels shaders.
- st/mesa: Don't open code the drawpixels vertex shader.

- st/mesa: Combine the DrawPixels and Bitmap passthrough VS programs.
- st/nir: Gather info after applying lowering FS variant features
- st/nir: Drop unused gl_program parameter in VS input handling helper.
- nir: Fix gl_nir_lower_samplers_as_deref's structure type handling.
- nir: Make gl_nir_lower_samplers use gl_nir_lower_samplers_as_deref
- blorp: Add blorp_get_surface_address to the driver interface.
- blorp: Pass the batch to lookup/upload_shader instead of context
- nir: Allow a non-existent sampler deref in nir_lower_samplers_as_deref
- st/nir: Lower TES gl_PatchVerticesIn to a constant if linked with a TCS.
- i965: Drop mark_surface_used mechanism.
- st/mesa: Make an enum for pipeline statistics query result indices.
- st/mesa: Rearrange PIPE_QUERY_PIPELINE_STATISTICS result fetching.
- gallium: Add the ability to query a single pipeline statistics counter
- st/mesa: Optionally override RGB/RGBX dst alpha blend factors
- gallium: Add forgotten docs for PIPE_CAP_GLSL_TESS_LEVELS_AS_INPUTS.
- st/mesa: Limit GL_MAX_[NATIVE_]PROGRAM_PARAMETERS_ARB to 2048
- anv: Put MOCS in the correct location
- nir: Don't reassociate add/mul chains containing only constants
- compiler: Mark clip/cull distance arrays as compact before lowering.
- spirv: Eliminate dead input/output variables after translation.

Kirill Burtsev (1):

- loader: free error state, when checking the drawable type

Kristian H. Kristensen (14):

- freedreno/a6xx: Clear z32 and separate stencil with blitter
- freedreno/a6xx: Move restore blits to IB
- freedreno/a6xx: Move resolve blits to an IB
- freedreno/a6xx: Clear gmem buffers at flush time
- gallium: Android build fixes
- mesa: Add core support for EXT_multisampled_render_to_texture{,2}
- gallium: Add new PIPE_CAP_SURFACE_SAMPLE_COUNT
- st/mesa: Add support for EXT_multisampled_render_to_texture
- freedreno: Add support for EXT_multisampled_render_to_texture
- freedreno: Fix the Makefile.am fix
- glapi: fixup EXT_multisampled_render_to_texture dispatch
- freedreno: Synchronize batch and flush for staging resource
- freedreno/a6xx: Turn on texture tiling by default

- freedreno/a6xx: Emit blitter dst with OUT_RELOCW

Leo Liu (2):

- st/va: fix the incorrect max profiles report
- st/va/vp9: set max reference as default of VP9 reference number

Lionel Landwerlin (47):

- intel/dump_gpu: add missing gdb option
- intel/sanitize_gpu: add help/gdb options to wrapper
- intel/sanitize_gpu: deal with non page multiple buffer sizes
- intel/sanitize_gpu: add debug message on mmap fail
- intel/decoders: fix instruction base address parsing
- anv: stub internal android code
- anv/android: mark gralloc allocated BOs as external
- intel/dump_gpu: move output option together
- intel/dump_gpu: add platform option
- intel/aub_read: remove useless breaks
- nir/lower_tex: add alpha channel parameter for yuv lowering
- nir/lower_tex: Add AYUV lowering support
- dri: add AYUV format
- i965: add support for sampling from AYUV
- anv: simplify internal address offset
- anv/image: remove unused parameter
- anv/lower_ycbcr: make sure to set 0s on all components
- anv: associate vulkan formats with aspects
- anv: use image aspects rather than computed ones
- anv: move helper function internally
- egl/dri: fix error value with unknown drm format
- intel/decoders: read ring buffer length
- intel/aubinator: fix ring buffer pointer
- intel/aub_viewer: fix dynamic state printing
- intel/aub_viewer: Print blend states properly
- anv: flush pipeline before query result copies
- anv/query: flush render target before copying results
- anv: don't do partial resolve on layer > 0
- intel/aub_viewer: fix shader get_bo
- intel/aub_viewer: fixup 0x address prefix
- intel/aub_viewer: print address of missing shader

- intel/aub_viewer: fix shader view
- intel/aub_viewer: fold binding/sampler table items
- intel/aub_viewer: highlight true booleans
- i965: limit VF caching workaround to gen8/9/10
- intel/blorp: emit VF caching workaround before 3DSTATE_VERTEX_BUFFERS
- i965: include draw_params/derived_draw_params for VF cache workaround
- i965: add CS stall on VF invalidation workaround
- anv: explictly specify format for blorp ccs/mcs op
- anv: flush fast clear colors into compressed surfaces
- anv: fix invalid binding table index computation
- anv: narrow flushing of the render target to buffer writes
- anv: document cache flushes & invalidations
- intel/genxml: add missing MI_PREDICATE compare operations
- vulkan: make generated enum to strings helpers available from c++
- intel: fix urb size for CFL GT1
- intel/compiler: use correct swizzle for replacement

Lucas Stach (6):

- etnaviv: use dummy RT buffer when rendering without color buffer
- etnaviv: use surface format directly
- st/dri: allow both render and sampler compatible dma-buf formats
- st/dri: replace format conversion functions with single mapping table
- etnaviv: enable full overwrite in a few more cases
- etnaviv: annotate variables only used in debug build

Marek Olšák (94):

- st/va: fix incorrect use of resource_destroy
- ac/surface: remove the overallocation workaround for Vega12
- radeonsi: use better DCC clear codes
- radeonsi: don't set the CB clear color registers for 0/1 clear colors on Raven2
- gallium: add PIPE_CONTEXT_LOSE_CONTEXT_ON_RESET
- radeonsi: stop command submission with PIPE_CONTEXT_LOSE_CONTEXT_ON_RESET only
- st/mesa: disable L3 thread pinning
- mesa: mark GL_SR8_EXT non-renderable on GLES
- radeonsi: fix video APIs on Raven2
- gallium/u_tests: add a compute shader test that clears an image
- gallium/u_tests: fix MSVC build by using old-style zero initializers
- mesa/glthread: pass the function name to _mesa_glthread_restore_dispatch

- mesa/glthread: enable immediate mode
- drirc: enable glthread for Talos Principle
- st/mesa: regularly re-pin driver threads to the CCX where the app thread is
- st/mesa: pin driver threads to a fixed CCX when glthread is enabled
- radeonsi: don't send data after write-confirm with BOTTOM_OF_PIPE_TS
- radeonsi: go back to using bottom-of-pipe for beginning of TIME_ELAPSED
- winsys/amdgpu: fix a buffer leak in amdgpu_bo_from_handle
- winsys/amdgpu: fix a device handle leak in amdgpu_winsys_create
- radeonsi: clean up primitive binning enablement
- radeonsi: use structured buffer intrinsics for image views
- radeonsi: fix is_oneway_access_only for image stores
- radeonsi: small cleanup for memory opcodes
- tgsi/scan: add more information about bindless usage
- radeonsi/nir: parse more information about bindless usage
- radeonsi: fix is_oneway_access_only for bindless images
- winsys/amdgpu: always reclaim/release slabs if there is not enough memory
- radeonsi: generalize the slab allocator code to allow layered slab allocators
- winsys/amdgpu: optimize slab allocation for 2 MB amdgpu page tables
- winsys/amdgpu: clean up code around BO VM alignment
- winsys/amdgpu: use >= instead of > for VM address alignment
- winsys/amdgpu: increase the VM alignment to the MSB of the size for Gfx9
- winsys/amdgpu: overallocate buffers for faster address translation on Gfx9
- winsys/amdgpu,radeon: pass vm_alignment to buffer_from_handle
- winsys/amdgpu: use optimal VM alignment for imported buffers
- winsys/amdgpu: use optimal VM alignment for CPU allocations
- radeonsi: allow si_cp_dma_clear_buffer to clear GDS from any IB
- winsys/amdgpu: add support for allocating GDS and OA resources
- radeonsi: add memory management stress tests for GDS
- Revert "winsys/amdgpu: overallocate buffers for faster address translation on Gfx9"
- st/mesa: expose GL_OES_texture_view
- mesa: expose GL_EXT_texture_view as an alias of GL_OES_texture_view
- mesa: expose EXT_texture_compression_rgtc on GLES
- mesa: expose EXT_texture_compression_bptc in GLES
- mesa: expose AMD_texture_texture4
- st/mesa: expose EXT_render_snorm on GLES
- radeonsi: don't emit redundant PKT3_NUM_INSTANCES packets

- radeonsi: call si_fix_resource_usage for the GS copy shader as well
- radeonsi: make si_cp_wait_mem more configurable
- radeonsi: use u_decomposed_prims_for_vertices instead of u_prims_for_vertices
- radeonsi: remove unused variables in si_insert_input_ptr
- radeonsi: always unmap texture CPU mappings on 32-bit CPU architectures
- ac: remove unused variable from ac_build_ddxy
- st/mesa: unify window-system renderbuffer initialization
- st/mesa: don't reference pipe_surface locally in PBO code
- st/mesa: don't leak pipe_surface if pipe_context is not current
- st/dri: fix dri2_format_table for argb1555 and rgb565
- radeonsi: also apply the GS hang workaround to draws without tessellation
- winsys/amdgpu: fix whitespace
- winsys/amdgpu: use the new BO list API
- radeonsi: fix a u_blitter crash after a shader with FBFETCH
- radeonsi: fix rendering to tiny viewports where the viewport center is > 8K
- radeonsi: use buffer_store_format_x & xy
- radeonsi: remove redundant call to emit_cache_flush in compute clear/copy
- radeonsi: compile clear and copy buffer compute shaders on demand
- radeonsi: correct WRITE_DATA.DST_SEL definitions
- radeonsi: fix the top-of-pipe fence on SI
- radeonsi: don't use WRITE_DATA.DST_SEL == MEM_GRBM on >= CIK
- radeonsi: move PKT3_WRITE_DATA generation into a helper function
- gallium: add SINT formats to have exact counterparts to SNORM formats
- gallium/util: add util_format_snorm8_to_sint8 (from radeonsi)
- radeonsi: disable render cond & pipeline stats for internal compute dispatches
- radeonsi: rename rscreen -> sscreen
- radeonsi: rename rview -> sview
- winsys/amdgpu: rename rfence, rsrc, rdst -> afence, asrc, adst
- radeonsi: remove r600 from comments
- radeonsi: rename r600_resource -> si_resource
- radeonsi: rename rquery -> squery
- radeonsi: rename rsrc -> ssrc, rdst -> sdst
- radeonsi: rename rbo, rbuffer to buf or buffer
- radeonsi: rename rfence -> sfence
- st/mesa: purge framebuffers when unbinding a context
- st/mesa: fix PRIMITIVES_GENERATED query after the "pipeline stat single" changes

- ac: use the correct LLVM processor name on Raven2
- radeonsi: fix crashing performance counters (division by zero)
- meson: drop the xcb-xrandr version requirement
- gallium/u_threaded: fix EXPLICIT_FLUSH for flush offsets > 0
- radeonsi: fix EXPLICIT_FLUSH for flush offsets > 0
- winsys/amdgpu: don't drop manually added fence dependencies
- radeonsi: add driconf option radeonsi_enable_nir
- radeonsi: always enable NIR for Civilization 6 to fix corruption
- driconf: add Civ6Sub executable for Civilization 6
- tgsi: don't set tgsi_info::uses_bindless_images for constbufs and hw atomics

Mario Kleiner (4):

- radeonsi: Fix use of 1- or 2- component GL_DOUBLE vbo's.
- egl/wayland: Allow client->server format conversion for PRIME offload. (v2)
- egl/wayland-drm: Only announce formats via wl_drm which the driver supports.
- drirc: Add sddm-greeter to adaptive_sync blacklist.

Mark Janes (3):

- Revert "i965/batch: avoid reverting batch buffer if saved state is an empty"
- Revert "Implementation of egl dri2 drivers for MESA_query_driver"
- Revert "Implement EGL API for MESA_query_driver"

Mathias Fröhlich (17):

- mesa: Remove needless indirection in some draw functions.
- mesa: Rename gl_vertex_array_object::_Enabled -> Enabled.
- mesa: Use the gl_vertex_array_object::Enabled bitfield.
- mesa: Use gl_vertex_array_object::Enabled for glGet.
- mesa: Remove gl_array_attributes::Enabled.
- mesa: Work with bitmasks when en/dis-abling VAO arrays.
- mesa: Unify glEdgeFlagPointer data type.
- nouveau: Use gl_array_attribute::_ElementSize.
- tnl: Use gl_array_attribute::_ElementSize.
- mesa: Factor out struct gl_vertex_format.
- mesa: Remove unneeded bitfield widths from the VAO.
- mesa/st: Only care about the uploader if it was used.
- mesa/st: Only unmap the uploader that was actually used.
- mesa/st: Factor out array and buffer setup from st_atom_array.c.
- mesa/st: Avoid extra references in the feedback draw function scope.
- mesa/st: Use binding information from the VAO in feedback rendering.

- mesa/st: Make st_pipe_vertex_format static.

Matt Turner (41):

- util/ralloc: Switch from DEBUG to NDEBUG
- util/ralloc: Make sizeof(linear_header) a multiple of 8
- nir: Call fflush() at the end of nir_print_shader()
- glsl: Remove unused member variable
- gallivm: Use nextafterf(0.5, 0.0) as rounding constant
- mesa: Revert INTEL_fragment_shader_ordering support
- Revert "st/mesa: silenced unhanded enum warning in st_glsl_to_tgsi.cpp"
- i965/fs: Handle V/UV immediates in dump_instructions()
- glsl: Add function support to glsl_to_nir
- glsl: Create file to contain software fp64 functions
- glsl: Add "built-in" functions to do ffma(fp64)
- glsl: Add "built-in" functions to do fmin/fmax(fp64)
- glsl: Add "built-in" function to do ffloor(fp64)
- glsl: Add "built-in" functions to do ffract(fp64)
- glsl: Add "built-in" functions to convert bool to double
- nir: Rework nir_lower_constant_initializers() to handle functions
- nir: Tag entrypoint for easy recognition by nir_shader_get_entrypoint()
- nir: Wire up int64 lowering functions
- nir: Implement lowering of 64-bit shift operations
- nir: Add and set info::uses_64bit
- nir: Create nir_builder in nir_lower_doubles_impl()
- nir: Add lowering support for 64-bit operations to software
- nir: Unset metadata debug bit if no progress made
- intel/compiler: Lower 64-bit MOV/SEL operations
- intel/compiler: Split 64-bit MOV-indirects if needed
- intel/compiler: Avoid false positive assertions
- intel/compiler: Rearrange code to avoid future problems
- intel/compiler: Prevent warnings in the following patch
- intel/compiler: Expand size of the 'nr' field
- intel/compiler: Heap-allocate temporary storage
- i965: Compile fp64 software routines and lower double-ops
- i965: Enable 64-bit GLSL extensions
- i965: Compile fp64 funcs only if we do not have 64-bit hardware support
- intel/compiler: Reset default flag register in brw_find_live_channel()

- gallium: Enable ASIMD/NEON on aarch64.
- gallivm: Return true from arch_rounding_available() if NEON is available
- intel/compiler: Add a file-level description of brw_eu_validate.c
- i965: Always compile fp64 funcs when needed
- nir: Optimize double-precision lower_round_even()
- intel/compiler: Avoid propagating inequality cmods if types are different
- intel/compiler/test: Add unit test for mismatched signedness comparison

Mauro Rossi (6):

- android: gallium/auxiliary: add include to get u_debug.h header
- android: radv: add libmesa_git_sha1 static dependency
- android: amd/addrlib: update Mesa's copy of addrlib
- android: st/mesa: fix building error due to sched_getcpu()
- android: anv: fix generated files depedencies (v2)
- android: anv: fix libexpat shared dependency

Maya Rashish (2):

- radeon: fix printf format specifier.
- configure: fix test portability

Michal Srb (2):

- gallium: Constify drisw_loader_funcs struct
- drisw: Use separate drisw_loader_funcs for shm

Michel Dänzer (4):

- winsys/amdgpu: Stop using amdgpu_bo_handle_type_kms_noimport
- winsys/amdgpu: Pull in LLVM CFLAGS
- amd/common: Restore v4i32 suffix for llvm.SI.load.const intrinsic
- loader/dri3: Use strlen instead of sizeof for creating VRR property atom

Neha Bhende (1):

- st/mesa: Fix topogun-1.06-orc-84k-resize.trace crash

Neil Roberts (4):

- freedreno: Add .dir-locals to the common directory
- spirv/nir: handle location decorations on block interface members
- glsl_types: Rename parameter of glsl_count_attribute_slots
- spirv: Don't use special semantics when counting vertex attribute size

Nicholas Kazlauskas (5):

- util: Get program name based on path when possible
- util: Add adaptive_sync driconf option
- drirc: Initial blacklist for adaptive sync

- loader/dri3: Enable adaptive_sync via _VARIABLE_REFRESH property
- radeonsi: Enable adaptive_sync by default for radeon

Nicolai Hähnle (37):

- radv: include LLVM IR in the VK_AMD_shader_info "disassembly"
- radeonsi: fix an out-of-bounds read reported by ASAN
- winsys/amdgpu: add amdgpu_winsys_bo::lock
- winsys/amdgpu: explicitly declare whether buffer_map is permanent or not
- egl/wayland: rather obvious build fix
- radv: remove dependency on addrlib gfx9_enum.h
- ac/surface/gfx9: let addrlib choose the preferred swizzle kind
- amd/addrlib: update Mesa's copy of addrlib
- meson: link LLVM 'native' component when LLVM is available
- ddebug: simplify watchdog loop and fix crash in the no-timeout case
- ddebug: always flush when requested, even when hang detection is disabled
- r600: remove redundant semicolon
- amd/sid_tables: add additional python3 compatibility imports
- amd/common: whitespace fixes
- amd/common: add ac_build_ifcc
- amd/common: scan/reduce across waves of a workgroup
- amd/common: add i1 special case to ac_build_{inclusive,exclusive}_scan
- ac/surface: 3D and cube surfaces are never displayable
- radeonsi: move SI_FORCE_FAMILY functionality to winsys
- radeonsi: extract declare_vs_blit_inputs
- radeonsi: add si_init_draw_functions and make some functions static
- radeonsi/gfx9: use SET_UCONFIG_REG_INDEX packets when available
- radeonsi: don't set RAW_WAIT for CP DMA clears
- radeonsi: rename SI_RESOURCE_FLAG_FORCE_TILING to clarify its purpose
- radeonsi: const-ify si_set_tesseval_regs
- radeonsi: show the fixed function TCS in debug dumps
- radeonsi: avoid using hard-coded SI_NUM_RW_BUFFERS
- radeonsi: add an si_set_rw_shader_buffer convenience function
- radeonsi: use si_set_rw_shader_buffer for setting streamout buffers
- radeonsi: track constant buffer bind history in si_pipe_set_constant_buffer
- radeonsi: move remaining perfcounter code into si_perfcounter.c
- radeonsi: move query suspend logic into the top-level si_query struct
- radeonsi: factor si_query_buffer logic out of si_query_hw

- radeonsi: split perfcounter queries from si_query_hw
- radeonsi: const-ify the si_query_ops
- amd/common: use llvm.amdgcn.s.buffer.load for LLVM 8.0
- amd/common/vi+: enable SMEM loads with GLC=1

Niklas Haas (3):

- glsl: fix block member alignment validation for vec3
- radv: correctly use vulkan 1.0 by default
- radv: add device->instance extension dependencies

Olivier Fourdan (1):

- wayland/egl: Resize EGL surface on update buffer for swrast

Oscar Blumberg (1):

- radeonsi: Fix guardband computation for large render targets

Pierre Moreau (2):

- clover/meson: Ignore 'svn' suffix when computing CLANG_RESOURCE_DIR
- meson: Fix with_gallium_icd to with_opencl_icd

Plamena Manolova (1):

- nir: Don't lower the local work group size if it's variable.

Rafael Antognolli (24):

- intel/genxml: Add register for object preemption.
- i965/gen10+: Enable object level preemption.
- i965/gen9: Add workarounds for object preemption.
- anv/tests: Fix block_pool_no_free test.
- anv/allocator: Add anv_state_table.
- anv/allocator: Add getter for anv_block_pool.
- anv/allocator: Add helper to push states back to the state table.
- anv/allocator: Use anv_state_table on anv_state_pool_alloc.
- anv/allocator: Use anv_state_table on back_alloc too.
- anv/allocator: Remove anv_free_list.
- anv/allocator: Rename anv_free_list2 to anv_free_list.
- anv/allocator: Remove pool->map.
- anv: Update usage of block_pool->bo.
- anv/allocator: Add support for a list of BOs in block pool.
- anv: Split code to add BO dependencies to execbuf.
- anv: Validate the list of BOs from the block pool.
- anv: Remove some asserts.
- anv/allocator: Rework chunk return to the state pool.

- anv/allocator: Add padding information.
- anv/allocator: Enable snooping on block pool and anv_bo_pool BOs.
- anv: Remove state flush.
- anv/allocator: Add support for non-userptr.
- anv/tests: Adding test for the state_pool padding.
- anv/allocator: Avoid race condition in anv_block_pool_map.

Ray Zhang (1):

- glx: fix shared memory leak in X11

Rhys Kidd (5):

- travis: radeonsi and radv require LLVM 7.0
- meson: libfreedreno depends upon libdrm (for fence support)
- v3d: Wire up core pipe_debug_callback
- vc4: Wire up core pipe_debug_callback
- nv50,nvc0: add missing CAPs for unsupported features

Rhys Perry (14):

- nir: fix constness in nir_intrinsic_align()
- ac: refactor visit_load_buffer
- ac: split 16-bit ssbo loads that may not be dword aligned
- radv: don't set surf_index for stencil-only images
- radv: switch from nir_bcsel to nir_b32csel
- ac/nir,radv,radeonsi/nir: use correct indices for interpolation intrinsics
- nir: fix copy-paste error in nir_lower_constant_initializers
- radv: use dithered alpha-to-coverage
- radv: pass radv_draw_info to radv_emit_draw_registers()
- radv: add missed situations for scissor bug workaround
- radv: avoid context rolls when binding graphics pipelines
- radv: prevent dirtying of dynamic state when it does not change
- radv: bitcast 16-bit outputs to integers
- radv: ensure export arguments are always float

Rob Clark (79):

- freedreno: update generated headers
- freedreno/a6xx: fix VSC bug with larger # of tiles
- freedreno/drm: fix unused 'entry' warnings
- freedreno/drm: remove dependency on gallium driver
- freedreno: move drm to common location
- freedreno/ir3: standalone compiler updates

- freedreno: shader_t -> gl_shader_stage
- freedreno: remove shader_stage_name()
- freedreno: FD_SHADER_DEBUG -> IR3_SHADER_DEBUG
- freedreno/ir3: move disasm and optmsgs debug flags
- util: env_var_as_unsigned() helper
- freedreno/ir3: use env_var_as_unsigned()
- freedreno/ir3: some header file cleanup
- freedreno/ir3: remove pipe_stream_output_info dependency
- freedreno/ir3: split up ir3_shader
- freedreno/ir3: remove u_inlines usage
- freedreno: move ir3 to common location
- mesa/st: swap order of clear() and clear_with_quad()
- mesa/st: better colormask check for clear fallback
- freedreno/a6xx: disable LRZ for z32
- freedreno/a6xx: set guardband clip
- freedreno: update generated headers
- freedreno/a3xx: also set FSSUPERTHREADENABLE
- freedreno/a6xx: MSAA
- freedreno: remove unused fd_surface fields
- gallium: fix typo
- freedreno/a5xx+a6xx: remove unused fs/vs pvt mem
- freedreno/drm: fix relocs in nested stateobjs
- freedreno: update generated headers
- freedreno/a6xx: blitter fixes
- freedreno/ir3: don't fetch unused tex components
- freedreno/ir3: sync instr/disasm
- freedreno/ir3: code-motion
- freedreno/ir3: track max flow control depth for a5xx/a6xx
- freedreno/drm: fix memory leak
- nir: fix spelling typo
- mesa/st/nir: fix missing nir_compact_varyings
- freedreno/drm: sync uapi and enable softpin
- freedreno: debug GEM obj names
- freedreno: also set DUMP flag on shaders
- freedreno/ir3: fix crash
- freedreno/ir3: don't remove unused input components

- freedreno/a6xx: fix blitter crash
- gallium/aux: add is_unorm() helper
- freedreno: update generated headers
- freedreno/a6xx: more blitter fixes
- freedreno: move fd_resource_copy_region()
- freedreno/a6xx: fix resource_copy_region()
- freedreno/a6xx: fix corrupted uniforms
- freedreno/ir3: fix fallout of extra assert
- freedreno/ir3: don't treat all inputs/outputs as vec4
- freedreno: combine fd_resource_layer_offset()/fd_resource_offset()
- freedreno/a6xx: simplify special case for 3d layout
- freedreno/a6xx: improve setup_slices() debug msgs
- freedreno: update generated headers
- freedreno/a6xx: fix 3d texture layout
- freedreno: skip depth resolve if not written
- freedreno: rework blit API
- freedreno: try blitter for fd_resource_copy_region()
- freedreno/a6xx: rework blitter API
- freedreno: remove blit_via_copy_region()
- freedreno: fix staging resource size for arrays
- freedreno: make cmdstream bo's read-only to GPU
- freedreno/a6xx: separate stencil restore/resolve fixes
- freedreno/a6xx: move tile_mode to sampler-view CSO
- freedreno/a6xx: fix 3d+tiled layout
- nir/vtn: add caps for some cl related capabilities
- loader: fix the no-modifiers case
- freedreno: core buffer modifier support
- freedreno: set modifier when exporting buffer
- freedreno: limit tiling to PIPE_BIND_SAMPLER_VIEW
- freedreno/a2xx: fix unused variable warning
- freedreno/a5xx: fix blitter nr_samples check
- freedreno/a6xx: fix blitter nr_samples check
- freedreno: stop frob'ing pipe_resource::nr_samples
- freedreno: minor cleanups
- mesa: wire up InvalidateFramebuffer
- freedreno: fix release tarball

- freedreno: more fixing release tarball

Rob Herring (3):

- pipe-loader: Fallback to kmsro driver when no matching driver name found
- kmsro: Add etnaviv renderonly support
- Switch imx to kmsro and remove the imx winsys

Robert Foss (3):

- virgl: native fence fd support
- virgl: Clean up fences commit
- virgl: add assert and missing function parameter

Rodrigo Vivi (1):

- intel: Add more PCI Device IDs for Coffee Lake and Ice Lake.

Roland Scheidegger (7):

- gallivm: fix improper clamping of vertex index when fetching gs inputs
- draw: fix infinite loop in line stippling
- gallivm: remove unused float coord wrapping for aos sampling
- gallivm: use llvm jit code for decoding s3tc
- gallivm: don't use pavg.b intrinsic on llvm >= 6.0
- gallivm: abort when trying to use non-existing intrinsic
- Revert "llvmpipe: Always return some fence in flush (v2)"

Sagar Ghuge (14):

- intel/compiler: Disassemble GEN6_SFID_DATAPORT_SAMPLER_CACHE as dp_sampler
- intel/compiler: Set swizzle to BRW_SWIZZLE_XXXX for scalar region
- intel/compiler: Always print flag subregister number
- nir: Add a new lowering option to lower 3D surfaces from txd to txl.
- glsl: Add "built-in" functions to do uint64_to_fp64(uint64_t)
- glsl: Add "built-in" functions to do int64_to_fp64(int64_t)
- glsl: Add "built-in" functions to do uint64_to_fp32(uint64_t)
- glsl: Add "built-in" functions to do int64_to_fp32(int64_t)
- glsl: Add utility function to round and pack uint64_t value
- glsl: Add "built-in" functions to do fp64_to_uint64(fp64)
- glsl: Add utility function to round and pack int64_t value
- glsl: Add "built-in" functions to do fp64_to_int64(fp64)
- glsl: Add "built-in" functions to do fp32_to_uint64(fp32)
- glsl: Add "built-in" functions to do fp32_to_int64(fp32)

Samuel Pitoiset (103):

- radv: remove useless sync after copying query results with compute

- radv: add missing TFB queries support to CmdCopyQueryPoolsResults()
- radv: replace si_emit_wait_fence() with radv_cp_wait_mem()
- radv: more use of radv_cp_wait_mem()
- radv: allocate enough space in CS when copying query results with compute
- radv: disable conditional rendering for vkCmdCopyQueryPoolResults()
- radv: only expose VK_SUBGROUP_FEATURE_ARITHMETIC_BIT for VI+
- radv: use LOAD_CONTEXT_REG when loading fast clear values
- radv: fix GPU hangs when loading depth/stencil clear values on SI/CIK
- radv: cleanup and document a Hawaii bug with offchip buffers
- radv: clean up setting partial_es_wave for distributed tess on VI
- radv: make use of num_good_cu_per_sh in si_emit_graphics() too
- radv: binding streamout buffers doesn't change context regs
- radv: set PA.SC_CONSERVATIVE_RASTERIZATION.NULL_SQUAD_AA_MASK_ENABLE
- radv: set optimal OVERWRITE_COMBINER_WATERMARK on GFX9
- radv: add a debug option for disabling primitive binning
- radv: enable primitive binning by default
- radv: tidy up radv_set_dcc_need_cmask_elim_pred()
- radv: always clear the FCE predicate after DCC/FMASK/CMASK decompressions
- radv/winsys: remove the max IBs per submit limit for the fallback path
- radv/winsys: remove the max IBs per submit limit for the sysmem path
- radv: remove unnecessary goto in the fast clear paths
- radv: add radv_get_htile_fast_clear_value() helper
- radv: add radv_is_fast_clear_{depth,stencil}_allowed() helpers
- radv: check allowed fast HTILE clears a bit earlier
- radv: rewrite the condition that checks allowed depth/stencil values
- radv: implement fast HTILE clears for depth or stencil only on GFX9
- ac/nir: fix intrinsic name string size in visit_image_atomic()
- radv: ignore subpass self-dependencies
- radv: only sync CP DMA for transfer operations or bottom pipe
- radv: remove useless sync after CmdClear{Color,DepthStencil}Image()
- radv: remove useless sync before CmdClear{Color,DepthStencil}Image()
- radv: ignore subpass self-dependencies for CreateRenderPass() too
- radv: remove useless check in emit_fast_color_clear()
- radv: add radv_image_can_fast_clear() helper
- radv: add radv_image_view_can_fast_clear() helper
- radv: add radv_can_fast_clear_{color,depth}() helpers

- radv: simplify a check in emit_fast_color_clear()
- radv: refactor the fast clear path for better re-use
- radv: optimize CmdClear{Color,DepthStencil}Image() for layered textures
- radv: remove unused pending_clears param in the transition path
- radv: drop few useless state changes when doing color/depth decompressions
- radv: rework the TC-compat HTILE hardware bug with COND_EXEC
- radv: reset pending_reset_query when flushing caches
- radv: wait on the high 32 bits of timestamp queries
- spirv: add SpvCapabilityInt64Atomics
- radv: expose VK_EXT_scalar_block_layout
- amd: remove support for LLVM 6.0
- gallium: add missing PIPE_CAP_SURFACE_SAMPLE_COUNT default value
- radv: bump reported version to 1.1.90
- radv: add a predicate for reflecting DCC decompression state
- radv: allow to skip DCC decompressions with the new predicate
- radv: switch on EOP when primitive restart is enabled with triangle strips
- radv: check if addrlib enabled HTILE in radv_image_can_enable_htile()
- radv: don't check if format is depth in radv_image_can_enable_hile()
- radv: report Vulkan version 1.1.90 for real
- ac/nir: remove the bitfield_extract workaround for LLVM 8
- radv: drop the amdgpu-skip-threshold=1 workaround for LLVM 8
- radv: fix subpass image transitions with multiviews
- radv: compute optimal VM alignment for imported buffers
- spirv: add support for SpvCapabilityStorageImageMultisample
- ac/nir: restrict fmask lookup to image load intrinsics
- radv: initialize FMASK for images in fully expanded mode
- radv: add support for FMASK expand
- radv: enable shaderStorageImageMultisample feature on GFX8+
- radv: get rid of bunch of KHR suffixes
- radv: enable variable pointers
- radv: skip draws with instance_count == 0
- ac/nir: add get_cache_policy() helper and use it
- ac/nir: set cache policy when loading/storing buffer images
- ac: add missing 16-bit types to glsl_base_to_llvm_type()
- radv: remove unnecessary returns in GetPhysicalDevice*Properties()
- radv: add two small helpers for getting VRAM and visible VRAM sizes

- radv: add support for VK_EXT_memory_budget
- ac/nir: don't trash L1 caches for store operations with writeonly memory
- radv: drop unused code related to 16 sample locations
- radv: reduce size of the per-queue descriptor BO
- radv: do not write unused descriptors to the per-queue BO
- radv: initialize the per-queue descriptor BO only once
- nir: do not remove varyings used for transform feedback
- nir: fix lowering arrays to elements for XFB outputs
- radv: improve gathering of load_push_constants with dynamic bindings
- radv: remove old_fence parameter from si_cs_emit_write_event_eop()
- radv: only allocate the GFX9 fence and EOP BOs for the gfx queue
- radv: compute the GFX9 fence VA at allocation time
- radv: always pass the GFX9 fence data to si_cs_emit_cache_flush()
- radv: fix computing number of user SGPRs for streamout buffers
- radv: remove radv_userdata_info::indirect field
- radv: simplify allocating user SGPRS for descriptor sets
- radv: set noalias/dereferenceable LLVM attributes based on param types
- radv: re-enable fast depth clears for 16-bit surfaces on VI
- radv/winsys: fix hash when adding internal buffers
- radv: fix compiler issues with GCC 9
- radv: fix using LOAD_CONTEXT_REG with old GFX ME firmwares on GFX8
- radv/winsys: fix BO list creation when RADV_DEBUG=allbos is set
- radv: always export gl_SampleMask when the fragment shader uses it
- radv: write the alpha channel of MRT0 when alpha coverage is enabled
- radv: fix writing the alpha channel of MRT0 when alpha coverage is enabled
- radv: fix out-of-bounds access when copying descriptors BO list
- radv: don't copy buffer descriptors list for samplers
- radv: fix clearing attachments in secondary command buffers
- radv: properly align the fence and EOP bug VA on GFX9
- radv: fix pointSizeRange limits

Sergii Romantsov (4):

- autotools: library-dependency when no sse and 32-bit
- i965/batch/debug: Allow log be dumped before assert
- nir: Length of boolean vtn_value now is 1
- dri: meson: do not prefix user provided dri-drivers-path

Sonny Jiang (1):

- radeonsi: use compute for resource_copy_region when possible

Tapani Pälli (27):

- anv: allow exporting an imported SYNC_FD semaphore type
- anv: add create_flags as part of anv_image
- anv: refactor make_surface to use data from anv_image
- anv: make anv_get_image_format_features public
- anv: add from/to helpers with android and vulkan formats
- anv/android: add GetAndroidHardwareBufferPropertiesANDROID
- anv: add anv_ahw_usage_from_vk_usage helper function
- anv: refactor, remove else block in AllocateMemory
- anv/android: support import/export of AHardwareBuffer objects
- anv/android: add ahardwarebuffer external memory properties
- anv/android: support creating images from external format
- anv: support VkExternalFormatANDROID in vkCreateSamplerYcbcrConversion
- anv: add VkFormat field as part of anv_format
- anv: support VkSamplerYcbcrConversionInfo in vkCreateImageView
- anv: ignore VkSamplerYcbcrConversion on non-yuv formats
- anv/android: turn on VK_ANDROID_external_memory_android_hardware_buffer
- dri3: initialize adaptive_sync as false before configQueryb
- intel/isl: move tiled_memcpy static libs from i965 to isl
- anv: do not advertise AHW support if extension not enabled
- nir: cleanup glsl_get_struct_field_offset, glsl_get_explicit_stride
- android: fix build issues with libmesa_anv_gen* libraries
- mesa: return NULL if we exceed MaxColorAttachments in get_fb_attachment
- nir: initialize value in copy_prop_vars_block
- anv: retain the is_array state in create_plane_tex_instr_implicit
- anv: release memory allocated by glsl types during spirv_to_nir
- anv: revert "anv: release memory allocated by glsl types during spirv_to_nir"
- anv: destroy descriptor sets when pool gets destroyed

Thomas Hellstrom (9):

- st/xa: Render update. Better support for solid pictures
- st/xa: Support higher color precision for solid pictures
- st/xa: Support a couple of new formats
- st/xa: Fix transformations when we have both source and mask samplers
- st/xa: Minor renderer cleanups
- st/xa: Support Component Alpha with trivial blending

- st/xa: Bump minor
- st/xa: Fix a memory leak
- winsys/svga: Fix a memory leak

Timothy Arceri (56):

- nir: allow propagation of if evaluation for bcsel
- nir: fix condition propagation when src has a swizzle
- ac/nir_to_llvm: fix b2f for f64
- nir: add new linking opt nir_link_constant_varyings()
- st/mesa: make use of nir_link_constant_varyings()
- nir: add glsl_type_is_integer() helper
- nir: don't pack varyings ints with floats unless flat
- anv/i965: make use of nir_link_constant_varyings()
- nir: add support for removing redundant stores to copy prop var
- radv: make use of nir_move_out_const_to_consumer()
- nir: small tidy ups for nir_loop_analyze()
- nir: clarify some nit_loop_info member names
- nir: add a new nir_cf_list_clone_and_reinsert() helper
- nir: make use of new nir_cf_list_clone_and_reinsert() helper
- nir: factor out some of the complex loop unroll code to a helper
- nir: rework force_unroll_array_access()
- nir: in loop analysis track actual control flow type
- nir: reword code comment
- nir: detect more induction variables
- nir: fix opt_if_loop_last_continue()
- tgsi/scan: fix loop exit point in tgsi_scan_tess_ctrl()
- tgsi/scan: correctly walk instructions in tgsi_scan_tess_ctrl()
- radeonsi: remove unrequired param in si_nir_scan_tess_ctrl()
- ac/nir_to_llvm: add ac_are_tessfactors_def_in_all_invocs()
- radeonsi: make use of ac_are_tessfactors_def_in_all_invocs()
- st/glsl_to_nir: call nir_lower_load_const_to_scalar() in the st
- nir: rename nir_link_constant_varyings() nir_link_opt_varyings()
- nir: add can_replace_varying() helper
- nir: rework nir_link_opt_varyings()
- nir: link time opt duplicate varyings
- nir: make nir_opt_remove_phis_impl() static
- nir: make use of does_varying_match() helper

- nir: simplify does_varying_match()
- nir: add rewrite_phi_predecessor_blocks() helper
- nir: merge some basic consecutive ifs
- st/glsl: refactor st_link_nir()
- nir: avoid uninitialized variable warning
- glsl: Copy function out to temp if we don't directly ref a variable
- ac/nir_to_llvm: fix type handling in image code
- radeonsi/nir: get correct type for images inside structs
- ac/nir_to_llvm: fix regression in bindless support
- ac/nir_to_llvm: add support for structs to get_sampler_desc()
- glsl: don't skip GLSL IR opts on first-time compiles
- glsl: be much more aggressive when skipping shader compilation
- Revert "glsl: be much more aggressive when skipping shader compilation"
- ac/nir_to_llvm: fix interpolateAt* for arrays
- glsl: be much more aggressive when skipping shader compilation
- radeonsi/nir: add missing piece for bindless image support
- ac/nir_to_llvm: add bindless support for uniform handles
- ac/nir_to_llvm: fix interpolateAt* for structs
- ac/nir_to_llvm: fix clamp shadow reference for more hardware
- tgsi: remove culldist semantic from docs
- radv/ac: fix some fp16 handling
- glsl: use remap location when serialising uniform program resource data
- radeonsi: fix query buffer allocation
- glsl: fix shader cache for packed param list

Tobias Klausmann (1):

- amd/vulkan: meson build - use radv_deps for libvulkan_radeon

Tomasz Figa (1):

- llvmpipe: Always return some fence in flush (v2)

Tomeu Vizoso (1):

- etnaviv: Consolidate buffer references from framebuffers

Toni Lönnberg (14):

- intel/decoder: Engine parameter for instructions
- intel/decoder: tools: gen_engine to drm_i915_gem_engine_class
- intel/decoder: tools: Use engine for decoding batch instructions
- intel/genxml: Add engine definition to render engine instructions (gen4)
- intel/genxml: Add engine definition to render engine instructions (gen45)

- intel/genxml: Add engine definition to render engine instructions (gen5)
- intel/genxml: Add engine definition to render engine instructions (gen6)
- intel/genxml: Add engine definition to render engine instructions (gen7)
- intel/genxml: Add engine definition to render engine instructions (gen75)
- intel/genxml: Add engine definition to render engine instructions (gen8)
- intel/genxml: Add engine definition to render engine instructions (gen9)
- intel/genxml: Add engine definition to render engine instructions (gen10)
- intel/genxml: Add engine definition to render engine instructions (gen11)
- intel/aubinator_error_decode: Get rid of warning for missing switch case

Topi Pohjolainen (1):

- i965/icl: Disable prefetching of sampler state entries

Veluri Mithun (5):

- Add extension doc for MESA_query_driver
- Implement EGL API for MESA_query_driver
- Implementation of egl dri2 drivers for MESA_query_driver
- egl: Implement EGL API for MESA_query_driver
- egl: Implementation of egl dri2 drivers for MESA_query_driver

Vinson Lee (7):

- r600/sb: Fix constant logical operand in assert.
- freedreno: Fix autotools build.
- st/xvmc: Add X11 include path.
- nir/algebraic: Make algebraic_parser_test.sh executable.
- meson: Fix typo.
- meson: Fix libsensors detection.
- meson: Fix typo.

Yevhenii Kolesnikov (1):

- i965: Fix allow_higher_compat_version workaround limited by OpenGL 3.0

pal1000 (1):

- scons: Compatibility with Scons development version string

4.58 Mesa 18.3.4 Release Notes / February 18, 2019

Mesa 18.3.4 is a bug fix release which fixes bugs found since the 18.3.3 release.

Mesa 18.3.4 implements the OpenGL 4.5 API, but the version reported by glGetString(GL_VERSION) or glGetIntegerv(GL_MAJOR_VERSION) / glGetIntegerv(GL_MINOR_VERSION) depends on the particular driver being used. Some drivers don't support all the features required in OpenGL 4.5. OpenGL 4.5 is **only** available if requested at context creation. Compatibility contexts may report a lower version depending on each driver.

4.58.1 SHA256 checksums

```
e22e6fe4c3aca80fe872a0a7285b6c5523e0cfc0bfb57ffcc3b3d66d292593e4  mesa-18.3.4.tar.gz
32314da4365d37f80d84f599bd9625b00161c273c39600ba63b45002d500bb07  mesa-18.3.4.tar.xz
```

4.58.2 New features

None

4.58.3 Bug fixes

- Bug 109107 - gallium/st/va: change va max_profiles when using Radeon VCN Hardware
- Bug 109401 - [DXVK] Project Cars rendering problems
- Bug 109543 - After upgrade mesa to 19.0.0~rc1 all vulkan based application stop working [“vulkan-cube” received SIGSEGV in radv_pipeline_init_blend_state at ../src/amd/vulkan/radv_pipeline.c:699]
- Bug 109603 - nir_instr_as_deref: Assertion ‘parent && parent->type == nir_instr_type_deref’ failed.

4.58.4 Changes

Bart Oldeman (1):

- gallium-xlib: query MIT-SHM before using it.

Bas Nieuwenhuizen (2):

- radv: Only look at pImmutableSamples if the descriptor has a sampler.
- amd/common: Use correct writemask for shared memory stores.

Dylan Baker (2):

- get-pick-list: Add –pretty=medium to the arguments for Cc patches
- meson: Add dependency on genxml to anvil

Emil Velikov (5):

- docs: add sha256 checksums for 18.3.3
- cherry-ignore: nv50,nvc0: add explicit settings for recent caps
- cherry-ignore: add more 19.0 only nominations from Ilia
- cherry-ignore: radv: fix using LOAD_CONTEXT_REG with old GFX ME firmwares on GFX8
- Update version to 18.3.4

Eric Anholt (1):

- vc4: Fix copy-and-paste fail in backport of NEON asm fixes.

Eric Engestrom (2):

- xvmc: fix string comparison
- xvmc: fix string comparison

Ernestas Kulik (2):

- vc4: Fix leak in HW queries error path
- v3d: Fix leak in resource setup error path

Iago Toral Quiroga (1):

- intel/compiler: do not copy-propagate strided regions to ddx/ddy arguments

Ilia Mirkin (1):

- nvc0: we have 16k-sized framebuffers, fix default scissors

Jason Ekstrand (3):

- intel/fs: Handle IMAGE_SIZE in size_read() and is_send_from_grf()
- intel/fs: Do the grf127 hack on SIMD8 instructions in SIMD16 mode
- nir/deref: Rematerialize parents in rematerialize_derefs_in_use_blocks

Juan A. Suarez Romero (1):

- anv/cmd_buffer: check for NULL framebuffer

Kenneth Graunke (1):

- st/mesa: Limit GL_MAX_[NATIVE_]PROGRAM_PARAMETERS_ARB to 2048

Kristian H. Kristensen (1):

- freedreno/a6xx: Emit blitter dst with OUT_RELOCW

Leo Liu (2):

- st/va: fix the incorrect max profiles report
- st/va/vp9: set max reference as default of VP9 reference number

Marek Olšák (4):

- meson: drop the xcb-xrandr version requirement
- gallium/u_threaded: fix EXPLICIT_FLUSH for flush offsets > 0
- radeonsi: fix EXPLICIT_FLUSH for flush offsets > 0
- winsys/amdgpu: don't drop manually added fence dependencies

Mario Kleiner (2):

- egl/wayland: Allow client->server format conversion for PRIME offload. (v2)
- egl/wayland-drm: Only announce formats via wl_drm which the driver supports.

Oscar Blumberg (1):

- radeonsi: Fix guardband computation for large render targets

Rob Clark (1):

- freedreno: stop frob'ing pipe_resource::nr_samples

Rodrigo Vivi (1):

- intel: Add more PCI Device IDs for Coffee Lake and Ice Lake.

Samuel Pitoiset (2):

- radv: fix compiler issues with GCC 9
- radv: always export gl_SampleMask when the fragment shader uses it

4.59 Mesa 18.3.3 Release Notes / January 31, 2019

Mesa 18.3.3 is a bug fix release which fixes bugs found since the 18.3.2 release.

Mesa 18.3.3 implements the OpenGL 4.5 API, but the version reported by glGetString(GL_VERSION) or glGetIntegerv(GL_MAJOR_VERSION) / glGetIntegerv(GL_MINOR_VERSION) depends on the particular driver being used. Some drivers don't support all the features required in OpenGL 4.5. OpenGL 4.5 is **only** available if requested at context creation. Compatibility contexts may report a lower version depending on each driver.

4.59.1 SHA256 checksums

```
6b9893942fe8011c7736d51448deb6ef80ece2257e0fac27b02e997a6605d5e4  mesa-18.3.3.tar.gz
2ab6886a6966c532ccbcc3b240925e681464b658244f0cbed752615af3936299  mesa-18.3.3.tar.xz
```

4.59.2 New features

None

4.59.3 Bug fixes

- Bug 108877 - OpenGL CTS gl43 test cases were interrupted due to segment fault
- Bug 109023 - error: inlining failed in call to always_inline '__m512 _mm512_and_ps(__m512, __m512)': target specific option mismatch
- Bug 109129 - format_types.h:1220: undefined reference to '_mm256_cvtps_ph'
- Bug 109229 - glLinkProgram locks up for ~30 seconds
- Bug 109242 - [RADV] The Witcher 3 system freeze
- Bug 109488 - Mesa 18.3.2 crash on a specific fragment shader (assert triggered) / already fixed on the master branch.

4.59.4 Changes

Andres Gomez (2):

- bin/get-pick-list.sh: fix the oneline printing
- bin/get-pick-list.sh: fix redirection in sh

Axel Davy (1):

- st/nine: Immediately upload user provided textures

Bas Nieuwenhuizen (3):

- radv: Only use 32 KiB per threadgroup on Stoney.
- radv: Set partial_vs_wave for pipelines with just GS, not tess.
- nir: Account for atomics in copy propagation.

Bruce Cherniak (1):

- gallium/swr: Fix multi-context sync fence deadlock.

Carsten Haitzler (Rasterman) (2):

- vc4: Use named parameters for the NEON inline asm.
- vc4: Declare the cpu pointers as being modified in NEON asm.

Danylo Piliaiev (1):

- glsl: Fix copying function's out to temp if dereferenced by array

Dave Airlie (3):

- dri_interface: add put shm image2 (v2)
- glx: add support for putimageshm2 path (v2)
- gallium: use put image shm2 path (v2)

Dylan Baker (4):

- meson: allow building dri driver without window system if osmesa is classic
- meson: fix swr KNL build
- meson: Fix compiler checks for SWR with ICC
- meson: Add warnings and errors when using ICC

Emil Velikov (4):

- docs: add sha256 checksums for 18.3.2
- cherry-ignore: radv: Fix multiview depth clears
- cherry-ignore: spirv: Handle arbitrary bit sizes for deref array indices
- cherry-ignore: WARNING: Commit XXX lists invalid sha

Eric Anholt (2):

- vc4: Don't leak the GPU fd for renderonly usage.
- vc4: Enable NEON asm on meson cross-builds.

Eric Engestrom (2):

- configure: EGL requirements only apply if EGL is built
- meson/vdpau: add missing soversion

Iago Toral Quiroga (1):

- anv/device: fix maximum number of images supported

Jason Ekstrand (3):

- anv/nir: Rework arguments to apply_pipeline_layout
- anv: Only parse pImmutableSamplers if the descriptor has samplers
- nir/xfb: Fix offset accounting for dvec3/4

Karol Herbst (2):

- nv50/ir: disable tryCollapseChainedMULs in ConstantFolding for precise instructions
- glsl/lower_output_reads: set invariant and precise flags on temporaries

Lionel Landwerlin (1):

- anv: fix invalid binding table index computation

Marek Olšák (4):

- radeonsi: also apply the GS hang workaround to draws without tessellation
- radeonsi: fix a u_blitter crash after a shader with FBFETCH
- radeonsi: fix rendering to tiny viewports where the viewport center is > 8K
- st/mesa: purge framebuffers when unbinding a context

Niklas Haas (1):

- radv: correctly use vulkan 1.0 by default

Pierre Moreau (1):

- meson: Fix with_gallium_icd to with_opencl_icd

Rob Clark (1):

- loader: fix the no-modifiers case

Samuel Pitoiset (1):

- radv: clean up setting partial_es_wave for distributed tess on VI

Timothy Arceri (5):

- ac/nir_to_llvm: fix interpolateAt* for arrays
- ac/nir_to_llvm: fix clamp shadow reference for more hardware
- radv/ac: fix some fp16 handling
- glsl: use remap location when serialising uniform program resource data
- glsl: Copy function out to temp if we don't directly ref a variable

Tomeu Vizoso (1):

- etnaviv: Consolidate buffer references from framebuffers

Vinson Lee (1):

- meson: Fix typo.

4.60 Mesa 18.3.2 Release Notes / January 17, 2019

Mesa 18.3.2 is a bug fix release which fixes bugs found since the 18.3.1 release.

Mesa 18.3.2 implements the OpenGL 4.5 API, but the version reported by glGetString(GL_VERSION) or glGetIntegerv(GL_MAJOR_VERSION) / glGetIntegerv(GL_MINOR_VERSION) depends on the particular driver being used. Some drivers don't support all the features required in OpenGL 4.5. OpenGL 4.5 is **only** available if requested at context creation. Compatibility contexts may report a lower version depending on each driver.

4.60.1 SHA256 checksums

```
1cde4fafd40cd1ad4ee3a13b364b7a0175a08b7afdd127fb46f918c1e1dfd4b0  mesa-18.3.2.tar.gz
f7ce7181c07b6d8e0132da879af1729523a6c8aa87f79a9d59dfd064024cfb35  mesa-18.3.2.tar.xz
```

4.60.2 New features

None

4.60.3 Bug fixes

- Bug 106595 - [RADV] Rendering distortions only when MSAA is enabled
- Bug 107728 - Wrong background in Sascha Willem's Multisampling Demo
- Bug 108114 - [vulkancts] new VK_KHR_16bit_storage tests fail.
- Bug 108116 - [vulkancts] stencil partial clear tests fail.
- Bug 108624 - [regression][bisected] "nir: Copy propagation between blocks" regression
- Bug 108910 - Vkd3d test failure test_multisample_array_texture()
- Bug 108911 - Vkd3d test failure test_clear_render_target_view()
- Bug 108943 - Build fails on ppc64le with meson
- Bug 109072 - GPU hang in blender 2.80
- Bug 109081 - [bisected] [HSW] Regression in clipping.user_defined.clip_* vulkancts tests
- Bug 109151 - [KBL-G][vulkan] dEQP-VK.texture.explicit_lod.2d.sizes.31x55_nearest_linear_mipmap_nearest_repeat failed verification.
- Bug 109202 - nv50_ir.cpp:749:19: error: cannot use typeid with -fno-rtti
- Bug 109204 - [regression, bisected] retroarch's crt-royale shader crash radv

4.60.4 Changes

Alex Deucher (3):

- pci_ids: add new vega10 pci ids
- pci_ids: add new vega20 pci id
- pci_ids: add new VegaM pci id

Alexander von Gluck IV (1):

- egl/haiku: Fix reference to disp vs dpy

Andres Gomez (2):

- glsl: correct typo in GLSL compilation error message
- glsl/linker: specify proper direction in location aliasing error

Axel Davy (3):

- st/nine: Fix volumetexture dtor on ctor failure
- st/nine: Bind src not dst in nine_context_box_upload
- st/nine: Add src reference to nine_context_range_upload

Bas Nieuwenhuizen (5):

- radv: Do a cache flush if needed before reading predicates.
- radv: Implement buffer stores with less than 4 components.

- anv/android: Do not reject storage images.
- radv: Fix rasterization precision bits.
- spirv: Fix matrix parameters in function calls.

Caio Marcelo de Oliveira Filho (3):

- nir: properly clear the entry sources in copy_prop_vars
- nir: properly find the entry to keep in copy_prop_vars
- nir: remove dead code from copy_prop_vars

Dave Airlie (2):

- radv/xfb: fix counter buffer bounds checks.
- virgl/vtest: fix front buffer flush with protocol version 0.

Dylan Baker (6):

- meson: Fix ppc64 little endian detection
- meson: Add support for gnu hurd
- meson: Add toggle for glx-direct
- meson: Override C++ standard to gnu++11 when building with altivec on ppc64
- meson: Error out if building nouveau and using LLVM without rtti
- autotools: Remove tegra vdpau driver

Emil Velikov (12):

- docs: add sha256 checksums for 18.3.1
- bin/get-pick-list.sh: rework handing of sha nominations
- bin/get-pick-list.sh: warn when commit lists invalid sha
- cherry-ignore: meson: libfreedreno depends upon libdrm (for fence support)
- glx: mandate xf86vidmode only for "drm" dri platforms
- meson: don't require glx/egl/gbm with gallium drivers
- pipe-loader: meson: reference correct library
- TODO: glx: meson: build dri based glx tests, only with -Dglx=dri
- glx: meson: drop includes from a link-only library
- glx: meson: wire up the dispatch-index-check test
- glx/test: meson: assorted include fixes
- Update version to 18.3.2

Eric Anholt (6):

- v3d: Fix a leak of the transfer helper on screen destroy.
- vc4: Fix a leak of the transfer helper on screen destroy.
- v3d: Fix a leak of the disassembled instruction string during debug dumps.
- v3d: Make sure that a thrsw doesn't split a multop from its umul24.
- v3d: Add missing flagging of SYNCB as a TSY op.

- gallium/ttn: Fix setup of outputs_written.

Erik Faye-Lund (2):

- virgl: wrap vertex element state in a struct
- virgl: work around bad assumptions in virglrenderer

Francisco Jerez (5):

- intel/fs: Handle source modifiers in lower_integer_multiplication().
- intel/fs: Implement quad swizzles on ICL+.
- intel/fs: Fix bug in lower_simd_width while splitting an instruction which was already split.
- intel/eu/gen7: Fix brw_MOV() with DF destination and strided source.
- intel/fs: Respect CHV/BXT regioning restrictions in copy propagation pass.

Ian Romanick (2):

- i965/vec4/dce: Don't narrow the write mask if the flags are used
- Revert "nir/lower_indirect: Bail early if modes == 0"

Jan Vesely (1):

- clover: Fix build after clang r348827

Jason Ekstrand (6):

- nir/constant_folding: Fix source bit size logic
- intel/blorp: Be more conservative about copying clear colors
- spirv: Handle any bit size in vector_insert/extract
- anv/apply_pipeline_layout: Set the cursor in lower_res_reindex_intrinsic
- spirv: Sign-extend array indices
- intel/peephole_ffma: Fix swizzle propagation

Karol Herbst (1):

- nv50/ir: fix use-after-free in ConstantFolding::visit

Kirill Burtsev (1):

- loader: free error state, when checking the drawable type

Lionel Landwerlin (5):

- anv: don't do partial resolve on layer > 0
- i965: include draw_params/derived_draw_params for VF cache workaround
- i965: add CS stall on VF invalidation workaround
- anv: explictly specify format for blorp ccs/mcs op
- anv: flush fast clear colors into compressed surfaces

Marek Olšák (1):

- st/mesa: don't leak pipe_surface if pipe_context is not current

Mario Kleiner (1):

- radeonsi: Fix use of 1- or 2- component GL_DOUBLE vbo's.

Nicolai Hähnle (1):

- meson: link LLVM 'native' component when LLVM is available

Rhys Perry (3):

- radv: don't set surf_index for stencil-only images
- ac/nir,radv,radeonsi/nir: use correct indices for interpolation intrinsics
- ac: split 16-bit ssbo loads that may not be dword aligned

Rob Clark (2):

- freedreno/drm: fix memory leak
- mesa/st/nir: fix missing nir_compact_varyings

Samuel Pitoiset (1):

- radv: switch on EOP when primitive restart is enabled with triangle strips

Timothy Arceri (2):

- tgsi/scan: fix loop exit point in tgsi_scan_tess_ctrl()
- tgsi/scan: correctly walk instructions in tgsi_scan_tess_ctrl()

Vinson Lee (2):

- meson: Fix typo.
- meson: Fix libsensors detection.

4.61 Mesa 18.2.8 Release Notes / December 27, 2018

Mesa 18.2.8 is a bug fix release which fixes bugs found since the 18.2.7 release.

Mesa 18.2.8 implements the OpenGL 4.5 API, but the version reported by glGetString(GL_VERSION) or glGetIntegerv(GL_MAJOR_VERSION) / glGetIntegerv(GL_MINOR_VERSION) depends on the particular driver being used. Some drivers don't support all the features required in OpenGL 4.5. OpenGL 4.5 is **only** available if requested at context creation. Compatibility contexts may report a lower version depending on each driver.

4.61.1 SHA256 checksums

```
77512edc0a84e19c7131a0e2e5ebf1beaf1494dc4b71508fcc92d06d65f9f4f5  mesa-18.2.8.tar.gz
1d2ed9fd435d86d95b7215b287258d3e6b1180293a36f688e5a2efc18298d863  mesa-18.2.8.tar.xz
```

4.61.2 New features

None

4.61.3 Bug fixes

- Bug 108114 - [vulkancts] new VK_KHR_16bit_storage tests fail.
- Bug 108116 - [vulkancts] stencil partial clear tests fail.
- Bug 108910 - Vkd3d test failure test_multisample_array_texture()

- Bug 108911 - Vkd3d test failure test_clear_render_target_view()
- Bug 109081 - [bisected] [HSW] Regression in clipping.user_defined.clip_* vulkancts tests

4.61.4 Changes

Alex Deucher (3):

- pci_ids: add new vega10 pci ids
- pci_ids: add new vega20 pci id
- pci_ids: add new VegaM pci id

Axel Davy (3):

- st/nine: Fix volumetexture dtor on ctor failure
- st/nine: Bind src not dst in nine_context_box_upload
- st/nine: Add src reference to nine_context_range_upload

Caio Marcelo de Oliveira Filho (1):

- nir: properly clear the entry sources in copy_prop_vars

Dylan Baker (1):

- meson: Fix ppc64 little endian detection

Emil Velikov (9):

- glx: mandate xf86vidmode only for "drm" dri platforms
- bin/get-pick-list.sh: rework handing of sha nominations
- bin/get-pick-list.sh: warn when commit lists invalid sha
- meson: don't require glx/egl/gbm with gallium drivers
- pipe-loader: meson: reference correct library
- TODO: glx: meson: build dri based glx tests, only with -Dglx=dri
- glx: meson: drop includes from a link-only library
- glx: meson: wire up the dispatch-index-check test
- glx/test: meson: assorted include fixes

Eric Anholt (2):

- v3d: Make sure that a thrsw doesn't split a multop from its umul24.
- v3d: Add missing flagging of SYNCB as a TSY op.

Erik Faye-Lund (2):

- virgl: wrap vertex element state in a struct
- virgl: work around bad assumptions in virglrenderer

Iago Toral Quiroga (1):

- intel/compiler: do not copy-propagate strided regions to ddx/ddy arguments

Ian Romanick (2):

- i965/vec4/dce: Don't narrow the write mask if the flags are used

- Revert "nir/lower_indirect: Bail early if modes == 0"

Jan Vesely (1):

- clover: Fix build after clang r348827

Jason Ekstrand (1):

- nir/constant_folding: Fix source bit size logic

Jon Turney (1):

- glx: Fix compilation with GLX_USE_WINDOWSGL

Juan A. Suarez Romero (7):

- docs: add sha256 checksums for 18.2.7
- cherry-ignore: add explicit 18.3 only nominations
- cherry-ignore: meson: libfreedreno depends upon libdrm (for fence support)
- cherry-ignore: radv: Fix multiview depth clears
- cherry-ignore: nir: properly find the entry to keep in copy_prop_vars
- cherry-ignore: intel/compiler: move nir_lower_bool_to_int32 before nir_lower_locals_to_regs
- Update version to 18.2.8

Kirill Burtsev (1):

- loader: free error state, when checking the drawable type

Lionel Landwerlin (1):

- anv: don't do partial resolve on layer > 0

Rhys Perry (2):

- radv: don't set surf_index for stencil-only images
- ac: split 16-bit ssbo loads that may not be dword aligned

Rob Clark (1):

- mesa/st/nir: fix missing nir_compact_varyings

Samuel Pitoiset (1):

- radv: switch on EOP when primitive restart is enabled with triangle strips

Vinson Lee (2):

- meson: Fix typo.
- meson: Fix libsensors detection.

4.62 Mesa 18.2.7 Release Notes / December 13, 2018

Mesa 18.2.7 is a bug fix release which fixes bugs found since the 18.2.6 release.

Mesa 18.2.7 implements the OpenGL 4.5 API, but the version reported by glGetString(GL_VERSION) or glGetIntegerv(GL_MAJOR_VERSION) / glGetIntegerv(GL_MINOR_VERSION) depends on the particular driver being used. Some drivers don't support all the features required in OpenGL 4.5. OpenGL 4.5 is **only** available if requested at context creation. Compatibility contexts may report a lower version depending on each driver.

4.62.1 SHA256 checksums

```
092351cfbcd430ec595fbd3a3d8d253fd62c29074e1740d7198b00289ab400f8  mesa-18.2.7.tar.gz
9c7b02560d89d77ca279cd21f36ea9a49e9ffc5611f6fe35099357d744d07ae6  mesa-18.2.7.tar.xz
```

4.62.2 New features

None

4.62.3 Bug fixes

- Bug 106577 - broken rendering with nine and nouveau (GM107)
- Bug 108245 - RADV/Vega: Low mip levels of large BCn textures get corrupted by vkCmdCopyBufferToImage
- Bug 108311 - Query buffer object support is broken on r600.
- Bug 108894 - [anv] vkCmdCopyBuffer() and vkCmdCopyQueryPoolResults() write-after-write hazard
- Bug 108909 - Vkd3d test failure test_resolve_non_issued_query_data()
- Bug 108914 - blocky shadow artifacts in The Forest with DXVK, RADV_DEBUG=nohiz fixes this
- Bug 108925 - vkCmdCopyQueryPoolResults(VK_QUERY_RESULT_WAIT_BIT) for timestamps with large query count hangs

4.62.4 Changes

Alex Smith (1):

- radv: Flush before vkCmdWriteTimestamp() if needed

Bas Nieuwenhuizen (4):

- radv: Align large buffers to the fragment size.
- radv: Clamp gfx9 image view extents to the allocated image extents.
- radv/android: Mark android WSI image as shareable.
- radv/android: Use buffer metadata to determine scanout compat.

Dave Airlie (2):

- r600: make suballocator 256-bytes align
- radv: use 3d shader for gfx9 copies if dst is 3d

Emil Velikov (2):

- egl/wayland: bail out when drmGetMagic fails
- egl/wayland: plug memory leak in drm_handle_device()

Eric Anholt (3):

- v3d: Fix a leak of the transfer helper on screen destroy.
- vc4: Fix a leak of the transfer helper on screen destroy.
- v3d: Fix a leak of the disassembled instruction string during debug dumps.

Eric Engestrom (3):

- anv: correctly use vulkan 1.0 by default
- wsi/display: fix mem leak when freeing swapchains
- vulkan/wsi: fix s/,/;/ typo

Gurchetan Singh (3):

- virgl: quadruple command buffer size
- virgl: avoid large inline transfers
- virgl: don't mark buffers as unclean after a write

Juan A. Suarez Romero (4):

- docs: add sha256 checksums for 18.2.6
- cherry-ignore: freedreno: Fix autotools build.
- cherry-ignore: mesa: Revert INTEL_fragment_shader_ordering support
- Update version to 18.2.7

Karol Herbst (1):

- nv50,nvc0: Fix gallium nine regression regarding sampler bindings

Lionel Landwerlin (2):

- anv: flush pipeline before query result copies
- anv/query: flush render target before copying results

Michal Srb (2):

- gallium: Constify drisw_loader_funcs struct
- drisw: Use separate drisw_loader_funcs for shm

Nicolai Hähnle (2):

- egl/wayland: rather obvious build fix
- meson: link LLVM 'native' component when LLVM is available

Samuel Pitoiset (1):

- radv: rework the TC-compat HTILE hardware bug with COND_EXEC

Thomas Hellstrom (2):

- st/xa: Fix a memory leak
- winsys/svga: Fix a memory leak

Tobias Klausmann (1):

- amd/vulkan: meson build - use radv_deps for libvulkan_radeon

Vinson Lee (1):

- st/xvmc: Add X11 include path.

4.63 Mesa 18.3.1 Release Notes / December 11, 2018

Mesa 18.3.1 is a bug fix release which fixes bugs found since the 18.3.0 release.

Mesa 18.3.0 implements the OpenGL 4.5 API, but the version reported by glGetString(GL_VERSION) or glGetIntegerv(GL_MAJOR_VERSION) / glGetIntegerv(GL_MINOR_VERSION) depends on the particular driver being used. Some drivers don't support all the features required in OpenGL 4.5. OpenGL 4.5 is **only** available if requested at context creation. Compatibility contexts may report a lower version depending on each driver.

4.63.1 SHA256 checksums

```
256d0c3d88e380c1b8e3fc5c6ac34001e3b7c30458b8b852407ec68b8ccd9fda  mesa-18.3.1.tar.gz
5b1f827d28684a25f6657289f8b7d47ac56395988c7ac23e0ec9a62b644bdc63  mesa-18.3.1.tar.xz
```

4.63.2 New features

None

4.63.3 Bug fixes

None

4.63.4 Changes

Emil Velikov (2):

- docs: add sha256 checksums for 18.3.0
- Update version to 18.3.1

Jason Ekstrand (1):

- anv,radv: Disable VK_EXT_pci_bus_info

4.64 Mesa 18.3.0 Release Notes / December 7, 2018

Mesa 18.3.0 is a new development release. People who are concerned with stability and reliability should stick with a previous release or wait for Mesa 18.3.1.

Mesa 18.3.0 implements the OpenGL 4.5 API, but the version reported by glGetString(GL_VERSION) or glGetIntegerv(GL_MAJOR_VERSION) / glGetIntegerv(GL_MINOR_VERSION) depends on the particular driver being used. Some drivers don't support all the features required in OpenGL 4.5. OpenGL 4.5 is **only** available if requested at context creation. Compatibility contexts may report a lower version depending on each driver.

libwayland-egl is now distributed by Wayland (since 1.15, see announcement), and has been removed from Mesa in this release. Make sure you're using an up-to-date version of Wayland to keep the functionality.

4.64.1 SHA256 checksums

```
17a124d4dbc712505d22a7815c9b0cee22214c96c8abb91539a2b1351e38a000  mesa-18.3.0.tar.gz
b63f947e735d6ef3dfaa30c789a9adfbae18aea671191eaacde95a18c17fc38a  mesa-18.3.0.tar.xz
```

4.64.2 New features

Note: some of the new features are only available with certain drivers.

- GL_AMD_depth_clamp_separate on r600, radeonsi.
- GL_AMD_framebuffer_multisample_advanced on radeonsi.
- GL_AMD_gpu_shader_int64 on i965, nvc0, radeonsi.
- GL_AMD_multi_draw_indirect on all GL 4.x drivers.
- GL_AMD_query_buffer_object on i965, nvc0, r600, radeonsi.
- GL_EXT_disjoint_timer_query on radeonsi and most other Gallium drivers (ES extension)
- GL_EXT_texture_compression_s3tc on all drivers (ES extension)
- GL_EXT_vertex_attrib_64bit on i965, nvc0, radeonsi.
- GL_EXT_window_rectangles on radeonsi.
- GL_KHR_texture_compression_astc_sliced_3d on radeonsi.
- GL_NV_fragment_shader_interlock on i965.
- EGL_EXT_device_base for all drivers.
- EGL_EXT_device_drm for all drivers.
- EGL_MESA_device_software for all drivers.

4.64.3 Bug fixes

- Bug 13728 - [G965] Some objects in Neverwinter Nights Linux version not displayed correctly
- Bug 91433 - piglit.spec.arb_depth_buffer_float.fbo-depth-gl_depth_component32f-copypixels fails
- Bug 93355 - [BXT,SKLGT4e] intermittent ext_framebuffer_multisample.accuracy fails
- Bug 94957 - dEQP failures on llvmpipe
- Bug 98699 - "float[a+++4 ? 1:1] f;" crashes glsl_compiler
- Bug 99507 - Corrupted frame contents with Vulkan version of DOTA2, Talos Principle and Sascha Willems' demos when they're run Vsynched in fullscreen
- Bug 99730 - Metro Redux game(s) needs override for midshader extension declaration
- Bug 100200 - Default Unreal Engine 4 frag shader fails to compile
- Bug 101247 - Mesa fails to link GLSL programs with unused output blocks
- Bug 102597 - [Regression] mpv, high rendering times (two to three times higher)
- Bug 103241 - Anv crashes when using 64-bit vertex inputs
- Bug 104602 - [apitrace] Graphical artifacts in Civilization VI on RX Vega

- Bug 104809 - anv: DOOM 2016 and Wolfenstein II:The New Colossus crash due to not having depthBoundsTest
- Bug 104926 - swrast: Mesa 17.3.3 produces: HW cursor for format 875713089 not supported
- Bug 105333 - [gallium-nine] missing geometry after commit ac: replace ac_build_kill with ac_build_kill_if_false
- Bug 105371 - r600_shader_from_tgsi - GPR limit exceeded - shader requires 360 registers
- Bug 105731 - linker error "fragment shader input … has no matching output in the previous stage" when previous stage's output declaration in a separate shader object
- Bug 105904 - Needed to delete mesa shader cache after driver upgrade for 32 bit wine vulkan programs to work.
- Bug 105975 - i965 always reports 0 viewport subpixel bits
- Bug 106231 - llvmpipe blends produce bad code after llvm patch https://reviews.llvm.org/D44785
- Bug 106283 - Shader replacements works only for limited use cases
- Bug 106577 - broken rendering with nine and nouveau (GM107)
- Bug 106833 - glLinkProgram is expected to fail when vertex attribute aliasing happens on ES3.0 context or later
- Bug 106865 - [GLK] piglit.spec.ext_framebuffer_multisample.accuracy stencil tests fail
- Bug 106980 - Basemark GPU vulkan benchmark hangs on GFX9
- Bug 106997 - [Regression]. Dying light game is crashing on latest mesa
- Bug 107088 - [GEN8+] Hang when discarding a fragment if dual source blending is enabled but shader doesn't support it
- Bug 107098 - Segfault after munmap(kms_sw_dt->ro_mapped)
- Bug 107212 - Dual-Core CPU E5500 / G45: RetroArch with reicast core results in corrupted graphics
- Bug 107223 - [GEN9+] 50% perf drop in SynMark Fill* tests (E2E RBC gets disabled?)
- Bug 107276 - radv: OpBitfieldUExtract returns incorrect result when count is zero
- Bug 107280 - [DXVK] Batman: Arkham City with tessellation enabled hangs on SKL GT4
- Bug 107313 - Meson instructions on web site are non-optimal
- Bug 107359 - [Regression] [bisected] [OpenGL CTS] [SKL,BDW] KHR-GL46.texture_barrier*-texels, GTF-GL46.gtf21.GL2FixedTests.buffer_corners.buffer_corners, and GTF-GL46.gtf21.GL2FixedTests.stencil_plane_corners.stencil_plane_corners fail with some configuration
- Bug 107460 - radv: OpControlBarrier does not always work correctly (bisected)
- Bug 107477 - [DXVK] Setting high shader quality in GTA V results in LLVM error
- Bug 107483 - DispatchSanity_test.GL31_CORE regression
- Bug 107487 - [intel] [tools] intel gpu tools don't honor -D tools=[]
- Bug 107488 - gl.h:2090: error: redefinition of typedef 'GLeglImageOES'
- Bug 107510 - [GEN8+] up to 10% perf drop on several 3D benchmarks
- Bug 107511 - KHR/khrplatform.h not always installed when needed
- Bug 107524 - Broken packDouble2x32 at llvmpipe
- Bug 107544 - intel/decoder: out of bounds group_iter
- Bug 107547 - shader crashing glsl_compiler (uniform block assigned to vec2, then component substraced by 1)
- Bug 107550 - "0[2]" as function parameter hits assert

- Bug 107563 - [RADV] Broken rendering in Unity demos
- Bug 107565 - TypeError: __init__() got an unexpected keyword argument 'future_imports'
- Bug 107579 - [SNB] The graphic corruption when we reuse the GS compiled and used for TFB when statebuffer contain magic trash in the unused space
- Bug 107601 - Rise of the Tomb Raider Segmentation Fault when the game starts
- Bug 107610 - Dolphin emulator mis-renders shadow overlay in Super Mario Sunshine
- Bug 107626 - [SNB] The graphical corruption and GPU hang occur sometimes on the piglit test "arb_texture_multisample-large-float-texture" with parameter –fp16
- Bug 107658 - [Regression] [bisected] [OpenGLES CTS] KHR-GLES3.packed_pixels.*rectangle.r*8_snorm
- Bug 107734 - [GLSL] glsl-fface-invariant, glsl-fcoord-invariant and glsl-pcoord-invariant should fail
- Bug 107745 - [bisected] [bdw bsw] piglit.spec.arb_fragment_shader_interlock.arb_fragment_shader_interlock-image-load-store failure
- Bug 107760 - GPU Hang when Playing DiRT 3 Complete Edition using Steam Play with DXVK
- Bug 107765 - [regression] Batman Arkham City crashes with DXVK under wine
- Bug 107772 - Mesa preprocessor matches if(def)s & endifs incorrectly
- Bug 107779 - Access violation with some games
- Bug 107786 - [DXVK] MSAA reflections are broken in GTA V
- Bug 107806 - glsl_get_natural_size_align_bytes() ABORT with GfxBench Vulkan AztecRuins
- Bug 107810 - The 'va_end' call is missed after 'va_copy' in 'util_vsnprintf' function under windows
- Bug 107832 - Gallium picking A16L16 formats when emulating INTENSITY16 conflicts with mesa
- Bug 107843 - 32bit Mesa build failes with meson.
- Bug 107856 - i965 incorrectly calculates the number of layers for texture views (assert)
- Bug 107857 - GPU hang - GS_EMIT without shader outputs
- Bug 107865 - swr fail to build with llvm-libs 6.0.1
- Bug 107869 - u_thread.h:87:4: error: use of undeclared identifier 'cpu_set_t'
- Bug 107870 - Undefined symbols for architecture x86_64: "_util_cpu_caps"
- Bug 107879 - crash happens when link program
- Bug 107891 - [wine, regression, bisected] RAGE, Wolfenstein The New Order hangs in menu
- Bug 107923 - build_id.c:126: multiple definition of 'build_id_length'
- Bug 107926 - [anv] Rise of the Tomb Raider always misrendering, segfault and gpu hang.
- Bug 107941 - GPU hang and system crash with Dota 2 using Vulkan
- Bug 107971 - SPV_GOOGLE_hlsl_functionality1 / SPV_GOOGLE_decorate_string
- Bug 108012 - Compiler crashes on access of non-existent member incremental operations
- Bug 108024 - [Debian Stretch]Fail to build because "xcb_randr_lease_t"
- Bug 108082 - warning: unknown warning option '-Wno-format-truncation' [-Wunknown-warning-option]
- Bug 108109 - [GLSL] no-overloads.vert fails
- Bug 108112 - [vulkancts] some of the coherent memory tests fail.

- Bug 108113 - [vulkancts] r32g32b32 transfer operations not implemented
- Bug 108115 - [vulkancts] dEQP-VK.subgroups.vote.graphics.subgroupallequal.* fails
- Bug 108164 - [radv] VM faults since 5d6a560a2986c9ab421b3c7904d29bb7bc35e36f
- Bug 108245 - RADV/Vega: Low mip levels of large BCn textures get corrupted by vkCmdCopyBufferToImage
- Bug 108272 - [polaris10] opencl-mesa: Anything using OpenCL segfaults, XFX Radeon RX 580
- Bug 108311 - Query buffer object support is broken on r600.
- Bug 108319 - [GLK BXT BSW] Assertion in piglit.spec.arb_gpu_shader_fp64.execution.built-in-functions.vs-sign-sat-neg-abs
- Bug 108491 - Commit baa38c14 causes output issues on my VEGA with RADV
- Bug 108524 - [RADV] GPU lockup on event synchronization
- Bug 108530 - (mesa-18.3) [Tracker] Mesa 18.3 Release Tracker
- Bug 108532 - make check nir_copy_prop_vars_test.store_store_load_different_components regression
- Bug 108560 - Mesa 32 is built without sse
- Bug 108595 - ir3_compiler valgrind build error
- Bug 108617 - [deqp] Mesa fails conformance for egl_ext_device
- Bug 108630 - [G965] piglit.spec.!opengl 1_2.tex3d-maxsize spins forever
- Bug 108635 - Mesa master commit 68dc591af16ebb36814e4c187e4998948103c99c causes XWayland to segfault
- Bug 108713 - Gallium: use after free with transform feedback
- Bug 108829 - [meson] libglapi exports internal API
- Bug 108894 - [anv] vkCmdCopyBuffer() and vkCmdCopyQueryPoolResults() write-after-write hazard
- Bug 108909 - Vkd3d test failure test_resolve_non_issued_query_data()
- Bug 108914 - blocky shadow artifacts in The Forest with DXVK, RADV_DEBUG=nohiz fixes this

4.64.4 Changes

- TBD

4.65 Mesa 18.2.6 Release Notes / November 28, 2018

Mesa 18.2.6 is a bug fix release which fixes bugs found since the 18.2.5 release.

Mesa 18.2.6 implements the OpenGL 4.5 API, but the version reported by glGetString(GL_VERSION) or glGetIntegerv(GL_MAJOR_VERSION) / glGetIntegerv(GL_MINOR_VERSION) depends on the particular driver being used. Some drivers don't support all the features required in OpenGL 4.5. OpenGL 4.5 is **only** available if requested at context creation. Compatibility contexts may report a lower version depending on each driver.

4.65.1 SHA256 checksums

```
e0ea1236dbc6c412b02e1b5d7f838072525971a6630246fa82ae4466a6d8a587  mesa-18.2.6.tar.gz
9ebafa4f8249df0c718e93b9ca155e3593a1239af303aa2a8b0f2056a7efdc12  mesa-18.2.6.tar.xz
```

4.65.2 New features

None

4.65.3 Bug fixes

- Bug 107626 - [SNB] The graphical corruption and GPU hang occur sometimes on the piglit test "arb_texture_multisample-large-float-texture" with parameter –fp16
- Bug 107856 - i965 incorrectly calculates the number of layers for texture views (assert)
- Bug 108630 - [G965] piglit.spec.!opengl 1_2.tex3d-maxsize spins forever
- Bug 108713 - Gallium: use after free with transform feedback
- Bug 108829 - [meson] libglapi exports internal API

4.65.4 Changes

Andrii Simiklit (1):

- i965/batch: avoid reverting batch buffer if saved state is an empty

Bas Nieuwenhuizen (1):

- radv: Fix opaque metadata descriptor last layer.

Brian Paul (1):

- scons/svga: remove opt from the list of valid build types

Danylo Piliaiev (1):

- i965: Fix calculation of layers array length for isl_view

Dylan Baker (2):

- meson: Don't set -Wall
- meson: Don't force libva to required from auto

Emil Velikov (13):

- bin/get-pick-list.sh: simplify git oneline printing
- bin/get-pick-list.sh: prefix output with "[stable] "
- bin/get-pick-list.sh: handle "typod" usecase.
- bin/get-pick-list.sh: handle the fixes tag
- bin/get-pick-list.sh: tweak the commit sha matching pattern
- bin/get-pick-list.sh: flesh out is_sha_nomination
- bin/get-pick-list.sh: handle fixes tag with missing colon
- bin/get-pick-list.sh: handle unofficial "broken by" tag
- bin/get-pick-list.sh: use test instead of [ ]
- bin/get-pick-list.sh: handle reverts prior to the branchpoint
- travis: drop unneeded x11proto-xf86vidmode-dev
- glx: make xf86vidmode mandatory for direct rendering

- travis: adding missing x11-xcb for meson+vulkan

Eric Anholt (1):

- vc4: Make sure we make ro scanout resources for create_with_modifiers.

Eric Engestrom (5):

- meson: only run vulkan's meson.build when building vulkan
- gbm: remove unnecessary meson include
- meson: fix wayland-less builds
- egl: add missing glvnd entrypoint for EGL_ANDROID_blob_cache
- glapi: add missing visibility args

Erik Faye-Lund (1):

- mesa/main: remove bogus error for zero-sized images

Gert Wollny (3):

- mesa: Reference count shaders that are used by transform feedback objects
- r600: clean up the GS ring buffers when the context is destroyed
- glsl: free or reuse memory allocated for TF varying

Jason Ekstrand (2):

- nir/lower_alu_to_scalar: Don't try to lower unpack_32_2x16
- anv: Put robust buffer access in the pipeline hash

Juan A. Suarez Romero (6):

- cherry-ignore: add explicit 18.3 only nominations
- cherry-ignore: intel/aub_viewer: fix dynamic state printing
- cherry-ignore: intel/aub_viewer: Print blend states properly
- cherry-ignore: mesa/main: fix incorrect depth-error
- docs: add sha256 checksums for 18.2.5
- Update version to 18.2.6

Karol Herbst (1):

- nir/spirv: cast shift operand to u32

Kenneth Graunke (1):

- i965: Add PCI IDs for new Amberlake parts that are Coffeelake based

Lionel Landwerlin (1):

- egl/dri: fix error value with unknown drm format

Marek Olšák (2):

- winsys/amdgpu: fix a buffer leak in amdgpu_bo_from_handle
- winsys/amdgpu: fix a device handle leak in amdgpu_winsys_create

Rodrigo Vivi (4):

- i965: Add a new CFL PCI ID.

- intel: aubinator: Adding missed platforms to the error message.
- intel: Introducing Amber Lake platform
- intel: Introducing Whiskey Lake platform

4.66 Mesa 18.2.5 Release Notes / November 15, 2018

Mesa 18.2.5 is a bug fix release which fixes bugs found since the 18.2.4 release.

Mesa 18.2.5 implements the OpenGL 4.5 API, but the version reported by glGetString(GL_VERSION) or glGetIntegerv(GL_MAJOR_VERSION) / glGetIntegerv(GL_MINOR_VERSION) depends on the particular driver being used. Some drivers don't support all the features required in OpenGL 4.5. OpenGL 4.5 is **only** available if requested at context creation. Compatibility contexts may report a lower version depending on each driver.

4.66.1 SHA256 checksums

```
dddc28928b6f4083a0d5120b58c1c8e2dc189ab5c14299c08a386607fdbbdce7  mesa-18.2.5.tar.gz
b12c32872832e5353155e1e8026e1f1ab75bba9dc5b178d712045684d26c2b73  mesa-18.2.5.tar.xz
```

4.66.2 New features

None

4.66.3 Bug fixes

- Bug 105731 - linker error "fragment shader input ... has no matching output in the previous stage" when previous stage's output declaration in a separate shader object
- Bug 107511 - KHR/khrplatform.h not always installed when needed
- Bug 107626 - [SNB] The graphical corruption and GPU hang occur sometimes on the piglit test "arb_texture_multisample-large-float-texture" with parameter –fp16
- Bug 108082 - warning: unknown warning option '-Wno-format-truncation' [-Wunknown-warning-option]
- Bug 108560 - Mesa 32 is built without sse

4.66.4 Changes

Andre Heider (1):

- st/nine: fix stack corruption due to ABI mismatch

Andrii Simiklit (1):

- i965/batch: don't ignore the 'brw_new_batch' call for a 'new batch'

Dylan Baker (2):

- meson: link gallium nine with pthreads
- meson: fix libatomic tests

Emil Velikov (2):

- egl/glvnd: correctly report errors when vendor cannot be found
- m4: add Werror when checking for compiler flags

Eric Engestrom (6):

- svga: add missing meson build dependency
- clover: add missing meson build dependency
- wsi/wayland: use proper VkResult type
- wsi/wayland: only finish() a successfully init()ed display
- configure: install KHR/khrplatform.h when needed
- meson: install KHR/khrplatform.h when needed

Gert Wollny (1):

- virgl/vtest-winsys: Use virgl version of bind flags

Jonathan Gray (1):

- intel/tools: include stdarg.h in error2aub

Juan A. Suarez Romero (4):

- docs: add sha256 checksums for 18.2.4
- cherry-ignore: add explicit 18.3 only nominations
- cherry-ignore: i965/batch: avoid reverting batch buffer if saved state is an empty
- Update version to 18.2.5

Lionel Landwerlin (1):

- anv/android: mark gralloc allocated BOs as external

Marek Olšák (3):

- ac: fix ac_build_fdiv for f64
- st/va: fix incorrect use of resource_destroy
- include: update GL & GLES headers (v2)

Matt Turner (2):

- util/ralloc: Switch from DEBUG to NDEBUG
- util/ralloc: Make sizeof(linear_header) a multiple of 8

Olivier Fourdan (1):

- wayland/egl: Resize EGL surface on update buffer for swrast

Rhys Perry (1):

- glsl_to_tgsi: don't create 64-bit integer MAD/FMA

Samuel Pitoiset (2):

- radv: disable conditional rendering for vkCmdCopyQueryPoolResults()
- radv: only expose VK_SUBGROUP_FEATURE_ARITHMETIC_BIT for VI+

Sergii Romantsov (1):

- autotools: library-dependency when no sse and 32-bit

Timothy Arceri (4):

- st/mesa: calculate buffer size correctly for packed uniforms
- st/glsl_to_nir: fix next_stage gathering
- nir: add glsl_type_is_integer() helper
- nir: don't pack varyings ints with floats unless flat

Vadym Shovkoplias (1):

- glsl/linker: Fix out variables linking during single stage

Vinson Lee (1):

- r600/sb: Fix constant logical operand in assert.

4.67 Mesa 18.2.4 Release Notes / October 31, 2018

Mesa 18.2.4 is a bug fix release which fixes bugs found since the 18.2.4 release.

Mesa 18.2.4 implements the OpenGL 4.5 API, but the version reported by glGetString(GL_VERSION) or glGetIntegerv(GL_MAJOR_VERSION) / glGetIntegerv(GL_MINOR_VERSION) depends on the particular driver being used. Some drivers don't support all the features required in OpenGL 4.5. OpenGL 4.5 is **only** available if requested at context creation. Compatibility contexts may report a lower version depending on each driver.

4.67.1 SHA256 checksums

```
968bfe78605e9397ddf244933b1fa62edb8429fc55aaec2ae7e20bb1c82abdea  mesa-18.2.4.tar.gz
621d1aebb57876d5b6a5d2dcf4eb7e0620e650c6fe5cf3655c65e243adc9cb4e  mesa-18.2.4.tar.xz
```

4.67.2 New features

None

4.67.3 Bug fixes

- Bug 107865 - swr fail to build with llvm-libs 6.0.1
- Bug 108272 - [polaris10] opencl-mesa: Anything using OpenCL segfaults, XFX Radeon RX 580
- Bug 108524 - [RADV] GPU lockup on event synchronization

4.67.4 Changes

Alex Smith (2):

- ac/nir: Use context-specific LLVM types
- anv: Fix sanitization of stencil state when the depth test is disabled

Alok Hota (2):

- swr/rast: ignore CreateElementUnorderedAtomicMemCpy

- swr/rast: fix intrinsic/function for LLVM 7 compatibility

Andres Rodriguez (1):

- radv: fix check for perftest options size

Bas Nieuwenhuizen (1):

- radv: Emit enqueued pipeline barriers on event write.

Connor Abbott (2):

- ac: Introduce ac_build_expand()
- ac: Fix loading a dvec3 from an SSBO

David McFarland (1):

- util: Change remaining uint32 cache ids to sha1

Dylan Baker (1):

- meson: don't require libelf for r600 without LLVM

Elie Tournier (1):

- gallium: Correctly handle no config context creation

Eric Engestrom (1):

- radv: s/abs/fabsf/ for floats

Jan Vesely (1):

- radeonsi: Bump number of allowed global buffers to 32

Jason Ekstrand (3):

- spirv: Use the right bit-size for spec constant ops
- blorp: Emit a dummy 3DSTATE_WM prior to 3DSTATE_WM_HZ_OP
- anv: Flag semaphore BOs as external

Juan A. Suarez Romero (3):

- docs: add sha256 checksums for 18.2.3
- cherry-ignore: Revert "anv/skylake: disable ForceThreadDispatchEnable"
- Update version to 18.2.4

Liviu Prodea (1):

- scons: Put to rest zombie texture_float build option.

Marek Olšák (1):

- radeonsi: fix a VGT hang with primitive restart on Polaris10 and later

Michel Dänzer (1):

- loader/dri3: Also wait for front buffer fence if we triggered it

Nanley Chery (1):

- intel/blorp: Define the clear value bounds for HiZ clears

Rob Clark (2):

- freedreno: fix inorder rendering case

- freedreno: don't flush when new and old pfb is identical

4.68 Mesa 18.2.3 Release Notes / October 19, 2018

Mesa 18.2.3 is a bug fix release which fixes bugs found since the 18.2.2 release.

Mesa 18.2.3 implements the OpenGL 4.5 API, but the version reported by glGetString(GL_VERSION) or glGetIntegerv(GL_MAJOR_VERSION) / glGetIntegerv(GL_MINOR_VERSION) depends on the particular driver being used. Some drivers don't support all the features required in OpenGL 4.5. OpenGL 4.5 is **only** available if requested at context creation. Compatibility contexts may report a lower version depending on each driver.

4.68.1 SHA256 checksums

```
0e13e2342eae74d8848df23595c4bb4b2f8874c9e1213b8466b1fbfa7ef99375  mesa-18.2.3.tar.gz
e2bf83c17e1abdecb1ee81af22652e27e9aa38f963e95e60f34275cc0376304f  mesa-18.2.3.tar.xz
```

4.68.2 New features

None

4.68.3 Bug fixes

- Bug 99507 - Corrupted frame contents with Vulkan version of DOTA2, Talos Principle and Sascha Willems' demos when they're run Vsynched in fullscreen
- Bug 107857 - GPU hang - GS_EMIT without shader outputs
- Bug 107926 - [anv] Rise of the Tomb Raider always misrendering, segfault and gpu hang.
- Bug 108012 - Compiler crashes on access of non-existent member incremental operations

4.68.4 Changes

Boyuan Zhang (1):

- st/va: use provided sizes and coords for vlVaGetImage

Dave Airlie (1):

- anv: add missing unlock in error path.

Dylan Baker (1):

- meson: Don't allow building EGL on Windows or MacOS

Emil Velikov (5):

- st/nine: do not double-close the fd on teardown
- egl: make eglSwapInterval a no-op for !window surfaces
- egl: make eglSwapBuffers* a no-op for !window surfaces
- vl/dri3: do full teardown on screen_destroy
- Revert "mesa: remove unnecessary 'sort by year' for the GL extensions"

Eric Engestrom (1):

- radv: add missing meson c++ visibility arguments

Fritz Koenig (1):

- i965: Replace checks for rb->Name with FlipY (v2)

Gert Wollny (1):

- virgl, vtest: Correct the transfer size calculation

Ilia Mirkin (4):

- glsl: fix array assignments of a swizzled vector
- nv50,nvc0: mark RGBX_UINT formats as renderable
- nv50,nvc0: guard against zero-size blits
- nvc0: fix blitting red to srgb8_alpha

Jason Ekstrand (7):

- nir/cf: Remove phi sources if needed in nir_handle_add_jump
- anv: Use separate MOCS settings for external BOs
- intel/fs: Fix a typo in need_matching_subreg_offset
- nir/from_ssa: Don't rewrite derefs destinations to registers
- anv/batch_chain: Don't start a new BO just for BATCH_BUFFER_START
- nir/alu_to_scalar: Use ssa_for_alu_src in hand-rolled expansions
- intel: Don't propagate conditional modifiers if a UD source is negated

Juan A. Suarez Romero (2):

- docs: add sha256 checksums for 18.2.2
- Update version to 18.2.3

Józef Kucia (1):

- radeonsi: avoid sending GS_EMIT in shaders without outputs

Marek Olšák (1):

- drirc: add a workaround for ARMA 3

Samuel Pitoiset (1):

- radv: add a workaround for a VGT hang with prim restart and strips

Tapani Pälli (1):

- glsl: do not attempt assignment if operand type not parsed correctly

Timothy Arceri (11):

- glsl: ignore trailing whitespace when define redefined
- util: disable cache if we have no build-id and timestamp is zero
- util: rename timestamp param in disk_cache_create()
- util: add disk_cache_get_function_identifier()
- radeonsi: use build-id when available for disk cache

- nouveau: use build-id when available for disk cache
- r600: use build-id when available for disk cache
- mesa/st: add force_compat_profile option to driconfig
- util: use force_compat_profile for Wolfenstein The Old Blood
- util: better handle program names from wine
- util: add drirc workarounds for RAGE

Vinson Lee (1):

- r600/sb: Fix constant-logical-operand warning.

4.69 Mesa 18.2.2 Release Notes / October 5, 2018

Mesa 18.2.2 is a bug fix release which fixes bugs found since the 18.2.1 release.

Mesa 18.2.2 implements the OpenGL 4.5 API, but the version reported by glGetString(GL_VERSION) or glGetIntegerv(GL_MAJOR_VERSION) / glGetIntegerv(GL_MINOR_VERSION) depends on the particular driver being used. Some drivers don't support all the features required in OpenGL 4.5. OpenGL 4.5 is **only** available if requested at context creation. Compatibility contexts may report a lower version depending on each driver.

4.69.1 SHA256 checksums

```
SHA256: c51711168971957037cc7e3e19e8abe1ec6eeab9cf236d419a1e7728a41cac8a  mesa-18.2.2.
↪tar.gz
SHA256: c3ba82b12a89d3d9fed2bdd96b4702dbb7ab675034650a8b1b718320daf073c4  mesa-18.2.2.
↪tar.xz
```

4.69.2 New features

None

4.69.3 Bug fixes

- Bug 104602 - [apitrace] Graphical artifacts in Civilization VI on RX Vega
- Bug 104926 - swrast: Mesa 17.3.3 produces: HW cursor for format 875713089 not supported
- Bug 107276 - radv: OpBitfieldUExtract returns incorrect result when count is zero
- Bug 107786 - [DXVK] MSAA reflections are broken in GTA V
- Bug 108024 - [Debian Stretch]Fail to build because "xcb_randr_lease_t"

4.69.4 Changes

Alex Deucher (1):

- pci_ids: add new polaris pci id

Andres Rodriguez (1):

- radv: only emit ZPASS_DONE for timestamp queries on gfx queues

Axel Davy (3):

- st/nine: Clamp RCP when 0*inf!=0
- st/nine: Avoid redundant SetCursorPos calls
- st/nine: Increase maximum number of temp registers

Dylan Baker (1):

- meson: Don't compile pipe loader with dri support when not using dri

Eric Anholt (1):

- vc4: Fix sin(0.0) and cos(0.0) accuracy to fix SDL rendering rotation.

Eric Engestrom (1):

- vulkan/wsi/display: check if wsi_swapchain_init() succeeded

Jason Ekstrand (1):

- anv,radv: Implement vkAcquireNextImage2

Juan A. Suarez Romero (2):

- docs: add sha256 checksums for 18.2.1
- Update version to 18.2.2

Leo Liu (1):

- radeon/uvd: use bitstream coded number for symbols of Huffman tables

Marek Olšák (2):

- glsl_to_tgsi: invert gl_SamplePosition.y for the default framebuffer
- radeonsi: NaN should pass kill_if

Maxime (1):

- vulkan: Disable randr lease for libxcb < 1.13

Michal Srb (1):

- st/dri: don't set queryDmaBufFormats/queryDmaBufModifiers if the driver does not implement it

Rhys Perry (2):

- nvc0: Update counter reading shaders to new NVC0_CB_AUX_MP_INFO
- nvc0: fix bindless multisampled images on Maxwell+

Samuel Iglesias Gonsálvez (1):

- anv: Add support for protected memory properties on anv_GetPhysicalDeviceProperties2()

Samuel Pitoiset (1):

- radv: use the resolve compute path if dest uses multiple layers

Stuart Young (1):

- docs: Update FAQ with respect to s3tc support

Timothy Arceri (1):

- radeonsi: add a workaround for bitfield_extract when count is 0

4.70 Mesa 18.1.8 Release Notes / September 24 2018

Mesa 18.1.9 is a bug fix release which fixes bugs found since the 18.1.8 release.

Mesa 18.1.9 implements the OpenGL 4.5 API, but the version reported by glGetString(GL_VERSION) or glGetIntegerv(GL_MAJOR_VERSION) / glGetIntegerv(GL_MINOR_VERSION) depends on the particular driver being used. Some drivers don't support all the features required in OpenGL 4.5. OpenGL 4.5 is **only** available if requested at context creation. Compatibility contexts may report a lower version depending on each driver.

4.70.1 SHA256 checksums

```
0f825dc834b1b3e3d9a6c3ce58b42977f0d9a248a7627a36dd3b313ffe41a499  mesa-18.1.9.tar.gz
55f5778d58a710a63d6635f000535768faf7db9e8144dc0f4fd1989f936c1a83  mesa-18.1.9.tar.xz
```

4.70.2 New features

None

4.70.3 Bug fixes

- Bug 103241 - Anv crashes when using 64-bit vertex inputs
- Bug 104926 - swrast: Mesa 17.3.3 produces: HW cursor for format 875713089 not supported
- Bug 107280 - [DXVK] Batman: Arkham City with tessellation enabled hangs on SKL GT4
- Bug 107772 - Mesa preprocessor matches if(def)s & endifs incorrectly
- Bug 107779 - Access violation with some games
- Bug 107810 - The 'va_end' call is missed after 'va_copy' in 'util_vsnprintf' function under windows

4.70.4 Changes

Andrii Simiklit (4):

- apple/glx/log: added missing va_end() after va_copy()
- mesa/util: don't use the same 'va_list' instance twice
- mesa/util: don't ignore NULL returned from 'malloc'
- mesa/util: add missing va_end() after va_copy()

Bas Nieuwenhuizen (4):

- radv: Use build ID if available for cache UUID.
- radv: Only allow 16 user SGPRs for compute on GFX9+.
- radv: Set the user SGPR MSB for Vega.
- radv: Fix driver UUID SHA1 init.

Christopher Egert (1):

- radeon: fix ColorMask

Dave Airlie (1):

- virgl: don't send a shader create with no data. (v2)

Dylan Baker (10):

- docs/relnotes: Add sha256 sums for mesa 18.1.8
- cherry-ignore: Add additional 18.2 patch
- meson: Print a message about why a libdrm version was selected
- cherry-ignore: add another 18.2 patch
- cherry-ignore: Add patches that don't apply cleanly and are for developer tools
- cherry-ignore: Add more 18.2 patches
- cherry-ignore: add 18.2 patchs
- cherry-ignore: add a patch that was reverted on master
- cherry-ignore: one final update
- Bump version to 18.1.9

Erik Faye-Lund (2):

- winsys/virgl: avoid unintended behavior
- virgl: adjust strides when mapping temp-resources

Gert Wollny (1):

- winsys/virgl: correct resource and handle allocation (v2)

Jason Ekstrand (6):

- anv/pipeline: Only consider double elements which actually exist
- i965: Workaround the gen9 hw astc5x5 sampler bug
- anv: Re-emit vertex buffers when the pipeline changes
- anv: Disable the vertex cache when tessellating on SKL GT4
- anv: Clamp scissors to the framebuffer boundary
- anv/query: Write both dwords in emit_zero_queries

Josh Pieper (1):

- st/mesa: Validate the result of pipe_transfer_map in make_texture (v2)

Kenneth Feng (1):

- amd: Add Picasso device id

Marek Olšák (4):

- st/mesa: help fix stencil border color for GL_DEPTH_STENCIL textures
- radeonsi: fix HTILE for NPOT textures with mipmapping on SI/CI
- r600: fix HTILE for NPOT textures with mipmapping
- radeonsi: fix printing a BO list into ddebug reports

Mathias Fröhlich (1):

- tnl: Fix green gun regression in xonotic.

Mauro Rossi (3):

- android: broadcom/genxml: fix collision with intel/genxml header-gen macro
- android: broadcom/cle: add gallium include path
- android: broadcom/cle: export the broadcom top level path headers

Michal Srb (1):

- st/dri: don't set queryDmaBufFormats/queryDmaBufModifiers if the driver does not implement it

Michel Dänzer (1):

- loader/dri3: Only wait for back buffer fences in dri3_get_buffer

Pierre Moreau (1):

- nvir: Always split 64-bit IMAD/IMUL operations

Sergii Romantsov (1):

- intel: compiler option msse2 and mstackrealign

Timothy Arceri (1):

- glsl: fixer lexer for unreachable defines

4.71 Mesa 18.2.1 Release Notes / September 21, 2018

Mesa 18.2.1 is a bug fix release which fixes bugs found since the 18.2.0 release.

Mesa 18.2.0 implements the OpenGL 4.5 API, but the version reported by glGetString(GL_VERSION) or glGetIntegerv(GL_MAJOR_VERSION) / glGetIntegerv(GL_MINOR_VERSION) depends on the particular driver being used. Some drivers don't support all the features required in OpenGL 4.5. OpenGL 4.5 is **only** available if requested at context creation. Compatibility contexts may report a lower version depending on each driver.

4.71.1 SHA256 checksums

```
SHA256: 45419ccbe1bf9a2e15ffe71ced34615002e1b42c24b917fbe2b2f58ab1970562  mesa-18.2.1.
↪tar.gz
SHA256: 9636dc6f3d188abdcca02da97cedd73640d9035224efd5db724187d062c81056  mesa-18.2.1.
↪tar.xz
```

4.71.2 New features

None

4.71.3 Bug fixes

- Bug 103241 - Anv crashes when using 64-bit vertex inputs
- Bug 107280 - [DXVK] Batman: Arkham City with tessellation enabled hangs on SKL GT4
- Bug 107772 - Mesa preprocessor matches if(def)s & endifs incorrectly
- Bug 107779 - Access violation with some games

- Bug 107810 - The 'va_end' call is missed after 'va_copy' in 'util_vsnprintf' function under windows
- Bug 107832 - Gallium picking A16L16 formats when emulating INTENSITY16 conflicts with mesa
- Bug 107843 - 32bit Mesa build failes with meson.
- Bug 107879 - crash happens when link program
- Bug 107891 - [wine, regression, bisected] RAGE, Wolfenstein The New Order hangs in menu

4.71.4 Changes

Andres Gomez (3):

- docs: add sha256 checksums for 18.2.0
- Revert "Revert "glsl: skip stringification in preprocessor if in unreachable branch""
- cherry-ignore: i965/tools: 32bit compilation with meson

Andrii Simiklit (4):

- apple/glx/log: added missing va_end() after va_copy()
- mesa/util: don't use the same 'va_list' instance twice
- mesa/util: don't ignore NULL returned from 'malloc'
- mesa/util: add missing va_end() after va_copy()

Bas Nieuwenhuizen (5):

- radv: Support v3 of VK_EXT_vertex_attribute_divisor.
- radv: Set the user SGPR MSB for Vega.
- radv: Only allow 16 user SGPRs for compute on GFX9+.
- radv: Use build ID if available for cache UUID.
- radv: Fix driver UUID SHA1 init.

Christopher Egert (1):

- radeon: fix ColorMask

Dave Airlie (1):

- virgl: don't send a shader create with no data. (v2)

Dylan Baker (1):

- meson: Print a message about why a libdrm version was selected

Eric Anholt (2):

- v3d: Fix SRC_ALPHA_SATURATE blending for RTs without alpha.
- v3d: Fix setup of the VCM cache size.

Erik Faye-Lund (2):

- winsys/virgl: avoid unintended behavior
- virgl: adjust strides when mapping temp-resources

Fritz Koenig (2):

- mesa: Additional FlipY applications

- mesa: FramebufferParameteri parameter checking

Gert Wollny (2):

- winsys/virgl: correct resource and handle allocation (v2)
- mesa/texture: Also check for LA texture when querying intensity component size

Ian Romanick (1):

- i965/fs: Don't propagate conditional modifiers from integer compares to adds

Jason Ekstrand (11):

- anv/pipeline: Only consider double elements which actually exist
- i965: Workaround the gen9 hw astc5x5 sampler bug
- anv: Re-emit vertex buffers when the pipeline changes
- anv: Disable the vertex cache when tessellating on SKL GT4
- anv: Clamp scissors to the framebuffer boundary
- vulkan: Update the XML and headers to 1.1.84
- anv: Support v3 of VK_EXT_vertex_attribute_divisor
- anv/query: Write both dwords in emit_zero_queries
- nir: Add a small pass to rematerialize derefs per-block
- nir/loop_unroll: Re-materialize derefs in use blocks before unrolling
- nir/opt_if: Re-materialize derefs in use blocks before peeling loops

Josh Pieper (1):

- st/mesa: Validate the result of pipe_transfer_map in make_texture (v2)

Juan A. Suarez Romero (2):

- cherry-ignore: radv: fix descriptor pool allocation size
- Update version to 18.2.1

Kenneth Feng (1):

- amd: Add Picasso device id

Marek Olšák (5):

- radeonsi: fix HTILE for NPOT textures with mipmapping on SI/CI
- winsys/radeon: fix CMASK fast clear for NPOT textures with mipmapping on SI/CI
- r600: fix HTILE for NPOT textures with mipmapping
- radeonsi: fix printing a BO list into ddebug reports
- ac: revert new LLVM 7.0 behavior for fdiv

Mathias Fröhlich (1):

- tnl: Fix green gun regression in xonotic.

Mauro Rossi (3):

- android: broadcom/genxml: fix collision with intel/genxml header-gen macro
- android: broadcom/cle: add gallium include path

- android: broadcom/cle: export the broadcom top level path headers

Michel Dänzer (1):

- loader/dri3: Only wait for back buffer fences in dri3_get_buffer

Pierre Moreau (1):

- nvir: Always split 64-bit IMAD/IMUL operations

Samuel Pitoiset (7):

- radv: fix function names for VK_EXT_conditional_rendering
- radv: fix VK_EXT_conditional_rendering visibility
- radv: bump the maximum number of arguments to 64
- radv: handle loc->indirect correctly for the first descriptor
- radv: fix GPU hangs with 32-bit indirect descriptors
- radv: fix flushing indirect descriptors
- radv: fix setting global locations for indirect descriptors

Sergii Romantsov (3):

- intel: compiler option msse2 and mstackrealign
- i965/tools: 32bit compilation with meson
- mesa/meson: 32bit xmlconfig linkage

Timothy Arceri (2):

- glsl: fixer lexer for unreachable defines
- Revert "radeonsi: avoid syncing the driver thread in si_fence_finish"

4.72 Mesa 18.2.0 Release Notes / September 7, 2018

Mesa 18.2.0 is a new development release. People who are concerned with stability and reliability should stick with a previous release or wait for Mesa 18.2.1.

Mesa 18.2.0 implements the OpenGL 4.5 API, but the version reported by glGetString(GL_VERSION) or glGetIntegerv(GL_MAJOR_VERSION) / glGetIntegerv(GL_MINOR_VERSION) depends on the particular driver being used. Some drivers don't support all the features required in OpenGL 4.5. OpenGL 4.5 is **only** available if requested at context creation. Compatibility contexts may report a lower version depending on each driver.

libwayland-egl is now distributed by Wayland (since 1.15, see announcement), and has been removed from Mesa in this release. Make sure you're using an up-to-date version of Wayland to keep the functionality.

4.72.1 SHA256 checksums

```
b9e6bb3eb7660b0726ba28405ffa0cb77de619e925b910b72f4d7a85c0098596  mesa-18.2.0.tar.gz
22452bdffff8e11bf4284278155a9f77cb28d6d73a12c507f1490732d0d9ddce  mesa-18.2.0.tar.xz
```

4.72.2 New features

Note: some of the new features are only available with certain drivers.

- OpenGL 4.3 on virgl
- OpenGL 4.4 Compatibility profile on radeonsi
- OpenGL ES 3.2 on radeonsi and virgl
- GL_ARB_ES3_2_compatibility on radeonsi
- GL_ARB_fragment_shader_interlock on i965
- GL_ARB_sample_locations and GL_NV_sample_locations on nvc0 (GM200+)
- GL_ANDROID_extension_pack_es31a on radeonsi.
- GL_KHR_texture_compression_astc_ldr on radeonsi
- GL_NV_conservative_raster and GL_NV_conservative_raster_dilate on nvc0 (GM200+)
- GL_NV_conservative_raster_pre_snap_triangles on nvc0 (GP102+)
- multisampled images on nvc0 (GM107+) (now supported on GF100+)

4.72.3 Bug fixes

- Bug 13728 - [G965] Some objects in Neverwinter Nights Linux version not displayed correctly
- Bug 61761 - glPolygonOffsetEXT, OFFSET_BIAS incorrectly set to a huge number
- Bug 65422 - Rename api_validate.[ch] to draw_validate.[ch]
- Bug 78097 - glUniform1ui and friends not supported by display lists
- Bug 91808 - trine1 misrender r600g
- Bug 93355 - [BXT,SKLGT4e] intermittent ext_framebuffer_multisample.accuracy fails
- Bug 95009 - [SNB] amd_shader_trinary_minmax.execution.built-in-functions.gs-mid3-ivec2-ivec2-ivec2 intermittent
- Bug 95012 - [SNB] glsl-1_50.execution.built-in-functions.gs-op tests intermittent
- Bug 98699 - “float[a+++4 ? 1:1] f;” crashes glsl_compiler
- Bug 99116 - Wine DirectDraw programs showing only a blackscreen when using Mesa Gallium drivers
- Bug 99730 - Metro Redux game(s) needs override for midshader extension declaration
- Bug 100177 - [GM206] Misrendering in XCOM Ennemy Within
- Bug 100430 - [radv] graphical glitches on dolphin emulator
- Bug 101247 - Mesa fails to link GLSL programs with unused output blocks
- Bug 102390 - centroid interpolation causes broken attribute values
- Bug 102678 - gl_BaseVertex should always be zero when the draw command has no <basevertex> parameter
- Bug 103274 - BRW allocates too much heap memory
- Bug 104388 - [snb] GPU HANG: ecode 6:0:0x85fffff8 in fgfs
- Bug 104626 - broadcom/vc5: double compare
- Bug 104809 - anv: DOOM 2016 and Wolfenstein II:The New Colossus crash due to not having depthBoundsTest

- Bug 105351 - [Gen6+] piglit's arb_shader_image_load_store-host-mem-barrier fails with a glGetTexSubImage fallback path
- Bug 105374 - texture3d, a SaschaWillems demo, assert fails
- Bug 105396 - tc compatible htile sets depth of htiles of discarded fragments to 1.0
- Bug 105399 - [snb] GPU hang: after geometry shader emits no geometry, the program hangs
- Bug 105497 - shader-db crashes on 72 core system after ast_type_qualifier bitset change
- Bug 105613 - Compute shader locks up within nested "for" loop
- Bug 105731 - linker error "fragment shader input … has no matching output in the previous stage" when previous stage's output declaration in a separate shader object
- Bug 105904 - Needed to delete mesa shader cache after driver upgrade for 32 bit wine vulkan programs to work.
- Bug 105975 - i965 always reports 0 viewport subpixel bits
- Bug 106090 - Compiling compute shader crashes RADV
- Bug 106133 - make check "OSError: [Errno 24] Too many open files"
- Bug 106163 - r600/sb: optimizer tries to schedule access to different array elements in one instruction group
- Bug 106174 - vulkan dota2 broken (segfaulting), found bug commit
- Bug 106180 - [bisected] radv vulkan smoke test black screen (Add support for DRI3 v1.2)
- Bug 106232 - LLVM unit tests have error in random number handling
- Bug 106243 - [kbl] GPU HANG: 9:0:0x85dffffb, in Cinnamon
- Bug 106315 - The witness + dxvk suffers flickering garbage
- Bug 106331 - radv doesnt support VK_FORMAT_R32G32B32_SFLOAT
- Bug 106382 - Shader cache breaks INTEL_DEBUG=shader_time
- Bug 106393 - glsl-fs-shader-stencil-export hangs forever
- Bug 106450 - glGetIntegerv return wrong value in some cases
- Bug 106462 - piglit.spec.arb_vertex_array_bgra.get regression
- Bug 106479 - NDEBUG not defined for libamdgpu_addrlib
- Bug 106480 - A2B10G10R10_SNORM vertex attribute doesn't work.
- Bug 106499 - [regression, bisected] Several games crash on start
- Bug 106504 - vulkan SPIR-V parsing failed at ../src/compiler/spirv/vtn_cfg.c:381
- Bug 106511 - radv: MSAA broken on SI (assertion failure in vkCreateImage)
- Bug 106587 - Dota2 is very dark when using vulkan render on a Intel << AMD prime setup
- Bug 106594 - [regression,apitrace,bisected] Prison Architect rendered unplayable by multicoloured flickering triangles and overlayed triangles when performing certain actions
- Bug 106619 - [OpenCL][llvm-svn]build failure addPassesToEmitFile candidate expects 6 arguments, 3 provided
- Bug 106629 - [SNB,IVB,HSW,BDW] dEQP-EGL.functional.image.create.gles2_cubemap_negative_z_rgb_read_pixels
- Bug 106642 - X server crashes in i965 on desktop startup when DRI3 v1.2 / modifier support is enabled
- Bug 106643 - double free when exporting a temporarily imported semaphore
- Bug 106673 - [bisected] Steam is unusable since commit 5c33e8c7

- Bug 106687 - radv: Fast color clears use incorrect format
- Bug 106708 - [SKL/KBL/GLK] 2-3% performance drop in SynMark DrvState and 5-9% drop on SynMark Multithread
- Bug 106748 - st/mesa: use PIPE_CAP_GLSL_FEATURE_LEVEL_COMPATIBILITY broke qemu -display sdl,gl=on
- Bug 106756 - Wine 3.9 crashes with DXVK on Just Cause 3 and Quantum Break on VEGA but works ON POLARIS
- Bug 106774 - GLSL IR copy propagates loads of SSBOs
- Bug 106776 - vma_random unrecognized command line option "-std=c++11"
- Bug 106778 - Files missing from tarball - intel_sanitize_gpu.*
- Bug 106779 - Files missing from tarball - u_debug_stack_android.cpp
- Bug 106784 - 18.1.1 autotools build fail without mako
- Bug 106801 - vma_random_test.cpp:239:18: error: non-constant-expression cannot be narrowed from type 'unsigned long' to 'uint_fast32_t' (aka 'unsigned int') in initializer list [-Wc++11-narrowing]
- Bug 106810 - ProgramBinary does not switch program correctly when using transform feedback
- Bug 106823 - Failed to recongnize keyword of shader code
- Bug 106830 - [bisected] 32 bit tests (deqp, piglit, glcts, vulkancts) crashing on all platforms
- Bug 106861 - fatal error: wayland-egl-backend.h: No such file or directory compilation terminated.
- Bug 106865 - [GLK] piglit.spec.ext_framebuffer_multisample.accuracy stencil tests fail
- Bug 106903 - radv: Fragment shader output goes to wrong attachments when render targets are sparse
- Bug 106906 - Failed to recongnize keyword "sampler2DRect" and "sampler2DRectShadow"
- Bug 106907 - Correct Transform Feedback Varyings information is expected after using ProgramBinary
- Bug 106912 - radv: 16-bit depth buffer causes artifacts in Shadow Warrior 2
- Bug 106928 - When starting a match Rocket League crashes on "Go"
- Bug 106941 - Intel ANV vulkan driver exposing version 1.1.0 which is incorrect
- Bug 106986 - glGetQueryiv error when querying number of result bits for GL_ANY_SAMPLES_PASSED_CONSERVATIVE
- Bug 106997 - [Regression]. Dying light game is crashing on latest mesa
- Bug 107098 - Segfault after munmap(kms_sw_dt->ro_mapped)
- Bug 107117 - mesa-18.1: regression with TFP on intel with modesettings and glamor acceleration
- Bug 107190 - Got seg fault on snb when use INTEL_DEBUG=bat
- Bug 107193 - piglit.spec.arb_compute_shader.linker.bug-93840 fails
- Bug 107212 - Dual-Core CPU E5500 / G45: RetroArch with reicast core results in corrupted graphics
- Bug 107223 - [GEN9+] 50% perf drop in SynMark Fill* tests (E2E RBC gets disabled?)
- Bug 107248 - [G45 ILK G965] Texture handling broken
- Bug 107275 - NIR segfaults after spirv-opt
- Bug 107276 - radv: OpBitfieldUExtract returns incorrect result when count is zero
- Bug 107295 - Access violation on glDrawArrays with count >= 2048

- Bug 107305 - glsl/opt_copy_propagation_elements.cpp:72:9: error: delegating constructors are permitted only in C++11
- Bug 107312 - Mesa-git RPM build fails after commit 8cacf38f527d42e41441ef8c25d95d4b2f4e8602
- Bug 107359 - [Regression] [bisected] [OpenGL CTS] [SKL,BDW] KHR-GL46.texture_barrier*-texels, GTF-GL46.gtf21.GL2FixedTests.buffer_corners.buffer_corners, and GTF-GL46.gtf21.GL2FixedTests.stencil_plane_corners.stencil_plane_corners fail with some configuration
- Bug 107366 - NIR verification crashes on piglit tests
- Bug 107423 - vc4 build failure: "v3d_decoder.c:893: undefined reference to 'clif_lookup_bo'"
- Bug 107443 - Build error on arm64: v3d_decoder.c:837:17: error: format not a string literal and no format arguments [-Werror=format-security]
- Bug 107460 - radv: OpControlBarrier does not always work correctly (bisected)
- Bug 107477 - [DXVK] Setting high shader quality in GTA V results in LLVM error
- Bug 107510 - [GEN8+] up to 10% perf drop on several 3D benchmarks
- Bug 107544 - intel/decoder: out of bounds group_iter
- Bug 107550 - "0[2]" as function parameter hits assert
- Bug 107579 - [SNB] The graphic corruption when we reuse the GS compiled and used for TFB when statebuffer contain magic trash in the unused space
- Bug 107601 - Rise of the Tomb Raider Segmentation Fault when the game starts
- Bug 107610 - Dolphin emulator mis-renders shadow overlay in Super Mario Sunshine

4.72.4 Changes

- Removed GL_EXT_polygon_offset applications should use glPolygonOffset instead.
- Removed libwayland-egl, now part of Wayland

4.73 Mesa 18.1.8 Release Notes / September 7 2018

Mesa 18.1.8 is a bug fix release which fixes bugs found since the 18.1.7 release.

Mesa 18.1.8 implements the OpenGL 4.5 API, but the version reported by glGetString(GL_VERSION) or glGetIntegerv(GL_MAJOR_VERSION) / glGetIntegerv(GL_MINOR_VERSION) depends on the particular driver being used. Some drivers don't support all the features required in OpenGL 4.5. OpenGL 4.5 is **only** available if requested at context creation. Compatibility contexts may report a lower version depending on each driver.

4.73.1 SHA256 checksums

```
8ec62f215dd1bb3910987f9941c6fc31632a0874e618815cf1e8e29445c86e0a  mesa-18.1.8.tar.gz
bd1be67fe9c73b517765264ac28911c84144682d28dbff140e1c2deb2f44c21b  mesa-18.1.8.tar.xz
```

4.73.2 New features

None

4.73.3 Bug fixes

- Bug 93355 - [BXT,SKLGT4e] intermittent ext_framebuffer_multisample.accuracy fails
- Bug 101247 - Mesa fails to link GLSL programs with unused output blocks
- Bug 104809 - anv: DOOM 2016 and Wolfenstein II:The New Colossus crash due to not having depthBoundsTest
- Bug 105904 - Needed to delete mesa shader cache after driver upgrade for 32 bit wine vulkan programs to work.
- Bug 106738 - No test for miptrees with DRI modifiers
- Bug 106865 - [GLK] piglit.spec.ext_framebuffer_multisample.accuracy stencil tests fail
- Bug 107359 - [Regression] [bisected] [OpenGL CTS] [SKL,BDW] KHR-GL46.texture_barrier*-texels, GTF-GL46.gtf21.GL2FixedTests.buffer_corners.buffer_corners, and GTF-GL46.gtf21.GL2FixedTests.stencil_plane_corners.stencil_plane_corners fail with some configuration
- Bug 107477 - [DXVK] Setting high shader quality in GTA V results in LLVM error
- Bug 107579 - [SNB] The graphic corruption when we reuse the GS compiled and used for TFB when statebuffer contain magic trash in the unused space
- Bug 107601 - Rise of the Tomb Raider Segmentation Fault when the game starts
- Bug 107760 - GPU Hang when Playing DiRT 3 Complete Edition using Steam Play with DXVK

4.73.4 Changes

Andrii Simiklit (1):

- i965/gen6/xfb: handle case where transform feedback is not active

Bas Nieuwenhuizen (3):

- radv: Add missing checks in radv_get_image_format_properties.
- radv: Fix CMASK dimensions.
- radv: Use a lower max offchip buffer count.

Christian Gmeiner (1):

- tegra: fix memory leak

Daniel Stone (1):

- st/dri: Don't expose sRGB formats to clients

Dave Airlie (1):

- ac/radeonsi: fix CIK copy max size

Dylan Baker (10):

- docs: Add mesa 18.1.7 notes
- cherry-ignore: add a patch
- cherry-ignore: Add more 18.2 only patches
- meson: Actually load translation files
- cherry-ignore: Add more 18.2 patches
- cherry-ignore: Add additional patch

- cherry-ignore: Add patch that doesn't apply to 18.1
- cherry-ignore: Add a couple of two fixes warning patches
- cherry-ignore: Add patch that needs more significant patches to function
- Bump version to 18.1.8

Emil Velikov (1):

- docs: update required mako version

Grazvydas Ignotas (1):

- radv: place pointer length into cache uuid

Gurchetan Singh (2):

- meson: fix egl build for surfaceless
- meson: fix egl build for android

Ian Romanick (2):

- i965/vec4: Clamp indirect tes input array reads with 0x0fffffff
- i965/vec4: Correctly handle uniform sources in generate_tes_add_indirect_urb_offset

Jason Ekstrand (5):

- anv: Fill holes in the VF VUE to zero
- nir/algebraic: Be more careful converting ushr to extract_u8/16
- egl/dri2: Add a helper for the number of planes for a FOURCC format
- egl/dri2: Guard against invalid fourcc formats
- anv/blorp: Do more flushing around HiZ clears

Juan A. Suarez Romero (1):

- egl/wayland: do not leak wl_buffer when it is locked

Lionel Landwerlin (1):

- anv: blorp: support multiple aspect blits

Marek Olšák (1):

- glapi: actually implement GL_EXT_robustness for GLES

Nanley Chery (7):

- intel/isl: Avoid tiling some 16K-wide render targets
- i965: Make blt_pitch public
- i965/miptree: Drop an if case from retile_as_linear
- i965/miptree: Use the correct BLT pitch
- i965/miptree: Use miptree_map in map_blit functions
- i965/miptree: Fix can_blit_slice()
- i965/gen7_urb: Re-emit PUSH_CONSTANT_ALLOC on some gen9

Samuel Pitoiset (1):

- radv: fix passing clip/cull distances from VS to PS

vadym.shovkoplias (1):

- glsl/linker: Allow unused in blocks which are not declated on previous stage

4.74 Mesa 18.1.7 Release Notes / August 24 2018

Mesa 18.1.7 is a bug fix release which fixes bugs found since the 18.1.6 release.

Mesa 18.1.7 implements the OpenGL 4.5 API, but the version reported by glGetString(GL_VERSION) or glGetIntegerv(GL_MAJOR_VERSION) / glGetIntegerv(GL_MINOR_VERSION) depends on the particular driver being used. Some drivers don't support all the features required in OpenGL 4.5. OpenGL 4.5 is **only** available if requested at context creation. Compatibility contexts may report a lower version depending on each driver.

4.74.1 SHA256 checksums

```
0c3c240bcd1352d179e65993214f9d55a399beac852c3ab4433e8df9b6c51c83  mesa-18.1.7.tar.gz
655e3b32ce3bdddd5e6e8768596e5d4bdef82d0dd37067c324cc4b2daa207306  mesa-18.1.7.tar.xz
```

4.74.2 New features

None

4.74.3 Bug fixes

- Bug 105975 - i965 always reports 0 viewport subpixel bits
- Bug 107098 - Segfault after munmap(kms_sw_dt->ro_mapped)

4.74.4 Changes

Alexander Tsoy (1):

- meson: fix build for egl platform_x11 without dri3 and gbm

Bas Nieuwenhuizen (1):

- radv: Fix missing Android platform define.

Danylo Piliaiev (1):

- i965: Advertise 8 bits subpixel precision for viewport bounds on gen6+

Dave Airlie (1):

- r600/eg: rework atomic counter emission with flushes

Dylan Baker (7):

- docs: Add sha256 sums for 18.1.6
- cherry-ignore: Add additional 18.2 only patches
- cherry-ignore: Add more 18.2 patches
- cherry-ignore: Add more 18.2 patches

- cherry-ignore: Add a couple of patches with > 1 fixes tags
- cherry-ignore: more 18.2 patches
- bump version for 18.1.7 release

Jason Ekstrand (2):

- intel: Switch the order of the 2x MSAA sample positions
- anv/lower_ycbcr: Use the binding array size for bounds checks

Ray Strode (1):

- gallium/winsys/kms: don't unmap what wasn't mapped

Samuel Pitoiset (1):

- radv/winsys: fix creating the BO list for virtual buffers

Timothy Arceri (1):

- radv: add Doom workaround

4.75 Mesa 18.1.6 Release Notes / August 13 2018

Mesa 18.1.6 is a bug fix release which fixes bugs found since the 18.1.5 release.

Mesa 18.1.6 implements the OpenGL 4.5 API, but the version reported by glGetString(GL_VERSION) or glGetIntegerv(GL_MAJOR_VERSION) / glGetIntegerv(GL_MINOR_VERSION) depends on the particular driver being used. Some drivers don't support all the features required in OpenGL 4.5. OpenGL 4.5 is **only** available if requested at context creation. Compatibility contexts may report a lower version depending on each driver.

4.75.1 SHA256 checksums

```
580e03328ffefe1fd43b19ab7669f20d931601a1c0a4c0f8b9c65d6e81a06df3  mesa-18.1.6.tar.gz
bb7ce759069801804fcfb8152da3457f76cd7b4e0096e4870ff5adcb5c894289  mesa-18.1.6.tar.xz
```

4.75.2 New features

None

4.75.3 Bug fixes

- Bug 13728 - [G965] Some objects in Neverwinter Nights Linux version not displayed correctly
- Bug 98699 - "float[a+++4 ? 1:1] f;" crashes glsl_compiler
- Bug 99730 - Metro Redux game(s) needs override for midshader extension declaration
- Bug 106382 - Shader cache breaks INTEL_DEBUG=shader_time
- Bug 107117 - mesa-18.1: regression with TFP on intel with modesettings and glamor acceleration
- Bug 107212 - Dual-Core CPU E5500 / G45: RetroArch with reicast core results in corrupted graphics

4.75.4 Changes

Adam Jackson (1):

- glx: GLX_MESA_multithread_makecurrent is direct-only

Andres Gomez (3):

- ddebug: use util_snprintf() in dd_get_debug_filename_and_mkdir
- gallium/aux/util: use util_snprintf() in test_texture_barrier
- glsl: use util_snprintf()

Christian Gmeiner (1):

- etnaviv: fix typo in query names

Dave Airlie (1):

- r600: reduce num compute threads to 1024.

Dylan Baker (6):

- docs: Add sha-256 sums for 18.1.5
- nir/meson: fix c vs cpp args for nir test
- gallium: fix ddebug on windows
- cherry-ignore: add patches that get-pick-list is finding in error
- cherry-ignore: Add some additional patches that are for 18.2
- bump version to 18.1.6

Emil Velikov (5):

- swr: don't export swr_create_screen_internal
- automake: require shared glapi when using DRI based libGL
- autotools: error out when using the broken –with-{gl, osmesa}-lib-name
- autotools: error out when building with mangling and glvnd
- autotools: use correct gl.pc LIBS when using glvnd

Eric Anholt (4):

- vc4: Fix a leak of the no-vertex-elements workaround BO.
- vc4: Respect a sampler view's first_layer field.
- vc4: Ignore samplers for finding uniform offsets.
- egl: Fix leak of X11 pixmaps backing pbuffers in DRI3.

Gert Wollny (1):

- meson, install_megadrivers: Also remove stale symlinks

Jan Vesely (2):

- clover: Reduce wait_count in abort path.
- clover: Don't extend illegal integer types.

Jason Ekstrand (2):

- nir: Take if uses into account in ssa_def_components_read

- i965/fs: Flag all slots of a flat input as flat

Jon Turney (1):

- meson: use correct keyword to fix a meson warning

Jordan Justen (2):

- i965, anv: Use INTEL_DEBUG for disk_cache driver flags
- i965: Disable shader cache with INTEL_DEBUG=shader_time

Juan A. Suarez Romero (2):

- wayland/egl: update surface size on window resize
- wayland/egl: initialize window surface size to window size

Karol Herbst (2):

- nir/lower_int64: mark all metadata as dirty
- nvc0/ir: return 0 in imageLoad on incomplete textures

Kenneth Graunke (1):

- intel: Fix SIMD16 unaligned payload GRF reads on Gen4-5.

Marek Olšák (1):

- ac/surface: fix MSAA corruption on Vega due to FMASK tile swizzle

Mauro Rossi (2):

- radv: generate entrypoints for VK_ANDROID_native_buffer
- radv: move vk_format_table.c to generated sources

Olivier Fourdan (1):

- dri3: For 1.2, use root window instead of pixmap drawable

Tapani Pälli (1):

- glsl: handle error case with ast_post_inc, ast_post_dec

Vlad Golovkin (1):

- swr: Remove unnecessary memset call

vadym.shovkoplias (1):

- drirc: Allow extension midshader for Metro Redux

4.76 Mesa 18.1.4 Release Notes / July 13 2018

Mesa 18.1.5 is a bug fix release which fixes bugs found since the 18.1.4 release.

Mesa 18.1.5 implements the OpenGL 4.5 API, but the version reported by glGetString(GL_VERSION) or glGetIntegerv(GL_MAJOR_VERSION) / glGetIntegerv(GL_MINOR_VERSION) depends on the particular driver being used. Some drivers don't support all the features required in OpenGL 4.5. OpenGL 4.5 is **only** available if requested at context creation. Compatibility contexts may report a lower version depending on each driver.

4.76.1 SHA256 checksums

```
SHA256: f966d5d5d373a5b8a16ed5036c1e7f05d4ad46d130f793bf9782c3ac9133a02e  mesa-18.1.5.
↪tar.gz
SHA256: 69dbe6f1a6660386f5beb85d4fcf003ee23023ed7b9a603de84e9a37e8d98dea  mesa-18.1.5.
↪tar.xz
```

4.76.2 New features

None

4.76.3 Bug fixes

- Bug 103274 - BRW allocates too much heap memory
- Bug 107275 - NIR segfaults after spirv-opt
- Bug 107295 - Access violation on glDrawArrays with count >= 2048
- Bug 107312 - Mesa-git RPM build fails after commit 8cacf38f527d42e41441ef8c25d95d4b2f4e8602
- Bug 107366 - NIR verification crashes on piglit tests

4.76.4 Changes

Alex Smith (1):

- anv: Pay attention to VK_ACCESS_MEMORY_(READ|WRITE)_BIT

Bas Nieuwenhuizen (7):

- radv: Select correct entries for binning.
- radv: Fix number of samples used for binning.
- radv: Disable disabled color buffers in rbplus opts.
- nir: Do not use continue block after removing it.
- util/disk_cache: Fix disk_cache_get_function_timestamp with disabled cache.
- nir: Fix end of function without return warning/error.
- radv: Still enable inmemory & API level caching if disk cache is not enabled.

Chad Versace (2):

- anv/android: Fix type error in call to vk_errorf()
- anv/android: Fix Autotools build for VK_ANDROID_native_buffer

Chih-Wei Huang (1):

- Android: fix a missing nir_intrinsics.h error

Danylo Piliaiev (1):

- i965: Sweep NIR after linking phase to free held memory

Dave Airlie (1):

- r600: enable tess_input_info for TES

Dylan Baker (5):

- docs: Add sha256 sums for 18.1.4 tarballs
- cherry-ignore: add 4a67ce886a7b3def5f66c1aedf9e5436d157a03c
- cherry-ignore: Add 1f616a840eac02241c585d28e9dac8f19a297f39
- cherry-ignore: add 11712b9ca17e4e1a819dcb7d020e19c6da77bc90
- bump version to 18.1.5

Eric Anholt (2):

- vc4: Don't automatically reallocate a PERSISTENT-mapped buffer.
- meson: Move xvmc test tools from unit tests to installed tools.

Harish Krupo (1):

- egl: Fix missing clamping in eglSetDamageRegionKHR

Jan Vesely (3):

- radeonsi: Refuse to accept code with unhandled relocations
- clover: Report error when pipe driver fails to create compute state
- clover: Catch errors from executing event action

Jason Ekstrand (6):

- anv: Stop setting 3DSTATE_PS_EXTRA::PixelShaderHasUAV
- nir/serialize: Alloc constants off the variable
- blorp: Handle the RGB workaround more like other workarounds
- intel/blorp: Handle 3-component formats in clears
- intel/compiler: Account for built-in uniforms in analyze_ubo_ranges
- spirv: Fix a couple of image atomic load/store bugs

José Fonseca (1):

- gallium/tests: Don't ignore S3TC errors.

Karol Herbst (1):

- nir: fix printing of vec16 type

Lepton Wu (1):

- virgl: Fix flush in virgl_encoder_inline_write.

Lucas Stach (1):

- st/mesa: call resource_changed when binding a EGLImage to a texture

Mauro Rossi (2):

- radv: winsys/amdgpu: include missing pthread.h header
- android: util/disk_cache: fix building errors in gallium drivers

Michel Dänzer (1):

- gallium: Check pipe_screen::resource_changed before dereferencing it

Roland Scheidegger (1):

- draw: force draw pipeline if there's more than 65535 vertices

Samuel Iglesias Gonsálvez (1):

- anv: fix assert in anv_CmdBindDescriptorSets()

Samuel Pitoiset (3):

- radv: make sure to wait for CP DMA when needed
- radv: emit a dummy ZPASS_DONE to prevent GPU hangs on GFX9
- radv: fix a memleak for merged shaders on GFX9

4.77 Mesa 18.1.4 Release Notes / July 13 2018

Mesa 18.1.4 is a bug fix release which fixes bugs found since the 18.1.3 release.

Mesa 18.1.4 implements the OpenGL 4.5 API, but the version reported by glGetString(GL_VERSION) or glGetIntegerv(GL_MAJOR_VERSION) / glGetIntegerv(GL_MINOR_VERSION) depends on the particular driver being used. Some drivers don't support all the features required in OpenGL 4.5. OpenGL 4.5 is **only** available if requested at context creation. Compatibility contexts may report a lower version depending on each driver.

4.77.1 SHA256 checksums

```
SHA256: 8acd42e4ac4d1e96ed22344073b3d4fef03d10f225f4eaf3f88c001dfc10e2db  mesa-18.1.4.
↪tar.gz
SHA256: 3061488b5d85504092cf4343816cfb2d96f2ad9bc2edec31fc96933d184cf58b  mesa-18.1.4.
↪tar.xz
```

4.77.2 New features

None

4.77.3 Bug fixes

- Bug 106906 - Failed to recongnize keyword "sampler2DRect" and "sampler2DRectShadow"
- Bug 106928 - When starting a match Rocket League crashes on "Go"
- Bug 107193 - piglit.spec.arb_compute_shader.linker.bug-93840 fails

4.77.4 Changes

Adam Jackson (1):

- glx: Don't allow glXMakeContextCurrent() with only one valid drawable

Dave Airlie (1):

- r600/sb: cleanup if_conversion iterator to be legal C++

Dylan Baker (2):

- docs: Add SHA256 sums to notes for 18.1.3

- Bump version for release

Iago Toral Quiroga (3):

- anv/cmd_buffer: make descriptors dirty when emitting base state address
- anv/cmd_buffer: clean dirty push constants flag after emitting push constants
- anv/cmd_buffer: never shrink the push constant buffer size

Ian Romanick (4):

- i965/vec4: Don't cmod propagate from CMP to ADD if the writemask isn't compatible
- intel/compiler: Relax mixed type restriction for saturating immediates
- i965/vec4: Properly handle sign(-abs(x))
- i965/fs: Properly handle sign(-abs(x))

Jason Ekstrand (3):

- intel/fs: Split instructions low to high in lower_simd_width
- anv: Be more careful about hashing pipeline layouts
- intel/fs: Mark LINTERP opcode as writing accumulator on platforms without PLN

Jose Maria Casanova Crespo (3):

- i965/fs: Register allocator shoudn't use grf127 for sends dest
- intel/compiler: grf127 can not be dest when src and dest overlap in send
- i965/fs: unspills shoudn't use grf127 as dest since Gen8+

Lionel Landwerlin (1):

- i965: fix clear color bo address relocation

Marek Olšák (3):

- radeonsi: fix memory exhaustion issue with DCC statistics gathering with DRI2
- glsl/cache: save and restore ExternalSamplersUsed
- st/dri: fix a crash in server_wait_sync

Neil Roberts (1):

- i965: Fix output register sizes when variable ranges are interleaved

Rhys Perry (1):

- nvc0/ir: fix TargetNVC0::insnCanLoadOffset()

Roland Scheidegger (1):

- r600/sb: fix crash in fold_alu_op3

Ross Burton (1):

- egl: fix build race in automake

Samuel Pitoiset (1):

- radv: fix emitting the view index on GFX9

Timothy Arceri (2):

- glsl: skip comparison opt when adding vars of different size

- nir: fix selection of loop terminator when two or more have the same limit

zhaowei yuan (1):

- glsl: Treat sampler2DRect and sampler2DRectShadow as reserved in ES2

4.78 Mesa 18.1.3 Release Notes / June 29 2018

Mesa 18.1.3 is a bug fix release which fixes bugs found since the 18.1.2 release.

Mesa 18.1.2 implements the OpenGL 4.5 API, but the version reported by glGetString(GL_VERSION) or glGetIntegerv(GL_MAJOR_VERSION) / glGetIntegerv(GL_MINOR_VERSION) depends on the particular driver being used. Some drivers don't support all the features required in OpenGL 4.5. OpenGL 4.5 is **only** available if requested at context creation. Compatibility contexts may report a lower version depending on each driver.

4.78.1 SHA256 checksums

```
2a1e36280d01ad18ba6d5b3fbd653ceaa109eaa031b78eb5dfaa4df452742b66  mesa-18.1.3.tar.gz
54f08deeda0cd2f818e8d40140040ed013de7852573002453b7f50da9ea738ce  mesa-18.1.3.tar.xz
```

4.78.2 New features

None

4.78.3 Bug fixes

- Bug 105396 - tc compatible htile sets depth of htiles of discarded fragments to 1.0
- Bug 105399 - [snb] GPU hang: after geometry shader emits no geometry, the program hangs
- Bug 106756 - Wine 3.9 crashes with DXVK on Just Cause 3 and Quantum Break on VEGA but works ON POLARIS
- Bug 106774 - GLSL IR copy propagates loads of SSBOs
- Bug 106903 - radv: Fragment shader output goes to wrong attachments when render targets are sparse
- Bug 106907 - Correct Transform Feedback Varyings information is expected after using ProgramBinary
- Bug 106912 - radv: 16-bit depth buffer causes artifacts in Shadow Warrior 2
- Bug 106980 - Basemark GPU vulkan benchmark fails.

4.78.4 Changes

Andrii Simiklit (1):

- i965/gen6/gs: Handle case where a GS doesn't allocate VUE

Bas Nieuwenhuizen (2):

- radv: Fix output for sparse MRTs.
- ac/surface: Set compressZ for stencil-only surfaces.

Christian Gmeiner (1):

- util/bitset: include util/macro.h

Dave Airlie (1):

- glsl: allow standalone semicolons outside main()

Dylan Baker (8):

- docs: Add release notes for 18.1.2
- cherry-ignore: Add 587e712eda95c31d88ea9d20e59ad0ae59afef4f
- meson: Fix auto option for va
- meson: Fix auto option for xvmc
- meson: Correct behavior of vdpau=auto
- cherry-ignore: Ignore cac7ab1192eefdd8d8b3f25053fb006b5c330eb8
- cherry-ignore: add a2f5292c82ad07731d633b36a663e46adc181db9
- VERSION: bump version to 18.1.3

Emil Velikov (2):

- configure: use compliant grep regex checks
- glsl/tests/glcpp: reinstate "error out if no tests found"

Eric Engestrom (3):

- radv: fix reported number of available VGPRs
- radv: fix bitwise check
- meson: fix i965/anv/isl genX static lib names

Ian Romanick (2):

- glsl: Don't copy propagate from SSBO or shared variables either
- glsl: Don't copy propagate elements from SSBO or shared variables either

Jason Ekstrand (2):

- nir: Handle call instructions in foreach_src
- nir/validate: Use the type from the tail of call parameter derefs

Lukas Rusak (2):

- meson: only build vl_winsys_dri.c when x11 platform is used
- meson: fix private libs when building without glx

Marek Olšák (5):

- radeonsi/gfx9: fix si_get_buffer_from_descriptors for 48-bit pointers
- ac/gpu_info: report real total memory sizes
- ac/gpu_info: add kernel_flushes_hdp_before_ib
- radeonsi: always put persistent buffers into GTT on radeon
- mesa: fix glGetInteger64v for arrays of integers

Rob Clark (1):

- freedreno/ir3: fix base_vertex

Samuel Pitoiset (6):

- radv: don't fast clear HTILE for 16-bit depth surfaces on GFX8
- radv: update the ZRANGE_PRECISION value for the TC-compat bug
- radv: fix emitting the TCS regs on GFX9
- radv: fix HTILE metadata initialization in presence of subpass clears
- radv: ignore pInheritanceInfo for primary command buffers
- radv: use separate bind points for the dynamic buffers

Tapani Pälli (1):

- glsl: serialize data from glTransformFeedbackVaryings

Tomeu Vizoso (1):

- virgl: Remove debugging left-overs

4.79 Mesa 18.1.2 Release Notes / June 15 2018

Mesa 18.1.2 is a bug fix release which fixes bugs found since the 18.1.1 release.

Mesa 18.1.2 implements the OpenGL 4.5 API, but the version reported by glGetString(GL_VERSION) or glGetIntegerv(GL_MAJOR_VERSION) / glGetIntegerv(GL_MINOR_VERSION) depends on the particular driver being used. Some drivers don't support all the features required in OpenGL 4.5. OpenGL 4.5 is **only** available if requested at context creation. Compatibility contexts may report a lower version depending on each driver.

4.79.1 SHA256 checksums

```
a644df23937f4078a2bd9a54349f6315c1955f5e3a4ac272832da51dea4d3c11  mesa-18.1.1.tar.gz
070bf0648ba5b242d7303ceed32aed80842f4c0ba16e5acc1a650a46eadfb1f9  mesa-18.1.1.tar.xz
```

4.79.2 New features

None

4.79.3 Bug fixes

None

4.79.4 Changes

Alex Smith (4):

- radv: Consolidate GFX9 merged shader lookup logic
- radv: Handle GFX9 merged shaders in radv_flush_constants()
- radeonsi: Fix crash on shaders using MSAA image load/store
- radv: Set active_stages the same whether or not shaders were cached

Andrew Galante (2):

- meson: Test for __atomic_add_fetch in atomic checks
- configure.ac: Test for __atomic_add_fetch in atomic checks

Bas Nieuwenhuizen (1):

- radv: Don't pass a TESS_EVAL shader when tesselation is not enabled.

Cameron Kumar (1):

- vulkan/wsi: Destroy swapchain images after terminating FIFO queues

Dylan Baker (6):

- docs/relnotes: Add sha256 sums for mesa 18.1.1
- cherry-ignore: add commits not to pull
- cherry-ignore: Add patches from Jason that he rebased on 18.1
- meson: work around gentoo applying -m32 to host compiler in cross builds
- cherry-ignore: Add another patch
- version: bump version for 18.1.2 release

Eric Engestrom (3):

- autotools: add missing android file to package
- configure: radv depends on mako
- i965: fix resource leak

Jason Ekstrand (10):

- intel/eu: Add some brw_get_default_ helpers
- intel/eu: Copy fields manually in brw_next_insn
- intel/eu: Set flag [sub]register number differently for 3src
- intel/blorp: Don't vertex fetch directly from clear values
- intel/isl: Add bounds-checking assertions in isl_format_get_layout
- intel/isl: Add bounds-checking assertions for the format_info table
- i965/screen: Refactor query_dma_buf_formats
- i965/screen: Use RGBA non-sRGB formats for images
- anv: Set fence/semaphore types to NONE in impl_cleanup
- i965/screen: Return false for unsupported formats in query_modifiers

Jordan Justen (1):

- mesa/program_binary: add implicit UseProgram after successful ProgramBinary

Juan A. Suarez Romero (1):

- glsl: Add ir_binop_vector_extract in NIR

Kenneth Graunke (2):

- i965: Fix batch-last mode to properly swap BOs.
- anv: Disable __gen_validate_value if NDEBUG is set.

Marek Olšák (1):

- r300g/swtcl: make pipe_context uploaders use malloc'd memory as before

Matt Turner (1):

- meson: Fix -latomic check

Michel Dänzer (1):

- glx: Fix number of property values to read in glXImportContextEXT

Nicolas Boichat (1):

- configure.ac/meson.build: Fix -latomic test

Philip Rebohle (1):

- radv: Use correct color format for fast clears

Samuel Pitoiset (3):

- radv: fix a GPU hang when MRTs are sparse
- radv: fix missing ZRANGE_PRECISION(1) for GFX9+
- radv: add a workaround for DXVK hangs by setting amdgpu-skip-threshold

Scott D Phillips (1):

- intel/tools: add intel_sanitize_gpu to EXTRA_DIST

Thomas Petazzoni (1):

- configure.ac: rework -latomic check

Timothy Arceri (2):

- ac: fix possible truncation of intrinsic name
- radeonsi: fix possible truncation on renderer string

4.80 Mesa 18.0.5 Release Notes / June 3, 2018

Mesa 18.0.5 is a bug fix release which fixes bugs found since the 18.0.4 release.

Mesa 18.0.5 implements the OpenGL 4.5 API, but the version reported by glGetString(GL_VERSION) or glGetIntegerv(GL_MAJOR_VERSION) / glGetIntegerv(GL_MINOR_VERSION) depends on the particular driver being used. Some drivers don't support all the features required in OpenGL 4.5. OpenGL 4.5 is **only** available if requested at context creation because compatibility contexts are not supported.

4.80.1 SHA256 checksums

```
ea3e00329cea899b1e32db812fd2f426832be37e4baa2e2fd9288a3480f30531  mesa-18.0.5.tar.gz
5187bba8d72aea78f2062d134ec6079a508e8216062dce9ec9048b5eb2c4fc6b  mesa-18.0.5.tar.xz
```

4.80.2 New features

None

4.80.3 Bug fixes

- Bug 78097 - glUniform1ui and friends not supported by display lists
- Bug 102390 - centroid interpolation causes broken attribute values
- Bug 105351 - [Gen6+] piglit's arb_shader_image_load_store-host-mem-barrier fails with a glGetTexSubImage fallback path
- Bug 106090 - Compiling compute shader crashes RADV
- Bug 106315 - The witness + dxvk suffers flickering garbage
- Bug 106465 - No test for Image Load/Store on format-incompatible texture buffer
- Bug 106479 - NDEBUG not defined for libamdgpu_addrlib
- Bug 106481 - No test for Image Load/Store on texture buffer sized greater than MAX_TEXTURE_BUFFER_SIZE_ARB
- Bug 106504 - vulkan SPIR-V parsing failed at ../src/compiler/spirv/vtn_cfg.c:381
- Bug 106587 - Dota2 is very dark when using vulkan render on a Intel << AMD prime setup
- Bug 106629 - [SNB,IVB,HSW,BDW] dEQP-EGL.functional.image.create.gles2_cubemap_negative_z_rgb_read_pixels

4.80.4 Changes

Anuj Phogat (1):

- i965/glk: Add l3 banks count for 2x6 configuration

Bas Nieuwenhuizen (2):

- amd/addrlib: Use defines in autotools build.
- radv: Fix SRGB compute copies.

Dave Airlie (1):

- tgsi/scan: add hw atomic to the list of memory accessing files

Francisco Jerez (4):

- Revert "mesa: simplify _mesa_is_image_unit_valid for buffers"
- i965: Move buffer texture size calculation into a common helper function.
- i965: Handle non-zero texture buffer offsets in buffer object range calculation.
- i965: Use intel_bufferobj_buffer() wrapper in image surface state setup.

Jan Vesely (1):

- eg/compute: Use reference counting to handle compute memory pool.

Jason Ekstrand (2):

- intel/eu: Set EXECUTE_1 when setting the rounding mode in cr0
- intel/blorp: Support blits and clears on surfaces with offsets

Jose Dapena Paz (1):

- mesa: do not leak ctx->Shader.ReferencedProgram references

Juan A. Suarez Romero (8):

- docs: add sha256 checksums for 18.0.4
- cherry-ignore: i965/miptree: Fix handling of uninitialized MCS buffers
- cherry-ignore: add explicit 18.1 only nominations
- cherry-ignore: mesa/st: handle vert_attrib_mask in nir case too
- cherry-ignore: Tegra is not supported
- cherry-ignore: st/mesa: fix assertion failures with GL_UNSIGNED_INT64_ARB (v2)
- cherry-ignore: nv30: ensure that displayable formats are marked accordingly
- Update version to 18.0.5

Marek Olšák (3):

- st/mesa: simplify lastLevel determination in st_finalize_texture
- radeonsi: fix incorrect parentheses around VS-PS varying elimination
- mesa: handle GL_UNSIGNED_INT64_ARB properly (v2)

Michel Dänzer (1):

- dri3: Stricter SBC wraparound handling

Nanley Chery (1):

- i965/miptree: Zero-initialize CCS_D buffers

Samuel Pitoiset (2):

- spirv: fix visiting inner loops with same break/continue block
- radv: fix centroid interpolation

Stuart Young (1):

- etnaviv: Fix missing rnndb file in tarballs

Timothy Arceri (1):

- mesa: add glUniform*ui{v} support to display lists

4.81 Mesa 18.1.1 Release Notes / June 1 2018

Mesa 18.1.1 is a bug fix release which fixes bugs found since the 18.1.0 release.

Mesa 18.1.1 implements the OpenGL 4.5 API, but the version reported by glGetString(GL_VERSION) or glGetIntegerv(GL_MAJOR_VERSION) / glGetIntegerv(GL_MINOR_VERSION) depends on the particular driver being used. Some drivers don't support all the features required in OpenGL 4.5. OpenGL 4.5 is **only** available if requested at context creation. Compatibility contexts may report a lower version depending on each driver.

4.81.1 SHA256 checksums

```
366a35f7530a016f2a8284fb0ee5759eeb216b4d6fa47f0e96b89ad2e43faf96  mesa-18.1.1.tar.gz
d3312a2ede5aac14a47476b208b8e3a401367838330197c4588ab8ad420d7781  mesa-18.1.1.tar.xz
```

4.81.2 New features

None

4.81.3 Bug fixes

None

4.81.4 Changes

Anuj Phogat (1):

- i965/glk: Add l3 banks count for 2x6 configuration

Bas Nieuwenhuizen (7):

- radv: Fix multiview queries.
- radv: Translate logic ops.
- radv: Fix up 2_10_10_10 alpha sign.
- radv: Disable texel buffers with A2 SNORM/SSCALED/SINT for pre-vega.
- amd/addrlib: Use defines in autotools build.
- radv: Fix SRGB compute copies.
- radv: Only expose subgroup shuffles on VI+.

Christoph Haag (1):

- radv: fix VK_EXT_descriptor_indexing

Dave Airlie (5):

- radv/resolve: do fmask decompress on all layers.
- radv: resolve all layers in compute resolve path.
- radv: use compute path for multi-layer images.
- virgl: set texture buffer offset alignment to disable ARB_texture_buffer_range.
- tgsi/scan: add hw atomic to the list of memory accessing files

Dylan Baker (2):

- docs: Add sha sums for release
- VERSION: bump to 18.1.1 for next release

Eric Engestrom (1):

- vulkan: don't free uninitialised memory

Francisco Jerez (4):

- Revert "mesa: simplify _mesa_is_image_unit_valid for buffers"
- i965: Move buffer texture size calculation into a common helper function.
- i965: Handle non-zero texture buffer offsets in buffer object range calculation.
- i965: Use intel_bufferobj_buffer() wrapper in image surface state setup.

Ilia Mirkin (1):

- nv30: ensure that displayable formats are marked accordingly

Jan Vesely (1):

- eg/compute: Use reference counting to handle compute memory pool.

Jason Ekstrand (2):

- intel/eu: Set EXECUTE_1 when setting the rounding mode in cr0
- intel/blorp: Support blits and clears on surfaces with offsets

Jose Dapena Paz (1):

- mesa: do not leak ctx->Shader.ReferencedProgram references

Kai Wasserbäch (1):

- opencl: autotools: Fix linking order for OpenCL target

Marek Olšák (3):

- st/mesa: simplify lastLevel determination in st_finalize_texture
- radeonsi: fix incorrect parentheses around VS-PS varying elimination
- mesa: handle GL_UNSIGNED_INT64_ARB properly (v2)

Michel Dänzer (1):

- dri3: Stricter SBC wraparound handling

Nanley Chery (4):

- i965: Add and use a getter for the miptree aux buffer
- i965: Add and use a single miptree aux_buf field
- i965/miptree: Fix handling of uninitialized MCS buffers
- i965/miptree: Zero-initialize CCS_D buffers

Samuel Pitoiset (2):

- spirv: fix visiting inner loops with same break/continue block
- radv: fix centroid interpolation

Stuart Young (1):

- etnaviv: Fix missing rnndb file in tarballs

Thierry Reding (3):

- tegra: Treat resources with modifiers as scanout
- tegra: Fix scanout resources without modifiers
- tegra: Remove usage of non-stable UAPI

Timothy Arceri (1):

- mesa: add glUniform*ui{v} support to display lists

4.82 Mesa 18.1.0 Release Notes / May 18 2018

Mesa 18.1.0 is a new development release. People who are concerned with stability and reliability should stick with a previous release or wait for Mesa 18.1.1.

Mesa 18.1.0 implements the OpenGL 4.5 API, but the version reported by glGetString(GL_VERSION) or glGetIntegerv(GL_MAJOR_VERSION) / glGetIntegerv(GL_MINOR_VERSION) depends on the particular driver being used. Some drivers don't support all the features required in OpenGL 4.5. OpenGL 4.5 is **only** available if requested at context creation. Compatibility contexts may report a lower version depending on each driver.

4.82.1 SHA256 checksums

```
b1c1dbb42597190503d3abc518b12de880623f097c6cb6c293ecf69ae87e6fbf  mesa-18.1.0.tar.gz
c855c5b67ef993b7621f76d8b120769ec0415f1c3616eaff44ef7f7f300aceba  mesa-18.1.0.tar.xz
```

4.82.2 New features

Note: some of the new features are only available with certain drivers.

- OpenGL 3.1 with ARB_compatibility on nv50, nvc0, r600, radeonsi, softpipe, llvmpipe, svga
- GL_ARB_bindless_texture on nvc0/maxwell+
- GL_ARB_transform_feedback_overflow_query on nvc0
- GL_EXT_semaphore on radeonsi
- GL_EXT_semaphore_fd on radeonsi
- GL_EXT_shader_framebuffer_fetch on i965 on desktop GL (GLES was already supported)
- GL_EXT_shader_framebuffer_fetch_non_coherent on i965
- GL_KHR_blend_equation_advanced on radeonsi
- Disk shader cache support for i965 enabled by default

4.82.3 Bug fixes

- Bug 90311 - Fail to build libglx with clang at linking stage
- Bug 91808 - trine1 misrender r600g
- Bug 95009 - [SNB] amd_shader_trinary_minmax.execution.built-in-functions.gs-mid3-ivec2-ivec2-ivec2 intermittent
- Bug 95012 - [SNB] glsl-1_50.execution.built-in-functions.gs-op tests intermittent
- Bug 98281 - 'message's in ctx->Debug.LogMessages[] seem to leak.
- Bug 99549 - pp: Failed to translate a shader
- Bug 100259 - [EGL] [GBM] undefined reference to 'gbm_bo_create_with_modifiers'
- Bug 101408 - [Gen8+] Xonotic fails to render one of the weapons
- Bug 101442 - Piglit shaders@ssa@fs-if-def-else-break fails with sb but passes with R600_DEBUG=nosb
- Bug 102342 - mesa-17.1.7/src/gallium/auxiliary/pipebuffer/pb_cache.c:169]: (style) Suspicious condition

- Bug 102542 - mesa-17.2.0/src/gallium/state_trackers/nine/nine_ff.c:1938: bad assignment ?
- Bug 102905 - [R600] Miscompilation of TGSI to VLIW causes artifacts in Gallium Nine with Crysis2 bump mapping
- Bug 103006 - [OpenGL CTS] [HSW] KHR-GL45.vertex_attrib_binding.basic-inputL-case1
- Bug 103142 - R600g+sb: optimizer apparently stuck in an endless loop
- Bug 103626 -
- Bug 103746 - [BDW BSW SKL KBL] dEQP-GLES31.functional.copy_image regressions
- Bug 104302 - Wolfenstein 2 (2017) under wine graphical artifacting on RADV
- Bug 104335 - [OpenGL CTS][SKL,KBL] KHR-GL45.vertex_attrib_64bit.limits_test occasionally fails
- Bug 104625 - semicolon after if
- Bug 104636 - [BSW/HD400] Aztec Ruins GL version GPU hangs
- Bug 104642 - Android: NULL pointer dereference with i965 mesa-dev, seems build_id_length related
- Bug 104654 - r600/sb: Alien Isolation GPU lock
- Bug 104668 - dEQP-GLES31.functional.shaders.linkage.uniform.block.differing_precision regression
- Bug 104717 - Rocket League: grass rendering broken with nir
- Bug 104732 - [radv] Binding descriptor sets disturbs other pipeline bindings
- Bug 104741 - Graphic corruption for Android apps Telegram and KineMaster
- Bug 104762 - Various segfaults/problems in qt/plasma
- Bug 104777 - Attaching multiple shader objects for the same stage to a GLSL program triggers a linker error
- Bug 104794 - piglit.spec.arb_internalformat_query2.samples and num_sample_counts pname checks
- Bug 104803 - SIGSEGV in state_tracker/st_glsl_to_tgsi_temprename.cpp
- Bug 104863 - 186 assertions in piglit
- Bug 104884 - memory leak with intel i965 mesa when running android container in Ubuntu
- Bug 104905 - SpvOpFOrdEqual doesn't return correct results for NaNs
- Bug 104908 - Texture Compression Hint not converted to enum16
- Bug 104915 - Indexed SHADING_LANGUAGE_VERSION query not supported
- Bug 104923 - anv: Dota2 rendering corruption
- Bug 104989 - [r600] [bisected] OpenGL applications can't render anything at all
- Bug 105013 - [regression] GLX+VA-API+clutter-gst video playback is corrupt with Mesa 17.3 (but is fine with 17.2)
- Bug 105026 - glxgears asserts with pp_jimenezmlaa=1
- Bug 105029 - simdlib_512_avx512.inl:371:57: error: could not convert '_mm512_mask_blend_epi32((__mmask16)(ImmT), a, b)' from '__m512i' {aka '__vector(8) long long int'} to 'SIMDImpl::SIMD512Impl::Float'
- Bug 105052 -
- Bug 105065 - Qt Programs occasionally fail to render with new Mesa (glGetProgramBinary)
- Bug 105067 -

- Bug 105088 - brw_nir_uniforms.cpp:256:10: error: non-constant-expression cannot be narrowed
- Bug 105098 - [RADV] GPU freeze with simple Vulkan App
- Bug 105103 - Wayland master causes Mesa to fail to compile
- Bug 105120 - meson build broken
- Bug 105161 - KHR_blend_equation_advanced doesn't work in GLSL 1.10-1.40 shaders
- Bug 105183 - Weird assertion in NIR linker
- Bug 105211 - build failure after zwp_dmabuf commit if wayland-protocols is not installed
- Bug 105224 - Webgl Pointclouds flickers
- Bug 105229 - [KBL SKL BDW HSW] [Regression] KHR-GLES31.core.shader_image_load_store.advanced-sso-simple failures
- Bug 105238 - ast.h:648:16: error: union member 'i' has a non-trivial constructor
- Bug 105255 - Waiting for fences without waitAll is not implemented
- Bug 105262 - [R600] [BISECTED] ttf fonts are invisible in many programs
- Bug 105271 - WebGL2 shader crashes i965_dri.so 17.3.3
- Bug 105274 -
- Bug 105290 -
- Bug 105292 - vkGetQueryPoolResults returns incorrect query status for large query buffers (bisected)
- Bug 105317 - The GPU Vega 56 was hang while try to pass #GraphicsFuzz shader15 test
- Bug 105320 - Storage texel buffer access produces wrong results (RX Vega)
- Bug 105374 - texture3d, a SaschaWillems demo, assert fails
- Bug 105436 - Blinking textures in UT2004 [bisected]
- Bug 105440 - GEN7: rendering issue on citra
- Bug 105442 - Hang when running nine ff lighting shader with radeonsi
- Bug 105444 - Enable GL disk shader cache when transform feedback is enabled
- Bug 105464 -
- Bug 105471 - [g33] [bisected] dEQP-GLES2.functional.shaders failures
- Bug 105497 - shader-db crashes on 72 core system after ast_type_qualifier bitset change
- Bug 105529 - u_debug_stack.c:268: error: #pragma GCC diagnostic not allowed inside functions
- Bug 105567 - meson/ninja: 1. mesa/vdpau incorrect symlinks in DESTDIR and 2. Ddri-drivers-path Dvdpau-libs-path overrides DESTDIR
- Bug 105621 - Build failure on GNOME Continuous
- Bug 105634 - Android build test fails when building brw_oa_metrics.c
- Bug 105670 -
- Bug 105704 -
- Bug 105717 - [bisected] Mesa build tests fails: BIGENDIAN_CPU or LITTLEENDIAN_CPU must be defined
- Bug 105737 - st_tests_common.cpp:140:42: error: no matching function for call to 'tgsi_get_opcode_info'
- Bug 105738 - commit f7ffa504a065dc2631fd38cc5fe885b277f4e7e7 causes artifacting in radv

- Bug 105740 - glsl_types.cpp(524): error: a dynamically-initialized local static variable is not allowed inside of a statement expression
- Bug 105775 - SI reaches the maximum IB size in dwords and fail to submit
- Bug 105807 - [Regression, bisected]: 3D Rendering not working correctly in Warhammer 40k: Dawn of War II
- Bug 105817 - scons build broken by glSpecializeShaderARB
- Bug 105820 - [m32] piglit regressions relinking program without shaders
- Bug 105942 - Graphical artefacts after update to mesa 18.0.0-2
- Bug 105952 - radv causes GPU hang on SI
- Bug 105960 - [bisected] meson build test fails with: undefined reference to 'etna_pm_create_query'
- Bug 105994 - surface state leak when creating and destroying image views with aspectMask depth and stencil
- Bug 106074 - radv: si_scissor_from_viewport returns incorrect result when using half-pixel viewport offset
- Bug 106126 - eglMakeCurrent does not always ensure dri_drawable->update_drawable_info has been called for a new EGLSurface if another has been created and destroyed first
- Bug 106131 - meson/ninja build missing file gtest.h
- Bug 106133 - make check "OSError: [Errno 24] Too many open files"
- Bug 106147 - SIGBUS in write_reloc() when Sacha Willems' "texture3d" Vulkan demo starts
- Bug 106174 - vulkan dota2 broken (segfaulting), found bug commit
- Bug 106180 - [bisected] radv vulkan smoke test black screen (Add support for DRI3 v1.2)
- Bug 106243 - [kbl] GPU HANG: 9:0:0x85dffffb, in Cinnamon
- Bug 106450 -
- Bug 106462 - piglit.spec.arb_vertex_array_bgra.get regression

4.82.4 Changes

- Remove incomplete GLX_SGIX_swap_barrier stubs from the Xlib libGL
- Remove incomplete GLX_SGIX_swap_group stubs from the Xlib libGL

4.83 Mesa 18.0.4 Release Notes / May 17, 2018

Mesa 18.0.4 is a bug fix release which fixes bugs found since the 18.0.3 release.

Mesa 18.0.4 implements the OpenGL 4.5 API, but the version reported by glGetString(GL_VERSION) or glGetIntegerv(GL_MAJOR_VERSION) / glGetIntegerv(GL_MINOR_VERSION) depends on the particular driver being used. Some drivers don't support all the features required in OpenGL 4.5. OpenGL 4.5 is **only** available if requested at context creation because compatibility contexts are not supported.

4.83.1 SHA256 checksums

```
d1dc3469faccdd73439479426952d71a9e8f684e8d03b6687063c12b13430801  mesa-18.0.4.tar.gz
1f3bcfe7cef0a5c20dae2b41df5d7e0a985e06be0183fa4d43b6068fcba2920f  mesa-18.0.4.tar.xz
```

4.83.2 New features

None

4.83.3 Bug fixes

- Bug 91808 - trine1 misrender r600g
- Bug 100430 - [radv] graphical glitches on dolphin emulator
- Bug 106243 - [kbl] GPU HANG: 9:0:0x85dffffb, in Cinnamon
- Bug 106480 - A2B10G10R10_SNORM vertex attribute doesn't work.

4.83.4 Changes

Bas Nieuwenhuizen (3):

- radv: Translate logic ops.
- radv: Fix up 2_10_10_10 alpha sign.
- radv: Disable texel buffers with A2 SNORM/SSCALED/SINT for pre-vega.

Dave Airlie (3):

- r600: fix constant buffer bounds.
- radv: resolve all layers in compute resolve path.
- radv: use compute path for multi-layer images.

Deepak Rawat (1):

- egl/x11: Send invalidate to driver on copy_region path in swap_buffer

Ian Romanick (1):

- mesa: Add missing support for glFogiv(GL_FOG_DISTANCE_MODE_NV)

Jan Vesely (8):

- clover: Add explicit virtual destructor to argument class
- eg/compute: Drop reference on code_bo in destructor.
- r600: Cleanup constant buffers on context destruction
- eg/compute: Drop reference to kernel_param bo in destructor
- pipe-loader: Free driver_name in error path
- gallium/auxiliary: Add helper function to count the number of entries in hash table
- winsys/radeon: Destroy fd_hash table when the last winsys is removed.
- winsys/amdgpu: Destroy dev_hash table when the last winsys is removed.

Jason Ekstrand (1):

- i965,anv: Set the CS stall bit on the ISP disable PIPE_CONTROL

Jose Maria Casanova Crespo (2):

- intel/compiler: fix 16-bit int brw_negate_immediate and brw_abs_immediate

- intel/compiler: fix brw_imm_w for negative 16-bit integers

Juan A. Suarez Romero (7):

- docs: add sha256 checksums for 18.0.3
- cherry-ignore: add explicit 18.1 only nominations
- cherry-ignore: glsl: change ast_type_qualifier bitset size to work around GCC 5.4 bug
- cherry-ignore: mesa: fix glGetInteger/Float/etc queries for vertex arrays attribs
- cherry-ignore: mesa: revert GL_[SECONDARY_]COLOR_ARRAY_SIZE glGet type to TYPE_INT
- cherry-ignore: radv/resolve: do fmask decompress on all layers.
- Update version to 18.0.4

Kai Wasserbäch (1):

- opencl: autotools: Fix linking order for OpenCL target

Kenneth Graunke (1):

- i965: Don't leak blorp on Gen4-5.

Lionel Landwerlin (2):

- i965: require pixel scoreboard stall prior to ISP disable
- anv: emit pixel scoreboard stall before ISP disable

Matthew Nicholls (1):

- radv: fix multisample image copies

Neil Roberts (1):

- spirv: Apply OriginUpperLeft to FragCoord

Rhys Perry (1):

- mesa: fix error handling in get_framebuffer_parameteriv

Ross Burton (1):

- src/intel/Makefile.vulkan.am: add missing MKDIR_GEN

4.84 Mesa 18.0.3 Release Notes / May 7, 2018

Mesa 18.0.3 is a bug fix release which fixes bugs found since the 18.0.2 release.

Mesa 18.0.3 implements the OpenGL 4.5 API, but the version reported by glGetString(GL_VERSION) or glGetIntegerv(GL_MAJOR_VERSION) / glGetIntegerv(GL_MINOR_VERSION) depends on the particular driver being used. Some drivers don't support all the features required in OpenGL 4.5. OpenGL 4.5 is **only** available if requested at context creation because compatibility contexts are not supported.

4.84.1 SHA256 checksums

```
58cc5c5b1ab2a44e6e47f18ef6c29836ad06f95450adce635ce3c317507a171b  mesa-18.0.3.tar.gz
099d9667327a76a61741a533f95067d76ea71a656e66b91507b3c0caf1d49e30  mesa-18.0.3.tar.xz
```

4.84.2 New features

None

4.84.3 Bug fixes

- Bug 105374 - texture3d, a SaschaWillems demo, assert fails
- Bug 106147 - SIGBUS in write_reloc() when Sacha Willems' "texture3d" Vulkan demo starts

4.84.4 Changes

Andres Rodriguez (1):

- radv/winsys: fix leaking resources from bo's imported by fd

Boyuan Zhang (1):

- radeon/vcn: fix mpeg4 msg buffer settings

Eric Anholt (1):

- gallium/util: Fix incorrect refcounting of separate stencil.

Jason Ekstrand (1):

- anv/allocator: Don't shrink either end of the block pool

Juan A. Suarez Romero (3):

- docs: add sha256 checksums for 18.0.2
- cherry-ignore: add explicit 18.1 only nominations
- Update version to 18.0.3

Leo Liu (1):

- st/omx/enc: fix blit setup for YUV LoadImage

Marek Olšák (2):

- util/u_queue: fix a deadlock in util_queue_finish
- radeonsi/gfx9: workaround for INTERP with indirect indexing

Nanley Chery (1):

- i965/tex_image: Avoid the ASTC LDR workaround on gen9lp

Samuel Pitoiset (1):

- radv: compute the number of subpass attachments correctly

4.85 Mesa 18.0.2 Release Notes / April 28, 2018

Mesa 18.0.2 is a bug fix release which fixes bugs found since the 18.0.1 release.

Mesa 18.0.2 implements the OpenGL 4.5 API, but the version reported by glGetString(GL_VERSION) or glGetIntegerv(GL_MAJOR_VERSION) / glGetIntegerv(GL_MINOR_VERSION) depends on the particular driver being used. Some drivers don't support all the features required in OpenGL 4.5. OpenGL 4.5 is **only** available if requested at context creation because compatibility contexts are not supported.

4.85.1 SHA256 checksums

```
SHA256: ffd8dfe3337b474a3baa085f0e7ef1a32c7cdc3bed1ad810b2633919a9324840  mesa-18.0.2.
↪tar.gz
SHA256: 98fa159768482dc568b9f8bf0f36c7acb823fa47428ffd650b40784f16b9e7b3  mesa-18.0.2.
↪tar.xz
```

4.85.2 New features

None

4.85.3 Bug fixes

- Bug 95009 - [SNB] amd_shader_trinary_minmax.execution.built-in-functions.gs-mid3-ivec2-ivec2-ivec2 intermittent
- Bug 95012 - [SNB] glsl-1_50.execution.built-in-functions.gs-op tests intermittent
- Bug 98281 - 'message's in ctx->Debug.LogMessages[] seem to leak.
- Bug 105320 - Storage texel buffer access produces wrong results (RX Vega)
- Bug 105775 - SI reaches the maximum IB size in dwords and fail to submit
- Bug 105994 - surface state leak when creating and destroying image views with aspectMask depth and stencil
- Bug 106074 - radv: si_scissor_from_viewport returns incorrect result when using half-pixel viewport offset
- Bug 106126 - eglMakeCurrent does not always ensure dri_drawable->update_drawable_info has been called for a new EGLSurface if another has been created and destroyed first

4.85.4 Changes

Bas Nieuwenhuizen (2):

- ac/nir: Make the GFX9 buffer size fix apply to image loads/atomics too.
- radv: Mark GTT memory as device local for APUs.

Dylan Baker (2):

- bin/install_megadrivers: fix DESTDIR and -D*-path
- meson: don't build classic mesa tests without dri_drivers

Ian Romanick (1):

- intel/compiler: Add scheduler deps for instructions that implicitly read g0

Jason Ekstrand (1):

- i965/fs: Return mlen * 8 for size_read() for INTERPOLATE_AT_*

Johan Klokkhammer Helsing (1):

- st/dri: Fix dangling pointer to a destroyed dri_drawable

Juan A. Suarez Romero (4):

- docs: add sha256 checksums for 18.0.1

- travis: radv needs LLVM 4.0
- cherry-ignore: add explicit 18.1 only nominations
- Update version to 18.0.2

Kenneth Graunke (1):

- i965: Fix shadow batches to be the same size as the real BO.

Lionel Landwerlin (1):

- anv: fix number of planes for depth & stencil

Lucas Stach (1):

- etnaviv: fix texture_format_needs_swiz

Marek Olšák (3):

- radeonsi/gfx9: fix a hang with an empty first IB
- glsl_to_tgsi: try harder to lower unsupported ir_binop_vector_extract
- Revert "st/dri: Fix dangling pointer to a destroyed dri_drawable"

Samuel Pitoiset (2):

- radv: fix scissor computation when using half-pixel viewport offset
- radv/winsys: allow to submit up to 4 IBs for chips without chaining

Thomas Hellstrom (1):

- svga: Fix incorrect advertizing of EGL_KHR_gl_colorspace

Timothy Arceri (1):

- mesa: free debug messages when destroying the debug state

4.86 Mesa 18.0.1 Release Notes / April 18, 2018

Mesa 18.0.1 is a bug fix release which fixes bugs found since the 18.0.0 release.

Mesa 18.0.1 implements the OpenGL 4.5 API, but the version reported by glGetString(GL_VERSION) or glGetIntegerv(GL_MAJOR_VERSION) / glGetIntegerv(GL_MINOR_VERSION) depends on the particular driver being used. Some drivers don't support all the features required in OpenGL 4.5. OpenGL 4.5 is **only** available if requested at context creation because compatibility contexts are not supported.

4.86.1 SHA256 checksums

```
0c93ba892c0610f5dd87f2e2673b9445187995c395b3ddb33fd4260bfb291e89  mesa-18.0.1.tar.gz
b2d2f5b5dbaab13e15cb0dcb5ec81887467f55ebc9625945b303a3647cd87954  mesa-18.0.1.tar.xz
```

4.86.2 New features

None

4.86.3 Bug fixes

- Bug 101408 - [Gen8+] Xonotic fails to render one of the weapons
- Bug 102342 - mesa-17.1.7/src/gallium/auxiliary/pipebuffer/pb_cache.c:169]: (style) Suspicious condition
- Bug 102542 - mesa-17.2.0/src/gallium/state_trackers/nine/nine_ff.c:1938: bad assignment ?
- Bug 105317 - The GPU Vega 56 was hang while try to pass #GraphicsFuzz shader15 test
- Bug 105440 - GEN7: rendering issue on citra
- Bug 105442 - Hang when running nine ff lighting shader with radeonsi
- Bug 105567 - meson/ninja: 1. mesa/vdpau incorrect symlinks in DESTDIR and 2. Ddri-drivers-path Dvdpau-libs-path overrides DESTDIR
- Bug 105670 - [regression][hang] Trine1EE hangs GPU after loading screen on Mesa3D-17.3 and later
- Bug 105704 - compiler assertion hit
- Bug 105717 - [bisected] Mesa build tests fails: BIGENDIAN_CPU or LITTLEENDIAN_CPU must be defined
- Bug 105942 - Graphical artefacts after update to mesa 18.0.0-2

4.86.4 Changes

Andres Gomez (2):

- dri_util: when overriding, always reset the core version
- mesa: adds some comments regarding MESA_GLES_VERSION_OVERRIDE usage

Axel Davy (5):

- st/nine: Fix bad tracking of vs textures for NINESBT_ALL
- st/nine: Fixes warning about implicit conversion
- st/nine: Fix non inversible matrix check
- st/nine: Declare lighting consts for ff shaders
- st/nine: Do not use scratch for face register

Bas Nieuwenhuizen (3):

- ac/nir: Add workaround for GFX9 buffer views.
- radv: Don't set instance count using predication.
- radv: Always reset draw user SGPRs after secondary command buffer.

Caio Marcelo de Oliveira Filho (1):

- anv/pipeline: fail if TCS/TES compile fail

Daniel Stone (1):

- st/dri: Initialise modifier to INVALID for DRI2

Derek Foreman (1):

- egl/wayland: Make swrast display_sync the correct queue

Dylan Baker (4):

- meson: don't use compiler.has_header

- autotools: include meson_get_version
- meson: Set .so version for xa like autotools does
- meson: fix megadriver symlinking

Emil Velikov (1):

- docs: add sha256 checksums for 18.0.0

Eric Engestrom (3):

- meson/configure: detect endian.h instead of trying to guess when it's available
- docs: fix 18.0 release note version
- gbm: remove never-implemented function

Henri Verbeet (1):

- mesa: Inherit texture view multi-sample information from the original texture images.

Iago Toral Quiroga (1):

- compiler/spirv: set is_shadow for depth comparitor sampling opcodes

Ian Romanick (1):

- i965/vec4: Fix null destination register in 3-source instructions

Jason Ekstrand (4):

- nir/vars_to_ssa: Remove copies from the correct set
- nir/lower_indirect_derefs: Support interp_var_at intrinsics
- intel/vec4: Set channel_sizes for MOV_INDIRECT sources
- nir/lower_vec_to_movs: Only coalesce if the vec had a SSA destination

Juan A. Suarez Romero (5):

- cherry-ignore anv: Be more careful about fast-clear colors
- cherry-ignore: ac/shader: fix vertex input with components.
- cherry-ignore: radv: handle exporting view index to fragment shader. (v1.1)
- cherry-ignore: omx: always define ENABLE_ST_OMX_{BELLAGIO,TIZONIA}
- Update version to 18.0.1

Leo Liu (1):

- radeon/vce: move feedback command inside of destroy function

Lionel Landwerlin (1):

- i965/perf: fix config registration when uploading to kernel

Marc Dietrich (1):

- meson: fix HAVE_LLVM version define in meson build

Marek Olšák (1):

- mesa: simplify MESA_GL_VERSION_OVERRIDE behavior of API override

Mark Thompson (1):

- st/va: Enable vaExportSurfaceHandle()

Rob Clark (3):

- nir: fix per_vertex_output intrinsic
- freedreno/a5xx: fix page faults on last level
- freedreno/a5xx: don't align height for PIPE_BUFFER

Samuel Pitoiset (2):

- radv: fix picking the method for resolve subpass
- radv: fix radv_layout_dcc_compressed() when image doesn't have DCC

Sergii Romantsov (1):

- i965: Extend the negative 32-bit deltas to 64-bits

Timothy Arceri (7):

- ac: add if/loop build helpers
- radeonsi: make use of if/loop build helpers in ac
- ac: make use of if/loop build helpers
- glsl: fix infinite loop caused by bug in loop unrolling pass
- nir: fix crash in loop unroll corner case
- gallium/pipebuffer: fix parenthesis location
- glsl: always call do_lower_jumps() after loop unrolling

Xiong, James (1):

- i965: return the fourcc saved in __DRIimage when possible

4.87 Mesa 17.3.9 Release Notes / April 18, 2018

Mesa 17.3.9 is a bug fix release which fixes bugs found since the 17.3.8 release.

Mesa 17.3.9 implements the OpenGL 4.5 API, but the version reported by glGetString(GL_VERSION) or glGetIntegerv(GL_MAJOR_VERSION) / glGetIntegerv(GL_MINOR_VERSION) depends on the particular driver being used. Some drivers don't support all the features required in OpenGL 4.5. OpenGL 4.5 is **only** available if requested at context creation because compatibility contexts are not supported.

4.87.1 SHA256 checksums

```
4d625f65a1ff4cd8cfeb39e38f047507c6dea047502a0d53113c96f54588f340  mesa-17.3.9.tar.gz
c5beb5fc05f0e0c294fefe1a393ee118cb67e27a4dca417d77c297f7d4b6e479  mesa-17.3.9.tar.xz
```

4.87.2 New features

None

4.87.3 Bug fixes

- Bug 98281 - 'message's in ctx->Debug.LogMessages[] seem to leak.
- Bug 101408 - [Gen8+] Xonotic fails to render one of the weapons
- Bug 102342 - mesa-17.1.7/src/gallium/auxiliary/pipebuffer/pb_cache.c:169]: (style) Suspicious condition
- Bug 105317 - The GPU Vega 56 was hang while try to pass #GraphicsFuzz shader15 test
- Bug 105440 - GEN7: rendering issue on citra
- Bug 105442 - Hang when running nine ff lighting shader with radeonsi
- Bug 105994 - surface state leak when creating and destroying image views with aspectMask depth and stencil

4.87.4 Changes

Andres Gomez (2):

- dri_util: when overriding, always reset the core version
- mesa: adds some comments regarding MESA_GLES_VERSION_OVERRIDE usage

Axel Davy (2):

- st/nine: Declare lighting consts for ff shaders
- st/nine: Do not use scratch for face register

Bas Nieuwenhuizen (1):

- ac/nir: Add workaround for GFX9 buffer views.

Daniel Stone (1):

- st/dri: Initialise modifier to INVALID for DRI2

Emil Velikov (1):

- glsl: remove unreachable assert()

Eric Engestrom (1):

- gbm: remove never-implemented function

Henri Verbeet (1):

- mesa: Inherit texture view multi-sample information from the original texture images.

Iago Toral Quiroga (1):

- compiler/spirv: set is_shadow for depth comparitor sampling opcodes

Jason Ekstrand (4):

- nir/vars_to_ssa: Remove copies from the correct set
- nir/lower_indirect_derefs: Support interp_var_at intrinsics
- intel/vec4: Set channel_sizes for MOV_INDIRECT sources
- nir/lower_vec_to_movs: Only coalesce if the vec had a SSA destination

Juan A. Suarez Romero (3):

- docs: add sha256 checksums for 17.3.8

- cherry-ignore: Explicit 18.0 only nominations
- Update version to 17.3.9

Lionel Landwerlin (1):

- anv: fix number of planes for depth & stencil

Marek Olšák (1):

- mesa: simplify MESA_GL_VERSION_OVERRIDE behavior of API override

Samuel Pitoiset (1):

- radv: fix picking the method for resolve subpass

Sergii Romantsov (1):

- i965: Extend the negative 32-bit deltas to 64-bits

Timothy Arceri (6):

- gallium/pipebuffer: fix parenthesis location
- glsl: always call do_lower_jumps() after loop unrolling
- ac: add if/loop build helpers
- radeonsi: make use of if/loop build helpers in ac
- ac: make use of if/loop build helpers
- mesa: free debug messages when destroying the debug state

Xiong, James (1):

- i965: return the fourcc saved in __DRIimage when possible

4.88 Mesa 17.3.8 Release Notes / April 03, 2018

Mesa 17.3.8 is a bug fix release which fixes bugs found since the 17.3.7 release.

Mesa 17.3.8 implements the OpenGL 4.5 API, but the version reported by glGetString(GL_VERSION) or glGetIntegerv(GL_MAJOR_VERSION) / glGetIntegerv(GL_MINOR_VERSION) depends on the particular driver being used. Some drivers don't support all the features required in OpenGL 4.5. OpenGL 4.5 is **only** available if requested at context creation because compatibility contexts are not supported.

4.88.1 SHA256 checksums

```
175d2ca9be2af3a8db6cd603986096d75da70f59699528d7b6675d542a305e23  mesa-17.3.8.tar.gz
8f9d9bf281c48e4a8f5228816577263b4c655248dc7666e75034ab422951a6b1  mesa-17.3.8.tar.xz
```

4.88.2 New features

None

4.88.3 Bug fixes

- Bug 102542 - mesa-17.2.0/src/gallium/state_trackers/nine/nine_ff.c:1938: bad assignment ?
- Bug 103746 - [BDW BSW SKL KBL] dEQP-GLES31.functional.copy_image regressions
- Bug 104636 - [BSW/HD400] Aztec Ruins GL version GPU hangs
- Bug 105290 - [BSW/HD400] SynMark OglCSDof GPU hangs when shaders come from cache
- Bug 105464 - Reading per-patch outputs in Tessellation Control Shader returns undefined values
- Bug 105670 - [regression][hang] Trine1EE hangs GPU after loading screen on Mesa3D-17.3 and later
- Bug 105704 - compiler assertion hit
- Bug 105717 - [bisected] Mesa build tests fails: BIGENDIAN_CPU or LITTLEENDIAN_CPU must be defined

4.88.4 Changes

Axel Davy (3):

- st/nine: Fix bad tracking of vs textures for NINESBT_ALL
- st/nine: Fixes warning about implicit conversion
- st/nine: Fix non inversible matrix check

Caio Marcelo de Oliveira Filho (1):

- anv/pipeline: fail if TCS/TES compile fail

Dave Airlie (1):

- radv: get correct offset into LDS for indexed vars.

Derek Foreman (1):

- egl/wayland: Make swrast display_sync the correct queue

Eric Engestrom (1):

- meson/configure: detect endian.h instead of trying to guess when it's available

Ian Romanick (2):

- mesa: Don't write to user buffer in glGetTexParameterIuiv on error
- i965/vec4: Fix null destination register in 3-source instructions

Jason Ekstrand (1):

- i965: Emit texture cache invalidates around blorp_copy

Jordan Justen (2):

- i965: Calculate thread_count in brw_alloc_stage_scratch
- i965: Hard code CS scratch_ids_per_subslice for Cherryview

Juan A. Suarez Romero (6):

- docs: add sha256 checksums for 17.3.7
- cherry-ignore: ac/nir: pass the nir variable through tcs loading.
- cherry-ignore: radv: handle exporting view index to fragment shader. (v1.1)

- cherry-ignore: omx: always define ENABLE_ST_OMX_{BELLAGIO,TIZONIA}
- cherry-ignore: docs: fix 18.0 release note version
- Update version to 17.3.8

Leo Liu (1):

- radeon/vce: move feedback command inside of destroy function

Marek Olšák (1):

- st/dri: fix OpenGL-OpenCL interop for GL_TEXTURE_BUFFER

Rob Clark (1):

- nir: fix per_vertex_output intrinsic

Timothy Arceri (2):

- glsl: fix infinite loop caused by bug in loop unrolling pass
- nir: fix crash in loop unroll corner case

4.89 Mesa 18.0.0 Release Notes / March 27 2018

Mesa 18.0.0 is a new development release. People who are concerned with stability and reliability should stick with a previous release or wait for Mesa 18.0.1.

Mesa 18.0.0 implements the OpenGL 4.5 API, but the version reported by glGetString(GL_VERSION) or glGetIntegerv(GL_MAJOR_VERSION) / glGetIntegerv(GL_MINOR_VERSION) depends on the particular driver being used. Some drivers don't support all the features required in OpenGL 4.5. OpenGL 4.5 is **only** available if requested at context creation because compatibility contexts are not supported.

4.89.1 SHA256 checksums

```
93c2d3504b2871ac2146603fb1270f341d36a39695e2950a469c5eac74f98457  mesa-18.0.0.tar.gz
694e5c3d37717d23258c1f88bc134223c5d1aac70518d2f9134d6df3ee791eea  mesa-18.0.0.tar.xz
```

4.89.2 New features

Note: some of the new features are only available with certain drivers.

- Disk shader cache support for i965 when MESA_GLSL_CACHE_DISABLE environment variable is set to "0" or "false"
- GL_ARB_shader_atomic_counters and GL_ARB_shader_atomic_counter_ops on r600/evergreen+
- GL_ARB_shader_image_load_store and GL_ARB_shader_image_size on r600/evergreen+
- GL_ARB_shader_storage_buffer_object on r600/evergreen+
- GL_ARB_compute_shader on r600/evergreen+
- GL_ARB_cull_distance on r600/evergreen+
- GL_ARB_enhanced_layouts on r600/evergreen+
- GL_ARB_bindless_texture on nvc0/kepler

- OpenGL 4.3 on r600/evergreen with hw fp64 support
- Support 1 binary format for GL_ARB_get_program_binary on i965. (For the 18.0 release, 0 formats continue to be supported in compatibility profiles.)
- Cannonlake support on i965 and anv

4.89.3 Bug fixes

- Bug 85564 - Dead Island rendering issues
- Bug 90311 - Fail to build libglx with clang at linking stage
- Bug 92363 - [BSW/BDW] ogles1conform Gets test fails
- Bug 94739 - Mesa 11.1.2 implementation error: bad format MESA_FORMAT_Z_FLOAT32 in _mesa_unpack_uint_24_8_depth_stencil_row
- Bug 97532 - Regression: GLB 2.7 & Glmark-2 GLES versions segfault due to linker precision error (259fc505) on dead variable
- Bug 97852 - Unreal Engine corrupted preview viewport
- Bug 100438 - glsl/ir.cpp:1376: ir_dereference_variable::ir_dereference_variable(ir_variable*): Assertion 'var != NULL' failed.
- Bug 101378 - interpolateAtSample check for input parameter is too strict
- Bug 101442 - Piglit shaders@ssa@fs-if-def-else-break fails with sb but passes with R600_DEBUG=nosb
- Bug 101560 - SPIR-V OpSwitch with int64 not supported even though shaderInt64 is true
- Bug 101691 - gfx corruption on windowed 3d-apps running on dGPU
- Bug 102177 - [SKL] ES31-CTS.core.sepshaderobjs.StateInteraction fails sporadically
- Bug 102264 - Missing MESA_FORMAT_{B8G8R8A8,B8G8R8X8}_SRGB formats
- Bug 102354 - Mesa 17.2 no longer can give SRGB-capable framebuffer on i965, even though Mesa 17.1.x does.
- Bug 102358 - WarThunder freezes at start, with activated vsync (vblank_mode=2)
- Bug 102435 - [skl,kbl] [drm] GPU hang in Valve games based on Source 1
- Bug 102503 - Report SRGB framebuffer to SuperTuxKart to workaround SuperTuxKart crash
- Bug 102665 - test_glsl_to_tgsi_lifetime.cpp:53:67: error: '>>' should be '> >' within a nested template argument list
- Bug 102677 - [OpenGL CTS] KHR-GL45.CommonBugs.CommonBug_PerVertexValidation fails
- Bug 102680 - [OpenGL CTS] KHR-GL45.shader_ballot_tests.ShaderBallotBitmasks fails
- Bug 102710 - vkCmdBlitImage with arrayLayers > 1 fails
- Bug 102774 - [BDW] [Bisected] Absolute constant buffers break VAAPI in mpv
- Bug 102809 - Rust shadows(?) flash random colours
- Bug 102897 - Separate bind points are not implemented correctly
- Bug 102955 - HyperZ related rendering issue in ARK: Survival Evolved
- Bug 103006 - [OpenGL CTS] [HSW] KHR-GL45.vertex_attrib_binding.basic-inputL-case1
- Bug 103007 - [OpenGL CTS] [HSW] KHR-GL45.gpu_shader_fp64.fp64.max_uniform_components fails
- Bug 103085 - [ivb byt hsw] piglit.spec.arb_indirect_parameters.tf-count-arrays

- Bug 103098 - [OpenGL CTS] KHR-GL45.enhanced_layouts.varying_structure_locations fails
- Bug 103101 - [SKL][bisected] DiRT Rally GPU hang
- Bug 103115 - [BSW BXT GLK] dEQP-VK.spirv_assembly.instruction.compute.sconvert.int32_to_int64
- Bug 103128 - [softpipe] piglit fs-ldexp regression
- Bug 103142 - R600g+sb: optimizer apparently stuck in an endless loop
- Bug 103227 - [G965 G45 ILK] ES2-CTS.gtf.GL2ExtensionTests.texture_float.texture_float regression
- Bug 103283 - drm_get_device_name_for_fd is broken on FreeBSD
- Bug 103388 - Linking libcltgsi.la (llvm/codegen/libclllvm_la-common.lo) fails with "error: no match for 'operator-'" with GCC-7, Mesa from Git and current LLVM revisions
- Bug 103393 - glDispatchComputeGroupSizeARB : gl_GlobalInvocationID.x != gl_WorkGroupID.x * gl_LocalGroupSizeARB.x + gl_LocalInvocationID.x
- Bug 103412 - gallium/wgl: Another fix to context creation without prior SetPixelFormat()
- Bug 103496 - svga_screen.c:26:46: error: git_sha1.h: No such file or directory
- Bug 103513 - [build failure] radv_shader.c:683:2: error: format not a string literal and no format arguments [-Werror=format-security]
- Bug 103519 - wayland egl apps crash on start with mesa 17.2
- Bug 103529 - [GM45] GPU hang with mpv fullscreen (bisected)
- Bug 103537 - i965: Shadow of Mordor broken since commit 379b24a40d3d34ffdaaeb1b328f50e28ecb01468 on Haswell
- Bug 103544 - Graphical glitches r600 in game this war of mine linux native
- Bug 103579 - Vertex shader causes compiler to crash in SPIRV-to-NIR
- Bug 103616 - Increased difference from reference image in shaders
- Bug 103626 - [SNB] ES3-CTS.functional.shaders.precision
- Bug 103628 - [BXT, GLK, BSW] KHR-GL46.shader_ballot_tests.ShaderBallotBitmasks
- Bug 103653 - Unreal segfault since gallium/u_threaded: avoid syncs for get_query_result
- Bug 103658 - addrlib/gfx9/gfx9addrlib.cpp:727:50: error: expected expression
- Bug 103674 - u_queue.c:173:7: error: implicit declaration of function 'timespec_get' is invalid in C99
- Bug 103746 - [BDW BSW SKL KBL] dEQP-GLES31.functional.copy_image regressions
- Bug 103759 - plasma desktop corrupted rendering
- Bug 103784 - [bisected] Egl changes breaks all of EGL
- Bug 103787 - [BDW,BSW] gpu hang on spec.arb_pipeline_statistics_query.arb_pipeline_statistics_query-comp
- Bug 103801 - [i965] >Observer_ issue
- Bug 103808 - [radeonsi, bisected] World of Warcraft scribbling all over screen
- Bug 103902 - Portal 2 game hangs at startup with latest mesa dev
- Bug 103904 - Source engine-based games won't hang at start without R600_DEBUG=vs
- Bug 103909 - anv_allocator.c:113:1: error: static declaration of 'memfd_create' follows non-static declaration
- Bug 103942 - KHR-GL46.enhanced_layouts.varying* regression

- Bug 103955 - Using array in structure results in wrong GLSL compilation output
- Bug 103966 - Mesa 17.2.5 implementation error: bad format MESA_FORMAT_Z_FLOAT32 in _mesa_unpack_uint_24_8_depth_stencil_row
- Bug 103988 - Intermittent piglit failures with shader cache enabled
- Bug 104005 - [sklgt4e] GPU hangs in Car_Chase
- Bug 104119 - radv: OpBitFieldInsert produces 0 with a loop counter for Insert
- Bug 104141 - include/c11/threads_posix.h:96: undefined reference to 'pthread_once'
- Bug 104143 - r600/sb: clobbers gl_Position -> gl_FragCoord
- Bug 104163 - [GEN9+] 2-3% perf drop in GfxBench Manhattan 3.1 from "i965: Disable regular fast-clears (CCS_D) on gen9+"
- Bug 104183 - mesa-17.3.0/src/broadcom/qpu/qpu_pack.c:171]: (error) Invalid memcmp() argument
- Bug 104199 - [i965 bisected] BIO and EM Vision in >Observer_ is broken since commit af2c320190f3c73180f1610c8df955a7fa2a4d09
- Bug 104213 - NULL pointer access crashes on compiling Vulkan compute shaders after "anv: Add support for the variablePointers feature"
- Bug 104214 - Dota crashes when switching from game to desktop
- Bug 104226 - [bisected] Anvil accesses uninitialized memory while compiling shaders
- Bug 104231 - DispatchSanity_test.GL30 regression
- Bug 104246 - Talos Principle Vulkan version crash: spirv_to_nir() returns NULL entry_point
- Bug 104271 - i965: Timeout in dEQP-GLES31.functional.ssbo.layout.random.all_shared_buffer.5
- Bug 104288 - Steamroll needs allow_glsl_cross_stage_interpolation_mismatch=true
- Bug 104302 - Wolfenstein 2 (2017) under wine graphical artifacting on RADV
- Bug 104331 - [r600g] Ogre demo "TutorialUAV01" crash at r600_decompress_color_images
- Bug 104338 - NULL pointer access crash on Sacha Willems' Vulkan raytracing demo after "spirv: Add basic type validation for OpLoad, OpStore, and OpCopyMemory"
- Bug 104359 - Mesa freezes in "vtn_cfg_walk_blocks" with Sacha Willems' hdr, parallaxmapping and specializationconstants Vulkan demos
- Bug 104381 - swr fails to build since llvm-svn r321257
- Bug 104383 - [KBL] Intel GPU hang with firefox
- Bug 104411 - [CCS] lemonbar-xft GPU hang
- Bug 104424 - DOOM 2016 broken by spirv OpStore validation
- Bug 104487 - [KBL] portal2_linux GPU hang
- Bug 104490 - [radeonsi/290x] Dota2 fails to start (can't create opengl context)
- Bug 104492 - Compute Shader: Wrong alignment when assigning struct value to structured SSBO
- Bug 104546 - Crash happens when running compute pipeline after calling glxMakeCurrent two times
- Bug 104551 - Check if Mako templates for Python are installed
- Bug 104625 - semicolon after if
- Bug 104636 - [BSW/HD400] Aztec Ruins GL version GPU hangs

- Bug 104642 - Android: NULL pointer dereference with i965 mesa-dev, seems build_id_length related
- Bug 104654 - r600/sb: Alien Isolation GPU lock
- Bug 104668 - dEQP-GLES31.functional.shaders.linkage.uniform.block.differing_precision regression
- Bug 104677 - radv_generate_graphics_pipeline_key reads input rate from incorrect binding
- Bug 104690 - [G33] regression: piglit.spec.!opengl 1_4.draw-batch and gl-1_4-dlist-multidrawarrays
- Bug 104711 - [skl CCS] Oxenfree (unity engine game) hangs GPU
- Bug 104741 - Graphic corruption for Android apps Telegram and KineMaster
- Bug 104742 - [swrast] piglit gl-1.4-dlist-multidrawarrays regression
- Bug 104746 - [swrast] piglit attribs regression
- Bug 104749 - rasterizer/jitter/JitManager.cpp:252:91: error: no matching function for call to 'llvm::DIBuilder::createBasicType(const char [8], int, llvm::dwarf::TypeKind)'
- Bug 104762 - Various segfaults/problems in qt/plasma
- Bug 104777 - Attaching multiple shader objects for the same stage to a GLSL program triggers a linker error
- Bug 104884 - memory leak with intel i965 mesa when running android container in Ubuntu
- Bug 104905 - SpvOpFOrdEqual doesn't return correct results for NaNs
- Bug 104915 - Indexed SHADING_LANGUAGE_VERSION query not supported
- Bug 104923 - anv: Dota2 rendering corruption
- Bug 105013 - [regression] GLX+VA-API+clutter-gst video playback is corrupt with Mesa 17.3 (but is fine with 17.2)
- Bug 105029 - simdlib_512_avx512.inl:371:57: error: could not convert '_mm512_mask_blend_epi32((__mmask16)(ImmT), a, b)' from '__m512i' {aka '__vector(8) long long int'} to 'SIMDImpl::SIMD512Impl::Float'
- Bug 105065 - Qt Programs occasionally fail to render with new Mesa (glGetProgramBinary)
- Bug 105098 - [RADV] GPU freeze with simple Vulkan App
- Bug 105103 - Wayland master causes Mesa to fail to compile
- Bug 105120 - meson build broken
- Bug 105224 - Webgl Pointclouds flickers
- Bug 105255 - Waiting for fences without waitAll is not implemented
- Bug 105271 - WebGL2 shader crashes i965_dri.so 17.3.3
- Bug 105290 - [BSW/HD400] SynMark OglCSDof GPU hangs when shaders come from cache
- Bug 105292 - vkGetQueryPoolResults returns incorrect query status for large query buffers (bisected)
- Bug 105436 - Blinking textures in UT2004 [bisected]
- Bug 105464 - Reading per-patch outputs in Tessellation Control Shader returns undefined values

4.89.4 Changes

- Remove incomplete GLX_MESA_set_3dfx_mode from the Xlib libGL

4.90 Mesa 17.3.7 Release Notes / March 21, 2018

Mesa 17.3.7 is a bug fix release which fixes bugs found since the 17.3.7 release.

Mesa 17.3.7 implements the OpenGL 4.5 API, but the version reported by glGetString(GL_VERSION) or glGetIntegerv(GL_MAJOR_VERSION) / glGetIntegerv(GL_MINOR_VERSION) depends on the particular driver being used. Some drivers don't support all the features required in OpenGL 4.5. OpenGL 4.5 is **only** available if requested at context creation because compatibility contexts are not supported.

4.90.1 SHA256 checksums

```
f08de6d0ccb3dbca04b44790d85c3ff9e7b1cc4189d1b7c7167e5ba7d98736c0  mesa-17.3.7.tar.gz
0595904a8fba65a8fe853a84ad3c940205503b94af41e8ceed245fada777ac1e  mesa-17.3.7.tar.xz
```

4.90.2 New features

None

4.90.3 Bug fixes

- Bug 103007 - [OpenGL CTS] [HSW] KHR-GL45.gpu_shader_fp64.fp64.max_uniform_components fails
- Bug 103988 - Intermittent piglit failures with shader cache enabled
- Bug 104302 - Wolfenstein 2 (2017) under wine graphical artifacting on RADV
- Bug 104381 - swr fails to build since llvm-svn r321257
- Bug 104625 - semicolon after if
- Bug 104642 - Android: NULL pointer dereference with i965 mesa-dev, seems build_id_length related
- Bug 104654 - r600/sb: Alien Isolation GPU lock
- Bug 104905 - SpvOpFOrdEqual doesn't return correct results for NaNs
- Bug 104915 - Indexed SHADING_LANGUAGE_VERSION query not supported
- Bug 104923 - anv: Dota2 rendering corruption
- Bug 105013 - [regression] GLX+VA-API+clutter-gst video playback is corrupt with Mesa 17.3 (but is fine with 17.2)
- Bug 105029 - simdlib_512_avx512.inl:371:57: error: could not convert '_mm512_mask_blend_epi32((__mmask16)(ImmT), a, b)' from '__m512i' {aka '__vector(8) long long int'} to 'SIMDImpl::SIMD512Impl::Float'
- Bug 105098 - [RADV] GPU freeze with simple Vulkan App
- Bug 105103 - Wayland master causes Mesa to fail to compile
- Bug 105224 - Webgl Pointclouds flickers
- Bug 105255 - Waiting for fences without waitAll is not implemented
- Bug 105271 - WebGL2 shader crashes i965_dri.so 17.3.3
- Bug 105436 - Blinking textures in UT2004 [bisected]

4.90.4 Changes

Alex Smith (1):

- radv: Fix CmdCopyImage between uncompressed and compressed images

Andriy Khulap (1):

- i965: Fix RELOC_WRITE typo in brw_store_data_imm64()

Anuj Phogat (1):

- isl: Don't use surface format R32_FLOAT for typed atomic integer operations

Bas Nieuwenhuizen (6):

- radv: Always lower indirect derefs after nir_lower_global_vars_to_local.
- radeonsi: Export signalled sync file instead of -1.
- radv: Implement WaitForFences with !waitAll.
- radv: Implement waiting on non-submitted fences.
- radv: Fix copying from 3D images starting at non-zero depth.
- radv: Increase the number of dynamic uniform buffers.

Brian Paul (1):

- mesa: add missing switch case for EXTRA_VERSION_40 in check_extra()

Chuck Atkins (1):

- glx: Properly handle cases where screen creation fails

Daniel Stone (3):

- i965: Fix bugs in intel_from_planar
- egl/wayland: Fix ARGB/XRGB transposition in config map
- egl/wayland: Always use in-tree wayland-egl-backend.h

Dave Airlie (9):

- r600: fix cubemap arrays
- r600/sb/cayman: fix indirect ubo access on cayman
- r600: fix xfb stream check.
- ac/nir: to integer the args to bcsel.
- r600/cayman: fix fragcood loading recip generation.
- radv: don't support tc-compat on multisample d32s8 at all.
- virgl: remap query types to hw support.
- ac/nir: don't apply slice rounding on txf_ms
- r600: implement callstack workaround for evergreen.

Dylan Baker (2):

- glapi/check_table: Remove 'extern "C"' block
- glapi: remove APPLE extensions from test

Emil Velikov (1):

- docs: add sha256 checksums for 17.3.6

Eric Anholt (4):

- mesa: Drop incorrect A4B4G4R4 _mesa_format_matches_format_and_type() cases.
- ac/nir: Fix compiler warning about uninitialized dw_addr.
- glsl/tests: Fix strict aliasing warning about int64/double.
- glsl/tests: Fix a compiler warning about signed/unsigned loop comparison.

Francisco Jerez (1):

- i965: Fix KHR_blend_equation_advanced with some render targets.

Frank Binns (1):

- egl/dri2: fix segfault when display initialisation fails

George Kyriazis (1):

- swr/rast: blend_epi32() should return Integer, not Float

Gert Wollny (1):

- r600: Take ALU_EXTENDED into account when evaluating jump offsets

Gurchetan Singh (1):

- mesa: don't clamp just based on ARB_viewport_array extension

Iago Toral Quiroga (2):

- i965/sbe: fix number of inputs for active components
- i965/vec4: use a temp register to compute offsets for pull loads

James Legg (1):

- radv: Really use correct HTILE expanded words.

Jason Ekstrand (3):

- intel/isl: Add an isl_color_value_is_zero helper
- vulkan/wsi/x11: Set OUT_OF_DATE if wait_for_special_event fails
- intel/fs: Set up sampler message headers in the visitor on gen7+

Jonathan Gray (1):

- configure.ac: pthread-stubs not present on OpenBSD

Jordan Justen (3):

- i965: Create new program cache bo when clearing the program cache
- program: Don't reset SamplersValidated when restoring from shader cache
- intel/vulkan: Hard code CS scratch_ids_per_subslice for Cherryview

Juan A. Suarez Romero (14):

- cherry-ignore: Explicit 18.0 only nominations
- cherry-ignore: r600/compute: only mark buffer/image state dirty for fragment shaders
- cherry-ignore: anv: Move setting current_pipeline to cmd_state_init
- cherry-ignore: anv: Be more careful about fast-clear colors

- cherry-ignore: Add patches that has a specific version for 17.3
- cherry-ignore: r600: Take ALU_EXTENDED into account when evaluating jump offsets
- cherry-ignore: intel/compiler: Memory fence commit must always be enabled for gen10+
- cherry-ignore: i965: Avoid problems from referencing orphaned BOs after growing.
- cherry-ignore: include all Meson related fixes
- cherry-ignore: ac/shader: fix vertex input with components.
- cherry-ignore: i965: Use absolute addressing for constant buffer 0 on Kernel 4.16+.
- cherry-ignore: anv/image: Separate modifiers from legacy scanout
- cherry-ignore: glsl: Fix memory leak with known glsl_type instances
- Update version to 17.3.7

Karol Herbst (1):

- nvir/nvc0: fix legalizing of ld unlock c0[0x10000]

Kenneth Graunke (1):

- i965: Emit CS stall before MEDIA_VFE_STATE.

Lionel Landwerlin (1):

- i965: perf: ensure reading config IDs from sysfs isn't interrupted

Marek Olšák (2):

- radeonsi: align command buffer starting address to fix some Raven hangs
- configure.ac: blacklist libdrm 2.4.90

Michal Navratil (1):

- winsys/amdgpu: allow non page-aligned size bo creation from pointer

Samuel Iglesias Gonsálvez (1):

- glsl/linker: fix bug when checking precision qualifier

Samuel Pitoiset (2):

- ac/nir: use ordered float comparisons except for not equal
- Revert "mesa: do not trigger _NEW_TEXTURE_STATE in glActiveTexture()"

Stephan Gerhold (1):

- util/build-id: Fix address comparison for binaries with LOAD vaddr > 0

Thomas Hellstrom (2):

- svga: Fix a leftover debug hack
- loader_dri3/glx/egl: Reinstate the loader_dri3_vtable get_dri_screen callback

Tim Rowley (1):

- swr/rast: fix MemoryBuffer build break for llvm-6

Timothy Arceri (1):

- nir: fix interger divide by zero crash during constant folding

Tobias Droste (1):

- gallivm: Use new LLVM fast-math-flags API

Vadym Shovkoplias (1):

- mesa: add glsl version query (v4)

Vinson Lee (1):

- swr/rast: Fix macOS macro.

4.91 Mesa 17.3.6 Release Notes / February 27, 2018

Mesa 17.3.6 is a bug fix release which fixes bugs found since the 17.3.5 release.

Mesa 17.3.6 implements the OpenGL 4.5 API, but the version reported by glGetString(GL_VERSION) or glGetIntegerv(GL_MAJOR_VERSION) / glGetIntegerv(GL_MINOR_VERSION) depends on the particular driver being used. Some drivers don't support all the features required in OpenGL 4.5. OpenGL 4.5 is **only** available if requested at context creation because compatibility contexts are not supported.

4.91.1 SHA256 checksums

```
d5e10ea3f0d11b06d2b0b235bba372a04278c39bc0e712090bda1f61842db188  mesa-17.3.6.tar.gz
e5915680d44ac9d05defdec529db7459ac9edd441c9845266eff2e2d3e57fbf8  mesa-17.3.6.tar.xz
```

4.91.2 New features

None

4.91.3 Bug fixes

- Bug 104383 - [KBL] Intel GPU hang with firefox
- Bug 104411 - [CCS] lemonbar-xft GPU hang
- Bug 104546 - Crash happens when running compute pipeline after calling glxMakeCurrent two times

4.91.4 Changes

Emil Velikov (2):

- docs: add sha256 checksums for 17.3.5
- Update version to 17.3.6

Jason Ekstrand (4):

- i965/draw: Do resolves properly for textures used by TXF
- i965: Replace draw_aux_buffer_disabled with draw_aux_usage
- i965/draw: Set NEW_AUX_STATE when draw aux changes
- i965: Stop disabling aux during texture preparation

Kenneth Graunke (1):

- i965: Don't disable CCS for RT dependencies when dispatching compute.

Topi Pohjolainen (1):

- i965: Don't try to disable render aux buffers for compute

4.92 Mesa 17.3.5 Release Notes / February 19, 2018

Mesa 17.3.5 is a bug fix release which fixes bugs found since the 17.3.4 release.

Mesa 17.3.5 implements the OpenGL 4.5 API, but the version reported by glGetString(GL_VERSION) or glGetIntegerv(GL_MAJOR_VERSION) / glGetIntegerv(GL_MINOR_VERSION) depends on the particular driver being used. Some drivers don't support all the features required in OpenGL 4.5. OpenGL 4.5 is **only** available if requested at context creation because compatibility contexts are not supported.

4.92.1 SHA256 checksums

```
bc1ee20366aae2affc37c89228f871f438136f70252005e9f842169bde976788  mesa-17.3.5.tar.gz
eb9228fc8aaa71e0205c1481c5b157752ebaec9b646b030d27478e25a6d7936a  mesa-17.3.5.tar.xz
```

4.92.2 New features

None

4.92.3 Bug fixes

None

4.92.4 Changes

Emil Velikov (2):

- docs: add sha256 checksums for 17.3.4
- Update version to 17.3.5

James Legg (1):

- ac/nir: Fix conflict resolution typo in handle_vs_input_decl

4.93 Mesa 17.3.4 Release Notes / January 15, 2018

Mesa 17.3.4 is a bug fix release which fixes bugs found since the 17.3.3 release.

Mesa 17.3.4 implements the OpenGL 4.5 API, but the version reported by glGetString(GL_VERSION) or glGetIntegerv(GL_MAJOR_VERSION) / glGetIntegerv(GL_MINOR_VERSION) depends on the particular driver being used. Some drivers don't support all the features required in OpenGL 4.5. OpenGL 4.5 is **only** available if requested at context creation because compatibility contexts are not supported.

4.93.1 SHA256 checksums

```
2d3a4c3cbc995b3e192361dce710d8c749e046e7575aa1b7d8fc9e6b4df28f84  mesa-17.3.4.tar.gz
71f995e233bc5df1a0dd46c980d1720106e7f82f02d61c1ca50854b5e02590d0  mesa-17.3.4.tar.xz
```

4.93.2 New features

None

4.93.3 Bug fixes

- Bug 90311 - Fail to build libglx with clang at linking stage
- Bug 101442 - Piglit shaders@ssa@fs-if-def-else-break fails with sb but passes with R600_DEBUG=nosb
- Bug 102435 - [skl,kbl] [drm] GPU HANG: ecode 9:0:0x86df7cf9, in csgo_linux64 [4947], reason: Hang on rcs, action: reset
- Bug 103006 - [OpenGL CTS] [HSW] KHR-GL45.vertex_attrib_binding.basic-inputL-case1
- Bug 103626 - [SNB] ES3-CTS.functional.shaders.precision
- Bug 104163 - [GEN9+] 2-3% perf drop in GfxBench Manhattan 3.1 from "i965: Disable regular fast-clears (CCS_D) on gen9+"
- Bug 104383 - [KBL] Intel GPU hang with firefox
- Bug 104411 - [CCS] lemonbar-xft GPU hang
- Bug 104487 - [KBL] portal2_linux GPU hang
- Bug 104711 - [skl CCS] Oxenfree (unity engine game) hangs GPU
- Bug 104741 - Graphic corruption for Android apps Telegram and KineMaster
- Bug 104745 - HEVC VDPAU decoding broken on RX 460 with UVD Firmware v1.130
- Bug 104818 - mesa fails to build on ia64

4.93.4 Changes

Andres Gomez (1):

- i965: perform 2 uploads with dual slot *64*PASSTHRU formats on gen<8

Bas Nieuwenhuizen (10):

- radv: Fix ordering issue in meta memory allocation failure path.
- radv: Fix memory allocation failure path in compute resolve init.
- radv: Fix freeing meta state if the device pipeline cache fails to allocate.
- radv: Fix fragment resolve init memory allocation failure paths.
- radv: Fix bufimage failure deallocation.
- radv: Init variant entry with memset.
- radv: Don't allow 3d or 1d depth/stencil textures.
- ac/nir: Use instance_rate_inputs per attribute, not per variable.

- ac/nir: Use correct 32-bit component writemask for 64-bit SSBO stores.
- ac/nir: Fix vector extraction if source vector has >4 elements.

Boyuan Zhang (2):

- radeon/vcn: add and manage render picture list
- radeon/uvd: add and manage render picture list

Chuck Atkins (1):

- configure.ac: add missing llvm dependencies to .pc files

Dave Airlie (10):

- r600/sb: fix a bug emitting ar load from a constant.
- ac/nir: account for view index in the user sgpr allocation.
- radv: add fs_key meta format support to resolve passes.
- radv: don't use hw resolve for integer image formats
- radv: don't use hw resolves for r16g16 norm formats.
- radv: move spi_baryc_cntl to pipeline
- r600/sb: insert the else clause when we might depart from a loop
- radv: don't enable tc compat for d32s8 + 4/8 samples (v1.1)
- radv/gfx9: fix block compression texture views. (v2)
- virgl: also remove dimension on indirect.

Eleni Maria Stea (1):

- mesa: Fix function pointers initialization in status tracker

Emil Velikov (18):

- cherry-ignore: i965: Accept CONTEXT_ATTRIB_PRIORITY for brwCreateContext
- cherry-ignore: swr: refactor swr_create_screen to allow for proper cleanup on error
- cherry-ignore: anv: add explicit 18.0 only nominations
- cherry-ignore: radv: fix sample_mask_in loading. (v3.1)
- cherry-ignore: meson: multiple fixes
- cherry-ignore: swr/rast: support llvm 3.9 type declarations
- Revert "cherry-ignore: intel/fs: Use the original destination region for int MUL lowering"
- cherry-ignore: ac/nir: set amdgpu.uniform and invariant.load for UBOs
- cherry-ignore: add gen10 fixes
- cherry-ignore: add r600/amdgpu 18.0 nominations
- cherry-ignore: add i965 shader cache fixes
- cherry-ignore: nir: mark unused space in packed_tex_data
- radv: Stop advertising VK_KHX_multiview
- cherry-ignore: radv: Don't expose VK_KHX_multiview on android.
- configure.ac: correct driglx-direct help text

- cherry-ignore: add meson fix
- cherry-ignore: add a few more meson fixes
- Update version to 17.3.4

Eric Engestrom (1):

- radeon: remove left over dead code

Gert Wollny (1):

- r600/shader: Initialize max_driver_temp_used correctly for the first time

Grazvydas Ignotas (2):

- st/va: release held locks in error paths
- st/vdpau: release held lock in error path

Igor Gnatenko (1):

- link mesautil with pthreads

Indrajit Das (4):

- st/omx_bellagio: Update default intra matrix per MPEG2 spec
- radeon/uvd: update quantiser matrices only when requested
- radeon/vcn: update quantiser matrices only when requested
- st/va: clear pointers for mpeg2 quantiser matrices

Jason Ekstrand (19):

- i965: Call brw_cache_flush_for_render in predraw_resolve_framebuffer
- i965: Add more precise cache tracking helpers
- i965/blorp: Add more destination flushing
- i965: Track the depth and render caches separately
- i965: Track format and aux usage in the render cache
- Re-enable regular fast-clears (CCS_D) on gen9+
- i965/miptree: Refactor CCS_E and CCS_D cases in render_aux_usage
- i965/miptree: Add an explicit tiling parameter to create_for_bo
- i965/miptree: Use the tiling from the modifier instead of the BO
- i965/bufmgr: Add a create_from_prime_tiled function
- i965: Set tiling on BOs imported with modifiers
- i965/miptree: Take an aux_usage in prepare/finish_render
- i965/miptree: Add an aux_disabled parameter to render_aux_usage
- i965/surface_state: Drop brw_aux_surface_disabled
- intel/fs: Use the original destination region for int MUL lowering
- anv/pipeline: Don't look at blend state unless we have an attachment
- anv/cmd_buffer: Re-emit the pipeline at every subpass
- anv: Stop advertising VK_KHX_multiview

- i965: Call prepare_external after implicit window-system MSAA resolves

Jon Turney (3):

- configure: Default to gbm=no on osx
- glx/apple: include util/debug.h for env_var_as_boolean prototype
- glx/apple: locate dispatch table functions to wrap by name

José Fonseca (1):

- svga: Prevent use after free.

Juan A. Suarez Romero (1):

- docs: add sha256 checksums for 17.3.3

Kenneth Graunke (2):

- i965: Bind null render targets for shadow sampling + color.
- i965: Bump official kernel requirement to Linux v3.9.

Lucas Stach (2):

- etnaviv: dirty TS state when framebuffer has changed
- renderonly: fix dumb BO allocation for non 32bpp formats

Marek Olšák (1):

- radeonsi: don't ignore pitch for imported textures

Matthew Nicholls (2):

- radv: restore previous stencil reference after depth-stencil clear
- radv: remove predication on cache flushes

Maxin B. John (1):

- anv_icd.py: improve reproducible builds

Michel Dänzer (1):

- winsys/radeon: Compute is_displayable in surf_drm_to_winsys

Roland Scheidegger (1):

- r600: don't do stack workarounds for hemlock

Samuel Pitoiset (1):

- radv: create pipeline layout objects for all meta operations

Samuel Thibault (1):

- glx: fix non-dri build

Timothy Arceri (2):

- ac: fix buffer overflow bug in 64bit SSBO loads
- ac: fix visit_ssa_undef() for doubles

4.94 Mesa 17.3.3 Release Notes / January 18, 2018

Mesa 17.3.3 is a bug fix release which fixes bugs found since the 17.3.2 release.

Mesa 17.3.3 implements the OpenGL 4.5 API, but the version reported by glGetString(GL_VERSION) or glGetIntegerv(GL_MAJOR_VERSION) / glGetIntegerv(GL_MINOR_VERSION) depends on the particular driver being used. Some drivers don't support all the features required in OpenGL 4.5. OpenGL 4.5 is **only** available if requested at context creation because compatibility contexts are not supported.

4.94.1 SHA256 checksums

```
c733d37a161501cd81dc9b309ccb613753b98eafc6d35e0847548a6642749772  mesa-17.3.3.tar.gz
41bac5de0ef6adc1f41a1ec0f80c19e361298ce02fa81b5f9ba4fdca33a9379b  mesa-17.3.3.tar.xz
```

4.94.2 New features

None

4.94.3 Bug fixes

- Bug 104214 - Dota crashes when switching from game to desktop
- Bug 104492 - Compute Shader: Wrong alignment when assigning struct value to structured SSBO
- Bug 104551 - Check if Mako templates for Python are installed

4.94.4 Changes

Alex Smith (3):

- anv: Add missing unlock in anv_scratch_pool_alloc
- anv: Take write mask into account in has_color_buffer_write_enabled
- anv: Make sure state on primary is correct after CmdExecuteCommands

Andres Gomez (1):

- anv: Import mako templates only during execution of anv_extensions

Bas Nieuwenhuizen (11):

- radv: Invert condition for all samples identical during resolve.
- radv: Flush caches before subpass resolve.
- radv: Fix fragment resolve destination offset.
- radv: Use correct framebuffer size for partial FS resolves.
- radv: Always use fragment resolve if dest uses DCC.
- Revert "radv/gfx9: fix block compression texture views."
- radv: Use correct HTILE expanded words.
- radv: Allow writing 0 scissors.

- ac/nir: Handle loading data from compact arrays.
- radv: Invalidate L1 for VK_ACCESS_VERTEX_ATTRIBUTE_READ_BIT.
- ac/nir: Sanitize location_frac for local variables.

Dave Airlie (8):

- radv: fix events on compute queues.
- radv: fix pipeline statistics end query on compute queue
- radv/gfx9: fix 3d image to image transfers on compute queues.
- radv/gfx9: fix 3d image clears on compute queues
- radv/gfx9: fix buffer to image for 3d images on compute queues
- radv/gfx9: fix block compression texture views.
- radv/gfx9: use a bigger hammer to flush cb/db caches.
- radv/gfx9: use correct swizzle parameter to work out border swizzle.

Emil Velikov (1):

- docs: add sha256 checksums for 17.3.2

Florian Will (1):

- glsl: Respect std430 layout in lower_buffer_access

Juan A. Suarez Romero (6):

- cherry-ignore: intel/fs: Use the original destination region for int MUL lowering
- cherry-ignore: i965/fs: Use UW types when using V immediates
- cherry-ignore: main: Clear shader program data whenever ProgramBinary is called
- cherry-ignore: egl: pass the dri2_dpy to the $plat_teardown functions
- cherry-ignore: vulkan/wsi: free cmd pools
- Update version to 17.3.3

Józef Kucia (1):

- radeonsi: fix alpha-to-coverage if color writes are disabled

Kenneth Graunke (2):

- i965: Require space for MI_BATCHBUFFER_END.
- i965: Torch public intel_batchbuffer_emit_dword/float helpers.

Lucas Stach (1):

- etnaviv: disable in-place resolve for non-supertiled surfaces

Samuel Iglesias Gonsálvez (1):

- anv: VkDescriptorSetLayoutBinding can have descriptorCount == 0

Thomas Hellstrom (1):

- loader/dri3: Avoid freeing renderbuffers in use

Tim Rowley (1):

- swr/rast: fix invalid sign masks in avx512 simdlib code

4.95 Mesa 17.3.2 Release Notes / January 8, 2018

Mesa 17.3.2 is a bug fix release which fixes bugs found since the 17.3.1 release.

Mesa 17.3.2 implements the OpenGL 4.5 API, but the version reported by glGetString(GL_VERSION) or glGetIntegerv(GL_MAJOR_VERSION) / glGetIntegerv(GL_MINOR_VERSION) depends on the particular driver being used. Some drivers don't support all the features required in OpenGL 4.5. OpenGL 4.5 is **only** available if requested at context creation because compatibility contexts are not supported.

4.95.1 SHA256 checksums

```
f997e80f14c385f9a2ba827c2b74aebf1b7426712ca4a81c631ef9f78e437bf4  mesa-17.3.2.tar.gz
e2844a13f2d6f8f24bee65804a51c42d8dc6ae9c36cff7ee61d0940e796d64c6  mesa-17.3.2.tar.xz
```

4.95.2 New features

None

4.95.3 Bug fixes

- Bug 97852 - Unreal Engine corrupted preview viewport
- Bug 103801 - [i965] >Observer_ issue
- Bug 104288 - Steamroll needs allow_glsl_cross_stage_interpolation_mismatch=true

4.95.4 Changes

Bas Nieuwenhuizen (1):

- radv: Fix DCC compatible formats.

Brendan King (1):

- egl: link libEGL against the dynamic version of libglapi

Dave Airlie (6):

- radv/gfx9: add support for 3d images to blit 2d paths
- radv: handle depth/stencil image copy with layouts better. (v3.1)
- radv/meta: fix blit paths for depth/stencil (v2.1)
- radv: fix issue with multisample positions and interp_var_at_sample.
- radv/gfx9: add 3d sampler image->buffer copy shader. (v3)
- radv: don't do format replacement on tc compat htile surfaces.

Emil Velikov (2):

- docs: add sha256 checksums for 17.3.1
- Update version to 17.3.2

Eric Engestrom (1):

- egl: let each platform decided how to handle LIBGL_ALWAYS_SOFTWARE

Rob Herring (1):

- egl/android: Fix build break with dri2_initialize_android _EGLDisplay parameter

Samuel Pitoiset (2):

- radv/gfx9: fix primitive topology when adjacency is used
- radv: use a faster version for nir_op_pack_half_2x16

Tapani Pälli (2):

- mesa: add AllowGLSLCrossStageInterpolationMismatch workaround
- drirc: set allow_glsl_cross_stage_interpolation_mismatch for more games

4.96 Mesa 17.2.8 Release Notes / December 22, 2017

Mesa 17.2.8 is a bug fix release which fixes bugs found since the 17.2.7 release.

Mesa 17.2.8 implements the OpenGL 4.5 API, but the version reported by glGetString(GL_VERSION) or glGetIntegerv(GL_MAJOR_VERSION) / glGetIntegerv(GL_MINOR_VERSION) depends on the particular driver being used. Some drivers don't support all the features required in OpenGL 4.5. OpenGL 4.5 is **only** available if requested at context creation because compatibility contexts are not supported.

4.96.1 SHA256 checksums

```
c715c3a3d6fe26a69c096f573ec416e038a548f0405e3befedd5136517527a84  mesa-17.2.8.tar.gz
6e940345cceaadfd805d701ed2b956589fa77fe8c39991da30ed51ea6b9d095f  mesa-17.2.8.tar.xz
```

4.96.2 New features

None

4.96.3 Bug fixes

- Bug 102710 - vkCmdBlitImage with arrayLayers > 1 fails
- Bug 103007 - [OpenGL CTS] [HSW] KHR-GL45.gpu_shader_fp64.fp64.max_uniform_components fails
- Bug 103544 - Graphical glitches r600 in game this war of mine linux native
- Bug 103579 - Vertex shader causes compiler to crash in SPIRV-to-NIR

4.96.4 Changes

Andres Gomez (6):

- cherry-ignore: swr: Fix KNOB_MAX_WORKER_THREADS thread creation override.
- cherry-ignore: added 17.3 nominations.
- cherry-ignore: radv: port merge tess info from anv

- cherry-ignore: main: Clear shader program data whenever ProgramBinary is called
- cherry-ignore: r600: set DX10_CLAMP for compute shader too
- Update version to 17.2.8

Bas Nieuwenhuizen (2):

- spirv: Fix loading an entire block at once.
- radv: Fix multi-layer blits.

Brian Paul (2):

- xlib: call _mesa_warning() instead of fprintf()
- gallium/aux: include nr_samples in util_resource_size() computation

Emil Velikov (1):

- docs: add sha256 checksums for 17.2.7

Iago Toral Quiroga (1):

- i965/vec4: use a temp register to compute offsets for pull loads

Leo Liu (1):

- radeon/vce: move destroy command before feedback command

Matt Turner (2):

- util: Assume little endian in the absence of platform-specific handling
- util: Add a SHA1 unit test program

Roland Scheidegger (2):

- r600: use min_dx10/max_dx10 instead of min/max
- r600: use DX10_CLAMP bit in shader setup

4.97 Mesa 17.3.1 Release Notes / December 21, 2017

Mesa 17.3.1 is a bug fix release which fixes bugs found since the 17.3.0 release.

Mesa 17.3.1 implements the OpenGL 4.5 API, but the version reported by glGetString(GL_VERSION) or glGetIntegerv(GL_MAJOR_VERSION) / glGetIntegerv(GL_MINOR_VERSION) depends on the particular driver being used. Some drivers don't support all the features required in OpenGL 4.5. OpenGL 4.5 is **only** available if requested at context creation because compatibility contexts are not supported.

4.97.1 SHA256 checksums

```
b0bb0419dbe3043ed4682a28eaf95721f427ca3f23a3c2a7dc77dbe8a3b6384d  mesa-17.3.1.tar.gz
9ae607e0998a586fb2c866cfc8e45e6f52d1c56cb1b41288253ea83eada824c1  mesa-17.3.1.tar.xz
```

4.97.2 New features

None

4.97.3 Bug fixes

- Bug 94739 - Mesa 11.1.2 implementation error: bad format MESA_FORMAT_Z_FLOAT32 in _mesa_unpack_uint_24_8_depth_stencil_row
- Bug 102710 - vkCmdBlitImage with arrayLayers > 1 fails
- Bug 103579 - Vertex shader causes compiler to crash in SPIRV-to-NIR
- Bug 103966 - Mesa 17.2.5 implementation error: bad format MESA_FORMAT_Z_FLOAT32 in _mesa_unpack_uint_24_8_depth_stencil_row
- Bug 104119 - radv: OpBitFieldInsert produces 0 with a loop counter for Insert
- Bug 104143 - r600/sb: clobbers gl_Position -> gl_FragCoord

4.97.4 Changes

Alex Smith (1):

- radv: Add LLVM version to the device name string

Bas Nieuwenhuizen (3):

- spirv: Fix loading an entire block at once.
- radv: Don't advertise VK_EXT_debug_report.
- radv: Fix multi-layer blits.

Ben Crocker (1):

- docs/llvmpipe: document ppc64le as alternative architecture to x86.

Brian Paul (2):

- xlib: call _mesa_warning() instead of fprintf()
- gallium/aux: include nr_samples in util_resource_size() computation

Bruce Cherniak (1):

- swr: Fix KNOB_MAX_WORKER_THREADS thread creation override.

Dave Airlie (1):

- radv: port merge tess info from anv

Emil Velikov (5):

- docs: add sha256 checksums for 17.3.0
- util: scons: wire up the sha1 test
- cherry-ignore: meson: fix strtof locale support check
- cherry-ignore: util: add mesa-sha1 test to meson
- Update version to 17.3.1

Eric Anholt (1):

- broadcom/vc4: Fix handling of GFXH-515 workaround with a start vertex count.

Eric Engestrom (1):

- compiler: use NDEBUG to guard asserts

Fabian Bieler (2):

- glsl: Match order of gl_LightSourceParameters elements.
- glsl: Fix gl_NormalScale.

Gert Wollny (1):

- r600/sb: do not convert if-blocks that contain indirect array access

James Legg (1):

- nir/opcodes: Fix constant-folding of bitfield_insert

Jason Ekstrand (1):

- i965: Switch over to fully external-or-not MOCS scheme

Juan A. Suarez Romero (1):

- travis: disable Meson build

Kenneth Graunke (2):

- meta: Initialize depth/clear values on declaration.
- meta: Fix ClearTexture with GL_DEPTH_COMPONENT.

Leo Liu (1):

- radeon/vce: move destroy command before feedback command

Marek Olšák (4):

- radeonsi: flush the context after resource_copy_region for buffer exports
- radeonsi: allow DMABUF exports for local buffers
- winsys/amdgpu: disable local BOs again due to worse performance
- radeonsi: don't call force_dcc_off for buffers

Matt Turner (2):

- util: Assume little endian in the absence of platform-specific handling
- util: Add a SHA1 unit test program

Nicolai Hähnle (1):

- radeonsi: fix the R600_RESOURCE_FLAG_UNMAPPABLE check

Pierre Moreau (1):

- nvc0/ir: Properly lower 64-bit shifts when the shift value is >32

Timothy Arceri (1):

- glsl: get correct member type when processing xfb ifc arrays

Vadym Shovkoplias (2):

- glx/dri3: Remove unused deviceName variable
- util/disk_cache: Remove unneeded free() on always null string

4.98 Mesa 17.2.7 Release Notes / December 14, 2017

Mesa 17.2.7 is a bug fix release which fixes bugs found since the 17.2.6 release.

Mesa 17.2.7 implements the OpenGL 4.5 API, but the version reported by glGetString(GL_VERSION) or glGetIntegerv(GL_MAJOR_VERSION) / glGetIntegerv(GL_MINOR_VERSION) depends on the particular driver being used. Some drivers don't support all the features required in OpenGL 4.5. OpenGL 4.5 is **only** available if requested at context creation because compatibility contexts are not supported.

4.98.1 SHA256 checksums

```
e8d837a1cd55014e636e9caf6c75cfbe1b3e4be9ab3fa125f5ef38398aa12e97  mesa-17.2.7.tar.gz
50cfdea8df55045797b4d0409591c04c784d9551c4da09b8178874dbe5a37a68  mesa-17.2.7.tar.xz
```

4.98.2 New features

None

4.98.3 Bug fixes

- Bug 94739 - Mesa 11.1.2 implementation error: bad format MESA_FORMAT_Z_FLOAT32 in _mesa_unpack_uint_24_8_depth_stencil_row
- Bug 101378 - interpolateAtSample check for input parameter is too strict
- Bug 102006 - gstreamer vaapih264enc segfault
- Bug 102435 - [skl,kbl] [drm] GPU HANG: ecode 9:0:0x86df7cf9, in csgo_linux64 [4947], reason: Hang on rcs, action: reset
- Bug 102552 - Null dereference due to not checking return value of util_format_description
- Bug 102677 - [OpenGL CTS] KHR-GL45.CommonBugs.CommonBug_PerVertexValidation fails
- Bug 103098 - [OpenGL CTS] KHR-GL45.enhanced_layouts.varying_structure_locations fails
- Bug 103227 - [G965 G45 ILK] ES2-CTS.gtf.GL2ExtensionTests.texture_float.texture_float regression
- Bug 103393 - glDispatchComputeGroupSizeARB : gl_GlobalInvocationID.x != gl_WorkGroupID.x * gl_LocalGroupSizeARB.x + gl_LocalInvocationID.x
- Bug 103412 - gallium/wgl: Another fix to context creation without prior SetPixelFormat()
- Bug 103616 - Increased difference from reference image in shaders
- Bug 103626 - [SNB] ES3-CTS.functional.shaders.precision
- Bug 103732 - [swr] often gets stuck in piglit's glx-multi-context-single-window test
- Bug 103909 - anv_allocator.c:113:1: error: static declaration of 'memfd_create' follows non-static declaration
- Bug 103966 - Mesa 17.2.5 implementation error: bad format MESA_FORMAT_Z_FLOAT32 in _mesa_unpack_uint_24_8_depth_stencil_row
- Bug 104119 - radv: OpBitFieldInsert produces 0 with a loop counter for Insert
- Bug 104143 - r600/sb: clobbers gl_Position -> gl_FragCoord

4.98.4 Changes

Alex Smith (1):

- radv: Add LLVM version to the device name string

Andres Gomez (2):

- docs: add sha256 checksums for 17.2.6
- docs: remove bug 103626 from fix list as per 17.2.6

Ben Crocker (2):

- docs/llvmpipe.html: Minor edits
- docs/llvmpipe: document ppc64le as alternative architecture to x86.

Dave Airlie (1):

- r600/sb: handle jump after target to end of program. (v2)

Denis Pauk (1):

- gallium/{r600, radeonsi}: Fix segfault with color format (v2)

Eduardo Lima Mitev (3):

- glsl_parser_extra: Add utility to copy symbols between symbol tables
- glsl: Use the utility function to copy symbols between symbol tables
- glsl/linker: Check that re-declared, inter-shader built-in blocks match

Emil Velikov (3):

- gl_table.py: add extern C guard for the generated glapitable.h
- cherry-ignore: radeonsi: allow DMABUF exports for local buffers
- Update version to 17.2.7

Eric Anholt (1):

- broadcom/vc4: Fix handling of GFXH-515 workaround with a start vertex count.

Eric Engestrom (1):

- compiler: use NDEBUG to guard asserts

Fabian Bieler (2):

- glsl: Match order of gl_LightSourceParameters elements.
- glsl: Fix gl_NormalScale.

Frank Richter (1):

- gallium/wgl: fix default pixel format issue

George Kyriazis (1):

- swr: Handle resource across context changes

Gert Wollny (2):

- r600: Emit EOP for more CF instruction types
- r600/sb: do not convert if-blocks that contain indirect array access

Ilia Mirkin (1):

- glsl: fix derived cs variables

James Legg (1):

- nir/opcodes: Fix constant-folding of bitfield_insert

Jason Ekstrand (1):

- i965: Disable regular fast-clears (CCS_D) on gen9+

Juan A. Suarez Romero (1):

- glsl: add varying resources for arrays of complex types

Julien Isorce (1):

- st/va: change frame_idx from array to hash table

Kai Wasserbäch (1):

- docs: Point to apt.llvm.org for development snapshot packages

Kenneth Graunke (3):

- meta: Initialize depth/clear values on declaration.
- meta: Fix ClearTexture with GL_DEPTH_COMPONENT.
- i965: Fix Smooth Point Enables.

Marek Olšák (3):

- radeonsi: fix layered DCC fast clear
- radeonsi/gfx9: fix importing shared textures with DCC
- radeonsi: flush the context after resource_copy_region for buffer exports

Matt Turner (4):

- i965/fs: Handle negating immediates on MADs when propagating saturates
- util: Fix SHA1 implementation on big endian
- util: Fix disk_cache index calculation on big endian
- i965/fs: Unpack count argument to 64-bit shift ops on Atom

Nicolai Hähnle (3):

- radeonsi: fix the R600_RESOURCE_FLAG_UNMAPPABLE check
- glsl: allow any l-value of an input variable as interpolant in interpolateAt*
- glsl: fix interpolateAtXxx(some_vec[idx], . . .) with dynamic idx

Pierre Moreau (1):

- nvc0/ir: Properly lower 64-bit shifts when the shift value is >32

Tapani Pälli (1):

- mesa/gles: adjust internal format in glTexSubImage2D error checks

Timothy Arceri (1):

- glsl: get correct member type when processing xfb ifc arrays

Vadym Shovkoplias (2):

- intel/blorp: Fix possible NULL pointer dereferencing

- glx/dri3: Remove unused deviceName variable

Vinson Lee (1):

- anv: Check if memfd_create is already defined.

4.99 Mesa 17.3.0 Release Notes / December 8. 2017

Mesa 17.3.0 is a new development release. People who are concerned with stability and reliability should stick with a previous release or wait for Mesa 17.3.1.

Mesa 17.3.0 implements the OpenGL 4.5 API, but the version reported by glGetString(GL_VERSION) or glGetIntegerv(GL_MAJOR_VERSION) / glGetIntegerv(GL_MINOR_VERSION) depends on the particular driver being used. Some drivers don't support all the features required in OpenGL 4.5. OpenGL 4.5 is **only** available if requested at context creation because compatibility contexts are not supported.

4.99.1 SHA256 checksums

```
0cb1ffe2b4637d80f08df3bdfeb300352dcffd8ff4f6711278639b084e3f07f9  mesa-17.3.0.tar.gz
29a0a3a6c39990d491a1a58ed5c692e596b3bfc6c01d0b45e0b787116c50c6d9  mesa-17.3.0.tar.xz
```

4.99.2 New features

Note: some of the new features are only available with certain drivers.

- libtxc_dxtn is now integrated into Mesa. GL_EXT_texture_compression_s3tc and GL_ANGLE_texture_compression_dxt are now always enabled on drivers that support them
- GL_ARB_indirect_parameters on i965/gen7+
- GL_ARB_polygon_offset_clamp on i965, nv50, nvc0, r600, radeonsi, llvmpipe, swr
- GL_ARB_transform_feedback_overflow_query on radeonsi
- GL_ARB_texture_filter_anisotropic on i965, nv50, nvc0, r600, radeonsi
- GL_EXT_memory_object on radeonsi
- GL_EXT_memory_object_fd on radeonsi
- EGL_ANDROID_native_fence_sync on radeonsi with a future kernel (possibly 4.15)
- EGL_IMG_context_priority on i965

4.99.3 Bug fixes

- Bug 97532 - Regression: GLB 2.7 & Glmark-2 GLES versions segfault due to linker precision error (259fc505) on dead variable
- Bug 100438 - glsl/ir.cpp:1376: ir_dereference_variable::ir_dereference_variable(ir_variable*): Assertion 'var != NULL' failed.
- Bug 100613 - Regression in Mesa 17 on s390x (zSystems)
- Bug 101334 - AMD SI cards: Some vulkan apps freeze the system
- Bug 101378 - interpolateAtSample check for input parameter is too strict

- Bug 101655 - Explicit sync support for android
- Bug 101691 - gfx corruption on windowed 3d-apps running on dGPU
- Bug 101709 - [llvmpipe] piglit gl-1.0-scissor-offscreen regression
- Bug 101766 - Assertion '!"invalid type"' failed when constant expression involves literal of different type
- Bug 101832 - [PATCH][regression][bisect] Xorg fails to start after f50aa21456d82c8cb6fbaa565835f1acc1720a5d
- Bug 101851 - [regression] libEGL_common.a undefined reference to '__gxx_personality_v0'
- Bug 101867 - Launch options window renders black in Feral Games in current Mesa trunk
- Bug 101876 - SIGSEGV when launching Steam
- Bug 101910 - [BYT] ES31-CTS.functional.copy_image.non_compressed.viewclass_96_bits.rgb32f_rgb32f
- Bug 101925 - playstore/webview crash
- Bug 101941 - Getting different output depending on attribute declaration order
- Bug 101961 - Serious Sam Fusion hangs system completely
- Bug 101981 - Commit ddc32537d6db69198e88ef0dfe19770bf9daa536 breaks rendering in multiple applications
- Bug 101982 - Weston crashes when running an OpenGL program on i965
- Bug 101983 - [G33] ES2-CTS.functional.shaders.struct.uniform.sampler_nested* regression
- Bug 101989 - ES3-CTS.functional.state_query.integers.viewport_getinteger regression
- Bug 102006 - gstreamer vaapih264enc segfault
- Bug 102014 - Mesa git build broken by commit bc7f41e11d325280db12e7b9444501357bc13922
- Bug 102015 - [Regression,bisected]: Segfaults with various programs
- Bug 102024 - FORMAT_FEATURE_SAMPLED_IMAGE_BIT not supported for D16_UNORM and D32_SFLOAT
- Bug 102038 - assertion failure in update_framebuffer_size
- Bug 102050 - commit b4f639d02a causes build breakage on Android 32bit builds
- Bug 102052 - No package 'expat' found
- Bug 102062 - Segfault at eglCreateContext in android-x86
- Bug 102125 - [softpipe] piglit arb_texture_view-targets regression
- Bug 102148 - Crash when running qopenglwidget example on mesa llvmpipe win32
- Bug 102177 - [SKL] ES31-CTS.core.sepshaderobjs.StateInteraction fails sporadically
- Bug 102201 - [regression, SI] GPU crash in Unigine Valley
- Bug 102241 - gallium/wgl: SwapBuffers freezing regularly with swap interval enabled
- Bug 102274 - assertion failure in ir_validate.cpp:240
- Bug 102308 - segfault in glCompressedTextureSubImage3D
- Bug 102358 - WarThunder freezes at start, with activated vsync (vblank_mode=2)
- Bug 102377 - PIPE_*_4BYTE_ALIGNED_ONLY caps crashing
- Bug 102429 - [regression, SI] Performance decrease in Unigine Valley & Heaven

- Bug 102435 - [skl,kbl] [drm] GPU HANG: ecode 9:0:0x86df7cf9, in csgo_linux64 [4947], reason: Hang on rcs, action: reset
- Bug 102454 - glibc 2.26 doesn't provide anymore xlocale.h
- Bug 102461 - [llvmpipe] piglit glean fragprog1 XPD test 1 regression
- Bug 102467 - src/mesa/state_tracker/st_cb_readpixels.c:178]: (warning) Redundant assignment
- Bug 102496 - Frontbuffer rendering corruption on mesa master
- Bug 102502 - [bisected] Kodi crashes since commit 707d2e8b - gallium: fold u_trim_pipe_prim call from st/mesa to drivers
- Bug 102530 - [bisected] Kodi crashes when launching a stream - commit bd2662bf
- Bug 102552 - Null dereference due to not checking return value of util_format_description
- Bug 102565 - u_debug_stack.c:114: undefined reference to '_Ux86_64_getcontext'
- Bug 102573 - fails to build on armel
- Bug 102665 - test_glsl_to_tgsi_lifetime.cpp:53:67: error: '>>' should be '> >' within a nested template argument list
- Bug 102677 - [OpenGL CTS] KHR-GL45.CommonBugs.CommonBug_PerVertexValidation fails
- Bug 102680 - [OpenGL CTS] KHR-GL45.shader_ballot_tests.ShaderBallotBitmasks fails
- Bug 102685 - piglit.spec.glsl-1_50.compiler.vs-redeclares-pervertex-out-before-global-redeclaration
- Bug 102774 - [BDW] [Bisected] Absolute constant buffers break VAAPI in mpv
- Bug 102809 - Rust shadows(?) flash random colours
- Bug 102844 - memory leak with glDeleteProgram for shader program type GL_COMPUTE_SHADER
- Bug 102847 - swr fail to build with llvm-5.0.0
- Bug 102852 - Scons: Support the new Scons 3.0.0
- Bug 102904 - piglit and gl45 cts linker tests regressed
- Bug 102924 - mesa (git version) images too dark
- Bug 102940 - Regression: Vulkan KMS rendering crashes since 17.2
- Bug 102955 - HyperZ related rendering issue in ARK: Survival Evolved
- Bug 102999 - [BISECTED,REGRESSION] Failing Android EGL dEQP with RGBA configs
- Bug 103002 - string_buffer_test.cpp:43: error: ISO C++ forbids initialization of member 'str1'
- Bug 103085 - [ivb byt hsw] piglit.spec.arb_indirect_parameters.tf-count-arrays
- Bug 103098 - [OpenGL CTS] KHR-GL45.enhanced_layouts.varying_structure_locations fails
- Bug 103101 - [SKL][bisected] DiRT Rally GPU hang
- Bug 103115 - [BSW BXT GLK] dEQP-VK.spirv_assembly.instruction.compute.sconvert.int32_to_int64
- Bug 103128 - [softpipe] piglit fs-ldexp regression
- Bug 103142 - R600g+sb: optimizer apparently stuck in an endless loop
- Bug 103214 - GLES CTS functional.state_query.indexed.atomic_counter regression
- Bug 103227 - [G965 G45 ILK] ES2-CTS.gtf.GL2ExtensionTests.texture_float.texture_float regression
- Bug 103247 - Performance regression: car chase, manhattan

- Bug 103253 - blob.h:138:1: error: unknown type name 'ssize_t'
- Bug 103265 - [llvmpipe] piglit depth-tex-compare regression
- Bug 103323 - Possible unintended error message in file pixel.c line 286
- Bug 103388 - Linking libcltgsi.la (llvm/codegen/libclllvm_la-common.lo) fails with "error: no match for 'operator-'" with GCC-7, Mesa from Git and current LLVM revisions
- Bug 103393 - glDispatchComputeGroupSizeARB : gl_GlobalInvocationID.x != gl_WorkGroupID.x * gl_LocalGroupSizeARB.x + gl_LocalInvocationID.x
- Bug 103412 - gallium/wgl: Another fix to context creation without prior SetPixelFormat()
- Bug 103519 - wayland egl apps crash on start with mesa 17.2
- Bug 103529 - [GM45] GPU hang with mpv fullscreen (bisected)
- Bug 103537 - i965: Shadow of Mordor broken since commit 379b24a40d3d34ffdaaeb1b328f50e28ecb01468 on Haswell
- Bug 103544 - Graphical glitches r600 in game this war of mine linux native
- Bug 103616 - Increased difference from reference image in shaders
- Bug 103628 - [BXT, GLK, BSW] KHR-GL46.shader_ballot_tests.ShaderBallotBitmasks
- Bug 103759 - plasma desktop corrupted rendering
- Bug 103787 - [BDW,BSW] gpu hang on spec.arb_pipeline_statistics_query.arb_pipeline_statistics_query-comp
- Bug 103909 - anv_allocator.c:113:1: error: static declaration of 'memfd_create' follows non-static declaration

4.99.4 Changes

4.100 Mesa 17.2.6 Release Notes / November 25, 2017

Mesa 17.2.6 is a bug fix release which fixes bugs found since the 17.2.5 release.

Mesa 17.2.6 implements the OpenGL 4.5 API, but the version reported by glGetString(GL_VERSION) or glGetIntegerv(GL_MAJOR_VERSION) / glGetIntegerv(GL_MINOR_VERSION) depends on the particular driver being used. Some drivers don't support all the features required in OpenGL 4.5. OpenGL 4.5 is **only** available if requested at context creation because compatibility contexts are not supported.

4.100.1 SHA256 checksums

```
a9ed76702ffb14ad674ad48899f5c8c7e3a0f987911878a5dfdc4117dce5b415  mesa-17.2.6.tar.gz
6ad85224620330be26ab68c8fc78381b12b38b610ade2db8716b38faaa8f30de  mesa-17.2.6.tar.xz
```

4.100.2 New features

None

4.100.3 Bug fixes

- Bug 100438 - glsl/ir.cpp:1376: ir_dereference_variable::ir_dereference_variable(ir_variable*): Assertion 'var != NULL' failed.
- Bug 102177 - [SKL] ES31-CTS.core.sepshaderobjs.StateInteraction fails sporadically
- Bug 103115 - [BSW BXT GLK] dEQP-VK.spirv_assembly.instruction.compute.sconvert.int32_to_int64
- Bug 103519 - wayland egl apps crash on start with mesa 17.2
- Bug 103529 - [GM45] GPU hang with mpv fullscreen (bisected)
- Bug 103628 - [BXT, GLK, BSW] KHR-GL46.shader_ballot_tests.ShaderBallotBitmasks
- Bug 103787 - [BDW,BSW] gpu hang on spec.arb_pipeline_statistics_query.arb_pipeline_statistics_query-comp

4.100.4 Changes

Adam Jackson (2):

- glx/drisw: Fix glXMakeCurrent(dpy, None, ctx)
- glx/dri3: Fix passing renderType into glXCreateContext

Alex Smith (2):

- spirv: Use correct type for sampled images
- nir/spirv: tg4 requires a sampler

Andres Gomez (14):

- docs: add sha256 checksums for 17.2.5
- cherry-ignore: intel/fs: Use a pure vertical stride for large register strides
- cherry-ignore: intel/nir: Use the correct indirect lowering masks in link_shaders
- cherry-ignore: intel/fs: Use the original destination region for int MUL lowering
- cherry-ignore: intel/fs: refactors
- cherry-ignore: r600/shader: reserve first register of vertex shader.
- cherry-ignore: anv/cmd_buffer: Advance the address when initializing clear colors
- cherry-ignore: anv/cmd_buffer: Take bo_offset into account in fast clear state addresses
- cherry-ignore: i965: Mark BOs as external when we export their handle
- cherry-ignore: added 17.3 nominations.
- cherry-ignore: glsl: Fix typo fragement -> fragment
- cherry-ignore: egl: pass the dri2_dpy to the $plat_teardown functions
- cherry-ignore: Revert "intel/fs: Use a pure vertical stride for large register strides"
- Update version to 17.2.6

Anuj Phogat (2):

- i965: Program DWord Length in MI_FLUSH_DW
- i965/gen8+: Fix the number of dwords programmed in MI_FLUSH_DW

Bas Nieuwenhuizen (2):

- radv: Free syncobj with multiple imports.
- radv: Free temporary syncobj after waiting on it.

Dave Airlie (1):

- r600: fix isoline tess factor component swapping.

Derek Foreman (1):

- egl/wayland: Add a fallback when fourcc query isn't supported

Dylan Baker (1):

- autotools: Set C++ visibility flags on Intel

Emil Velikov (3):

- targets/opencl: don't hardcode the icd file install to /etc/...
- configure.ac: loosen –enable-glvnd check to honour egl
- configure.ac: require xcb* for the omx/va/... when using x11 platform

George Barrett (1):

- glsl: Catch subscripted calls to undeclared subroutines

Jason Ekstrand (9):

- intel/fs: Use ANY/ALL32 predicates in SIMD32
- intel/fs: Use an explicit D type for vote any/all/eq intrinsics
- intel/fs: Use a pair of 1-wide MOVs instead of SEL for any/all
- intel/eu/reg: Add a subscript() helper
- intel/fs: Fix MOV_INDIRECT for 64-bit values on little-core
- intel/fs: Fix integer multiplication lowering for src/dst hazards
- intel/fs: Mark 64-bit values as being contiguous
- intel/fs: Rework zero-length URB write handling
- i965: Add stencil buffers to cache set regardless of stencil texturing

Kenneth Graunke (5):

- i965: properly initialize brw->cs.base.stage to MESA_SHADER_COMPUTE
- i965: Make L3 configuration atom listen for TCS/TES program updates.
- intel/tools: Fix detection of enabled shader stages.
- i965: Implement another VF cache invalidate workaround on Gen8+.
- i965: Upload invariant state once at the start of the batch on Gen4-5.

Matt Turner (2):

- i965/fs: Fix extract_i8/u8 to a 64-bit destination
- i965/fs: Split all 32->64-bit MOVs on CHV, BXT, GLK

Neil Roberts (1):

- glsl: Transform fb buffers are only active if a variable uses them

Nicolai Hähnle (1):

- ddebug: fix use-after-free of streamout targets

Tim Rowley (2):

- swr/rast: Use gather instruction for i32gather_ps on simd16/avx512
- swr/rast: Faster emulated simd16 permute

Timothy Arceri (3):

- glsl: drop cache_fallback
- glsl: use the correct parent when allocating program data members
- mesa: rework how we free gl_shader_program_data

4.101 Mesa 17.2.5 Release Notes / November 10, 2017

Mesa 17.2.5 is a bug fix release which fixes bugs found since the 17.2.4 release.

Mesa 17.2.5 implements the OpenGL 4.5 API, but the version reported by glGetString(GL_VERSION) or glGetIntegerv(GL_MAJOR_VERSION) / glGetIntegerv(GL_MINOR_VERSION) depends on the particular driver being used. Some drivers don't support all the features required in OpenGL 4.5. OpenGL 4.5 is **only** available if requested at context creation because compatibility contexts are not supported.

4.101.1 SHA256 checksums

```
25b40e72fad64b096c2d8d6fe9579369954debe7970d4ad53e5033c7eec2918b  mesa-17.2.5.tar.gz
7f7f914b7b9ea0b15f2d9d01a4375e311b0e90e55683b8e8a67ce8691eb1070f  mesa-17.2.5.tar.xz
```

4.101.2 New features

None

4.101.3 Bug fixes

- Bug 97532 - Regression: GLB 2.7 & Glmark-2 GLES versions segfault due to linker precision error (259fc505) on dead variable
- Bug 102680 - [OpenGL CTS] KHR-GL45.shader_ballot_tests.ShaderBallotBitmasks fails
- Bug 102809 - Rust shadows(?) flash random colours
- Bug 103142 - R600g+sb: optimizer apparently stuck in an endless loop

4.101.4 Changes

Andres Gomez (8):

- docs: add sha256 checksums for 17.2.4
- cherry-ignore: radv: copy indirect lowering settings from radeonsi
- cherry-ignore: i965: fix blorp stage_prog_data->param leak
- cherry-ignore: etnaviv: don't do resolve-in-place without valid TS

- cherry-ignore: intel/fs: Alloc pull constants off mem_ctx
- cherry-ignore: added 17.3 nominations.
- cherry-ignore: automake: include git_sha1.h.in in release tarball
- Update version to 17.2.5

Bas Nieuwenhuizen (3):

- radv: Don't expose heaps with 0 memory.
- radv: Don't use vgpr indexing for outputs on GFX9.
- radv: Disallow indirect outputs for GS on GFX9 as well.

Dave Airlie (3):

- i915g: make gears run again.
- radv: free attachments on end command buffer.
- radv: add initial copy descriptor support. (v2)

Eric Engestrom (1):

- vc4: fix release build

Gert Wollny (1):

- r600/sb: bail out if prepare_alu_group() doesn't find a proper scheduling

Jason Ekstrand (4):

- spirv: Claim support for the simple memory model
- i965/blorp: Use blorp_to_isl_format for src_isl_format in blit_miptrees
- i965/blorp: Use more temporary isl_format variables
- i965/miptree: Take an isl_format in render_aux_usage

Kenneth Graunke (1):

- mesa: Accept GL_BACK in get_fb0_attachment with ARB_ES3_1_compatibility.

Leo Liu (1):

- radeon/video: add gfx9 offsets when rejoin the video surface

Marek Olšák (2):

- st/dri: don't expose modifiers in EGL if the driver doesn't implement them
- ac/surface/gfx9: don't allow DCC for the smallest mipmap levels

Nanley Chery (1):

- i965: Check CCS_E compatibility for texture view rendering

Neil Roberts (1):

- nir/opt_intrinsics: Fix values for gl_SubGroupG{e,t}MaskARB

Nicolai Hähnle (1):

- amd/common/gfx9: workaround DCC corruption more conservatively

Tapani Pälli (1):

- i965: unref push_const_bo in intelDestroyContext

Timothy Arceri (1):

- radv: copy indirect lowering settings from radeonsi

Tomasz Figa (1):

- glsl: Allow precision mismatch on dead data with GLSL ES 1.00

Topi Pohjolainen (1):

- intel/compiler/gen9: Pixel shader header only workaround

4.102 Mesa 17.2.4 Release Notes / October 30, 2017

Mesa 17.2.4 is a bug fix release which fixes bugs found since the 17.2.3 release.

Mesa 17.2.4 implements the OpenGL 4.5 API, but the version reported by glGetString(GL_VERSION) or glGetIntegerv(GL_MAJOR_VERSION) / glGetIntegerv(GL_MINOR_VERSION) depends on the particular driver being used. Some drivers don't support all the features required in OpenGL 4.5. OpenGL 4.5 is **only** available if requested at context creation because compatibility contexts are not supported.

4.102.1 SHA256 checksums

```
cb266edc5cf7226219ebaf556ca2e03dff282e0324d20afd80423a5754d1272c  mesa-17.2.4.tar.gz
5ba408fecd6e1132e5490eec1a2f04466214e4c65c8b89b331be844768c2e550  mesa-17.2.4.tar.xz
```

4.102.2 New features

None

4.102.3 Bug fixes

- Bug 102774 - [BDW] [Bisected] Absolute constant buffers break VAAPI in mpv
- Bug 103388 - Linking libcltgsi.la (llvm/codegen/libclllvm_la-common.lo) fails with "error: no match for 'operator-'" with GCC-7, Mesa from Git and current LLVM revisions

4.102.4 Changes

Andres Gomez (8):

- cherry-ignore: configure.ac: rework llvm detection and handling
- cherry-ignore: glsl: fix derived cs variables
- cherry-ignore: added 17.3 nominations.
- cherry-ignore: radv: Don't use vgpr indexing for outputs on GFX9.
- cherry-ignore: radv: Disallow indirect outputs for GS on GFX9 as well.
- cherry-ignore: mesa/bufferobj: don't double negate the range
- cherry-ignore: broadcom/vc5: Propagate vc4 aliasing fix to vc5.
- Update version to 17.2.4

Bas Nieuwenhuizen (1):

- ac/nir: Fix nir_texop_lod on GFX for 1D arrays.

Dave Airlie (1):

- radv/image: bump all the offset to uint64_t.

Emil Velikov (1):

- docs: add sha256 checksums for 17.2.3

Henri Verbeet (1):

- vulkan/wsi: Free the event in x11_manage_fifo_queues().

Jan Vesely (1):

- clover: Fix compilation after clang r315871

Jason Ekstrand (4):

- nir/intrinsics: Set the correct num_indices for load_output
- intel/fs: Handle flag read/write aliasing in needs_src_copy
- anv/pipeline: Call nir_lower_system_valaues after brw_preprocess_nir
- intel/eu: Use EXECUTE_1 for JMPI

Kenneth Graunke (1):

- i965: Revert absolute mode for constant buffer pointers.

Marek Olšák (1):

- Revert "mesa: fix texture updates for ATI_fragment_shader"

Matthew Nicholls (1):

- ac/nir: generate correct instruction for atomic min/max on unsigned images

Michel Dänzer (1):

- st/mesa: Initialize textures array in st_framebuffer_validate

Samuel Pitoiset (1):

- radv: add the draw count buffer to the list of buffers

Stefan Schake (1):

- broadcom/vc4: Fix aliasing issue

4.103 Mesa 17.2.3 Release Notes / October 19, 2017

Mesa 17.2.3 is a bug fix release which fixes bugs found since the 17.2.2 release.

Mesa 17.2.3 implements the OpenGL 4.5 API, but the version reported by glGetString(GL_VERSION) or glGetIntegerv(GL_MAJOR_VERSION) / glGetIntegerv(GL_MINOR_VERSION) depends on the particular driver being used. Some drivers don't support all the features required in OpenGL 4.5. OpenGL 4.5 is **only** available if requested at context creation because compatibility contexts are not supported.

4.103.1 SHA256 checksums

```
fb305eecfeec1fd771fdc96fff973c51871f7bd35fd2bd56cacc27b4b8823220  mesa-17.2.3.tar.gz
a0b0ec8f7b24dd044d7ab30a8c7e6d3767521e245f88d4ed5dd93315dc56f837  mesa-17.2.3.tar.xz
```

4.103.2 New features

None

4.103.3 Bug fixes

- Bug 101832 - [PATCH][regression][bisect] Xorg fails to start after f50aa21456d82c8cb6fbaa565835f1acc1720a5d
- Bug 102852 - Scons: Support the new Scons 3.0.0
- Bug 102940 - Regression: Vulkan KMS rendering crashes since 17.2

4.103.4 Changes

Alex Smith (1):

- radv: Add R16G16B16A16_SNORM fast clear support

Bas Nieuwenhuizen (2):

- nir/spirv: Allow loop breaks in a switch body.
- radv: Only set the MTYPE flags on GFX9+.

Ben Crocker (4):

- gallivm: fix typo in debug_printf message
- gallivm: allow additional llc options
- gallivm/ppc64le: adjust VSX code generation control.
- gallivm/ppc64le: allow environmental control of Altivec code generation

Daniel Stone (2):

- egl/wayland: Check queryImage return for wl_buffer
- egl/wayland: Don't use dmabuf with no modifiers

Dave Airlie (2):

- radv: emit fmuladd instead of fma to llvm.
- radv: lower ffma in nir.

Emil Velikov (6):

- cherry-ignore: add "anv: Remove unreachable cases from isl_format_for_size"
- cherry-ignore: add "anv/wsi: Allocate enough memory for the entire image"
- swr/rast: do not crash on NULL strings returned by getenv
- wayland-drm: use a copy of the wayland_drm_callbacks struct

- eglmesaext: add forward declaration for struct wl_buffers
- Update version to 17.2.3

Eric Engestrom (1):

- scons: use python3-compatible print()

Ilia Mirkin (2):

- nv50/ir: fix 64-bit integer shifts
- nv50,nvc0: fix push hint logic in presence of a start offset

Jason Ekstrand (6):

- intel/compiler: Don't cmod propagate into a saturated operation
- intel/compiler: Don't propagate cmod into integer multiplies
- glsl/blob: Return false from ensure_can_read on overrun
- glsl/blob: Return false from grow_to_fit if we've ever failed
- nir/opcodes: Fix constant-folding of ufind_msb
- nir: Get rid of the variable on vote intrinsics

Juan A. Suarez Romero (1):

- docs: add sha256 checksums for 17.2.2

Józef Kucia (3):

- anv: Fix vkCmdFillBuffer()
- spirv: Fix SpvOpAtomicISub
- anv: Do not assert() on VK_ATTACHMENT_UNUSED

Leo Liu (3):

- st/va: use pipe transfer_map to map upload buffer
- st/vdpau: don't re-allocate interlaced buffer with packed YUV format
- st/va: don't re-allocate interlaced buffer with pakced format

Lionel Landwerlin (4):

- intel: compiler: vec4: add missing default 0 lod
- anv/cmd_buffer: fix push descriptors with set > 0
- anv/cmd_buffer: Reset state in cmd_buffer_destroy
- anv: bo_cache: allow importing a BO larger than needed

Marek Olšák (3):

- mesa: fix texture updates for ATI_fragment_shader
- st/mesa: don't use pipe_surface for passing information about EGLImage
- glsl_to_tgsi: fix instruction order for bindless textures

Nicolai Hähnle (14):

- st/glsl_to_tgsi: fix conditional assignments to packed shader outputs
- amd/common: fix build_cube_select

- radeonsi/gfx9: fix geometry shaders without output vertices
- util/queue: fix a race condition in the fence code
- glsl/lower_instruction: handle denorms and overflow in ldexp correctly
- radeonsi: move current_rast_prim to r600_common_context
- radeonsi: don't discard points and lines
- radeonsi: deduce rast_prim correctly for tessellation point mode
- radeonsi: fix maximum advertised point size / line width
- st/mesa: don't clobber glGetInternalformat* buffer for GL_NUM_SAMPLE_COUNTS
- st/glsl_to_tgsi: fix indirect access to 64-bit integer
- st/glsl_to_tgsi: fix a use-after-free in merge_two_dsts
- radeonsi: clamp depth comparison value only for fixed point formats
- radeonsi: clamp border colors for upgraded depth textures

Rob Clark (2):

- freedreno/a5xx: align height to GMEM
- freedreno/a5xx: fix missing restore state

4.104 Mesa 17.2.2 Release Notes / October 2, 2017

Mesa 17.2.2 is a bug fix release which fixes bugs found since the 17.2.1 release.

Mesa 17.2.2 implements the OpenGL 4.5 API, but the version reported by glGetString(GL_VERSION) or glGetIntegerv(GL_MAJOR_VERSION) / glGetIntegerv(GL_MINOR_VERSION) depends on the particular driver being used. Some drivers don't support all the features required in OpenGL 4.5. OpenGL 4.5 is **only** available if requested at context creation because compatibility contexts are not supported.

4.104.1 SHA256 checksums

```
8242256f3243ed3f35184ed7bf0a9070439ccdf477a3bd9cfd2437c0b2f9bc7f  mesa-17.2.2.tar.gz
cf522244d6a5a1ecde3fc00e7c96935253fe22f808f064cab98be6f3faa65782  mesa-17.2.2.tar.xz
```

4.104.2 New features

None

4.104.3 Bug fixes

- Bug 102573 - fails to build on armel
- Bug 102844 - memory leak with glDeleteProgram for shader program type GL_COMPUTE_SHADER
- Bug 102847 - swr fail to build with llvm-5.0.0
- Bug 102904 - piglit and gl45 cts linker tests regressed

4.104.4 Changes

Alexandru-Liviu Prodea (1):

- Scons: Add LLVM 5.0 support

Bas Nieuwenhuizen (1):

- radv: Check for GFX9 for 1D arrays in image_size intrinsic.

Boris Brezillon (1):

- broadcom/vc4: Fix infinite retry in vc4_bo_alloc()

Dave Airlie (3):

- radv/nir: call opt_remove_phis after trivial continues.
- ac/surface: handle S8 on gfx9
- st/glsl->tgsi: fix u64 to bool comparisons.

David Airlie (1):

- radv: add gfx9 scissor workaround

Emil Velikov (2):

- docs: add sha256 checksums for 17.2.1
- automake: enable libunwind in 'make distcheck'

Eric Anholt (4):

- broadcom/vc4: Fix use-after-free for flushing when writing to a texture.
- broadcom/vc4: Fix use-after-free trying to mix a quad and tile clear.
- broadcom/vc4: Fix use-after-free when deleting a program.
- broadcom/vc4: Keep pipe_sampler_view->texture matching the original texture.

Gert Wollny (2):

- travis: force llvm-3.3 for "make Gallium ST Other"
- travis: Add libunwind-dev to gallium/make builds

Grazvydas Ignotas (1):

- configure: check if -latomic is needed for __atomic_*

Ian Romanick (1):

- nv20: Fix GL_CLAMP

Jason Ekstrand (6):

- i965/blorp: Set r8stencil_needs_update when writing stencil
- vulkan/wsi/wayland: Stop printing out the DRM device
- vulkan/wsi/wayland: Refactor wsi_wl_display code
- vulkan/wsi/wayland: Stop caching Wayland displays
- vulkan/wsi/wayland: Copy wl_proxy objects from oldSwapchain if available
- vulkan/wsi/wayland: Return better error messages

Juan A. Suarez Romero (4):

- cherry-ignore: add "radeonsi/gfx9: proper workaround for LS/HS VGPR initialization bug"
- cherry-ignore: add "radv: Check for GFX9 for 1D arrays in image_size intrinsic."
- cherry-ignore: add "radv: copy the number of viewports/scissors at pipeline bind time"
- Update version to 17.2.2

Józef Kucia (1):

- anv: Fix descriptors copying

Kenneth Graunke (2):

- i965/vec4: Actually handle atomic op intrinsics.
- i965/vec4: Fix swizzles on atomic sources.

Leo Liu (1):

- st/va/postproc: use video original size for postprocessing

Lucas Stach (1):

- etnaviv: fix 16bpp clears

Matt Turner (2):

- util: Link libmesautil into u_atomic_test
- util/u_atomic: Add implementation of __sync_val_compare_and_swap_8

Nicolai Hähnle (9):

- radeonsi: workaround for gather4 on integer cube maps
- amd/common: round cube array slice in ac_prepare_cube_coords
- amd/common: add workaround for cube map array layer clamping
- glsl/linker: fix output variable overlap check
- radeonsi: fix array textures layer coordinate
- radeonsi: set MIP_POINT_PRECLAMP to 0
- amd/addrlib: fix missing va_end() after va_copy()
- amd/common: move ac_build_phi from radeonsi
- radeonsi: fix a regression in integer cube map handling

Samuel Iglesias Gonsálvez (1):

- anv: fix viewport transformation for z component

Samuel Pitoiset (1):

- radv: fix saved compute state when doing statistics/occlusion queries

Tapani Pälli (1):

- mesa: free current ComputeProgram state in _mesa_free_context_data

Tim Rowley (1):

- swr/rast: remove llvm fence/atomics from generated files

Tomasz Figa (1):

- egl/dri2: Implement swapInterval fallback in a conformant way

4.105 Mesa 17.1.10 Release Notes / September 25, 2017

Mesa 17.1.10 is a bug fix release which fixes bugs found since the 17.1.9 release.

Mesa 17.1.10 implements the OpenGL 4.5 API, but the version reported by glGetString(GL_VERSION) or glGetIntegerv(GL_MAJOR_VERSION) / glGetIntegerv(GL_MINOR_VERSION) depends on the particular driver being used. Some drivers don't support all the features required in OpenGL 4.5. OpenGL 4.5 is **only** available if requested at context creation because compatibility contexts are not supported.

4.105.1 SHA256 checksums

```
a48ce6b643a728b2b0f926151930525b3670fbff1fb688527fd9051eab9f30a4  mesa-17.1.10.tar.gz
cbc0d681cc4df47d8deb5a36f45b420978128522fd665b2cd4c7096316f11bdb  mesa-17.1.10.tar.xz
```

4.105.2 New features

None

4.105.3 Bug fixes

- Bug 102844 - memory leak with glDeleteProgram for shader program type GL_COMPUTE_SHADER

4.105.4 Changes

Alexandre Demers (1):

- osmesa: link with libunwind if enabled (v2)

Andres Gomez (12):

- docs: add sha256 checksums for 17.1.9
- cherry-ignore: add "st/mesa: skip draw calls with pipe_draw_info::count == 0"
- cherry-ignore: add "radv: use amdgpu_bo_va_op_raw."
- cherry-ignore: add "radv: use simpler indirect packet 3 if possible."
- cherry-ignore: add "radeonsi: don't always apply the PrimID instancing bug workaround on SI"
- cherry-ignore: add "intel/eu/validate: Look up types on demand in execution_type()"
- cherry-ignore: add "radv: gfx9 fixes"
- cherry-ignore: add "radv/gfx9: set mip0-depth correctly for 2d arrays/3d images"
- cherry-ignore: add "radv/gfx9: fix image resource handling."
- cherry-ignore: add "docs/egl: remove reference to EGL_DRIVERS_PATH"
- cherry-ignore: add "radv: Disable multilayer & multilevel DCC."
- cherry-ignore: add "radv: Don't allocate CMASK for linear images."

Dave Airlie (2):

- radv/ac: bump params array for image atomic comp swap

- st/glsl->tgsi: fix u64 to bool comparisons.

Emil Velikov (2):

- egl/x11/dri3: adding missing __DRI_BACKGROUND_CALLABLE extension
- automake: enable libunwind in 'make distcheck'

Eric Anholt (3):

- broadcom/vc4: Fix use-after-free for flushing when writing to a texture.
- broadcom/vc4: Fix use-after-free trying to mix a quad and tile clear.
- broadcom/vc4: Fix use-after-free when deleting a program.

George Kyriazis (1):

- swr: invalidate attachment on transition change

Gert Wollny (2):

- travis: force llvm-3.3 for "make Gallium ST Other"
- travis: Add libunwind-dev to gallium/make builds

Jason Ekstrand (1):

- i965/blorp: Set r8stencil_needs_update when writing stencil

Juan A. Suarez Romero (9):

- cherry-ignore: add "ac/surface: match Z and stencil tile config"
- cherry-ignore: add "radv/nir: call opt_remove_phis after trivial continues."
- cherry-ignore: add "amd/common: add workaround for cube map array layer clamping"
- cherry-ignore: add "radeonsi: workaround for gather4 on integer cube maps"
- cherry-ignore: add "Scons: Add LLVM 5.0 support"
- cherry-ignore: add "ac/surface: handle S8 on gfx9"
- cherry-ignore: add "radv: Check for GFX9 for 1D arrays in image_size intrinsic."
- cherry-ignore: add "glsl/linker: fix output variable overlap check"
- Update version to 17.1.10

Józef Kucia (1):

- anv: Fix descriptors copying

Matt Turner (2):

- util: Link libmesautil into u_atomic_test
- util/u_atomic: Add implementation of __sync_val_compare_and_swap_8

Nicolai HÃƒÂ¤hnle (1):

- radeonsi: apply a mask to gl_SampleMaskIn in the PS prolog

Nicolai Hähnle (4):

- st/glsl_to_tgsi: only the first (inner-most) array reference can be a 2D index
- amd/common: round cube array slice in ac_prepare_cube_coords
- radeonsi: set MIP_POINT_PRECLAMP to 0

- radeonsi: fix array textures layer coordinate

Tapani Pälli (1):

- mesa: free current ComputeProgram state in _mesa_free_context_data

4.106 Mesa 17.2.1 Release Notes / September 17, 2017

Mesa 17.2.1 is a bug fix release which fixes bugs found since the 17.2.0 release.

Mesa 17.2.1 implements the OpenGL 4.5 API, but the version reported by glGetString(GL_VERSION) or glGetIntegerv(GL_MAJOR_VERSION) / glGetIntegerv(GL_MINOR_VERSION) depends on the particular driver being used. Some drivers don't support all the features required in OpenGL 4.5. OpenGL 4.5 is **only** available if requested at context creation because compatibility contexts are not supported.

4.106.1 SHA256 checksums

```
c902d8dc2540195bc570d88af1a8fd8a1774373660a27bb1d539551f46824bc1  mesa-17.2.1.tar.gz
77385d17827cff24a3bae134342234f2efe7f7f990e778109682571dbbc9ba1e  mesa-17.2.1.tar.xz
```

4.106.2 New features

None

4.106.3 Bug fixes

- Bug 100613 - Regression in Mesa 17 on s390x (zSystems)
- Bug 101709 - [llvmpipe] piglit gl-1.0-scissor-offscreen regression
- Bug 102454 - glibc 2.26 doesn't provide anymore xlocale.h
- Bug 102467 - src/mesa/state_tracker/st_cb_readpixels.c:178]: (warning) Redundant assignment
- Bug 102502 - [bisected] Kodi crashes since commit 707d2e8b - gallium: fold u_trim_pipe_prim call from st/mesa to drivers

4.106.4 Changes

Bas Nieuwenhuizen (4):

- radv: Actually set the cmd_buffer usage_flags.
- radv: Fix vkCopyImage with both depth and stencil aspects.
- radv: Disable multilayer & multilevel DCC.
- radv: Don't allocate CMASK for linear images.

Ben Crocker (1):

- llvmpipe: lp_build_gather_elem_vec BE fix for 3x16 load

Brian Paul (1):

- llvmpipe: initialize llvmpipe->dirty with LP_NEW_SCISSOR

Charmaine Lee (1):

- vbo: fix offset in minmax cache key

Dave Airlie (12):

- radv: disable 1d/2d linear optimisation on gfx9.
- radv/gfx9: set descriptor up for base_mip to level range.
- Revert "radv: disable support for VEGA for now."
- radv/winsys: use amdgpu_bo_va_op_raw.
- radv/gfx9: allocate events from uncached VA space
- radv: use simpler indirect packet 3 if possible.
- radv: don't use iview for meta image width/height.
- radv: handle GFX9 1D textures
- radv/gfx9: set mip0-depth correctly for 2d arrays/3d images
- radv/ac: bump params array for image atomic comp swap
- radv/gfx9: fix image resource handling.
- radv/winsys: fix flags vs va_flags thinko.

Emil Velikov (7):

- docs: add sha256 checksums for 17.2.0
- cherry-ignore: add getCapability patches
- cherry-ignore: ignore gfx9 tile swizzle fix
- cherry-ignore: add execution_type() fix to the list
- cherry-ignore: add EGL+gbm swast patches
- egl/x11/dri3: adding missing __DRI_BACKGROUND_CALLABLE extension
- Update version to 17.2.1

Eric Engestrom (3):

- util: improve compiler guard
- mesa/st: remove unwanted backup file
- docs/egl: remove reference to EGL_DRIVERS_PATH

Grazvydas Ignotas (1):

- radv: don't assert on empty hash table

Jason Ekstrand (2):

- anv/formats: Nicely handle unknown VkFormat enums
- spirv: Add support for the HelperInvocation builtin

Karol Herbst (1):

- nvc0: write 0 to pipeline_statistics.cs_invocations

Kenneth Graunke (2):

- i965: Fix crash in fallback GTT mapping.

- i965: Set “Subslice Hashing Mode” to 16x16 on Apollolake.

Marek Olšák (1):

- st/mesa: skip draw calls with pipe_draw_info::count == 0

Michael Olbrich (1):

- egl/dri2: only destroy created objects

Nicolai HÃƒÂ¤hnle (1):

- radeonsi: apply a mask to gl_SampleMaskIn in the PS prolog

Nicolai Hähnle (4):

- radeonsi/gfx9: always flush DB metadata on framebuffer changes
- st/glsl_to_tgsi: only the first (inner-most) array reference can be a 2D index
- ac/surface: match Z and stencil tile config
- glsl: fix glsl_struct_field size calculations for shader cache

Ray Strode (1):

- gallivm: correct channel shift logic on big endian

Rob Clark (1):

- freedreno: skip batch-cache for compute shaders

Roland Scheidegger (1):

- st/mesa: fix view template initialization in try_pbo_readpixels

Samuel Pitoiset (1):

- radeonsi: update dirty_level_mask before dispatching

Timothy Arceri (9):

- glsl: allow NULL to be passed to encode_type_to_blob()
- glsl: stop adding pointers from gl_shader_variable to the cache
- glsl: stop adding pointers from glsl_struct_field to the cache
- glsl: add has_uniform_storage() helper to shader cache
- glsl: don’t write uniform storage offset if there isn’t one
- glsl: always write a name/label string to the cache
- compiler: move pointers to the start of shader_info
- glsl: stop adding pointers from shader_info to the cache
- glsl: stop adding pointers from bindless structs to the cache

4.107 Mesa 17.1.9 Release Notes / September 8, 2017

Mesa 17.1.9 is a bug fix release which fixes bugs found since the 17.1.8 release.

Mesa 17.1.9 implements the OpenGL 4.5 API, but the version reported by glGetString(GL_VERSION) or glGetIntegerv(GL_MAJOR_VERSION) / glGetIntegerv(GL_MINOR_VERSION) depends on the particular driver being used.

Some drivers don't support all the features required in OpenGL 4.5. OpenGL 4.5 is **only** available if requested at context creation because compatibility contexts are not supported.

4.107.1 SHA256 checksums

```
4325401b07b5f44759da781bc8d7c0a4a7244e09a702d16c037090986e07ee22  mesa-17.1.9.tar.gz
5f51ad94341696097d5df7b83818353447821685 8ac0fc8de183671a36ffea1a  mesa-17.1.9.tar.xz
```

4.107.2 New features

None

4.107.3 Bug fixes

- Bug 100613 - Regression in Mesa 17 on s390x (zSystems)
- Bug 102454 - glibc 2.26 doesn't provide anymore xlocale.h
- Bug 102467 - src/mesa/state_tracker/st_cb_readpixels.c:178]: (warning) Redundant assignment

4.107.4 Changes

Andres Gomez (8):

- docs: add sha256 checksums for 17.1.8
- cherry-ignore: added 17.2 nominations.
- cherry-ignore: add "nir: Fix system_value_from_intrinsic for subgroups"
- cherry-ignore: add "i965: Fix crash in fallback GTT mapping."
- cherry-ignore: add "radeonsi/gfx9: always flush DB metadata on framebuffer changes"
- cherry-ignore: add "radv: Fix vkCopyImage with both depth and stencil aspects."
- cherry-ignore: add "radeonsi/gfx9: proper workaround for LS/HS VGPR initialization bug"
- Update version to 17.1.9

Bas Nieuwenhuizen (3):

- radv: Fix off by one in MAX_VBS assert.
- radv: Fix sparse BO mapping merging.
- radv: Actually set the cmd_buffer usage_flags.

Ben Crocker (1):

- llvmpipe: lp_build_gather_elem_vec BE fix for 3x16 load

Charmaine Lee (1):

- vbo: fix offset in minmax cache key

Christian Gmeiner (1):

- etnaviv: use correct param for etna_compatible_rs_format(..)

Emil Velikov (3):

- egl: don't NULL deref the .get_capabilities function pointer
- egl/wayland: plug leaks in dri2_wl_create_window_surface() error path
- egl/wayland: polish object teardown in dri2_wl_destroy_surface

Eric Engestrom (1):

- util: improve compiler guard

Grazvydas Ignotas (2):

- radv: clear dynamic_shader_stages on create
- radv: don't assert on empty hash table

Ilia Mirkin (2):

- glsl: fix counting of vertex shader output slots used by explicit vars
- st/mesa: fix handling of vertex array double inputs

Jason Ekstrand (2):

- anv/formats: Nicely handle unknown VkFormat enums
- spirv: Add support for the HelperInvocation builtin

Karol Herbst (1):

- nvc0: write 0 to pipeline_statistics.cs_invocations

Michael Olbrich (1):

- egl/dri2: only destroy created objects

Ray Strode (1):

- gallivm: correct channel shift logic on big endian

Roland Scheidegger (1):

- st/mesa: fix view template initialization in try_pbo_readpixels

4.108 Mesa 17.2.0 Release Notes / September 4, 2017

Mesa 17.2.0 is a new development release. People who are concerned with stability and reliability should stick with a previous release or wait for Mesa 17.2.1.

Mesa 17.2.0 implements the OpenGL 4.5 API, but the version reported by glGetString(GL_VERSION) or glGetIntegerv(GL_MAJOR_VERSION) / glGetIntegerv(GL_MINOR_VERSION) depends on the particular driver being used. Some drivers don't support all the features required in OpenGL 4.5. OpenGL 4.5 is **only** available if requested at context creation because compatibility contexts are not supported.

4.108.1 SHA256 checksums

```
9484ad96b4bb6cda5bbf1aef52dfa35183dc21aa6258a2991c245996c2fdaf85  mesa-17.2.0.tar.gz
3123448f770eae58bc73e15480e78909defb892f10ab777e9116c9b218094943  mesa-17.2.0.tar.xz
```

4.108.2 New features

Note: some of the new features are only available with certain drivers.

- GL_ARB_bindless_texture on radeonsi
- GL_ARB_post_depth_coverage on nvc0 (GM200+)
- GL_ARB_shader_ballot on i965/gen8+
- GL_ARB_shader_group_vote on i965 (with a no-op vec4 implementation)
- GL_ARB_shader_viewport_layer_array on nvc0 (GM200+)
- GL_AMD_vertex_shader_layer on nvc0 (GM200+)
- GL_AMD_vertex_shader_viewport_index on nvc0 (GM200+)

4.108.3 Bug fixes

- Bug 68365 - [SNB Bisected]Piglit spec_ARB_framebuffer_object_fbo-blit-stretch fail
- Bug 77240 - khrplatform.h not installed if EGL is disabled
- Bug 95530 - Stellaris - colored overlay of sectors doesn't render on i965
- Bug 96449 - Dying Light reports OpenGL version 3.0 with mesa-git
- Bug 96958 - [SKL] Improper rendering in Europa Universalis IV
- Bug 97524 - Samplers referring to the same texture unit with different types should raise GL_INVALID_OPERATION
- Bug 97957 - Awful screen tearing in a separate X server with DRI3
- Bug 98238 - Witcher 2: objects are black when changing lod on Radeon Pitcairn
- Bug 98428 - Undefined non-weak-symbol in dri-drivers
- Bug 98833 - [REGRESSION, bisected] Wayland revert commit breaks non-Vsync fullscreen frame updates
- Bug 99467 - [radv] DOOM 2016 + wine. Green screen everywhere (but can be started)
- Bug 100070 - Rocket League: grass gets rendered incorrectly
- Bug 100242 - radeon buffer allocation failure during startup of Factorio
- Bug 100620 - [SKL] 48-bit addresses break DOOM
- Bug 100690 - [Regression, bisected] TotalWar: Warhammer corrupted graphics
- Bug 100741 - Chromium - Memory leak
- Bug 100785 - [regression, bisected] arb_gpu_shader5 piglit fail
- Bug 100854 - YUV to RGB Color Space Conversion result is not precise
- Bug 100871 - gles cts hangs mesa indefinitely
- Bug 100877 - vulkan/tests/block_pool_no_free regression
- Bug 100892 - Polaris 12: winsys init bad switch (missing break) initializing addrlib
- Bug 100925 - [HSW/BSW/BDW/SKL] Google Earth is not resolving all the details in the map correctly
- Bug 100937 - Mesa fails to build with GCC 4.8
- Bug 100945 - Build failure in GNOME Continuous

- Bug 100988 - glXGetCurrentDisplay() no longer works for FakeGLX contexts?
- Bug 101071 - compiling glsl fails with undefined reference to 'pthread_create'
- Bug 101088 - 'gallium: remove pipe_index_buffer and set_index_buffer' causes glitches and crash in gallium nine
- Bug 101110 - Build failure in GNOME Continuous
- Bug 101189 - Latest git fails to compile with radeon
- Bug 101252 - eglGetDisplay() is not thread safe
- Bug 101254 - VDPAU videos don't start playing with r600 gallium driver
- Bug 101283 - skylake: page fault accessing address 0
- Bug 101284 - [G45] ES2-CTS.functional.texture.specification.basic_copytexsubimage2d.cube_rgba
- Bug 101294 - radeonsi minecraft forge splash freeze since 17.1
- Bug 101306 - [BXT] gles asserts in cts
- Bug 101326 - gallium/wgl: Allow context creation without prior SetPixelFormat()
- Bug 101334 - AMD SI cards: Some vulkan apps freeze the system
- Bug 101336 - glcpp-test.sh regression
- Bug 101340 - i915_surface.c:108:4: error: too few arguments to function 'util_blitter_default_src_texture'
- Bug 101360 - Assertion failure comparing result of ballotARB
- Bug 101401 - [REGRESSION][BISECTED] GDM fails to start after 8ec4975cd83365c791a1
- Bug 101418 - Build failure in GNOME Continuous
- Bug 101451 - [G33] ES2-CTS.functional.clipping.polygon regression
- Bug 101464 - PrimitiveRestartNV inside a render list causes a crash
- Bug 101471 - Mesa fails to build: unknown typename bool
- Bug 101535 - [bisected] [Skylake] Kwin won't start and glxgears coredumps
- Bug 101538 - From "Use isl for hiz layouts" commit onwards, everything crashes with Mesa
- Bug 101539 - [Regresion] [IVB] Segment fault in recent commit in intel_miptree_level_has_hiz under Ivy bridge
- Bug 101558 - [regression][bisected] MPV playing video via opengl "randomly" results in only part of the window / screen being rendered with Mesa GIT.
- Bug 101596 - Blender renders black UI elements
- Bug 101607 - Regression in anisotropic filtering from "i965: Convert fs sampler state to use genxml"
- Bug 101657 - strtod.c:32:10: fatal error: xlocale.h: No such file or directory
- Bug 101666 - bitfieldExtract is marked as a built-in function on OpenGL ES 3.0, but was added in OpenGL ES 3.1
- Bug 101683 - Some games hang while loading when compositing is shut off or absent
- Bug 101703 - No stencil buffer allocated when requested by GLUT
- Bug 101704 - [regression][bisected] glReadPixels() from pbuffer failing in Android CTS camera tests
- Bug 101766 - Assertion '!"invalid type"' failed when constant expression involves literal of different type

- Bug 101774 - gen_clflush.h:37:7: error: implicit declaration of function '__builtin_ia32_clflush'
- Bug 101775 - Xorg segfault since 147d7fb "st/mesa: add a winsys buffers list in st_context"
- Bug 101829 - read-after-free in st_framebuffer_validate
- Bug 101831 - Build failure in GNOME Continuous
- Bug 101851 - [regression] libEGL_common.a undefined reference to '__gxx_personality_v0'
- Bug 101867 - Launch options window renders black in Feral Games in current Mesa trunk
- Bug 101876 - SIGSEGV when launching Steam
- Bug 101910 - [BYT] ES31-CTS.functional.copy_image.non_compressed.viewclass_96_bits.rgb32f_rgb32f
- Bug 101925 - playstore/webview crash
- Bug 101961 - Serious Sam Fusion hangs system completely
- Bug 101982 - Weston crashes when running an OpenGL program on i965
- Bug 101983 - [G33] ES2-CTS.functional.shaders.struct.uniform.sampler_nested* regression
- Bug 102024 - FORMAT_FEATURE_SAMPLED_IMAGE_BIT not supported for D16_UNORM and D32_SFLOAT
- Bug 102148 - Crash when running qopenglwidget example on mesa llvmpipe win32
- Bug 102241 - gallium/wgl: SwapBuffers freezing regularly with swap interval enabled
- Bug 102308 - segfault in glCompressedTextureSubImage3D

4.108.4 Changes

- GL_APPLE_vertex_array_object support removed.

4.109 Mesa 17.1.8 Release Notes / August 28, 2017

Mesa 17.1.8 is a bug fix release which fixes bugs found since the 17.1.7 release.

Mesa 17.1.8 implements the OpenGL 4.5 API, but the version reported by glGetString(GL_VERSION) or glGetIntegerv(GL_MAJOR_VERSION) / glGetIntegerv(GL_MINOR_VERSION) depends on the particular driver being used. Some drivers don't support all the features required in OpenGL 4.5. OpenGL 4.5 is **only** available if requested at context creation because compatibility contexts are not supported.

4.109.1 SHA256 checksums

```
faa59a677e88fd5224cdfebcdb6ca9ad3e3c64bd562baa8d5c3c1faeef1066b6  mesa-17.1.8.tar.gz
75ed2eaeae26ddd536150f294386468ae2e1a7717948c41cd14b7875be5269db  mesa-17.1.8.tar.xz
```

4.109.2 New features

None

4.109.3 Bug fixes

- Bug 101910 - [BYT] ES31-CTS.functional.copy_image.non_compressed.viewclass_96_bits.rgb32f_rgb32f
- Bug 102308 - segfault in glCompressedTextureSubImage3D

4.109.4 Changes

Andres Gomez (6):

- docs: add sha256 checksums for 17.1.7
- cherry-ignore: cherry-ignore: added 17.2 nominations.
- cherry-ignore: add "i965/tex: Don't pass samples to miptree_create_for_teximage"
- cherry-ignore: add "i965: Make a BRW_NEW_FAST_CLEAR_COLOR dirty bit."
- cherry-ignore: add "egl/drm: Fix misused x and y offsets in swrast_*_image*"
- Update version to 17.1.8

Christoph Haag (1):

- mesa: only copy requested compressed teximage cubemap faces

Dave Airlie (1):

- radv: don't crash if we have no framebuffer

Ilia Mirkin (2):

- glsl: add a few missing int64 constant propagation cases
- nv50/ir: properly set sType for TXF ops to U32

Jason Ekstrand (1):

- i965: Stop looking at NewDriverState when emitting 3DSTATE_URB

Kai Chen (1):

- egl/wayland: Use roundtrips when awaiting buffer release

Lionel Landwerlin (1):

- i965: perf: minimize the chances to spread queries across batchbuffers

Marek Olšák (1):

- radeonsi/gfx9: add a temporary workaround for a tessellation driver bug

Tim Rowley (1):

- swr/rast: switch gen_knobs.cpp license

Topi Pohjolainen (1):

- intel/blorp: Adjust intra-tile x when faking rgb with red-only

4.110 Mesa 17.1.7 Release Notes / August 21, 2017

Mesa 17.1.7 is a bug fix release which fixes bugs found since the 17.1.6 release.

Mesa 17.1.7 implements the OpenGL 4.5 API, but the version reported by glGetString(GL_VERSION) or glGetIntegerv(GL_MAJOR_VERSION) / glGetIntegerv(GL_MINOR_VERSION) depends on the particular driver being used. Some drivers don't support all the features required in OpenGL 4.5. OpenGL 4.5 is **only** available if requested at context creation because compatibility contexts are not supported.

4.110.1 SHA256 checksums

```
7ca484fe3194e8185d9a20261845bfd284cc40d0f3fda690d317f85ac7b91af5  mesa-17.1.7.tar.gz
69f472a874b1122404fa0bd13e2d6bf87eb3b9ad9c21d2f39872a96d83d9e5f5  mesa-17.1.7.tar.xz
```

4.110.2 New features

None

4.110.3 Bug fixes

- Bug 101334 - AMD SI cards: Some vulkan apps freeze the system
- Bug 101766 - Assertion '!"invalid type"' failed when constant expression involves literal of different type
- Bug 102024 - FORMAT_FEATURE_SAMPLED_IMAGE_BIT not supported for D16_UNORM and D32_SFLOAT
- Bug 102148 - Crash when running qopenglwidget example on mesa llvmpipe win32
- Bug 102241 - gallium/wgl: SwapBuffers freezing regularly with swap interval enabled

4.110.4 Changes

Andres Gomez (8):

- cherry-ignore: add "swr: use the correct variable for no undefined symbols"
- cherry-ignore: add "radeon/ac: use ds_swizzle for derivs on si/cik."
- cherry-ignore: add "configure: remove trailing "-a" in swr architecture teststable: 17.2 nomination only."
- cherry-ignore: added 17.2 nominations.
- cherry-ignore: add "radv: Handle VK_ATTACHMENT_UNUSED in color attachments."
- cherry-ignore: add "virgl: drop precise modifier."
- cherry-ignore: add "radv: handle 10-bit format clamping workaround."
- Update version to 17.1.7

Chris Wilson (1):

- i965/blit: Remember to include miptree buffer offset in relocs

Connor Abbott (1):

- ac/nir: fix lsb emission

Dave Airlie (5):

- intel/vec4/gs: reset nr_pull_param if DUAL_INSTANCED compile failed.
- radv: avoid GPU hangs if someone does a resolve with non-multisample src (v2)
- radv: fix f16->f32 denorm handling for SI/CIK. (v2)
- radv: fix MSAA on SI gpus.
- radv: force cs/ps/l2 flush at end of command stream. (v2)

Emil Velikov (3):

- docs: add sha256 checksums for 17.1.6
- egl/x11: don't leak xfixes_query in the error path
- egl: avoid eglCreatePlatform*Surface{EXT,} crash with invalid dpy

Eric Anholt (1):

- util: Fix build on old glibc.

Frank Richter (3):

- st/mesa: fix a null pointer access
- st/wgl: check for negative delta in wait_swap_interval()
- gallium/os: fix os_time_get_nano() to roll over less

Ilia Mirkin (3):

- glsl/ast: update rhs in addition to the var's constant_value
- nv50/ir: fix srcMask computation for TG4 and TXF
- nv50/ir: fix TXQ srcMask

Jason Ekstrand (1):

- anv/formats: Allow sampling on depth-only formats on gen7

Karol Herbst (1):

- nv50/ir: fix ConstantFolding with saturation

Kenneth Graunke (1):

- i965: Delete pitch alignment assertion in get_blit_intratile_offset_el.

Marek Olšák (2):

- ac: fail shader compilation if libelf is replaced by an incompatible version
- radeonsi: disable CE by default

Tim Rowley (1):

- swr/rast: Fix invalid casting for calls to Interlocked* functions

4.111 Mesa 17.1.6 Release Notes / August 7, 2017

Mesa 17.1.6 is a bug fix release which fixes bugs found since the 17.1.5 release.

Mesa 17.1.6 implements the OpenGL 4.5 API, but the version reported by glGetString(GL_VERSION) or glGetIntegerv(GL_MAJOR_VERSION) / glGetIntegerv(GL_MINOR_VERSION) depends on the particular driver being used.

Some drivers don't support all the features required in OpenGL 4.5. OpenGL 4.5 is **only** available if requested at context creation because compatibility contexts are not supported.

4.111.1 SHA256 checksums

```
971831bc1e748b3e8367eee6b9eb509bad2970e3c2f8520ad25f5caa12ca5491  mesa-17.1.6.tar.gz
0686deadde1f126b20aa67e47e8c50502043eee4ecdf60d5009ffda3cebfee50  mesa-17.1.6.tar.xz
```

4.111.2 New features

None

4.111.3 Bug fixes

- Bug 97957 - Awful screen tearing in a separate X server with DRI3
- Bug 101683 - Some games hang while loading when compositing is shut off or absent
- Bug 101867 - Launch options window renders black in Feral Games in current Mesa trunk

4.111.4 Changes

Andres Gomez (1):

- docs: add sha256 checksums for 17.1.5

Bas Nieuwenhuizen (1):

- radv: Don't underflow non-visible VRAM size.

Brian Paul (1):

- svga: fix texture swizzle writemasking

Chad Versace (1):

- anv/image: Fix VK_IMAGE_CREATE_CUBE_COMPATIBLE_BIT

Chris Wilson (1):

- i965: Resolve framebuffers before signaling the fence

Connor Abbott (1):

- nir: fix algebraic optimizations

Daniel Stone (1):

- st/dri: Check get-handle return value in queryImage

Dave Airlie (5):

- radv: fix non-0 based layer clears.
- radv: fix buffer views on SI/CIK.
- radv/ac: realign SI workaround with radeonsi.
- radv/ac: port SI TC L1 write corruption fix.

- radv: for stencil only set Z tile mode index to same value

Emil Velikov (23):

- cherry-ignore: add "anv: Round u_vector element sizes to a power of two"
- anv: advertise v6 of the wayland surface extension
- radv: advertise v6 of the wayland surface extension
- swrast: add dri2ConfigQueryExtension to the correct extension list
- cherry-ignore: add "anv: Transition MCS buffers from the undefined layout"
- swr: don't forget to link AVX/AVX2 against pthreads
- cherry-ignore: add "i965: Fix offset addition in get_isl_surf"
- cherry-ignore: add "i965: Fix = vs == in MCS aux usage assert."
- cherry-ignore: add a couple of radeon commits
- cherry-ignore: add "swr/rast: non-regex knob fallback code for gcc < 4.9"
- cherry-ignore: add "swr: fix transform feedback logic"
- cherry-ignore: add a couple of radeonsi/gfx9 commits
- cherry-ignore: ignore reverted st/mesa commit
- cherry-ignore: add bindless textures fix
- cherry-ignore: add "st/glsl_to_tgsi: fix getting the image type for array of structs"
- cherry-ignore: add yet another bindless textures fix
- bin/cherry-ignore: add radeonsi "fix of a fix"
- travis: lower SWR requirement to GCC 4.8, aka std=c++11
- i965: use strtol to convert the integer deviceID override
- swr: remove unneeded fallback strcasecmp define
- cherry-ignore: add a bunch more commits to the list
- fixup! cherry-ignore: add a bunch more commits to the list
- Update version to 17.1.6

Eric Anholt (1):

- broadcom/vc4: Prefer blit via rendering to the software fallback.

Eric Engestrom (1):

- configure: only install khrplatform.h if needed

Iago Toral Quiroga (2):

- anv/cmd_buffer: fix off by one error in assertion
- anv: only expose up to 28 vertex attributes

Ilia Mirkin (1):

- nv50/ir: fix threads calculation for non-compute shaders

Jason Ekstrand (5):

- anv/cmd_buffer: Properly handle render passes with 0 attachments

- anv: Stop leaking the no_aux sampler surface state
- anv/image: Add INPUT_ATTACHMENT to the list of required usages
- nir/vars_to_ssa: Handle missing struct members in foreach_deref_node
- spirv: Fix SpvImageFormatR16ui

Juan A. Suarez Romero (2):

- anv/pipeline: use unsigned long long constant to check enable vertex inputs
- anv/pipeline: do not use BITFIELD64_BIT()

Kenneth Graunke (1):

- nir: Use nir_src_copy instead of direct assignments.

Lionel Landwerlin (1):

- i965: perf: flush batchbuffers at the beginning of queries

Lucas Stach (1):

- etnaviv: fix memory leak when BO allocation fails

Marek Olšák (2):

- st/mesa: always unconditionally revalidate main framebuffer after SwapBuffers
- gallium/radeon: make S_FIXED function signed and move it to shared code

Mark Thompson (1):

- st/va: Fix scaling list ordering for H.265

Nicolai Hähnle (4):

- radeonsi/gfx9: fix crash building monolithic merged ES-GS shader
- radeonsi: fix detection of DRAW_INDIRECT_MULTI on SI
- radeonsi/gfx9: reduce max threads per block to 1024 on gfx9+
- gallium/radeon: fix ARB_query_buffer_object conversion to boolean

Thomas Hellstrom (2):

- loader/dri3: Use dri3_find_back in loader_dri3_swap_buffers_msc
- dri3: Wait for all pending swapbuffers to be scheduled before touching the front

Tim Rowley (3):

- gallium/util: fix nondeterministic avx512 detection
- swr/rast: quit using linux-specific gettid()
- swr/rast: fix scons gen_knobs.h dependency

Timothy Arceri (1):

- nir: fix nir_opt_copy_prop_vars() for arrays of arrays

Wladimir J. van der Laan (1):

- etnaviv: Clear lbl_usage array correctly

4.112 Mesa 17.1.5 Release Notes / July 14, 2017

Mesa 17.1.5 is a bug fix release which fixes bugs found since the 17.1.4 release.

Mesa 17.1.5 implements the OpenGL 4.5 API, but the version reported by glGetString(GL_VERSION) or glGetIntegerv(GL_MAJOR_VERSION) / glGetIntegerv(GL_MINOR_VERSION) depends on the particular driver being used. Some drivers don't support all the features required in OpenGL 4.5. OpenGL 4.5 is **only** available if requested at context creation because compatibility contexts are not supported.

4.112.1 SHA256 checksums

```
7e3eeee8f9c28052796eb18133c2be12c38ba34864cc496382a2fa20c29b0317  mesa-17.1.5.tar.gz
378516b171712687aace4c7ea8b37c85895231d7a6d61e1e27362cf6034fded9  mesa-17.1.5.tar.xz
```

4.112.2 New features

None

4.112.3 Bug fixes

- Bug 100242 - radeon buffer allocation failure during startup of Factorio
- Bug 101657 - strtod.c:32:10: fatal error: xlocale.h: No such file or directory
- Bug 101666 - bitfieldExtract is marked as a built-in function on OpenGL ES 3.0, but was added in OpenGL ES 3.1
- Bug 101703 - No stencil buffer allocated when requested by GLUT

4.112.4 Changes

Aaron Watry (1):

- radeon/winsys: Limit max allocation size to 70% of VRAM

Aleksander Morgado (2):

- etnaviv: fix refcnt initialization in etna_screen
- etnaviv: don't dereference etna_resource pointer if allocation fails

Alex Smith (2):

- ac/nir: Use correct LLVM intrinsics for atomic ops on imageBuffers
- ac/nir: Fix ordering of parameters for image atomic cmpswap intrinsics

Andres Gomez (3):

- docs: add sha256 checksums for 17.1.4
- cherry-ignore: i965: Fix anisotropic filtering for mag filter
- Update version to 17.1.5

Anuj Phogat (2):

- intel/isl: Use uint64_t to store total surface size

- intel/isl: Add the maximum surface size limit

Brian Paul (3):

- draw: check for line_width != 1.0f in validate_pipeline()
- svga: clamp device line width to at least 1 to fix HWv8 line stippling
- svga: fix PIPE_CAP_MAX_TEXTURE_BUFFER_SIZE value

Bruce Cherniak (1):

- swr: Limit memory held by defer deleted resources.

Chandu Babu N (1):

- st/va: Fix leak in VAAPI subpictures

Charmaine Lee (1):

- svga: fixed surface size to include array size

Connor Abbott (2):

- spirv: fix OpBitcast when the src and dst bitsize are different (v3)
- ac/nir: implement 64-bit packing and unpacking

Iago Toral Quiroga (1):

- glsl: gl_Max{Vertex,Fragment}UniformComponents exist in all desktop GL versions

Ilia Mirkin (1):

- glsl: check if any of the named builtins are available first

James Legg (2):

- ac/nir: Make intrinsic_name buffer long enough
- spirv: Fix reaching unreachable for compare exchange on images

Jason Ekstrand (1):

- nir/spirv: Use the type from the deref for atomics

Juan A. Suarez Romero (1):

- glsl: do not call link_xfb_stride_layout_qualifiers() for fragment shaders

Kenneth Graunke (2):

- i965: Use true AA line distance on G45/Ironlake.
- i965: Always set AALINEDISTANCE_TRUE on Sandybridge.

Lucas Stach (1):

- etnaviv: fix shader miscompilation with more than 16 labels

Marek Olšák (1):

- gallium/radeon: fix a possible crash for buffer exports

Neha Bhende (1):

- svga: loop over box.depth for ReadBack_image on each slice

Nicolai Hähnle (1):

- winsys/radeon: only call pb_slabs_reclaim when slabs are actually used

Olivier Lauffenburger (1):

- st/wgl: improve selection of pixel format

Philipp Zabel (1):

- st/mesa: release EGLImage on EGLImageTarget* error

Plamena Manolova (1):

- mesa/main: Move NULL pointer check.

Tim Rowley (2):

- swr/rast: _mm*_undefined_* implementations for gcc<4.9
- swr/rast: Correctly allocate SWR_STATS memory as cacheline aligned

Tomasz Figa (1):

- intel: common: Fix link failure with standalone Android build

Vinson Lee (1):

- scons: Check for xlocale.h before defining HAVE_XLOCALE_H.

4.113 Mesa 17.1.4 Release Notes / June 30, 2017

Mesa 17.1.4 is a bug fix release which fixes bugs found since the 17.1.3 release.

Mesa 17.1.4 implements the OpenGL 4.5 API, but the version reported by glGetString(GL_VERSION) or glGetIntegerv(GL_MAJOR_VERSION) / glGetIntegerv(GL_MINOR_VERSION) depends on the particular driver being used. Some drivers don't support all the features required in OpenGL 4.5. OpenGL 4.5 is **only** available if requested at context creation because compatibility contexts are not supported.

4.113.1 SHA256 checksums

```
f82fbbdf2dcec0e7e5aa3a8fe4bacd50bf4b7293cc6e1a56658ae6504d732362  mesa-17.1.4.tar.gz
06f3b0e6a28f0d20b7f3391cf67fe89ae98ecd0a686cd545da76557b6cec9cad  mesa-17.1.4.tar.xz
```

4.113.2 New features

None

4.113.3 Bug fixes

- Bug 77240 - khrplatform.h not installed if EGL is disabled
- Bug 95530 - Stellaris - colored overlay of sectors doesn't render on i965
- Bug 96958 - [SKL] Improper rendering in Europa Universalis IV
- Bug 99467 - [radv] DOOM 2016 + wine. Green screen everywhere (but can be started)
- Bug 101071 - compiling glsl fails with undefined reference to 'pthread_create'
- Bug 101252 - eglGetDisplay() is not thread safe
- Bug 101294 - radeonsi minecraft forge splash freeze since 17.1

- Bug 101451 - [G33] ES2-CTS.functional.clipping.polygon regression

4.113.4 Changes

Alex Deucher (1):

- radeonsi: add new polaris12 pci id

Andres Gomez (3):

- cherry-ignore: 17.1.4 rejected commits
- cherry-ignore: bin/get-fixes-pick-list.sh: better identify multiple "fixes:" tags
- Update version to 17.1.4

Anuj Phogat (2):

- i965: Add and initialize l3_banks field for gen7+
- i965: Fix broxton 2x6 l3 config

Ben Crocker (1):

- egl_dri2: swrastGetDrawableInfo: set *x, common.py [v2]

Brian Paul (2):

- svga: check return value from svga_set_shader(SVGA3D_SHADERTYPE_GS, NULL)
- gallium/vbuf: avoid segfault when we get invalid glDrawRangeElements()

Chad Versace (1):

- egl/android: Change order of EGLConfig generation (v2)

Chandu Babu N (1):

- change va max_entrypoints

Charmaine Lee (1):

- svga: use the winsys interface to invalidate surface

Emil Velikov (3):

- docs: add sha256 checksums for 17.1.3
- configure.ac: add -pthread to PTHREAD_LIBS
- radeonsi: include ac_binary.h for struct ac_shader_binary

Eric Engestrom (3):

- egl: properly count configs
- egl/display: only detect the platform once
- egl/display: make platform detection thread-safe

Eric Le Bihan (1):

- Fix khrplatform.h not installed if EGL is disabled.

Iago Toral Quiroga (1):

- i965: update MaxTextureRectSize to match PRMs and comply with OpenGL 4.1+

Ilia Mirkin (2):

- nv50/ir: fetch indirect sources BEFORE the op that uses them
- nv50/ir: fix combineLd/St to update existing records as necessary

Jason Ekstrand (10):

- i965: Flush around state base address
- i965: Take a uint64_t immediate in emit_pipe_control_write
- i965: Unify the two emit_pipe_control functions
- i965: Do an end-of-pipe sync prior to STATE_BASE_ADDRESS
- i965/blorp: Do an end-of-pipe sync around CCS ops
- i965: Do an end-of-pipe sync after flushes
- i965: Disable the interleaved vertex optimization when instancing
- i965: Set step_rate = 0 for interleaved vertex buffers
- spirv: Work around the Doom shader bug
- i965: Clamp clear colors to the representable range

Jonas Kulla (1):

- anv: Fix L3 cache programming on Bay Trail

Kenneth Graunke (1):

- i965: Ignore anisotropic filtering in nearest mode.

Lucas Stach (7):

- etnaviv: don't try RS blit if blit region is unaligned
- etnaviv: use padded width/height for resource copies
- etnaviv: remove bogus assert
- etnaviv: replace translate_clear_color with util_pack_color
- etnaviv: mask correct channel for RB swapped rendertargets
- etnaviv: advertise correct max LOD bias
- etnaviv: only flush resource to self if no scanout buffer exists

Marek Olšák (4):

- winsys/amdgpu: fix a deadlock when waiting for submission_in_progress
- mesa: flush vertices before changing viewports
- mesa: flush vertices before updating ctx->_Shader
- st/mesa: fix pipe_rasterizer_state::scissor with multiple viewports

Michel Dänzer (1):

- gallium/util: Break recursion in pipe_resource_reference

Nicolai Hähnle (2):

- gallium/radeon/gfx9: fix PBO texture uploads to compressed textures
- amd/common: fix off-by-one in sid_tables.py

Pierre Moreau (1):

- nv50/ir: Properly fold constants in SPLIT operation

Rob Herring (1):

- Android: major/minor/makedev live in <sys/sysmacros.h>

Topi Pohjolainen (2):

- i965: Add an end-of-pipe sync helper
- i965/gen4: Set depth offset when there is stencil attachment only

Ville Syrjälä (2):

- i915: Fix gl_Fragcoord interpolation
- i915: Fix wpos_tex vs. -1 comparison

4.114 Mesa 17.1.3 Release Notes / June 19, 2017

Mesa 17.1.3 is a bug fix release which fixes bugs found since the 17.1.2 release.

Mesa 17.1.3 implements the OpenGL 4.5 API, but the version reported by glGetString(GL_VERSION) or glGetIntegerv(GL_MAJOR_VERSION) / glGetIntegerv(GL_MINOR_VERSION) depends on the particular driver being used. Some drivers don't support all the features required in OpenGL 4.5. OpenGL 4.5 is **only** available if requested at context creation because compatibility contexts are not supported.

4.114.1 SHA256 checksums

```
81ae9127286ff8d631e466d258608d6dea9854fe7bee2e8521da44c7544f01e5  mesa-17.1.3.tar.gz
5f1ee9a8aea2880f887884df2dea0c16dd1b13eb42fd2e52265db0dc1b380e8c  mesa-17.1.3.tar.xz
```

4.114.2 New features

None

4.114.3 Bug fixes

- Bug 100988 - glXGetCurrentDisplay() no longer works for FakeGLX contexts?

4.114.4 Changes

Bas Nieuwenhuizen (3):

- radv: Set both compute and graphics SGPRS on descriptor set flush.
- radv: Dirty all descriptors sets when changing the pipeline.
- radv: Remove SI num RB override for occlusion queries.

Brian Paul (1):

- xlib: fix glXGetCurrentDisplay() failure

Chad Versace (1):

- i965/dri: Fix bad GL error in intel_create_winsys_renderbuffer()

Chuck Atkins (1):

- configure.ac: Reduce zlib requirement from 1.2.8 to 1.2.3.

Dave Airlie (3):

- radv: expose integrated device type for APUs.
- radv: set fmask state to all 0s when no fmask. (v2)
- glsl/lower_distance: only set max_array_access for 1D clip dist arrays

Emil Velikov (1):

- Update version to 17.1.3

Grazvydas Ignotas (1):

- radv: fix trace dumping for !use_ib_bos

Jason Ekstrand (4):

- i965/blorp: Take a layer range in intel_hiz_exec
- i965: Move the pre-depth-clear flush/stalls to intel_hiz_exec
- i965: Perform HiZ flush/stall prior to HiZ resolves
- i965: Mark depth surfaces as needing a HiZ resolve after blitting

José Fonseca (1):

- automake: Link all libGL.so variants with -Bsymbolic.

Juan A. Suarez Romero (1):

- docs: add sha256 checksums for 17.1.2

Lucas Stach (1):

- etnaviv: always do cpu_fini in transfer_unmap

Lyude (1):

- nvc0: disable BGRA8 images on Fermi

Marek Olšák (3):

- st/mesa: don't load cached TGSI shaders on demand
- radeonsi: fix a GPU hang with tessellation on 2-CU configs
- radeonsi: disable the patch ID workaround on SI when the patch ID isn't used (v2)

Nicolai Hähnle (1):

- radv: fewer than 8 RBs are possible

Nicolas Dechesne (1):

- util/rand_xor: add missing include statements

Tapani Pälli (1):

- egl: fix _eglQuerySurface in EGL_BUFFER_AGE_EXT case

Thomas Hellstrom (1):

- dri3/GLX: Fix drawable invalidation v2

Tim Rowley (1):

- swr: relax c++ requirement from c++14 to c++11

4.115 Mesa 17.1.2 Release Notes / June 5, 2017

Mesa 17.1.2 is a bug fix release which fixes bugs found since the 17.1.1 release.

Mesa 17.1.2 implements the OpenGL 4.5 API, but the version reported by glGetString(GL_VERSION) or glGetIntegerv(GL_MAJOR_VERSION) / glGetIntegerv(GL_MINOR_VERSION) depends on the particular driver being used. Some drivers don't support all the features required in OpenGL 4.5. OpenGL 4.5 is **only** available if requested at context creation because compatibility contexts are not supported.

4.115.1 SHA256 checksums

```
0d2020c2115db0d13a5be0075abf0da143290f69f5817a2f277861e89166a3e1  mesa-17.1.2.tar.gz
0937804f43746339b1f9540d8f9c8b4a1bb3d3eec0e4020eac283b8799798239  mesa-17.1.2.tar.xz
```

4.115.2 New features

None

4.115.3 Bug fixes

- Bug 98833 - [REGRESSION, bisected] Wayland revert commit breaks non-Vsync fullscreen frame updates
- Bug 100741 - Chromium - Memory leak
- Bug 100877 - vulkan/tests/block_pool_no_free regression
- Bug 101110 - Build failure in GNOME Continuous

4.115.4 Changes

Bartosz Tomczyk (1):

- mesa: Avoid leaking surface in st_renderbuffer_delete

Bas Nieuwenhuizen (1):

- radv: Reserve space for descriptor and push constant user SGPR setting.

Daniel Stone (7):

- vulkan: Fix Wayland uninitialised registry
- vulkan/wsi/wayland: Remove roundtrip when creating image
- vulkan/wsi/wayland: Use per-display event queue
- vulkan/wsi/wayland: Use proxy wrappers for swapchain
- egl/wayland: Don't open-code roundtrip
- egl/wayland: Use per-surface event queues

- egl/wayland: Ensure we get a back buffer

Emil Velikov (24):

- docs: add sha256 checksums for 17.1.1
- configure: move platform handling further up
- configure: rename remaining HAVE_EGL_PLATFORM_* guards
- configure: update remaining –with-egl-platforms references
- configure: loosen –with-platforms heuristics
- configure: enable the surfaceless platform by default
- configure: set HAVE_foo_PLATFORM as applicable
- configure: error out when building GLX w/o the X11 platform
- configure: check once for DRI3 dependencies
- loader: build libloader_dri3_helper.la only with HAVE_PLATFORM_X11
- configure: error out when building X11 Vulkan without DRI3
- auxiliary/vl: use vl_*_screen_create stubs when building w/o platform
- st/va: fix misplaced closing bracket
- st/omx: remove unneeded X11 include
- st/omx: fix building against X11-less setups
- gallium/targets: link against XCB only as needed
- configure: error out if building VA w/o supported platform
- configure: error out if building OMX w/o supported platform
- configure: error out if building VDPAU w/o supported platform
- configure: error out if building XVMC w/o supported platform
- travis: remove workarounds for the Vulkan target
- anv: automake: list shared libraries after the static ones
- radv: automake: list shared libraries after the static ones
- egl/wayland: select the format based on the interface used

Ian Romanick (3):

- r100: Don't assume that the base mipmap of a texture exists
- r100,r200: Don't assume glVisual is non-NULL during context creation
- r100: Use _mesa_get_format_base_format in radeon_update_wrapper

Jason Ekstrand (17):

- anv: Handle color layout transitions from the UNINITIALIZED layout
- anv: Handle transitioning depth from UNDEFINED to other layouts
- anv/image: Get rid of the memset(aux, 0, sizeof(aux)) hack
- anv: Predicate 48bit support on gen >= 8
- anv: Set up memory types and heaps during physical device init

- anv: Set image memory types based on the type count
- i965/blorp: Do and end-of-pipe sync on both sides of fast-clear ops
- i965: Round copy size to the nearest block in intel_miptree_copy
- anv: Set EXEC_OBJECT_ASYNC when available
- anv: Determine the type of mapping based on type metadata
- anv: Add valid_bufer_usage to the memory type metadata
- anv: Stop setting BO flags in bo_init_new
- anv: Make supports_48bit_addresses a heap property
- anv: Refactor memory type setup
- anv: Advertise both 32-bit and 48-bit heaps when we have enough memory
- i965: Rework Sandy Bridge HiZ and stencil layouts
- anv: Require vertex buffers to come from a 32-bit heap

Juan A. Suarez Romero (13):

- Revert "android: fix segfault within swap_buffers"
- cherry-ignore: radeonsi: load patch_id for TES-as-ES when exporting for PS
- cherry-ignore: anv: Determine the type of mapping based on type metadata
- cherry-ignore: anv: Stop setting BO flags in bo_init_new
- cherry-ignore: anv: Make supports_48bit_addresses a heap property
- cherry-ignore: anv: Advertise both 32-bit and 48-bit heaps when we have enough memory
- cherry-ignore: anv: Require vertex buffers to come from a 32-bit heap
- cherry-ignore: radv: fix regression in descriptor set freeing
- cherry-ignore: anv: Add valid_bufer_usage to the memory type metadata
- cherry-ignore: anv: Refactor memory type setup
- Revert "cherry-ignore: anv: [...]"
- Revert "cherry-ignore: anv: Require vertex buffers to come from a 32-bit heap"
- Update version to 17.1.2

Marek Olšák (1):

- radeonsi/gfx9: compile shaders with +xnack

Nicolai Hähnle (1):

- st/mesa: remove redundant stfb->iface checks

Nicolas Boichat (1):

- configure.ac: Also match -androideabi tuple

Rob Clark (1):

- freedreno: fix fence creation fail if no rendering

Tapani Pälli (1):

- egl/android: fix segfault within swap_buffers

Timothy Arceri (1):

- st/mesa: don't mark the program as in cache_fallback when there is cache miss

4.116 Mesa 17.0.7 Release Notes / June 1, 2017

Mesa 17.0.7 is a bug fix release which fixes bugs found since the 17.0.6 release.

Mesa 17.0.7 implements the OpenGL 4.5 API, but the version reported by glGetString(GL_VERSION) or glGetIntegerv(GL_MAJOR_VERSION) / glGetIntegerv(GL_MINOR_VERSION) depends on the particular driver being used. Some drivers don't support all the features required in OpenGL 4.5. OpenGL 4.5 is **only** available if requested at context creation because compatibility contexts are not supported.

4.116.1 SHA256 checksums

```
bc68d13c6b1a053b855ac453ebf7e62bd89511adf44bad6c613e09f7fa13390a  mesa-17.0.7.tar.gz
f6d75304a229c8d10443e219d6b6c0c342567dbab5a879ebe7cfa3c9139c4492  mesa-17.0.7.tar.xz
```

4.116.2 New features

None

4.116.3 Bug fixes

- Bug 98833 - [REGRESSION, bisected] Wayland revert commit breaks non-Vsync fullscreen frame updates
- Bug 100741 - Chromium - Memory leak
- Bug 100925 - [HSW/BSW/BDW/SKL] Google Earth is not resolving all the details in the map correctly

4.116.4 Changes

Andres Gomez (1):

- docs: add sha256 checksums for 17.0.6

Bartosz Tomczyk (1):

- mesa: Avoid leaking surface in st_renderbuffer_delete

Chad Versace (1):

- egl: Partially revert 23c86c74, fix eglMakeCurrent

Daniel Stone (7):

- vulkan: Fix Wayland uninitialised registry
- vulkan/wsi/wayland: Remove roundtrip when creating image
- vulkan/wsi/wayland: Use per-display event queue
- vulkan/wsi/wayland: Use proxy wrappers for swapchain
- egl/wayland: Don't open-code roundtrip

- egl/wayland: Use per-surface event queues
- egl/wayland: Ensure we get a back buffer

Emil Velikov (5):

- st/va: fix misplaced closing bracket
- anv: automake: list shared libraries after the static ones
- radv: automake: list shared libraries after the static ones
- egl/wayland: select the format based on the interface used
- Update version to 17.0.7

Eric Anholt (2):

- renderonly: Initialize fields of struct winsys_handle.
- vc4: Don't allocate new BOs to avoid synchronization when they're shared.

Hans de Goede (1):

- glxglvnddispatch: Add missing dispatch for GetDriverConfig

Ilia Mirkin (1):

- nvc0/ir: SHLADD's middle source must be an immediate

Jason Ekstrand (2):

- i965/blorp: Do and end-of-pipe sync on both sides of fast-clear ops
- i965: Round copy size to the nearest block in intel_miptree_copy

Lucas Stach (1):

- etnaviv: stop oversizing buffer resources

Nanley Chery (2):

- anv/formats: Update the three-channel BC1 mappings
- i965/formats: Update the three-channel DXT1 mappings

Pohjolainen, Topi (1):

- intel/isl/gen7: Use stencil vertical alignment of 8 instead of 4

Samuel Iglesias Gonsálvez (3):

- i965/vec4/gs: restore the uniform values which was overwritten by failed vec4_gs_visitor execution
- i965/vec4: fix swizzle and writemask when loading an uniform with constant offset
- i965/vec4: load dvec3/4 uniforms first in the push constant buffer

Tom Stellard (1):

- gallivm: Make sure module has the correct data layout when pass manager runs

4.117 Mesa 17.1.1 Release Notes / March 25, 2017

Mesa 17.1.1 is a bug fix release which fixes bugs found since the 17.1.0 release.

Mesa 17.1.1 implements the OpenGL 4.5 API, but the version reported by glGetString(GL_VERSION) or glGetIntegerv(GL_MAJOR_VERSION) / glGetIntegerv(GL_MINOR_VERSION) depends on the particular driver being used.

Some drivers don't support all the features required in OpenGL 4.5. OpenGL 4.5 is **only** available if requested at context creation because compatibility contexts are not supported.

4.117.1 SHA256 checksums

```
652315af87f2bb015ce99ee3b90d9d115d53cbf9e052493bd13d521a753b1930  mesa-17.1.1.tar.gz
aed503f94c0c1630a162a3e276f4ee12a86764cee4cb92338ea2dea99a04e7ef  mesa-17.1.1.tar.xz
```

4.117.2 New features

None

4.117.3 Bug fixes

- Bug 100854 - YUV to RGB Color Space Conversion result is not precise
- Bug 100925 - [HSW/BSW/BDW/SKL] Google Earth is not resolving all the details in the map correctly

4.117.4 Changes

Alex Deucher (1):

- radeonsi: add new vega10 pci ids

Andres Gomez (2):

- bin/get-fixes-pick-list.sh: don't warn if more than one, go over them
- bin/get-fixes-pick-list.sh: bring back the warning

Bruce Cherniak (1):

- swr: move msaa resolve to generalized StoreTile

Chad Versace (1):

- egl: Partially revert 23c86c74, fix eglMakeCurrent

Chih-Wei Huang (1):

- Android: correct libz dependency

Daniel Stone (1):

- gbm/dri: Fix sign-extension in modifier query

Emil Velikov (6):

- docs: add sha256 checksums for 17.1.0
- radeon: automake: remove unneeded elf Cflags/Libs
- configure: remove unneeded bits around libunwind handling
- egl: add g_egldispatchstubs.h to the release tarball
- automake: add SWR LLVM gen_builder.hpp workaround
- Update version to 17.1.1

Eric Anholt (2):

- renderonly: Initialize fields of struct winsys_handle.
- vc4: Don't allocate new BOs to avoid synchronization when they're shared.

Grazvydas Ignotas (2):

- anv: fix possible stack corruption
- anv: don't leak DRM devices

Hans de Goede (1):

- glxglvnddispatch: Add missing dispatch for GetDriverConfig

Ilia Mirkin (1):

- nvc0/ir: SHLADD's middle source must be an immediate

Johnson Lin (1):

- nir/lower_tex: Fix minor error in YUV color conversion matrix

Juan A. Suarez Romero (2):

- bin/get-{extra,fixes}-pick-list.sh: add support for ignore list
- bin/get-{extra,fixes}-pick-list.sh: improve output

Lucas Stach (2):

- etnaviv: stop oversizing buffer resources
- etnaviv: allow R/B swapped surfaces to be cleared

Marek Olšák (2):

- amd/addrlib: import Raven support
- radeonsi/gfx9: add support for Raven

Nanley Chery (2):

- anv/formats: Update the three-channel BC1 mappings
- i965/formats: Update the three-channel DXT1 mappings

Nicolai Hähnle (5):

- radeonsi: mark fast-cleared textures as compressed when dirtying
- radeonsi: fix primitive ID in fragment shader when using tessellation
- radeonsi: fix gl_PrimitiveID in tessellation with instanced draws on SI
- radeonsi: fix gl_PrimitiveIDIn in geometry shader when using tessellation
- st/mesa: remove an incorrect assertion

Pohjolainen, Topi (1):

- intel/isl/gen7: Use stencil vertical alignment of 8 instead of 4

Rob Clark (2):

- mesa/st: fix yuv EGLImage's
- freedreno: fix crash when flush() but no rendering

Rob Herring (1):

- virgl: fix virgl_bo_transfer_{put, get} box struct copy

Samuel Iglesias Gonsálvez (3):

- i965/vec4/gs: restore the uniform values which was overwritten by failed vec4_gs_visitor execution
- i965/vec4: fix swizzle and writemask when loading an uniform with constant offset
- i965/vec4: load dvec3/4 uniforms first in the push constant buffer

Tom Stellard (1):

- gallivm: Make sure module has the correct data layout when pass manager runs

4.118 Mesa 17.0.6 Release Notes / May 12, 2017

Mesa 17.0.6 is a bug fix release which fixes bugs found since the 17.0.5 release.

Mesa 17.0.6 implements the OpenGL 4.5 API, but the version reported by glGetString(GL_VERSION) or glGetIntegerv(GL_MAJOR_VERSION) / glGetIntegerv(GL_MINOR_VERSION) depends on the particular driver being used. Some drivers don't support all the features required in OpenGL 4.5. OpenGL 4.5 is **only** available if requested at context creation because compatibility contexts are not supported.

4.118.1 SHA256 checksums

```
f1b2497d553e9a584f0caa3a2d9d310e27ead15fb0af170da69f6e70fb5031cd  mesa-17.0.6.tar.gz
89ecf3bcd0f18dcca5aaa42bf36bb52a2df33be89889f94aaaad91f7a504a69d  mesa-17.0.6.tar.xz
```

4.118.2 New features

None

4.118.3 Bug fixes

- Bug 98428 - Undefined non-weak-symbol in dri-drivers
- Bug 100854 - YUV to RGB Color Space Conversion result is not precise

4.118.4 Changes

Adam Jackson (1):

- egl/platform/drm: Don't take display ownership until gbm is initialized

Andres Gomez (7):

- docs: add sha256 checksums for 17.0.5
- travis: replace Trusty-based LLVM toolchain apt-get with apt addon
- travis: add the possibility of using the txc-dxtn library
- cherry-ignore: 17.1 nominations only
- cherry-ignore: fix regression in descriptor set freeing.

- cherry-ignore: rejected commits
- Update version to 17.0.6

Ben Boeckel (1):

- scons: update for LLVM 4.0

Brian Paul (1):

- st/mesa: move duplicated st_ws_framebuffer() function into header file

Chad Versace (3):

- egl: Emit error when EGLSurface is lost
- egl/android: Cancel any outstanding ANativeBuffer in surface destructor
- egl/android: Mark surface as lost when dequeueBuffer fails

Christian Gmeiner (1):

- etnaviv: add L8A8_UNORM texture format

Dave Airlie (2):

- radv/wsi: report presentation error per image request
- radv: enable POLARIS12 support.

Emil Velikov (21):

- travis: correct libdrm required regex to also track libdrm itself
- travis: add nearly all gallium drivers to the list
- travis: use both cores for make/make check
- travis: bring the scons build on par with AppVeyor
- travis: explicitly LD_LIBRARY_PATH the local libraries
- travis: enable apt cache
- travis: automatically manage ccache caching
- travis: remove unused -dev packages
- travis: rework "if test" blocks in the script section
- travis: split out matrix from env
- travis: add separate "scons" and "scons llvm" targets
- travis: add "scons swr" to the build matrix
- travis: add "make swr" to the build matrix
- travis: split the make target to three separate ones
- travis: model scons check target like the make one
- travis: add Gallium state-tracker targets
- travis: enable wayland support
- travis: bump MAKEFLAGS to -j4
- gallium/dri: always link against shared glapi
- mesa/dri: always link against shared glapi

- glx: glX_proto_send.py: use correct compile guard GLX_INDIRECT_RENDERING

Eric Anholt (1):

- nir: Pick just the channels we want for bitmap and drawpixels lowering.

Ilia Mirkin (1):

- gallium/targets: fix bool setting on BE architectures

Jason Ekstrand (1):

- anv/cmd_buffer: Use the device allocator for QueueSubmit

Johnson Lin (1):

- nir/lower_tex: Fix minor error in YUV color conversion matrix

Marek Olšák (2):

- radeonsi: adjust ESGS ring buffer size computation on VI
- radeonsi: apply the tess+GS hang workaround to Polaris12 as well

Nicolai Hähnle (1):

- radeonsi: fix gl_PrimitiveID in tessellation with instanced draws on SI

Philipp Zabel (3):

- renderonly: close transfer prime_fd
- renderonly: drop resources on destroy
- renderonly: use drmIoctl

Rhys Kidd (3):

- travis: Support LLVM 3.8+ on Trusty-based Travis-CI via apt-get not apt addon
- travis: Add radv vulkan driver to continuous integration
- travis: Add radeonsi to continuous integration

Rob Clark (1):

- freedreno/a3xx: fix hang w/ large render targets and small gmem

Samuel Iglesias Gonsálvez (5):

- i965/vec4: fix vertical stride to avoid breaking region parameter rule
- i965/vec4: fix register width for DF VGRF and UNIFORM
- i965/vec4: don't modify regioning parameters to the sources of DF align1 instructions
- anv: anv_gem_mmap() returns MAP_FAILED as mapping error
- anv: vkBindImageMemory() should return VK_ERROR_OUT_OF_{HOST,DEVICE}_MEMORY on failure

4.119 Mesa 17.1.0 Release Notes / May 10, 2017

Mesa 17.1.0 is a new development release. People who are concerned with stability and reliability should stick with a previous release or wait for *Mesa 17.1.1*.

Mesa 17.1.0 implements the OpenGL 4.5 API, but the version reported by glGetString(GL_VERSION) or glGetIntegerv(GL_MAJOR_VERSION) / glGetIntegerv(GL_MINOR_VERSION) depends on the particular driver being used.

Some drivers don't support all the features required in OpenGL 4.5. OpenGL 4.5 is **only** available if requested at context creation because compatibility contexts are not supported.

4.119.1 SHA256 checksums

```
c388069581a72853161657ac365f2c083afabd7cffd53f80513dacfa1cfa58a8  mesa-17.1.0.tar.gz
cf234a6ed4764673886b6661553b54675776ef0898f774716173cec890ac3b17  mesa-17.1.0.tar.xz
```

4.119.2 New features

Note: some of the new features are only available with certain drivers.

- OpenGL 4.2 on i965/ivb
- GL_ARB_gpu_shader_fp64 on i965/ivybridge
- GL_ARB_gpu_shader_int64 on i965/gen8+, nvc0, radeonsi, softpipe, llvmpipe
- GL_ARB_shader_ballot on nvc0, radeonsi
- GL_ARB_shader_clock on nv50, nvc0, radeonsi
- GL_ARB_shader_group_vote on radeonsi
- GL_ARB_shader_precision on i965/ivb
- GL_ARB_shader_viewport_layer_array on radeonsi
- GL_ARB_sparse_buffer on radeonsi/CIK+
- GL_ARB_transform_feedback2 on i965/gen6
- GL_ARB_transform_feedback_overflow_query on i965/gen6+
- GL_ARB_vertex_attrib_64bit on i965/ivb
- GL_NV_fill_rectangle on nvc0
- Geometry shaders enabled on swr

4.119.3 Bug fixes

- Bug 68504 - 9.2-rc1 workaround for clover build failure on ppc/altivec: cannot convert 'bool' to '__vector(4) __bool int' in return
- Bug 84325 - X.Org segfaults when starting DE on an Intel+Radeon laptop, caused by libpciaccess cleanup, patch attached
- Bug 93089 - mesa fails to check for gcc atomic primitives before using them
- Bug 95460 - Please add more drivers (freedreno, virgl) to features.txt status document
- Bug 96743 - [BYT, HSW, SKL, BXT, KBL] GPU hangs with GfxBench 4.0 CarChase
- Bug 97102 - [dri][swr] stack overflow / infinite loop with GALLIUM_DRIVER=swr
- Bug 97338 - Black squares in the Spec Ops: The Line chapter select screen
- Bug 97524 - Samplers referring to the same texture unit with different types should raise GL_INVALID_OPERATION
- Bug 97967 - glsl/tests/cache-test regression

- Bug 97988 - [radeonsi] playing back videos with VDPAU exhibits deinterlacing/anti-aliasing issues not visible with VA-API
- Bug 98263 - [radv] The Talos Principle fails to launch with "Fatal error: Cannot set display mode."
- Bug 98428 - Undefined non-weak-symbol in dri-drivers
- Bug 98502 - Delay when starting firefox, thunderbird or chromium and dmesg spam
- Bug 98869 - Electronic Super Joy graphic artefacts (regression,bisected)
- Bug 98975 - Wasteland 2 Directors Cut: Hangs. GPU fault
- Bug 99010 - –disable-gallium-llvm no longer recognized
- Bug 99246 - [d3dadapter+radeonsi & bisect] EVE-Online : hang on wormhole sight
- Bug 99265 - i965: Piglit egl_khr_gl_renderbuffer_image-clear-shared-image fails
- Bug 99339 - Blender line rendering broken after removing XY clipping of lines
- Bug 99401 - [g33] regression: piglit.spec.!opengl 1_0.gl-1_0-beginend-coverage
- Bug 99450 - [amdgpu] Payday 2 visual glitches on some models
- Bug 99451 - polygon offset use after free
- Bug 99456 - Firefox crashing when opening about:support with WebGL2 enabled
- Bug 99465 - vtn_vector_construct writing out of bounds when given multiple non-zero length sources
- Bug 99484 - Crusader Kings 2 - Loading bars, siege bars, morale bars, etc. do not render correctly
- Bug 99532 - Compute shader doesn't give right result under some circumstances
- Bug 99542 - vdpau logging errors since gallium/radeon: adjust the rule for using the LINEAR_ALIGNED layout
- Bug 99631 - segfault with OSVRTrackerView and openscenegraph git master
- Bug 99633 - rasterizer/core/clip.h:279:49: error: 'const struct API_STATE' has no member named 'linkageCount'
- Bug 99660 - Not all of the int64 conversion opcodes got implemented
- Bug 99677 - heap-use-after-free in glsl
- Bug 99692 - [radv] Mostly broken on Hawaii PRO/CIK ASICs
- Bug 99701 - loader.c:353:8: error: implicit declaration of function 'geteuid' is invalid in C99 [-Werror,-Wimplicit-function-declaration]
- Bug 99715 - Don't print: "Note: Buggy applications may crash, if they do please report to vendor"
- Bug 99789 - Memory leak on failure to create an ir_constant in calculate_iterations in loop_controls.cpp
- Bug 99817 - [softpipe] piglit glsl-fs-tan-1 regression
- Bug 99842 - GL_ARB_transform_feedback2 on i965 gen6
- Bug 99850 - Tessellation bug on Carrizo
- Bug 99918 - disk_cache.h:57:20: error: no member named 'st_mtim' in 'struct stat'
- Bug 99953 - device9.c:122:49: error: 'PIPE_CAP_USER_INDEX_BUFFERS' undeclared (first use in this function)
- Bug 99955 - [r600g] GPU load always displayed at 100% with GALLIUM_HUD=GPU-load
- Bug 100026 - piglit.spec.arb_shader_subroutine.compiler.direct-call_vert regression

- Bug 100049 - "ralloc: Make sure ralloc() allocations match malloc()'s alignment." causes seg fault in 32bit build
- Bug 100060 - wsi/wsi_common_wayland.c:25:41: fatal error: wayland-drm-client-protocol.h: No such file or directory
- Bug 100061 - LODQ instruction generated with invalid dst mask
- Bug 100068 - LLVM ERROR: Cannot select: intrinsic %llvm.amdgcn.buffer.load.format
- Bug 100088 - piglit.spec.arb_get_texture_sub_image.arb_get_texture_sub_image regressions
- Bug 100091 - Failure to create folder for on-disk shader cache
- Bug 100133 - swr_context.cpp:336:44: error: invalid conversion from 'uint {aka unsigned int}' to 'pipe_render_cond_flag' [-fpermissive]
- Bug 100154 - test_eu_compact regression
- Bug 100180 - Build failure in GNOME Continuous
- Bug 100182 - Flickering in The Talos Principle on Sky Lake GT4.
- Bug 100201 - Windows scons build with MSVC toolchain and LLVM 4.0 fails
- Bug 100223 - marshal_generated.c:38:10: fatal error: 'X11/Xlib-xcb.h' file not found
- Bug 100236 - Undefined symbols for architecture x86_64: "typeinfo for llvm::RTDyldMemoryManager"
- Bug 100259 - [EGL] [GBM] undefined reference to 'gbm_bo_create_with_modifiers'
- Bug 100288 - clover unable to run OpenCL kernels since 03127bb radeonsi: compile all TGSI compute shaders asynchronously
- Bug 100303 - Adding a single, meaningless if-else to a shader source leads to different image
- Bug 100391 - SachaWillems deferredmultisampling asserts
- Bug 100452 - push_constants host memory leak when resetting command buffer
- Bug 100531 - [regression] Broken graphics in several games
- Bug 100562 - u_debug_stack.c:59: undefined reference to '_Ux86_64_getcontext'
- Bug 100569 - core/resource.cpp:36:33: error: non-constant-expression cannot be narrowed from type 'int' to 'int16_t' (aka 'short') in initializer list [-Wc++11-narrowing]
- Bug 100574 - anv_device.c:189: undefined reference to 'anv_gem_supports_48b_addresses'
- Bug 100582 - [GEN8+] piglit.spec.arb_stencil_texturing.glblitframebuffer corrupts state.gl_texture* assertions
- Bug 100600 - anv_device.c:1337: undefined reference to 'anv_gem_busy'
- Bug 100620 - [SKL] 48-bit addresses break DOOM
- Bug 100663 - commit 61e47d92c5196 breaks RS780
- Bug 100690 - [Regression, bisected] TotalWar: Warhammer corrupted graphics
- Bug 100892 - Polaris 12: winsys init bad switch (missing break) initializing addrlib

4.119.4 Changes

- Removed the ilo gallium driver.
- The configure option –enable-gallium-llvm is superseded by –enable-llvm.
- The swr driver now requires LLVM >= 3.9.0 and a C++14 capable compiler.

- The radeonsi driver now requires LLVM 3.8.0.
- The MESA_GLSL=opt and MESA_GLSL=no_opt environment vars have been removed.
- The –with-egl-platforms configure option is deprecated. Use –with-platforms instead.

4.120 Mesa 17.0.5 Release Notes / April 28, 2017

Mesa 17.0.5 is a bug fix release which fixes bugs found since the 17.0.4 release.

Mesa 17.0.5 implements the OpenGL 4.5 API, but the version reported by glGetString(GL_VERSION) or glGetIntegerv(GL_MAJOR_VERSION) / glGetIntegerv(GL_MINOR_VERSION) depends on the particular driver being used. Some drivers don't support all the features required in OpenGL 4.5. OpenGL 4.5 is **only** available if requested at context creation because compatibility contexts are not supported.

4.120.1 SHA256 checksums

```
7510eee0d0077860b250d30d73305048c2df4ba09ea8fc04e4f3eec7beece301  mesa-17.0.5.tar.gz
668efa445d2f57a26e5c096b1965a685733a3b57d9c736f9d6460263847f9bfe  mesa-17.0.5.tar.xz
```

4.120.2 New features

None

4.120.3 Bug fixes

- Bug 97524 - Samplers referring to the same texture unit with different types should raise GL_INVALID_OPERATION

4.120.4 Changes

Andres Gomez (16):

- cherry-ignore: Add the pci_id into the shader cache UUID
- cherry-ignore: fix crash if ctx torn down with no rendering
- cherry-ignore: Fix typos.
- cherry-ignore: Revert "etnaviv: Cannot render to rb-swapped formats"
- cherry-ignore: Revert "i965/fs: Don't emit SEL instructions for type-converting MOVs."
- cherry-ignore: fix typo in a2b10g10r10 fast clear calculation
- cherry-ignore: remove unused anv_dispatch_table dtable
- cherry-ignore: remove unused radv_dispatch_table dtable
- cherry-ignore: make radv_resolve_entrypoint static
- cherry-ignore: vulkan: add support for libmesa_vulkan_util
- cherry-ignore: r600: fix libmesa_amd_common dependency
- cherry-ignore: remove dead brw_new_shader() declaration

- cherry-ignore: remove i965_symbols_test reference from .gitignore
- cherry-ignore: automake: ensure that the destination directory is created
- cherry-ignore: provide required gem stubs for the tests
- Update version to 17.0.5

Boyan Ding (2):

- nvc0/ir: Properly handle a "split form" of predicate destination
- nir: Destination component count of shader_clock intrinsic is 2

Emil Velikov (5):

- docs: add sha256 checksums for 17.0.4
- winsys/sw/dri: don't use GNU void pointer arithmetic
- st/clover: add space between < and ::
- configure.ac: check require_basic_egl only if egl enabled
- st/mesa: automake: honour the vdpau header install location

Francisco Jerez (2):

- intel/fs: Use regs_written() in spilling cost heuristic for improved accuracy.
- intel/fs: Take into account amount of data read in spilling cost heuristic.

Grazvydas Ignotas (1):

- radv: report timestampPeriod correctly

Jason Ekstrand (5):

- anv/blorp: Flush the texture cache in UpdateBuffer
- anv/cmd_buffer: Flush the VF cache at the top of all primaries
- anv/cmd_buffer: Always set up a null surface state
- anv/cmd_buffer: Use the null surface state for ATTACHMENT_UNUSED
- anv/blorp: Properly handle VK_ATTACHMENT_UNUSED

Kenneth Graunke (1):

- i965/vec4: Avoid reswizzling MACH instructions in opt_register_coalesce().

Marek Olšák (1):

- st/mesa: invalidate the readpix cache in st_indirect_draw_vbo

Nanley Chery (1):

- anv/cmd_buffer: Disable CCS on BDW input attachments

Nicolai Hähnle (4):

- mesa: fix remaining xfb prims check for GLES with multiple instances
- mesa: extract need_xfb_remaining_prims_check
- mesa: move glMultiDrawArrays to vbo and fix error handling
- vbo: fix gl_DrawID handling in glMultiDrawArrays

Rob Clark (1):

- util/queue: don't hang at exit

Timothy Arceri (1):

- mesa: validate sampler type across the whole program

4.121 Mesa 17.0.4 Release Notes / April 17, 2017

Mesa 17.0.4 is a bug fix release which fixes bugs found since the 17.0.3 release.

Mesa 17.0.4 implements the OpenGL 4.5 API, but the version reported by glGetString(GL_VERSION) or glGetIntegerv(GL_MAJOR_VERSION) / glGetIntegerv(GL_MINOR_VERSION) depends on the particular driver being used. Some drivers don't support all the features required in OpenGL 4.5. OpenGL 4.5 is **only** available if requested at context creation because compatibility contexts are not supported.

4.121.1 SHA256 checksums

```
c4c34ba05d48f76b45bc05bc4b6e9242077f403d63c4f0c355c7b07786de233e  mesa-17.0.4.tar.gz
1269dc8545a193932a0779b2db5bce9be4a5f6813b98c38b93b372be8362a346  mesa-17.0.4.tar.xz
```

4.121.2 Next release

Mesa 17.0.5 is expected in approximatelly two weeks. See the release *calendar* for details.

4.121.3 New features

None

4.121.4 Bug fixes

- Bug 99515 - SIGSEGV MAPERR on Android nougat-x86 with mesa 17.0.0rc
- Bug 100391 - SachaWillems deferredmultisampling asserts
- Bug 100452 - push_constants host memory leak when resetting command buffer
- Bug 100582 - [GEN8+] piglit.spec.arb_stencil_texturing.glblitframebuffer corrupts state.gl_texture* assertions

4.121.5 Changes

Alex Deucher (1):

- radeonsi: add new polaris10 pci id

Alex Smith (1):

- radv: Invalidate L2 for TRANSFER_WRITE barriers

Andres Gomez (1):

- docs: add sha256 checksums for 17.0.3

Craig Stout (1):

- anv/cmd_buffer: fix host memory leak

Emil Velikov (3):

- Revert "cherry-ignore: add the Flush after unmap in gbm/dri fix"
- Revert "freedreno: fix memory leak"
- Update version to 17.0.4

Fabio Estevam (1):

- loader: Move non-error message to debug level

Ilia Mirkin (4):

- nvc0/ir: fix LSB/BFE/BFI implementations
- nvc0/ir: fix overwriting of offset register with interpolateAtOffset
- nvc0: increase texture buffer object alignment to 256 for pre-GM107
- nouveau: when mapping a persistent buffer, synchronize on former xfers

Jason Ekstrand (5):

- i965/fs: Always provide a default LOD of 0 for TXS and TXL
- anv/pipeline: Properly handle unset gl_Layer and gl_ViewportIndex
- anv/blorp: Align vertex buffers to 64B
- i965/blorp: Align vertex buffers to 64B
- i965/blorp: Bump the batch space estimate

Jerome Duval (2):

- haiku: build fixes around debug defines
- haiku/winsys: fix dt prototype args

Julien Isorce (4):

- winsys/radeon: check null in radeon_cs_create_fence
- winsys/radeon: check null return from radeon_cs_create_fence in cs_flush
- radeon: initialize hole variable before calling container_of
- radeon_drm_bo: explicitly check return value of drmCommandWriteRead

Kenneth Graunke (4):

- i965: Document the sad story of the kernel command parser.
- i965: Set screen->cmd_parser_version to 0 if we can't write registers.
- i965: Skip register write detection when possible.
- i965: Set kernel features before computing max GL version.

Marek Olšák (1):

- targets: export radeon winsys_create functions to silence LLVM warning

Michal Srb (1):

- st: Add cubeMapFace parameter to st_finalize_texture.

Thomas Hellstrom (1):

- gbm/dri: Flush after unmap

4.122 Mesa 17.0.3 Release Notes / April 1, 2017

Mesa 17.0.3 is a bug fix release which fixes bugs found since the 17.0.2 release.

Mesa 17.0.3 implements the OpenGL 4.5 API, but the version reported by glGetString(GL_VERSION) or glGetIntegerv(GL_MAJOR_VERSION) / glGetIntegerv(GL_MINOR_VERSION) depends on the particular driver being used. Some drivers don't support all the features required in OpenGL 4.5. OpenGL 4.5 is **only** available if requested at context creation because compatibility contexts are not supported.

4.122.1 SHA256 checksums

```
8253edf1bdd7b14ab63d5982349143a5c9ac3767f39a63257cc9d7e7d92f60f1  mesa-17.0.3.tar.gz
ca646f5075a002d60ef9123c8a4331cede155c01712ef945a65c59a5e69fe7ed  mesa-17.0.3.tar.xz
```

4.122.2 New features

None

4.122.3 Bug fixes

- Bug 96743 - [BYT, HSW, SKL, BXT, KBL] GPU hangs with GfxBench 4.0 CarChase
- Bug 99246 - [d3dadapter+radeonsi & bisect] EVE-Online : hang on wormhole sight
- Bug 100061 - LODQ instruction generated with invalid dst mask
- Bug 100182 - Flickering in The Talos Principle on Sky Lake GT4.
- Bug 100201 - Windows scons build with MSVC toolchain and LLVM 4.0 fails

4.122.4 Changes

Alex Deucher (1):

- radeonsi: add new polaris12 pci id

Andres Gomez (5):

- glsl: on UBO/SSBOs link error reset the number of active blocks to 0
- cherry-ignore: add the Invalidate L2 for TRANSFER_WRITE barriers fix
- cherry-ignore: add the Flush after unmap in gbm/dri fix
- cherry-ignore: corrected typo in the Flush after unmap in gbm/dri fix
- Update version to 17.0.3

Axel Davy (2):

- st/nine: Resolve deadlock in surface/volume dtors when using csmt
- st/nine: Use atomics for available_texture_mem

Bas Nieuwenhuizen (1):

- radv: flush DB cache before and after HTILE decompress.

Dave Airlie (1):

- radv: fix primitive reset index emission

Emil Velikov (1):

- docs: add sha256 checksums for 17.0.2

Ilia Mirkin (1):

- st/mesa: set result writemask based on ir type

Jan Vesely (1):

- clover: use pipe_resource references

Jason Ekstrand (9):

- anv/query: Invalidate the correct range
- anv/GetQueryPoolResults: Actually implement the spec
- anv/image: Return early when unbinding an image
- anv/query: Fix the location of timestamp availability
- anv: Make anv_get_layerCount a macro
- anv/blorp: Use anv_get_layerCount everywhere
- anv/cmd_buffer: Apply flush operations prior to executing secondaries
- anv/cmd_buffer: Fix bad indentation
- anv: Flush caches prior to PIPELINE_SELECT on all gens

José Fonseca (1):

- c11/threads: Include thr/xtimec.h for xtime definition when building with MSVC.

Juan A. Suarez Romero (1):

- tests/cache_test: allow crossing mount points

Karol Herbst (1):

- nvc0/ir: treat FMA like MAD for operand propagation

Kenneth Graunke (1):

- i965: Fall back to GL 4.2/4.3 on Haswell if the kernel isn't new enough.

Marek Olšák (1):

- radeonsi: don't hang on shader compile failure

Matt Turner (1):

- i965/fs: Don't emit SEL instructions for type-converting MOVs.

Nanley Chery (1):

- intel: Correct the BDW surface state size

Nicolai Hähnle (1):

- mesa/main: fix MultiDrawElements[BaseVertex] validation of primcount

Rob Clark (1):

- freedreno: fix memory leak

Tim Rowley (1):

- swr: [rasterizer jitter] fix llvm >= 5.0 build break

Timothy Arceri (2):

- glsl: fix lower jumps for returns when loop is inside an if
- mesa: update lower_jumps tests after bug fix

Topi Pohjolainen (1):

- i965/gen8+: Do full stall when switching pipeline

Xu Randy (2):

- anv/blorp: Fix a crash in CmdClearColorImage
- anv/genX: Solve the vkCreateGraphicsPipelines crash

4.123 Mesa 17.0.2 Release Notes / March 20, 2017

Mesa 17.0.2 is a bug fix release which fixes bugs found since the 17.0.1 release.

Mesa 17.0.2 implements the OpenGL 4.5 API, but the version reported by glGetString(GL_VERSION) or glGetIntegerv(GL_MAJOR_VERSION) / glGetIntegerv(GL_MINOR_VERSION) depends on the particular driver being used. Some drivers don't support all the features required in OpenGL 4.5. OpenGL 4.5 is **only** available if requested at context creation because compatibility contexts are not supported.

4.123.1 SHA256 checksums

```
2e0f41e7974ba7a36ca32bbeaf8ebcd65c8fd4d2dc9872f04d4becbd5e7a8cb5  mesa-17.0.2.tar.gz
f8f191f909e01e65de38d5bdea5fb057f21649a3aed20948be02348e77a689d4  mesa-17.0.2.tar.xz
```

4.123.2 New features

None

4.123.3 Bug fixes

- Bug 68504 - 9.2-rc1 workaround for clover build failure on ppc/altivec: cannot convert 'bool' to '__vector(4) __bool int' in return
- Bug 97988 - [radeonsi] playing back videos with VDPAU exhibits deinterlacing/anti-aliasing issues not visible with VA-API
- Bug 99484 - Crusader Kings 2 - Loading bars, siege bars, morale bars, etc. do not render correctly
- Bug 99715 - Don't print: "Note: Buggy applications may crash, if they do please report to vendor"
- Bug 100049 - "ralloc: Make sure ralloc() allocations match malloc()'s alignment." causes seg fault in 32bit build

4.123.4 Changes

Alex Smith (3):

- radv: Emit pending flushes before executing a secondary command buffer
- radv: Flush before copying with PKT3_WRITE_DATA in CmdUpdateBuffer
- radv/ac: Fix shared memory offset calculation

Bas Nieuwenhuizen (3):

- radv: Disable HTILE for textures with multiple layers/levels.
- radv: Emit cache flushes before CP DMA.
- Revert "radv: Emit cache flushes before CP DMA."

Dave Airlie (3):

- radv: drop Z24 support.
- radv: disable mip point pre clamping.
- radv: setup llvm target data layout

Emil Velikov (4):

- docs: add sha256 checksums for 17.0.1
- cherry-ignore: add the swizzle blorp_clear fix
- i965: move brw_define.h ifndef guard to the top
- Update version to 17.0.2

Fredrik Höglund (2):

- radv: fix the dynamic buffer index in vkCmdBindDescriptorSets
- radv/ac: fix multiple descriptor sets with dynamic buffers

Gregory Hainaut (1):

- glapi: fix typo in count_scale

Ilia Mirkin (2):

- nvc0: take extra pushbuf space into account for pushbuf_space calls
- nvc0: increase alignment to 256 for texture buffers on fermi

Jacob Lifshay (1):

- vulkan/wsi: Improve the DRI3 error message

James Legg (1):

- radv: Fix using more than 4 bound descriptor sets

Jason Ekstrand (7):

- anv/blorp/clear_subpass: Only set surface clear color for fast clears
- anv: Accurately advertise dynamic descriptor limits
- anv: Stall before fast-clear operations
- anv: Properly handle destroying NULL devices and instances
- anv/blorp: Turn off AUX after doing a CCS_D resolve

- anv/blorp: Only set a clear color for resolves if fast-cleared
- nir/intrinsics: Make load_barycentric_input take a 2-component coor

Jonas Pfeil (1):

- ralloc: Make sure ralloc() allocations match malloc()'s alignment.

Kenneth Graunke (1):

- egl: Ensure ResetNotificationStrategy matches for shared contexts.

Marek Olšák (3):

- st/mesa: reset sample_mask, min_sample, and render_condition for PBO ops
- st/mesa: set blend state for PBO readbacks
- radeonsi: mark all bound shader buffer ranges as initialized

Matt Turner (1):

- clover: Work around build failure with AltiVec.

Nanley Chery (2):

- anv/pass: Avoid accessing attachment array out of bounds
- anv/image: Remove extra dependency on HiZ-specific variable

Nicolai Hähnle (2):

- st/glsl_to_tgsi: avoid iterating past the head of the instruction list
- st/mesa: inform the driver of framebuffer changes before compute dispatches

Robert Foss (1):

- mesa: Avoid read of uninitialized variable

Samuel Iglesias Gonsálvez (5):

- i965/fs: mark last DF uniform array element as 64 bit live one
- i965/fs: detect different bit size accesses to uniforms to push them in proper locations
- i965/fs: fix indirect load DF uniforms on BSW/BXT
- i965/fs: fix source type when emitting MOV_INDIRECT to read ICP handles
- i965/fs: emit MOV_INDIRECT with the source with the right register type

Samuel Pitoiset (1):

- radeonsi: disable sinking common instructions down to the end block

4.124 Mesa 13.0.6 Release Notes / March 20, 2017

Mesa 13.0.6 is a bug fix release which fixes bugs found since the 13.0.5 release.

Mesa 13.0.6 implements the OpenGL 4.4 API, but the version reported by glGetString(GL_VERSION) or glGetIntegerv(GL_MAJOR_VERSION) / glGetIntegerv(GL_MINOR_VERSION) depends on the particular driver being used. Some drivers don't support all the features required in OpenGL 4.4. OpenGL 4.4 is **only** available if requested at context creation because compatibility contexts are not supported.

4.124.1 SHA256 checksums

```
1076590f29103f022a2cd87e6dff6ae77072013745603d06b0410c373ab2bb1a  mesa-13.0.6.tar.gz
29ef104a7fc082d352b1599bd6cb1d040be424ccd22f5e0eb7ee9b0e9acd3597  mesa-13.0.6.tar.xz
```

4.124.2 New features

None

4.124.3 Bug fixes

- Bug 68504 - 9.2-rc1 workaround for clover build failure on ppc/altivec: cannot convert 'bool' to '__vector(4) __bool int' in return
- Bug 97102 - [dri][swr] stack overflow / infinite loop with GALLIUM_DRIVER=swr
- Bug 98869 - Electronic Super Joy graphic artefacts (regression,bisected)
- Bug 99401 - [g33] regression: piglit.spec.!opengl 1_0.gl-1_0-beginend-coverage
- Bug 99456 - Firefox crashing when opening about:support with WebGL2 enabled
- Bug 99677 - heap-use-after-free in glsl
- Bug 99715 - Don't print: "Note: Buggy applications may crash, if they do please report to vendor"
- Bug 99850 - Tessellation bug on Carrizo
- Bug 100049 - "ralloc: Make sure ralloc() allocations match malloc()'s alignment." causes seg fault in 32bit build

4.124.4 Changes

Alex Smith (2):

- radv: Emit pending flushes before executing a secondary command buffer
- radv: Flush before copying with PKT3_WRITE_DATA in CmdUpdateBuffer

Bartosz Tomczyk (1):

- glsl: fix heap-buffer-overflow

Bas Nieuwenhuizen (8):

- radv: Pass CMASK alignment to application.
- radv: Pass DCC alignment to application.
- radv: Never try to create more than max_sets descriptor sets.
- radv: Reset emitted compute pipeline when calling secondary cmd buffer.
- radv: Only use PKT3_OCCLUSION_QUERY when it doesn't hang.
- radv: Use correct size for availability flag.
- radv: Disable HTILE for textures with multiple layers/levels.
- radv: Emit cache flushes before CP DMA.

Ben Crocker (3):

- gallivm: Improve debug output (V2)
- gallivm: Override getHostCPUName() "generic" w/ "pwr8" (v4)
- gallivm: Reenable PPC VSX (v3)

Brendan King (1):

- egl/dri3: implement query surface hook

Bruce Cherniak (1):

- swr: Prune empty nodes in CalculateProcessorTopology.

Connor Abbott (1):

- anv: fix Get*MemoryRequirements for !LLC

Dave Airlie (13):

- radv: program a default point size.
- radv: handle transfer_write as a dst flag.
- radv/ac: handle nir irem opcode.
- radv/ac: implement txs for buffer textures.
- radv/ac: correctly size shared memory usage.
- radv/ac: avoid the fmask path when doing txs.
- radv: pass FMASK alignment to application
- tgsi: fix memory leak in tgsi sanity check
- radv: fix depth format in blit2d.
- radv: fix txs for sampler buffers
- radv: drop Z24 support.
- radv: disable mip point pre clamping.
- radv: setup llvm target data layout

Emil Velikov (6):

- docs: add sha256 checksums for 13.0.5
- Revert "get-pick-list.sh: Require explicit "13.0" for nominating stable patches"
- cherry-ignore: don't pick nir_op_pack_double optimisation fix
- i965: move brw_define.h ifndef guard to the top
- cherry-ignore: add ANV fast clears related fixes
- Update version to 13.0.6

Fredrik Höglund (2):

- radv: fix the dynamic buffer index in vkCmdBindDescriptorSets
- radv/ac: fix multiple descriptor sets with dynamic buffers

George Kyriazis (1):

- swr: Align query results allocation

Grazvydas Ignotas (3):

- r300g: only allow byteswapped formats on big endian
- gallium/u_queue: fix a crash with atexit handlers
- gallium/u_queue: set num_threads correctly if not all threads start

Gregory Hainaut (1):

- glapi: fix typo in count_scale

Ian Romanick (1):

- mesa: Don't advertise GL_OES_read_format in core profile

Ilia Mirkin (8):

- nvc0: increase number of ubo binding points
- nvc0/ir: fix robustness guarantees for constbuf loads on kepler+ compute
- nvc0/ir: fix ubo max clamp, reset file index
- gm107/ir: fix address offset bitfield for ATOMS
- nvc0: set the render condition in the compute object
- st/mesa: don't pass compare mode for stencil-sampled textures
- nvc0: take extra pushbuf space into account for pushbuf_space calls
- nvc0: increase alignment to 256 for texture buffers on fermi

Jacob Lifshay (1):

- vulkan/wsi: Improve the DRI3 error message

Jason Ekstrand (11):

- i965: Use a better guardband calculation.
- intel/blorp: Swizzle clear colors on the CPU
- i965/fs: Remove the inline pack_double_2x32 optimization
- anv: Add an invalidate_range helper
- anv/query: clflush the bo map on non-LLC platforms
- genxml: Make MI_STORE_DATA_IMM more consistent
- anv/query: Perform CmdResetQueryPool on the GPU
- blorp/exec: Use uint32_t for copying varying data
- intel/blorp: Explicitly flush all allocated state
- anv: Accurately advertise dynamic descriptor limits
- anv: Properly handle destroying NULL devices and instances

Jonas Pfeil (1):

- ralloc: Make sure ralloc() allocations match malloc()'s alignment.

Jose Maria Casanova Crespo (1):

- glsl: non-last member unsized array on SSBO must fail compilation on GLSL ES 3.1

Kenneth Graunke (7):

- i965: Fix fast depth clears for surfaces with a dimension of 16384.

- i965: Use a UW source type for CS_OPCODE_CS_TERMINATE.
- i965: Fix check for negative pitch in can_do_fast_copy_blit().
- i965: Support the force_glsl_version driconf option.
- i965: Combine the Gen6 SF and Clip viewport atoms.
- mesa: Do (TCS && !TES) draw time validation in ES as well.
- egl: Ensure ResetNotificationStrategy matches for shared contexts.

Lionel Landwerlin (3):

- spirv: don't assert with location decorations on non i/o variables
- anv: wsi: report presentation error per image request
- i965/fs: fix uninitialized memory access

Marc Di Luzio (1):

- glsl: correct compute shader checks for memoryBarrier functions

Marek Olšák (10):

- st/mesa: destroy pipe_context before destroying st_context (v2)
- radeonsi: don't invoke DCC decompression in update_all_texture_descriptors
- radeonsi: fix UNSIGNED_BYTE index buffer fallback with non-zero start (v2)
- gallium/util: remove unused u_index_modify helpers
- gallium/u_index_modify: don't add PIPE_TRANSFER_UNSYNCHRONIZED unconditionally
- gallium/u_queue: fix random crashes when the app calls exit()
- st/mesa: reset sample_mask, min_sample, and render_condition for PBO ops
- st/mesa: set blend state for PBO readbacks
- radeonsi: fix broken tessellation on Carrizo and Stoney
- radeonsi: mark all bound shader buffer ranges as initialized

Matt Turner (1):

- clover: Work around build failure with AltiVec.

Nicolai Hähnle (12):

- mesa/main: fix meta caller of _mesa_ClampColor
- radeonsi: fix texture gather on stencil textures
- glsl: split DIV_TO_MUL_RCP into single- and double-precision flags
- glx/dri3: handle NULL pointers in loader-to-DRI3 drawable conversion
- glx/dri3: guard in_current_context against a disappeared drawable
- glx: guard swap-interval functions against destroyed drawables
- dri/common: clear the loaderPrivate pointer in driDestroyDrawable
- winsys/amdgpu: reduce max_alloc_size based on GTT limits
- radeonsi: handle MultiDrawIndirect in si_get_draw_start_count
- radeonsi: fix UINT/SINT clamping for 10-bit formats on <= CIK

- st/glsl_to_tgsi: avoid iterating past the head of the instruction list
- st/mesa: inform the driver of framebuffer changes before compute dispatches

Samuel Iglesias Gonsálvez (6):

- glsl: fix heap-use-after-free in ast_declarator_list::hir()
- i965/fs: mark last DF uniform array element as 64 bit live one
- i965/fs: detect different bit size accesses to uniforms to push them in proper locations
- i965/fs: fix indirect load DF uniforms on BSW/BXT
- i965/fs: fix source type when emitting MOV_INDIRECT to read ICP handles
- i965/fs: emit MOV_INDIRECT with the source with the right register type

Samuel Pitoiset (1):

- winsys/amdgpu: avoid potential segfault in amdgpu_bo_map()

4.125 Mesa 17.0.1 Release Notes / March 4, 2017

Mesa 17.0.1 is a bug fix release which fixes bugs found since the 17.0.0 release.

Mesa 17.0.1 implements the OpenGL 4.5 API, but the version reported by glGetString(GL_VERSION) or glGetIntegerv(GL_MAJOR_VERSION) / glGetIntegerv(GL_MINOR_VERSION) depends on the particular driver being used. Some drivers don't support all the features required in OpenGL 4.5. OpenGL 4.5 is **only** available if requested at context creation because compatibility contexts are not supported.

4.125.1 SHA256 checksums

```
e819bd3e515dac26faf9836d8f27a4ddf05323b9b23afb6c06536d4ac82e2743  mesa-17.0.1.tar.gz
96fd70ef5f31d276a17e424e7e1bb79447ccbbe822b56844213ef932e7ad1b0c  mesa-17.0.1.tar.xz
```

4.125.2 New features

None

4.125.3 Bug fixes

- Bug 98869 - Electronic Super Joy graphic artefacts (regression,bisected)
- Bug 99532 - Compute shader doesn't give right result under some circumstances
- Bug 99677 - heap-use-after-free in glsl
- Bug 99692 - [radv] Mostly broken on Hawaii PRO/CIK ASICs
- Bug 99850 - Tessellation bug on Carrizo

4.125.4 Changes

Bas Nieuwenhuizen (4):

- radv: Never try to create more than max_sets descriptor sets.
- radv: Reset emitted compute pipeline when calling secondary cmd buffer.
- radv: Only use PKT3_OCCLUSION_QUERY when it doesn't hang.
- radv: Use correct size for availability flag.

Ben Crocker (3):

- gallivm: Reenable PPC VSX (v3)
- gallivm: Improve debug output (V2)
- gallivm: Override getHostCPUName() "generic" w/ "pwr8" (v4)

Brendan King (1):

- egl/dri3: implement query surface hook

Christian Gmeiner (2):

- etnaviv: move pctx initialisation to avoid a null dereference
- etnaviv: remove number of pixel pipes validation

Connor Abbott (1):

- anv: fix Get*MemoryRequirements for !LLC

Daniel Stone (1):

- egl/wayland: Don't use DRM format codes for SHM

Dave Airlie (6):

- tgsi: fix memory leak in tgsi sanity check
- radv: change base aligmment for allocated memory.
- radv: fix cik macroModeIndex.
- radv: adopt some init config workarounds from radeonsi.
- radv: fix depth format in blit2d.
- radv: fix txs for sampler buffers

Emil Velikov (8):

- docs: add sha256 checksums for 17.0.0
- bin/get-extra-pick-list: use git merge-base to get the branchpoint
- bin/get-extra-pick-list: rework to use already_picked list
- bin/get-typod-pick-list.sh: limit 'git grep . . . ' to only as needed
- bin/get-pick-list.sh: limit 'git grep . . . ' only as needed
- bin/get-pick-list.sh: remove ancient way of nominating patches
- bin/get-fixes-pick-list.sh: add new script
- Update version to 17.0.1

Eric Anholt (1):

- vc4: Avoid emitting small immediates for UBO indirect load address guards.

Grazvydas Ignotas (3):

- r300g: only allow byteswapped formats on big endian
- gallium/u_queue: fix a crash with atexit handlers
- gallium/u_queue: set num_threads correctly if not all threads start

Hans de Goede (1):

- glx/glvnd: Fix GLXdispatchIndex sorting

Ilia Mirkin (4):

- gm107/ir: fix address offset bitfield for ATOMS
- nvc0: set the render condition in the compute object
- st/mesa: don't pass compare mode for stencil-sampled textures
- nvc0: disable linked tsc mode in compute launch descriptor

Jason Ekstrand (10):

- i965/sampler_state: Clamp min/max LOD to 14 on gen7+
- i965/sampler_state: Pass texObj into update_sampler_state
- i965/sampler_state: Set the "Base Mip Level" field on Sandy Bridge
- intel/blorp: Swizzle clear colors on the CPU
- i965/fs: Fix the inline nir_op_pack_double optimization
- anv: Add an invalidate_range helper
- anv/query: clflush the bo map on non-LLC platforms
- genxml: Make MI_STORE_DATA_IMM more consistent
- anv/query: Perform CmdResetQueryPool on the GPU
- intel/blorp: Explicitly flush all allocated state

Jose Maria Casanova Crespo (1):

- glsl: non-last member unsized array on SSBO must fail compilation on GLSL ES 3.1

Kenneth Graunke (1):

- mesa: Do (TCS && !TES) draw time validation in ES as well.

Leo Liu (1):

- configure.ac: check require_basic_egl only if egl enabled

Lionel Landwerlin (2):

- anv: wsi: report presentation error per image request
- i965/fs: fix uninitialized memory access

Marek Olšák (6):

- radeonsi: fix UNSIGNED_BYTE index buffer fallback with non-zero start (v2)
- gallium/util: remove unused u_index_modify helpers
- gallium/u_index_modify: don't add PIPE_TRANSFER_UNSYNCHRONIZED unconditionally

- gallium/u_queue: fix random crashes when the app calls exit()
- radeonsi: fix broken tessellation on Carrizo and Stoney
- amd/common: fix ASICREV_IS_POLARIS11_M for Polaris12

Mauro Rossi (2):

- android: radeonsi: fix sid_table.h generated header include path
- android: glsl: build shader cache sources

Michel Dänzer (1):

- configure.ac: Drop LLVM compiler flags more radically

Nicolai Hähnle (3):

- winsys/amdgpu: reduce max_alloc_size based on GTT limits
- radeonsi: handle MultiDrawIndirect in si_get_draw_start_count
- radeonsi: fix UINT/SINT clamping for 10-bit formats on <= CIK

Samuel Iglesias Gonsálvez (1):

- glsl: fix heap-use-after-free in ast_declarator_list::hir()

Tapani Pälli (1):

- android: fix droid_create_image_from_prime_fd_yuv for YV12

4.126 Mesa 13.0.5 Release Notes / February 20, 2017

Mesa 13.0.5 is a bug fix release which fixes bugs found since the 13.0.4 release.

Mesa 13.0.5 implements the OpenGL 4.4 API, but the version reported by glGetString(GL_VERSION) or glGetIntegerv(GL_MAJOR_VERSION) / glGetIntegerv(GL_MINOR_VERSION) depends on the particular driver being used. Some drivers don't support all the features required in OpenGL 4.4. OpenGL 4.4 is **only** available if requested at context creation because compatibility contexts are not supported.

4.126.1 SHA256 checksums

```
7e45e3812078726eabca6d9384364bf035a3c4279024ec9090dd1b19a8989926  mesa-13.0.5.tar.gz
bfcea7e2c801525a60895c8aff11aa68457ee9aa35d01a4638e1f310a3f5ef87  mesa-13.0.5.tar.xz
```

4.126.2 New features

None

4.126.3 Bug fixes

- Bug 98329 - [dEQP, EGL, SKL, BDW, BSW] dEQP-EGL.functional.image.render_multiple_contexts.gles2_renderbuffer_depth16
- Bug 98421 - src/loader/loader.c:111:40: error: unknown type name 'drmDevicePtr'
- Bug 98526 - glsl/tests/general-ir-test regression
- Bug 99532 - Compute shader doesn't give right result under some circumstances

- Bug 99631 - segfault with OSVRTrackerView and openscenegraph git master
- Bug 99633 - rasterizer/core/clip.h:279:49: error: 'const struct API_STATE' has no member named 'linkageCount'
- Bug 99692 - [radv] Mostly broken on Hawaii PRO/CIK ASICs

4.126.4 Changes

Bartosz Tomczyk (2):

- r600: Fix stack overflow
- r600/sb: Fix memory leak

Bruce Cherniak (1):

- swr: [rasterizer core] Remove dead code Clipper::ClipScalar()

Chad Versace (1):

- i965/mt: Disable HiZ when sharing depth buffer externally (v2)

Dave Airlie (3):

- radv: change base aligmment for allocated memory.
- radv: fix cik macroModeIndex.
- radv: adopt some init config workarounds from radeonsi.

Derek Foreman (1):

- egl/dri2: add image_loader_extension back into loader extensions for wayland

Emil Velikov (26):

- docs: add sha256 checksums for 13.0.4
- configure.ac: list radeon in –with-vulkan-drivers help string
- i965: automake: correctly set MKDIR_GEN
- freedreno: automake: correctly set MKDIR_GEN
- i965: automake: include builddir prior to srcdir
- i915: automake: include builddir prior to srcdir
- egl: automake: include builddir prior to srcdir
- clover: automake: include builddir prior to srcdir
- st/dri: automake: include builddir prior to srcdir
- d3dadapter9: automake: include builddir prior to srcdir
- glx: automake: include builddir prior to srcdir
- glx/apple: automake: include builddir prior to srcdir
- glx/windows: automake: include builddir prior to srcdir
- loader: automake: include builddir prior to srcdir
- mapi: automake: include builddir prior to srcdir
- radeon, r200: automake: include builddir prior to srcdir

- dri/swrast: automake: include builddir prior to srcdir
- dri/osmesa: automake: include builddir prior to srcdir
- mesa/tests: automake: include builddir prior to srcdir
- bin/get-extra-pick-list: use git merge-base to get the branchpoint
- bin/get-extra-pick-list: rework to use already_picked list
- bin/get-typod-pick-list.sh: limit 'git grep . . . ' to only as needed
- bin/get-pick-list.sh: limit 'git grep . . . ' only as needed
- bin/get-pick-list.sh: remove ancient way of nominating patches
- bin/get-fixes-pick-list.sh: add new script
- Update version to 13.0.5

Eric Anholt (1):

- vc4: Avoid emitting small immediates for UBO indirect load address guards.

Hans de Goede (1):

- glx/glvnd: Fix GLXdispatchIndex sorting

Ian Romanick (11):

- linker: Slight code rearrange to prevent duplication in the next commit
- linker: Accurately track gl_uniform_block::stageref
- glsl: Split process_block_array into two functions
- glsl: Fix wonkey indentation left from previous commit
- glsl: Track the linearized array index for each UBO instance array element
- glsl: Use simpler visitor to determine which UBO and SSBO blocks are used
- glsl: Add tracking for elements of an array-of-arrays that have been accessed
- glsl: Add structures to track accessed elements of a single array
- glsl: Mark a set of array elements as accessed using a list of array_deref_range
- glsl: Walk a list of ir_dereference_array to mark array elements as accessed
- linker: Accurately mark a uniform block instance array element as used in a stage

Ilia Mirkin (3):

- vbo: process buffer binding state changes on draw when recording
- st/mesa: MAX_VARYING is the max supported number of patch varyings, not min
- nvc0: disable linked tsc mode in compute launch descriptor

Jason Ekstrand (11):

- nir/search: Use the correct bit size for integer comparisons
- i965/blorp: Use the correct ISL format for combined depth/stencil
- intel/blorp: Handle clearing of A4B4G4R4 on all platforms
- isl/formats: Only advertise sampling for A4B4G4R4 on Broadwell
- anv: Flush render cache before STATE_BASE_ADDRESS on gen7

- anv: Improve flushing around STATE_BASE_ADDRESS
- vulkan/wsi/wayland: Handle VK_INCOMPLETE for GetFormats
- vulkan/wsi/wayland: Handle VK_INCOMPLETE for GetPresentModes
- vulkan/wsi: Lower the maximum image sizes
- i965/sampler_state: Pass texObj into update_sampler_state
- i965/sampler_state: Set the "Base Mip Level" field on Sandy Bridge

Kenneth Graunke (1):

- i965: Unbind deleted shaders from brw_context, fixing malloc heisenbug.

Lionel Landwerlin (5):

- anv: don't require render target isl bit for depth/stencil surfaces
- anv: set command buffer to NULL when allocations fail
- anv: fix descriptor pool internal size allocation
- spirv: handle OpUndef as part of the variable parsing pass
- spirv: handle undefined components for OpVectorShuffle

Marc-André Lureau (1):

- tgsi-dump: dump label if instruction has one

Marek Olšák (2):

- radeonsi: always set the TCL1_ACTION_ENA when invalidating L2
- gallium/radeon: fix performance of buffer readbacks

Topi Pohjolainen (2):

- i965: Make depth clear flushing more explicit
- i965/gen6: Issue direct depth stall and flush after depth clear

Vinson Lee (2):

- scons: Require libdrm >= 2.4.66 for DRM.
- util: Fix Clang trivial destructor check.

4.127 Mesa 17.0.0 Release Notes / February 13, 2017

Mesa 17.0.0 is a new development release. People who are concerned with stability and reliability should stick with a previous release or wait for Mesa 17.0.1.

Mesa 17.0.0 implements the OpenGL 4.5 API, but the version reported by glGetString(GL_VERSION) or glGetIntegerv(GL_MAJOR_VERSION) / glGetIntegerv(GL_MINOR_VERSION) depends on the particular driver being used. Some drivers don't support all the features required in OpenGL 4.5. OpenGL 4.5 is **only** available if requested at context creation because compatibility contexts are not supported.

4.127.1 SHA256 checksums

```
696578f0b83796470511a88a95fff15a2a25fa201a9e487716f2ca20c177c3ab  mesa-17.0.0.tar.gz
39db3d59700159add7f977307d12a7dfe016363e760ad82280ac4168ea668481  mesa-17.0.0.tar.xz
```

4.127.2 New features

Note: some of the new features are only available with certain drivers.

- GL_ARB_post_depth_coverage on i965/gen9+
- GL_KHR_blend_equation_advanced on nvc0
- GL_INTEL_conservative_rasterization on i965/gen9+
- GL_NV_image_formats on any driver supporting GL_ARB_shader_image_load_store (i965, nvc0, radeonsi, softpipe)
- GL_ARB_gpu_shader_fp64 in i965/haswell
- GL_ARB_vertex_attrib_64bit in i965/haswell
- GL_ARB_shader_precision in i965/haswell
- Intel Haswell now supports OpenGL 4.2
- GL_OES_geometry_shader on i965/haswell
- GL_OES_texture_cube_map_array on i965/haswell
- GL_OES_viewport_array on i965/haswell
- Vulkan Float64 capability support on Intel's ANV driver

4.127.3 Bug fixes

- Bug 70623 - libglx.so: undefined symbol: _glapi_tls_Context
- Bug 72902 - [IVB/HSW/BDW] DOTA2 segfaults unless Mesa is configured with (non-default) –enable-glx-tls
- Bug 73778 - _glapi_tls_Dispatch undefined
- Bug 77662 - Fail to render to different faces of depth-stencil cube map
- Bug 89043 - undefined symbol: _glapi_tls_Dispatch
- Bug 91281 - Tonga VCE 2160p encode fails with BO to small for addr
- Bug 92234 - [BDW] GPU hang in Shogun2
- Bug 92634 - gallium's vl_mpeg12_decoder does not work with st/va
- Bug 92760 - Add FP64 support to the i965 shader backends
- Bug 92925 - Incorrect GEN for ASTC in Surface Format Table
- Bug 93551 - Divinity: Original Sin Enhanced Edition(Native) crash on start
- Bug 94512 - X segfaults with glx-tls enabled in a x32 environment
- Bug 94900 - HD6950 GPU lockup loop with various steam games (octodad[always], saints row 4[always], dead island[always], grid autosport[sometimes])

- Bug 94904 - [vulkan, BSW] dEQP-VK.api.object_management.multithreaded_per_thread_device intermittent crash
- Bug 95460 - Please add more drivers (freedreno, virgl) to features.txt status document
- Bug 96959 - nop.sat generated by pow workaround?
- Bug 97102 - [dri][swr] stack overflow / infinite loop with GALLIUM_DRIVER=swr
- Bug 97232 - Line rendering broken in Dolphin when using gl_ClipDistance
- Bug 97287 - GL45-CTS.vertex_attrib_binding.basic-inputL-case1 fails
- Bug 97321 - Query INFO_LOG_LENGTH for empty info log should return 0
- Bug 97420 - "#version 0" crashes glsl_compiler
- Bug 97422 - trying to call a number as a function results into a crash
- Bug 97447 - GL 3.0 compatibility context exposes GL_ARB_compute_shader
- Bug 97473 - Memory corruption when uploading DXT5 cubemap faces
- Bug 97715 - [ILK,G45,G965] piglit.spec.arb_separate_shader_objects.misc api error checks
- Bug 97779 - [regression, bisected][BDW, GPU hang] stuck on render ring, always reproducible
- Bug 97804 - Later precision statement isn't overriding earlier one
- Bug 97952 - /usr/include/string.h:518:12: error: exception specification in declaration does not match previous declaration
- Bug 97967 - glsl/tests/cache-test regression
- Bug 98005 - VCE dual instance encoding inconsistent since st/va: enable dual instances encode by sync surface
- Bug 98012 - [IVB] Segfault when running Dolphin twice with Vulkan
- Bug 98134 - dEQP-GLES31.functional.debug.negative_coverage.get_error.buffer.draw_buffers wants a different GL error code
- Bug 98172 - Concurrent call to glClientWaitSync results in segfault in one of the waiters.
- Bug 98238 - witcher 2: objects are black when changing lod
- Bug 98243 - dEQP mismatched UBO precision qualifiers
- Bug 98245 - GLES3.1 link negative dEQP "expected linking to fail, but passed."
- Bug 98250 - dEQP-GLES31.functional.debug.negative_coverage.get_error.texture.texparameterIiv/texparameterIuiv failure
- Bug 98263 - [radv] The Talos Principle fails to launch with "Fatal error: Cannot set display mode."
- Bug 98297 - Can't configure a desktop with 3x4k monitors in one row
- Bug 98299 - Compute shaders generate stupid divides
- Bug 98307 - "st/glsl_to_tgsi: explicitly track all input and output declaration" broke flightgear colors on rs780
- Bug 98326 - [dEQP, EGL] pbuffer depth/stencil tests fail
- Bug 98327 - [dEQP, EGL] dEQP-EGL.functional.resize not supported
- Bug 98328 - [dEQP, EGL] luminance tests fail
- Bug 98329 - [dEQP, EGL, SKL, BDW, BSW] dEQP-EGL.functional.image.render_multiple_contexts.gles2_renderbuffer_depth16
- Bug 98330 - [dEQP, EGL] dEQP-EGL.functional.buffer_age.no_preserve fails

- Bug 98339 - dEQP-EGL: Got EGL_BAD_MATCH: eglCreateSyncKHR()
- Bug 98343 - dEQP-EGL: GL_INVALID_ENUM at teglCreateContextExtTests
- Bug 98415 - Vulkan Driver JSON file contains incorrect field
- Bug 98421 - src/loader/loader.c:111:40: error: unknown type name 'drmDevicePtr'
- Bug 98431 - UnrealEngine v4 demos startup fails to blorp blit assert
- Bug 98480 - Support R8 image texture in ES 3.1
- Bug 98512 - radeon r600 vdpau: Invalid command stream: texture bo too small
- Bug 98518 - [r600g, bisected] regression: NI/Turks MSAA texture corruption with FreeCAD and Wine games
- Bug 98526 - glsl/tests/general-ir-test regression
- Bug 98595 - glsl: ralloc assertion "info->canary == CANARY" failed
- Bug 98599 - xterm menus corrupt since tgsi/scan: handle indirect image indexing correctly
- Bug 98632 - Fix build on Hurd without PATH_MAX
- Bug 98681 - ir_builder_print_visitor.cpp:401:67: error: expected ')' before 'PRIx64'
- Bug 98694 - "(5=2)?1:1" as array size decleration crashes glsl_compiler
- Bug 98740 - bitcode.cpp:102:8: error: 'Error' is not a member of 'llvm'
- Bug 98767 - [swrast] ralloc.c:84: get_header: Assertion 'info->canary == CANARY' failed.
- Bug 98774 - glsl/tests/warnings-test regression
- Bug 98815 - [SKL/BDW GT2] large perf regression in TessMark
- Bug 98840 - nir clone test fails
- Bug 98893 - [SKL] piglit.spec.arb_shader_image_load_store.semantics intermittent
- Bug 98914 - mesa-vdpau-drivers: breaks vdpau for mpeg2video
- Bug 98917 - [BDW SKL BSW KBL] Tessellation CTS tests regression
- Bug 98975 - Wasteland 2 Directors Cut: Hangs. GPU fault
- Bug 99010 - –disable-gallium-llvm no longer recognized
- Bug 99013 - [regression, bisected] radeonsi: commit 4c8c13b3 "Use amdgcn intrinsics for fs interpolation" makes system unusable
- Bug 99030 - [HSW, regression] transform feedback fails on Linux 4.8
- Bug 99038 - [dEQP, EGL, SKL, BDW, BSW] dEQP-EGL.functional.negative_api.create_pixmap_surface crashes
- Bug 99072 - [byt,ivb,snb] ES3-CTS.gtf.GL3Tests.shadow regression
- Bug 99085 - [EGL] dEQP-EGL.functional.sharing.gles2.multithread intermittent
- Bug 99097 - [vulkancts] dEQP-VK.image.store regression
- Bug 99100 - [SKL,BDW,BSW,KBL] dEQP-VK.glsl.return.return_in_dynamic_loop_dynamic_vertex regression
- Bug 99119 - swr_fence_work.cpp(42): error: argument of type "std::nullptr_t" is incompatible with parameter of type "unsigned long"
- Bug 99144 - Incorrect rendering using glDrawArraysInstancedBaseInstance and first != 0 on Skylake

- Bug 99154 - Link time error when using multiple builtin functions
- Bug 99158 - vdpau segfaults and gpu locks with kodi on R9285
- Bug 99185 - dEQP-EGL.functional.image.modify.tex_rgb5_a1_tex_subimage_rgba8
- Bug 99188 - dEQP-EGL.functional.create_context_ext.robust_gl_30.rgb565_no_depth_no_stencil
- Bug 99210 - ES3-CTS.functional.texture.mipmap.cube.generate.rgba5551_*
- Bug 99214 - Crash in library libswrAVX.so when assigning vertex buffer object pointers with elements of type GL_DOUBLE
- Bug 99219 - The Stanley Parable GPU hang when starting a new game
- Bug 99229 - [G33] thousands of tests crash
- Bug 99231 - [HSW][i965] Crash in upload_3dstate_streamout()
- Bug 99287 - piglit.spec.glsl-1_10.execution.vs-nested-return-sibling-loop regression
- Bug 99303 - [REGRESSION][BISECTED] DMs are crashing on start with "radeon"
- Bug 99314 - [g33] glsl regressions
- Bug 99339 - Blender line rendering broken after removing XY clipping of lines
- Bug 99354 - [G71] "Assertion 'bkref' failed" reproducible with glmark2
- Bug 99389 - Mesa build broken: sid_tables.h
- Bug 99391 - [ILK,G45,G965] piglit regressions
- Bug 99401 - [g33] regression: piglit.spec.!opengl 1_0.gl-1_0-beginend-coverage
- Bug 99419 - Crash(Segmentation fault) si_shader_select in Master Of Orion
- Bug 99450 - [amdgpu] Payday 2 visual glitches on some models
- Bug 99451 - polygon offset use after free
- Bug 99456 - Firefox crashing when opening about:support with WebGL2 enabled
- Bug 99631 - segfault with OSVRTrackerView and openscenegraph git master
- Bug 99633 - rasterizer/core/clip.h:279:49: error: 'const struct API_STATE' has no member named 'linkageCount'
- Bug 99637 - VLC video has corrupted colors when using VDPAU output on Radeon SI

4.127.4 Changes

- Building RADV requires –enable-gallium-llvm
- The vulkan headers vk_platform.h and vulkan.h are no longer installed
- The configure options –with-sha1 and –disable-shader-cache are removed alongside their respective library requirements

4.128 Mesa 13.0.4 Release Notes / February 1, 2017

Mesa 13.0.4 is a bug fix release which fixes bugs found since the 13.0.3 release.

Mesa 13.0.4 implements the OpenGL 4.4 API, but the version reported by glGetString(GL_VERSION) or glGetIntegerv(GL_MAJOR_VERSION) / glGetIntegerv(GL_MINOR_VERSION) depends on the particular driver being used. Some drivers don't support all the features required in OpenGL 4.4. OpenGL 4.4 is **only** available if requested at context creation because compatibility contexts are not supported.

4.128.1 SHA256 checksums

```
a78518030b0b7d77a6c426ac3ff40f4b27fb0e2cdb0dfbe685024a46cae59bad  mesa-13.0.4.tar.gz
a95d7ce8f7bd5f88585e4be3144a341236d8c0fc91f6feaec59bb8ba3120e726  mesa-13.0.4.tar.xz
```

4.128.2 New features

None

4.128.3 Bug fixes

- Bug 92634 - gallium's vl_mpeg12_decoder does not work with st/va
- Bug 94512 - X segfaults with glx-tls enabled in a x32 environment
- Bug 94900 - HD6950 GPU lockup loop with various steam games (octodad[always], saints row 4[always], dead island[always], grid autosport[sometimes])
- Bug 98263 - [radv] The Talos Principle fails to launch with "Fatal error: Cannot set display mode."
- Bug 98914 - mesa-vdpau-drivers: breaks vdpau for mpeg2video
- Bug 98975 - Wasteland 2 Directors Cut: Hangs. GPU fault
- Bug 99030 - [HSW, regression] transform feedback fails on Linux 4.8
- Bug 99085 - [EGL] dEQP-EGL.functional.sharing.gles2.multithread intermittent
- Bug 99097 - [vulkancts] dEQP-VK.image.store regression
- Bug 99100 - [SKL,BDW,BSW,KBL] dEQP-VK.glsl.return.return_in_dynamic_loop_dynamic_vertex regression
- Bug 99144 - Incorrect rendering using glDrawArraysInstancedBaseInstance and first != 0 on Skylake
- Bug 99154 - Link time error when using multiple builtin functions
- Bug 99158 - vdpau segfaults and gpu locks with kodi on R9285
- Bug 99185 - dEQP-EGL.functional.image.modify.tex_rgb5_a1_tex_subimage_rgba8
- Bug 99188 - dEQP-EGL.functional.create_context_ext.robust_gl_30.rgb565_no_depth_no_stencil
- Bug 99210 - ES3-CTS.functional.texture.mipmap.cube.generate.rgba5551_*
- Bug 99354 - [G71] "Assertion 'bkref' failed" reproducible with glmark2
- Bug 99450 - [amdgpu] Payday 2 visual glitches on some models
- Bug 99451 - polygon offset use after free

4.128.4 Changes

Andres Rodriguez (2):

- vulkan/wsi: clarify the severity of lack of DRI3 v2
- radv: fix include order for installed headers v2

Arda Coskunses (2):

- vulkan/wsi/x11: don't crash on null visual
- vulkan/wsi/x11: don't crash on null wsi x11 connection

Bas Nieuwenhuizen (1):

- radv: Support loader interface version 3.

Chad Versace (10):

- egl: Check config's surface types in eglCreate*Surface()
- dri: Add __DRI_IMAGE_FORMAT_ARGB1555
- mesa/texformat: Handle GL_RGBA + GL_UNSIGNED_SHORT_5_5_5_1
- egl: Emit correct error when robust context creation fails
- anv: Handle vkGetPhysicalDeviceQueueFamilyProperties with count == 0
- mesa/shaderobj: Fix races on refcounts
- meta: Disable dithering during glGenerateMipmap
- vulkan: Add new cast macros for VkIcd types
- vulkan: Update vk_icd.h to interface version 3
- anv: Support loader interface version 3 (patch v2)

Christian König (1):

- vl/zscan: fix "Fix trivial sign compare warnings"

Chuck Atkins (1):

- glx: Add missing glproto dependency for gallium-xlib glx

Damien Grassart (1):

- anv: return count of queue families written

Dave Airlie (1):

- radv: flush smem for uniform buffer bit.

Emil Velikov (10):

- docs: add sha256 checksums for 13.0.3
- cherry-ignore: add couple of intel_miptree_copy related patches
- cherry-ignore: add radv: Call nir_lower_constant_initializers."
- get-typod-pick-list.sh: add new script
- cherry-ignore: add "_mesa_ClampColor extension/version fix"
- cherry-ignore: add wayland race condition fix
- egl/wayland: use the destroy_window_callback for swrast

- automake: use shared llvm libs for make distcheck
- get-pick-list.sh: Require explicit "13.0" for nominating stable patches
- Update version to 13.0.4

Francisco Jerez (1):

- anv: Fix uniform and storage buffer offset alignment limits.

Fredrik Höglund (2):

- radv: fix dual source blending
- dri3: Fix MakeCurrent without a default framebuffer

Grazvydas Ignotas (1):

- mapi: update the asm code to support x32

Heiko Przybyl (1):

- r600/sb: Fix loop optimization related hangs on eg

Ilia Mirkin (1):

- nouveau: take extra push space into account for pushbuf_space calls

Jason Ekstrand (4):

- i965/generator/tex: Handle an immediate sampler with an indirect texture
- anv/formats: Use the real format for B4G4R4A4_UNORM_PACK16 on gen8
- nir/search: Only allow matching SSA values
- isl: Mark A4B4G4R4_UNORM as supported on gen8

Jonas Ådahl (1):

- egl/wayland: Cleanup private display connection when init fails

Kenneth Graunke (7):

- i965: Don't bail on vertex element processing if we need draw params.
- i965: Fix last slot calculations
- i965: Fix texturing in the vec4 TCS and GS backends.
- spirv: Move cursor before calling vtn_ssa_value() in phi 2nd pass.
- i965: Make BLORP disable the NP Z PMA stall fix.
- glsl: Use ir_var_temporary when generating inline functions.
- i965: Properly flush in hsw_pause_transform_feedback().

Marek Olšák (4):

- vdpau: call texture_get_handle while the mutex is being held
- va: call texture_get_handle while the mutex is being held
- radeonsi: for the tess barrier, only use emit_waitcnt on SI and LLVM 3.9+
- radeonsi: don't forget to add HTILE to the buffer list for texturing

Michel Dänzer (1):

- cso: Don't restore nr_samplers in cso_restore_fragment_samplers

Nanley Chery (3):

- anv/cmd_buffer: Fix arrayed depth/stencil attachments
- anv/cmd_buffer: Fix programmed HiZ qpitch
- anv/image: Disable HiZ for depth buffer arrays

Nayan Deshmukh (1):

- st/va: delay calling begin_frame until we have all parameters

Rob Clark (1):

- freedreno: some fence cleanup

Samuel Pitoiset (1):

- gallium/hud: add missing break in hud_cpufreq_graph_install()

Timothy Arceri (3):

- nir: Turn imov/fmov of undef into undef
- glsl: fix opt_minmax redundancy checks against baserange
- util: fix list_is_singular()

Zachary Michaels (1):

- radeonsi: Always leave poly_offset in a valid state

4.129 Mesa 12.0.6 Release Notes / January 23, 2017

Mesa 12.0.6 is a bug fix release which fixes bugs found since the 12.0.5 release.

Mesa 12.0.6 implements the OpenGL 4.3 API, but the version reported by glGetString(GL_VERSION) or glGetIntegerv(GL_MAJOR_VERSION) / glGetIntegerv(GL_MINOR_VERSION) depends on the particular driver being used. Some drivers don't support all the features required in OpenGL 4.3. OpenGL 4.3 is **only** available if requested at context creation because compatibility contexts are not supported.

4.129.1 SHA256 checksums

```
65339ba5d76a45225b8b56f9a1da9db15c569e1d163760faa2921da0a8461741  mesa-12.0.6.tar.gz
7d6da9744c1022a4c2ab6ad01a206984d00443fb691568011d01b3dd97e36448  mesa-12.0.6.tar.xz
```

4.129.2 New features

None

4.129.3 Bug fixes

This list is likely incomplete.

- Bug 92234 - [BDW] GPU hang in Shogun2
- Bug 95130 - Derivatives of gl_Color wrong when helper pixels used
- Bug 98329 - [dEQP, EGL, SKL, BDW, BSW] dEQP-EGL.functional.image.render_multiple_contexts.gles2_renderbuffer_depth16

- Bug 99030 - [HSW, regression] transform feedback fails on Linux 4.8
- Bug 99354 - [G71] "Assertion 'bkref' failed" reproducible with glmark2

4.129.4 Changes

Chad Versace (3):

- i965/mt: Disable aux surfaces after making miptree shareable
- i965/mt: Disable HiZ when sharing depth buffer externally (v2)
- anv: Handle vkGetPhysicalDeviceQueueFamilyProperties with count == 0

Emil Velikov (5):

- docs: add sha256 checksums for 12.0.5
- get-typod-pick-list.sh: add new script
- automake: use shared llvm libs for make distcheck
- egl/wayland: use the destroy_window_callback for swrast
- Update version to 12.0.6

Fredrik Höglund (1):

- dri3: Fix MakeCurrent without a default framebuffer

Ilia Mirkin (1):

- nouveau: take extra push space into account for pushbuf_space calls

Jason Ekstrand (19):

- spirv/nir: Fix some texture opcode asserts
- spirv/nir: Add support for shadow samplers that return vec4
- spirv/nir: Properly handle gather components
- anv/pipeline: Set binding_table.gather_texture_start
- nir: Add a helper for determining the type of a texture source
- nir/lower_tex: Add some helpers for working with tex sources
- nir/lower_tex: Add support for lowering coordinate offsets
- i965/nir: Enable NIR lowering of txf and rect offsets
- i965: Get rid of the do_lower_unnormalized_offsets pass
- spirv/nir: Don't increment coord_components for array lod queries
- anv/image: Assert that the image format is actually supported
- spirv/nir: Move opcode selection higher up in handle_texture
- spirv/nir: Refactor type handling in handle_texture
- nir/spirv: Refactor coordinate handling in handle_texture
- spirv/nir: Handle texture projectors
- spirv/nir: Add support for ImageQuerySamples
- anv/device: Return the right error for failed maps

- anv/device: Implicitly unmap memory objects in FreeMemory
- anv/descriptor_set: Write the state offset in the surface state free list.

Kenneth Graunke (2):

- spirv: Move cursor before calling vtn_ssa_value() in phi 2nd pass.
- i965: Properly flush in hsw_pause_transform_feedback().

Marek Olšák (6):

- cso: don't release sampler states that are bound
- radeonsi: always restore sampler states when unbinding sampler views
- radeonsi: fix incorrect FMASK checking in bind_sampler_states
- radeonsi: disable CE on SI + AMDGPU
- radeonsi: disable the constant engine (CE) on Carrizo and Stoney
- gallium/radeon: fix the draw-calls HUD query

Matt Turner (3):

- i965/fs: Rename opt_copy_propagate -> opt_copy_propagation.
- i965/fs: Add unit tests for copy propagation pass.
- i965/fs: Reject copy propagation into SEL if not min/max.

Michel Dänzer (1):

- cso: Don't restore nr_samplers in cso_restore_fragment_samplers

Nicolai Hähnle (1):

- radeonsi: enable WQM in PS prolog when needed

4.130 Mesa 13.0.3 Release Notes / January 5, 2017

Mesa 13.0.3 is a bug fix release which fixes bugs found since the 13.0.2 release.

Mesa 13.0.3 implements the OpenGL 4.4 API, but the version reported by glGetString(GL_VERSION) or glGetIntegerv(GL_MAJOR_VERSION) / glGetIntegerv(GL_MINOR_VERSION) depends on the particular driver being used. Some drivers don't support all the features required in OpenGL 4.4. OpenGL 4.4 is **only** available if requested at context creation because compatibility contexts are not supported.

4.130.1 SHA256 checksums

```
55b07d056f9b855ba9d7c8b2ddc7d3b220a61c6ab1bdc73cbfc2f607721094c2  mesa-13.0.3.tar.gz
d9aa8be5c176d00d0cd503cb2f64a5a403ea471ec819c022581414860d7ba40e  mesa-13.0.3.tar.xz
```

4.130.2 New features

None

4.130.3 Bug fixes

- Bug 77662 - Fail to render to different faces of depth-stencil cube map
- Bug 92234 - [BDW] GPU hang in Shogun2
- Bug 98329 - [dEQP, EGL, SKL, BDW, BSW] dEQP-EGL.functional.image.render_multiple_contexts.gles2_renderbuffer_depth16
- Bug 99038 - [dEQP, EGL, SKL, BDW, BSW] dEQP-EGL.functional.negative_api.create_pixmap_surface crashes

4.130.4 Changes

Chad Versace (2):

- i965/mt: Disable aux surfaces after making miptree shareable
- egl: Fix crashes in eglCreate*Surface()

Dave Airlie (4):

- anv: set maxFragmentDualSrcAttachments to 1
- radv: set maxFragmentDualSrcAttachments to 1
- radv: fix another regression since shadow fixes.
- radv: add missing license file to radv_meta_bufimage.

Emil Velikov (5):

- docs: add sha256 checksums for 13.0.2
- anv: don't double-close the same fd
- anv: don't leak memory if anv_init_wsi() fails
- radv: don't leak the fd if radv_physical_device_init() succeeds
- Update version to 13.0.3

Eric Anholt (1):

- vc4: In a loop break/continue, jump if everyone has taken the path.

Gwan-gyeong Mun (3):

- anv: Add missing error-checking to anv_block_pool_init (v2)
- anv: Update the teardown in reverse order of the anv_CreateDevice
- vulkan/wsi: Fix resource leak in success path of wsi_queue_init()

Haixia Shi (1):

- compiler/glsl: fix precision problem of tanh

Ilia Mirkin (1):

- mesa: only verify that enabled arrays have backing buffers

Jason Ekstrand (8):

- anv/cmd_buffer: Re-emit MEDIA_CURBE_LOAD when CS push constants are dirty
- anv/image: Rename hiz_surface to aux_surface
- anv/cmd_buffer: Remove the 1-D case from the HiZ QPitch calculation

- genxml/gen9: Change the default of MI_SEMAPHORE_WAIT::RegisterPoleMode
- anv/device: Return the right error for failed maps
- anv/device: Implicitly unmap memory objects in FreeMemory
- anv/descriptor_set: Write the state offset in the surface state free list.
- spirv: Use a simpler and more correct implementaiton of tanh()

Kenneth Graunke (1):

- i965: Allocate at least some URB space even when max_vertices = 0.

Marek Olšák (17):

- radeonsi: always set all blend registers
- radeonsi: set CB_BLEND1_CONTROL.ENABLE for dual source blending
- radeonsi: disable RB+ blend optimizations for dual source blending
- radeonsi: consolidate max-work-group-size computation
- radeonsi: apply a multi-wave workgroup SPI bug workaround to affected CIK chips
- radeonsi: apply a TC L1 write corruption workaround for SI
- radeonsi: apply a tessellation bug workaround for SI
- radeonsi: add a tess+GS hang workaround for VI dGPUs
- radeonsi: apply the double EVENT_WRITE_EOP workaround to VI as well
- cso: don't release sampler states that are bound
- radeonsi: always restore sampler states when unbinding sampler views
- radeonsi: fix incorrect FMASK checking in bind_sampler_states
- radeonsi: allow specifying simm16 of emit_waitcnt at call sites
- radeonsi: wait for outstanding memory instructions in TCS barriers
- tgsi: fix the src type of TGSI_OPCODE_MEMBAR
- radeonsi: wait for outstanding LDS instructions in memory barriers if needed
- radeonsi: disable the constant engine (CE) on Carrizo and Stoney

Matt Turner (3):

- i965/fs: Rename opt_copy_propagate -> opt_copy_propagation.
- i965/fs: Add unit tests for copy propagation pass.
- i965/fs: Reject copy propagation into SEL if not min/max.

Nanley Chery (1):

- mesa/fbobject: Update CubeMapFace when reusing textures

Nicolai Hähnle (4):

- radeonsi: fix isolines tess factor writes to control ring
- radeonsi: update all GSVS ring descriptors for new buffer allocations
- radeonsi: do not kill GS with memory writes
- radeonsi: fix an off-by-one error in the bounds check for max_vertices

Rhys Kidd (1):

- glsl: Add pthread libs to cache_test

Timothy Arceri (2):

- mesa: fix active subroutine uniforms properly
- Revert "nir: Turn imov/fmov of undef into undef."

4.131 Mesa 12.0.5 Release Notes / December 5, 2016

Mesa 12.0.5 is a bug fix release which fixes bugs found since the 12.0.5 release.

Mesa 12.0.5 implements the OpenGL 4.3 API, but the version reported by glGetString(GL_VERSION) or glGetIntegerv(GL_MAJOR_VERSION) / glGetIntegerv(GL_MINOR_VERSION) depends on the particular driver being used. Some drivers don't support all the features required in OpenGL 4.3. OpenGL 4.3 is **only** available if requested at context creation because compatibility contexts are not supported.

4.131.1 SHA256 checksums

```
44d08a27d98bfeacd864381189e434d98afbf451689d01f80380dc1d66450e5b  mesa-12.0.5.tar.gz
2b0a972d8282860a11291c09c3ef01ac45171405951eb21a83c45ed2b4321924  mesa-12.0.5.tar.xz
```

4.131.2 New features

None

4.131.3 Bug fixes

This list is likely incomplete.

- Bug 77662 - Fail to render to different faces of depth-stencil cube map
- Bug 97779 - [regression, bisected][BDW, GPU hang] stuck on render ring, always reproducible
- Bug 98415 - Vulkan Driver JSON file contains incorrect field

4.131.4 Changes

Adam Jackson (2):

- glx/glvnd: Don't modify the dummy slot in the dispatch table
- glx/glvnd: Fix dispatch function names and indices

Anuj Phogat (1):

- i965: Fix GPU hang related to multiple render targets and alpha testing

Emil Velikov (4):

- docs: add release notes for 12.0.4
- docs: add sha256 checksums for 12.0.4

- cherry-ignore: add reverted LLVM_LIBDIR patch
- Update version to 12.0.5

Haixia Shi (1):

- mesa: change state query return value for RGB565

Jason Ekstrand (3):

- i965/fs/generator: Don't use the address immediate for MOV_INDIRECT
- anv/cmd_buffer: Take a command buffer instead of a batch in two helpers
- anv/cmd_buffer: Enable a CS stall workaround for Sky Lake gt4

Kenneth Graunke (1):

- intel: Fix pixel shader scratch space allocation on Gen9+ platforms.

Marek Olšák (13):

- gallium/radeon: fix behavior of GLSL findLSB(0)
- gallium/radeon: make sure HTILE address is aligned properly
- radeonsi: fix an assertion failure in si_decompress_sampler_color_textures
- gallium/radeon: unify viewport emission code
- gallium/radeon: set VPORT_ZMIN/MAX registers correctly
- radeonsi: fix gl_PatchVerticesIn for tessellation evaluation shader
- radeonsi: fix a crash in imageSize for cubemap arrays
- radeonsi: emit TA_CS_BC_BASE_ADDR on SI only if the kernel allows it
- gallium/radeon: add support for sharing textures with DCC between processes
- radeonsi: always set all blend registers
- radeonsi: set CB_BLEND1_CONTROL.ENABLE for dual source blending
- radeonsi: disable RB+ blend optimizations for dual source blending
- radeonsi: silence runtime warnings with LLVM 3.9

Matt Turner (1):

- anv: Replace "abi_versions" with correct "api_version".

Nanley Chery (1):

- mesa/fbobject: Update CubeMapFace when reusing textures

Steinar H. Gunderson (1):

- Fix races during _mesa_HashWalk().

Tim Rowley (3):

- swr: [rasterizer jitter] cleanup supporting different llvm versions
- swr: [rasterizer jitter] fix llvm-3.7 compile
- swr: [rasterizer] add support for llvm-3.9

4.132 Mesa 13.0.2 Release Notes / November 28, 2016

Mesa 13.0.2 is a bug fix release which fixes bugs found since the 13.0.1 release.

Mesa 13.0.2 implements the OpenGL 4.4 API, but the version reported by glGetString(GL_VERSION) or glGetIntegerv(GL_MAJOR_VERSION) / glGetIntegerv(GL_MINOR_VERSION) depends on the particular driver being used. Some drivers don't support all the features required in OpenGL 4.4. OpenGL 4.4 is **only** available if requested at context creation because compatibility contexts are not supported.

4.132.1 SHA256 checksums

```
6014233a5db6032ab8de4881384871bbe029de684502707794ce7b3e6beec308  mesa-13.0.2.tar.gz
a6ed622645f4ed61da418bf65adde5bcc4bb79023c36ba7d6b45b389da4416d5  mesa-13.0.2.tar.xz
```

4.132.2 New features

None

4.132.3 Bug fixes

- Bug 97321 - Query INFO_LOG_LENGTH for empty info log should return 0
- Bug 97420 - "#version 0" crashes glsl_compiler
- Bug 98632 - Fix build on Hurd without PATH_MAX

4.132.4 Changes

Ben Widawsky (3):

- i965: Add some APL and KBL SKU strings
- i965: Reorder PCI ID list to match release order
- i965/glk: Add basic Geminilake support

Dave Airlie (14):

- radv: fix texturesamples to handle single sample case
- wsi: fix VK_INCOMPLETE for vkGetSwapchainImagesKHR
- radv: don't crash on null swapchain destroy.
- ac/nir/llvm: fix channel in texture gather lowering code.
- radv: make sure to flush input attachments correctly.
- radv: fix image view creation for depth and stencil only
- radv: spir-v allows texture size query with and without lod.
- vulkan/wsi/x11: handle timeouts properly in next image acquire (v1.1)
- vulkan/wsi: store present mode in swapchain base class
- vulkan/wsi/x11: add support for IMMEDIATE present mode

- radv: fix texel fetch offset with 2d arrays.
- radv/si: fix optimal micro tile selection
- radv/ac/llvm: shadow samplers only return one value.
- radv: fix 3D clears with baseMiplevel

Eduardo Lima Mitev (2):

- vulkan/wsi/x11: Fix behavior of vkGetPhysicalDeviceSurfaceFormatsKHR
- vulkan/wsi/x11: Fix behavior of vkGetPhysicalDeviceSurfacePresentModesKHR

Emil Velikov (5):

- docs: add sha256 checksums for 13.0.1
- cherry-ignore: add reverted LLVM_LIBDIR patch
- anv: fix enumeration of properties
- radv: honour the number of properties available
- Update version to 13.0.2

Eric Anholt (3):

- vc4: Don't abort when a shader compile fails.
- vc4: Clamp the shadow comparison value.
- vc4: Fix register class handling of DDX/DDY arguments.

Gwan-gyeong Mun (2):

- util/disk_cache: close a previously opened handle in disk_cache_put (v2)
- anv: Fix unintentional integer overflow in anv_CreateDmaBufImageINTEL

Iago Toral Quiroga (1):

- anv/format: handle unsupported formats properly

Ian Romanick (2):

- glcpp: Handle '#version 0' and other invalid values
- glsl: Parse 0 as a preprocessor INTCONSTANT

Jason Ekstrand (15):

- anv/gen8: Stall when needed in Cmd(Set|Reset)Event
- anv/wsi: Set the fence to signaled in AcquireNextImageKHR
- anv: Rework fences
- vulkan/wsi/wayland: Include pthread.h
- vulkan/wsi/wayland: Clean up some error handling paths
- vulkan/wsi: Report the correct min/maxImageCount
- i965/gs: Allow primitive id to be a system value
- anv: Handle null in all destructors
- anv/fence: Handle ANV_FENCE_CREATE_SIGNALED_BIT
- nir/spirv: Fix handling of gl_PrimitiveId

- anv/blorp: Ignore clears for attachments first used as resolve destinations
- anv: Implement a depth stall restriction on gen7
- anv/cmd_buffer: Handle running out of binding tables in compute shaders
- anv/cmd_buffer: Emit a CS stall before setting a CS pipeline
- vulkan/wsi/x11: Implement FIFO mode.

Jordan Justen (2):

- isl: Fix height calculation in isl_msaa_interleaved_scale_px_to_sa
- i965/hsw: Set integer mode in sampling state for stencil texturing

Kenneth Graunke (4):

- intel: Set min_ds_entries on Broxton.
- i965: Fix compute shader crash.
- mesa: Drop PATH_MAX usage.
- i965: Fix GS push inputs with enhanced layouts.

Kevin Strasser (1):

- vulkan/wsi: Add a thread-safe queue implementation

Lionel Landwerlin (1):

- anv: fix multi level clears with VK_REMAINING_MIP_LEVELS

Lucas Stach (1):

- gbm: request correct version of the DRI2_FENCE extension

Nicolai Hähnle (2):

- radeonsi: store group_size_variable in struct si_compute
- glsl/lower_output_reads: fix geometry shader output handling with conditional emit

Steinar H. Gunderson (1):

- Fix races during _mesa_HashWalk().

Tapani Pälli (1):

- mesa: fix empty program log length

4.133 Mesa 13.0.1 Release Notes / November 14, 2016

Mesa 13.0.1 is a bug fix release which fixes bugs found since the 13.0.0 release.

Mesa 13.0.1 implements the OpenGL 4.4 API, but the version reported by glGetString(GL_VERSION) or glGetIntegerv(GL_MAJOR_VERSION) / glGetIntegerv(GL_MINOR_VERSION) depends on the particular driver being used. Some drivers don't support all the features required in OpenGL 4.4. OpenGL 4.4 is **only** available if requested at context creation because compatibility contexts are not supported.

4.133.1 SHA256 checksums

```
7cbb91dead05cde279ee95f86e8321c8e1c8fc9deb88f12e0f587672a10d88c5  mesa-13.0.1.tar.gz
71962fb2bf77d33b0ad4a565b490dbbeaf4619099c6d9722f04a73187957a731  mesa-13.0.1.tar.xz
```

4.133.2 New features

None

4.133.3 Bug fixes

- Bug 97715 - [ILK,G45,G965] piglit.spec.arb_separate_shader_objects.misc api error checks
- Bug 98012 - [IVB] Segfault when running Dolphin twice with Vulkan
- Bug 98512 - radeon r600 vdpau: Invalid command stream: texture bo too small

4.133.4 Changes

Adam Jackson (2):

- glx/glvnd: Don't modify the dummy slot in the dispatch table
- glx/glvnd: Fix dispatch function names and indices

Andreas Boll (1):

- glx/windows: Add wgl.h to the sources list

Anuj Phogat (1):

- i965: Fix GPU hang related to multiple render targets and alpha testing

Chih-Wei Huang (1):

- android: avoid using libdrm with host modules

Darren Salt (1):

- radv/pipeline: Don't dereference NULL dynamic state pointers

Dave Airlie (8):

- radv: expose xlib platform extension
- radv: fix dual source blending
- Revert "st/vdpau: use linear layout for output surfaces"
- radv: emit correct last export when Z/stencil export is enabled
- ac/nir: add support for discard_if intrinsic (v2)
- nir: add conditional discard optimisation (v4)
- radv: enable conditional discard optimisation on radv.
- radv: fix GetFenceStatus for signaled fences

Emil Velikov (6):

- docs: add sha256 checksums for 13.0.0

- amd/addrlib: limit fastcall/regparm to GCC i386
- anv: use correct .specVersion for extensions
- radv: use correct .specVersion for extensions
- radv: Suffix the radeon_icd file with the host CPU
- Update version to 13.0.1

Eric Anholt (1):

- vc4: Use Newton-Raphson on the 1/W write to fix glmark2 terrain.

Francisco Jerez (1):

- nir: Flip gl_SamplePosition in nir_lower_wpos_ytransform().

Fredrik Höglund (1):

- radv: add support for anisotropic filtering on VI+

Jason Ekstrand (21):

- anv/device: Return DEVICE_LOST if execbuf2 fails
- vulkan/wsi/x11: Better handle wsi_x11_connection_create failure
- vulkan/wsi/x11: Clean up connections in finish_wsi
- anv: Better handle return codes from anv_physical_device_init
- intel/blorp: Use wm_prog_data instead of hand-rolling our own
- intel/blorp: Pass a brw_stage_prog_data to upload_shader
- anv/pipeline: Put actual pointers in anv_shader_bin
- anv/pipeline: Properly cache prog_data::param
- intel/blorp: Emit all the binding tables
- anv/device: Add an execbuf wrapper
- anv: Add a cmd_buffer_execbuf helper
- anv: Don't presume to know what address is in a surface relocation
- anv: Add a new bo_pool_init helper
- anv/allocator: Simplify anv_scratch_pool
- anv: Initialize anv_bo::offset to -1
- anv/batch_chain: Improve write_reloc
- anv: Add an anv_execbuf helper struct
- anv/batch: Move last_ss_pool_bo_offset to the command buffer
- anv: Move relocation handling from EndCommandBuffer to QueueSubmit
- anv/cmd_buffer: Take a command buffer instead of a batch in two helpers
- anv/cmd_buffer: Enable a CS stall workaround for Sky Lake gt4

Kenneth Graunke (2):

- glsl: Update deref types when resizing implicitly sized arrays.
- mesa: Fix pixel shader scratch space allocation on Gen9+ platforms.

Kristian Høgsberg (1):

- anv: Do relocations in userspace before execbuf ioctl

Marek Olšák (4):

- egl: use util/macros.h
- egl: make interop ABI visible again
- glx: make interop ABI visible again
- radeonsi: fix an assertion failure in si_decompress_sampler_color_textures

Nicolai Hähnle (4):

- radeonsi: fix BFE/BFI lowering for GLSL semantics
- glsl: fix lowering of UBO references of named blocks
- st/glsl_to_tgsi: fix dvec[34] loads from SSBO
- st/mesa: fix the layer of VDPAU surface samplers

Steven Toth (3):

- gallium/hud: fix a problem where objects are free'd while in use.
- gallium/hud: close a previously opened handle
- gallium/hud: protect against and initialization race

Timothy Arceri (1):

- mesa/glsl: delete previously linked shaders earlier when linking

4.134 Mesa 12.0.4 Release Notes / November 10, 2016

Mesa 12.0.4 is a bug fix release which fixes bugs found since the 12.0.4 release.

Mesa 12.0.4 implements the OpenGL 4.3 API, but the version reported by glGetString(GL_VERSION) or glGetIntegerv(GL_MAJOR_VERSION) / glGetIntegerv(GL_MINOR_VERSION) depends on the particular driver being used. Some drivers don't support all the features required in OpenGL 4.3. OpenGL 4.3 is **only** available if requested at context creation because compatibility contexts are not supported.

4.134.1 SHA256 checksums

```
22026ce4f1c6a7908b0d10ff057dececc0a5633afe7f38a0cef5c08d0689f02a6 mesa-12.0.4.tar.gz
5d6003da867d3f54e5000b4acdfc37e6cce5b6a4459274fdad73e24bd2f0065e mesa-12.0.4.tar.xz
```

4.134.2 New features

None

4.134.3 Bug fixes

This list is likely incomplete.

- Bug 71759 - Intel driver fails with "intel_do_flush_locked failed: No such file or directory" if buffer imported with EGL_NATIVE_PIXMAP_KHR
- Bug 94354 - R9285 Unigine Valley perf regression since radeonsi: use re-Z
- Bug 96770 - include/GL/mesa_glinterop.h:62: error: redefinition of typedef 'GLXContext'
- Bug 97231 - GL_DEPTH_CLAMP doesn't clamp to the far plane
- Bug 97233 - vkQuake VkSpecializationMapEntry related bug
- Bug 97260 - R9 290 low performance in Linux 4.7
- Bug 97549 - [SNB, BXT] up to 40% perf drop from "loader/dri3: Overhaul dri3_update_num_back" commit
- Bug 97887 - llvm segfault in janusvr -render vive
- Bug 98025 - [radeonsi] incorrect primitive restart index used
- Bug 98134 - dEQP-GLES31.functional.debug.negative_coverage.get_error.buffer.draw_buffers wants a different GL error code
- Bug 98326 - [dEQP, EGL] pbuffer depth/stencil tests fail

4.134.4 Changes

Axel Davy (4):

- gallium/util: Really allow aliasing of dst for u_box_union_*
- st/nine: Fix the calculation of the number of vs inputs
- st/nine: Fix mistake in Volume9 UnlockBox
- st/nine: Fix locking CubeTexture surfaces.

Brendan King (1):

- configure.ac: fix the name of the Wayland Scanner pc file

Brian Paul (1):

- st/mesa: fix swizzle issue in st_create_sampler_view_from_stobj()

Chad Versace (3):

- egl: Fix truncation error in _eglParseSyncAttribList64
- i965/sync: Fix uninitalized usage and leak of mutex
- egl: Don't advertise unsupported platform extensions

Chuanbo Weng (1):

- gbm: fix potential NULL deref of mapImage/unmapImage.

Chuck Atkins (1):

- autoconf: Make header install distinct for various APIs (v2)

Dave Airlie (3):

- anv: initialise and increment send_sbc

- anv/wsi: fix apps that acquire multiple images up front
- Revert "st/vdpau: use linear layout for output surfaces"

Emil Velikov (12):

- docs: add sha256 checksums for 12.0.3
- cherry-ignore: add non-applicable i965 commit
- cherry-ignore: add vaapi encode fix
- cherry-ignore: add EGL_KHR_debug fix
- cherry-ignore: add update_renderbuffer_read_surfaces()
- isl/gen6: correctly check msaa layout samples count
- egl/x11: don't crash if dri2_dpy->conn is NULL
- get-pick-list.sh: Require explicit "12.0" for nominating stable patches
- automake: don't forget to pick wglext.h in the tarball
- cherry-ignore: add N/A EGL revert
- cherry-ignore: add ClientWaitSync fixes
- Update version to 12.0.4

Eric Anholt (5):

- travis: Parse configure.ac to pick an updated LIBDRM_VERSION.
- travis: Update to the Ubuntu Trusty image.
- travis: Enable vc4 in libdrm to satisfy vc4 test build dependency.
- travis: Upgrade LLVM dependency to 3.5 and enable LLVM drivers.
- gallium: Fix install-gallium-links.mk on non-bash /bin/sh

Hans de Goede (1):

- pipe_loader_sw: Fix fd leak when instantiated via pipe_loader_sw_probe_kms

Ian Romanick (1):

- glsl: Fix cut-and-paste bug in hierarchical visitor ir_expression::accept

Ilia Mirkin (16):

- nv30: set usage to staging so that the buffer is allocated in GART
- a3xx: make sure to actually clamp depth as requested
- a3xx: make use of software clipping when hw can't handle it
- a3xx: use window scissor to simulate viewport xy clip
- main: GL_RGB10_A2UI does not come with GL 3.0/EXT_texture_integer
- mesa/formatquery: limit ES target support, fix core context support
- nir: fix definition of pack_uvec2_to_uint
- gm107/ir: AL2P writes to a predicate register
- st/mesa: fix is_scissor_enabled when X/Y are negative
- nvc0/ir: fix overwriting of value backing non-constant gather offset

- nv50/ir: copy over value's register id when resolving merge of a phi
- nvc0/ir: fix textureGather with a single offset
- gm107/ir: fix texturing with indirect samplers
- gm107/ir: fix bit offset of tex lod setting for indirect texturing
- nv50,nvc0: avoid reading out of bounds when getting bogus so info
- nv50/ir: process texture offset sources as regular sources

James Legg (1):

- radeonsi: Fix primitive restart when index changes

Jason Ekstrand (9):

- nir/spirv: Swap the argument order for AtomicCompareExchange
- nir/spirv: Use the correct sources for CompareExchange on images
- nir/spirv: Break variable decoration handling into a helper
- nir/spirv: Refactor variable deocration handling
- nir/spirv/cfg: Handle switches whose break block is a loop continue
- nir/spirv/cfg: Detect switch_break after loop_break/continue
- nir: Add a nop intrinsic
- nir/spirv/cfg: Use a nop intrinsic for tagging the ends of blocks
- intel/blorp: Rework our usage of ralloc when compiling shaders

Jonathan Gray (3):

- genxml: add generated headers to EXTRA_DIST
- mapi: automake: set VISIBILITY_CFLAGS for shared glapi
- mesa: automake: include mesa_glinterop.h in distfile

Julien Isorce (1):

- st/va: also honors interlaced preference when providing a video format

Kenneth Graunke (8):

- nir: Call nir_metadata_preserve from nir_lower_alu_to_scalar().
- mesa: Expose RESET_NOTIFICATION_STRATEGY with KHR_robustness.
- i965: Fix missing _NEW_TRANSFORM in Gen8+ 3DSTATE_DS atom.
- i965: Add missing BRW_NEW_VS_PROG_DATA to 3DSTATE_CLIP.
- i965: Move BRW_NEW_FRAGMENT_PROGRAM from 3DSTATE_PS to PS_EXTRA.
- i965: Add missing BRW_NEW_CS_PROG_DATA to compute constant atom.
- i965: Add missing BRW_CS_PROG_DATA to CS work group surface atom.
- i965: Fix gl_InvocationID in dual object GS where invocations == 1.

Marek Olšák (12):

- radeonsi: fix cubemaps viewed as 2D
- radeonsi: take compute shader and dispatch indirect memory usage into account

- radeonsi: fix FP64 UBO loads with indirect uniform block indexing
- mesa: fix glGetFramebufferAttachmentParameteriv w/ on-demand FRONT_BACK alloc
- radeonsi: fix interpolateAt opcodes for .zw components
- radeonsi: fix texture border colors for compute shaders
- radeonsi: disable ReZ
- gallium/radeon: make sure the address of separate CMASK is aligned properly
- winsys/amdgpu: fix radeon_surf::macro_tile_index for imported textures
- egl: use util/macros.h
- egl: make interop ABI visible again
- glx: make interop ABI visible again

Mario Kleiner (1):

- glx: Perform check for valid fbconfig against proper X-Screen.

Martin Peres (2):

- loader/dri3: add get_dri_screen() to the vtable
- loader/dri3: import prime buffers in the currently-bound screen

Matt Whitlock (5):

- egl/android: replace call to dup(2) with fcntl(F_DUPFD_CLOEXEC)
- gallium/auxiliary: replace call to dup(2) with fcntl(F_DUPFD_CLOEXEC)
- st/dri: replace calls to dup(2) with fcntl(F_DUPFD_CLOEXEC)
- st/xa: replace call to dup(2) with fcntl(F_DUPFD_CLOEXEC)
- gallium/winsys: replace calls to dup(2) with fcntl(F_DUPFD_CLOEXEC)

Max Staudt (1):

- r300g: Set R300_VAP_CNTL on RSxxx to avoid triangle flickering

Michel Dänzer (1):

- loader/dri3: Overhaul dri3_update_num_back

Nicholas Bishop (2):

- gbm: return appropriate error when queryImage() fails
- st/dri: check pipe_screen->resource_get_handle() return value

Nicolai Hähnle (10):

- gallium/radeon: cleanup and fix branch emits
- st/glsl_to_tgsi: disable on-the-fly peephole for 64-bit operations
- st/glsl_to_tgsi: simplify translate_tex_offset
- st/glsl_to_tgsi: fix textureGatherOffset with indirectly loaded offsets
- st/mesa: fix vertex elements setup for doubles
- radeonsi: fix indirect loads of 64 bit constants
- st/glsl_to_tgsi: fix atomic counter addressing

- st/glsl_to_tgsi: fix block copies of arrays of doubles
- st/mesa: only set primitive_restart when the restart index is in range
- radeonsi: fix 64-bit loads from LDS

Samuel Pitoiset (4):

- nvc0/ir: fix subops for IMAD
- gk110/ir: fix wrong emission of OP_NOT
- nvc0: use correct bufctx when invalidating CP textures
- nvc0/ir: fix emission of IMAD with NEG modifiers

Stencel, Joanna (1):

- egl/wayland: add missing destroy_window callback

Tapani Pälli (5):

- egl: stop claiming support for pbuffer + msaa
- egl/dri2: set max values for pbuffer width and height
- egl: add check that eglCreateContext gets a valid config
- mesa: fix error handling in DrawBuffers
- egl: set preserved behavior for surface only if config supports it

Tim Rowley (1):

- configure.ac: add llvm inteljitevents component if enabled

Vedran Miletić (1):

- clover: Fix build against clang SVN >= r273191

Vinson Lee (1):

- Revert "mesa_glinterop: remove inclusion of GLX header"

4.135 Mesa 13.0.0 Release Notes / November 1, 2016

Mesa 13.0.0 is a new development release. People who are concerned with stability and reliability should stick with a previous release or wait for Mesa 13.0.1.

Mesa 13.0.0 implements the OpenGL 4.4 API, but the version reported by glGetString(GL_VERSION) or glGetIntegerv(GL_MAJOR_VERSION) / glGetIntegerv(GL_MINOR_VERSION) depends on the particular driver being used. Some drivers don't support all the features required in OpenGL 4.4. OpenGL 4.4 is **only** available if requested at context creation because compatibility contexts are not supported.

4.135.1 SHA256 checksums

```
4a54d7cdc1a94a8dae05a75ccff48356406d51b0d6a64cbdc641c266e3e008eb  mesa-13.0.0.tar.gz
94edb4ebff82066a68be79d9c2627f15995e1fe10f67ab3fc63deb842027d727  mesa-13.0.0.tar.xz
```

4.135.2 New features

Note: some of the new features are only available with certain drivers.

- OpenGL ES 3.1 on i965/hsw
- OpenGL ES 3.2 on i965/gen9+ (Skylake and later)
- GL_ARB_ES3_1_compatibility on i965
- GL_ARB_ES3_2_compatibility on i965/gen8+
- GL_ARB_clear_texture on r600, radeonsi
- GL_ARB_compute_variable_group_size on nvc0, radeonsi
- GL_ARB_cull_distance on radeonsi
- GL_ARB_enhanced_layouts on i965, nv50, nvc0, radeonsi, llvmpipe, softpipe
- GL_ARB_indirect_parameters on radeonsi
- GL_ARB_query_buffer_object on radeonsi
- GL_ARB_shader_draw_parameters on radeonsi
- GL_ARB_shader_group_vote on nvc0
- GL_ARB_shader_viewport_layer_array on i965/gen6+
- GL_ARB_stencil_texturing on i965/hsw
- GL_ARB_texture_stencil8 on i965/hsw
- GL_EXT_window_rectangles on nv50, nvc0
- GL_KHR_blend_equation_advanced on i965
- GL_KHR_robustness on nvc0, radeonsi
- GL_KHR_texture_compression_astc_sliced_3d on i965
- GL_OES_copy_image on nv50, nvc0, r600, radeonsi, softpipe, llvmpipe
- GL_OES_geometry_shader on i965/gen8+, nvc0, radeonsi
- GL_OES_primitive_bounding_box on i965/gen7+, nvc0, radeonsi
- GL_OES_texture_cube_map_array on i965/gen8+, nvc0, radeonsi
- GL_OES_tessellation_shader on i965/gen7+, nvc0, radeonsi
- GL_OES_viewport_array on nvc0, radeonsi
- GL_ANDROID_extension_pack_es31a on i965/gen9+

4.135.3 Bug fixes

- Bug 61907 - Indirect rendering of multi-texture vertex arrays broken
- Bug 69622 - eglTerminate then eglMakeCurrent crahes
- Bug 71759 - Intel driver fails with "intel_do_flush_locked failed: No such file or directory" if buffer imported with EGL_NATIVE_PIXMAP_KHR
- Bug 83036 - [ILK]Piglit spec_ARB_copy_image_arb_copy_image-formats fails
- Bug 89599 - symbol 'x86_64_entry_start' is already defined when building with LLVM/clang

- Bug 90513 - Odd gray and red flicker in The Talos Principle on GK104
- Bug 91342 - Very dark textures on some objects in indoors environments in Postal 2
- Bug 92306 - GL Excess demo renders incorrectly on nv43
- Bug 94148 - Framebuffer considered invalid when a draw call is done before glCheckFramebufferStatus
- Bug 94354 - R9285 Unigine Valley perf regression since radeonsi: use re-Z
- Bug 94561 - [llvmpipe] PIPE_CAP_VIDEO_MEMORY reports negative value on 32 bits (with 16GB ram)
- Bug 94627 - Game Risen on wine black grass
- Bug 94681 - dEQP-GLES31.functional.ssbo.layout.random.all_shared_buffer.23 takes 25 minutes to compile
- Bug 95000 - deqp: assert in dEQP-GLES3.functional.vertex_arrays.single_attribute.strides.fixed.user_ptr_stride17_components2_
- Bug 95130 - Derivatives of gl_Color wrong when helper pixels used
- Bug 95246 - Segfault in glBindFramebuffer()
- Bug 95419 - [HSW][regression][bisect] RPG Maker game gives "invalid floating point operation" at startup
- Bug 95462 - [BXT,BSW] arb_gpu_shader_fp64 causes gpu hang
- Bug 95529 - [regression, bisected] Image corruption in Chrome
- Bug 96235 - st_nir.h:34: error: redefinition of typedef 'nir_shader'
- Bug 96274 - [NVC0] Failure when compiling compute shader: Assertion 'bb->getFirst()->serial <= bb->getExit()->serial' failed
- Bug 96285 - Mesa build broken
- Bug 96299 - [vulkan] 64 regressions due to mesa d5f2f32
- Bug 96343 - oom since st/mesa: implement PBO downloads for ReadPixels
- Bug 96346 - [SNB,CTS] es2-cts.gtf.gl.atan regression
- Bug 96349 - [CTS,SKL,BSW,BDW,KBL,BXT] es31-cts.arrays_of_arrays.interactionuniformbuffers3
- Bug 96351 - [CTS,SKL,KBL,BXT] es2-cts.gtf.gl2extensiontests.egl_image.egl_image
- Bug 96358 - SSO: wrong interface validation between GS and VS (regresion due to latest gles 3.1)
- Bug 96425 - [bisected] occasional dark render in The Talos Principle
- Bug 96484 - [vulkan] deqp-vk.glsl.builtin.precision.sin / cos regression
- Bug 96504 - [vulkancts] compute tests crash
- Bug 96516 - [bisected: 482526] "clover: Update OpenCL version string to match OpenGL": clover's build fails because of missing git_sha1.h
- Bug 96528 - Location qualifier segfaults during shader compilation
- Bug 96541 - Tonga Unreal elemental bad rendering since radeonsi: Decompress DCC textures in a render feedback loop
- Bug 96565 - Clive Barker's Jericho displays strange,vivid colors when motion blur enabled
- Bug 96607 - [bisected] texture misrender / flicker in The Talos Principle on SKL
- Bug 96617 - gl_SecondaryFragDataEXT doesn't work for extended blend func
- Bug 96629 - dEQP-GLES2.functional.texture.completeness.cube.not_positive_level_0: Assertion 'width >= 1' failed.

- Bug 96639 - st/mesa: transfer_map with too-high level with dEQP-GLES2.functional.texture.completeness.cube.extra_level
- Bug 96674 - [SNB, ILK] spec.ext_image_dma_buf_import.ext_image_dma_buf_import-sample_nv1
- Bug 96729 - Wrong shader compilation error message
- Bug 96762 - [radeonsi,apitrace] Firewatch: nothing rendered in scrollable (text) areas
- Bug 96765 - BindFragDataLocationIndexed on array fragment shader output.
- Bug 96770 - include/GL/mesa_glinterop.h:62: error: redefinition of typedef 'GLXContext'
- Bug 96782 - [regression bisected] R600 fp64 and glsl-4.00 piglit failures
- Bug 96791 - Cannot use image from swapchains for sampling
- Bug 96825 - anv_device.c:31:27: fatal error: anv_timestamp.h: No such file or directory
- Bug 96835 - "gallium: Force blend color to 16-byte alignment" crash with "-march=native -O3" causes some 32bit games to crash
- Bug 96850 - Crucible tests fail for 32bit mesa
- Bug 96878 - [Bisected: cc2d0e6][HSW] "GPU HANG" msg after autologin to gnome-session
- Bug 96908 - [radeonsi] MSAA causes graphical artifacts
- Bug 96911 - webgl2 conformance2/textures/misc/tex-mipmap-levels.html crashes 12.1 Intel driver
- Bug 96949 - [regression] Piglit numSamples assertion failures with 9a23a177b90
- Bug 96950 - Another regression from bc4e0c486: vbo: Use a bitmask to track the active arrays in vbo_exec*.
- Bug 96971 - invariant qualifier is not valid for shader inputs
- Bug 97019 - [clover] build failure in llvm/codegen/native.cpp:129:52
- Bug 97032 - [BDW,SKL] piglit.spec.arb_gpu_shader5.arb_gpu_shader5-interpolateatcentroid-flat
- Bug 97033 - [BDW,SKL] piglit.spec.arb_gpu_shader_fp64.varying-packing.simple regressions
- Bug 97039 - The Talos Principle and Serious Sam 3 GPU faults
- Bug 97083 - [IVB,BYT] GPU hang on deqp-gles31.functional.separate.shader.random
- Bug 97140 - dd_draw.c:949:11: error: implicit declaration of function 'fmemopen' is invalid in C99 [-Werror,-Wimplicit-function-declaration]
- Bug 97207 - [IVY BRIDGE] Fragment shader discard writing to depth
- Bug 97214 - X not running with error "Failed to make EGL context current"
- Bug 97225 - [i965 on HD4600 Haswell] xcom switch to ingame cinematics cause segmentation fault
- Bug 97231 - GL_DEPTH_CLAMP doesn't clamp to the far plane
- Bug 97233 - vkQuake VkSpecializationMapEntry related bug
- Bug 97260 - R9 290 low performance in Linux 4.7
- Bug 97267 - [BDW] GL45-CTS.texture_cube_map_array.sampling asserts inside brw_fs.cpp
- Bug 97278 - [vulkancts,HSW] all vulkancts tests assert on HSW
- Bug 97285 - Darkness in Dota 2 after Patch "Make Gallium's BlitFramebuffer follow the GL 4.4 sRGB rules"
- Bug 97286 - 'make check' fails uniform-initializer-test
- Bug 97305 - Gallium: TBOs and images set the offset in elements, not bytes

- Bug 97307 - glsl/glcpp/tests/glcpp-test regression
- Bug 97309 - piglit.spec.glsl-1_30.compiler.switch-statement.switch-case-duplicated.vert regression
- Bug 97322 - GenerateMipmap creates wrong mipmap for sRGB texture
- Bug 97331 - glDrawElementsBaseVertex doesn't work in display list on i915
- Bug 97351 - DrawElementsBaseVertex with VBO ignores base vertex on Intel GMA 9xx in some cases
- Bug 97413 - BioShock Infinite crashes on startup with Mesa Git version, R7 370
- Bug 97426 - glScissor gives vertically inverted result
- Bug 97448 - [HSW] deqp-vk.api_.copy_and_blit.image_to_image_stencil regression
- Bug 97476 - Shader binaries should not be stored in the PipelineCache
- Bug 97477 - i915g: gl_FragCoord is always (0.0, max_y)
- Bug 97513 - clover reports wrong device pointer size
- Bug 97549 - [SNB, BXT] up to 40% perf drop from "loader/dri3: Overhaul dri3_update_num_back" commit
- Bug 97587 - make check nir/tests/control_flow_tests regression
- Bug 97761 - es2-cts.gtf.gl2extensiontests.egl_image_external.testsimpleunassociated crashes
- Bug 97773 - New Mesa master now results in warnings in glrender (and subsurfaces and simple-egl), black screen
- Bug 97779 - [regression, bisected][BDW, GPU hang] stuck on render ring, always reproducible
- Bug 97790 - Vulkan cts regressions due to 24be63066
- Bug 97804 - Later precision statement isn't overriding earlier one
- Bug 97808 - "tgsi/scan: don't set interp flags for inputs only used by INTERP instructions" causes glitches in wine with gallium nine
- Bug 97887 - llvm segfault in janusvr -render vive
- Bug 97894 - Crash in u_transfer_unmap_vtbl when unmapping a buffer mapped in different context
- Bug 97952 - /usr/include/string.h:518:12: error: exception specification in declaration does not match previous declaration
- Bug 97969 - [radeonsi, bisected: fb827c0] Video decoding shows green artifacts
- Bug 97976 - VCE regression BO to small for addr since winsys/amdgpu: enable buffer allocation from slabs
- Bug 98005 - VCE dual instance encoding inconsistent since st/va: enable dual instances encode by sync surface
- Bug 98025 - [radeonsi] incorrect primitive restart index used
- Bug 98128 - nir/tests/control_flow_tests.cpp:79:73: error: 'nir_loop_first_cf_node' was not declared in this scope
- Bug 98131 - Compiler should reject lowp/mediump qualifiers on atomic_uints
- Bug 98133 - GetSynciv should raise an error if bufSize < 0
- Bug 98134 - dEQP-GLES31.functional.debug.negative_coverage.get_error.buffer.draw_buffers wants a different GL error code
- Bug 98135 - dEQP-GLES31.functional.debug.negative_coverage.get_error.shader.transform_feedback_varyings wants a different GL error code
- Bug 98167 - [vulkan, radv] missing libgcrypt and openssl devel results in linker error in libvulkan_common

- Bug 98172 - Concurrent call to glClientWaitSync results in segfault in one of the waiters.
- Bug 98244 - dEQP: textureOffset(sampler2DArrayShadow, . . .) should not exist.
- Bug 98264 - Build broken for i965 due to multiple deifnitions of intelFenceExtension
- Bug 98307 - "st/glsl_to_tgsi: explicitly track all input and output declaration" broke flightgear colors on rs780
- Bug 98326 - [dEQP, EGL] pbuffer depth/stencil tests fail
- Bug 98415 - Vulkan Driver JSON file contains incorrect field
- Bug 98431 - UnrealEngine v4 demos startup fails to blorp blit assert

4.135.4 Changes

Mesa no longer depends on libudev.

4.136 Mesa 12.0.3 Release Notes / September 15, 2016

Mesa 12.0.3 is a bug fix release which fixes bugs found since the 12.0.3 release.

Mesa 12.0.3 implements the OpenGL 4.3 API, but the version reported by glGetString(GL_VERSION) or glGetIntegerv(GL_MAJOR_VERSION) / glGetIntegerv(GL_MINOR_VERSION) depends on the particular driver being used. Some drivers don't support all the features required in OpenGL 4.3. OpenGL 4.3 is **only** available if requested at context creation because compatibility contexts are not supported.

4.136.1 SHA256 checksums

```
79abcfab3de30dbd416d1582a3cf6b1be308466231488775f1b7bb43be353602 mesa-12.0.3.tar.gz
1dc86dd9b51272eee1fad3df65e18cda2e556ef1bc0b6e07cd750b9757f493b1 mesa-12.0.3.tar.xz
```

4.136.2 New features

None

4.136.3 Bug fixes

This list is likely incomplete.

- Bug 97781 - [HSW, BYT, IVB] es2-cts.gtf.gl2extensiontests.depth_texture_cube_map.depth_texture_cube_map

4.136.4 Changes

Emil Velikov (3):

- docs: add sha256 checksums for 12.0.2
- Revert "i965/miptree: Stop multiplying cube depth by 6 in HiZ calculations"
- Update version to 12.0.3

José Fonseca (1):

- appveyor: Update winflexbison download URL.

4.137 Mesa 12.0.2 Release Notes / September 2, 2016

Mesa 12.0.2 is a bug fix release which fixes bugs found since the 12.0.1 release.

Mesa 12.0.2 implements the OpenGL 4.3 API, but the version reported by glGetString(GL_VERSION) or glGetIntegerv(GL_MAJOR_VERSION) / glGetIntegerv(GL_MINOR_VERSION) depends on the particular driver being used. Some drivers don't support all the features required in OpenGL 4.3. OpenGL 4.3 is **only** available if requested at context creation because compatibility contexts are not supported.

4.137.1 SHA256 checksums

```
a08565ab1273751ebe2ffa928cbf785056594c803077c9719d0763da780f2918  mesa-12.0.2.tar.gz
d957a5cc371dcd7ff2aa0d87492f263aece46f79352f4520039b58b1f32552cb  mesa-12.0.2.tar.xz
```

4.137.2 New features

None

4.137.3 Bug fixes

This list is likely incomplete.

- Bug 69622 - eglTerminate then eglMakeCurrent crahes
- Bug 89599 - symbol 'x86_64_entry_start' is already defined when building with LLVM/clang
- Bug 91342 - Very dark textures on some objects in indoors environments in Postal 2
- Bug 92306 - GL Excess demo renders incorrectly on nv43
- Bug 94148 - Framebuffer considered invalid when a draw call is done before glCheckFramebufferStatus
- Bug 96274 - [NVC0] Failure when compiling compute shader: Assertion 'bb->getFirst()->serial <= bb->getExit()->serial' failed
- Bug 96358 - SSO: wrong interface validation between GS and VS (regresion due to latest gles 3.1)
- Bug 96381 - Texture artifacts with immutable texture storage and mipmaps
- Bug 96762 - [radeonsi,apitrace] Firewatch: nothing rendered in scrollable (text) areas
- Bug 96835 - "gallium: Force blend color to 16-byte alignment" crash with "-march=native -O3" causes some 32bit games to crash
- Bug 96850 - Crucible tests fail for 32bit mesa
- Bug 96908 - [radeonsi] MSAA causes graphical artifacts
- Bug 96911 - webgl2 conformance2/textures/misc/tex-mipmap-levels.html crashes 12.1 Intel driver
- Bug 96971 - invariant qualifier is not valid for shader inputs
- Bug 97039 - The Talos Principle and Serious Sam 3 GPU faults
- Bug 97207 - [IVY BRIDGE] Fragment shader discard writing to depth

- Bug 97214 - X not running with error "Failed to make EGL context current"
- Bug 97225 - [i965 on HD4600 Haswell] xcom switch to ingame cinematics cause segmentation fault
- Bug 97231 - GL_DEPTH_CLAMP doesn't clamp to the far plane
- Bug 97307 - glsl/glcpp/tests/glcpp-test regression
- Bug 97331 - glDrawElementsBaseVertex doesn't work in display list on i915
- Bug 97351 - DrawElementsBaseVertex with VBO ignores base vertex on Intel GMA 9xx in some cases
- Bug 97426 - glScissor gives vertically inverted result
- Bug 97476 - Shader binaries should not be stored in the PipelineCache
- Bug 97567 - [SNB, ILK] ctl, piglit regressions in mesa 12.0.2rc1

4.137.4 Changes

Andreas Boll (1):

- configure.ac: Use ${datarootdir} for –with-vulkan-icddir help string too

Bernard Kilarski (1):

- glx: fix error code when there is no context bound

Brian Paul (4):

- svga: handle mismatched number of samplers, sampler views
- mesa: use _mesa_clear_texture_image() in clear_texture_fields()
- swrast: fix incorrectly positioned putImage() in swrast driver
- mesa: fix format conversion bug in get_tex_rgba_uncompressed()

Chad Versace (2):

- i965: Fix miptree layout for EGLImage-based renderbuffers
- i965: Respect miptree offsets in intel_readpixels_tiled_memcpy()

Christian König (1):

- st/mesa: fix reference counting bug in st_vdpau

Chuck Atkins (1):

- swr: Refactor checks for compiler feature flags

Daniel Scharrer (1):

- mesa: Fix fixed function spot lighting on newer hardware (again)

Dave Airlie (2):

- anv: fix writemask on blit fragment shader.
- st/glsl_to_tgsi: fix st_src_reg_for_double constant.

Emil Velikov (15):

- docs: add sha256 checksums for 12.0.1
- mesa: automake: list builddir before srcdir
- mesa: scons: list builddir before srcdir

- i965: store reference to the context within struct brw_fence (v2)
- anv: remove internal 'validate' layer
- anv: automake: use VISIBILITY_CFLAGS to restrict symbol visibility
- anv: automake: build with -Bsymbolic
- anv: do not export the Vulkan API
- anv: remove dummy VK_DEBUG_MARKER_EXT entry points
- isl: automake: use VISIBILITY_CFLAGS to restrict symbol visibility
- cherry-ignore: temporary(?) drop "a4xx: make sure to actually clamp depth"
- i915: Check return value of screen->image.loader->getBuffers
- Revert "i965/miptree: Set logical_depth0 == 6 for cube maps"
- glx/glvnd: list the strcmp arguments in correct order
- Update version to 12.0.2

Eric Anholt (4):

- vc4: Close our screen's fd on screen close.
- vc4: Disable early Z with computed depth.
- vc4: Fix a leak of the src[] array of VPM reads in optimization.
- vc4: Fix leak of the bo_handles table.

Francisco Jerez (3):

- i965: Emit SKL VF cache invalidation W/A from brw_emit_pipe_control_flush.
- i965: Make room in the batch epilogue for three more pipe controls.
- i965: Fix remaining flush vs invalidate race conditions in brw_emit_pipe_control_flush.

Haixia Shi (1):

- platform_android: prevent deadlock in droid_swap_buffers

Ian Romanick (5):

- mesa: Strip arrayness from interface block names in some IO validation
- glsl: Pack integer and double varyings as flat even if interpolation mode is none
- glcpp: Track the actual version instead of just the version_resolved flag
- glcpp: Only disallow #undef of pre-defined macros on GLSL ES >= 3.00 shaders
- glsl: Mark cube map array sampler types as reserved in GLSL ES 3.10

Ilia Mirkin (16):

- mesa: etc2 online compression is unsupported, don't attempt it
- st/mesa: return appropriate mesa format for ETC texture formats
- mesa: set _NEW_BUFFERS when updating texture bound to current buffers
- nv50,nvc0: srgb rendering is only available for rgba/bgra
- vbo: allow DrawElementsBaseVertex in display lists
- gallium/util: add helper to compute zmin/zmax for a viewport state

- nv50,nvc0: fix depth range when halfz is enabled
- nv50/ir: fix bb positions after exit instructions
- vbo: add basevertex when looking up elements for vbo splitting
- a4xx: only disable depth clipping, not all clipping, when requested
- nv50/ir: make sure cfg iterator always hits all blocks
- main: add missing EXTRA_END in OES_sample_variables get check
- nouveau: always enable at least one RC
- nv30: only bail on color/depth bpp mismatch when surfaces are swizzled
- a4xx: make sure to actually clamp depth as requested
- gk110/ir: fix quadop dall emission

Jan Ziak (2):

- egl/x11: avoid using freed memory if dri2 init fails
- loader: fix memory leak in loader_dri3_open

Jason Ekstrand (31):

- nir/spirv: Don't multiply the push constant block size by 4
- anv: Add a stub for CmdCopyQueryPoolResults on Ivy Bridge
- glsl/types: Fix function type comparison function
- glsl/types: Use _mesa_hash_data for hashing function types
- genxml: Make gen6-7 blending look more like gen8
- anv/pipeline: Unify blend state setup between gen7 and gen8
- anv: Enable independentBlend on gen7
- anv: Add an align_down_npot_u32 helper
- anv: Handle VK_WHOLE_SIZE properly for buffer views
- i965/miptree: Enforce that height == 1 for 1-D array textures
- i965/miptree: Set logical_depth0 == 6 for cube maps
- nir: Add a nir_deref_foreach_leaf helper
- nir/inline: Constant-initialize local variables in the callee if needed
- anv/pipeline: Set up point coord enables
- i965/miptree: Stop multiplying cube depth by 6 in HiZ calculations
- i965/vec4: Make opt_vector_float reset at the top of each block
- anv/blit2d: Add a format parameter to bind_dst and create_iview
- anv/blit2d: Add support for RGB destinations
- anv/clear: Make cmd_clear_image take an actual VkClearValue
- anv/clear: Clear E5B9G9R9 images as R32_UINT
- anv: Include the pipeline layout in the shader hash
- isl: Allow multisampled array textures

- anv/descriptor_set: memset anv_descriptor_set_layout
- anv/pipeline: Fix bind maps for fragment output arrays
- anv/allocator: Correctly set the number of buckets
- anv/pipeline: Properly handle OOM during shader compilation
- anv: Remove unused fields from anv_pipeline_bind_map
- anv: Add pipeline_has_stage guards a few places
- anv: Add a struct for storing a compiled shader
- anv/pipeline: Add support for caching the push constant map
- anv: Rework pipeline caching

José Fonseca (2):

- appveyor: Install pywin32 extensions.
- appveyor: Force Visual Studio 2013 image.

Kenneth Graunke (21):

- genxml: Add CLIPMODE_* prefix to 3DSTATE_CLIP's "Clip Mode" enum values.
- genxml: Add APIMODE_D3D missing enum values and improve consistency.
- anv: Fix near plane clipping on Gen7/7.5.
- anv: Enable early culling on Gen7.
- anv: Unify 3DSTATE_CLIP code across generations.
- genxml: Rename "API Rendering Disable" to "Rendering Disable".
- anv: Properly call gen75_emit_state_base_address on Haswell.
- i965: Include VUE handles for GS with invocations > 1.
- nir: Add a base const_index to shared atomic intrinsics.
- i965: Fix shared atomic intrinsics to pay attention to base.
- mesa: Add GL_BGRA_EXT to the list of GenerateMipmap internal formats.
- mesa: Don't call GenerateMipmap if Width or Height == 0.
- glsl: Delete bogus ir_set_program_inouts assert.
- glsl: Fix the program resource names of gl_TessLevelOuter/Inner[].
- glsl: Fix location bias for patch variables.
- glsl: Fix invariant matching in GLSL 4.30 and GLSL ES 1.00.
- mesa: Fix uf10_to_f32() scale factor in the E == 0 and M != 0 case.
- nir/builder: Add bany_inequal and bany helpers.
- i965: Implement the WaPreventHSTessLevelsInterference workaround.
- i965: Fix execution size of scalar TCS barrier setup code.
- i965: Fix barrier count shift in scalar TCS backend.

Leo Liu (2):

- st/omx/enc: check uninitialized list from task release

- vl/dri3: fix a memory leak from front buffer

Marek Olšák (7):

- glsl_to_tgsi: don't use the negate modifier in integer ops after bitcast
- radeonsi: add a workaround for a compute VGPR-usage LLVM bug
- winsys/amdgpu: disallow DCC with mipmaps
- gallium/util: fix align64
- radeonsi: only set dual source blending for MRT0
- radeonsi: fix VM faults due NULL internal const buffers on CIK
- radeonsi: disable SDMA texture copying on Carrizo

Matt Turner (4):

- mapi: Massage code to allow clang to compile.
- i965/vec4: Ignore swizzle of VGRF for use by var_range_end().
- mesa: Use AC_HEADER_MAJOR to include correct header for major().
- nir: Walk blocks in source code order in lower_vars_to_ssa.

Michel Dänzer (1):

- glx: Don't use current context in __glXSendError

Miklós Máté (1):

- vbo: set draw_id

Nanley Chery (5):

- anv/descriptor_set: Fix binding partly undefined descriptor sets
- isl: Fix assert on raw buffer surface state size
- anv/device: Fix max buffer range limits
- isl: Fix isl_tiling_is_any_y()
- anv/gen7_pipeline: Set PixelShaderKillPixel for discards

Nicolai Hähnle (7):

- radeonsi: explicitly choose center locations for 1xAA on Polaris
- radeonsi: fix Polaris MSAA regression
- radeonsi: ensure sample locations are set for line and polygon smoothing
- st_glsl_to_tgsi: only skip over slots of an input array that are present
- glsl: fix optimization of discard nested multiple levels
- radeonsi: flush TC L2 cache for indirect draw data
- radeonsi: add si_set_rw_buffer to be used for internal descriptors

Nicolas Boichat (6):

- egl/dri2: dri2_make_current: Set EGL error if bindContext fails
- egl/wayland: Set disp->DriverData to NULL on error
- egl/surfaceless: Set disp->DriverData to NULL on error

- egl/drm: Set disp->DriverData to NULL on error
- egl/android: Set dpy->DriverData to NULL on error
- egl/dri2: Add reference count for dri2_egl_display

Rob Herring (3):

- Android: add missing u_math.h include path for libmesa_isl
- vc4: fix vc4_resource_from_handle() stride calculation
- vc4: add hash table look-up for exported dmabufs

Samuel Pitoiset (7):

- nvc0/ir: fix images indirect access on Fermi
- nvc0: fix the driver cb size when draw parameters are used
- gm107/ir: add missing NEG modifier for IADD32I
- gm107/ir: make use of ADD32I for all immediates
- nvc0: upload sample locations on GM20x
- nvc0: invalidate textures/samplers on GK104+
- nv50/ir: always emit the NDV bit for OP_QUADOP

Stefan Dirsch (1):

- Avoid overflow in 'last' variable of FindGLXFunction(...)

Stencel, Joanna (1):

- egl/wayland-egl: Fix for segfault in dri2_wl_destroy_surface.

Tim Rowley (2):

- Revert "gallium: Force blend color to 16-byte alignment"
- swr: switch from overriding -march to selecting features

Tomasz Figa (8):

- gallium/dri: Add shared glapi to LIBADD on Android
- egl/android: Remove unused variables
- egl/android: Check return value of dri2_get_dri_config()
- egl/android: Stop leaking DRI images
- gallium/winsys/kms: Fix double refcount when importing from prime FD (v2)
- gallium/winsys/kms: Fully initialize kms_sw_dt at prime import time (v2)
- gallium/winsys/kms: Move display target handle lookup to separate function
- gallium/winsys/kms: Look up the GEM handle after importing a prime FD

4.138 Mesa 12.0.1 Release Notes / July 8, 2016

Mesa 12.0.1 is a bug fix release which fixes bugs found since the 12.0.1 release.

Mesa 12.0.1 implements the OpenGL 4.3 API, but the version reported by glGetString(GL_VERSION) or glGetIntegerv(GL_MAJOR_VERSION) / glGetIntegerv(GL_MINOR_VERSION) depends on the particular driver being used.

Some drivers don't support all the features required in OpenGL 4.3. OpenGL 4.3 is **only** available if requested at context creation because compatibility contexts are not supported.

4.138.1 SHA256 checksums

```
28dff9c045f4305c96a875a487b9f06c7e88d910511cd6016dbddcd1f53ade0d  mesa-12.0.1.tar.gz
bab24fb79f78c876073527f515ed871fc9c81d816f66c8a0b051d8d653896389  mesa-12.0.1.tar.xz
```

4.138.2 New features

None

4.138.3 Bug fixes

- Bug 96864 - Mesa 12.0 radeon build broken

4.138.4 Changes

Emil Velikov (4):

- docs: add sha256 checksums for 12.0.0
- radeon: reference the correct cdw/max_dw
- Update version to 12.0.1
- docs: add release notes for 12.0.1

4.139 Mesa 12.0.0 Release Notes / July 8, 2016

Mesa 12.0.0 is a new development release. People who are concerned with stability and reliability should stick with a previous release or wait for Mesa 12.0.1.

Mesa 12.0.0 implements the OpenGL 4.3 API, but the version reported by glGetString(GL_VERSION) or glGetIntegerv(GL_MAJOR_VERSION) / glGetIntegerv(GL_MINOR_VERSION) depends on the particular driver being used. Some drivers don't support all the features required in OpenGL 4.3. OpenGL 4.3 is **only** available if requested at context creation because compatibility contexts are not supported.

4.139.1 SHA256 checksums

```
3b8fa4d86d78f8f6ec86055b92ad1afe869001483593b3dd4531184b8bc4fcfb  mesa-12.0.0.tar.gz
0090c02521931893512429 2b482e3439bc43e8c074ad01086449fcad88547dc6  mesa-12.0.0.tar.xz
```

4.139.2 New features

Note: some of the new features are only available with certain drivers.

- OpenGL 4.3 on nvc0, radeonsi, i965 (Gen8+)

- OpenGL ES 3.1 on nvc0, radeonsi
- GL_ARB_ES3_1_compatibility on nvc0, radeonsi
- GL_ARB_compute_shader on nvc0, radeonsi, softpipe
- GL_ARB_cull_distance on i965/gen6+, nv50, nvc0, llvmpipe, softpipe
- GL_ARB_framebuffer_no_attachments on nvc0, r600, radeonsi, softpipe
- GL_ARB_internalformat_query2 on all drivers
- GL_ARB_query_buffer_object on i965/hsw+
- GL_ARB_robust_buffer_access_behavior on i965, nvc0, radeonsi
- GL_ARB_shader_atomic_counters on radeonsi, softpipe
- GL_ARB_shader_atomic_counter_ops on nvc0, radeonsi, softpipe
- GL_ARB_shader_image_load_store on nvc0, radeonsi, softpipe
- GL_ARB_shader_image_size on nvc0, radeonsi, softpipe
- GL_ARB_shader_storage_buffer_objects on radeonsi, softpipe
- GL_ATI_fragment_shader on all Gallium drivers
- GL_EXT_base_instance on all drivers that support GL_ARB_base_instance
- GL_EXT_clip_cull_distance on all drivers that support GL_ARB_cull_distance
- GL_KHR_robustness on i965
- GL_OES_copy_image on i965 (Baytrail and Gen8+)
- GL_OES_draw_buffers_indexed and GL_EXT_draw_buffers_indexed on all drivers that support GL_ARB_draw_buffers_blend
- GL_OES_gpu_shader5 and GL_EXT_gpu_shader5 on all drivers that support GL_ARB_gpu_shader5
- GL_OES_sample_shading on i965, nvc0, r600, radeonsi
- GL_OES_sample_variables on i965, nvc0, r600, radeonsi
- GL_OES_shader_image_atomic on all drivers that support GL_ARB_shader_image_load_store
- GL_OES_shader_io_blocks on i965, nvc0, radeonsi
- GL_OES_shader_multisample_interpolation on i965, nvc0, r600, radeonsi
- GL_OES_texture_border_clamp and GL_EXT_texture_border_clamp on all drivers that support GL_ARB_texture_border_clamp
- GL_OES_texture_buffer and GL_EXT_texture_buffer on i965, nvc0, radeonsi
- EGL_KHR_reusable_sync on all drivers
- GL_ARB_stencil_texture8 and GL_OES_stencil_texture8 on i965/gen8+

4.139.3 Bug fixes

- Bug 42187 - ES 1.1 conformance pntszary.c fail
- Bug 71789 - [r300g] Visuals not found in (default) depth = 24
- Bug 81585 - piglit spec_glsl-1.10_compiler_literals_invalid-float-suffix-capital-f.vert fails
- Bug 83036 - [ILK]Piglit spec_ARB_copy_image_arb_copy_image-formats fails

- Bug 89607 - Assertion hit in opt_array_splitting with recursive array indexing
- Bug 90513 - Odd gray and red flicker in The Talos Principle on GK104
- Bug 91526 - World of Warcraft (on Wine) has UI corruption with nouveau
- Bug 92363 - [BSW/BDW] ogles1conform Gets test fails
- Bug 92628 - HTTP site for Mesa downloads
- Bug 92743 - Centroid shouldn't have to match between the FS and the VS
- Bug 92850 - Segfault loading War Thunder
- Bug 93054 - [BDW] DiRT Showdown and Bioshock Infinite only render half the screen (bottom left triangle)
- Bug 93524 - Clover doesn't build
- Bug 93551 - Divinity: Original Sin Enhanced Edition(Native) crash on start
- Bug 93667 - Crash in eglCreateImageKHR with huge texture size
- Bug 93767 - Glitches with soft shadows and MSAA in Knights of the Old Republic 2
- Bug 93840 - [i965] Alien: Isolation fails with GL_ARB_compute_shader enabled
- Bug 93962 - [HSW, regression, bisected, CTS] ES2-CTS.gtf.GL2FixedTests.scissor.scissor - segfault/asserts
- Bug 94081 - [HSW] compute shader shared var + atomic op = fail
- Bug 94086 - Multiple conflicting libGL libraries installed
- Bug 94116 - program interface queries not returning right data for UBO / GL_BLOCK_INDEX
- Bug 94129 - Mesa's compiler should warn about undefined values
- Bug 94181 - [regression] piglit.spec.ext_framebuffer_object.getteximage-formats init-by-clear-and-render
- Bug 94193 - [llvmpipe] Line antialiasing looks different when GL_LINE_STIPPLE is enabled with pattern 0xffff
- Bug 94198 - [HSW] segfault in copy image when copying from cubemap to 2d
- Bug 94199 - Shader abort/crash
- Bug 94253 - [llvmpipe] piglit gl-1.0-swapbuffers-behavior regression
- Bug 94254 - [llvmpipe] [softpipe] piglit read-front regression
- Bug 94257 - [softpipe] piglit glx-copy-sub-buffer regression
- Bug 94274 - [swrast] piglit arb_occlusion_query2-render regression
- Bug 94284 - [radeonsi] outlast segfault on start
- Bug 94291 - llvmpipe tests fail if built on skylake i7-6700k
- Bug 94348 - vkBindImageMemory doesn't take into account the offset when the image is used as a depth buffer
- Bug 94383 - build error on i386 when enabling swr
- Bug 94388 - r600_blit.c:281: r600_decompress_depth_textures: Assertion 'tex->is_depth && !tex->is_flushing_texture' failed.
- Bug 94412 - Trine 3 misrender
- Bug 94447 - glsl/glcpp/tests/glcpp-test-cr-lf regression
- Bug 94453 - dEQP-GLES3.functional.clipping.line.wide_line_clip_viewport_{center,corner} fail
- Bug 94454 - dEQP-GLES3.functional.clipping.point.wide_point_clip* fails

- Bug 94456 - dEQP-GLES3.functional.state_query.floats.{blend_color,color_clear_value,depth_clear_value}_getinteger64 fail
- Bug 94458 - dEQP-GLES3.functional.state_query.fbo.framebuffer_attachment_x_size_initial fails
- Bug 94468 - [HSW, regression, bisected] numerous Sascha demos render incorrectly
- Bug 94481 - softpipe - access violation in img_filter_2d_nearest
- Bug 94485 - dEQP-GLES3.functional.negative_api.shader.compile_shader and delete_shader broken by Meta
- Bug 94524 - Wrong gl_TessLevelOuter interpretation for isolines
- Bug 94595 - [Mesa AMD&swrast] Texture views attached as framebuffers return their viewed tecture's color encoding and render incorrectly
- Bug 94657 - [llvmpipe] [softpipe] piglit arb_texture_view-getteximage-srgb regression
- Bug 94661 - [bdw, skl] vk-cts: new test failing
- Bug 94671 - [radeonsi] Blue-ish textures in Shadow of Mordor
- Bug 94713 - [Gen8+] ES 3.1 Stencil texturing broken for 2DArray/Cubes
- Bug 94747 - Convert phi nodes to logical operations
- Bug 94835 - Increase fragment shader sample limits from 16 to 32 (AMD Linux - Mesa/RadeonSi)
- Bug 94847 - [ES3.1CTS] es31-cts.draw_buffers_indexed.color_masks fails
- Bug 94896 - [vulkan] new CTS tests fail on i965
- Bug 94904 - [vulkan, BSW] dEQP-VK.api.object_management.multithreaded_per_thread_device intermittent crash
- Bug 94907 - codegen/nv50_ir_ra.cpp:1330:29: error: 'isinf' was not declared in this scope
- Bug 94909 - [llvmpipe] piglit fs-roundEven-float regression
- Bug 94917 - radeonsi supports GL_ARB_shader_storage_buffer_object with 0 GL_MAX_COMBINED_SHADER_STORAGE_BLOCKS
- Bug 94924 - [GEN8] Ungine Valley fails to run due to "intel_do_flush_locked failed: Input/output error"
- Bug 94925 - Crash in egl_dri3_get_dri_context with Dolphin EGL/X11 in single-core mode
- Bug 94944 - [regression, hswgt1] gpu hang on arb_shader_image_load_store
- Bug 94955 - Uninitialized variables leads to random segfaults (valgrind log, apitrace attached)
- Bug 94969 - build fails because install-data-local doesn't follow $DESTDIR
- Bug 94972 - blend failures on llvmpipe with llvm 3.7 due to vector selects
- Bug 94979 - dolphin-emu rendering broken on gallium/SWR + crashing often
- Bug 94984 - XCom2 crashes with SIGSEGV on radeonsi
- Bug 94994 - OSMesaGetProcAdress always fails on mangled OSMesa
- Bug 94997 - [vulkan, SKL,BDW,HSW] deqp-vk.spirv_assembly.instruction.compute.opcopymemory.array regression
- Bug 94998 - [vulkan] deqp-vk.pipeline.push_constant.graphics_pipeline.count_3shader_vgf regression
- Bug 95001 - [vulkan] deqp-vk.binding_model.shader_access regression
- Bug 95005 - Unreal engine demos segfault after shader compilation error with OpenGL 4.3
- Bug 95026 - Alien Isolation segfault after initial loading screen/video

- Bug 95034 - vkResetCommandPool should not destroy the command buffers.
- Bug 95071 - [bisected] Wrong colors in KDE/Qt applications
- Bug 95133 - X-COM Enemy Within crashes when entering tactical mission with Bonaire
- Bug 95138 - [deqp, 32bit, gen8+] deqp-gles31.functional.draw_indirect.negative
- Bug 95142 - [ES3.1CTS,GEN8] ESEXT-CTS.draw_elements_base_vertex_tests.invalid_mapped_bos assertion
- Bug 95158 - glx-test compilation fails in 'make check'
- Bug 95164 - GLSL compiler (linker I think) emits assertion upon call to glAttachShader
- Bug 95180 - rasterizer/memory/Convert.h:170:9: error: '__builtin_isnan' is not a member of 'std'
- Bug 95198 - Shadow of Mordor beta has missing geometry with gl 4.3
- Bug 95203 - Tonga GST/OMX/VCE encode broken since mesa: st/omx: Fix resource leak on OMX_ErrorNone
- Bug 95211 - scons TypeError: 'tuple' object is not callable
- Bug 95246 - Segfault in glBindFramebuffer()
- Bug 95251 - vdpau decoder capabilities: not supported
- Bug 95252 - [deqp] deqp-gles31.functional.debug.object_labels.query_length_only crashes
- Bug 95292 - [IVB,SKL] vulkan: stride/tiling issue with vkCmdCopyBufferToImage from larger source buffer into destination image
- Bug 95296 - nir_lower_double_packing.c:79:4: error: void function 'lower_double_pack_impl' should not return a value [-Wreturn-type]
- Bug 95324 - GL33-CTS.gtf32.GL3Tests.packed_pixels.packed_pixels_pbo fails in one case on Haswell
- Bug 95370 - [965GM] piglit fails many tests after a5d7e144
- Bug 95373 - Suspicious warning in brw_blorp_clear.cpp
- Bug 95403 - [GK110] misaligned_gpr spamming dmesg when playing victor vran
- Bug 95419 - [HSW][regression][bisect] RPG Maker game gives "invalid floating point operation" at startup
- Bug 95456 - glXGetFBConfigs has invalid screen bounds
- Bug 95462 - [BXT,BSW] arb_gpu_shader_fp64 causes gpu hang
- Bug 95529 - [regression, bisected] Image corruption in Chrome
- Bug 95537 - Invalid argument in anv_ioctl called from anv_physical_device_init
- Bug 96221 - nir/nir_lower_tex.c:202: error: unknown field 'f32' specified in initializer
- Bug 96228 - SSBO test regressions from mesa 5b267509
- Bug 96236 - dri_interface.h:404: error: redefinition of typedef 'mesa_glinterop_device_info'
- Bug 96238 - swr fails to build outside of the main directory
- Bug 96239 - [radeonsi tessellation] [R9 290/390] Random "texture flickering" (Shadow of Mordor, Tomb Raider, Unigine Heaven 4.0)
- Bug 96258 - [NVC0] Hang when running compute program
- Bug 96285 - Mesa build broken
- Bug 96299 - [vulkan] 64 regressions due to mesa d5f2f32
- Bug 96346 - [SNB,CTS] es2-cts.gtf.gl.atan regression

- Bug 96349 - [CTS,SKL,BSW,BDW,KBL,BXT] es31-cts.arrays_of_arrays.interactionuniformbuffers3
- Bug 96351 - [CTS,SKL,KBL,BXT] es2-cts.gtf.gl2extensiontests.egl_image.egl_image
- Bug 96358 - SSO: wrong interface validation between GS and VS (regresion due to latest gles 3.1)
- Bug 96425 - [bisected] occasional dark render in The Talos Principle
- Bug 96504 - [vulkancts] compute tests crash
- Bug 96516 - [bisected: 482526] "clover: Update OpenCL version string to match OpenGL": clover's build fails because of missing git_sha1.h
- Bug 96565 - Clive Barker's Jericho displays strange,vivid colors when motion blur enabled
- Bug 96607 - [bisected] texture misrender / flicker in The Talos Principle on SKL
- Bug 96617 - gl_SecondaryFragDataEXT doesn't work for extended blend func
- Bug 96629 - dEQP-GLES2.functional.texture.completeness.cube.not_positive_level_0: Assertion 'width >= 1' failed.
- Bug 96639 - st/mesa: transfer_map with too-high level with dEQP-GLES2.functional.texture.completeness.cube.extra_level
- Bug 96674 - [SNB, ILK] spec.ext_image_dma_buf_import.ext_image_dma_buf_import-sample_nv1
- Bug 96765 - BindFragDataLocationIndexed on array fragment shader output.
- Bug 96791 - Cannot use image from swapchains for sampling
- Bug 96825 - anv_device.c:31:27: fatal error: anv_timestamp.h: No such file or directory

4.139.4 Changes

Radeon drivers (r600 and radeonsi) now require LLVm 3.6 as a minimum.

4.140 Mesa 11.2.2 Release Notes / May 9, 2016

Mesa 11.2.2 is a bug fix release which fixes bugs found since the 11.2.1 release.

Mesa 11.2.2 implements the OpenGL 4.1 API, but the version reported by glGetString(GL_VERSION) or glGetIntegerv(GL_MAJOR_VERSION) / glGetIntegerv(GL_MINOR_VERSION) depends on the particular driver being used. Some drivers don't support all the features required in OpenGL 4.1. OpenGL 4.1 is **only** available if requested at context creation because compatibility contexts are not supported.

4.140.1 SHA256 checksums

```
e2453014cd2cc5337a5180cdeffe8cf24fffbb83e20a96888e2b01df868eaae6  mesa-11.2.2.tar.gz
40e148812388ec7c6d7b6657d5a16e2e8dabba8b97ddfceea5197947647bdfb4  mesa-11.2.2.tar.xz
```

4.140.2 New features

None

4.140.3 Bug fixes

This list is likely incomplete.

- Bug 92850 - Segfault loading War Thunder
- Bug 93767 - Glitches with soft shadows and MSAA in Knights of the Old Republic 2
- Bug 94955 - Uninitialized variables leads to random segfaults (valgrind log, apitrace attached)
- Bug 94994 - OSMesaGetProcAdress always fails on mangled OSMesa
- Bug 95026 - Alien Isolation segfault after initial loading screen/video
- Bug 95133 - X-COM Enemy Within crashes when entering tactical mission with Bonaire
- Bug 95164 - GLSL compiler (linker I think) emits assertion upon call to glAttachShader
- Bug 95251 - vdpau decoder capabilities: not supported

4.140.4 Changes

Boyuan Zhang (1):

- radeon/uvd: alignment fix for decode message buffer

Brian Paul (2):

- st/mesa: fix sampler view leak in st_DrawAtlasBitmaps()
- gallium/util: initialize pipe_framebuffer_state to zeros

Chad Versace (1):

- dri: Fix robust context creation via EGL attribute

Egbert Eich (1):

- dri2: Check for dummyContext to see if the glx_context is valid

Emil Velikov (5):

- docs: add sha256 checksums for 11.2.1
- docs: update the sha256 checksums for 11.2.1
- cherry-ignore: remove duplicate commit
- cherry-ignore: ignore the GetSamplerParameterIuiv{EXT,OES} fixups
- Update version to 11.2.2

Eric Anholt (4):

- vc4: Fix subimage accesses to LT textures.
- vc4: Add support for rendering to cube map surfaces.
- vc4: Fix tests for format supported with nr_samples == 1.
- vc4: Make sure we recompile when sample_mask changes.

Frederic Devernay (1):

- glapi: fix _glapi_get_proc_address() for mangled function names

Ilia Mirkin (2):

- nvc0: fix retrieving query results into buffer for timestamps
- nouveau/video: properly detect the decoder class for availability checks

Jason Ekstrand (1):

- i965/fs: Properly report regs_written from SAMPLEINFO

Jonathan Gray (1):

- egl/x11: authenticate before doing chipset id ioctls

Jose Fonseca (1):

- winsys/sw/xlib: use correct free function for xlib_dt->data

Kenneth Graunke (3):

- i965: Fix clear code for ignoring colormask for XRGB formats on Gen9+.
- glsl: Convert lower_vec_index_to_swizzle to a rvalue visitor.
- glsl: Lower vector_extracts to swizzles after lower_vector_derefs.

Leo Liu (1):

- radeon/uvd: fix tonga feedback buffer size

Marek Olšák (1):

- st/mesa: fix blit-based GetTexImage for non-finalized textures

Nicolai Hähnle (5):

- gallium/radeon: handle failure when mapping staging buffer
- st/glsl_to_tgsi: reduce stack explosion in recursive expression visitor
- gallium/radeon: fix crash in r600_set_streamout_targets
- radeonsi: correct NULL-pointer check in si_upload_const_buffer
- radeonsi: work around an MSAA fast stencil clear problem

Oded Gabbay (4):

- r600g/radeonsi: send endian info to format translation functions
- r600g: set endianess of 16/32-bit buffers according to do_endian_swap
- r600g: use do_endian_swap in color swapping functions
- r600g: use do_endian_swap in texture swapping function

Patrick Rudolph (1):

- r600g: fix and optimize tgsi_cmp when using ABS and NEG modifier

Roland Scheidegger (3):

- llvmpipe: (trivial) initialize src1_alpha var to NULL
- gallivm: fix bogus argument order to lp_build_sample_mipmap function
- gallivm: make sampling more robust against bogus coordinates

Samuel Pitoiset (6):

- gk110/ir: do not overwrite def value with zero for EXCH ops
- gk110/ir: make use of IMUL32I for all immediates

- nvc0/ir: fix wrong emission of (a OP b) OP c
- gk110/ir: add emission for (a OP b) OP c
- nvc0: reduce GL_MAX_3D_TEXTURE_SIZE to 2048 on Kepler+
- st/glsl_to_tgsi: fix potential crash when allocating temporaries

Stefan Dirsch (1):

- dri3: Check for dummyContext to see if the glx_context is valid

Topi Pohjolainen (2):

- i965/blorp/gen7: Prepare re-using for gen8
- i965/blorp: Use 8k chunk size for urb allocation

WuZhen (3):

- tgsi: initialize stack allocated struct
- winsys/sw/dri: use correct free function for dri_sw_dt->data
- android: enable dlopen() on all architectures

4.141 Mesa 11.1.4 Release Notes / May 9, 2016

Mesa 11.1.4 is a bug fix release which fixes bugs found since the 11.1.3 release.

Mesa 11.1.4 implements the OpenGL 4.1 API, but the version reported by glGetString(GL_VERSION) or glGetIntegerv(GL_MAJOR_VERSION) / glGetIntegerv(GL_MINOR_VERSION) depends on the particular driver being used. Some drivers don't support all the features required in OpenGL 4.1. OpenGL 4.1 is **only** available if requested at context creation because compatibility contexts are not supported.

4.141.1 SHA256 checksums

```
034231fffb22621dadb8e4a968cb44752b8b68db7a2417568d63c275b3490cea  mesa-11.1.4.tar.gz
0f781e9072655305f576efd4204d183bf99ac8cb8d9e0dd9fc2b4093230a0eba  mesa-11.1.4.tar.xz
```

4.141.2 New features

None

4.141.3 Bug fixes

This list is likely incomplete.

- Bug 92850 - Segfault loading War Thunder
- Bug 93962 - [HSW, regression, bisected, CTS] ES2-CTS.gtf.GL2FixedTests.scissor.scissor - segfault/asserts
- Bug 94955 - Uninitialized variables leads to random segfaults (valgrind log, apitrace attached)
- Bug 94994 - OSMesaGetProcAdress always fails on mangled OSMesa
- Bug 95026 - Alien Isolation segfault after initial loading screen/video
- Bug 95133 - X-COM Enemy Within crashes when entering tactical mission with Bonaire

4.141.4 Changes

Brian Paul (1):

- gallium/util: initialize pipe_framebuffer_state to zeros

Chad Versace (1):

- dri: Fix robust context creation via EGL attribute

Egbert Eich (1):

- dri2: Check for dummyContext to see if the glx_context is valid

Emil Velikov (5):

- docs: add sha256 checksums for 11.1.3
- cherry-ignore: add non-applicable "fix of a fix"
- cherry-ignore: ignore st_DrawAtlasBitmaps mem leak fix
- cherry-ignore: add CodeEmitterGK110::emitATOM() fix
- Update version to 11.1.4

Eric Anholt (4):

- vc4: Fix subimage accesses to LT textures.
- vc4: Add support for rendering to cube map surfaces.
- vc4: Fix tests for format supported with nr_samples == 1.
- vc4: Make sure we recompile when sample_mask changes.

Frederic Devernay (1):

- glapi: fix _glapi_get_proc_address() for mangled function names

Jason Ekstrand (2):

- i965/tiled_memcopy: Add aligned mem_copy parameters to the [de]tiling functions
- i965/tiled_memcpy: Rework the RGBA -> BGRA mem_copy functions

Jonathan Gray (1):

- egl/x11: authenticate before doing chipset id ioctls

Jose Fonseca (1):

- winsys/sw/xlib: use correct free function for xlib_dt->data

Leo Liu (1):

- radeon/uvd: fix tonga feedback buffer size

Marek Olšák (2):

- drirc: add a workaround for blackness in Warsow
- st/mesa: fix blit-based GetTexImage for non-finalized textures

Nicolai Hähnle (5):

- radeonsi: fix bounds check in si_create_vertex_elements
- gallium/radeon: handle failure when mapping staging buffer
- st/glsl_to_tgsi: reduce stack explosion in recursive expression visitor

- gallium/radeon: fix crash in r600_set_streamout_targets
- radeonsi: correct NULL-pointer check in si_upload_const_buffer

Oded Gabbay (4):

- r600g/radeonsi: send endian info to format translation functions
- r600g: set endianess of 16/32-bit buffers according to do_endian_swap
- r600g: use do_endian_swap in color swapping functions
- r600g: use do_endian_swap in texture swapping function

Roland Scheidegger (3):

- llvmpipe: (trivial) initialize src1_alpha var to NULL
- gallivm: fix bogus argument order to lp_build_sample_mipmap function
- gallivm: make sampling more robust against bogus coordinates

Samuel Pitoiset (5):

- gk110/ir: make use of IMUL32I for all immediates
- nvc0/ir: fix wrong emission of (a OP b) OP c
- gk110/ir: add emission for (a OP b) OP c
- nvc0: reduce GL_MAX_3D_TEXTURE_SIZE to 2048 on Kepler+
- st/glsl_to_tgsi: fix potential crash when allocating temporaries

Stefan Dirsch (1):

- dri3: Check for dummyContext to see if the glx_context is valid

Thomas Hindoe Paaboel Andersen (1):

- st/va: avoid dereference after free in vlVaDestroyImage

WuZhen (3):

- tgsi: initialize stack allocated struct
- winsys/sw/dri: use correct free function for dri_sw_dt->data
- android: enable dlopen() on all architectures

4.142 Mesa 11.2.1 Release Notes / April 17, 2016

Mesa 11.2.1 is a bug fix release which fixes bugs found since the 11.2.0 release.

Mesa 11.2.1 implements the OpenGL 4.1 API, but the version reported by glGetString(GL_VERSION) or glGetIntegerv(GL_MAJOR_VERSION) / glGetIntegerv(GL_MINOR_VERSION) depends on the particular driver being used. Some drivers don't support all the features required in OpenGL 4.1. OpenGL 4.1 is **only** available if requested at context creation because compatibility contexts are not supported.

4.142.1 SHA256 checksums

```
cc2a024204564a71acc95cf262bf618fe49b1d77d351e5755eea705cadac5167  mesa-11.2.1.tar.gz
a65207e9ae5c5f1c29f863c6a2cc98a7ab99762a24b82a248337f0ea9cfce01b  mesa-11.2.1.tar.xz
```

4.142.2 New features

None

4.142.3 Bug fixes

This list is likely incomplete.

- Bug 93962 - [HSW, regression, bisected, CTS] ES2-CTS.gtf.GL2FixedTests.scissor.scissor - segfault/asserts

4.142.4 Changes

Brian Paul (2):

- st/mesa: fix glReadBuffer() assertion failure
- st/mesa: fix memleak in glDrawPixels cache code

Christian Schmidbauer (1):

- st/nine: specify WINAPI only for i386 and amd64

Emil Velikov (3):

- docs: add sha256 checksums for 11.2.0
- configure.ac: update the path of the generated files
- Update version to 11.2.1

Ilia Mirkin (1):

- glsl: allow usage of the keyword buffer before GLSL 430 / ESSL 310

Iurie Salomov (1):

- va: check null context in vlVaDestroyContext

Jason Ekstrand (2):

- i965/tiled_memcopy: Add aligned mem_copy parameters to the [de]tiling functions
- i965/tiled_memcpy: Rework the RGBA -> BGRA mem_copy functions

Kenneth Graunke (3):

- i965: Fix textureSize() depth value for 1 layer surfaces on Gen4-6.
- i965: Use brw->urb.min_vs_urb_entries instead of 32 for BLORP.
- glsl: Lower variable indexing of system value arrays unconditionally.

Marek Olšák (1):

- drirc: add a workaround for blackness in Warsow

Nicolai Hähnle (1):

- radeonsi: fix bounds check in si_create_vertex_elements

Samuel Pitoiset (1):

- nv50/ir: do not try to attach JOIN ops to ATOM

Thomas Hindoe Paaboel Andersen (1):

- st/va: avoid dereference after free in vlVaDestroyImage

4.143 Mesa 11.1.3 Release Notes / April 17, 2016

Mesa 11.1.3 is a bug fix release which fixes bugs found since the 11.1.2 release.

Mesa 11.1.3 implements the OpenGL 4.1 API, but the version reported by glGetString(GL_VERSION) or glGetIntegerv(GL_MAJOR_VERSION) / glGetIntegerv(GL_MINOR_VERSION) depends on the particular driver being used. Some drivers don't support all the features required in OpenGL 4.1. OpenGL 4.1 is **only** available if requested at context creation because compatibility contexts are not supported.

4.143.1 SHA256 checksums

```
9e86c72b6b2e8adb53c1c4a0002ab267b45094d753eb9404b1db34f81ce94ccf  mesa-11.1.3.tar.gz
51f6658a214d75e4d9f05207586d7ed56ebba75c6b10841176fb6675efa310ac  mesa-11.1.3.tar.xz
```

4.143.2 New features

None

4.143.3 Bug fixes

This list is likely incomplete.

- Bug 27512 - Illegal instruction _mesa_x86_64_transform_points4_general
- Bug 91526 - World of Warcraft (on Wine) has UI corruption with nouveau
- Bug 92193 - [SKL] ES2-CTS.gtf.GL2ExtensionTests.compressed_astc_texture.compressed_astc_texture fails
- Bug 93358 - [HSW] Unreal Elemental demo - assertion error in copy_image_with_blitter
- Bug 93418 - Geometry Shaders output wrong vertices on Sandy Bridge
- Bug 93524 - Clover doesn't build
- Bug 93667 - Crash in eglCreateImageKHR with huge texture size
- Bug 93813 - Incorrect viewport range when GL_CLIP_ORIGIN is GL_UPPER_LEFT
- Bug 94050 - test_vec4_register_coalesce regression
- Bug 94073 - Miscompilation of abs_vec3_vert_xvary_ref.vert in WebGL conformance
- Bug 94088 - [llvmpipe] SIGFPE pthread_barrier_destroy.c:40
- Bug 94193 - [llvmpipe] Line antialiasing looks different when GL_LINE_STIPPLE is enabled with pattern 0xffff
- Bug 94195 - [llvmpipe] Does not build with LLVM 3.7.x on Windows
- Bug 94388 - r600_blit.c:281: r600_decompress_depth_textures: Assertion 'tex->is_depth && !tex->is_flushing_texture' failed.
- Bug 94412 - Trine 3 misrender
- Bug 94481 - softpipe - access violation in img_filter_2d_nearest

- Bug 94595 - [Mesa AMD&swrast] Texture views attached as framebuffers return their viewed tecture's color encoding and render incorrectly
- Bug 94954 - test_vec4_copy_propagation fails in 'make check'

4.143.4 Changes

Anuj Phogat (1):

- i965: Fix assert conditions for src/dst x/y offsets

Ben Widawsky (2):

- i965: Make sure we blit a full compressed block
- i965/skl: Add two missing device IDs

Brian Paul (1):

- mesa: fix incorrect viewport position when GL_CLIP_ORIGIN = GL_LOWER_LEFT

Chris Forbes (1):

- i965/blorp: Fix hiz ops on MSAA surfaces

Christian König (1):

- radeon/uvd: disable MPEG1

Christian Schmidbauer (1):

- st/nine: specify WINAPI only for i386 and amd64

Daniel Czarnowski (3):

- egl_dri2: NULL check for xcb_dri2_get_buffers_reply()
- egl_dri2: set correct error code if swapbuffers fails
- egl: support EGL_LARGEST_PBUFFER in eglCreatePbufferSurface(...)

Dave Airlie (1):

- mesa/fbobject: propogate Layered when reusing attachments.

Derek Foreman (1):

- egl/wayland: Try to use wl_surface.damage_buffer for SwapBuffersWithDamage

Dongwon Kim (1):

- egl: move Null check to eglGetSyncAttribKHR to prevent Segfault

Emil Velikov (10):

- docs: add sha256 checksums for 11.1.2
- get-pick-list.sh: Require explicit "11.1" for nominating stable patches
- cherry-ignore: do not pick nv50/ir commit
- automake: add nine to make distcheck
- install-gallium-links: port changes from install-lib-links
- automake: add more missing options for make distcheck
- mesa; add get-extra-pick-list.sh script into bin/

- egl/x11: check the return value of xcb_dri2_get_buffers_reply()
- nvc/ir: remove duplicate variable declaration
- Update version to 11.1.3

Francisco Jerez (4):

- i965: Reupload push and pull constants when we get new shader image unit state.
- i965/fs: Add missing analysis invalidation in opt_sampler_eot().
- i965/fs: Add missing analysis invalidation in fixup_3src_null_dest().
- i965/vec4: Consider removal of no-op MOVs as progress during register coalesce.

Ilia Mirkin (21):

- nvc0/ir: fix converting between predicate and gpr
- nvc0: add some missing PUSH_SPACE's
- nvc0: avoid negatives in PUSH_SPACE argument
- glsl: make sure builtins are initialized before getting the shader
- glsl: return cloned signature, not the builtin one
- nv50/ir: fix quadop emission in the presence of predication
- st/mesa: fix up result_src.type when doing i2u/u2i conversions
- meta/copy_image: use precomputed dst_internal_format to avoid segfault
- st/mesa: force depth mode to GL_RED for sized depth/stencil formats
- glx: update to updated version of EXT_create_context_es2_profile
- nv50,nvc0: bump minimum texture buffer offset alignment
- nvc0: reset TFB bufctx when we no longer hold a reference to the buffers
- glsl: avoid stack smashing when there are too many attributes
- nvc0: fix blit triangle size to fully cover FB's > 8192x8192
- nv50: reset TFB bufctx when we no longer hold a reference to the buffers
- nv50/ir: force-enable derivatives on TXD ops
- st/mesa: only minify depth for 3d targets
- nv50/ir: fix indirect texturing for non-array textures on nvc0
- nvc0/ir: fix picking of coordinates from tex instruction for textureGrad
- nvc0: disable primitive restart and index bias during blits
- nv50/ir: we can't load local memory directly into an output

Jason Ekstrand (1):

- nir/lower_vec_to_movs: Better report channels handled by insert_mov

Kenneth Graunke (3):

- mesa: Make glGet queries initialize ctx->Debug when necessary.
- mesa: Allow Get*() of several forgotten IsEnabled() pnames.
- i965: Only magnify depth for 3D textures, not array textures.

Koop Mast (1):

- st/clover: Add libelf cflags to the build

Marc-André Lureau (1):

- virtio_gpu: Add virtio 1.0 PCI ID to driver map

Marek Olšák (3):

- radeonsi: fix Hyper-Z on Stoney
- gallium/radeon: don't use temporary buffers for persistent mappings
- radeonsi: fix Hyper-Z hangs on P2 configs

Matt Turner (3):

- i965/vec4: don't copy ATTR into 3src instructions with complex swizzles
- i965/fs: Don't CSE negated multiplies with saturation.
- i965/vec4: Update vec4 unit tests for commit 01dacc83ff.

Nanley Chery (2):

- mesa/image: Make _mesa_clip_readpixels() work with renderbuffers
- mesa/readpix: Clip ReadPixels() area to the ReadBuffer's

Nicolai Hähnle (2):

- r600g: clear compressed_depthtex/colortex_mask when binding buffer texture
- st/mesa: use the texture view's format for render-to-texture

Nishanth Peethambaran (2):

- st/omx: Remove trailing spaces
- st/omx/dec: Correct the timestamping

Oded Gabbay (8):

- gallium/radeon: Correctly translate colorswaps for big endian
- llvmpipe: use vpkswss when dst is signed
- gallium/radeon: return correct values for BE in r600_translate_colorswap
- gallium/radeon: remove separate BE path in r600_translate_colorswap
- gallium/r600: Don't let h/w do endian swap for colorformat
- gallium/radeon: disable evergreen_do_fast_color_clear for BE
- r600g: Do colorformat endian swap for PIPE_USAGE_STAGING
- radeonsi: Do colorformat endian swap for PIPE_USAGE_STAGING

Olivier Pena (1):

- scons: support for LLVM 3.7.

Patrick Baggett (1):

- mesa: Use SSE prefetch instructions rather than 3DNow instructions

Rob Herring (10):

- Android: remove dependence on .SECONDEXPANSION

- Android: glsl: fix dependence on YACC_HEADER_SUFFIX from build system
- Android: add -Wno-date-time flag for clang
- Android: remove headers from LOCAL_SRC_FILES
- Android: clean-up and fix DRI module path handling
- freedreno: drop unnecessary -Wno-packed-bitfield-compat
- gallium/radeon: Add space between string literal and identifier
- r600: Make enum alu_op_flags unsigned
- virtio_gpu: Add PCI ID to driver map
- Android: fix x86 gallium builds

Roland Scheidegger (2):

- softpipe: fix anisotropic filtering crash
- draw: fix line stippling

Samuel Pitoiset (1):

- nvc0: make sure to delete samplers used by compute shaders

Steinar H. Gunderson (1):

- mesa: Fix locking of GLsync objects.

Tamil velan (1):

- radeon/uvd: increase max height to 4096 for VI and newer

Thomas Hellstrom (2):

- winsys/svga: Fix an uninitialized return value
- winsys/svga: Increase the fence timeout

Vinson Lee (1):

- llvmpipe: Do not use barriers if not using threads.

xavier (1):

- r600/sb: Do not distribute neg in expr_handler::fold_assoc() when folding multiplications.

4.144 Mesa 11.2.0 Release Notes / 4 April 2016

Mesa 11.2.0 is a new development release. People who are concerned with stability and reliability should stick with a previous release or wait for Mesa 11.2.1.

Mesa 11.2.0 implements the OpenGL 4.1 API, but the version reported by glGetString(GL_VERSION) or glGetIntegerv(GL_MAJOR_VERSION) / glGetIntegerv(GL_MINOR_VERSION) depends on the particular driver being used. Some drivers don't support all the features required in OpenGL 4.1. OpenGL 4.1 is **only** available if requested at context creation because compatibility contexts are not supported.

4.144.1 SHA256 checksums

```
dea3d8143929aad5c24ef0993ddb05807b30c284b488fc62903adfcc1c127887  mesa-11.2.0.tar.gz
1c1fed2674abf3f16ed2623e9a5694d6752c293194e18462ebc644a19cfaafb2  mesa-11.2.0.tar.xz
```

4.144.2 New features

Note: some of the new features are only available with certain drivers.

- GL_ARB_arrays_of_arrays on all gallium drivers that provide GLSL 1.30
- GL_ARB_base_instance on freedreno/a4xx
- GL_ARB_compute_shader on i965
- GL_ARB_copy_image on r600
- GL_ARB_indirect_parameters on nvc0
- GL_ARB_query_buffer_object on nvc0
- GL_ARB_shader_atomic_counters on nvc0
- GL_ARB_shader_draw_parameters on i965, nvc0
- GL_ARB_shader_storage_buffer_object on nvc0
- GL_ARB_tessellation_shader on i965 and r600 (evergreen/cayman only)
- GL_ARB_texture_buffer_object_rgb32 on freedreno/a4xx
- GL_ARB_texture_buffer_range on freedreno/a4xx
- GL_ARB_texture_query_lod on freedreno/a4xx
- GL_ARB_texture_rgb10_a2ui on freedreno/a4xx
- GL_ARB_texture_view on freedreno/a4xx
- GL_ARB_vertex_type_10f_11f_11f_rev on freedreno/a4xx
- GL_KHR_texture_compression_astc_ldr on freedreno/a4xx
- GL_AMD_performance_monitor on radeonsi (CIK+ only)
- GL_ATI_meminfo on r600, radeonsi
- GL_NVX_gpu_memory_info on r600, radeonsi
- New OSMesaCreateContextAttribs() function (for creating core profile contexts)

4.144.3 Bug fixes

- Bug 27512 - Illegal instruction _mesa_x86_64_transform_points4_general
- Bug 75165 - compute.c:464:49: error: function definition is not allowed here
- Bug 79783 - Distorted output in obs-studio where other vendors “work”
- Bug 89330 - piglit glsl-1.50 invariant-qualifier-in-out-block-01 regression
- Bug 89969 - nouveau: add support for chunk decoding in order to support vaapi (st/va)
- Bug 90348 - Spilling failure of b96 merged value

- Bug 91526 - World of Warcraft (on Wine) has UI corruption with nouveau
- Bug 91596 - EGL_KHR_gl_colorspace (v2) causes problem with Android-x86 GUI
- Bug 91806 - configure does not test whether assembler supports sse4.1
- Bug 91927 - [SKL] [regression] piglit compressed textures tests fail with kernel upgrade
- Bug 92193 - [SKL] ES2-CTS.gtf.GL2ExtensionTests.compressed_astc_texture.compressed_astc_texture fails
- Bug 92229 - [APITRACE] SOMA have serious graphical errors
- Bug 92233 - Unigine Heaven 4.0 silhuette run
- Bug 92363 - [BSW/BDW] ogles1conform Gets test fails
- Bug 92438 - Segfault in pushbuf_kref when running the android emulator (qemu) on nv50
- Bug 92589 - [BDW BSW SKL CTS] ES31-CTS.texture_gather.* GPU_HANG
- Bug 92595 - [HSW,BDW,SKL][GLES 3.1 CTS] Big difference in the results for the ES31-CTS.shader_bitfield_operation.* tests
- Bug 92609 - [BDW, BSW] piglit sampling-2d-array-as-2d-layer fails
- Bug 92687 - Add support for ARB_internalformat_query2
- Bug 92706 - glBlitFramebuffer refuses to blit RGBA to RGB with MSAA
- Bug 92709 - "LLVM triggered Diagnostic Handler: unsupported call to function ldexpf in main" when starting race in stuntrally
- Bug 92743 - Centroid shouldn't have to match between the FS and the VS
- Bug 92759 - [Regression, bisected] Visuals without alpha bits are not sRGB-capable
- Bug 92849 - [IVB HSW BDW] piglit image load/store load-from-cleared-image.shader_test fails
- Bug 92909 - Offset/alignment issue with layout std140 and vec3
- Bug 93004 - Guild Wars 2 crash on nouveau DX11 cards
- Bug 93048 - [CTS regression] mesa af2723 breaks GL Conformance for debug extension
- Bug 93063 - drm_helper.h:227:1: error: static declaration of 'pipe_virgl_create_screen' follows non-static declaration
- Bug 93091 - [opencl] segfault when running any opencl programs (like clinfo)
- Bug 93092 - lp_test_format regression
- Bug 93126 - wrongly claim supporting GL_EXT_texture_rg
- Bug 93180 - [regression] arb_separate_shader_objects.active sampler conflict fails
- Bug 93189 - "./util/u_inlines.h", line 83: operands have incompatible types: void ":" int
- Bug 93215 - [Regression bisected] Ogles1conform Automatic mipmap generation test is fail
- Bug 93235 - [regression] dispatch sanity broken by GetPointerv
- Bug 93257 - [SKL, bisected] ASTC dEQP tests segfault
- Bug 93264 - Tonga VM Faults since llvm ScheduleDAGInstrs: Rework schedule graph builder.
- Bug 93266 - gl_arb_shading_language_420pack does not allow binding of image variables
- Bug 93300 - Two Worlds 2 renders water incorrectly
- Bug 93312 - [SKL][GLES 3.1 CTS] ES31-CTS.layout_binding* GPU_HANG

- Bug 93320 - [HSW,BDW,SKL][GLES 3.1 CTS] ES31-CTS.vertex_attrib_binding.advanced-bindingUpdate fail
- Bug 93322 - [HSW,BDW,SKL][GLES 3.1 CTS] ES31-CTS.compute_shader.resource-ubo fail
- Bug 93323 - [HSW,BDW,SKL][GLES 3.1 CTS]ES31-CTS.shader_image_load_store.basic-allTargets-store-fs fail
- Bug 93325 - [HSW,BDW,SKL]ES31-CTS.explicit_uniform_location.uniform-loc-* 2 tests fail
- Bug 93339 - glLinkProgram() should fail when a varying is never written to in a previous stage
- Bug 93348 - [HSW,BDW,SKL][GLES 3.1 CTS] ES31-CTS.compute_shader.* segfault
- Bug 93358 - [HSW] Unreal Elemental demo - assertion error in copy_image_with_blitter
- Bug 93387 - inverse() shouldn't be exposed in GLSL 1.20 and 1.30
- Bug 93388 - [i965, regression, bisection] MESA_FORMAT_B8G8R8X8_SRGB changes break kwin
- Bug 93407 - [SKL][GLES 3.1 CTS]ES31-CTS.compute_shader.resources-texture fail
- Bug 93410 - [BDW,SKL][GLES 3.1 CTS]ES31-CTS.shader_image_load_store.negative-linkErrors fail
- Bug 93418 - Geometry Shaders output wrong vertices on Sandy Bridge
- Bug 93426 - [SKL,BDW,BSW,BXT] CTS regression: es2-cts.gtf.gl2fixedtests.buffer_objects.buffer_object,s
- Bug 93524 - Clover doesn't build
- Bug 93526 - GfxBench 4 tessellation demos misrender
- Bug 93532 - [HSW,BDW,SKL][GLES 3.1 CTS] ES31-CTS.compute_shader.*. Regression, bisected.
- Bug 93540 - [BISECTED, HSW] Rendering issue in Heaven (and other benchmarks)
- Bug 93560 - opt_combine_constants failing fabsf(reg->f) == table.imm[i].val assertion
- Bug 93599 - Strange green flashes with "Metro: Last Light Redux" + "Metro 2033 Redux" with Intel Mesa driver
- Bug 93648 - Random lines being rendered when playing Dolphin (geometry shaders related, w/ apitrace)
- Bug 93650 - GL_ARB_separate_shader_objects is buggy (PCSX2)
- Bug 93667 - Crash in eglCreateImageKHR with huge texture size
- Bug 93696 - [HSW,BDW;SKL][GLES 3.1 CTS]ES31-CTS.explicit_uniform_location.uniform-loc-mix-with-implicit-max-* fail
- Bug 93700 - [SKL, regression] deqp-gles2.functional.texture.completeness
- Bug 93717 - Meta mipmap generation can corrupt texture state
- Bug 93722 - Segfault when compiling shader with a subroutine that takes a parameter
- Bug 93725 - [HSW, regression, bisected] ES31-CTS.texture_gather.*depth*
- Bug 93731 - glUniformSubroutinesuiv segfaults when subroutine uniform is bound to a specific location
- Bug 93761 - A conditional discard in a fragment shader causes no depth writing at all
- Bug 93790 - [HSW] Use after free with compute programs
- Bug 93792 - [HSW] intel_mipmap_tree.c:1325: intel_miptree_copy_slice: Assertion 'src_mt->format == dst_mt->format
- Bug 93813 - Incorrect viewport range when GL_CLIP_ORIGIN is GL_UPPER_LEFT
- Bug 93840 - [i965] Alien: Isolation fails with GL_ARB_compute_shader enabled

- Bug 93862 - [Bisected] "drm/amdgpu: fix amdgpu_bo_pin_restricted VRAM placing v2" is bad
- Bug 93878 - [llvmpipe][softpipe] piglit arb_gpu_shader_fp64-double-gettransformfeedbackvarying regression
- Bug 93957 - [HSW] Mishandling of sample count when using an attachment-less framebuffer (assertion error)
- Bug 93961 - virgl build failure after 2016-02-01 changes - no previous prototype for 'virgl_drm_winsys_create'
- Bug 93962 - [HSW, regression, bisected, CTS] ES2-CTS.gtf.GL2FixedTests.scissor.scissor - segfault/asserts
- Bug 93989 - build: flex-2.5.39 seems to be failing for glsl_lexer.ll
- Bug 94016 - make check MesaExtensionsTest.AlphabeticallySorted regression
- Bug 94019 - [bisected] 3D acceleration broken with gallium/radeon: just get num_tile_pipes from the winsys
- Bug 94050 - test_vec4_register_coalesce regression
- Bug 94073 - Miscompilation of abs_vec3_vert_xvary_ref.vert in WebGL conformance
- Bug 94081 - [HSW] compute shader shared var + atomic op = fail
- Bug 94088 - [llvmpipe] SIGFPE pthread_barrier_destroy.c:40
- Bug 94091 - Tonga unreal elemental segfault since radeonsi: put image, fmask, and sampler descriptors into one array
- Bug 94100 - [HSW] compute indirect dispatch with 0 work groups causes gpu hang
- Bug 94134 - [regression] piglit.spec.arb_texture_view.sampling-2d-array-as-2d-layer assertion
- Bug 94139 - [regression, HSW, IVB] piglit.spec.arb_compute_shader.minmax
- Bug 94150 - UE4 Suntemple rendering errors
- Bug 94186 - Crash when launching glxinfo and World of Warcraft with RV790
- Bug 94188 - define (or undef) defined behaves stupidly
- Bug 94193 - [llvmpipe] Line antialiasing looks different when GL_LINE_STIPPLE is enabled with pattern 0xffff
- Bug 94199 - Shader abort/crash
- Bug 94253 - [llvmpipe] piglit gl-1.0-swapbuffers-behavior regression
- Bug 94254 - [llvmpipe] [softpipe] piglit read-front regression
- Bug 94257 - [softpipe] piglit glx-copy-sub-buffer regression
- Bug 94274 - [swrast] piglit arb_occlusion_query2-render regression
- Bug 94284 - [radeonsi] outlast segfault on start
- Bug 94388 - r600_blit.c:281: r600_decompress_depth_textures: Assertion 'tex->is_depth && !tex->is_flushing_texture' failed.
- Bug 94412 - Trine 3 misrender
- Bug 94481 - softpipe - access violation in img_filter_2d_nearest
- Bug 94524 - Wrong gl_TessLevelOuter interpretation for isolines
- Bug 94595 - [Mesa AMD&swrast] Texture views attached as framebuffers return their viewed tecture's color encoding and render incorrectly

4.144.4 Changes

Microsoft Visual Studio 2013 or later is now required for building on Windows. Previously, Visual Studio 2008 and later were supported.

4.145 Mesa 11.1.2 Release Notes / February 10, 2016

Mesa 11.1.2 is a bug fix release which fixes bugs found since the 11.1.1 release.

Mesa 11.1.2 implements the OpenGL 4.1 API, but the version reported by glGetString(GL_VERSION) or glGetIntegerv(GL_MAJOR_VERSION) / glGetIntegerv(GL_MINOR_VERSION) depends on the particular driver being used. Some drivers don't support all the features required in OpenGL 4.1. OpenGL 4.1 is **only** available if requested at context creation because compatibility contexts are not supported.

4.145.1 SHA256 checksums

```
ba0e7462b2936b86e6684c26fbb55519f8d9ad31d13a1c1e1afbe41e73466eea  mesa-11.1.2.tar.gz
8f72aead896b340ba0f7a4a474bfaf71681f5d675592aec1cb7ba698e319148b  mesa-11.1.2.tar.xz
```

4.145.2 New features

None

4.145.3 Bug fixes

This list is likely incomplete.

- Bug 91596 - EGL_KHR_gl_colorspace (v2) causes problem with Android-x86 GUI
- Bug 93628 - Exception: attempt to use unavailable module DRM when building MesaGL 11.1.0 on windows
- Bug 93648 - Random lines being rendered when playing Dolphin (geometry shaders related, w/ apitrace)
- Bug 93650 - GL_ARB_separate_shader_objects is buggy (PCSX2)
- Bug 93717 - Meta mipmap generation can corrupt texture state
- Bug 93722 - Segfault when compiling shader with a subroutine that takes a parameter
- Bug 93731 - glUniformSubroutinesuiv segfaults when subroutine uniform is bound to a specific location
- Bug 93761 - A conditional discard in a fragment shader causes no depth writing at all

4.145.4 Changes

Ben Widawsky (1):

- i965/bxt: Fix conservative wm thread counts.

Dave Airlie (1):

- glsl: fix subroutine lowering reusing actual parmaters

Emil Velikov (6):

- docs: add sha256 checksums for 11.1.1
- cherry-ignore: drop the i965/kbl .num_slices patch
- i915: correctly parse/set the context flags
- targets/dri: android: use WHOLE static libraries
- egl/dri2: expose srgb configs when KHR_gl_colorspace is available
- Update version to 11.1.2

Eric Anholt (2):

- vc4: Don't record the seqno of a failed job submit.
- vc4: Throttle outstanding rendering after submission.

François Tigeot (1):

- gallium: Add DragonFly support

Grazvydas Ignotas (1):

- r600g: don't leak driver const buffers

Ian Romanick (2):

- meta/blit: Restore GL_DEPTH_STENCIL_TEXTURE_MODE state for GL_TEXTURE_RECTANGLE
- meta: Use internal functions to set texture parameters

Ilia Mirkin (6):

- st/mesa: use surface format to generate mipmaps when available
- glsl: always compute proper varying type, irrespective of varying packing
- nvc0: avoid crashing when there are holes in vertex array bindings
- nv50,nvc0: fix buffer clearing to respect engine alignment requirements
- nv50/ir: fix false global CSE on instructions with multiple defs
- st/mesa: treat a write as a read for range purposes

Jason Ekstrand (3):

- i965/vec4: Use UW type for multiply into accumulator on GEN8+
- i965/fs/generator: Take an actual shader stage rather than a string
- i965/fs: Always set channel 2 of texture headers in some stages

Jose Fonseca (2):

- scons: Conditionally use DRM module on pipe-loader.
- pipe-loader: Fix PATH_MAX define on MSVC.

Karol Herbst (1):

- nv50/ir: fix memory corruption when spilling and redoing RA

Kenneth Graunke (2):

- glsl: Make bitfield_insert/extract and bfi/bfm non-vectorizable.
- glsl: Allow implicit int -> uint conversions for bitwise operators (&, ^, |).

Leo Liu (2):

- vl: add zig zag scan for list 4x4
- st/omx/dec/h264: fix corruption when scaling matrix present flag set

Marek Olšák (1):

- radeonsi: don't miss changes to SPI_TMPRING_SIZE

Nicolai Hähnle (11):

- mesa/bufferobj: make _mesa_delete_buffer_object externally accessible
- st/mesa: use _mesa_delete_buffer_object
- radeon: use _mesa_delete_buffer_object
- i915: use _mesa_delete_buffer_object
- i965: use _mesa_delete_buffer_object
- util/u_pstipple.c: copy immediates during transformation
- radeonsi: extract the VGT_GS_MODE calculation into its own function
- radeonsi: ensure that VGT_GS_MODE is sent when necessary
- radeonsi: add DCC buffer for sampler views on new CS
- st/mesa: use the correct address generation functions in st_TexSubImage blit
- radeonsi: fix discard-only fragment shaders (11.1 version)

Timothy Arceri (4):

- glsl: fix segfault linking subroutine uniform with explicit location
- mesa: fix segfault in glUniformSubroutinesuiv()
- glsl: fix interface block error message
- glsl: create helper to remove outer vertex index array used by some stages

4.146 Mesa 11.0.9 Release Notes / January 22, 2016

Mesa 11.0.9 is a bug fix release which fixes bugs found since the 11.0.8 release.

Mesa 11.0.9 implements the OpenGL 4.1 API, but the version reported by glGetString(GL_VERSION) or glGetIntegerv(GL_MAJOR_VERSION) / glGetIntegerv(GL_MINOR_VERSION) depends on the particular driver being used. Some drivers don't support all the features required in OpenGL 4.1. OpenGL 4.1 is **only** available if requested at context creation because compatibility contexts are not supported.

4.146.1 SHA256 checksums

```
1597c2e983f476f98efdd6cd58b5298896d18479ff542bdeff28b98b129ede05  mesa-11.0.9.tar.gz
a1262ff1c66a16ccf341186cf0e57b306b8589eb2cc5ce92ffb6788ab01d2b01  mesa-11.0.9.tar.xz
```

4.146.2 New features

None

4.146.3 Bug fixes

This list is likely incomplete.

- Bug 91596 - EGL_KHR_gl_colorspace (v2) causes problem with Android-x86 GUI
- Bug 92229 - [APITRACE] SOMA have serious graphical errors
- Bug 93257 - [SKL, bisected] ASTC dEQP tests segfault

4.146.4 Changes

Emil Velikov (6):

- docs: add sha256 checksums for 11.0.8
- cherry-ignore: add patch already in branch
- cherry-ignore: add the dri3 glx null check patch
- i915: correctly parse/set the context flags
- egl/dri2: expose srgb configs when KHR_gl_colorspace is available
- Update version to 11.0.9

Grazvydas Ignotas (1):

- r600: fix constant buffer size programming

Ilia Mirkin (5):

- nvc0: don't forget to reset VTX_TMP bufctx slot after blit completion
- nv50/ir: float(s32 & 0xff) = float(u8), not s8
- nv50,nvc0: make sure there's pushbuf space and that we ref the bo early
- nv50,nvc0: fix crash when increasing bsp bo size for h264
- nvc0: scale up inter_bo size so that it's 16M for a 4K video

Kenneth Graunke (2):

- ralloc: Fix ralloc_adopt() to the old context's last child's parent.
- nvc0: Set winding order regardless of domain.

Marek Olšák (1):

- radeonsi: don't miss changes to SPI_TMPRING_SIZE

Miklós Máté (1):

- mesa: Don't leak ATIfs instructions in DeleteFragmentShader

Neil Roberts (1):

- i965: Fix crash when calling glViewport with no surface bound

Nicolai Hähnle (6):

- gallium/radeon: only dispose locally created target machine in radeon_llvm_compile
- mesa/bufferobj: make _mesa_delete_buffer_object externally accessible
- st/mesa: use _mesa_delete_buffer_object

- radeon: use _mesa_delete_buffer_object
- i915: use _mesa_delete_buffer_object
- i965: use _mesa_delete_buffer_object

Oded Gabbay (1):

- llvmpipe: use vpkswss when dst is signed

Rob Herring (1):

- freedreno/ir3: fix 32-bit builds with pointer-to-int-cast error enabled

4.147 Mesa 11.1.1 Release Notes / January 13, 2016

Mesa 11.1.1 is a bug fix release which fixes bugs found since the 11.1.0 release.

Mesa 11.1.1 implements the OpenGL 4.1 API, but the version reported by glGetString(GL_VERSION) or glGetIntegerv(GL_MAJOR_VERSION) / glGetIntegerv(GL_MINOR_VERSION) depends on the particular driver being used. Some drivers don't support all the features required in OpenGL 4.1. OpenGL 4.1 is **only** available if requested at context creation because compatibility contexts are not supported.

4.147.1 SHA256 checksums

```
b15089817540ba0bffd0aad323ecf3a8ff6779568451827c7274890b4a269d58  mesa-11.1.1.tar.gz
64db074fc514136b5fb3890111f0d50604db52f0b1e94ba3fcb0fe8668a7fd20  mesa-11.1.1.tar.xz
```

4.147.2 New features

None

4.147.3 Bug fixes

This list is likely incomplete.

- Bug 91806 - configure does not test whether assembler supports sse4.1
- Bug 92229 - [APITRACE] SOMA have serious graphical errors
- Bug 92233 - Unigine Heaven 4.0 silhuette run
- Bug 93004 - Guild Wars 2 crash on nouveau DX11 cards
- Bug 93215 - [Regression bisected] Ogles1conform Automatic mipmap generation test is fail
- Bug 93257 - [SKL, bisected] ASTC dEQP tests segfault

4.147.4 Changes

Brian Paul (1):

- st/mesa: check state->mesa in early return check in st_validate_state()

Dave Airlie (6):

- mesa/varray: set double arrays to non-normalised.
- mesa/shader: return correct attribute location for double matrix arrays
- glsl: pass stage into mark function
- glsl/fp64: add helper for dual slot double detection.
- glsl: fix count_attribute_slots to allow for different 64-bit handling
- glsl: only update doubles inputs for vertex inputs.

Emil Velikov (4):

- docs: add sha256 checksums for 11.0.1
- cherry-ignore: drop the "re-enable" DCC on Stoney
- cherry-ignore: don't pick a specific i965 formats patch
- Update version to 11.1.1

Eric Anholt (2):

- vc4: Warn instead of abort()ing on exec ioctl failures.
- vc4: Keep sample mask writes from being reordered after TLB writes

Grazvydas Ignotas (1):

- r600: fix constant buffer size programming

Ian Romanick (1):

- meta/generate_mipmap: Work-around GLES 1.x problem with GL_DRAW_FRAMEBUFFER

Ilia Mirkin (9):

- nv50/ir: can't have predication and immediates
- gk104/ir: simplify and fool-proof texbar algorithm
- glsl: assign varying locations to tess shaders when doing SSO
- glx/dri3: a drawable might not be bound at wait time
- nvc0: don't forget to reset VTX_TMP bufctx slot after blit completion
- nv50/ir: float(s32 & 0xff) = float(u8), not s8
- nv50,nvc0: make sure there's pushbuf space and that we ref the bo early
- nv50,nvc0: fix crash when increasing bsp bo size for h264
- nvc0: scale up inter_bo size so that it's 16M for a 4K video

Jonathan Gray (2):

- configure.ac: use pkg-config for libelf
- configure: check for python2.7 for PYTHON2

Kenneth Graunke (5):

- ralloc: Fix ralloc_adopt() to the old context's last child's parent.
- drirc: Disable ARB_blend_func_extended for Heaven 4.0/Valley 1.0.
- glsl: Fix varying struct locations when varying packing is disabled.
- nvc0: Set winding order regardless of domain.

- nir: Add a lower_fdiv option, turn fdiv into fmul/frcp.

Marek Olšák (7):

- tgsi/scan: add flag colors_written
- r600g: write all MRTs only if there is exactly one output (fixes a hang)
- radeonsi: don't call of u_prims_for_vertices for patches and rectangles
- radeonsi: apply the streamout workaround to Fiji as well
- gallium/radeon: fix Hyper-Z hangs by programming PA_SC_MODE_CNTL_1 correctly
- program: add _mesa_reserve_parameter_storage
- st/mesa: fix GLSL uniform updates for glBitmap & glDrawPixels (v2)

Mark Janes (1):

- Add missing platform information for KBL

Miklós Máté (1):

- mesa: Don't leak ATIfs instructions in DeleteFragmentShader

Neil Roberts (3):

- i965: Add MESA_FORMAT_B8G8R8X8_SRGB to brw_format_for_mesa_format
- i965: Add B8G8R8X8_SRGB to the alpha format override
- i965: Fix crash when calling glViewport with no surface bound

Nicolai Hähnle (2):

- gallium/radeon: only dispose locally created target machine in radeon_llvm_compile
- gallium/radeon: fix regression in a number of driver queries

Oded Gabbay (1):

- configura.ac: fix test for SSE4.1 assembler support

Patrick Rudolph (2):

- nv50,nvc0: fix use-after-free when vertex buffers are unbound
- gallium/util: return correct number of bound vertex buffers

Rob Herring (1):

- freedreno/ir3: fix 32-bit builds with pointer-to-int-cast error enabled

Samuel Pitoiset (3):

- nvc0: free memory allocated by the prog which reads MP perf counters
- nv50,nvc0: free memory allocated by performance metrics
- nv50: free memory allocated by the prog which reads MP perf counters

Sarah Sharp (1):

- mesa: Add KBL PCI IDs and platform information.

4.148 Mesa 11.0.8 Release Notes / December 9, 2015

Mesa 11.0.8 is a bug fix release which fixes bugs found since the 11.0.7 release.

Mesa 11.0.8 implements the OpenGL 4.1 API, but the version reported by glGetString(GL_VERSION) or glGetIntegerv(GL_MAJOR_VERSION) / glGetIntegerv(GL_MINOR_VERSION) depends on the particular driver being used. Some drivers don't support all the features required in OpenGL 4.1. OpenGL 4.1 is **only** available if requested at context creation because compatibility contexts are not supported.

4.148.1 SHA256 checksums

```
ab9db87b54d7525e4b611b82577ea9a9eae55927558df57b190059d5ecd9406f  mesa-11.0.8.tar.gz
5696e4730518b6805d2ed5def393c4293f425a2c2c01bd5ed4bdd7ad62f7ad75  mesa-11.0.8.tar.xz
```

4.148.2 New features

None

4.148.3 Bug fixes

This list is likely incomplete.

- Bug 91806 - configure does not test whether assembler supports sse4.1
- Bug 92849 - [IVB HSW BDW] piglit image load/store load-from-cleared-image.shader_test fails
- Bug 92909 - Offset/alignment issue with layout std140 and vec3
- Bug 93004 - Guild Wars 2 crash on nouveau DX11 cards
- Bug 93215 - [Regression bisected] Ogles1conform Automatic mipmap generation test is fail
- Bug 93266 - gl_arb_shading_language_420pack does not allow binding of image variables

4.148.4 Changes

Boyuan Zhang (1):

- radeon/uvd: uv pitch separation for stoney

Dave Airlie (9):

- r600: do SQ flush ES ring rolling workaround
- r600: SMX returns CONTEXT_DONE early workaround
- r600/shader: split address get out to a function.
- r600/shader: add utility functions to do single slot arithmatic
- r600g: fix geom shader input indirect indexing.
- r600: handle geometry dynamic input array index
- radeonsi: handle doubles in lds load path.
- mesa/varray: set double arrays to non-normalised.

- mesa/shader: return correct attribute location for double matrix arrays

Emil Velikov (8):

- docs: add sha256 checksums for 11.0.7
- cherry-ignore: don't pick a specific i965 formats patch
- Revert "i965/nir: Remove unused indirect handling"
- Revert "i965/state: Get rid of dword_pitch arguments to buffer functions"
- Revert "i965/vec4: Use a stride of 1 and byte offsets for UBOs"
- Revert "i965/fs: Use a stride of 1 and byte offsets for UBOs"
- Revert "i965/vec4: Use byte offsets for UBO pulls on Sandy Bridge"
- Update version to 11.0.8

Francisco Jerez (1):

- i965: Resolve color and flush for all active shader images in intel_update_state().

Ian Romanick (1):

- meta/generate_mipmap: Work-around GLES 1.x problem with GL_DRAW_FRAMEBUFFER

Ilia Mirkin (17):

- freedreno/a4xx: support lod_bias
- freedreno/a4xx: fix 5_5_5_1 texture sampler format
- freedreno/a4xx: point regid to "red" even for alpha-only rb formats
- nvc0/ir: fold postfactor into immediate
- nv50/ir: deal with loops with no breaks
- nv50/ir: the mad source might not have a defining instruction
- nv50/ir: fix instruction permutation logic
- nv50/ir: don't forget to mark flagsDef on cvt in txb lowering
- nv50/ir: fix DCE to not generate 96-bit loads
- nv50/ir: avoid looking at uninitialized srcMods entries
- gk110/ir: fix imul hi emission with limm arg
- gk104/ir: sampler doesn't matter for txf
- gk110/ir: fix imad sat/hi flag emission for immediate args
- nv50/ir: fix cutoff for using r63 vs r127 when replacing zero
- nv50/ir: can't have predication and immediates
- glsl: assign varying locations to tess shaders when doing SSO
- ttn: add TEX2 support

Jason Ekstrand (5):

- i965/vec4: Use byte offsets for UBO pulls on Sandy Bridge
- i965/fs: Use a stride of 1 and byte offsets for UBOs
- i965/vec4: Use a stride of 1 and byte offsets for UBOs

- i965/state: Get rid of dword_pitch arguments to buffer functions
- i965/nir: Remove unused indirect handling

Jonathan Gray (2):

- configure.ac: use pkg-config for libelf
- configure: check for python2.7 for PYTHON2

Kenneth Graunke (2):

- i965: Fix fragment shader struct inputs.
- i965: Fix scalar vertex shader struct outputs.

Marek Olšák (8):

- radeonsi: fix occlusion queries on Fiji
- radeonsi: fix a hang due to uninitialized border color registers
- radeonsi: fix Fiji for LLVM <= 3.7
- radeonsi: don't call of u_prims_for_vertices for patches and rectangles
- radeonsi: apply the streamout workaround to Fiji as well
- gallium/radeon: fix Hyper-Z hangs by programming PA_SC_MODE_CNTL_1 correctly
- tgsi/scan: add flag colors_written
- r600g: write all MRTs only if there is exactly one output (fixes a hang)

Matt Turner (1):

- glsl: Allow binding of image variables with 420pack.

Neil Roberts (2):

- i965: Add MESA_FORMAT_B8G8R8X8_SRGB to brw_format_for_mesa_format
- i965: Add B8G8R8X8_SRGB to the alpha format override

Oded Gabbay (1):

- configura.ac: fix test for SSE4.1 assembler support

Patrick Rudolph (2):

- nv50,nvc0: fix use-after-free when vertex buffers are unbound
- gallium/util: return correct number of bound vertex buffers

Samuel Pitoiset (1):

- nvc0: free memory allocated by the prog which reads MP perf counters

Tapani Pälli (1):

- i965: use _Shader to get fragment program when updating surface state

Tom Stellard (2):

- radeonsi: Rename si_shader::ls_rsrc{1,2} to si_shader::rsrc{1,2}
- radeonsi/compute: Use the compiler's COMPUTE_PGM_RSRC* register values

4.149 Mesa 11.1.0 Release Notes / 15 December 2015

Mesa 11.1.0 is a new development release. People who are concerned with stability and reliability should stick with a previous release or wait for Mesa 11.1.1.

Mesa 11.1.0 implements the OpenGL 4.1 API, but the version reported by glGetString(GL_VERSION) or glGetIntegerv(GL_MAJOR_VERSION) / glGetIntegerv(GL_MINOR_VERSION) depends on the particular driver being used. Some drivers don't support all the features required in OpenGL 4.1. OpenGL 4.1 is **only** available if requested at context creation because compatibility contexts are not supported.

4.149.1 SHA256 checksums

```
e3bc44be4df5e4dc728dfda7b55b1aaeadfce36eca6a367b76cc07598070cb2d  mesa-11.1.0.tar.gz
9befe03b04223eb1ede177fa8cac001e2850292c8c12a3ec9929106afad9cf1f  mesa-11.1.0.tar.xz
```

4.149.2 New features

Note: some of the new features are only available with certain drivers.

- OpenGL 3.1 support on freedreno (a3xx, a4xx)
- OpenGL 3.3 support for VMware guest VM driver (supported by Workstation 12 and Fusion 8).
- GL_AMD_performance_monitor on nv50
- GL_ARB_arrays_of_arrays on i965
- GL_ARB_blend_func_extended on freedreno (a3xx)
- GL_ARB_clear_texture on nv50, nvc0
- GL_ARB_clip_control on freedreno/a4xx
- GL_ARB_copy_image on nv50, nvc0, radeonsi
- GL_ARB_depth_clamp on freedreno/a4xx
- GL_ARB_fragment_layer_viewport on i965 (gen6+)
- GL_ARB_gpu_shader_fp64 on r600 for Cypress/Cayman/Aruba chips
- GL_ARB_gpu_shader5 on r600 for Evergreen and later chips
- GL_ARB_seamless_cubemap_per_texture on freedreno/a4xx
- GL_ARB_shader_clock on i965 (gen7+)
- GL_ARB_shader_stencil_export on i965 (gen9+)
- GL_ARB_shader_storage_buffer_object on i965
- GL_ARB_shader_texture_image_samples on i965, nv50, nvc0, r600, radeonsi
- GL_ARB_texture_barrier / GL_NV_texture_barrier on i965
- GL_ARB_texture_buffer_range on freedreno/a3xx
- GL_ARB_texture_compression_bptc on freedreno/a4xx
- GL_ARB_texture_query_lod on softpipe
- GL_ARB_texture_view on radeonsi and r600 (for evergeen and newer)

- GL_ARB_vertex_type_2_10_10_10_rev on freedreno (a3xx, a4xx)
- GL_EXT_blend_func_extended on all drivers that support the ARB version
- GL_EXT_buffer_storage implemented for when ES 3.1 support is gained
- GL_EXT_draw_elements_base_vertex on all drivers
- GL_EXT_texture_compression_rgtc / latc on freedreno (a3xx & a4xx)
- GL_KHR_debug (GLES)
- GL_NV_conditional_render on freedreno
- GL_OES_draw_elements_base_vertex on all drivers
- EGL_KHR_create_context on softpipe, llvmpipe
- EGL_KHR_gl_colorspace on softpipe, llvmpipe
- new virgl gallium driver for qemu virtio-gpu
- 16x multisampling on i965 (gen9+)
- GL_EXT_shader_samples_identical on i965.

4.149.3 Bug fixes

This list is likely incomplete.

- Bug 28130 - vbo: premature flushing breaks GL_LINE_LOOP
- Bug 38109 - i915 driver crashes if too few vertices are submitted (Mesa 7.10.2)
- Bug 49779 - Extra line segments in GL_LINE_LOOP
- Bug 55552 - Compile errors with –enable-mangling
- Bug 71789 - [r300g] Visuals not found in (default) depth = 24
- Bug 79783 - Distorted output in obs-studio where other vendors "work"
- Bug 80821 - When LIBGL_ALWAYS_SOFTWARE is set, KHR_create_context is not supported
- Bug 81174 - Gallium: GL_LINE_LOOP broken with more than 512 points
- Bug 83508 - [UBO] Assertion for array of blocks
- Bug 84677 - Triangle disappears with glPolygonMode GL_LINE
- Bug 86281 - brw_meta_fast_clear (brw=brw@entry=0x7fffd4097a08, fb=fb@entry=0x7fffd40fa900, buffers=buffers@entry=2, partial_clear=partial_clear@entry=false)
- Bug 86469 - Unreal Engine demo doesn't run
- Bug 86720 - [radeon] Europa Universalis 4 freezing during game start (10.3.3+, still broken on 11.0.2)
- Bug 89014 - PIPE_QUERY_GPU_FINISHED is not acting as expected on SI
- Bug 90175 - [hsw bisected][PATCH] atomic counters doesn't work for a binding point different to zero
- Bug 90348 - Spilling failure of b96 merged value
- Bug 90631 - Compilation failure for fragment shader with many branches on Sandy Bridge
- Bug 90734 - glBufferSubData is corrupting data when buffer is > 32k
- Bug 90887 - PhiMovesPass in register allocator broken

- Bug 91044 - piglit spec/egl_khr_create_context/valid debug flag gles* fail
- Bug 91114 - ES3-CTS.gtf.GL3Tests.shadow.shadow_execution_vert fails
- Bug 91254 - (regresion) video using VA-API on Intel slow and freeze system with mesa 10.6 or 10.6.1
- Bug 91292 - [BDW+] glVertexAttribDivisor not working in combination with glPolygonMode
- Bug 91342 - Very dark textures on some objects in indoors environments in Postal 2
- Bug 91526 - World of Warcraft (on Wine) has UI corruption with nouveau
- Bug 91551 - DXTn compressed normal maps produce severe artifacts on all NV5x and NVDx chipsets
- Bug 91596 - EGL_KHR_gl_colorspace (v2) causes problem with Android-x86 GUI
- Bug 91716 - [bisected] piglit.shaders.glsl-vs-int-attrib regresses on 32 bit BYT, HSW, IVB, SNB
- Bug 91718 - piglit.spec.arb_shader_image_load_store.invalid causes intermittent GPU HANG
- Bug 91719 - [SNB,HSW,BYT] dEQP regressions associated with using NIR for vertex shaders
- Bug 91726 - R600 asserts in tgsi_cmp/make_src_for_op3
- Bug 91780 - Rendering issues with geometry shader
- Bug 91785 - make check DispatchSanity_test.GLES31 regression
- Bug 91788 - [HSW Regression] Synmark2_v6 Multithread performance case FPS reduced by 36%
- Bug 91847 - glGenerateTextureMipmap not working (no errors) unless glActiveTexture(GL_TEXTURE1) is called before
- Bug 91857 - Mesa 10.6.3 linker is slow
- Bug 91881 - regression: GPU lockups since mesa-11.0.0_rc1 on RV620 (r600) driver
- Bug 91890 - [nve7] witcher2: blurry image & DATA_ERRORs (class 0xa097 mthd 0x2380/0x238c)
- Bug 91898 - src/util/mesa-sha1.c:250:25: fatal error: openssl/sha.h: No such file or directory
- Bug 91927 - [SKL] [regression] piglit compressed textures tests fail with kernel upgrade
- Bug 91930 - Program with GtkGLArea widget does not redraw
- Bug 91970 - [BSW regression] dEQP-GLES3.functional.shaders.precision.int.highp_mul_vertex
- Bug 91985 - [regression, bisected] FTBFS with commit f9caabe8f1: R600_UCP_CONST_BUFFER is undefined
- Bug 91993 - Graphical glitch in Astromenace (open-source game).
- Bug 92009 - ES3-CTS.gtf.GL3Tests.packed_pixels.packed_pixels fails
- Bug 92033 - [SNB,regression,dEQP,bisected] functional.shaders.random tests regressed
- Bug 92052 - nir/nir_builder.h:79: error: expected primary-expression before '.' token
- Bug 92054 - make check gbm-symbols-check regression
- Bug 92066 - [ILK,G45,regression] New assertion on BRW_MAX_MRF breaks ilk and g45
- Bug 92072 - Wine breakage since d082c5324 (st/mesa: don't call st_validate_state in BlitFramebuffer)
- Bug 92095 - [Regression, bisected] arb_shader_atomic_counters.compiler.builtins.frag
- Bug 92122 - [bisected, cts] Regression with Assault Android Cactus
- Bug 92124 - shader_query.cpp:841:34: error: 'strndup' was not declared in this scope
- Bug 92183 - linker.cpp:3187:46: error: 'strtok_r' was not declared in this scope

- Bug 92193 - [SKL] ES2-CTS.gtf.GL2ExtensionTests.compressed_astc_texture.compressed_astc_texture fails
- Bug 92214 - Flightgear crashes during splashboot with R600 driver, LLVM 3.7.0 and mesa 11.0.2
- Bug 92221 - Unintended code changes in _mesa_base_tex_format commit
- Bug 92265 - Black windows in weston after update mesa to 11.0.2-1
- Bug 92304 - [cts] cts.shaders.negative conformance tests fail
- Bug 92363 - [BSW/BDW] ogles1conform Gets test fails
- Bug 92437 - osmesa: Expose GL entry points for Windows build, via .def file
- Bug 92438 - Segfault in pushbuf_kref when running the android emulator (qemu) on nv50
- Bug 92476 - [cts] ES2-CTS.gtf.GL2ExtensionTests.egl_image.egl_image fails
- Bug 92588 - [HSW,BDW,BSW,SKL-Y][GLES 3.1 CTS] ES31-CTS.arrays_of_arrays.InteractionFunctionCalls2 - assert
- Bug 92621 - [G965 ILK G45] Regression: 24 piglit regressions in glsl-1.10
- Bug 92623 - Differences in prog_data ignored when caching fragment programs (causes hangs)
- Bug 92634 - gallium's vl_mpeg12_decoder does not work with st/va
- Bug 92639 - [Regression bisected] Ogles1conform mustpass.c fail
- Bug 92641 - [SKL BSW] [Regression] Ogles1conform userclip.c fail
- Bug 92645 - kodi vdpau interop fails since mesa,meta: move gl_texture_object::TargetIndex initializations
- Bug 92705 - [clover] fail to build with llvm-svn/clang-svn 3.8
- Bug 92709 - "LLVM triggered Diagnostic Handler: unsupported call to function ldexpf in main" when starting race in stuntrally
- Bug 92738 - Randon R7 240 doesn't work on 16KiB page size platform
- Bug 92744 - [g965 Regression bisected] Performance regression and piglit assertions due to liveness analysis
- Bug 92770 - [SNB, regression, dEQP] deqp-gles3.functional.shaders.discard.dynamic_loop_texture
- Bug 92824 - [regression, bisected] 'make check' dispatch-sanity broken by GL_EXT_buffer_storage
- Bug 92849 - [IVB HSW BDW] piglit image load/store load-from-cleared-image.shader_test fails
- Bug 92859 - [regression, bisected] validate_intrinsic_instr: Assertion triggered
- Bug 92860 - [radeonsi][bisected] st/mesa: implement ARB_copy_image - Corruption in ARK Survival Evolved
- Bug 92900 - [regression bisected] About 700 piglit regressions is what could go wrong
- Bug 92909 - Offset/alignment issue with layout std140 and vec3
- Bug 92985 - Mac OS X build error "ar: no archive members specified"
- Bug 93015 - Tonga Elemental segfault + VM faults since radeon: implement r600_query_hw_get_result via function pointers
- Bug 93048 - [CTS regression] mesa af2723 breaks GL Conformance for debug extension
- Bug 93063 - drm_helper.h:227:1: error: static declaration of 'pipe_virgl_create_screen' follows non-static declaration
- Bug 93091 - [opencl] segfault when running any opencl programs (like clinfo)
- Bug 93126 - wrongly claim supporting GL_EXT_texture_rg

- Bug 93180 - [regression] arb_separate_shader_objects.active sampler conflict fails
- Bug 93235 - [regression] dispatch sanity broken by GetPointerv
- Bug 93266 - gl_arb_shading_language_420pack does not allow binding of image variables

4.149.4 Changes

- MPEG4 decoding has been disabled by default in the VAAPI driver

4.150 Mesa 11.0.7 Release Notes / December 9, 2015

Mesa 11.0.7 is a bug fix release which fixes bugs found since the 11.0.6 release.

Mesa 11.0.7 implements the OpenGL 4.1 API, but the version reported by glGetString(GL_VERSION) or glGetIntegerv(GL_MAJOR_VERSION) / glGetIntegerv(GL_MINOR_VERSION) depends on the particular driver being used. Some drivers don't support all the features required in OpenGL 4.1. OpenGL 4.1 is **only** available if requested at context creation because compatibility contexts are not supported.

4.150.1 SHA256 checksums

```
07c27004ff68b288097d17b2faa7bdf15ec73c96b7e6c9835266e544adf0a62f  mesa-11.0.7.tar.gz
e7e90a332ede6c8fd08eff90786a3fd1605a4e62ebf3a9b514047838194538cb  mesa-11.0.7.tar.xz
```

4.150.2 New features

None

4.150.3 Bug fixes

This list is likely incomplete.

- Bug 90348 - Spilling failure of b96 merged value
- Bug 92363 - [BSW/BDW] ogles1conform Gets test fails
- Bug 92438 - Segfault in pushbuf_kref when running the android emulator (qemu) on nv50
- Bug 93110 - [NVE4] textureSize() and textureQueryLevels() uses a texture bound during the previous draw call
- Bug 93126 - wrongly claim supporting GL_EXT_texture_rg

4.150.4 Changes

Chris Wilson (1):

- meta: Compute correct buffer size with SkipRows/SkipPixels

Daniel Stone (1):

- egl/wayland: Ignore rects from SwapBuffersWithDamage

Dave Airlie (4):

- texgetimage: consolidate 1D array handling code.
- r600: geometry shader gsvs itemsize workaround
- r600: rv670 use at least 16es/gs threads
- r600: workaround empty geom shader.

Emil Velikov (4):

- docs: add sha256 checksums for 11.0.6
- get-pick-list.sh: Require explicit "11.0" for nominating stable patches
- mesa; add get-extra-pick-list.sh script into bin/
- Update version to 11.0.7

François Tigeot (1):

- xmlconfig: Add support for DragonFly

Ian Romanick (22):

- mesa: Make bind_vertex_buffer avilable outside varray.c
- mesa: Refactor update_array_format to make _mesa_update_array_format_public
- mesa: Refactor enable_vertex_array_attrib to make _mesa_enable_vertex_array_attrib
- i965: Pass brw_context instead of gl_context to brw_draw_rectlist
- i965: Use DSA functions for VBOs in brw_meta_fast_clear
- i965: Use internal functions for buffer object access
- i965: Don't pollute the buffer object namespace in brw_meta_fast_clear
- meta: Use DSA functions for PBO in create_texture_for_pbo
- meta: Use _mesa_NamedBufferData and _mesa_NamedBufferSubData for users of _mesa_meta_setup_vertex_objects
- i965: Use _mesa_NamedBufferSubData for users of _mesa_meta_setup_vertex_objects
- meta: Don't leave the VBO bound after _mesa_meta_setup_vertex_objects
- meta: Track VBO using gl_buffer_object instead of GL API object handle
- meta: Use DSA functions for VBOs in _mesa_meta_setup_vertex_objects
- meta: Use internal functions for buffer object and VAO access
- meta: Don't pollute the buffer object namespace in _mesa_meta_setup_vertex_objects
- meta: Partially convert _mesa_meta_DrawTex to DSA
- meta: Track VBO using gl_buffer_object instead of GL API object handle in _mesa_meta_DrawTex
- meta: Use internal functions for buffer object and VAO access in _mesa_meta_DrawTex
- meta: Don't pollute the buffer object namespace in _mesa_meta_DrawTex
- meta/TexSubImage: Don't pollute the buffer object namespace
- meta/generate_mipmap: Don't leak the framebuffer object
- glsl: Fix off-by-one error in array size check assertion

Ilia Mirkin (7):

- nvc0/ir: actually emit AFETCH on kepler
- nir: fix typo in idiv lowering, causing large-udiv-udiv failures
- nouveau: use the buffer usage to determine placement when no binding
- nv50,nvc0: properly handle buffer storage invalidation on dsa buffer
- nv50/ir: fix (un)spilling of 3-wide results
- mesa: support GL_RED/GL_RG in ES2 contexts when driver support exists
- nvc0/ir: start offset at texBindBase for txq, like regular texturing

Jonathan Gray (1):

- automake: fix some occurrences of hardcoded -ldl and -lpthread

Leo Liu (1):

- radeon/vce: disable Stoney VCE for 11.0

Marta Lofstedt (1):

- gles2: Update gl2ext.h to revision: 32120

Oded Gabbay (1):

- llvmpipe: disable VSX in ppc due to LLVM PPC bug

4.151 Mesa 11.0.6 Release Notes / November 21, 2015

Mesa 11.0.6 is a bug fix release which fixes bugs found since the 11.0.5 release.

Mesa 11.0.6 implements the OpenGL 4.1 API, but the version reported by glGetString(GL_VERSION) or glGetIntegerv(GL_MAJOR_VERSION) / glGetIntegerv(GL_MINOR_VERSION) depends on the particular driver being used. Some drivers don't support all the features required in OpenGL 4.1. OpenGL 4.1 is **only** available if requested at context creation because compatibility contexts are not supported.

4.151.1 SHA256 checksums

```
4bdf054af66ebabf3eca0616f9f5e44c2f234695661b570261c391bc2f4f7482  mesa-11.0.6.tar.gz
8340e64cdc91999840404c211496f3de38e7b4cb38db34e2f72f1642c5134760  mesa-11.0.6.tar.xz
```

4.151.2 New features

None

4.151.3 Bug fixes

This list is likely incomplete.

- Bug 91780 - Rendering issues with geometry shader
- Bug 92588 - [HSW,BDW,BSW,SKL-Y][GLES 3.1 CTS] ES31-CTS.arrays_of_arrays.InteractionFunctionCalls2 - assert
- Bug 92738 - Randon R7 240 doesn't work on 16KiB page size platform

- Bug 92860 - [radeonsi][bisected] st/mesa: implement ARB_copy_image - Corruption in ARK Survival Evolved
- Bug 92900 - [regression bisected] About 700 piglit regressions is what could go wrong

4.151.4 Changes

Alex Deucher (1):

- radeonsi: enable optimal raster config setting for fiji (v2)

Ben Widawsky (1):

- i965/skl/gt4: Fix URB programming restriction.

Boyuan Zhang (2):

- st/vaapi: fix vaapi VC-1 simple/main corruption v2
- radeon/uvd: fix VC-1 simple/main profile decode v2

Dave Airlie (1):

- r600: initialised PGM_RESOURCES_2 for ES/GS

Emil Velikov (4):

- docs: add sha256 checksums for 11.0.5
- cherry-ignore: add the swrast front buffer support
- automake: use static llvm for make distcheck
- Update version to 11.0.6

Eric Anholt (3):

- vc4: Return GL_OUT_OF_MEMORY when buffer allocation fails.
- vc4: Return NULL when we can't make our shadow for a sampler view.
- vc4: Add support for nir_op_uge, using the carry bit on QPU_A_SUB.

Ian Romanick (2):

- meta/generate_mipmap: Don't leak the sampler object
- meta/generate_mipmap: Only modify the draw framebuffer binding in fallback_required

Ilia Mirkin (2):

- mesa/copyimage: allow width/height to not be multiples of block
- nouveau: don't expose HEVC decoding support

Jason Ekstrand (1):

- nir/vars_to_ssa: Rework copy set handling in lower_copies_to_load_store

Kenneth Graunke (1):

- glsl: Allow implicit int -> uint conversions for the % operator.

Marek Olšák (1):

- radeonsi: initialize SX_PS_DOWNCONVERT to 0 on Stoney

Michel Dänzer (1):

- winsys/radeon: Use CPU page size instead of hardcoding 4096 bytes v3

Oded Gabbay (1):

- llvmpipe: use simple coeffs calc for 128bit vectors

Roland Scheidegger (2):

- radeon: fix bgrx8/xrgb8 blits
- r200: fix bgrx8/xrgb8 blits

4.152 Mesa 11.0.5 Release Notes / November 11, 2015

Mesa 11.0.5 is a bug fix release which fixes bugs found since the 11.0.4 release.

Mesa 11.0.5 implements the OpenGL 4.1 API, but the version reported by glGetString(GL_VERSION) or glGetIntegerv(GL_MAJOR_VERSION) / glGetIntegerv(GL_MINOR_VERSION) depends on the particular driver being used. Some drivers don't support all the features required in OpenGL 4.1. OpenGL 4.1 is **only** available if requested at context creation because compatibility contexts are not supported.

4.152.1 SHA256 checksums

```
8495ef5c06f7f726452462b7d408a5b40048373ff908f2283a3b4d1f49b45ee6  mesa-11.0.5.tar.gz
9c255a2a6695fcc6ef4a279e1df0aeaf417dc142f39ee59dfb533d80494bb67a  mesa-11.0.5.tar.xz
```

4.152.2 New features

None

4.152.3 Bug fixes

This list is likely incomplete.

- Bug 91993 - Graphical glitch in Astromenace (open-source game).
- Bug 92214 - Flightgear crashes during splashboot with R600 driver, LLVM 3.7.0 and mesa 11.0.2
- Bug 92437 - osmesa: Expose GL entry points for Windows build, via .def file
- Bug 92476 - [cts] ES2-CTS.gtf.GL2ExtensionTests.egl_image.egl_image fails
- Bug 92623 - Differences in prog_data ignored when caching fragment programs (causes hangs)

4.152.4 Changes

Alex Deucher (1):

- radeon/uvd: don't expose HEVC on old UVD hw (v3)

Ben Widawsky (1):

- i965/skl: Add GT4 PCI IDs

Emil Velikov (4):

- docs: add sha256 checksums for 11.0.4

- cherry-ignore: ignore a possible wrong nomination
- Revert "mesa/glformats: Undo code changes from _mesa_base_tex_format() move"
- Update version to 11.0.5

Emmanuel Gil Peyrot (1):

- gbm.h: Add a missing stddef.h include for size_t.

Eric Anholt (1):

- vc4: When the create ioctl fails, free our cache and try again.

Ian Romanick (1):

- i965: Fix is-renderable check in intel_image_target_renderbuffer_storage

Ilia Mirkin (3):

- nvc0: respect edgeflag attribute width
- nouveau: set MaxDrawBuffers to the same value as MaxColorAttachments
- nouveau: relax fence emit space assert

Ivan Kalvachev (1):

- r600g: Fix special negative immediate constants when using ABS modifier.

Jason Ekstrand (2):

- nir/lower_vec_to_movs: Pass the shader around directly
- nir: Report progress from lower_vec_to_movs().

Jose Fonseca (2):

- gallivm: Translate all util_cpu_caps bits to LLVM attributes.
- gallivm: Explicitly disable unsupported CPU features.

Julien Isorce (4):

- st/va: pass picture desc to begin and decode
- nvc0: fix crash when nv50_miptree_from_handle fails
- st/va: do not destroy old buffer when new one failed
- st/va: add more errors checks in vlVaBufferSetNumElements and vlVaMapBuffer

Kenneth Graunke (6):

- i965: Fix missing BRW_NEW_*_PROG_DATA flagging caused by cache reuse.
- nir: Report progress from nir_split_var_copies().
- nir: Properly invalidate metadata in nir_split_var_copies().
- nir: Properly invalidate metadata in nir_opt_copy_prop().
- nir: Properly invalidate metadata in nir_lower_vec_to_movs().
- nir: Properly invalidate metadata in nir_opt_remove_phis().

Marek Olšák (1):

- radeonsi: add register definitions for Stoney

Nanley Chery (1):

- mesa/glformats: Undo code changes from _mesa_base_tex_format() move

Nicolai Hähnle (1):

- st/mesa: fix mipmap generation for immutable textures with incomplete pyramids

Nigel Stewart (1):

- osmesa: Expose GL entry points for Windows build via DEF file.

Roland Scheidegger (1):

- gallivm: disable f16c when not using AVX

Samuel Li (2):

- radeonsi: add support for Stoney asics (v3)
- radeonsi: add Stoney pci ids

4.153 Mesa 11.0.4 Release Notes / October 24, 2015

Mesa 11.0.4 is a bug fix release which fixes bugs found since the 11.0.3 release.

Mesa 11.0.4 implements the OpenGL 4.1 API, but the version reported by glGetString(GL_VERSION) or glGetIntegerv(GL_MAJOR_VERSION) / glGetIntegerv(GL_MINOR_VERSION) depends on the particular driver being used. Some drivers don't support all the features required in OpenGL 4.1. OpenGL 4.1 is **only** available if requested at context creation because compatibility contexts are not supported.

4.153.1 SHA256 checksums

```
ed412ca6a46d1bd055120e5c12806c15419ae8c4dd6d3f6ea20a83091d5c78bf  mesa-11.0.4.tar.gz
40201bf7fc6fa12a6d9edfe870b41eb4dd6669154e3c42c48a96f70805f5483d  mesa-11.0.4.tar.xz
```

4.153.2 New features

None

4.153.3 Bug fixes

This list is likely incomplete.

- Bug 86281 - brw_meta_fast_clear (brw=brw@entry=0x7fffd4097a08, fb=fb@entry=0x7fffd40fa900, buffers=buffers@entry=2, partial_clear=partial_clear@entry=false)
- Bug 86720 - [radeon] Europa Universalis 4 freezing during game start (10.3.3+, still broken on 11.0.2)
- Bug 91788 - [HSW Regression] Synmark2_v6 Multithread performance case FPS reduced by 36%
- Bug 92304 - [cts] cts.shaders.negative conformance tests fail

4.153.4 Changes

Alejandro Piñeiro (2):

- i965/vec4: check writemask when bailing out at register coalesce
- i965/vec4: fill src_reg type using the constructor type parameter

Brian Paul (2):

- vbo: fix incorrect switch statement in init_mat_currval()
- mesa: fix incorrect opcode in save_BlendFunci()

Chih-Wei Huang (3):

- mesa: android: Fix the incorrect path of sse_minmax.c
- nv50/ir: use C++11 standard std::unordered_map if possible
- nv30: include the header of ffs prototype

Chris Wilson (1):

- i965: Remove early release of DRI2 miptree

Dave Airlie (1):

- mesa/uniforms: fix get_uniform for doubles (v2)

Emil Velikov (1):

- docs: add sha256 checksums for 11.0.3

Francisco Jerez (5):

- i965: Don't tell the hardware about our UAV access.
- mesa: Expose function to calculate whether a shader image unit is valid.
- mesa: Skip redundant texture completeness checking during image validation.
- i965: Use _mesa_is_image_unit_valid() instead of gl_image_unit::_Valid.
- mesa: Get rid of texture-dependent image unit derived state.

Ian Romanick (8):

- glsl: Allow built-in functions as constant expressions in OpenGL ES 1.00
- ff_fragment_shader: Use binding to set the sampler unit
- glsl/linker: Use constant_initializer instead of constant_value to initialize uniforms
- glsl: Use constant_initializer instead of constant_value to determine whether to keep an unused uniform
- glsl: Only set ir_variable::constant_value for const-decorated variables
- glsl: Restrict initializers for global variables to constant expression in ES
- glsl: Add method to determine whether an expression contains the sequence operator
- glsl: In later GLSL versions, sequence operator is cannot be a constant expression

Ilia Mirkin (1):

- nouveau: make sure there's always room to emit a fence

Indrajit Das (1):

- st/va: Used correct parameter to derive the value of the "h" variable in vlVaCreateImage

Jonathan Gray (1):

- configure.ac: ensure RM is set

Krzysztof Sobiecki (1):

- st/fbo: use pipe_surface_release instead of pipe_surface_reference

Leo Liu (1):

- st/omx/dec/h264: fix field picture type 0 poc disorder

Marek Olšák (3):

- st/mesa: fix clip state dependencies
- radeonsi: fix a GS copy shader leak
- gallium: add PIPE_SHADER_CAP_MAX_UNROLL_ITERATIONS_HINT

Nicolai Hähnle (1):

- u_vbuf: fix vb slot assignment for translated buffers

Rob Clark (1):

- freedreno/a3xx: cache-flush is needed after MEM_WRITE

Tapani Pälli (3):

- mesa: add GL_UNSIGNED_INT_24_8 to _mesa_pack_depth_span
- mesa: Set api prefix to version string when overriding version
- mesa: fix ARRAY_SIZE query for GetProgramResourceiv

4.154 Mesa 11.0.3 Release Notes / October 10, 2015

Mesa 11.0.3 is a bug fix release which fixes bugs found since the 11.0.2 release.

Mesa 11.0.3 implements the OpenGL 4.1 API, but the version reported by glGetString(GL_VERSION) or glGetIntegerv(GL_MAJOR_VERSION) / glGetIntegerv(GL_MINOR_VERSION) depends on the particular driver being used. Some drivers don't support all the features required in OpenGL 4.1. OpenGL 4.1 is **only** available if requested at context creation because compatibility contexts are not supported.

4.154.1 SHA256 checksums

```
c2210e3daecc10ed9fdcea500327652ed6effc2f47c4b9cee63fb08f560d7117  mesa-11.0.3.tar.gz
ab2992eece21adc23c398720ef8c6933cb69ea42e1b2611dc09d031e17e033d6  mesa-11.0.3.tar.xz
```

4.154.2 New features

None

4.154.3 Bug fixes

This list is likely incomplete.

- Bug 55552 - Compile errors with –enable-mangling
- Bug 71789 - [r300g] Visuals not found in (default) depth = 24
- Bug 91044 - piglit spec/egl_khr_create_context/valid debug flag gles* fail
- Bug 91342 - Very dark textures on some objects in indoors environments in Postal 2
- Bug 91596 - EGL_KHR_gl_colorspace (v2) causes problem with Android-x86 GUI
- Bug 91718 - piglit.spec.arb_shader_image_load_store.invalid causes intermittent GPU HANG
- Bug 92072 - Wine breakage since d082c5324 (st/mesa: don't call st_validate_state in BlitFramebuffer)
- Bug 92265 - Black windows in weston after update mesa to 11.0.2-1

4.154.4 Changes

Brian Paul (1):

- st/mesa: try PIPE_BIND_RENDER_TARGET when choosing float texture formats

Daniel Scharrer (1):

- mesa: Add abs input modifier to base for POW in ffvertex_prog

Emil Velikov (3):

- docs: add sha256 checksums for 11.0.2
- Revert "nouveau: make sure there's always room to emit a fence"
- Update version to 11.0.3

Francisco Jerez (1):

- i965/fs: Fix hang on IVB and VLV with image format mismatch.

Ian Romanick (1):

- meta: Handle array textures in scaled MSAA blits

Ilia Mirkin (6):

- nouveau: be more careful about freeing temporary transfer buffers
- nouveau: delay deleting buffer with unflushed fence
- nouveau: wait to unref the transfer's bo until it's no longer used
- nv30: pretend to have packed texture/surface formats
- nv30: always go through translate module on big-endian
- nouveau: make sure there's always room to emit a fence

Jason Ekstrand (1):

- mesa: Correctly handle GL_BGRA_EXT in ES3 format_and_type checks

Kyle Brenneman (3):

- glx: Fix build errors with –enable-mangling (v2)

- mapi: Make _glapi_get_stub work with "gl" or "mgl" prefix.
- glx: Don't hard-code the name "libGL.so.1" in driOpenDriver (v3)

Leo Liu (1):

- radeon/vce: fix vui time_scale zero error

Marek Olšák (21):

- st/mesa: fix front buffer regression after dropping st_validate_state in Blit
- radeonsi: handle index buffer alloc failures
- radeonsi: handle constant buffer alloc failures
- gallium/radeon: handle buffer_map staging buffer failures better
- gallium/radeon: handle buffer alloc failures in r600_draw_rectangle
- gallium/radeon: add a fail path for depth MSAA texture readback
- radeonsi: report alloc failure from si_shader_binary_read
- radeonsi: add malloc fail paths to si_create_shader_state
- radeonsi: skip drawing if the tess factor ring allocation fails
- radeonsi: skip drawing if GS ring allocations fail
- radeonsi: handle shader precompile failures
- radeonsi: handle fixed-func TCS shader create failure
- radeonsi: skip drawing if VS, TCS, TES, GS fail to compile or upload
- radeonsi: skip drawing if PS fails to compile or upload
- radeonsi: skip drawing if updating the scratch buffer fails
- radeonsi: don't forget to update scratch relocations for LS, HS, ES shaders
- radeonsi: handle dummy constant buffer allocation failure
- gallium/u_blitter: handle allocation failures
- radeonsi: add scratch buffer to the buffer list when it's re-allocated
- st/dri: don't use _ctx in client_wait_sync
- egl/dri2: don't require a context for ClientWaitSync (v2)

Matthew Waters (1):

- egl: rework handling EGL_CONTEXT_FLAGS

Michel Dänzer (1):

- st/dri: Use packed RGB formats

Roland Scheidegger (1):

- mesa: fix mipmap generation for immutable, compressed textures

Tom Stellard (3):

- gallium/radeon: Use call_once() when initailizing LLVM targets
- gallivm: Allow drivers and state trackers to initialize gallivm LLVM targets v2
- radeon/llvm: Initialize gallivm targets when initializing the AMDGPU target v2

Varad Gautam (1):

- egl: restore surface type before linking config to its display

Ville Syrjälä (3):

- i830: Fix collision between I830_UPLOAD_RASTER_RULES and I830_UPLOAD_TEX(0)
- i915: Fix texcoord vs. varying collision in fragment programs
- i915: Remember to call intel_prepare_render() before blitting

4.155 Mesa 10.6.9 Release Notes / Octover 03, 2015

Mesa 10.6.9 is a bug fix release which fixes bugs found since the 10.6.8 release.

Mesa 10.6.9 implements the OpenGL 3.3 API, but the version reported by glGetString(GL_VERSION) or glGetIntegerv(GL_MAJOR_VERSION) / glGetIntegerv(GL_MINOR_VERSION) depends on the particular driver being used. Some drivers don't support all the features required in OpenGL 3.3. OpenGL 3.3 is **only** available if requested at context creation because compatibility contexts are not supported.

4.155.1 SHA256 checksums

```
3406876aac67546d0c3e2cb97da330b62644c313e7992b95618662e13c54296a  mesa-10.6.9.tar.gz
b04c4de6280b863babc2929573da17218d92e9e4ba6272d548d135415723e8c3  mesa-10.6.9.tar.xz
```

4.155.2 New features

None

4.155.3 Bug fixes

This list is likely incomplete.

- Bug 38109 - i915 driver crashes if too few vertices are submitted (Mesa 7.10.2)
- Bug 55552 - Compile errors with –enable-mangling
- Bug 86281 - brw_meta_fast_clear (brw=brw@entry=0x7fffd4097a08, fb=fb@entry=0x7fffd40fa900, buffers=buffers@entry=2, partial_clear=partial_clear@entry=false)
- Bug 91970 - [BSW regression] dEQP-GLES3.functional.shaders.precision.int.highp_mul_vertex
- Bug 92072 - Wine breakage since d082c5324 (st/mesa: don't call st_validate_state in BlitFramebuffer)

4.155.4 Changes

Brian Paul (1):

- st/mesa: try PIPE_BIND_RENDER_TARGET when choosing float texture formats

Chris Wilson (1):

- i965: Remove early release of DRI2 miptree

Emil Velikov (4):

- docs: add sha256 checksums for 10.6.8
- cherry-ignore: add commit non applicable for 10.6
- cherry-ignore: add commit non applicable for 10.6
- Update version to 10.6.9

Iago Toral Quiroga (1):

- mesa: Fix GL_FRAMEBUFFER_ATTACHMENT_OBJECT_TYPE for default framebuffer.

Ian Romanick (5):

- t_dd_dmatmp: Make "count" actually be the count
- t_dd_dmatmp: Clean up improper code formatting from previous patch
- t_dd_dmatmp: Use '& 3' instead of '% 4' everywhere
- t_dd_dmatmp: Pull out common 'count -= count & 3' code
- t_dd_dmatmp: Use addition instead of subtraction in loop bounds

Jeremy Huddleston (1):

- configure.ac: Add support to enable read-only text segment on x86.

Kristian Høgsberg Kristensen (1):

- i965: Respect stride and subreg_offset for ATTR registers

Kyle Brenneman (3):

- glx: Fix build errors with –enable-mangling (v2)
- mapi: Make _glapi_get_stub work with "gl" or "mgl" prefix.
- glx: Don't hard-code the name "libGL.so.1" in driOpenDriver (v3)

Leo Liu (1):

- radeon/vce: fix vui time_scale zero error

Marek Olšák (1):

- st/mesa: fix front buffer regression after dropping st_validate_state in Blit

Roland Scheidegger (1):

- mesa: fix mipmap generation for immutable, compressed textures

4.156 Mesa 11.0.2 Release Notes / September 28, 2015

Mesa 11.0.2 is a bug fix release which fixes bugs found since the 11.0.1 release.

Mesa 11.0.2 implements the OpenGL 4.1 API, but the version reported by glGetString(GL_VERSION) or glGetIntegerv(GL_MAJOR_VERSION) / glGetIntegerv(GL_MINOR_VERSION) depends on the particular driver being used. Some drivers don't support all the features required in OpenGL 4.1. OpenGL 4.1 is **only** available if requested at context creation because compatibility contexts are not supported.

4.156.1 SHA256 checksums

```
45170773500d6ae2f9eb93fc85efee69f7c97084411ada4eddf92f78bca56d20  mesa-11.0.2.tar.gz
fce11fb27eb87adf1e620a76455d635c6136dfa49ae58c53b34ef8d0c7b7eae4  mesa-11.0.2.tar.xz
```

4.156.2 New features

None

4.156.3 Bug fixes

This list is likely incomplete.

- Bug 91582 - [bisected] Regression in DEQP gles2.functional.negative_api.texture.texsubimage2d_neg_offset
- Bug 91970 - [BSW regression] dEQP-GLES3.functional.shaders.precision.int.highp_mul_vertex
- Bug 92095 - [Regression, bisected] arb_shader_atomic_counters.compiler.builtins.frag

4.156.4 Changes

Eduardo Lima Mitev (3):

- mesa: Fix order of format+type and internal format checks for glTexImageXD ops
- mesa: Move _mesa_base_tex_format() from teximage to glformats files
- mesa: Use the effective internal format instead for validation

Emil Velikov (2):

- docs: add sha256 checksums for 11.0.1
- Update version to 11.0.2

Kristian Høgsberg Kristensen (1):

- i965: Respect stride and subreg_offset for ATTR registers

Matt Turner (1):

- glsl: Expose gl_MaxTess{Control,Evaluation}AtomicCounters.

4.157 Mesa 11.0.1 Release Notes / September 26, 2015

Mesa 11.0.1 is a bug fix release which fixes bugs found since the 11.0.0 release.

Mesa 11.0.1 implements the OpenGL 4.1 API, but the version reported by glGetString(GL_VERSION) or glGetIntegerv(GL_MAJOR_VERSION) / glGetIntegerv(GL_MINOR_VERSION) depends on the particular driver being used. Some drivers don't support all the features required in OpenGL 4.1. OpenGL 4.1 is **only** available if requested at context creation because compatibility contexts are not supported.

4.157.1 SHA256 checksums

```
6dab262877e12c0546a0e2970c6835a0f217e6d4026ccecb3cd5dd733d1ce867  mesa-11.0.1.tar.gz
43d0dfcd1f1e36f07f8228cd76d90175d3fc74c1ed25d7071794a100a98ef2a6  mesa-11.0.1.tar.xz
```

4.157.2 New features

None

4.157.3 Bug fixes

This list is likely incomplete.

- Bug 38109 - i915 driver crashes if too few vertices are submitted (Mesa 7.10.2)
- Bug 91114 - ES3-CTS.gtf.GL3Tests.shadow.shadow_execution_vert fails
- Bug 91716 - [bisected] piglit.shaders.glsl-vs-int-attrib regresses on 32 bit BYT, HSW, IVB, SNB
- Bug 91719 - [SNB,HSW,BYT] dEQP regressions associated with using NIR for vertex shaders
- Bug 92009 - ES3-CTS.gtf.GL3Tests.packed_pixels.packed_pixels fails

4.157.4 Changes

Antia Puentes (2):

- i965/vec4: Fix saturation errors when coalescing registers
- i965/vec4_nir: Load constants as integers

Anuj Phogat (1):

- meta: Abort meta pbo path if TexSubImage need signed unsigned conversion

Emil Velikov (2):

- docs: add sha256 checksums for 11.0.0
- Update version to 11.0.1

Iago Toral Quiroga (1):

- mesa: Fix GL_FRAMEBUFFER_ATTACHMENT_OBJECT_TYPE for default framebuffer.

Ian Romanick (5):

- t_dd_dmatmp: Make "count" actually be the count
- t_dd_dmatmp: Clean up improper code formatting from previous patch
- t_dd_dmatmp: Use '& 3' instead of '% 4' everywhere
- t_dd_dmatmp: Pull out common 'count -= count & 3' code
- t_dd_dmatmp: Use addition instead of subtraction in loop bounds

Ilia Mirkin (6):

- st/mesa: avoid integer overflows with buffers >= 512MB
- nv50, nvc0: fix max texture buffer size to 128M elements

- freedreno/a3xx: fix blending of L8 format
- nv50,nvc0: detect underlying resource changes and update tic
- nv50,nvc0: flush texture cache in presence of coherent bufs
- radeonsi: load fmask ptr relative to the resources array

Jason Ekstrand (2):

- nir: Fix a bunch of ralloc parenting errors
- i965/vec4: Don't reswizzle hardware registers

Jeremy Huddleston (1):

- configure.ac: Add support to enable read-only text segment on x86.

Ray Strode (1):

- gbm: convert gbm bo format to fourcc format on dma-buf import

Tapani Pälli (2):

- mesa: fix errors when reading depth with glReadPixels
- i965: fix textureGrad for cubemaps

Ulrich Weigand (1):

- mesa: Fix texture compression on big-endian systems

4.158 Mesa 10.6.8 Release Notes / September 20, 2015

Mesa 10.6.8 is a bug fix release which fixes bugs found since the 10.6.7 release.

Mesa 10.6.8 implements the OpenGL 3.3 API, but the version reported by glGetString(GL_VERSION) or glGetIntegerv(GL_MAJOR_VERSION) / glGetIntegerv(GL_MINOR_VERSION) depends on the particular driver being used. Some drivers don't support all the features required in OpenGL 3.3. OpenGL 3.3 is **only** available if requested at context creation because compatibility contexts are not supported.

4.158.1 SHA256 checksums

```
1f34dba2a8059782e3e4e0f18b9628004e253b2c69085f735b846d2e63c9e250  mesa-10.6.8.tar.gz
e36ee5ceeadb3966fb5ce5b4cf18322dbb76a4f075558ae49c3bba94f57d58fd  mesa-10.6.8.tar.xz
```

4.158.2 New features

None

4.158.3 Bug fixes

This list is likely incomplete.

- Bug 90621 - Mesa fail to build from git
- Bug 91526 - World of Warcraft (on Wine) has UI corruption with nouveau
- Bug 91719 - [SNB,HSW,BYT] dEQP regressions associated with using NIR for vertex shaders

4.158.4 Changes

Alejandro Piñeiro (1):

- i965/vec4: fill src_reg type using the constructor type parameter

Antia Puentes (1):

- i965/vec4: Fix saturation errors when coalescing registers

Emil Velikov (2):

- docs: add sha256 checksums for 10.6.7
- cherry-ignore: add commit non applicable for 10.6

Hans de Goede (4):

- nv30: Fix creation of scanout buffers
- nv30: Implement color resolve for msaa
- nv30: Fix max width / height checks in nv30 sifm code
- nv30: Disable msaa unless requested from the env by NV30_MAX_MSAA

Ian Romanick (2):

- mesa: Pass the type to _mesa_uniform_matrix as a glsl_base_type
- mesa: Don't allow wrong type setters for matrix uniforms

Ilia Mirkin (5):

- st/mesa: don't fall back to 16F when 32F is requested
- nvc0: always emit a full shader colormask
- nvc0: remove BGRA4 format support
- st/mesa: avoid integer overflows with buffers >= 512MB
- nv50, nvc0: fix max texture buffer size to 128M elements

Jason Ekstrand (1):

- i965/vec4: Don't reswizzle hardware registers

Jose Fonseca (1):

- gallivm: Workaround LLVM PR23628.

Kenneth Graunke (1):

- i965: Momentarily pretend to support ARB_texture_stencil8 for blits.

Oded Gabbay (1):

- llvmpipe: convert double to long long instead of unsigned long long

Ray Strode (1):

- gbm: convert gbm bo format to fourcc format on dma-buf import

Ulrich Weigand (1):

- mesa: Fix texture compression on big-endian systems

Vinson Lee (1):

- gallivm: Do not use NoFramePointerElim with LLVM 3.7.

4.159 Mesa 11.0.0 Release Notes / September 12, 2015

Mesa 11.0.0 is a new development release. People who are concerned with stability and reliability should stick with a previous release or wait for Mesa 11.0.1.

Mesa 11.0.0 implements the OpenGL 4.1 API, but the version reported by glGetString(GL_VERSION) or glGetIntegerv(GL_MAJOR_VERSION) / glGetIntegerv(GL_MINOR_VERSION) depends on the particular driver being used. Some drivers don't support all the features required in OpenGL 4.1. OpenGL 4.1 is **only** available if requested at context creation because compatibility contexts are not supported.

4.159.1 SHA256 checksums

```
7d7e4ddffa3b162506efa01e2cc41e329caa4995336b92e5cc21f2e1fb36c1b3  mesa-11.0.0.tar.gz
e095a3eb2eca9dfde7efca8946527c8ae20a0cc938a8c78debc7f158ad44af32  mesa-11.0.0.tar.xz
```

4.159.2 New features

Note: some of the new features are only available with certain drivers.

- New hardware support for AMD GCN 1.2 GPUs: Tonga, Iceland, Carrizo, Fiji
- OpenGL 4.1 on radeonsi, nvc0
- OpenGL ES 3.0 on freedreno (a3xx, a4xx)
- GL_AMD_vertex_shader_viewport_index on radeonsi
- GL_ARB_conditional_render_inverted on r600, radeonsi
- GL_ARB_depth_buffer_float on a4xx
- GL_ARB_derivative_control on radeonsi
- GL_ARB_draw_buffers, GL_ARB_draw_buffers_blend on a4xx
- GL_ARB_fragment_layer_viewport on radeonsi
- GL_ARB_framebuffer_no_attachments on i965
- GL_ARB_get_texture_sub_image for all drivers
- GL_ARB_gpu_shader5 on radeonsi
- GL_ARB_gpu_shader_fp64 on llvmpipe, radeonsi
- GL_ARB_shader_image_load_store on i965
- GL_ARB_shader_precision on radeonsi, nvc0
- GL_ARB_shader_image_size on i965
- GL_ARB_shader_stencil_export on llvmpipe
- GL_ARB_shader_subroutine on core profile all drivers
- GL_ARB_tessellation_shader on nvc0, radeonsi
- GL_ARB_transform_feedback2, GL_ARB_transform_feedback_instanced, GL_EXT_transform_feedback on a3xx, a4xx
- GL_ARB_vertex_attrib_64bit on llvmpipe, radeonsi

- GL_ARB_viewport_array on radeonsi
- GL_EXT_depth_bounds_test on radeonsi, nv30, nv50, nvc0
- GL_EXT_texture_compression_s3tc on freedreno (a3xx)
- GL_NV_read_depth (GLES) on all drivers
- GL_NV_read_depth_stencil (GLES) on all drivers
- GL_NV_read_stencil (GLES) on all drivers
- GL_OES_texture_float on all r300, r600, radeonsi, nv30, nv50, nvc0, softpipe, llvmpipe
- GL_OES_texture_half_float on all r300, r600, radeonsi, nv30, nv50, nvc0, softpipe, llvmpipe
- GL_OES_texture_float_linear on all r300, r600, radeonsi, nv30, nv50, nvc0, softpipe, llvmpipe
- GL_OES_texture_half_float_linear on all r300, r600, radeonsi, nv30, nv50, nvc0, softpipe, llvmpipe
- GL_EXT_draw_buffers2 on a4xx
- GLX_ARB_create_context_robustness on r600, radeonsi
- EGL_EXT_create_context_robustness on r600, radeonsi
- EGL_KHR_gl_colorspace on r600, radeonsi, nv50, nvc0
- EGL_KHR_gl_texture_3D_image on r600, radeonsi, nv50, nvc0
- EGL 1.5 on r600, radeonsi, nv50, nvc0

4.159.3 Bug fixes

This list is likely incomplete.

- Bug 51658 - r200 (& possibly radeon) DRI fixes for gnome shell on Mesa 8.0.3
- Bug 65525 - [llvmpipe] lp_scene.h:210:lp_scene_alloc: Assertion 'size <= (64 * 1024)' failed.
- Bug 66346 - shader_query.cpp:49: error: invalid conversion from 'void*' to 'GLuint'
- Bug 73512 - [clover] mesa.icd. should contain full path
- Bug 73528 - Deferred lighting in Second Life causes system hiccups and screen flickering
- Bug 74329 - Please expose OES_texture_float and OES_texture_half_float on the ES3 context
- Bug 80500 - Flickering shadows in unreleased title trace
- Bug 82186 - [r600g] BARTS GPU lockup with minecraft shaders
- Bug 84225 - Allow constant-index-expression sampler array indexing with GLSL-ES < 300
- Bug 84677 - Triangle disappears with glPolygonMode GL_LINE
- Bug 85252 - Segfault in compiler while processing ternary operator with void arguments
- Bug 89131 - [Bisected] Graphical corruption in Weston, shows old framebuffer pieces
- Bug 90000 - [i965 Bisected NIR] Piglit/gglean_fragprog1-z-write_test fail
- Bug 90073 - Leaks in xcb_dri3_open_reply_fds() and get_render_node_from_id_path_tag
- Bug 90249 - Fails to build egl_dri2 on osx
- Bug 90310 - Fails to build gallium_dri.so at linking stage with clang because of multiple redefinitions
- Bug 90347 - [NVE0+] Failure to insert texbar under some circumstances (causing bad colors in Terasology)

- Bug 90466 - arm: linker error ndefined reference to 'nir_metadata_preserve'
- Bug 90520 - Register spilling clobbers registers used elsewhere in the shader
- Bug 90537 - radeonsi bo/va conflict on RADEON_GEM_VA (rscreen->ws->buffer_from_handle returns NULL)
- Bug 90547 - [BDW/BSW/SKL Bisected]Piglit/glean@vertprog1-rsq_test_2_(reciprocal_square_root_of_negative_value) fais
- Bug 90580 - [HSW bisected] integer multiplication bug
- Bug 90600 - IOError: [Errno 2] No such file or directory: 'gl_API.xml'
- Bug 90621 - Mesa fail to build from git
- Bug 90629 - [i965] SIMD16 dual_source_blend assertion 'src[i].file != GRF || src[i].width == dst.width' failed
- Bug 90691 - [BSW]Piglit/spec/nv_conditional_render/dlist fails intermittently
- Bug 90728 - dvd playback with vlc and vdpau causes segmentation fault
- Bug 90734 - glBufferSubData is corrupting data when buffer is > 32k
- Bug 90748 - [BDW Bisected]dEQP-GLES3.functional.fbo.completeness.renderable.texture.depth.rg_half_float_oes fails
- Bug 90749 - [BDW Bisected]dEQP-GLES3.functional.rasterization.fbo.rbo_multisample_max.primitives.lines_wide fails
- Bug 90751 - [BDW Bisected]dEQP-GLES3.functional.fbo.completeness.renderable.texture.stencil.stencil_index8 fails
- Bug 90797 - [ALL bisected] Mesa change cause performance case manhattan fail.
- Bug 90817 - swrast fails to load with certain remote X servers
- Bug 90830 - [bsw bisected regression] GPU hang for spec.arb_gpu_shader5.execution.sampler_array_indexing.vs-nonzero-base
- Bug 90839 - [10.5.5/10.6 regression, bisected] PBO glDrawPixels no longer using blit fastpath
- Bug 90873 - Kernel hang, TearFree On, Mate desktop environment
- Bug 90887 - PhiMovesPass in register allocator broken
- Bug 90895 - [IVB/HSW/BDW/BSW Bisected] GLB2.7 Egypt, GfxBench3.0 T-Rex & ALU and many SynMark cases performance reduced by 10-23%
- Bug 90902 - [bsw][regression] dEQP: "Found invalid pixel values"
- Bug 90903 - egl_dri2.c:dri2_load fails to load libglapi on osx
- Bug 90904 - OSX: EXC_BAD_ACCESS when using translate_sse + gallium + softpipe/llvmpipe
- Bug 90905 - mesa: Finish subdir-objects transition
- Bug 90925 - "high fidelity": Segfault in _mesa_program_resource_find_name
- Bug 91022 - [g45 g965 bisected] assertions generated from textureGrad cube samplers fix
- Bug 91047 - [SNB Bisected] Messed up Fog in Super Smash Bros. Melee in Dolphin
- Bug 91056 - The Bard's Tale (2005, native) has rendering issues
- Bug 91077 - dri2_glx.c:1186: undefined reference to 'loader_open_device'
- Bug 91099 - [llvmpipe] piglit glsl-max-varyings >max_varying_components regression

- Bug 91101 - [softpipe] piglit glsl-1.50@execution@geometry@max-input-components regression
- Bug 91117 - Nimbus (running in wine) has rendering issues, objects are semi-transparent
- Bug 91124 - Civilization V (in Wine) has rendering issues: text missing, menu bar corrupted
- Bug 91173 - Oddworld: Stranger's Wrath HD: disfigured models in wrong colors
- Bug 91193 - [290x] Dota2 reborn ingame rendering breaks with git-af4b9c7
- Bug 91222 - lp_test_format regression on CentOS 7
- Bug 91226 - Crash in glLinkProgram (NEW)
- Bug 91231 - [NV92] Psychonauts (native) segfaults on start when DRI3 enabled
- Bug 91254 - (regresion) video using VA-API on Intel slow and freeze system with mesa 10.6 or 10.6.1
- Bug 91290 - SIGSEGV glcpp/glcpp-parse.y:1077
- Bug 91292 - [BDW+] glVertexAttribDivisor not working in combination with glPolygonMode
- Bug 91337 - OSMesaGetProcAdress("OSMesaPixelStore") returns nil
- Bug 91418 - Visual Studio 2015 vsnprintf build error
- Bug 91425 - [regression, bisected] Piglit spec/ext_packed_float/ getteximage-invalid-format-for-packed-type fails
- Bug 91441 - make check DispatchSanity_test.GL30 regression
- Bug 91444 - regression bisected radeonsi: don't change pipe_resource in resource_copy_region
- Bug 91461 - gl_TessLevel* writes have no effect for all but the last TCS invocation
- Bug 91513 - [IVB/HSW/BDW/SKL Bisected] Lightsmark performance reduced by 7%-10%
- Bug 91526 - World of Warcraft (on Wine) has UI corruption with nouveau
- Bug 91544 - [i965, regression, bisected] regression of several tests in 93977d3a151675946c03e
- Bug 91551 - DXTn compressed normal maps produce severe artifacts on all NV5x and NVDx chipsets
- Bug 91570 - Upgrading mesa to 10.6 causes segfault in OpenGL applications with GeForce4 MX 440 / AGP 8X
- Bug 91591 - rounding.h:102:2: error: #error "Unsupported or undefined LONG_BIT"
- Bug 91610 - [BSW] GPU hang for spec.shaders.point-vertex-id gl_instanceid divisor
- Bug 91673 - Segfault when calling glTexSubImage2D on storage texture to bound FBO
- Bug 91726 - R600 asserts in tgsi_cmp/make_src_for_op3
- Bug 91847 - glGenerateTextureMipmap not working (no errors) unless glActiveTexture(GL_TEXTURE1) is called before
- Bug 91857 - Mesa 10.6.3 linker is slow
- Bug 91881 - regression: GPU lockups since mesa-11.0.0_rc1 on RV620 (r600) driver
- Bug 91890 - [nve7] witcher2: blurry image & DATA_ERRORs (class 0xa097 mthd 0x2380/0x238c)

4.159.4 Changes

- Removed the EGL loader from the Linux SCons build.

4.160 Mesa 10.6.7 Release Notes / September 10, 2015

Mesa 10.6.7 is a bug fix release which fixes bugs found since the 10.6.6 release.

Mesa 10.6.7 implements the OpenGL 3.3 API, but the version reported by glGetString(GL_VERSION) or glGetIntegerv(GL_MAJOR_VERSION) / glGetIntegerv(GL_MINOR_VERSION) depends on the particular driver being used. Some drivers don't support all the features required in OpenGL 3.3. OpenGL 3.3 is **only** available if requested at context creation because compatibility contexts are not supported.

4.160.1 SHA256 checksums

```
4ba10c59abee30d72476543a57afd2f33803dabf4620dc333b335d47966ff842  mesa-10.6.7.tar.gz
feb1f640b915dada88a7c793dfaff0ae23580f8903f87a6b76469253de0d28d8  mesa-10.6.7.tar.xz
```

4.160.2 New features

None

4.160.3 Bug fixes

This list is likely incomplete.

- Bug 90751 - [BDW Bisected]dEQP-GLES3.functional.fbo.completeness.renderable.texture.stencil.stencil_index8 fails

4.160.4 Changes

Dave Airlie (1):

- mesa/teximage: use correct extension for accept stencil texture.

Emil Velikov (3):

- docs: add sha256 checksums for 10.6.6
- Revert "i965: Momentarily pretend to support ARB_texture_stencil8 for blits."
- Update version to 10.6.7

Kenneth Graunke (1):

- glsl: Handle attribute aliasing in attribute storage limit check.

4.161 Mesa 10.6.6 Release Notes / September 04, 2015

Mesa 10.6.6 is a bug fix release which fixes bugs found since the 10.6.5 release.

Mesa 10.6.6 implements the OpenGL 3.3 API, but the version reported by glGetString(GL_VERSION) or glGetIntegerv(GL_MAJOR_VERSION) / glGetIntegerv(GL_MINOR_VERSION) depends on the particular driver being used. Some drivers don't support all the features required in OpenGL 3.3. OpenGL 3.3 is **only** available if requested at context creation because compatibility contexts are not supported.

4.161.1 SHA256 checksums

```
416517aa9df4791f97d34451a9e4da33c966afcd18c115c5769b92b15b018ef5  mesa-10.6.6.tar.gz
570f2154b7340ff5db61ff103bc6e85165b8958798b78a50fa2df488e98e5778  mesa-10.6.6.tar.xz
```

4.161.2 New features

None

4.161.3 Bug fixes

This list is likely incomplete.

- Bug 84677 - Triangle disappears with glPolygonMode GL_LINE
- Bug 90734 - glBufferSubData is corrupting data when buffer is > 32k
- Bug 90748 - [BDW Bisected]dEQP-GLES3.functional.fbo.completeness.renderable.texture.depth.rg_half_float_oes fails
- Bug 90902 - [bsw][regression] dEQP: "Found invalid pixel values"
- Bug 90925 - "high fidelity": Segfault in _mesa_program_resource_find_name
- Bug 91254 - (regresion) video using VA-API on Intel slow and freeze system with mesa 10.6 or 10.6.1
- Bug 91292 - [BDW+] glVertexAttribDivisor not working in combination with glPolygonMode
- Bug 91673 - Segfault when calling glTexSubImage2D on storage texture to bound FBO
- Bug 91726 - R600 asserts in tgsi_cmp/make_src_for_op3

4.161.4 Changes

Chris Wilson (2):

- i965: Prevent coordinate overflow in intel_emit_linear_blit
- i965: Always re-emit the pipeline select during invariant state emission

Daniel Scharrer (1):

- mesa: add missing queries for ARB_direct_state_access

Dave Airlie (8):

- mesa/arb_gpu_shader_fp64: add support for glGetUniformdv
- mesa/texgetimage: fix missing stencil check
- st/readpixels: fix accel path for skipimages.
- texcompress_s3tc/fxt1: fix stride checks (v1.1)
- mesa/readpixels: check strides are equal before skipping conversion
- mesa: enable texture stencil8 for multisample
- r600/sb: update last_cf for finalize if.
- r600g: fix calculation for gpr allocation

David Heidelberg (1):

- st/nine: Require gcc >= 4.6

Emil Velikov (2):

- docs: add sha256 checksums for 10.6.5
- get-pick-list.sh: Require explicit "10.6" for nominating stable patches

Glenn Kennard (4):

- r600g: Fix assert in tgsi_cmp
- r600g/sb: Handle undef in read port tracker
- r600g/sb: Don't read junk after EOP
- r600g/sb: Don't crash on empty if jump target

Ilia Mirkin (5):

- st/mesa: fix assignments with 4-operand arguments (i.e. BFI)
- st/mesa: pass through 4th opcode argument in bitmap/pixel visitors
- nv50,nvc0: disable depth bounds test on blit
- nv50: fix 2d engine blits for 64- and 128-bit formats
- mesa: only copy the requested teximage faces

Jason Ekstrand (1):

- i965/fs: Split VGRFs after lowering pull constants

Kenneth Graunke (3):

- i965: Fix copy propagation type changes.
- Revert "i965: Advertise a line width of 40.0 on Cherryview and Skylake."
- i965: Momentarily pretend to support ARB_texture_stencil8 for blits.

Marek Olšák (3):

- gallium/radeon: fix the ADDRESS_HI mask for EVENT_WRITE CIK packets
- mesa: create multisample fallback textures like normal textures
- radeonsi: fix a Unigine Heaven hang when drirc is missing

Matt Turner (1):

- i965/fs: Handle MRF destinations in lower_integer_multiplication().

Neil Roberts (2):

- i965: Swap the order of the vertex ID and edge flag attributes
- i965/bdw: Fix 3DSTATE_VF_INSTANCING when the edge flag is used

Tapani Pälli (5):

- mesa: update fbo state in glTexStorage
- glsl: build stageref mask using IR, not symbol table
- glsl: expose build_program_resource_list function
- glsl: create program resource list after LinkShader

- mesa: add GL_RED, GL_RG support for floating point textures

4.162 Mesa 10.6.5 Release Notes / August 22, 2015

Mesa 10.6.5 is a bug fix release which fixes bugs found since the 10.6.4 release.

Mesa 10.6.5 implements the OpenGL 3.3 API, but the version reported by glGetString(GL_VERSION) or glGetIntegerv(GL_MAJOR_VERSION) / glGetIntegerv(GL_MINOR_VERSION) depends on the particular driver being used. Some drivers don't support all the features required in OpenGL 3.3. OpenGL 3.3 is **only** available if requested at context creation because compatibility contexts are not supported.

4.162.1 SHA256 checksums

```
afe290fc7af75a25df5ee52396a9f09e5dba85fb3e159304bdda265b8564b0d4  mesa-10.6.5.tar.gz
fb6fac3c85bcfa9d06b8dd439169f23f0c0924a88e44362e738b99b1feff762f  mesa-10.6.5.tar.xz
```

4.162.2 New features

None

4.162.3 Bug fixes

This list is likely incomplete.

- Bug 85252 - Segfault in compiler while processing ternary operator with void arguments
- Bug 91570 - Upgrading mesa to 10.6 causes segfault in OpenGL applications with GeForce4 MX 440 / AGP 8X
- Bug 91610 - [BSW] GPU hang for spec.shaders.point-vertex-id gl_instanceid divisor

4.162.4 Changes

Adam Jackson (1):

- glx: Fix __glXWireToEvent for BufferSwapComplete

Alex Deucher (2):

- radeonsi: add new OLAND pci id
- radeonsi: properly set the raster_config for KV

Emil Velikov (4):

- docs: add sha256 checksums for 10.6.4
- vc4: add missing nir include, to fix the build
- Revert "radeonsi: properly set the raster_config for KV"
- Update version to 10.6.5

Frank Binns (1):

- egl/x11: don't abort when creating a DRI2 drawable fails

Ilia Mirkin (3):

- nouveau: no need to do tnl wakeup, state updates are always hooked up
- gm107/ir: indirect handle goes first on maxwell also
- nv50,nvc0: take level into account when doing eng2d multi-layer blits

Jason Ekstrand (4):

- meta/copy_image: Stash off the scissor
- mesa/formats: Only do byteswapping for packed formats
- mesa/formats: Fix swizzle flipping for big-endian targets
- mesa/formats: Don't flip channels of null array formats

Marek Olšák (3):

- radeonsi: fix polygon offset scale
- r600g: fix polygon offset scale
- r600g: allow setting geometry shader sampler states

Neil Roberts (1):

- i965/bdw: Fix setting the instancing state for the SGVS element

Oded Gabbay (2):

- mesa: clear existing swizzle info before bitwise-OR
- mesa/formats: don't byteswap when building array formats

Renaud Gaubert (1):

- glsl: avoid compiler's segfault when processing operators with void arguments

4.163 Mesa 10.6.4 Release Notes / August 11, 2015

Mesa 10.6.4 is a bug fix release which fixes bugs found since the 10.6.3 release.

Mesa 10.6.4 implements the OpenGL 3.3 API, but the version reported by glGetString(GL_VERSION) or glGetIntegerv(GL_MAJOR_VERSION) / glGetIntegerv(GL_MINOR_VERSION) depends on the particular driver being used. Some drivers don't support all the features required in OpenGL 3.3. OpenGL 3.3 is **only** available if requested at context creation because compatibility contexts are not supported.

4.163.1 SHA256 checksums

```
4960bf17d8b5d6a6503c6954ec6cf480b5cd930797bac901c60bea192675f85e  mesa-10.6.4.tar.gz
8f5ac103f0f503de2f7a985b0df349bd4ecdfe7f51c714be146fa5a9a3c07b77  mesa-10.6.4.tar.xz
```

4.163.2 New features

None

4.163.3 Bug fixes

This list is likely incomplete.

- Bug 73512 - [clover] mesa.icd. should contain full path
- Bug 91290 - SIGSEGV glcpp/glcpp-parse.y:1077

4.163.4 Changes

Anuj Phogat (6):

- mesa: Turn get_readpixels_transfer_ops() in to a global function
- meta: Fix transfer operations check in meta pbo path for readpixels
- meta: Abort meta pbo path if readpixels need signed-unsigned conversion
- meta: Don't do fragment color clamping in _mesa_meta_pbo_GetTexSubImage
- mesa: Add a helper function _mesa_need_luminance_to_rgb_conversion()
- meta: Fix reading luminance texture as rgba in _mesa_meta_pbo_GetTexSubImage()

Ben Widawsky (1):

- i965/skl: Add production thread counts and URB size

Eduardo Lima Mitev (3):

- mesa: Fix errors values returned by glShaderBinary()
- mesa: Validate target before resolving tex obj in glTex(ture)SubImageXD
- mesa: Fix error returned by glCopyTexImage2D() upon an invalid internal format

Emil Velikov (6):

- docs: Add checksums for mesa 10.6.3 tarballs
- configure.ac: do not set HAVE_DRI(23) when libdrm is missing
- egl/wayland: libdrm is a hard requirement, treat it as such
- winsys/radeon: don't leak the fd when it is 0
- bugzilla_mesa.sh: sort the bugs list by number
- Update version to 10.6.4

Francisco Jerez (1):

- i965/fs: Fix fs_inst::regs_read() for sources in the ATTR file.

Frank Binns (2):

- egl/dri: Add error info needed for EGL_EXT_image_dma_buf_import extension
- egl: Add eglQuerySurface surface type check for EGL_LARGEST_PBUFFER attrib

Igor Gnatenko (1):

- opencl: use versioned .so in mesa.icd

Ilia Mirkin (1):

- nvc0: fix geometry program revalidation of clipping params

Kenneth Graunke (1):

- glsl: Fix a bug where LHS swizzles of swizzles were too small.

Marek Olšák (6):

- st/mesa: don't call st_validate_state in BlitFramebuffer
- radeonsi: upload shader rodata after updating scratch relocations
- st/mesa: don't ignore texture buffer state changes
- radeonsi: rework how shader pointers to descriptors are set
- radeonsi: completely rework updating descriptors without CP DMA
- r600g: fix the CB_SHADER_MASK setup

Samuel Iglesias Gonsalvez (1):

- glsl/glcpp: fix SIGSEGV when checking error condition for macro redefinition

Samuel Pitoiset (1):

- nv50: avoid segfault with enabled but unbound vertex attrib

4.164 Mesa 10.6.3 Release Notes / July 26, 2015

Mesa 10.6.3 is a bug fix release which fixes bugs found since the 10.6.2 release.

Mesa 10.6.3 implements the OpenGL 3.3 API, but the version reported by glGetString(GL_VERSION) or glGetIntegerv(GL_MAJOR_VERSION) / glGetIntegerv(GL_MINOR_VERSION) depends on the particular driver being used. Some drivers don't support all the features required in OpenGL 3.3. OpenGL 3.3 is **only** available if requested at context creation because compatibility contexts are not supported.

4.164.1 SHA256 checksums

```
c27e1e33798e69a6d2d2425aee8ac7b4c0b243066a65dd76cbb182ea31b1c7f2  mesa-10.6.3.tar.gz
58592e07c350cd2e8969b73fa83048c657a39fe2f13f3b88f5e5818fe2e4676d  mesa-10.6.3.tar.xz
```

4.164.2 New features

None

4.164.3 Bug fixes

This list is likely incomplete.

- Bug 90728 - dvd playback with vlc and vdpau causes segmentation fault
- Bug 91337 - OSMesaGetProcAdress("OSMesaPixelStore") returns nil

4.164.4 Changes

Brian Paul (1):

- osmesa: fix OSMesaPixelsStore typo

Chad Versace (1):

- mesa: Fix generation of git_sha1.h.tmp for gitlinks

Christian König (2):

- vl: cleanup video buffer private when the decoder is destroyed
- st/vdpau: fix mixer size checks

Emil Velikov (3):

- docs: Add sha256 checksums for the 10.6.2 release
- auxiliary/vl: use the correct screen index
- Update version to 10.6.3

Francisco Jerez (1):

- i965/gen9: Use custom MOCS entries set up by the kernel.

Ilia Mirkin (5):

- nv50, nvc0: enable at least one color RT if alphatest is enabled
- nvc0/ir: fix txq on indirect samplers
- nvc0/ir: don't worry about sampler in txq handling
- gm107/ir: fix indirect txq emission
- nv50: fix max level clamping on G80

Kenneth Graunke (1):

- program: Allow redundant OPTION ARB_fog_* directives.

Rob Clark (1):

- xa: don't leak fences

4.165 Mesa 10.6.2 Release Notes / July 11, 2015

Mesa 10.6.2 is a bug fix release which fixes bugs found since the 10.6.1 release.

Mesa 10.6.2 implements the OpenGL 3.3 API, but the version reported by glGetString(GL_VERSION) or glGetIntegerv(GL_MAJOR_VERSION) / glGetIntegerv(GL_MINOR_VERSION) depends on the particular driver being used. Some drivers don't support all the features required in OpenGL 3.3. OpenGL 3.3 is **only** available if requested at context creation because compatibility contexts are not supported.

4.165.1 SHA256 checksums

```
9c7ab9300dda6c912faaaff97995ec1820ba21d114d9cf555f145cbad90995f4  mesa-10.6.2.tar.gz
05753d3db4212900927b9894221a1669a10f56786e86a7e818b6e18a0817dca9  mesa-10.6.2.tar.xz
```

4.165.2 New features

None

4.165.3 Bug fixes

This list is likely incomplete.

- Bug 73528 - Deferred lighting in Second Life causes system hiccups and screen flickering
- Bug 80500 - Flickering shadows in unreleased title trace
- Bug 82186 - [r600g] BARTS GPU lockup with minecraft shaders
- Bug 84225 - Allow constant-index-expression sampler array indexing with GLSL-ES < 300
- Bug 90537 - radeonsi bo/va conflict on RADEON_GEM_VA (rscreen->ws->buffer_from_handle returns NULL)
- Bug 90873 - Kernel hang, TearFree On, Mate desktop environment
- Bug 91022 - [g45 g965 bisected] assertions generated from textureGrad cube samplers fix
- Bug 91047 - [SNB Bisected] Messed up Fog in Super Smash Bros. Melee in Dolphin
- Bug 91056 - The Bard's Tale (2005, native) has rendering issues
- Bug 91117 - Nimbus (running in wine) has rendering issues, objects are semi-transparent
- Bug 91124 - Civilization V (in Wine) has rendering issues: text missing, menu bar corrupted
- Bug 91173 - Oddworld: Stranger's Wrath HD: disfigured models in wrong colors
- Bug 91226 - Crash in glLinkProgram (NEW)
- Bug 91231 - [NV92] Psychonauts (native) segfaults on start when DRI3 enabled

4.165.4 Changes

Chris Wilson (1):

- loader: Look for any version of currently linked libudev.so

Emil Velikov (2):

- docs: Add sha256 checksums for the 10.6.1 release
- Update version to 10.6.2

Ilia Mirkin (8):

- nv50/ir: propagate modifier to right arg when const-folding mad
- nv50/ir: fix emission of address reg in 3rd source
- nv50/ir: copy joinAt when splitting both before and after
- mesa: reset the source packing when creating temp transfer image
- nv50/ir: don't emit src2 in immediate form
- mesa/prog: relative offsets into constbufs are not constant
- nv50/ir: UCMP arguments are float, so make sure modifiers are applied
- nvc0: turn sample counts off during blit

Kenneth Graunke (5):

- i965/fs: Fix ir_txs in emit_texture_gen4_simd16().
- i965: Reserve more batch space to accomodate Gen6 perfmonitors.
- i965/vs: Fix matNxM vertex attributes where M != 4.
- Revert "glsl: clone inputs and outputs during linking"
- Revert "i965: Delete linked GLSL IR when using NIR."

Marek Olšák (3):

- r600g: disable single-sample fast color clear due to hangs
- radeonsi: fix a hang with DrawTransformFeedback on 4 SE chips
- st/dri: don't set PIPE_BIND_SCANOUT for MSAA surfaces

Mario Kleiner (2):

- nouveau: Use dup fd as key in drm-winsys hash table to fix ZaphodHeads.
- winsys/radeon: Use dup fd as key in drm-winsys hash table to fix ZaphodHeads.

Matt Turner (2):

- i965/fs: Don't mess up stride for uniform integer multiplication.
- Revert SHA1 additions.

Michel Dänzer (1):

- winsys/radeon: Unmap GPU VM address range when destroying BO

Mike Stroyan (2):

- meta: Only change and restore viewport 0 in mesa meta mode
- i965: allocate at least 1 BLEND_STATE element

Neil Roberts (4):

- i965/skl: Set the pulls bary bit in 3DSTATE_PS_EXTRA
- glsl: Add missing check for whether an expression is an add operation
- glsl: Make sure not to dereference NULL
- i965: Don't try to print the GLSL IR if it has been freed

Tapani Pälli (8):

- glsl: clone inputs and outputs during linking
- i965: Delete linked GLSL IR when using NIR.
- glsl: Allow dynamic sampler array indexing with GLSL ES < 3.00
- mesa/glsl: new compiler option EmitNoIndirectSampler
- i965: use EmitNoIndirectSampler for gen < 7
- i915: use EmitNoIndirectSampler
- mesa/st: use EmitNoIndirectSampler if !ARB_gpu_shader5
- glsl: validate sampler array indexing for 'constant-index-expression'

4.166 Mesa 10.5.9 Release Notes / July 04, 2015

Mesa 10.5.9 is a bug fix release which fixes bugs found since the 10.5.8 release.

Mesa 10.5.9 implements the OpenGL 3.3 API, but the version reported by glGetString(GL_VERSION) or glGetIntegerv(GL_MAJOR_VERSION) / glGetIntegerv(GL_MINOR_VERSION) depends on the particular driver being used. Some drivers don't support all the features required in OpenGL 3.3. OpenGL 3.3 is **only** available if requested at context creation because compatibility contexts are not supported.

4.166.1 SHA256 checksums

```
0c081b59572ee9732e7438d34adc3817fe8cc8d4b58abc0e71fd4b4c904945cb  mesa-10.5.9.tar.gz
71c69f31d3dbc35cfa79950e58a01d27030378d8c7ef1259a0b31d4d0487f4ec  mesa-10.5.9.tar.xz
```

4.166.2 New features

None

4.166.3 Bug fixes

This list is likely incomplete.

- Bug 84225 - Allow constant-index-expression sampler array indexing with GLSL-ES < 300
- Bug 88999 - [SKL] Compiz crashes after opening unity dash
- Bug 89118 - [SKL Bisected]many Ogles3conform cases core dumped
- Bug 90537 - radeonsi bo/va conflict on RADEON_GEM_VA (rscreen->ws->buffer_from_handle returns NULL)
- Bug 90839 - [10.5.5/10.6 regression, bisected] PBO glDrawPixels no longer using blit fastpath
- Bug 90873 - Kernel hang, TearFree On, Mate desktop environment
- Bug 91056 - The Bard's Tale (2005, native) has rendering issues
- Bug 91117 - Nimbus (running in wine) has rendering issues, objects are semi-transparent
- Bug 91124 - Civilization V (in Wine) has rendering issues: text missing, menu bar corrupted

4.166.4 Changes

Ben Widawsky (2):

- i965/gen9: Implement Push Constant Buffer workaround
- i965/skl: Use 1 register for uniform pull constant payload

Boyan Ding (1):

- egl/x11: Remove duplicate call to dri2_x11_add_configs_for_visuals

Chris Wilson (3):

- i965: Fix HW blitter pitch limits
- i915: Blit RGBX<->RGBA drawpixels

- i965: Export format comparison for blitting between miptrees

Emil Velikov (6):

- docs: Add sha256sums for the 10.5.8 release
- configure: warn about shared_glapi & xlib-glx only when both are set
- configure: error out when building backend-less libEGL
- configure: error out when building libEGL without shared-glapi
- gbm: do not (over)link against libglapi.so
- Update version to 10.5.9

Frank Henigman (1):

- gbm: dlopen libglapi so gbm_create_device works

Ilia Mirkin (8):

- glsl: add version checks to conditionals for builtin variable enablement
- mesa: add GL_PROGRAM_PIPELINE support in KHR_debug calls
- glsl: binding point is a texture unit, which is a combined space
- nvc0: always put all tfb bufs into bufctx
- nv50,nvc0: make sure to pushbuf_refn before putting bo into pushbuf_data
- nv50/ir: propagate modifier to right arg when const-folding mad
- nv50/ir: fix emission of address reg in 3rd source
- nv50/ir: copy joinAt when splitting both before and after

Mario Kleiner (2):

- nouveau: Use dup fd as key in drm-winsys hash table to fix ZaphodHeads.
- winsys/radeon: Use dup fd as key in drm-winsys hash table to fix ZaphodHeads.

Michel Dänzer (1):

- winsys/radeon: Unmap GPU VM address range when destroying BO

Tapani Pälli (6):

- glsl: Allow dynamic sampler array indexing with GLSL ES < 3.00
- mesa/glsl: new compiler option EmitNoIndirectSampler
- i915: use EmitNoIndirectSampler
- mesa/st: use EmitNoIndirectSampler if !ARB_gpu_shader5
- i965: use EmitNoIndirectSampler for gen < 7
- glsl: validate sampler array indexing for 'constant-index-expression'

4.167 Mesa 10.6.1 Release Notes / June 29, 2015

Mesa 10.6.1 is a bug fix release which fixes bugs found since the 10.6.0 release.

Mesa 10.6.1 implements the OpenGL 3.3 API, but the version reported by glGetString(GL_VERSION) or glGetIntegerv(GL_MAJOR_VERSION) / glGetIntegerv(GL_MINOR_VERSION) depends on the particular driver being used.

Some drivers don't support all the features required in OpenGL 3.3. OpenGL 3.3 is **only** available if requested at context creation because compatibility contexts are not supported.

4.167.1 SHA256 checksums

```
b4cccd4d0eabcc2bca00c3175d3ad88fdda57ffdb883a7998525b873a21fe607  mesa-10.6.1.tar.gz
6c80a2b647e57c85dc36e609d9aed17f878f0d8e0cf9ace86d14cf604101e1eb  mesa-10.6.1.tar.xz
```

4.167.2 New features

None

4.167.3 Bug fixes

This list is likely incomplete.

- Bug 90347 - [NVE0+] Failure to insert texbar under some circumstances (causing bad colors in Terasology)

4.167.4 Changes

Anuj Phogat (4):

- mesa: Handle integer formats in need_rgb_to_luminance_conversion()
- mesa: Use helper function need_rgb_to_luminance_conversion()
- mesa: Turn need_rgb_to_luminance_conversion() in to a global function
- meta: Abort meta path if ReadPixels need rgb to luminance conversion

Ben Widawsky (1):

- i965/gen9: Implement Push Constant Buffer workaround

Boyan Ding (2):

- egl/x11: Set version of swrastLoader to 2
- egl/x11: Remove duplicate call to dri2_x11_add_configs_for_visuals

Emil Velikov (6):

- docs: Add sha256sums for the 10.6.0 release
- configure: warn about shared_glapi & xlib-glx only when both are set
- configure: error out when building backend-less libEGL
- configure: error out when building libEGL without shared-glapi
- gbm: do not (over)link against libglapi.so
- Update version to 10.6.1

Frank Henigman (1):

- gbm: dlopen libglapi so gbm_create_device works

Ilia Mirkin (9):

- nvc0/ir: fix collection of first uses for texture barrier insertion
- nv50,nvc0: clamp uniform size to 64k
- nvc0/ir: can't have a join on a load with an indirect source
- glsl: handle conversions to double when comparing param matches
- glsl: add version checks to conditionals for builtin variable enablement
- mesa: add GL_PROGRAM_PIPELINE support in KHR_debug calls
- glsl: binding point is a texture unit, which is a combined space
- nvc0: always put all tfb bufs into bufctx
- nv50,nvc0: make sure to pushbuf_refn before putting bo into pushbuf_data

4.168 Mesa 10.5.8 Release Notes / June 20, 2015

Mesa 10.5.8 is a bug fix release which fixes bugs found since the 10.5.7 release.

Mesa 10.5.8 implements the OpenGL 3.3 API, but the version reported by glGetString(GL_VERSION) or glGetIntegerv(GL_MAJOR_VERSION) / glGetIntegerv(GL_MINOR_VERSION) depends on the particular driver being used. Some drivers don't support all the features required in OpenGL 3.3. OpenGL 3.3 is **only** available if requested at context creation because compatibility contexts are not supported.

4.168.1 SHA256 checksums

```
611ddcfa3c1bf13f7e6ccac785c8749c3b74c9a78452bac70f8372cf6b209aa0  mesa-10.5.8.tar.gz
2866b855c5299a4aed066338c77ff6467c389b2c30ada7647be8758663da2b54  mesa-10.5.8.tar.xz
```

4.168.2 New features

None

4.168.3 Bug fixes

This list is likely incomplete.

- Bug 90310 - Fails to build gallium_dri.so at linking stage with clang because of multiple redefinitions
- Bug 90347 - [NVE0+] Failure to insert texbar under some circumstances (causing bad colors in Terasology)
- Bug 90520 - Register spilling clobbers registers used elsewhere in the shader
- Bug 90905 - mesa: Finish subdir-objects transition

4.168.4 Changes

Ben Widawsky (1):

- i965: Disable compaction for EOT send messages

Boyan Ding (1):

- egl/x11: Set version of swrastLoader to 2

Emil Velikov (2):

- docs: Add sha256sums for the 10.5.7 release
- Update version to 10.5.8

Erik Faye-Lund (1):

- mesa: build xmlconfig to a separate static library

Francisco Jerez (1):

- i965: Don't compact instructions with unmapped bits.

Ilia Mirkin (3):

- nvc0/ir: fix collection of first uses for texture barrier insertion
- nv50,nvc0: clamp uniform size to 64k
- nvc0/ir: can't have a join on a load with an indirect source

Jason Ekstrand (1):

- i965/fs: Don't let the EOT send message interfere with the MRF hack

Marek Olšák (1):

- egl: fix setting context flags

Roland Scheidegger (1):

- draw: (trivial) fix NULL pointer dereference

4.169 Mesa 10.6.0 Release Notes / June 14, 2015

Mesa 10.6.0 is a new development release. People who are concerned with stability and reliability should stick with a previous release or wait for Mesa 10.6.1.

Mesa 10.6.0 implements the OpenGL 3.3 API, but the version reported by glGetString(GL_VERSION) or glGetIntegerv(GL_MAJOR_VERSION) / glGetIntegerv(GL_MINOR_VERSION) depends on the particular driver being used. Some drivers don't support all the features required in OpenGL 3.3. OpenGL 3.3 is **only** available if requested at context creation because compatibility contexts are not supported.

4.169.1 SHA256 checksums

```
9bc659abdba26202509304f259723aaa4343dba6aac4bd87d5baea11d23c8c63  mesa-10.6.0.tar.gz
f37e2633978deed02ff0522abc36c709586e2b555fd439a82ab71dce2c866c76  mesa-10.6.0.tar.xz
```

4.169.2 New features

Note: some of the new features are only available with certain drivers.

- GL_AMD_pinned_memory on r600, radeonsi
- GL_ARB_clip_control on i965
- GL_ARB_depth_buffer_float on freedreno

- GL_ARB_depth_clamp on freedreno
- GL_ARB_direct_state_access on all drivers that support GL 2.0+
- GL_ARB_draw_indirect, GL_ARB_multi_draw_indirect on r600
- GL_ARB_draw_instanced on freedreno
- GL_ARB_gpu_shader_fp64 on nvc0, softpipe
- GL_ARB_gpu_shader5 on i965/gen8+
- GL_ARB_instanced_arrays on freedreno
- GL_ARB_pipeline_statistics_query on i965, nv50, nvc0, r600, radeonsi, softpipe
- GL_ARB_program_interface_query (all drivers)
- GL_ARB_texture_stencil8 on nv50, nvc0, r600, radeonsi, softpipe
- GL_ARB_texture_view on llvmpipe, softpipe
- GL_ARB_uniform_buffer_object on freedreno
- GL_ARB_vertex_attrib_64bit on nvc0, softpipe
- GL_ARB_viewport_array, GL_AMD_vertex_shader_viewport_index on i965/gen6
- GL_EXT_draw_buffers2 on freedreno
- GL_OES_EGL_sync on all drivers
- EGL_KHR_fence_sync on i965, freedreno, nv50, nvc0, r600, radeonsi
- EGL_KHR_wait_sync on i965, freedreno, nv50, nvc0, r600, radeonsi
- EGL_KHR_cl_event2 on freedreno, nv50, nvc0, r600, radeonsi
- GL_AMD_performance_monitor on nvc0

4.169.3 Bug fixes

This list is likely incomplete.

- Bug 15006 - translate & rotate the line cause Aliasing
- Bug 27007 - Lines disappear with GL_LINE_SMOOTH
- Bug 28832 - piglit/general/line-aa-width fail
- Bug 45348 - [swrast] piglit fbo-drawbuffers-arbfp regression
- Bug 60797 - 1px lines in octave plot aliased to 0
- Bug 67564 - HiZ buffers are much larger than necessary
- Bug 69226 - Cannot enable basic shaders with Second Life aborts attempt
- Bug 71591 - Second Life shaders fail to compile (extension declared in middle of shader)
- Bug 79202 - valgrind errors in glsl-fs-uniform-array-loop-unroll.shader_test; random code generation
- Bug 81025 - [IVB/BYT Bisected]Piglit spec_ARB_draw_indirect_arb_draw_indirect-draw-elements-prim-restart-ugly fails
- Bug 82477 - [softpipe] piglit fp-long-alu regression
- Bug 82668 - Can't set int attributes to certain values on 32-bit

- Bug 82831 - i965: Support GL_ARB_blend_func_extended in SIMD16
- Bug 83962 - [HSW/BYT]Piglit spec_ARB_gpu_shader5_arb_gpu_shader5-emitstreamvertex_nodraw fails
- Bug 84613 - [G965, bisected] piglit regressions : glslparsertest.glsl2
- Bug 86747 - Noise in Football Manager 2014 textures
- Bug 86792 - [NVC0] Portal 2 Crashes in Wine
- Bug 86811 - [BDW/BSW Bisected]Piglit spec_arb_shading_language_packing_execution_built-in-functions_vs-unpackSnorm4x8 fails
- Bug 86837 - kodi segfault since auxiliary/vl: rework the build of the VL code
- Bug 86944 - glsl_parser_extras.cpp", line 1455: Error: Badly formed expression. (Oracle Studio)
- Bug 86974 - INTEL_DEBUG=shader_time always asserts in fs_generator::generate_code() when Mesa is built with –enable-debug (= with asserts)
- Bug 86980 - [swrast] piglit fp-rfl regression
- Bug 87258 - [BDW/BSW Bisected]Piglit spec_ARB_shader_atomic_counters_array-indexing fails
- Bug 88246 - Commit 2881b12 causes 43 DrawElements test regressions
- Bug 88248 - Calling glClear while there is an occlusion query in progress messes up the results
- Bug 88521 - GLBenchmark 2.7 TRex renders with artifacts on Gen8 with !UXA
- Bug 88534 - include/c11/threads_posix.h PTHREAD_MUTEX_RECURSIVE_NP not defined
- Bug 88561 - [radeonsi][regression,bisected] Depth test/buffer issues in Portal
- Bug 88793 - [BDW/BSW Bisected]Piglit/shaders_glsl-max-varyings fails
- Bug 88815 - Incorrect handling of GLSL #line directive
- Bug 88883 - ir-a2xx.c: variable changed in assert statement
- Bug 88885 - Transform feedback uses incorrect interleaving if a previous draw did not write gl_Position
- Bug 88905 - [SNB+ Bisected]Ogles3conform ES3-CTS.gtf.GL3Tests.packed_pixels.packed_pixels fails
- Bug 88999 - [SKL] Compiz crashes after opening unity dash
- Bug 89014 - PIPE_QUERY_GPU_FINISHED is not acting as expected on SI
- Bug 89026 - Renderbuffer layered state used for framebuffer completeness test
- Bug 89032 - [BDW/BSW/SKL Bisected]Piglit spec_OpenGL_1.1_infinite-spot-light fails
- Bug 89037 - [SKL]Piglit spec_EXT_texture_array_copyteximage_1D_ARRAY_samples=2 sporadically causes GPU hang
- Bug 89039 - [SKL]etqw system hang
- Bug 89058 - [SKL]Render error in some games (etqw-demo, nexuiz, portal)
- Bug 89068 - glTexImage2D regression by texstore_rgba switch to _mesa_format_convert
- Bug 89069 - Lack of grass in The Talos Principle on radeonsi (nativewinenine)
- Bug 89094 - [SNB/IVB/HSW/BYT Bisected]Ogles3conform ES3-CTS.gtf.GL3Tests.shadow.shadow_execution_vert fails
- Bug 89095 - [SNB/IVB/BYT Bisected]Webglc conformance/glsl/functions/glsl-function-mix-float.html fails
- Bug 89112 - u_atomic_test: u_atomic_test.c:124: test_atomic_8bits_bool: Assertion 'r == 65 && "p_atomic_add"' failed.

- Bug 89118 - [SKL Bisected]many Ogles3conform cases core dumped
- Bug 89131 - [Bisected] Graphical corruption in Weston, shows old framebuffer pieces
- Bug 89156 - r300g: GL_COMPRESSED_RED_RGTC1 / ATI1N support broken
- Bug 89180 - [IVB regression] Rendering issues in Mass Effect through VMware Workstation
- Bug 89210 - GS statistics fail on SNB
- Bug 89218 - lower_instructions.cpp:648:48: error: invalid suffix 'd' on floating constant
- Bug 89224 - Incorrect rendering of Unigine Valley running in VM on VMware Workstation
- Bug 89260 - macros.h:34:25: fatal error: util/u_math.h: No such file or directory
- Bug 89292 - [regression,bisected] incomplete screenshots in some cases
- Bug 89311 - [regression, bisected] dEQP: Added entry points for glCompressedTextureSubImage*D.
- Bug 89312 - [regression, bisected] main: Added entry points for CopyTextureSubImage*D. (d6b7c40cecfe01)
- Bug 89315 - [HSW, regression, bisected] i965/fs: Emit MAD instructions when possible.
- Bug 89317 - [HSW, regression, bisected] i965: Add LINTERP/CINTERP to can_do_cmod() (d91390634)
- Bug 89328 - python required to build Mesa release tarballs
- Bug 89342 - main/light.c:159:62: error: 'M_PI' undeclared (first use in this function)
- Bug 89343 - compiler/tests/radeon_compiler_optimize_tests.c:43:3: error: implicit declaration of function 'fprintf' [-Werror=implicit-function-declaration]
- Bug 89345 - imports.h:452:58: error: expected declaration specifiers or '...' before 'va_list'
- Bug 89364 - c99_alloca.h:40:22: fatal error: alloca.h: No such file or directory
- Bug 89372 - [softpipe] piglit glsl-1.50 generate-zero-primitives regression
- Bug 89387 - Double delete in lp_bld_misc.cpp
- Bug 89416 - UE4Editor crash after load project
- Bug 89430 - [g965][bisected] arb_copy_image-targets gl_texture* tests fail
- Bug 89433 - GCC 4.2 does not support -Wvla
- Bug 89455 - [NVC0/Gallium] Unigine Heaven black and white boxes
- Bug 89457 - [BSW Bisected]ogles3conform ES3-CTS.gtf.GL3Tests.shadow.shadow_execution_vert fails
- Bug 89477 - include/no_extern_c.h:47:1: error: template with C linkage
- Bug 89508 - Bad int(floatBitsToInt(vec4))
- Bug 89530 - FTBFS in loader: missing fstat
- Bug 89569 - Papo & Yo crash on startup [HSW]
- Bug 89590 - Crash in glLinkProgram with shaders with multiple constant arrays
- Bug 89662 - context.c:943: undefined reference to '_glapi_new_nop_table'
- Bug 89670 - cmod_propagation_test.andnz_one regression
- Bug 89679 - [NV50] Portal/Half-Life 2 will not start (native Steam)
- Bug 89689 - [Regression] Weston on DRM backend won't start with new version of mesa
- Bug 89722 - [ILK Bisected]Ogles2conform/ES2-CTS.gtf.GL.equal.equal_vec2_frag fails

- Bug 89726 - [Bisected] dEQP-GLES3: uniform linking logic in the presence of structs
- Bug 89746 - Mesa and LLVM 3.6+ break opengl for genymotion
- Bug 89754 - vertexAttrib fails WebGL Conformance test with mesa drivers
- Bug 89758 - pow WebGL Conformance test with mesa drivers
- Bug 89759 - WebGL OGL ES GLSL conformance test with mesa drivers fails
- Bug 89831 - [r600] r600_asm.c:310:assign_alu_units: Assertion '0' failed.
- Bug 89899 - nir/nir_lower_tex_projector.c:112: error: unknown field 'ssa' specified in initializer
- Bug 89957 - vm protection faults in piglit lest: texsubimage cube_map_array pbo
- Bug 89960 - [softpipe] piglit copy-pixels regreession
- Bug 89961 - [BDW/BSW Bisected]Synmark2_v6 OglDrvRes/OglDrvShComp/OglDrvState/OglPSPom Image Validation fail
- Bug 89963 - lp_bld_debug.cpp:100:31: error: no matching function for call to 'llvm::raw_ostream::raw_ostream()'
- Bug 90000 - [i965 Bisected NIR] Piglit/gglean_fragprog1-z-write_test fail
- Bug 90109 - [SNB+ Bisected]Ogles3conform ES3-CTS.shaders.uniform_block.random.basic_arrays.3 fails
- Bug 90114 - [SNB+ Bisected]Ogles3conform ES3-CTS.shaders.struct.uniform.sampler_array_fragment fails
- Bug 90130 - gl_PrimitiveId seems to reset at 340
- Bug 90147 - swrast: build error undeclared _SC_PHYS_PAGES on osx
- Bug 90149 - [SNB+ Bisected]ES3-CTS.gtf.GL3Tests.uniform_buffer_object.uniform_buffer_object_getactiveuniformsiv_for_nor fails
- Bug 90153 - [SKL Bisected]ES3-CTS.gtf.GL3Tests.uniform_buffer_object.uniform_buffer_object_all_valid_basic_types fails
- Bug 90167 - [softpipe] piglit depthstencil-default_fb-drawpixels-32f_24_8_rev regression
- Bug 90207 - [r600g, bisected] regression: NI/Turks crash on WebGL Water (most WebGL stuff)
- Bug 90213 - glDrawPixels with GL_COLOR_INDEX never returns.
- Bug 90243 - [bisected] regression: spec.!opengl 3_2.get-active-attrib-returns-all-inputs
- Bug 90258 - [IVB] spec.glsl-1_10.execution.fs-dfdy-accuracy fails intermittently
- Bug 90310 - Fails to build gallium_dri.so at linking stage with clang because of multiple redefinitions
- Bug 90350 - [G96] Portal's portal are incorrectly rendered
- Bug 90363 - [nv50] HW state is not reset correctly when using a new GL context
- Bug 90397 - ARB_program_interface_query: glGetProgramResourceiv() returns wrong value for GL_REFERENCED_BY_*_SHADER prop for GL_UNIFORM for members of an interface block with an instance name
- Bug 90466 - arm: linker error ndefined reference to 'nir_metadata_preserve'
- Bug 90520 - Register spilling clobbers registers used elsewhere in the shader
- Bug 90547 - [BDW/BSW/SKL Bisected]Piglit/glean@vertprog1-rsq_test_2_(reciprocal_square_root_of_negative_value) fais
- Bug 90580 - [HSW bisected] integer multiplication bug

- Bug 90629 - [i965] SIMD16 dual_source_blend assertion 'src[i].file != GRF || src[i].width == dst.width' failed
- Bug 90749 - [BDW Bisected]dEQP-GLES3.functional.rasterization.fbo.rbo_multisample_max.primitives.lines_wide fails
- Bug 90830 - [bsw bisected regression] GPU hang for spec.arb_gpu_shader5.execution.sampler_array_indexing.vs-nonzero-base
- Bug 90839 - [10.5.5/10.6 regression, bisected] PBO glDrawPixels no longer using blit fastpath
- Bug 90905 - mesa: Finish subdir-objects transition
- Bug 9951 - GL_LINE_SMOOTH and GL_POLYGON_SMOOTH with i965 driver

4.169.4 Changes

- Removed classic Windows software rasterizer.
- Removed egl_gallium EGL driver.
- Removed gbm_gallium GBM driver.
- Removed OpenVG support.
- Removed the galahad gallium driver.
- Removed the identity gallium driver.
- Removed the EGL loader from the Windows SCons build.
- Removed the classic osmesa from the Windows SCons build.

4.170 Mesa 10.5.7 Release Notes / June 07, 2015

Mesa 10.5.7 is a bug fix release which fixes bugs found since the 10.5.6 release.

Mesa 10.5.7 implements the OpenGL 3.3 API, but the version reported by glGetString(GL_VERSION) or glGetIntegerv(GL_MAJOR_VERSION) / glGetIntegerv(GL_MINOR_VERSION) depends on the particular driver being used. Some drivers don't support all the features required in OpenGL 3.3. OpenGL 3.3 is **only** available if requested at context creation because compatibility contexts are not supported.

4.170.1 SHA256 checksums

```
8f865ce497435fdf25d4e35f3b5551b2bcd5f9bc6570561183be82af20d18b82  mesa-10.5.7.tar.gz
04d06890cd69af8089d6ca76f40e46dcf9cacfe4a9788b32be620574d4638818  mesa-10.5.7.tar.xz
```

4.170.2 New features

None

4.170.3 Bug fixes

This list is likely incomplete.

- Bug 89131 - [Bisected] Graphical corruption in Weston, shows old framebuffer pieces

4.170.4 Changes

Ben Widawsky (1):

- i965: Emit 3DSTATE_MULTISAMPLE before WM_HZ_OP (gen8+)

Emil Velikov (4):

- docs: Add sha256sums for the 10.5.6 release
- get-pick-list.sh: Require explicit "10.5" for nominating stable patches
- cherry-ignore: add clover build fix not applicable for 10.5
- Update version to 10.5.7

Ilia Mirkin (18):

- nvc0/ir: set ftz when sources are floats, not just destinations
- nv50/ir: guess that the constant offset is the starting slot of array
- nvc0/ir: LOAD's can't be used for shader inputs
- nvc0: a geometry shader can have up to 1024 vertices output
- nv50/ir: avoid messing up arg1 of PFETCH
- nv30: don't leak fragprog consts
- nv30: avoid leaking render state and draw shaders
- nv30: fix clip plane uploads and enable changes
- nv30/draw: avoid leaving stale pointers in draw state
- nv30/draw: draw expects constbuf size in bytes, not vec4 units
- st/mesa: don't leak glsl_to_tgsi object on link failure
- glsl: avoid leaking linked gl_shader when there's a late linker error
- nv30/draw: fix indexed draws with swtnl path and a resource index buffer
- nv30/draw: only use the DMA1 object (GART) if the bo is not in VRAM
- nv30/draw: allocate vertex buffers in gart
- nv30/draw: switch varying hookup logic to know about texcoords
- nv30: falling back to draw path for edgeflag does no good
- nv30: avoid doing extra work on clear and hitting unexpected states

Jason Ekstrand (1):

- i965/fs: Fix implied_mrf_writes for scratch writes

Marek Olšák (1):

- st/dri: fix postprocessing crash when there's no depth buffer

4.171 Mesa 10.5.6 Release Notes / May 23, 2015

Mesa 10.5.6 is a bug fix release which fixes bugs found since the 10.5.5 release.

Mesa 10.5.6 implements the OpenGL 3.3 API, but the version reported by glGetString(GL_VERSION) or glGetIntegerv(GL_MAJOR_VERSION) / glGetIntegerv(GL_MINOR_VERSION) depends on the particular driver being used. Some drivers don't support all the features required in OpenGL 3.3. OpenGL 3.3 is **only** available if requested at context creation because compatibility contexts are not supported.

4.171.1 SHA256 checksums

```
89ff9cb08d0f6e3f34154864c3071253057cd21020759457c8ae27e0f70985d3  mesa-10.5.6.tar.gz
66017853bde5f7a6647db3eede30512a091a3491daa1708e0ad8027c328ba595  mesa-10.5.6.tar.xz
```

4.171.2 New features

None

4.171.3 Bug fixes

This list is likely incomplete.

- Bug 86792 - [NVC0] Portal 2 Crashes in Wine
- Bug 90147 - swrast: build error undeclared _SC_PHYS_PAGES on osx
- Bug 90350 - [G96] Portal's portal are incorrectly rendered
- Bug 90363 - [nv50] HW state is not reset correctly when using a new GL context

4.171.4 Changes

Alex Deucher (1):

- radeonsi: add new bonaire pci id

Axel Davy (2):

- egl/wayland: properly destroy wayland objects
- glx/dri3: Add additional check for gpu offloading case

Emil Velikov (4):

- docs: Add sha256 sums for the 10.5.5 release
- egl/main: fix EGL_KHR_get_all_proc_addresses
- targets/osmesa: drop the -module tag from LDFLAGS
- Update version to 10.5.6

Francisco Jerez (4):

- clover: Refactor event::trigger and ::abort to prevent deadlock and reentrancy issues.
- clover: Wrap event::_status in a method to prevent unlocked access.
- clover: Implement locking of the wait_count, _chain and _status members of event.
- i965: Fix PBO cache coherency issue after _mesa_meta_pbo_GetTexSubImage().

Fredrik Höglund (2):

- main: Require that the texture exists in framebuffer_texture
- mesa: Generate GL_INVALID_VALUE in framebuffer_texture when layer < 0

Ilia Mirkin (7):

- nv50/ir: only propagate saturate up if some actual folding took place
- nv50: keep track of PGRAPH state in nv50_screen
- nvc0: keep track of PGRAPH state in nvc0_screen
- nvc0: reset the instanced elements state when doing blit using 3d engine
- nv50/ir: only enable mul saturate on G200+
- st/mesa: make sure to create a "clean" bool when doing i2b
- nvc0: switch mechanism for shader eviction to be a while loop

Jeremy Huddleston Sequoia (2):

- swrast: Build fix for darwin
- darwin: Fix install name of libOSMesa

Laura Ekstrand (2):

- main: Fix an error generated by FramebufferTexture
- main: Complete error conditions for glInvalidate*Framebuffer.

Marta Lofstedt (1):

- main: glGetIntegeri_v fails for GL_VERTEX_BINDING_STRIDE

Rob Clark (2):

- freedreno: enable a306
- freedreno: fix bug in tile/slot calculation

Roland Scheidegger (1):

- draw: (trivial) fix out-of-bounds vector initialization

Tim Rowley (1):

- mesa: fix shininess check for ffvertex_prog v2

Tom Stellard (2):

- clover: Add a mutex to guard queue::queued_events
- clover: Fix a bug with multi-threaded events v2

4.172 Mesa 10.5.5 Release Notes / May 11, 2015

Mesa 10.5.5 is a bug fix release which fixes bugs found since the 10.5.4 release.

Mesa 10.5.5 implements the OpenGL 3.3 API, but the version reported by glGetString(GL_VERSION) or glGetIntegerv(GL_MAJOR_VERSION) / glGetIntegerv(GL_MINOR_VERSION) depends on the particular driver being used. Some drivers don't support all the features required in OpenGL 3.3. OpenGL 3.3 is **only** available if requested at context creation because compatibility contexts are not supported.

4.172.1 SHA256 checksums

```
c10f00fd792b8290dd51ebcc48a9016c4cafab19ec205423c6fcadfd7f3a59f2  mesa-10.5.5.tar.gz
4ac4e4ea3414f1cadb1467f2f173f9e56170d31e8674f7953a46f0549d319f28  mesa-10.5.5.tar.xz
```

4.172.2 New features

None

4.172.3 Bug fixes

This list is likely incomplete.

- Bug 88521 - GLBenchmark 2.7 TRex renders with artifacts on Gen8 with !UXA
- Bug 89455 - [NVC0/Gallium] Unigine Heaven black and white boxes
- Bug 89689 - [Regression] Weston on DRM backend won't start with new version of mesa
- Bug 90130 - gl_PrimitiveId seems to reset at 340

4.172.4 Changes

Boyan Ding (1):

- i965: Add XRGB8888 format to intel_screen_make_configs

Emil Velikov (3):

- docs: Add sha256 sums for the 10.5.4 release
- r300: do not link against libdrm_intel
- Update version to 10.5.5

Ilia Mirkin (4):

- nvc0/ir: flush denorms to zero in non-compute shaders
- gk110/ir: fix set with a register dest to not auto-set the abs flag
- nvc0/ir: fix predicated PFETCH emission
- nv50/ir: fix asFlow() const helper for OP_JOIN

Kenneth Graunke (2):

- i965: Make intel_emit_linear_blit handle Gen8+ alignment restrictions.
- i965: Disallow linear blits that are not cacheline aligned.

Roland Scheidegger (1):

- draw: fix prim ids when there's no gs

4.173 Mesa 10.5.4 Release Notes / April 24, 2015

Mesa 10.5.4 is a bug fix release which fixes bugs found since the 10.5.3 release.

Mesa 10.5.4 implements the OpenGL 3.3 API, but the version reported by glGetString(GL_VERSION) or glGetIntegerv(GL_MAJOR_VERSION) / glGetIntegerv(GL_MINOR_VERSION) depends on the particular driver being used. Some drivers don't support all the features required in OpenGL 3.3. OpenGL 3.3 is **only** available if requested at context creation because compatibility contexts are not supported.

4.173.1 SHA256 checksums

```
e1089567fc7bf8d9b2d8badcc9f2fc3b758701c8c0ccfe7af1805549fea53f11  mesa-10.5.4.tar.gz
b51e723f3a20d842c88a92d809435b229fc4744ca0dbec0317d9d4a3ac4c6803  mesa-10.5.4.tar.xz
```

4.173.2 New features

None

4.173.3 Bug fixes

This list is likely incomplete.

- Bug 69226 - Cannot enable basic shaders with Second Life aborts attempt
- Bug 71591 - Second Life shaders fail to compile (extension declared in middle of shader)
- Bug 81025 - [IVB/BYT Bisected]Piglit spec_ARB_draw_indirect_arb_draw_indirect-draw-elements-prim-restart-ugly fails
- Bug 89457 - [BSW Bisected]ogles3conform ES3-CTS.gtf.GL3Tests.shadow.shadow_execution_vert fails
- Bug 89957 - vm protection faults in piglit lest: texsubimage cube_map_array pbo

4.173.4 Changes

Brian Paul (1):

- glsl: rewrite glsl_type::record_key_hash() to avoid buffer overflow

Dave Airlie (2):

- st/mesa: convert sub image for cube map arrays to 2d arrays for upload
- st/mesa: align cube map arrays layers

Emil Velikov (11):

- docs: Add 256 sums for the 10.5.3 release
- radeonsi: remove unused si_dump_key()
- android: use LOCAL_SHARED_LIBRARIES over TARGET_OUT_HEADERS
- android: add $(mesa_top)/src include to the whole of mesa
- android: egl: add libsync_cflags to the build
- android: dri/common: conditionally include drm_cflags/set __NOT_HAVE_DRM_H

- android: add HAVE__BUILTIN_* and HAVE_FUNC_ATTRIBUTE_* defines
- android: add $(mesa_top)/src/mesa/main to the includes list
- android: dri: link against libmesa_util
- android: mesa: fix the path of the SSE4_1 optimisations
- Update version to 10.5.4

Ian Romanick (1):

- nir: Fix typo in "ushr by 0" algebraic replacement

Kenneth Graunke (2):

- i965: Fix software primitive restart with indirect draws.
- drirc: Add "Second Life" quirk (allow_glsl_extension_directive_midshader).

Kristian Høgsberg (1):

- i965: Rewrite ir_tex to ir_txl with lod 0 for vertex shaders

Marek Olšák (2):

- glsl_to_tgsi: fix out-of-bounds constant access and crash for uniforms
- glsl_to_tgsi: don't use a potentially-undefined immediate for ir_query_levels

Mathias Froehlich (1):

- i965: Flush batchbuffer containing the query on glQueryCounter.

Mauro Rossi (2):

- android: mesa: generate the format_{un,}pack.[ch] sources
- android: add inital NIR build

4.174 Mesa 10.5.3 Release Notes / April 12, 2015

Mesa 10.5.3 is a bug fix release which fixes bugs found since the 10.5.2 release.

Mesa 10.5.3 implements the OpenGL 3.3 API, but the version reported by glGetString(GL_VERSION) or glGetIntegerv(GL_MAJOR_VERSION) / glGetIntegerv(GL_MINOR_VERSION) depends on the particular driver being used. Some drivers don't support all the features required in OpenGL 3.3. OpenGL 3.3 is **only** available if requested at context creation because compatibility contexts are not supported.

4.174.1 SHA256 checksums

```
2371b8e210ccd19f61dd94b6664d612e5a479ba7d431a074512d87633bd6aeb4  mesa-10.5.3.tar.gz
8701ee1be4f5c03238f5e63c1a9bd4cc03a2f6c0155ed42a1ae7d58f18912ba2  mesa-10.5.3.tar.xz
```

4.174.2 New features

None

4.174.3 Bug fixes

This list is likely incomplete.

- Bug 83962 - [HSW/BYT]Piglit spec_ARB_gpu_shader5_arb_gpu_shader5-emitstreamvertex_nodraw fails
- Bug 89679 - [NV50] Portal/Half-Life 2 will not start (native Steam)
- Bug 89746 - Mesa and LLVM 3.6+ break opengl for genymotion
- Bug 89754 - vertexAttrib fails WebGL Conformance test with mesa drivers
- Bug 89758 - pow WebGL Conformance test with mesa drivers
- Bug 89759 - WebGL OGL ES GLSL conformance test with mesa drivers fails
- Bug 89905 - scons build broken on 10.5.2 due to activated vega st

4.174.4 Changes

Dave Airlie (1):

- st_glsl_to_tgsi: only do mov copy propagation on temps (v2)

Emil Velikov (5):

- docs: Add sha256 sums for the 10.5.2 release
- xmlpool: don't forget to ship the MOS
- configure.ac: error out if python/mako is not found when required
- dist: add the VG depedencies into the tarball
- Update version to 10.5.3

Iago Toral Quiroga (1):

- i965: Do not render primitives in non-zero streams then TF is disabled

Ilia Mirkin (7):

- st/mesa: update arrays when the current attrib has been updated
- nv50/ir: take postFactor into account when doing peephole optimizations
- nv50/ir/gk110: fix offset flag position for TXD opcode
- freedreno/a3xx: fix 3d texture layout
- freedreno/a3xx: point size should not be divided by 2
- nv50: allocate more offset space for occlusion queries
- nv50,nvc0: limit the y-tiling of 3d textures to the first level's tiling

Kenneth Graunke (2):

- i965: Fix instanced geometry shaders on Gen8+.
- i965: Add forgotten multi-stream code to Gen8 SOL state.

Marcin Ślusarz (1):

- nouveau: synchronize "scratch runout" destruction with the command stream

Michel Dänzer (1):

- radeonsi: Cache LLVMTargetMachineRef in context instead of in screen

Tom Stellard (1):

- clover: Return CL_BUILD_ERROR for CL_PROGRAM_BUILD_STATUS when compilation fails v2

Ville Syrjälä (1):

- i965: Fix URB size for CHV

4.175 Mesa 10.5.2 Release Notes / March 28, 2015

Mesa 10.5.2 is a bug fix release which fixes bugs found since the 10.5.1 release.

Mesa 10.5.2 implements the OpenGL 3.3 API, but the version reported by glGetString(GL_VERSION) or glGetIntegerv(GL_MAJOR_VERSION) / glGetIntegerv(GL_MINOR_VERSION) depends on the particular driver being used. Some drivers don't support all the features required in OpenGL 3.3. OpenGL 3.3 is **only** available if requested at context creation because compatibility contexts are not supported.

4.175.1 SHA256 checksums

```
755220e160a9f22fda0dffd47746f997b6e196d03f8edc390df7793aecaaa541  mesa-10.5.2.tar.gz
2f4b6fb77c3e7d6f861558d0884a3073f575e1e673dad8d1b0624e78e9c4dd44  mesa-10.5.2.tar.xz
```

4.175.2 New features

None

4.175.3 Bug fixes

This list is likely incomplete.

- Bug 88534 - include/c11/threads_posix.h PTHREAD_MUTEX_RECURSIVE_NP not defined
- Bug 89328 - python required to build Mesa release tarballs
- Bug 89530 - FTBFS in loader: missing fstat
- Bug 89590 - Crash in glLinkProgram with shaders with multiple constant arrays
- Bug 89680 - Hard link exist in Mesa 10.5.1 sources

4.175.4 Changes

Anuj Phogat (1):

- glsl: Generate link error for non-matching gl_FragCoord redeclarations

Emil Velikov (7):

- docs: Add sha256 sums for the 10.5.1 release
- automake: add missing egl files to the tarball
- st/egl: don't ship the dri2.c link at the tarball

- loader: include <sys/stat.h> for non-sysfs builds
- auxiliary/os: fix the android build - s/drm_munmap/os_munmap/
- cherry-ignore: add commit non applicable for 10.5
- Update version to 10.5.2

Felix Janda (1):

- c11/threads: Use PTHREAD_MUTEX_RECURSIVE by default

Francisco Jerez (1):

- i965: Set nr_params to the number of uniform components in the VS/GS path.

Ilia Mirkin (2):

- freedreno/a3xx: use the same layer size for all slices
- freedreno: fix slice pitch calculations

Marek Olšák (1):

- radeonsi: increase coords array size for radeon_llvm_emit_prepare_cube_coords

Mario Kleiner (2):

- glx: Handle out-of-sequence swap completion events correctly. (v2)
- mapi: Make private copies of name strings provided by client.

Rob Clark (1):

- freedreno: update generated headers

Samuel Iglesias Gonsalvez (2):

- glsl: optimize (0 cmp x + y) into (-x cmp y).
- configure: Introduce new output variable to ax_check_python_mako_module.m4

Tapani Pälli (1):

- glsl: fix names in lower_constant_arrays_to_uniforms

Tom Stellard (1):

- clover: Return 0 as storage size for local kernel args that are not set v2

4.176 Mesa 10.4.7 Release Notes / March 20, 2015

Mesa 10.4.7 is a bug fix release which fixes bugs found since the 10.4.6 release.

Mesa 10.4.7 implements the OpenGL 3.3 API, but the version reported by glGetString(GL_VERSION) or glGetIntegerv(GL_MAJOR_VERSION) / glGetIntegerv(GL_MINOR_VERSION) depends on the particular driver being used. Some drivers don't support all the features required in OpenGL 3.3. OpenGL 3.3 is **only** available if requested at context creation because compatibility contexts are not supported.

4.176.1 SHA256 checksums

```
9e7b59267199658808f8b33e0410b86fbafbdcd52378658b9df65fac9d24947f  MesaLib-10.4.7.tar.
↪gz
2c351c98671f9a7ab3fd9c601bb7a255801b1580f5dd0992639f99152801b0d2  MesaLib-10.4.7.tar.
↪bz2
d14ac578b5ce16560757b53fbd1cb4d6b34652f8e110e4b10a019adc82e67ffd  MesaLib-10.4.7.zip
```

4.176.2 New features

None

4.176.3 Bug fixes

This list is likely incomplete.

- Bug 79202 - valgrind errors in glsl-fs-uniform-array-loop-unroll.shader_test; random code generation
- Bug 89156 - r300g: GL_COMPRESSED_RED_RGTC1 / ATI1N support broken
- Bug 89224 - Incorrect rendering of Unigine Valley running in VM on VMware Workstation
- Bug 89530 - FTBFS in loader: missing fstat

4.176.4 Changes

Andrey Sudnik (1):

- i965/vec4: Don't lose the saturate modifier in copy propagation.

Daniel Stone (1):

- egl: Take alpha bits into account when selecting GBM formats

Emil Velikov (6):

- docs: Add sha256 sums for the 10.4.6 release
- cherry-ignore: add not applicable/rejected commits
- mesa: rename format_info.c to format_info.h
- loader: include <sys/stat.h> for non-sysfs builds
- auxiliary/os: fix the android build - s/drm_munmap/os_munmap/
- Update version to 10.4.7

Iago Toral Quiroga (1):

- i965: Fix out-of-bounds accesses into pull_constant_loc array

Ilia Mirkin (4):

- freedreno: move fb state copy after checking for size change
- freedreno/ir3: fix array count returned by TXQ
- freedreno/ir3: get the # of miplevels from getinfo
- freedreno: fix slice pitch calculations

Marc-Andre Lureau (1):

- gallium/auxiliary/indices: fix start param

Marek Olšák (4):

- r300g: fix RGTC1 and LATC1 SNORM formats
- r300g: fix a crash when resolving into an sRGB texture
- r300g: fix sRGB->sRGB blits
- radeonsi: increase coords array size for radeon_llvm_emit_prepare_cube_coords

Mario Kleiner (1):

- glx: Handle out-of-sequence swap completion events correctly. (v2)

Matt Turner (2):

- r300g: Use PATH_MAX instead of limiting ourselves to 100 chars.
- r300g: Check return value of snprintf().

Rob Clark (2):

- freedreno/ir3: fix silly typo for binning pass shaders
- freedreno: update generated headers

Samuel Iglesias Gonsalvez (1):

- glsl: optimize (0 cmp x + y) into (-x cmp y).

Stefan Dösinger (1):

- r300g: Fix the ATI1N swizzle (RGTC1 and LATC1)

4.177 Mesa 10.5.1 Release Notes / March 13, 2015

Mesa 10.5.1 is a bug fix release which fixes bugs found since the 10.5.0 release.

Mesa 10.5.1 implements the OpenGL 3.3 API, but the version reported by glGetString(GL_VERSION) or glGetIntegerv(GL_MAJOR_VERSION) / glGetIntegerv(GL_MINOR_VERSION) depends on the particular driver being used. Some drivers don't support all the features required in OpenGL 3.3. OpenGL 3.3 is **only** available if requested at context creation because compatibility contexts are not supported.

4.177.1 SHA256 checksums

```
b5b6256a6d46023e16a675257fd11a0f94d7b3e60a76cf112952da3d0fef8e9b  mesa-10.5.1.tar.gz
ffc51943d15c6812ee7611d053d8980a683fbd6a4986cff567b12cc66637d679  mesa-10.5.1.tar.xz
```

4.177.2 New features

None

4.177.3 Bug fixes

This list is likely incomplete.

- Bug 79202 - valgrind errors in glsl-fs-uniform-array-loop-unroll.shader_test; random code generation
- Bug 84613 - [G965, bisected] piglit regressions : glslparsertest.glsl2
- Bug 86747 - Noise in Football Manager 2014 textures
- Bug 86974 - INTEL_DEBUG=shader_time always asserts in fs_generator::generate_code() when Mesa is built with –enable-debug (= with asserts)
- Bug 88246 - Commit 2881b12 causes 43 DrawElements test regressions
- Bug 88793 - [BDW/BSW Bisected]Piglit/shaders_glsl-max-varyings fails
- Bug 88883 - ir-a2xx.c: variable changed in assert statement
- Bug 88885 - Transform feedback uses incorrect interleaving if a previous draw did not write gl_Position
- Bug 89095 - [SNB/IVB/BYT Bisected]Webglc conformance/glsl/functions/glsl-function-mix-float.html fails
- Bug 89156 - r300g: GL_COMPRESSED_RED_RGTC1 / ATI1N support broken
- Bug 89224 - Incorrect rendering of Unigine Valley running in VM on VMware Workstation
- Bug 89292 - [regression,bisected] incomplete screenshots in some cases
- Bug 89311 - [regression, bisected] dEQP: Added entry points for glCompressedTextureSubImage*D.
- Bug 89312 - [regression, bisected] main: Added entry points for CopyTextureSubImage*D. (d6b7c40cecfe01)
- Bug 89315 - [HSW, regression, bisected] i965/fs: Emit MAD instructions when possible.
- Bug 89317 - [HSW, regression, bisected] i965: Add LINTERP/CINTERP to can_do_cmod() (d91390634)
- Bug 89416 - UE4Editor crash after load project
- Bug 89430 - [g965][bisected] arb_copy_image-targets gl_texture* tests fail

4.177.4 Changes

Andrey Sudnik (1):

- i965/vec4: Don't lose the saturate modifier in copy propagation.

Chris Forbes (1):

- i965/gs: Check newly-generated GS-out VUE map against correct stage

Daniel Stone (1):

- egl: Take alpha bits into account when selecting GBM formats

Emil Velikov (5):

- docs: Add sha256 sums for the 10.5.0 release
- egl/main: no longer export internal function
- cherry-ignore: ignore a few more commits picked without -x
- mapi: fix commit 90411b56f6bc817e229d8801ac0adad6d4e3fb7a
- Update version to 10.5.1

Frank Henigman (1):

- intel: fix EGLImage renderbuffer _BaseFormat

Iago Toral Quiroga (1):

- i965: Fix out-of-bounds accesses into pull_constant_loc array

Ian Romanick (1):

- i965/fs/nir: Use emit_math for nir_op_fpow

Ilia Mirkin (3):

- freedreno: move fb state copy after checking for size change
- freedreno/ir3: fix array count returned by TXQ
- freedreno/ir3: get the # of miplevels from getinfo

Jason Ekstrand (2):

- meta/TexSubImage: Stash everything other than PIXEL_TRANSFER/store in meta_begin
- main/base_tex_format: Properly handle STENCIL_INDEX1/4/16

Kenneth Graunke (8):

- i965: Split Gen4-5 BlitFramebuffer code; prefer BLT over Meta.
- glsl: Mark array access when copying to a temporary for the ?: operator.
- i965/fs: Set force_writemask_all on shader_time instructions.
- i965/fs: Set smear on shader_time diff register.
- i965/fs: Make emit_shader_time_write return rather than emit.
- i965/fs: Make get_timestamp() pass back the MOV rather than emitting it.
- i965/fs: Make emit_shader_time_end() insert before EOT.
- i965/fs: Don't issue FB writes for bound but unwritten color targets.

Laura Ekstrand (2):

- main: Fix target checking for CompressedTexSubImage*D.
- main: Fix target checking for CopyTexSubImage*D.

Marc-Andre Lureau (1):

- gallium/auxiliary/indices: fix start param

Marek Olšák (3):

- r300g: fix RGTC1 and LATC1 SNORM formats
- r300g: fix a crash when resolving into an sRGB texture
- r300g: fix sRGB->sRGB blits

Matt Turner (12):

- i965/vec4: Fix implementation of i2b.
- mesa: Indent break statements and add a missing one.
- mesa: Free memory allocated for luminance in readpixels.
- mesa: Correct backwards NULL check.
- i965: Consider scratch writes to have side effects.

- i965/fs: Don't use backend_visitor::instructions after creating the CFG.
- r300g: Use PATH_MAX instead of limiting ourselves to 100 chars.
- r300g: Check return value of snprintf().
- i965/fs: Don't propagate cmod to inst with different type.
- i965: Tell intel_get_memcpy() which direction the memcpy() is going.
- Revert SHA1 additions.
- i965: Avoid applying negate to wrong MAD source.

Neil Roberts (4):

- meta: In pbo_{Get,}TexSubImage don't repeatedly rebind the source tex
- Revert "common: Fix PBOs for 1D_ARRAY."
- meta: Allow GL_UN/PACK_IMAGE_HEIGHT in _mesa_meta_pbo_Get/TexSubImage
- meta: Fix the y offset for 1D_ARRAY in _mesa_meta_pbo_TexSubImage

Rob Clark (11):

- freedreno/ir3: fix silly typo for binning pass shaders
- freedreno/a2xx: fix increment in assert
- freedreno/a4xx: bit of cleanup
- freedreno: update generated headers
- freedreno/a4xx: set PC_PRIM_VTX_CNTL.VAROUT properly
- freedreno: update generated headers
- freedreno/a4xx: aniso filtering
- freedreno/ir3: fix up cat6 instruction encodings
- freedreno/ir3: add support for memory (cat6) instructions
- freedreno/ir3: handle flat bypass for a4xx
- freedreno/ir3: fix failed assert in grouping

Stefan Dösinger (1):

- r300g: Fix the ATI1N swizzle (RGTC1 and LATC1)

4.178 Mesa 10.5.0 Release Notes / March 06, 2015

Mesa 10.5.0 is a new development release. People who are concerned with stability and reliability should stick with a previous release or wait for Mesa 10.5.1.

Mesa 10.5.0 implements the OpenGL 3.3 API, but the version reported by glGetString(GL_VERSION) or glGetIntegerv(GL_MAJOR_VERSION) / glGetIntegerv(GL_MINOR_VERSION) depends on the particular driver being used. Some drivers don't support all the features required in OpenGL 3.3. OpenGL 3.3 is **only** available if requested at context creation because compatibility contexts are not supported.

4.178.1 SHA256 checksums

```
2bb6e2e982ee4d8264d52d638c2a4e3f8a164190336d72d4e34ae1304d87ed91  mesa-10.5.0.tar.gz
d7ca9f9044bbdd674377e3eebceef1fae339c8817b9aa435c2053e4fea44e5d3  mesa-10.5.0.tar.xz
```

4.178.2 New features

Note: some of the new features are only available with certain drivers.

- GL_ARB_framebuffer_sRGB on freedreno
- GL_ARB_texture_rg on freedreno
- GL_EXT_packed_float on freedreno
- GL_EXT_polygon_offset_clamp on i965, nv50, nvc0, r600, radeonsi, llvmpipe
- GL_EXT_texture_shared_exponent on freedreno
- GL_EXT_texture_snorm on freedreno

4.178.3 Bug fixes

This list is likely incomplete.

- Bug 10370 - Incorrect pixels read back if draw bitmap texture through Display list
- Bug 45348 - [swrast] piglit fbo-drawbuffers-arbfp regression
- Bug 60879 - [radeonsi] X11 can't start with acceleration enabled
- Bug 67672 - [llvmpipe] lp_test_arit fails on old CPUs
- Bug 77544 - i965: Try to use LINE instructions to perform MAD with immediate arguments
- Bug 78770 - [SNB bisected]Webglc conformance/textures/texture-size-limit.html fails
- Bug 80568 - [gen4] GPU Crash During Google Chrome Operation
- Bug 82477 - [softpipe] piglit fp-long-alu regression
- Bug 82585 - geometry shader with optional out variable segfaults
- Bug 82991 - Inverted bumpmap in webgl applications
- Bug 83463 - [swrast] piglit glsl-vs-clamp-1 regression
- Bug 83500 - si_dma_copy_tile causes GPU hangs
- Bug 83510 - Graphical glitches in Unreal Engine 4
- Bug 83908 - [i965] Incorrect icon colors in Steam Big Picture
- Bug 84212 - [BSW]ES3-CTS.shaders.loops.do_while_dynamic_iterations.vector_counter_vertex fails and causes GPU hang
- Bug 84651 - Distorted graphics or black window when running Battle.net app on Intel hardware via wine
- Bug 84777 - [BSW]Piglit spec_glsl-1.50_execution_geometry-basic fails
- Bug 85367 - [gen4] GPU hang in glmark-es2
- Bug 85467 - [llvmpipe] piglit gl-1.0-dlist-beginend failure with llvm-3.6.0svn

- Bug 85529 - Surfaces not drawn in Unvanquished
- Bug 85647 - Random radeonsi crashes with mesa 10.3.x
- Bug 85696 - r600g+nine: Bioshock shader failure after 7b1c0cbc90d456384b0950ad21faa3c61a6b43ff
- Bug 86089 - [r600g][mesa 10.4.0-dev] shader failure - r600_sb::bc_finalizer::cf_peephole() when starting Second Life
- Bug 86618 - [NV96] neg modifiers not working in MIN and MAX operations
- Bug 86760 - mesa doesn't build: recipe for target 'r600_llvm.lo' failed
- Bug 86764 - [SNB+ Bisected]Piglit glean/pointSprite fails
- Bug 86788 - (bisected) 32bit UrbanTerror 4.1 timedemo sse4.1 segfault...
- Bug 86811 - [BDW/BSW Bisected]Piglit spec_arb_shading_language_packing_execution_built-in-functions_vs-unpackSnorm4x8 fails
- Bug 86837 - kodi segfault since auxiliary/vl: rework the build of the VL code
- Bug 86939 - test_vf_float_conversions.cpp:63:12: error: expected primary-expression before 'union'
- Bug 86944 - glsl_parser_extras.cpp", line 1455: Error: Badly formed expression. (Oracle Studio)
- Bug 86958 - lp_bld_misc.cpp:503:40: error: no matching function for call to 'llvm::EngineBuilder::setMCJITMemoryManager(ShaderMemoryManager*&)'
- Bug 86969 - _drm_intel_gem_bo_references() function takes half the CPU with Witcher2 game
- Bug 87076 - Dead Island needs allow_glsl_extension_directive_midshader
- Bug 87516 - glProgramBinary violates spec
- Bug 87619 - Changes to state such as render targets change fragment shader without marking it dirty.
- Bug 87658 - [llvmpipe] SEGV in sse2_has_daz on ancient Pentium4-M
- Bug 87694 - [SNB] Crash in brw_begin_transform_feedback
- Bug 87886 - constant fps drops with Intel and Radeon
- Bug 87887 - [i965 Bisected]ES2-CTS.gtf.GL.cos.cos_float_vert_xvary fails
- Bug 87913 - CPU cacheline size of 0 can be returned by CPUID leaf 0x80000006 in some virtual machines
- Bug 88079 - dEQP-GLES3.functional.fbo.completeness.renderable.renderbuffer.color0 tests fail due to enabling of GL_RGB and GL_RGBA
- Bug 88170 - 32 bits opengl apps crash with latest llvm 3.6 git / mesa git / radeonsi
- Bug 88219 - include/c11/threads_posix.h:197: undefined reference to 'pthread_mutex_lock'
- Bug 88227 - Radeonsi: High GTT usage in Prison Architect large map
- Bug 88248 - Calling glClear while there is an occlusion query in progress messes up the results
- Bug 88335 - format_pack.c:9567:22: error: expected ')'
- Bug 88385 - [SNB+ Bisected]Ogles3conform ES3-CTS.gtf.GL3Tests.packed_pixels.packed_pixels core dumped
- Bug 88467 - nir.c:140: error: 'nir_src' has no member named 'ssa'
- Bug 88478 - #error "<malloc.h> has been replaced by <stdlib.h>"
- Bug 88519 - sha1.c:210:22: error: 'grcy_md_hd_t' undeclared (first use in this function)
- Bug 88523 - sha1.c:37: error: 'SHA1_CTX' undeclared (first use in this function)

- Bug 88561 - [radeonsi][regression,bisected] Depth test/buffer issues in Portal
- Bug 88658 - (bisected) Slow video playback on Kabini
- Bug 88662 - unaligned access to gl_dlist_node
- Bug 88783 - FTBFS: Clover: src/gallium/state_trackers/clover/llvm/invocation.cpp:335:49: error: no matching function for call to 'llvm::TargetLibraryInfo::TargetLibraryInfo(llvm::Triple)
- Bug 88792 - [BDW/BSW Bisected]Piglit spec_ARB_pixel_buffer_object_pbo-read-argb8888 fails
- Bug 88806 - nir/nir_constant_expressions.c:2754:15: error: controlling expression type 'unsigned int' not compatible with any generic association type
- Bug 88841 - [SNB/IVB/HSW/BDW Bisected]Piglit spec_EGL_NOK_texture_from_pixmap_basic fails
- Bug 88852 - macros.h(181) : error C2143: syntax error : missing '{' before 'enum [tag]'
- Bug 88905 - [SNB+ Bisected]Ogles3conform ES3-CTS.gtf.GL3Tests.packed_pixels.packed_pixels fails
- Bug 88930 - [osmesa] osbuffer->textures should be indexed by attachment type
- Bug 88962 - [osmesa] Crash on postprocessing if z buffer is NULL
- Bug 89032 - [BDW/BSW/SKL Bisected]Piglit spec_OpenGL_1.1_infinite-spot-light fails
- Bug 89037 - [SKL]Piglit spec_EXT_texture_array_copyteximage_1D_ARRAY_samples=2 sporadically causes GPU hang
- Bug 89068 - glTexImage2D regression by texstore_rgba switch to _mesa_format_convert
- Bug 89069 - Lack of grass in The Talos Principle on radeonsi (nativewinenine)
- Bug 89180 - [IVB regression] Rendering issues in Mass Effect through VMware Workstation
- Bug 86330 - lp_bld_debug.cpp:112: multiple definition of 'raw_debug_ostream::write_impl(char const*, unsigned long)'

4.178.4 Changes

- Removed support for GCC versions earlier than 4.2.0.

4.179 Mesa 10.4.6 Release Notes / March 06, 2015

Mesa 10.4.6 is a bug fix release which fixes bugs found since the 10.4.5 release.

Mesa 10.4.6 implements the OpenGL 3.3 API, but the version reported by glGetString(GL_VERSION) or glGetIntegerv(GL_MAJOR_VERSION) / glGetIntegerv(GL_MINOR_VERSION) depends on the particular driver being used. Some drivers don't support all the features required in OpenGL 3.3. OpenGL 3.3 is **only** available if requested at context creation because compatibility contexts are not supported.

4.179.1 SHA256 checksums

```
46c9082142e811c01e49a2c332a9ac0a1eb98f2908985fb9df216539d7eaeaf4  MesaLib-10.4.6.tar.
↪gz
d8baedd20e79ccd98a5a7b05e23d59a30892e68de1fcc057ca6873dafca02735  MesaLib-10.4.6.tar.
↪bz2
6aded6eac7f0d4d55117b8b581d8424710bbb4c768fc90f7b881f29311a751aa  MesaLib-10.4.6.zip
```

4.179.2 New features

None

4.179.3 Bug fixes

This list is likely incomplete.

- Bug 45348 - [swrast] piglit fbo-drawbuffers-arbfp regression
- Bug 84613 - [G965, bisected] piglit regressions : glslparsertest.glsl2
- Bug 87516 - glProgramBinary violates spec
- Bug 88885 - Transform feedback uses incorrect interleaving if a previous draw did not write gl_Position
- Bug 89180 - [IVB regression] Rendering issues in Mass Effect through VMware Workstation

4.179.4 Changes

Abdiel Janulgue (2):

- glsl: Don't optimize min/max into saturate when EmitNoSat is set
- st/mesa: For vertex shaders, don't emit saturate when SM 3.0 is unsupported

Andreas Boll (1):

- glx: Fix returned values of GLX_RENDERER_PREFERRED_PROFILE_MESA

Brian Paul (2):

- swrast: fix multiple color buffer writing
- st/mesa: fix sampler view reference counting bug in glDraw/CopyPixels

Chris Forbes (1):

- i965/gs: Check newly-generated GS-out VUE map against correct stage

Eduardo Lima Mitev (1):

- mesa: Fix error validating args for TexSubImage3D

Emil Velikov (6):

- docs: Add sha256 sums for the 10.4.5 release
- install-lib-links: remove the .install-lib-links file
- Revert "mesa: Correct backwards NULL check."
- mesa: cherry-pick the second half of commit 2aa71e9485a
- Revert "gallivm: Update for RTDyldMemoryManager becoming an unique_ptr."
- Update version to 10.4.6

Ian Romanick (3):

- mesa: Add missing error checks in _mesa_ProgramBinary
- mesa: Ensure that length is set to zero in _mesa_GetProgramBinary
- mesa: Always generate GL_INVALID_OPERATION in _mesa_GetProgramBinary

Jonathan Gray (1):

- auxilary/os: correct sysctl use in os_get_total_physical_memory()

José Fonseca (1):

- gallivm: Update for RTDyldMemoryManager becoming an unique_ptr.

Leo Liu (1):

- st/omx/dec/h264: fix picture out-of-order with poc type 0 v2

Lucas Stach (1):

- install-lib-links: don't depend on .libs directory

Marek Olšák (2):

- vbo: fix an unitialized-variable warning
- radeonsi: fix point sprites

Matt Turner (4):

- glsl: Rewrite and fix min/max to saturate optimization.
- mesa: Correct backwards NULL check.
- i965/fs: Don't use backend_visitor::instructions after creating the CFG.
- mesa: Correct backwards NULL check.

4.180 Mesa 10.4.5 Release Notes / February 21, 2015

Mesa 10.4.5 is a bug fix release which fixes bugs found since the 10.4.4 release.

Mesa 10.4.5 implements the OpenGL 3.3 API, but the version reported by glGetString(GL_VERSION) or glGetIntegerv(GL_MAJOR_VERSION) / glGetIntegerv(GL_MINOR_VERSION) depends on the particular driver being used. Some drivers don't support all the features required in OpenGL 3.3. OpenGL 3.3 is **only** available if requested at context creation because compatibility contexts are not supported.

4.180.1 SHA256 checksums

```
e12bbdaee9a758617e8ebd0bb0e987f72addd11db2e4da25ba695e386cd63843  MesaLib-10.4.5.tar.
↪gz
bf60000700a9d58e3aca2bfeee7e781053b0d839e61a95b1883e05a2dee247a0  MesaLib-10.4.5.tar.
↪bz2
3b926de8eee500bb67cf85332c51292f826cc539b8636382aadbb8e70c76527a  MesaLib-10.4.5.zip
```

4.180.2 New features

None

4.180.3 Bug fixes

This list is likely incomplete.

- Bug 82477 - [softpipe] piglit fp-long-alu regression
- Bug 88658 - (bisected) Slow video playback on Kabini
- Bug 89069 - Lack of grass in The Talos Principle on radeonsi (nativewinenine)

4.180.4 Changes

Carl Worth (1):

- Revert use of Mesa IR optimizer for ARB_fragment_programs

Emil Velikov (3):

- docs: Add sha256 sums for the 10.4.4 release
- get-pick-list.sh: Require explicit "10.4" for nominating stable patches
- Update version to 10.4.5

Ilia Mirkin (3):

- nvc0: bail out of 2d blits with non-A8_UNORM alpha formats
- st/mesa: treat resource-less xfb buffers as if they weren't there
- nvc0: allow holes in xfb target lists

Jeremy Huddleston Sequoia (2):

- darwin: build fix
- darwin: build fix

Kenneth Graunke (4):

- i965: Override swizzles for integer luminance formats.
- i965: Use a gl_color_union for sampler border color.
- i965: Fix integer border color on Haswell.
- glsl: Reduce memory consumption of copy propagation passes.

Laura Ekstrand (1):

- main: Fixed _mesa_GetCompressedTexImage_sw to copy slices correctly.

Marek Olšák (5):

- r600g,radeonsi: don't append to streamout buffers that haven't been used yet
- radeonsi: fix instanced arrays with non-zero start instance
- radeonsi: small fix in SPI state
- mesa: fix AtomicBuffer typo in _mesa_DeleteBuffers
- radeonsi: fix a crash if a stencil ref state is set before a DSA state

Michel Dänzer (2):

- st/mesa: Don't use PIPE_USAGE_STREAM for GL_PIXEL_UNPACK_BUFFER_ARB

- Revert "radeon/llvm: enable unsafe math for graphics shaders"

4.181 Mesa 10.4.4 Release Notes / February 06, 2015

Mesa 10.4.4 is a bug fix release which fixes bugs found since the 10.4.3 release.

Mesa 10.4.4 implements the OpenGL 3.3 API, but the version reported by glGetString(GL_VERSION) or glGetIntegerv(GL_MAJOR_VERSION) / glGetIntegerv(GL_MINOR_VERSION) depends on the particular driver being used. Some drivers don't support all the features required in OpenGL 3.3. OpenGL 3.3 is **only** available if requested at context creation because compatibility contexts are not supported.

4.181.1 SHA256 checksums

```
5cb427eaf980cb8555953e9928f5797979ed783e277745d5f8cbae8bc5364086  MesaLib-10.4.4.tar.
↪gz
f18a967e9c4d80e054b2fdff8c130ce6e6d1f8eecfc42c9f354f8628d8b4df1c  MesaLib-10.4.4.tar.
↪bz2
86baad73b77920c80fe58402a905e7dd17e3ea10ead6ea7d3afdc0a56c860bd7  MesaLib-10.4.4.zip
```

4.181.2 New features

None

4.181.3 Bug fixes

This list is likely incomplete.

- Bug 88662 - unaligned access to gl_dlist_node
- Bug 88930 - [osmesa] osbuffer->textures should be indexed by attachment type

4.181.4 Changes

Brian Paul (1):

- mesa: fix display list 8-byte alignment issue

Emil Velikov (2):

- docs: Add sha256 sums for the 10.4.3 release
- Update version to 10.4.4

José Fonseca (1):

- egl: Pass the correct X visual depth to xcb_put_image().

Mario Kleiner (1):

- glx/dri3: Request non-vsynced Present for swapinterval zero. (v3)

Matt Turner (1):

- gallium/util: Don't use __builtin_clrsb in util_last_bit().

Niels Ole Salscheider (1):

- configure: Link against all LLVM targets when building clover

Park, Jeongmin (1):

- st/osmesa: Fix osbuffer->textures indexing

Ville Syrjälä (1):

- i965: Fix max_wm_threads for CHV

4.182 Mesa 10.4.3 Release Notes / January 24, 2015

Mesa 10.4.3 is a bug fix release which fixes bugs found since the 10.4.2 release.

Mesa 10.4.3 implements the OpenGL 3.3 API, but the version reported by glGetString(GL_VERSION) or glGetIntegerv(GL_MAJOR_VERSION) / glGetIntegerv(GL_MINOR_VERSION) depends on the particular driver being used. Some drivers don't support all the features required in OpenGL 3.3. OpenGL 3.3 is **only** available if requested at context creation because compatibility contexts are not supported.

4.182.1 SHA256 checksums

```
c53eaafc83d9c6315f63e0904d9954d929b841b0b2be7a328eeb6e14f1376129  MesaLib-10.4.3.tar.
↪gz
ef6ecc9c2f36c9f78d1662382a69ae961f38f03af3a0c3268e53f351aa1978ad  MesaLib-10.4.3.tar.
↪bz2
179325fc8ec66529d3b0d0c43ef61a33a44d91daa126c3bbdd1efdfd25a7db1d  MesaLib-10.4.3.zip
```

4.182.2 New features

None

4.182.3 Bug fixes

This list is likely incomplete.

- Bug 80568 - [gen4] GPU Crash During Google Chrome Operation
- Bug 85367 - [gen4] GPU hang in glmark-es2
- Bug 85696 - r600g+nine: Bioshock shader failure after 7b1c0cbc90d456384b0950ad21faa3c61a6b43ff
- Bug 88219 - include/c11/threads_posix.h:197: undefined reference to 'pthread_mutex_lock'

4.182.4 Changes

Axel Davy (39):

- st/nine: Add new texture format strings
- st/nine: Correctly advertise D3DPMISCCAPS_CLIPTLVERTS
- st/nine: NineBaseTexture9: fix setting of last_layer

- st/nine: CubeTexture: fix GetLevelDesc
- st/nine: Fix crash when deleting non-implicit swapchain
- st/nine: Return D3DERR_INVALIDCALL when trying to create a texture of bad format
- st/nine: NineBaseTexture9: update sampler view creation
- st/nine: Check if srgb format is supported before trying to use it.
- st/nine: Add ATI1 and ATI2 support
- st/nine: Rework of boolean constants
- st/nine: Convert integer constants to floats before storing them when cards don't support integers
- st/nine: Remove some shader unused code
- st/nine: Saturate oFog and oPts vs outputs
- st/nine: Correctly declare NineTranslateInstruction_Mkxn inputs
- st/nine: Fix typo for M4x4
- st/nine: Fix POW implementation
- st/nine: Handle RSQ special cases
- st/nine: Handle NRM with input of null norm
- st/nine: Correct LOG on negative values
- st/nine: Rewrite LOOP implementation, and a0 aL handling
- st/nine: Fix CND implementation
- st/nine: Clamp ps 1.X constants
- st/nine: Fix some fixed function pipeline operation
- st/nine: Implement TEXCOORD special behaviours
- st/nine: Fill missing dst and src number for some instructions.
- st/nine: Fix TEXM3x3 and implement TEXM3x3VSPEC
- st/nine: implement TEXM3x2DEPTH
- st/nine: Implement TEXM3x2TEX
- st/nine: Implement TEXM3x3SPEC
- st/nine: Implement TEXDEPTH
- st/nine: Implement TEXDP3
- st/nine: Implement TEXDP3TEX
- st/nine: Implement TEXREG2AR, TEXREG2GB and TEXREG2RGB
- st/nine: Correct rules for relative adressing and constants.
- st/nine: Remove unused code for ps
- st/nine: Fix sm3 relative addressing for non-debug build
- st/nine: Add variables containing the size of the constant buffers
- st/nine: Allocate the correct size for the user constant buffer
- st/nine: Allocate vs constbuf buffer for indirect addressing once.

Emil Velikov (2):

- docs: Add sha256 sums for the 10.4.2 release
- Update version to 10.4.3

Jason Ekstrand (1):

- mesa: Fix clamping to -1.0 in snorm_to_float

Jonathan Gray (1):

- glsl: Link glsl_test with pthreads library.

Jose Fonseca (1):

- nine: Drop use of TGSI_OPCODE_CND.

Kenneth Graunke (2):

- i965: Respect the no_8 flag on Gen6, not just Gen7+.
- i965: Work around mysterious Gen4 GPU hangs with minimal state changes.

Stanislaw Halik (1):

- st/nine: Hack to generate resource if it doesn't exist when getting view

Xavier Bouchoux (3):

- st/nine: Additional defines to d3dtypes.h
- st/nine: Add missing c++ declaration for IDirect3DVolumeTexture9
- st/nine: Fix D3DRS_POINTSPRITE support

4.183 Mesa 10.4.2 Release Notes / January 12, 2015

Mesa 10.4.2 is a bug fix release which fixes bugs found since the 10.4.1 release.

Mesa 10.4.2 implements the OpenGL 3.3 API, but the version reported by glGetString(GL_VERSION) or glGetIntegerv(GL_MAJOR_VERSION) / glGetIntegerv(GL_MINOR_VERSION) depends on the particular driver being used. Some drivers don't support all the features required in OpenGL 3.3. OpenGL 3.3 is **only** available if requested at context creation because compatibility contexts are not supported.

4.183.1 SHA256 checksums

```
e303e77dd774df0d051b2870b165f98c97084a55980f884731df89c1b56a6146  MesaLib-10.4.2.tar.
↪gz
08a119937d9f2aa2f66dd5de97baffc2a6e675f549e40e699a31f5485d15327f  MesaLib-10.4.2.tar.
↪bz2
c2c2921a80a3395824f02bee4572a6a17d6a12a928a3e497618eeea04fb06490  MesaLib-10.4.2.zip
```

4.183.2 New features

None

4.183.3 Bug fixes

This list is likely incomplete.

- Bug 85529 - Surfaces not drawn in Unvanquished
- Bug 87619 - Changes to state such as render targets change fragment shader without marking it dirty.
- Bug 87658 - [llvmpipe] SEGV in sse2_has_daz on ancient Pentium4-M
- Bug 87913 - CPU cacheline size of 0 can be returned by CPUID leaf 0x80000006 in some virtual machines

4.183.4 Changes

Chad Versace (2):

- i965: Use safer pointer arithmetic in intel_texsubimage_tiled_memcpy()
- i965: Use safer pointer arithmetic in gather_oa_results()

Dave Airlie (3):

- Revert "r600g/sb: fix issues cause by GLSL switching to loops for switch"
- r600g: fix regression since UCMP change
- r600g/sb: implement r600 gpr index workaround. (v3.1)

Emil Velikov (2):

- docs: Add sha256 sums for the 10.4.1 release
- Update version to 10.4.2

Ilia Mirkin (2):

- nv50,nvc0: set vertex id base to index_bias
- nv50/ir: fix texture offsets in release builds

Kenneth Graunke (2):

- i965: Add missing BRW_NEW_*_PROG_DATA to texture/renderbuffer atoms.
- i965: Fix start/base_vertex_location for >1 prims but !BRW_NEW_VERTICES.

Leonid Shatz (1):

- gallium/util: make sure cache line size is not zero

Marek Olšák (4):

- glsl_to_tgsi: fix a bug in copy propagation
- vbo: ignore primitive restart if FixedIndex is enabled in DrawArrays
- st/mesa: fix GL_PRIMITIVE_RESTART_FIXED_INDEX
- radeonsi: fix VertexID for OpenGL

Michel Dänzer (1):

- radeonsi: Don't modify PA_SC_RASTER_CONFIG register value if rb_mask == 0

Roland Scheidegger (1):

- gallium/util: fix crash with daz detection on x86

Tiziano Bacocco (1):

- nv50,nvc0: implement half_pixel_center

Vadim Girlin (1):

- r600g/sb: fix issues with loops created for switch

4.184 Mesa 10.3.7 Release Notes / January 12, 2015

Mesa 10.3.7 is a bug fix release which fixes bugs found since the 10.3.6 release.

Mesa 10.3.7 implements the OpenGL 3.3 API, but the version reported by glGetString(GL_VERSION) or glGetIntegerv(GL_MAJOR_VERSION) / glGetIntegerv(GL_MINOR_VERSION) depends on the particular driver being used. Some drivers don't support all the features required in OpenGL 3.3. OpenGL 3.3 is **only** available if requested at context creation because compatibility contexts are not supported.

4.184.1 SHA256 checksums

```
bc13f33c19bc9f44a0565fdd51a8f9d1c0153a3365c429ceaf4ef43b7022b052  MesaLib-10.3.7.tar.
↪gz
43c6ced15e237cbb21b3082d7c0b42777c50c1f731d0d4b5efb5231063fb6a5b  MesaLib-10.3.7.tar.
↪bz2
d821fd46baf804fecfcf403e901800a4b996c7dd1c83f20a354b46566a49026f  MesaLib-10.3.7.zip
```

4.184.2 New features

None

4.184.3 Bug fixes

This list is likely incomplete.

- Bug 85529 - Surfaces not drawn in Unvanquished
- Bug 87619 - Changes to state such as render targets change fragment shader without marking it dirty.

4.184.4 Changes

Chad Versace (2):

- i965: Use safer pointer arithmetic in intel_texsubimage_tiled_memcpy()
- i965: Use safer pointer arithmetic in gather_oa_results()

Emil Velikov (2):

- docs: Add sha256 sums for the 10.3.6 release
- Update version to 10.3.7

Ilia Mirkin (2):

- nv50,nvc0: set vertex id base to index_bias

- nv50/ir: fix texture offsets in release builds

Kenneth Graunke (2):

- i965: Add missing BRW_NEW_*_PROG_DATA to texture/renderbuffer atoms.
- i965: Fix start/base_vertex_location for >1 prims but !BRW_NEW_VERTICES.

Marek Olšák (3):

- glsl_to_tgsi: fix a bug in copy propagation
- vbo: ignore primitive restart if FixedIndex is enabled in DrawArrays
- st/mesa: fix GL_PRIMITIVE_RESTART_FIXED_INDEX

Michel Dänzer (1):

- radeonsi: Don't modify PA_SC_RASTER_CONFIG register value if rb_mask == 0

4.185 Mesa 10.4.1 Release Notes / December 29, 2014

Mesa 10.4.1 is a bug fix release which fixes bugs found since the 10.4.0 release.

Mesa 10.4.1 implements the OpenGL 3.3 API, but the version reported by glGetString(GL_VERSION) or glGetIntegerv(GL_MAJOR_VERSION) / glGetIntegerv(GL_MINOR_VERSION) depends on the particular driver being used. Some drivers don't support all the features required in OpenGL 3.3. OpenGL 3.3 is **only** available if requested at context creation because compatibility contexts are not supported.

4.185.1 SHA256 checksums

```
5311285e791a6bfaa468ad002bd1e1164acb3eaa040b5a1bf958bdb7c27e0a9d  MesaLib-10.4.1.tar.
↪gz
91e8b71c8aff4cb92022a09a872b1c5d1ae5bfec8c6c84dbc4221333da5bf1ca  MesaLib-10.4.1.tar.
↪bz2
e09c8135f5a86ecb21182c6f8959aafd39ae2f98858fdf7c0e25df65b5abcdb8  MesaLib-10.4.1.zip
```

4.185.2 New features

None

4.185.3 Bug fixes

This list is likely incomplete.

- Bug 82585 - geometry shader with optional out variable segfaults
- Bug 82991 - Inverted bumpmap in webgl applications
- Bug 83908 - [i965] Incorrect icon colors in Steam Big Picture

4.185.4 Changes

Andres Gomez (1):

- i965/brw_reg: struct constructor now needs explicit negate and abs values.

Cody Northrop (1):

- i965: Require pixel alignment for GPU copy blit

Emil Velikov (3):

- docs: Add 10.4 sha256 sums, news item and link release notes
- Revert "glx/dri3: Request non-vsynced Present for swapinterval zero. (v3)"
- Update version to 10.4.1

Ian Romanick (2):

- linker: Wrap access of producer_var with a NULL check
- linker: Assign varying locations geometry shader inputs for SSO

Mario Kleiner (4):

- glx/dri3: Fix glXWaitForSbcOML() to handle targetSBC==0 correctly. (v2)
- glx/dri3: Track separate (ust, msc) for PresentPixmap vs. PresentNotifyMsc (v2)
- glx/dri3: Request non-vsynced Present for swapinterval zero. (v3)
- glx/dri3: Don't fail on glXSwapBuffersMscOML(dpy, window, 0, 0, 0) (v2)

Maxence Le Doré (1):

- glsl: Add gl_MaxViewports to available builtin constants

4.186 Mesa 10.3.6 Release Notes / December 29, 2014

Mesa 10.3.6 is a bug fix release which fixes bugs found since the 10.3.5 release.

Mesa 10.3.6 implements the OpenGL 3.3 API, but the version reported by glGetString(GL_VERSION) or glGetIntegerv(GL_MAJOR_VERSION) / glGetIntegerv(GL_MINOR_VERSION) depends on the particular driver being used. Some drivers don't support all the features required in OpenGL 3.3. OpenGL 3.3 is **only** available if requested at context creation because compatibility contexts are not supported.

4.186.1 SHA256 checksums

```
c4d053d6bc6604cb5c93c99e0ef2e815c539f26dc5a03737eb3809bc1767d12f  MesaLib-10.3.6.tar.
↪gz
8d43673c6788fbf85f9c36c3a95c61ccf46f8835fc9c0d85d34474490d80572b  MesaLib-10.3.6.tar.
↪bz2
6b5b1e9a13949cfdb76fe51e8dcc3ea71e464a5ca73d11fdc29c20c4ba3f411a  MesaLib-10.3.6.zip
```

4.186.2 New features

None

4.186.3 Bug fixes

This list is likely incomplete.

- Bug 60879 - [radeonsi] X11 can't start with acceleration enabled
- Bug 82585 - geometry shader with optional out variable segfaults
- Bug 82991 - Inverted bumpmap in webgl applications
- Bug 84777 - [BSW]Piglit spec_glsl-1.50_execution_geometry-basic fails

4.186.4 Changes

Andres Gomez (1):

- i965/brw_reg: struct constructor now needs explicit negate and abs values.

Ben Widawsky (1):

- i965/gs: Avoid DW * DW mul

Dave Airlie (1):

- r600g: only init GS_VERT_ITEMSIZE on r600

Emil Velikov (3):

- docs: Add sha256 sums for the 10.3.5 release
- Revert "glx/dri3: Request non-vsynced Present for swapinterval zero. (v3)"
- Update version to 10.3.6

Ian Romanick (2):

- linker: Wrap access of producer_var with a NULL check
- linker: Assign varying locations geometry shader inputs for SSO

Ilia Mirkin (3):

- util/primconvert: pass index bias through
- util/primconvert: support instanced rendering
- util/primconvert: take ib offset into account

José Fonseca (1):

- util/primconvert: Avoid point arithmetic; apply offset on all cases.

Marek Olšák (1):

- docs/relnotes: document the removal of GALLIUM_MSAA

Mario Kleiner (4):

- glx/dri3: Fix glXWaitForSbcOML() to handle targetSBC==0 correctly. (v2)
- glx/dri3: Track separate (ust, msc) for PresentPixmap vs. PresentNotifyMsc (v2)
- glx/dri3: Request non-vsynced Present for swapinterval zero. (v3)
- glx/dri3: Don't fail on glXSwapBuffersMscOML(dpy, window, 0, 0, 0) (v2)

Maxence Le Doré (1):

- glsl: Add gl_MaxViewports to available builtin constants

Tom Stellard (1):

- radeonsi: Program RASTER_CONFIG for harvested GPUs v5

4.187 Mesa 10.4 Release Notes / December 14, 2014

Mesa 10.4 is a new development release. People who are concerned with stability and reliability should stick with a previous release or wait for Mesa 10.4.1.

Mesa 10.4 implements the OpenGL 3.3 API, but the version reported by glGetString(GL_VERSION) or glGetIntegerv(GL_MAJOR_VERSION) / glGetIntegerv(GL_MINOR_VERSION) depends on the particular driver being used. Some drivers don't support all the features required in OpenGL 3.3. OpenGL 3.3 is **only** available if requested at context creation because compatibility contexts are not supported.

4.187.1 SHA256 checksums

```
abfbfd2d91ce81491c5bb6923ae649212ad5f82d0bee277de8704cc948dc221e  MesaLib-10.4.0.tar.
↪gz
98a7dff3a1a6708c79789de8b9a05d8042e867067f70e8f30387c15026233219  MesaLib-10.4.0.tar.
↪bz2
443a6d46d0691b5ac811d8d30091b1716c365689b16d49c57cf273c2b76086fe  MesaLib-10.4.0.zip
```

4.187.2 New features

Note: some of the new features are only available with certain drivers.

- GL_ARB_conditional_render_inverted on nv50
- GL_ARB_sample_shading on r600
- GL_ARB_texture_view on nv50, nvc0
- GL_ARB_clip_control on nv50, nvc0, r300, r600, radeonsi, llvmpipe, softpipe
- GL_KHR_context_flush_control on all drivers

4.187.3 Bug fixes

This list is likely incomplete.

- Bug 79963 - [ILK Bisected]some piglit and ogles2conform cases fail
- Bug 29661 - MSVC built u_format_test fails on Windows
- Bug 38873 - [855gm] gnome-shell misrendered
- Bug 54372 - GLX_INTEL_swap_event crashes driver when swapping window buffers
- Bug 60879 - [radeonsi] X11 can't start with acceleration enabled
- Bug 61415 - Clover ignores –with-opencl-libdir path
- Bug 64471 - Radeon HD6570 lockup in Brütal Legend with HyperZ

- Bug 66184 - src/mesa/state_tracker/st_glsl_to_tgsi.cpp:3216:simplify_cmp: Assertion 'inst->dst.index < 4096' failed.
- Bug 67672 - [llvmpipe] lp_test_arit fails on old CPUs
- Bug 69200 - [Bisected]Piglit glx/glx-multithread-shader-compile aborted
- Bug 70410 - egl-static/Makefile: linking fails with llvm >= 3.4
- Bug 72685 - [radeonsi hyperz] Artifacts in Unigine Sanctuary
- Bug 72819 - [855GM] Incorrect drop shadow color on windows and strange white rectangle when showing/hiding GLX-dock...
- Bug 74563 - Surfaceless contexts are not properly released by DRI drivers
- Bug 74863 - [r600g] HyperZ broken on RV770 and CYPRESS (Left 4 Dead 2 trees corruption) bisected!
- Bug 75011 - [hyperz] Performance drop since git-01e6371 (disable hyperz by default) with radeonsi
- Bug 75112 - Meta Bug for HyperZ issues on r600g and radeonsi
- Bug 76252 - Dynamic loading/unloading of opengl32.dll results in a deadlock
- Bug 76861 - mid3 generates slow code for constant arguments
- Bug 77957 - Variably-indexed constant arrays result in terrible shader code
- Bug 78468 - Compiling of shader gets stuck in infinite loop
- Bug 78770 - [SNB bisected]Webglc conformance/textures/texture-size-limit.html fails
- Bug 79155 - [Tesseract Game] Global Illumination: Medium Causes Color Distortion
- Bug 79462 - [NVC0/Codegen] Shader compilation falis in spill logic
- Bug 80011 - [softpipe] tgsi/tgsi_exec.c:2023:exec_txf: Assertion '0' failed.
- Bug 80012 - [softpipe] draw/draw_gs.c:113:tgsi_fetch_gs_outputs: Assertion '!util_is_inf_or_nan(output[slot][0])' failed.
- Bug 80050 - [855GM] Incorrect drop shadow color under windows in Cinnamon persists with MESA 10.1.
- Bug 80247 - Khronos conformance test ES3-CTS.gtf.GL3Tests.transform_feedback.transform_feedback_vertex_id fails
- Bug 80561 - Incorrect implementation of some VDPAU APIs.
- Bug 80615 - Files in bellagio directory [omx tracker] don't respect installation folder
- Bug 80848 - [dri3] Building mesa fails with dri3 enabled
- Bug 81680 - [r600g] Firefox crashes with hardware acceleration turned on
- Bug 82255 - [VP2] Chroma planes are vertically stretched during VDPAU playback
- Bug 82472 - piglit 16385-consecutive-chars regression
- Bug 82537 - Stunt Rally GLSL compiler assertion failure
- Bug 82538 - Super Maryo Chronicles fails with st/mesa assertion failure
- Bug 82539 - vmw_screen_dri.lo In file included from vmw_screen_dri.c:41: vmwgfx_drm.h:32:17: error: drm.h: No such file or directory
- Bug 82796 - [IVB/BYT-M/HSW/BDW Bisected]Synmark2_v6.0_OglTerrainFlyInst/OglTerrainPanInst cannot run as image validation failed
- Bug 82804 - unreal engine 4 rendering errors

- Bug 82828 - Regression: Crash in 3Dmark2001
- Bug 82846 - [BDW Bisected] Gpu hang when running Lightsmark v2008/Warsow v1.0/Xonotic v0.7/unigine-demos
- Bug 82881 - test_vec4_register_coalesce regression
- Bug 82882 - [swrast] piglit glsl-fs-uniform-bool-1 regression
- Bug 82921 - layout(location=0) emits error >= MAX_UNIFORM_LOCATIONS due to integer underflow
- Bug 82929 - [BDW Bisected]glxgears causes X hang
- Bug 82932 - [SNB+ Bisected]Ogles3conform ES3-CTS.shaders.indexing.vector_subscript.vec3_static_loop_subscript_write_dire fails
- Bug 83079 - [NVC0] Dota 2 (Linux native and Wine) crash with Nouveau Drivers
- Bug 83080 - [SNB+ Bisected]ES3-CTS.shaders.loops.do_while_constant_iterations.mixed_break_continue_fragment fails
- Bug 83081 - [BDW Bisected]Piglit spec_ARB_sample_shading_builtin-gl-sample-mask_2 is core dumped
- Bug 83127 - [ILK Bisected]Piglit glean_texCombine fails
- Bug 83148 - Unity invisible under Ubuntu 14.04 and 14.10
- Bug 83355 - FTBFS: src/mesa/program/program_lexer.l:122:64: error: unknown type name 'YYSTYPE'
- Bug 83380 - Linking fails when not writing gl_Position.
- Bug 83418 - EU IV is incorrectly rendered after git1409011930.d571f2
- Bug 83432 - r600_query.c:269:r600_emit_query_end: Assertion 'ctx->num_pipelinestat_queries > 0' failed [Gallium HUD]
- Bug 83463 - [swrast] piglit glsl-vs-clamp-1 regression
- Bug 83468 - [UBO] Using bool from UBO as if-statement condition asserts
- Bug 83500 - si_dma_copy_tile causes GPU hangs
- Bug 83506 - [UBO] row_major layout ignored inside structures
- Bug 83533 - [UBO] nested structures don't get appropriate padding
- Bug 83573 - [swrast] piglit fs-op-not-bool-using-if regression
- Bug 83574 - [llvmpipe] [softpipe] piglit arb_explicit_uniform_location-use-of-unused-loc regression
- Bug 83741 - [UBO] row_major layout partially ignored for arrays of structures
- Bug 83777 - [regression] ilo fails to build
- Bug 83934 - Structures must have same name to be considered same type.
- Bug 84140 - mplayer crashes playing some files using vdpau output
- Bug 84145 - UE4: Realistic Rendering Demo render blue
- Bug 84178 - Big glamor regression in Xorg server 1.6.99.1 GIT: x11perf 1.5 Test: PutImage XY 500x500 Square
- Bug 84355 - texture2DProjLod and textureCubeLod are not supported when using GLES.
- Bug 84529 - [IVB bisected] glean fragProg1 CMP test failed
- Bug 84538 - lp_test_format.c:226:4: error: too few arguments to function 'gallivm_create'

- Bug 84539 - brw_fs_register_coalesce.cpp:183: bool fs_visitor::register_coalesce(): Assertion 'src_size <= 11' failed.
- Bug 84557 - [HSW] "Emit ELSE/ENDIF JIP with type D on Gen 7" causes Atomic Afterlife and GPU hangs
- Bug 84651 - Distorted graphics or black window when running Battle.net app on Intel hardware via wine
- Bug 84662 - Long pauses with Unreal demo Elemental on R9270X since : Always flush the HDP cache before submitting a CS to the GPU
- Bug 84777 - [BSW]Piglit spec_glsl-1.50_execution_geometry-basic fails
- Bug 84807 - Build issue starting between bf4aecfb2acc8d0dc815105d2f36eccbc97c284b and a3e9582f09249ad27716ba82c7dfcee685b65d51
- Bug 85189 - llvm/invocation.cpp: In function 'void {anonymous}::optimize(llvm::Module*, unsigned int, const std::vector<llvm::Function*>&)': llvm/invocation.cpp:324:18: error: expected type-specifier
- Bug 85267 - vlc crashes with vdpau (Radeon 3850HD) [r600]
- Bug 85377 - lp_test_format failure with llvm-3.6
- Bug 85425 - [bisected] Compiler error in clip control operations in meta
- Bug 85429 - indirect.c:296: multiple definition of '__indirect_glNewList'
- Bug 85454 - Unigine Sanctuary with Wine crashes on Mesa Git
- Bug 85647 - Random radeonsi crashes with mesa 10.3.x
- Bug 85683 - [i965 Bisected]Piglit shaders_glsl-vs-raytrace-bug26691 segfault
- Bug 85691 - 'glsl: Drop constant 0.0 components from dot products.' broke piglit shaders/glsl-gnome-shell-dim-window and a few others with Gallium
- Bug 86025 - srcglsllist.h(535) : error C2143: syntax error : missing ';' before 'type'
- Bug 86089 - [r600g][mesa 10.4.0-dev] shader failure - r600_sb::bc_finalizer::cf_peephole() when starting Second Life
- Bug 86145 - Pipeline statistic counter values for VF always 0
- Bug 86618 - [NV96] neg modifiers not working in MIN and MAX operations
- Bug 86760 - mesa doesn't build: recipe for target 'r600_llvm.lo' failed
- Bug 86764 - [SNB+ Bisected]Piglit glean/pointSprite fails
- Bug 86788 - (bisected) 32bit UrbanTerror 4.1 timedemo sse4.1 segfault...

4.187.4 Changes

- The environment variable GALLIUM_MSAA that forced a multisample GLX visual was removed.

4.188 Mesa 10.3.5 Release Notes / December 5, 2014

Mesa 10.3.5 is a bug fix release which fixes bugs found since the 10.3.4 release.

Mesa 10.3.5 implements the OpenGL 3.3 API, but the version reported by glGetString(GL_VERSION) or glGetIntegerv(GL_MAJOR_VERSION) / glGetIntegerv(GL_MINOR_VERSION) depends on the particular driver being used. Some drivers don't support all the features required in OpenGL 3.3. OpenGL 3.3 is **only** available if requested at context creation because compatibility contexts are not supported.

4.188.1 SHA256 checksums

```
7ea71c3cce89114df3dc050376afa1c6f6bf235d77a68f9703273603d6a90621  MesaLib-10.3.5.tar.
↪gz
eb75d2790f1606d59d50a6acaa637b6c75f2155b3e0eca3d5099165c0d9556ae  MesaLib-10.3.5.tar.
↪bz2
164bc64ba63fb07ff255ff8de6ed3c95ff545dfe8f864c44c33abe94788da910  MesaLib-10.3.5.zip
```

4.188.2 New features

None

4.188.3 Bug fixes

This list is likely incomplete.

- Bug 86618 - [NV96] neg modifiers not working in MIN and MAX operations

4.188.4 Changes

Brian Paul (2):

- mesa: fix arithmetic error in _mesa_compute_compressed_pixelstore()
- mesa: fix height error check for 1D array textures

Chris Forbes (2):

- i965: Handle nested uniform array indexing
- mesa: Fix Get(GL_TRANSPOSE_CURRENT_MATRIX_ARB) to transpose

Emil Velikov (2):

- docs: Add sha256 sums for the 10.3.5 release
- Update version to 10.3.5

Ilia Mirkin (6):

- nv50/ir: set neg modifiers on min/max args
- nv50,nvc0: actually check constbufs for invalidation
- nv50,nvc0: buffer resources can be bound as other things down the line
- freedreno/ir3: don't pass consts to madsh.m16 in MOD logic
- freedreno/a3xx: only enable blend clamp for non-float formats
- freedreno/ir3: fix UMAD

Rob Clark (1):

- configure.ac: bump libdrm_freedreno requirement

4.189 Mesa 10.3.4 Release Notes / November 21, 2014

Mesa 10.3.4 is a bug fix release which fixes bugs found since the 10.3.3 release.

Mesa 10.3.4 implements the OpenGL 3.3 API, but the version reported by glGetString(GL_VERSION) or glGetIntegerv(GL_MAJOR_VERSION) / glGetIntegerv(GL_MINOR_VERSION) depends on the particular driver being used. Some drivers don't support all the features required in OpenGL 3.3. OpenGL 3.3 is **only** available if requested at context creation because compatibility contexts are not supported.

4.189.1 SHA256 checksums

```
26482495ef6177f889dbd87c7edcccfedd995598785bbbd7e3e066352574c8e0  MesaLib-10.3.4.tar.
↪gz
e6373913142338d10515daf619d659433bfd2989988198930c13b0945a15e98a  MesaLib-10.3.4.tar.
↪bz2
8c3ebbb6535daf3414305860ebca6ac67dbb6e3d35058c7a6ce18b84b5945b7f  MesaLib-10.3.4.zip
```

4.189.2 New features

None

4.189.3 Bug fixes

This list is likely incomplete.

- Bug 76252 - Dynamic loading/unloading of opengl32.dll results in a deadlock
- Bug 78770 - [SNB bisected]Webglc conformance/textures/texture-size-limit.html fails
- Bug 83500 - si_dma_copy_tile causes GPU hangs
- Bug 85647 - Random radeonsi crashes with mesa 10.3.x

4.189.4 Changes

Brian Paul (1):

- st/mesa: copy sampler_array_size field when copying instructions

Chad Versace (1):

- i965: Fix segfault in WebGL Conformance on Ivybridge

Dave Airlie (5):

- r600g/cayman: fix integer multiplication output overwrite (v2)
- r600g/cayman: fix texture gather tests
- r600g/cayman: handle empty vertex shaders
- r600g: geom shaders: always load texture src regs from inputs
- r600g: limit texture offset application to specific types (v2)

Emil Velikov (3):

- docs: Add sha256 sums for the 10.3.3 release
- configure.ac: roll up a program for the sse4.1 check
- get-pick-list.sh: Require explicit "10.3" for nominating stable patches

Ilia Mirkin (1):

- st/mesa: add a fallback for clear_with_quad when no vs_layer

José Fonseca (1):

- llvmpipe: Avoid deadlock when unloading opengl32.dll

Kenneth Graunke (1):

- i915g: we also have more than 0 viewports!

Michel Dänzer (1):

- radeonsi: Disable asynchronous DMA except for PIPE_BUFFER

4.190 Mesa 10.3.3 Release Notes / November 8, 2014

Mesa 10.3.3 is a bug fix release which fixes bugs found since the 10.3.2 release.

Mesa 10.3.3 implements the OpenGL 3.3 API, but the version reported by glGetString(GL_VERSION) or glGetIntegerv(GL_MAJOR_VERSION) / glGetIntegerv(GL_MINOR_VERSION) depends on the particular driver being used. Some drivers don't support all the features required in OpenGL 3.3. OpenGL 3.3 is **only** available if requested at context creation because compatibility contexts are not supported.

4.190.1 SHA256 checksums

```
23a0c36d88cd5d8968ae6454160de2878192fd1d37b5d606adca1f1b7e788b79  MesaLib-10.3.3.tar.
↪gz
0e4eee4a2ddf86456eed2fc44da367f95471f74249636710491e85cc256c4753  MesaLib-10.3.3.tar.
↪bz2
a83648f17d776b7cf6c813fbb15782d2644b937dc6a7c53d8c0d1b35411f4840  MesaLib-10.3.3.zip
```

4.190.2 New features

None

4.190.3 Bug fixes

This list is likely incomplete.

- Bug 70410 - egl-static/Makefile: linking fails with llvm >= 3.4
- Bug 82921 - layout(location=0) emits error >= MAX_UNIFORM_LOCATIONS due to integer underflow
- Bug 83574 - [llvmpipe] [softpipe] piglit arb_explicit_uniform_location-use-of-unused-loc regression
- Bug 85454 - Unigine Sanctuary with Wine crashes on Mesa Git
- Bug 85918 - Mesa: MSVC 2010/2012 Compile error

4.190.4 Changes

Anuj Phogat (2):

- glsl: Fix crash due to negative array index
- glsl: Use signed array index in update_max_array_access()

Brian Paul (1):

- mesa: fix UNCLAMPED_FLOAT_TO_UBYTE() macro for MSVC

Emil Velikov (2):

- docs: Add sha256 sums for the 10.3.2 release
- Update version to 10.3.3

Ilia Mirkin (27):

- freedreno/ir3: fix FSLT/etc handling to return 0/-1 instead of 0/1.0
- freedreno/ir3: INEG operates on src0, not src1
- freedreno/ir3: add UARL support
- freedreno/ir3: negate result of USLT/etc
- freedreno/ir3: use unsigned comparison for UIF
- freedreno/ir3: add TXL support
- freedreno/ir3: fix UCMP handling
- freedreno/ir3: implement UMUL correctly
- freedreno: add default .dir-locals.el for emacs settings
- freedreno/ir3: make texture instruction construction more dynamic
- freedreno/ir3: fix TXB/TXL to actually pull the bias/lod argument
- freedreno/ir3: add TXQ support
- freedreno/ir3: add TXB2 support
- freedreno: dual-source render targets are not supported
- freedreno: instanced drawing/compute not yet supported
- freedreno/ir3: avoid fan-in sources referring to same instruction
- freedreno/ir3: add IDIV/UDIV support
- freedreno/ir3: add UMOD support, based on UDIV
- freedreno/ir3: add MOD support
- freedreno/ir3: add ISSG support
- freedreno/ir3: add UMAD support
- freedreno/ir3: make TXQ return integers, not floats
- freedreno/ir3: shadow comes before array
- freedreno/ir3: add texture offset support
- freedreno/ir3: add TXD support and expose ARB_shader_texture_lod
- freedreno/ir3: add TXF support

- freedreno: positions come out as integers, not half-integers

Jan Vesely (1):

- configure: include llvm systemlibs when using static llvm

Marek Olšák (5):

- r600g: fix polygon mode for points and lines and point/line fill modes
- radeonsi: fix polygon mode for points and lines and point/line fill modes
- radeonsi: fix incorrect index buffer max size for lowered 8-bit indices
- Revert "st/mesa: set MaxUnrollIterations = 255"
- r300g: remove enabled/disabled hyperz and AA compression messages

Mauro Rossi (1):

- gallium/nouveau: fully build the driver under android

Michel Dänzer (1):

- radeon/llvm: Dynamically allocate branch/loop stack arrays

Rob Clark (62):

- freedreno/ir3: detect scheduler fail
- freedreno/ir3: add TXB
- freedreno/ir3: add DDX/DDY
- freedreno/ir3: bit of debug
- freedreno/ir3: fix error in bail logic
- freedreno/ir3: fix constlen with relative addressing
- freedreno/ir3: add no-copy-propagate fallback step
- freedreno: don't overflow cmdstream buffer so much
- freedreno/ir3: fix potential segfault in RA
- freedreno: update generated headers
- freedreno/a3xx: enable hw primitive-restart
- freedreno/a3xx: handle rendering to layer != 0
- freedreno: update generated headers
- freedreno/a3xx: format fixes
- util/u_format: add _is_alpha()
- freedreno/a3xx: alpha render-target shenanigans
- freedreno/ir3: catch incorrect usage of tmp-dst
- freedreno/ir3: add missing put_dst
- freedreno: "fix" problems with excessive flushes
- freedreno: update generated headers
- freedreno/a3xx: 3d/array textures
- freedreno: add DRM_CONF_SHARE_FD

- freedreno/a3xx: more texture array fixes
- freedreno/a3xx: initial texture border-color
- freedreno: fix compiler warning
- freedreno: don't advertise mirror-clamp support
- freedreno: update generated headers
- freedreno: we have more than 0 viewports!
- freedreno: turn missing caps into compile warnings
- freedreno/a3xx: add LOD_BIAS
- freedreno/a3xx: add flat interpolation mode
- freedreno/a3xx: add 32bit integer vtx formats
- freedreno/a3xx: fix border color order
- freedreno: move bind_sampler_states to per-generation
- freedreno: add texcoord clamp support to lowering
- freedreno/a3xx: add support to emulate GL_CLAMP
- freedreno/a3xx: re-emit shaders on variant change
- freedreno/lowering: fix token calculation for lowering
- freedreno: destroy transfer pool after blitter
- freedreno: max-texture-lod-bias should be 15.0f
- freedreno: update generated headers
- freedreno/a3xx: handle large shader program sizes
- freedreno/a3xx: emit all immediates in one shot
- freedreno/ir3: fix lockups with lame FRAG shaders
- freedreno/a3xx: handle VS only outputting BCOLOR
- freedreno: query fixes
- freedreno/a3xx: refactor vertex state emit
- freedreno/a3xx: refactor/optimize emit
- freedreno/ir3: optimize shader key comparision
- freedreno: inline fd_draw_emit()
- freedreno: fix layer_stride
- freedreno: update generated headers
- freedreno/ir3: large const support
- freedreno/a3xx: more layer/level fixes
- freedreno/ir3: comment + better fxn name
- freedreno/ir3: fix potential gpu lockup with kill
- freedreno/a3xx: disable early-z when we have kill's
- freedreno/ir3: add debug flag to disable cp

- freedreno: clear vs scissor
- freedreno: mark scissor state dirty when enable bit changes
- freedreno/a3xx: fix viewport state during clear
- freedreno/a3xx: fix depth/stencil restore format

Tapani Pälli (2):

- glsl: fix uniform location count used for glsl types
- mesa: check that uniform exists in glUniform* functions

4.191 Mesa 10.3.2 Release Notes / October 24, 2014

Mesa 10.3.2 is a bug fix release which fixes bugs found since the 10.3 release.

Mesa 10.3.2 implements the OpenGL 3.3 API, but the version reported by glGetString(GL_VERSION) or glGetIntegerv(GL_MAJOR_VERSION) / glGetIntegerv(GL_MINOR_VERSION) depends on the particular driver being used. Some drivers don't support all the features required in OpenGL 3.3. OpenGL 3.3 is **only** available if requested at context creation because compatibility contexts are not supported.

4.191.1 SHA256 checksums

```
e65f8e691f06f111c1aeb3a376b13c9cc88cb162bee2709e0e7e6b0e6628ca75  MesaLib-10.3.2.tar.
↪gz
e9849bcb9aa9acd98a753d6d46d2e7d7238d3367036e11357a60efd16de8bea3  MesaLib-10.3.2.tar.
↪bz2
427dc0d670d38e713ebff2675665ec2fe4ff7d04ce227bd54de946999fc1d234  MesaLib-10.3.2.zip
```

4.191.2 New features

None

4.191.3 Bug fixes

This list is likely incomplete.

- Bug 54372 - GLX_INTEL_swap_event crashes driver when swapping window buffers
- Bug 81680 - [r600g] Firefox crashes with hardware acceleration turned on
- Bug 84140 - mplayer crashes playing some files using vdpau output
- Bug 84662 - Long pauses with Unreal demo Elemental on R9270X since : Always flush the HDP cache before submitting a CS to the GPU
- Bug 85267 - vlc crashes with vdpau (Radeon 3850HD) [r600]

4.191.4 Changes

Brian Paul (3):

- mesa: fix spurious wglGetProcAddress / GL_INVALID_OPERATION error
- st/wgl: add WINAPI qualifiers on wgl function typedefs
- glsl: fix several use-after-free bugs

Daniel Manjarres (1):

- glx: Fix glxUseXFont for glxWindow and glxPixmaps

Dave Airlie (1):

- mesa: fix GetTexImage for 1D array depth textures

Emil Velikov (2):

- docs: Add sha256 sums for the 10.3.1 release
- Update VERSION to 10.3.2

Ilia Mirkin (4):

- gm107/ir: add dnz emission for fmul
- gk110/ir: add dnz flag emission for fmul/fmad
- nouveau: 3d textures are unsupported, limit 3d levels to 1
- st/gbm: fix order of arguments passed to is_format_supported

Kenneth Graunke (3):

- i965: Add a BRW_MOCS_PTE #define.
- i965: Use BDW_MOCS_PTE for renderbuffers.
- i965: Fix register write checks.

Marek Olšák (2):

- st/mesa: use pipe_sampler_view_release for releasing sampler views
- glsl_to_tgsi: fix the value of gl_FrontFacing with native integers

Michel Dänzer (4):

- radeonsi: Clear sampler view flags when binding a buffer
- r600g,radeonsi: Always use GTT again for PIPE_USAGE_STREAM buffers
- winsys/radeon: Use separate caching buffer manager for each set of flags
- r600g: Drop references to destroyed blend state

4.192 Mesa 10.3.1 Release Notes / October 12, 2014

Mesa 10.3.1 is a bug fix release which fixes bugs found since the 10.3 release.

Mesa 10.3.1 implements the OpenGL 3.3 API, but the version reported by glGetString(GL_VERSION) or glGetIntegerv(GL_MAJOR_VERSION) / glGetIntegerv(GL_MINOR_VERSION) depends on the particular driver being used. Some drivers don't support all the features required in OpenGL 3.3. OpenGL 3.3 is **only** available if requested at context creation because compatibility contexts are not supported.

4.192.1 SHA256 checksums

```
155afcbad17be8bb80282c761b957d5cc716c14a1fa16c4f5ee04e76df729c6d  MesaLib-10.3.1.tar.
↪gz
b081d077d717e5d56f2d59677490856052c41573e50378ff86d6c72456714add  MesaLib-10.3.1.tar.
↪bz2
07a14febfed06412d519e091a62d24513fee6745f1a6f8a8f1956bfe04b77d15  MesaLib-10.3.1.zip
```

4.192.2 New features

None

4.192.3 Bug fixes

This list is likely incomplete.

- Bug 79462 - [NVC0/Codegen] Shader compilation falis in spill logic
- Bug 82932 - [SNB+ Bisected]Ogles3conform ES3-CTS.shaders.indexing.vector_subscript.vec3_static_loop_subscript_write_dire fails
- Bug 83506 - [UBO] row_major layout ignored inside structures
- Bug 83533 - [UBO] nested structures don't get appropriate padding
- Bug 83570 - Glyphy demo throws unhandled Integer division by zero exception
- Bug 83741 - [UBO] row_major layout partially ignored for arrays of structures
- Bug 84178 - Big glamor regression in Xorg server 1.6.99.1 GIT: x11perf 1.5 Test: PutImage XY 500x500 Square

4.192.4 Changes

Andreas Pokorny (2):

- egl/drm: expose KHR_image_pixmap extension
- i915: Fix black buffers when importing prime fds

Brian Paul (1):

- mesa: fix prog_optimize.c assertions triggered by SWZ opcode

Emil Velikov (2):

- docs: Add 10.3 sha256 sums, news item and link release notes
- Update VERSION to 10.3.1

Ian Romanick (4):

- glsl: Make sure fields after small structs have correct padding
- glsl: Make sure row-major array-of-structure get correct layout
- glsl: Round struct size up to at least 16 bytes
- glsl: Strip arrayness from ir_type_dereference_variable too

Ilia Mirkin (5):

- nv50/ir: avoid deleting pseudo instructions too early
- gm107/ir: fix manual TXD for array targets
- gm107/ir: fix texture argument order
- gm107/ir: add support for indirect const buffer selection
- gm107/ir: take relative pfetch offset into account

Keith Packard (1):

- glx/dri3: Provide error diagnostics when DRI3 allocation fails

Kenneth Graunke (2):

- mesa: Use proper structure for glGet*(GL_TEXTURE_COORD_ARRAY*).
- mesa: Set correct array element in vbo_exec_vtx_init.

Marek Olšák (3):

- radeonsi: release GS rings at context destruction
- radeonsi: properly destroy the GS copy shader and scratch_bo for compute
- st/dri: remove GALLIUM_MSAA and __GL_FSAA_MODE environment variables

Michel Dänzer (1):

- st/mesa: Use PIPE_USAGE_STAGING for GL_STATIC/DYNAMIC/STREAM_READ buffers

Richard Sandiford (2):

- mesa: Fix alpha component in unpack_R8G8B8X8_SRGB.
- swrast: Fix handling of MESA_FORMAT_L8A8_SRGB for big-endian

Roland Scheidegger (1):

- gallivm: fix idiv

Thomas Hellstrom (1):

- st/xa: Fix regression in xa_yuv_planar_blit()

Tom Stellard (2):

- clover: Add support to mem objects for multiple destructor callbacks v2
- configure.ac: Compute LLVM_VERSION_PATCH using llvm-config

Tomasz Figa (3):

- util: Include in Android builds
- st/mesa: Generate format_info.c in Android builds
- st/mesa: Fix paths used in Android builds

rconde (1):

- gallivm,tgsi: fix idiv by zero crash

4.193 Mesa 10.2.9 Release Notes / October 12, 2014

Mesa 10.2.9 is a bug fix release which fixes bugs found since the 10.2.8 release. This is the final planned release for the 10.2 branch.

Mesa 10.2.9 implements the OpenGL 3.3 API, but the version reported by glGetString(GL_VERSION) or glGetIntegerv(GL_MAJOR_VERSION) / glGetIntegerv(GL_MINOR_VERSION) depends on the particular driver being used. Some drivers don't support all the features required in OpenGL 3.3. OpenGL 3.3 is **only** available if requested at context creation because compatibility contexts are not supported.

4.193.1 SHA256 checksums

```
f8d62857eed8f604a57710c58a8ffcfb8dab2dc4977ec27c956c7c4fd14032f6  MesaLib-10.2.9.tar.
↪gz
f6031f8b7113a92325b60635c504c510490eebb2e707119bbff7bd86aa34657d  MesaLib-10.2.9.tar.
↪bz2
11c0ef4f3308fc29d9f15a77fd8f4842a946fce9e830250a1c95b171a446171a  MesaLib-10.2.9.zip
```

4.193.2 New features

None

4.193.3 Bug fixes

This list is likely incomplete.

- Bug 79462 - [NVC0/Codegen] Shader compilation falis in spill logic
- Bug 83570 - Glyphy demo throws unhandled Integer division by zero exception

4.193.4 Changes

Andreas Pokorny (2):

- egl/drm: expose KHR_image_pixmap extension
- i915: Fix black buffers when importing prime fds

Emil Velikov (2):

- docs: Add sha256 sums for the 10.2.8 release
- Update VERSION to 10.2.9

Ilia Mirkin (1):

- nv50/ir: avoid deleting pseudo instructions too early

Marek Olšák (3):

- radeonsi: release GS rings at context destruction
- radeonsi: properly destroy the GS copy shader and scratch_bo for compute
- st/dri: remove GALLIUM_MSAA and __GL_FSAA_MODE environment variables

Roland Scheidegger (1):

- gallivm: fix idiv

Thomas Hellstrom (1):

- st/xa: Fix regression in xa_yuv_planar_blit()

Tom Stellard (1):

- configure.ac: Compute LLVM_VERSION_PATCH using llvm-config

rconde (1):

- gallivm,tgsi: fix idiv by zero crash

4.194 Mesa 10.3 Release Notes / September 19, 2014

Mesa 10.3 is a new development release. People who are concerned with stability and reliability should stick with a previous release or wait for Mesa 10.3.1.

Mesa 10.3 implements the OpenGL 3.3 API, but the version reported by glGetString(GL_VERSION) or glGetIntegerv(GL_MAJOR_VERSION) / glGetIntegerv(GL_MINOR_VERSION) depends on the particular driver being used. Some drivers don't support all the features required in OpenGL 3.3. OpenGL 3.3 is **only** available if requested at context creation because compatibility contexts are not supported.

4.194.1 SHA256 checksums

```
9a1bf52040fc3dda81e83a35f944f1c3f532847dbe9fdf57161265cf71ea1bae  MesaLib-10.3.0.tar.
↪gz
0283bfe710fa449ed82e465cfa09612a269e19abb7e0382082608062ce7960b5  MesaLib-10.3.0.tar.
↪bz2
221420763c2c3a244836a736e735612c4a6a0377b4e5223fca1e612f49906789  MesaLib-10.3.0.zip
```

4.194.2 New features

Note: some of the new features are only available with certain drivers.

- GL_ARB_ES3_compatibility on nv50, nvc0, r600, radeonsi, softpipe, llvmpipe
- GL_ARB_clear_texture on i965
- GL_ARB_compressed_texture_pixel_storage on all drivers
- GL_ARB_conditional_render_inverted on i965, nvc0, softpipe, llvmpipe
- GL_ARB_derivative_control on i965, nv50, nvc0, r600
- GL_ARB_draw_indirect on nvc0, radeonsi
- GL_ARB_explicit_uniform_location (all drivers that support GLSL)
- GL_ARB_fragment_layer_viewport on nv50, nvc0, llvmpipe, r600
- GL_ARB_gpu_shader5 on i965/gen7, nvc0
- GL_ARB_multi_draw_indirect on nvc0, radeonsi
- GL_ARB_sample_shading on radeonsi
- GL_ARB_seamless_cubemap_per_texture on i965, llvmpipe, nvc0, r600, radeonsi, softpipe

- GL_ARB_stencil_texturing on nv50, nvc0, r600, and radeonsi
- GL_ARB_texture_barrier on nv50, nvc0, r300, r600, radeonsi
- GL_ARB_texture_compression_bptc on i965/gen7+, nvc0, r600/evergreen+, radeonsi
- GL_ARB_texture_cube_map_array on radeonsi
- GL_ARB_texture_gather on r600, radeonsi
- GL_ARB_texture_query_levels on nv50, nvc0, llvmpipe, r600, radeonsi, softpipe
- GL_ARB_texture_query_lod on r600, radeonsi
- GL_ARB_viewport_array on nvc0
- GL_AMD_vertex_shader_viewport_index on i965/gen7+, r600
- GL_OES_compressed_ETC1_RGB8_texture on nv30, nv50, nvc0, r300, r600, radeonsi, softpipe, llvmpipe
- GLX_MESA_query_renderer on nv30, nv50, nvc0, r300, r600, radeonsi, softpipe, llvmpipe
- A new software rasterizer driver (kms_swrast_dri.so) that works with DRM drivers that don't have a full-fledged GEM (such as qxl or simpledrm)
- Distribute the Khronos GL/glcorearb.h header file.

4.194.3 Bug fixes

- Bug 50754 - Building 32 bit mesa on 64 bit OS fails since change for automake
- Bug 53617 - [llvmpipe] piglit fbo-depthtex regression
- Bug 54372 - GLX_INTEL_swap_event crashes driver when swapping window buffers
- Bug 56127 - [ILK bisected]unigine-sanctruary performance reduced by 98%
- Bug 66184 - src/mesa/state_tracker/st_glsl_to_tgsi.cpp:3216:simplify_cmp: Assertion 'inst->dst.index < 4096' failed.
- Bug 66452 - JUNIPER UVD accelerated playback of WMV3 streams does not work
- Bug 68365 - [SNB Bisected]Piglit spec_ARB_framebuffer_object_fbo-blit-stretch fail
- Bug 70441 - [Gen4-5 clip] Piglit spec_OpenGL_1.1_polygon-offset hits (execsize >= width) assertion
- Bug 73846 - [llvmpipe] lp_test_format fails with llvm-3.5svn >= r199602
- Bug 74005 - [i965 Bisected]Piglit/glx_glx-make-glxdrawable-current fails
- Bug 74863 - [r600g] HyperZ broken on RV770 and CYPRESS (Left 4 Dead 2 trees corruption) bisected!
- Bug 75010 - clang: error: unknown argument: '-fstack-protector-strong'
- Bug 75478 - [BDW]Some Piglit and Ogles2conform cases cause GPU hang
- Bug 75664 - Unigine Valley & Heaven "error: syntax error, unexpected EXTENSION, expecting $end" IVB HD4000
- Bug 75878 - [BDW] GPU hang running Raytracer WebGL demo
- Bug 76188 - EGL_EXT_image_dma_buf_import fd ownership is incorrect
- Bug 76223 - [radeonsi] luxmark segfault
- Bug 76939 - [BDW] GPU hang when running "Metro:Last Light " /"Crusader Kings II"
- Bug 77245 - Bogus GL_ARB_explicit_attrib_location layout identifier warnings

- Bug 77493 - lp_test_arit fails with llvm >= llvm-3.5svn r206094
- Bug 77703 - [ILK Bisected]Piglit glean_texCombine4 fails
- Bug 77704 - [IVB/HSW Bisected]Ogles3conform GL3Tests_shadow_shadow_execution_frag.test fails
- Bug 77705 - [SNB/IVB/HSW/BYT/BDW Bisected]Ogles3conform GL3Tests/packed_pixels/packed_pixels_pixelstore.test segfault
- Bug 77707 - [ILK Bisected]Ogles2conform GL_sin_sin_float_frag_xvary.test fails
- Bug 77740 - i965: Relax accumulator dependency scheduling on Gen < 6
- Bug 77852 - [BDW]Piglit spec_ARB_framebuffer_object_fbo-drawbuffers-none_glBlitFramebuffer fails
- Bug 77856 - [BDW]Piglit spec_OpenGL_3.0_clearbuffer-mixed-format fails
- Bug 77865 - [BDW] Many Ogles3conform framebuffer_blit cases fail
- Bug 78225 - Compile error due to undefined reference to 'gbm_dri_backend', fix attached
- Bug 78258 - make check link_varyings.gl_ClipDistance failure
- Bug 78403 - query_renderer_implementation_unittest.cpp:144:4: error: expected primary-expression before '.' token
- Bug 78468 - Compiling of shader gets stuck in infinite loop
- Bug 78537 - no anisotropic filtering in a native Half-Life 2
- Bug 78546 - [swrast] piglit copyteximage-border regression
- Bug 78581 - OpenCL: clBuildProgram prints error messages directly rather than storing them
- Bug 78648 - Texture artifacts in Kerbal Space Program
- Bug 78665 - macros in builtin_functions.cpp make invalid assumptions about M_PI definitions
- Bug 78679 - Gen4-5 code lost: runtime_check_aads_emit
- Bug 78691 - [G45 - Tesseract] Mesa 10.1.2 implementation error: Unsupported opcode 169872468 in FS
- Bug 78692 - Football Manager 2014, gameplay rendered black & white
- Bug 78716 - Fix Mesa bugs for running Unreal Engine 4.1 Cave effects demo compiled for Linux
- Bug 78803 - gallivm/lp_bld_debug.cpp:42:28: fatal error: llvm/IR/Module.h: No such file or directory
- Bug 78842 - [swrast] piglit fcc-read-after-clear copy rb regression
- Bug 78843 - [swrast] piglit copyteximage 1D regression
- Bug 78872 - [ILK Bisected]Piglit spec_ARB_depth_buffer_float_fbo-depthstencil-GL_DEPTH32F_STENCIL8-blit Aborted
- Bug 78875 - [ILK Bisected]Webglc conformance/uniforms/uniform-default-values.html fails
- Bug 78888 - test_eu_compact.c:54:3: error: implicit declaration of function 'brw_disasm' [-Werror=implicit-function-declaration]
- Bug 79029 - INTEL_DEBUG=shader_time is full of lies
- Bug 79095 - x86/common_x86.c:348:14: error: use of undeclared identifier 'bit_SSE4_1'
- Bug 79115 - glFramebufferRenderbuffer(GL_DRAW_FRAMEBUFFER, GL_DEPTH_STENCIL_ATTACHMENT, GL_RENDERBUFFER, 0) doesn't unbind stencil buffer
- Bug 79263 - Linking error in egl_gallium.la when compiling 32 bit on multiarch
- Bug 79294 - Xlib-based build broken on non x86/x86-64 architectures

- Bug 79373 - Non-const initializers for matrix and vector constructors
- Bug 79382 - build error: multiple definition of 'loader_get_pci_id_for_fd'
- Bug 79421 - [llvmpipe] SIGSEGV src/gallium/drivers/llvmpipe/lp_rast_priv.h:218
- Bug 79440 - prog_hash_table.c:146: undefined reference to '_mesa_error_no_memory'
- Bug 79469 - Commit e3cc0d90e14e62a0a787b6c07a6df0f5c84039be breaks unigine heaven
- Bug 79534 - gen<7 renders garbage
- Bug 79616 - L4D2 crash on startup
- Bug 79724 - switch statement type check
- Bug 79729 - [i965] glClear on a multisample texture doesn't work
- Bug 79809 - radeonsi: mouse cursor corruption using weston on AMD Kaveri
- Bug 79823 - [NV30/gallium] Mozilla apps freeze on startup with nouveau-dri-10.2.1 libs on dual-screen
- Bug 79885 - commit b52a530 (gallium/egl: st_profiles are build time decision, treat them as such) broke egl
- Bug 79903 - [HSW Bisected]Some Piglit and Ogles2conform cases fail
- Bug 79907 - Mesa 10.2.1 –enable-vdpau default=auto broken
- Bug 79948 - [i965] Incorrect pixels when using discard and uniform loads
- Bug 80015 - Transparency glitches in native Civilization 5 (Civ5) port
- Bug 80115 - MESA_META_DRAW_BUFFERS induced GL_INVALID_VALUE errors
- Bug 80211 - [ILK/SNB Bisected]Piglit shaders_glsl-fs-copy-propagation-texcoords-1 fails
- Bug 80247 - Khronos conformance test ES3-CTS.gtf.GL3Tests.transform_feedback.transform_feedback_vertex_id fails
- Bug 80254 - pipe_loader_sw.c:90: undefined reference to 'dri_create_sw_winsys'
- Bug 80541 - [softpipe] piglit levelclamp regression
- Bug 80561 - Incorrect implementation of some VDPAU APIs.
- Bug 80614 - [regression] Error in 'omxregister-bellagio': munmap_chunk(): invalid pointer: 0x00007f5f76626dab
- Bug 80778 - [bisected regression] piglit spec/glsl-1.50/compiler/incorrect-in-layout-qualifier-repeated-prim.geom
- Bug 80827 - [radeonsi,R9 270X] Corruptions in window menus in KDE
- Bug 80880 - Unreal Engine 4 demos fail GLSL compiler assertion
- Bug 80991 - [BDW]Piglit spec_ARB_sample_shading_builtin-gl-sample-mask_2 fails
- Bug 81020 - [radeonsi][regresssion] Wireframe of background rendered through objects in Half-Life 2: Episode 2 with MSAA enabled
- Bug 81150 - [SNB]Piglit spec_arb_shading_language_packing_execution_built-in-functions_fs-packSnorm4x8 fails
- Bug 81157 - [BDW]Piglit some spec_glsl-1.50_execution_built-in-functions* cases fail
- Bug 81450 - [BDW]Piglit spec_glsl-1.30_execution_tex-miplevel-selection_textureGrad_1DArray cases intel_do_flush_locked failed
- Bug 81828 - [BDW Bisected]Ogles3conform GL3Tests_packed_pixels_packed_pixels_pbo.test fails

- Bug 81834 - TGSI constant buffer overrun causes assertion failure
- Bug 81857 - [SNB+]Piglit spec_glsl-1.30_execution_switch_fs-default_last sporadically fail
- Bug 81967 - [regression] Selections in Blender renders wrong
- Bug 82139 - [r600g, bisected] multiple ubo piglit regressions
- Bug 82159 - No rule to make target '../../../../src/mesa/libmesa.la', needed by 'collision'.
- Bug 82255 - [VP2] Chroma planes are vertically stretched during VDPAU playback
- Bug 82268 - Add support for the OpenRISC architecture (or1k)
- Bug 82428 - [radeonsi,R9 270X] System lockup when using mplayer/mpv with VDPAU
- Bug 82472 - piglit 16385-consecutive-chars regression
- Bug 82483 - format_srgb.h:145: undefined reference to 'util_format_srgb_to_linear_8unorm_table'
- Bug 82517 - [RADEONSI,VDPAU] SIGSEGV in map_msg_fb_buf called from ruvd_destroy, when closing a Tab with accelerated video player
- Bug 82534 - srceglmaineglapi.h : fatal error LNK1107: invalid or corrupt file: cannot read at 0x2E02
- Bug 82536 - u_current.h:72: undefined reference to '__imp__glapi_Dispatch'
- Bug 82538 - Super Maryo Chronicles fails with st/mesa assertion failure
- Bug 82539 - vmw_screen_dri.lo In file included from vmw_screen_dri.c:41: vmwgfx_drm.h:32:17: error: drm.h: No such file or directory
- Bug 82546 - [regression] libOSMesa build failure
- Bug 82574 - GLSL: opt_vectorize goes wrong on texture lookups
- Bug 82628 - bisected: GALLIUM_HUD hangs radeon 7970M (PRIME)
- Bug 82671 - [r600g-evergreen][compute]Empty kernel execution causes crash
- Bug 82709 - OpenCL not working on radeon hainan
- Bug 82796 - [IVB/BYT-M/HSW/BDW Bisected]Synmark2_v6.0_OglTerrainFlyInst/OglTerrainPanInst cannot run as image validation failed
- Bug 82804 - unreal engine 4 rendering errors
- Bug 82814 - glDrawBuffers(0, NULL) segfaults in _mesa_drawbuffers
- Bug 82828 - Regression: Crash in 3Dmark2001
- Bug 82846 - [BDW Bisected] Gpu hang when running Lightsmark v2008/Warsow v1.0/Xonotic v0.7/unigine-demos
- Bug 82881 - test_vec4_register_coalesce regression
- Bug 82882 - [swrast] piglit glsl-fs-uniform-bool-1 regression
- Bug 82929 - [BDW Bisected]glxgears causes X hang
- Bug 82932 - [SNB+ Bisected]Ogles3conform ES3-CTS.shaders.indexing.vector_subscript.vec3_static_loop_subscript_write_dire fails
- Bug 83046 - [BDW bisected]] Warsow v1.0/Xonotic v0.7/Gputest v0.5_triangle_fullscreen/synmark2_v6/GLBenchmark v2.5.0/GLBenchmark v2.7.0/Ungine-demos performance reduced 30%~60%
- Bug 83079 - [NVC0] Dota 2 (Linux native and Wine) crash with Nouveau Drivers
- Bug 83081 - [BDW Bisected]Piglit spec_ARB_sample_shading_builtin-gl-sample-mask_2 is core dumped

- Bug 83127 - [ILK Bisected]Piglit glean_texCombine fails
- Bug 83355 - FTBFS: src/mesa/program/program_lexer.l:122:64: error: unknown type name 'YYSTYPE'
- Bug 83432 - r600_query.c:269:r600_emit_query_end: Assertion 'ctx->num_pipelinestat_queries > 0' failed [Gallium HUD]
- Bug 83468 - [UBO] Using bool from UBO as if-statement condition asserts

4.194.4 Changes

- Removed support for the GL_ATI_envmap_bumpmap extension
- The hacky –enable-32/64-bit is no longer available in configure. To build 32/64 bit mesa refer to the default method recommended by your distribution
- The environment variable GALLIUM_MSAA that forced a multisample GLX visual was removed.

4.195 Mesa 10.2.8 Release Notes / September 19, 2014

Mesa 10.2.8 is a bug fix release which fixes bugs found since the 10.2.7 release.

Mesa 10.2.8 implements the OpenGL 3.3 API, but the version reported by glGetString(GL_VERSION) or glGetIntegerv(GL_MAJOR_VERSION) / glGetIntegerv(GL_MINOR_VERSION) depends on the particular driver being used. Some drivers don't support all the features required in OpenGL 3.3. OpenGL 3.3 is **only** available if requested at context creation because compatibility contexts are not supported.

4.195.1 SHA256 checksums

```
4c5a25ccaf1a9734bbd10d62a1420cc8fd35a1060ce679f2fc846769a25fbeec  MesaLib-10.2.8.tar.
↪gz
1ef9ad3f241788d454f2ff8c9d65b6849dfc31c8fe91f70fd2930b81c8af1398  MesaLib-10.2.8.tar.
↪bz2
d26218da3b44734b1d555267b4c63c48803c4c8b14d2bc53071be57014da37fa  MesaLib-10.2.8.zip
```

4.195.2 New features

None

4.195.3 Bug fixes

This list is likely incomplete.

- Bug 77493 - lp_test_arit fails with llvm >= llvm-3.5svn r206094
- Bug 82539 - vmw_screen_dri.lo In file included from vmw_screen_dri.c:41: vmwgfx_drm.h:32:17: error: drm.h: No such file or directory
- Bug 82882 - [swrast] piglit glsl-fs-uniform-bool-1 regression
- Bug 83432 - r600_query.c:269:r600_emit_query_end: Assertion 'ctx->num_pipelinestat_queries > 0' failed [Gallium HUD]
- Bug 83567 - Mesa 10.2.6 does not compile with llvm 3.5

- Bug 83735 - [mesa-10.2.x] broken with llvm-3.5 and old CPUs

4.195.4 Changes

Aaron Watry (1):

- gallivm: Fix build after LLVM commit 211259

Christoph Bumiller (2):

- nv50/ir/util: fix BitSet issues
- nvc0/ir: clarify recursion fix to finding first tex uses

Emil Velikov (3):

- docs: Add sha256 sums for the 10.2.7 release
- configure: bail out if building svga without libdrm
- Update VERSION to 10.2.8

Ilia Mirkin (4):

- nv50/ir: avoid array overrun when checking for supported mods
- nouveau: only enable the depth test if there actually is a depth buffer
- nouveau: only enable stencil func if the visual has stencil bits
- nouveau: change internal variables to avoid conflicts with macro args

Jonathan Gray (1):

- configure.ac: strip _GNU_SOURCE from llvm-config output

José Fonseca (1):

- gallivm: Disable workaround for PR12833 on LLVM 3.2+.

Maarten Lankhorst (4):

- nouveau: re-allocate bo's on overflow
- nouveau: fix MPEG4 hw decoding
- nouveau: rework reference frame handling
- nouveau: remove unneeded assert

Marek Olšák (3):

- r600g,radeonsi: make sure there's enough CS space before resuming queries
- mesa: set UniformBooleanTrue = 1.0f by default
- st/mesa: use 1.0f as boolean true on drivers without integer support

Richard Sandiford (1):

- gallivm: Fix uses of 2^24

Roland Scheidegger (1):

- gallivm: set mcpu when initializing llvm execution engine

Thomas Hellstrom (1):

- winsys/svga: Fix incorrect type usage in IOCTL v2

4.196 Mesa 10.2.7 Release Notes / September 06, 2014

Mesa 10.2.7 is a bug fix release which fixes bugs found since the 10.2.6 release.

Mesa 10.2.7 implements the OpenGL 3.3 API, but the version reported by glGetString(GL_VERSION) or glGetIntegerv(GL_MAJOR_VERSION) / glGetIntegerv(GL_MINOR_VERSION) depends on the particular driver being used. Some drivers don't support all the features required in OpenGL 3.3. OpenGL 3.3 is **only** available if requested at context creation because compatibility contexts are not supported.

4.196.1 SHA256 checksums

```
cb67dfaabf88acba29aa2cf0dd58ee17b21ebf9594f8d1226c41794da8de3e9d  MesaLib-10.2.7.tar.
↪gz
27b958063a4c002071f14ed45c7d2a1ee52cd85e4ac8876e8a1c273495a7d43f  MesaLib-10.2.7.tar.
↪bz2
a2796a2d5bbbc2edd22857ecc267cba68dfe5d0296f5d84ba7510877b216cc40  MesaLib-10.2.7.zip
```

4.196.2 New features

None

4.196.3 Bug fixes

This list is likely incomplete.

- Bug 36193 - [i965] brw_eu_emit.c:182: validate_reg: Assertion 'execsize >= width' failed.
- Bug 66184 - src/mesa/state_tracker/st_glsl_to_tgsi.cpp:3216:simplify_cmp: Assertion 'inst->dst.index < 4096' failed.
- Bug 70441 - [Gen4-5 clip] Piglit spec_OpenGL_1.1_polygon-offset hits (execsize >= width) assertion
- Bug 76188 - EGL_EXT_image_dma_buf_import fd ownership is incorrect
- Bug 76789 - [radeonsi] si_descriptors.c requires -std=gnu99 or -fms-extensions
- Bug 82139 - [r600g, bisected] multiple ubo piglit regressions
- Bug 82255 - [VP2] Chroma planes are vertically stretched during VDPAU playback
- Bug 82671 - [r600g-evergreen][compute]Empty kernel execution causes crash
- Bug 82709 - OpenCL not working on radeon hainan
- Bug 82814 - glDrawBuffers(0, NULL) segfaults in _mesa_drawbuffers
- Bug 83079 - [NVC0] Dota 2 (Linux native and Wine) crash with Nouveau Drivers
- Bug 83355 - FTBFS: src/mesa/program/program_lexer.l:122:64: error: unknown type name 'YYSTYPE'

4.196.4 Changes

Adam Jackson (1):

- radeonsi: Don't use anonymous struct trick in atom tracking

Alex Deucher (2):

- radeonsi: add new CIK pci ids
- radeonsi: add new SI pci ids

Andreas Boll (1):

- winsys/radeon: fix nop packet padding for hawaii

Anuj Phogat (1):

- i965: Bail on vec4 copy propagation for scratch writes with source modifiers

Brian Paul (1):

- mesa: fix NULL pointer deref bug in _mesa_drawbuffers()

Carl Worth (2):

- docs: Add sha256 sums for the 10.2.6 release
- Makefile: Switch from md5sums to sha256sums

Dave Airlie (1):

- i965: add missing parens in vec4 visitor

Emil Velikov (17):

- configure.ac: bail out if building gallium_gbm without gallium_egl
- android: gallium/nouveau: fix include folders, link against libstlport
- android: egl/main: fixup the nouveau build
- automake: gallium/freedreno: drop spurious include dirs
- android: gallium/freedreno: add preliminary build
- android: egl/main: add/enable freedreno
- android: gallium/auxiliary: drop log2/log2f redefitions
- android: drop HAL_PIXEL_FORMAT_RGBA_{5551,4444}
- android: glsl: the stlport over the limited Android STL
- android: dri/i915: do not build an 'empty' driver
- cherry-ignore: remove patch that lacking previous dependencies
- cherry-ignore: PIPE_SHADER_CAP_MAX_CONST_BUFFER_SIZE is not it 10.2
- cherry-ignore: drop whitespace fix
- cherry-ignore: reject a15088338eb
- get-pick-list.sh: Require explicit "10.2" for nominating stable patches
- mesa: fix make tarballs
- Update VERSION to 10.2.7

Ian Romanick (1):

- mesa: Handle uninitialized textures like other textures in get_tex_level_parameter_image

Ilia Mirkin (9):

- nouveau: make sure to invalidate any vbo state as well
- nouveau: don't keep stale pointer to free'd data

- nvc0/ir: avoid infinite recursion when finding first uses of tex
- nv50: zero out unbound samplers
- nvc0: don't make 1d staging textures linear
- nv50/ir: avoid creating instructions that can't be emitted
- nv50: set the miptree address when clearing bo's in vp2 init
- nv50: mt address may not be the underlying bo's start address
- nv50: attach the buffer bo to the miptree structures

Jan Vesely (1):

- gallivm: Fix build with latest LLVM

José Fonseca (1):

- mesa: Move declaration to top of block.

Kenneth Graunke (3):

- i965/vec4: Set NoMask for GS_OPCODE_SET_VERTEX_COUNT on Gen8+.
- i965/vec4: Respect ir->force_writemask_all in Gen8 code generation.
- i965/clip: Fix brw_clip_unfilled.c/compute_offset's assembly.

Marek Olšák (3):

- r600g: fix constant buffer fetches
- radeonsi: save scissor state and sample mask for u_blitter
- glsl_to_tgsi: allocate and enlarge arrays for temporaries on demand

Paulo Sergio Travaglia (2):

- android: gallium/radeon: attempt to fix the android build
- android: egl/main: resolve radeon linking issues

Pekka Paalanen (1):

- egl_dri2: fix EXT_image_dma_buf_import fds

Robert Bragg (1):

- meta: save and restore swizzle for _GenerateMipmap

Tom Stellard (7):

- radeon/compute: Fix reported values for MAX_GLOBAL_SIZE and MAX_MEM_ALLOC_SIZE
- radeonsi/compute: Update reference counts for buffers in si_set_global_binding()
- radeonsi/compute: Call si_pm4_free_state() after emitting compute state
- clover: Flush the command queue in clReleaseCommandQueue()
- radeon: Add work-around for missing Hainan support in clang < 3.6 v2
- pipe-loader: Fix memory leak v2
- r600g/compute: Don't initialize vertex_buffer_state masks to 0x2

Vinson Lee (1):

- gallivm: Fix build with LLVM >= 3.6 r215967.

4.197 Mesa 10.2.6 Release Notes / August 19, 2014

Mesa 10.2.6 is a bug fix release which fixes bugs found since the 10.2.5 release.

Mesa 10.2.6 implements the OpenGL 3.3 API, but the version reported by glGetString(GL_VERSION) or glGetIntegerv(GL_MAJOR_VERSION) / glGetIntegerv(GL_MINOR_VERSION) depends on the particular driver being used. Some drivers don't support all the features required in OpenGL 3.3. OpenGL 3.3 is **only** available if requested at context creation because compatibility contexts are not supported.

4.197.1 SHA256 checksums

```
193314d2adba98e43697d726739ac46b4299aae324fa1821aa226890c28ac806  MesaLib-10.2.6.tar.
↪bz2
f7a45a5977b485eb95ac024205c584a0c112fe3951c2313c797579bb16a7a448  MesaLib-10.2.6.tar.
↪gz
6d086d6fcda8f317adfaaae40011decf2f2e2dc80819c4a7a77c76f73512e8d8  MesaLib-10.2.6.zip
```

4.197.2 New features

None

4.197.3 Bug fixes

This list is likely incomplete.

- Bug 81450 - [BDW]Piglit spec_glsl-1.30_execution_tex-miplevel-selection_textureGrad_1DArray cases intel_do_flush_locked failed

4.197.4 Changes

Anuj Phogat (15):

- mesa: Fix error condition for valid texture targets in glTexStorage* functions
- mesa: Turn target_can_be_compressed() in to a utility function
- mesa: Add error condition for using compressed internalformat in glTexStorage3D()
- mesa: Fix condition for using compressed internalformat in glCompressedTexImage3D()
- mesa: Add utility function _mesa_is_enum_format_snorm()
- mesa: Don't allow snorm internal formats in glCopyTexImage*() in GLES3
- mesa: Add a helper function _mesa_is_enum_format_unsized()
- mesa: Add a gles3 error condition for sized internalformat in glCopyTexImage*()
- mesa: Add gles3 error condition for GL_RGBA10_A2 buffer format in glCopyTexImage*()
- mesa: Add utility function _mesa_is_enum_format_unorm()
- mesa: Add gles3 condition for normalized internal formats in glCopyTexImage*()
- mesa: Allow GL_TEXTURE_CUBE_MAP target with compressed internal formats
- meta: Use _mesa_get_format_bits() to get the GL_RED_BITS

- egl: Fix OpenGL ES version checks in _eglParseContextAttribList()
- meta: Fix datatype computation in get_temp_image_type()

Brian Paul (1):

- mesa: fix assertion in _mesa_drawbuffers()

Carl Worth (2):

- docs: Add sha256 sums to the 10.2.5 release notes
- Update VERSION to 10.2.6

Ilia Mirkin (1):

- mesa/st: only convert AND(a, NOT(b)) into MAD when not using native integers

Jordan Justen (1):

- i965/miptree: Layout 1D Array as 2D Array with height of 1

Maarten Lankhorst (1):

- configure.ac: Do not require llvm on x32

Marek Olšák (4):

- st/mesa: fix blit-based partial TexSubImage for 1D arrays
- radeon,r200: fix buffer validation after CS flush
- radeonsi: fix a hang with instancing in Unigine Heaven/Valley on Hawaii
- radeonsi: fix CMASK and HTILE allocation on Tahiti

Pali Rohár (1):

- configure: check for dladdr via AC_CHECK_FUNC/AC_CHECK_LIB

Roland Scheidegger (1):

- gallivm: fix up out-of-bounds level when using conformant out-of-bound behavior

4.198 Mesa 10.2.5 Release Notes / August 2, 2014

Mesa 10.2.5 is a bug fix release which fixes bugs found since the 10.2.4 release.

Mesa 10.2.5 implements the OpenGL 3.3 API, but the version reported by glGetString(GL_VERSION) or glGetIntegerv(GL_MAJOR_VERSION) / glGetIntegerv(GL_MINOR_VERSION) depends on the particular driver being used. Some drivers don't support all the features required in OpenGL 3.3. OpenGL 3.3 is **only** available if requested at context creation because compatibility contexts are not supported.

4.198.1 SHA256 checksums

```
b4459f0bf7f4a3c8fb78ece3c9d2eac3d0e5bf38cb470f2a72705e744bd0310d  MesaLib-10.2.5.tar.
↪bz2
7b4dd0cb683f8c7dc48a3e7a315742bed58ddcd7b756c462aca4177bd1acdc79  MesaLib-10.2.5.tar.
↪gz
6180565914fb238dd77ccdaff96b6155d9a6e1b3e981ebbf6a6851301b384fed  MesaLib-10.2.5.zip
```

4.198.2 New features

None

4.198.3 Bug fixes

This list is likely incomplete.

- Bug 80991 - [BDW]Piglit spec_ARB_sample_shading_builtin-gl-sample-mask_2 fails

4.198.4 Changes

Abdiel Janulgue (3):

- i965/fs: Refactor check for potential copy propagated instructions.
- i965/fs: skip copy-propate for logical instructions with negated src entries
- i965/vec4: skip copy-propate for logical instructions with negated src entries

Adel Gadllah (1):

- i915: Fix up intelInitScreen2 for DRI3

Anuj Phogat (2):

- i965: Fix z_offset computation in intel_miptree_unmap_depthstencil()
- mesa: Don't use memcpy() in _mesa_texstore() for float depth texture data

Brian Paul (3):

- mesa: fix geometry shader memory leaks
- st/mesa: fix geometry shader memory leak
- gallium/u_blitter: fix some shader memory leaks

Carl Worth (6):

- docs: Add sha256 checksums for the 10.2.3 release
- Update VERSION to 10.2.4
- Add release notes for 10.2.4
- docs: Add SHA256 checksums for the 10.2.4 release
- cherry-ignore: Ignore a few patches picked in the previous stable release
- Update version to 10.2.5

Christian König (1):

- radeonsi: fix order of r600_need_dma_space and r600_context_bo_reloc

Eric Anholt (1):

- i965: Generalize the pixel_x/y workaround for all UW types.

Ian Romanick (2):

- mesa: Don't allow GL_TEXTURE_BORDER queries outside compat profile
- mesa: Don't allow GL_TEXTURE_{LUMINANCE,INTENSITY}_* queries outside compat profile

Ilia Mirkin (5):

- nv50/ir: retrieve shadow compare from first arg
- nv50/ir: ignore bias for samplerCubeShadow on nv50
- nvc0/ir: do quadops on the right texture coordinates for TXD
- nvc0/ir: use manual TXD when offsets are involved
- nvc0: make sure that the local memory allocation is aligned to 0x10

Jason Ekstrand (2):

- main/format_pack: Fix a wrong datatype in pack_ubyte_R8G8_UNORM
- main/get_hash_params: Add GL_SAMPLE_SHADING_ARB

Jordan Justen (1):

- i965: Add auxiliary surface field #defines for Broadwell.

José Fonseca (1):

- st/wgl: Clamp wglChoosePixelFormatARB's output nNumFormats to nMaxFormats.

Kenneth Graunke (13):

- i965: Don't copy propagate abs into Broadwell logic instructions.
- i965: Set execution size to 8 for instructions with force_sechalf set.
- i965/fs: Set force_uncompressed and force_sechalf on samplepos setup.
- i965/fs: Use WE_all for gl_SampleID header register munging.
- i965: Add plumbing for Broadwell's auxiliary surface support.
- i965: Drop SINT workaround for CMS layout on Broadwell.
- i965: Hook up the MCS buffers in SURFACE_STATE on Broadwell.
- i965: Add 2x MSAA support to the MCS allocation function.
- i965: Enable compressed multisample support (CMS) on Broadwell.
- i965: Add missing persample_shading field to brw_wm_debug_recompile.
- i965/fs: Fix gl_SampleID for 2x MSAA and SIMD16 mode.
- i965/fs: Fix gl_SampleMask handling for SIMD16 on Gen8+.
- i965/fs: Set LastRT on the final FB write on Broadwell.

Marek Olšák (14):

- gallium: fix u_default_transfer_inline_write for textures
- st/mesa: fix samplerCubeShadow with bias
- radeonsi: fix samplerCubeShadow with bias
- radeonsi: add support for TXB2
- r600g: switch SNORM conversion to DX and GLES behavior
- radeonsi: fix CMASK and HTILE calculations for Hawaii
- gallium/util: add a helper for calculating primitive count from vertex count
- radeonsi: fix a hang with instancing on Hawaii

- radeonsi: fix a hang with streamout on Hawaii
- winsys/radeon: fix vram_size overflow with Hawaii
- radeonsi: fix occlusion queries on Hawaii
- r600g,radeonsi: switch all occurences of array_size to util_max_layer
- radeonsi: fix build because of lack of draw_indirect infrastructure in 10.2
- radeonsi: use DRAW_PREAMBLE on CIK

Matt Turner (8):

- i965/vec4: Don't return void from a void function.
- i965/vec4: Don't fix_math_operand() on Gen >= 8.
- i965/fs: Don't fix_math_operand() on Gen >= 8.
- i965/fs: Make try_constant_propagate() static.
- i965/fs: Constant propagate into 2-src math instructions on Gen8.
- i965/vec4: Constant propagate into 2-src math instructions on Gen8.
- i965/fs: Don't use brw_imm_* unnecessarily.
- i965/fs: Set correct number of regs_written for MCS fetches.

Thorsten Glaser (1):

- nv50: fix build failure on m68k due to invalid struct alignment assumptions

Tom Stellard (1):

- clover: Call end_query before getting timestamp result v2

4.199 Mesa 10.2.4 Release Notes / July 18, 2014

Mesa 10.2.4 is a bug fix release which fixes bugs found since the 10.2.3 release.

Mesa 10.2.4 implements the OpenGL 3.3 API, but the version reported by glGetString(GL_VERSION) or glGetIntegerv(GL_MAJOR_VERSION) / glGetIntegerv(GL_MINOR_VERSION) depends on the particular driver being used. Some drivers don't support all the features required in OpenGL 3.3. OpenGL 3.3 is **only** available if requested at context creation because compatibility contexts are not supported.

4.199.1 SHA256 checksums

```
06a2341244eb85c283f59f70161e06ded106f835ed9b6be1ef0243bd9344811a  MesaLib-10.2.4.tar.
↪bz2
33e3c8b4343503e7d7d17416c670438860a2fd99ec93ea3327f73c3abe33b5e4  MesaLib-10.2.4.tar.
↪gz
e26791a4a62a61b82e506e6ba031812d09697d1a831e8239af67e5722a8ee538  MesaLib-10.2.4.zip
```

4.199.2 New features

None

4.199.3 Bug fixes

This list is likely incomplete.

- Bug 81157 - [BDW]Piglit some spec_glsl-1.50_execution_built-in-functions* cases fail

4.199.4 Changes

Abdiel Janulgue (3):

- i965/fs: Refactor check for potential copy propagated instructions.
- i965/fs: skip copy-propate for logical instructions with negated src entries
- i965/vec4: skip copy-propate for logical instructions with negated src entries

Brian Paul (3):

- mesa: fix geometry shader memory leaks
- st/mesa: fix geometry shader memory leak
- gallium/u_blitter: fix some shader memory leaks

Carl Worth (2):

- docs: Add sha256 checksums for the 10.2.3 release
- Update VERSION to 10.2.4

Eric Anholt (1):

- i965: Generalize the pixel_x/y workaround for all UW types.

Ilia Mirkin (4):

- nv50/ir: retrieve shadow compare from first arg
- nv50/ir: ignore bias for samplerCubeShadow on nv50
- nvc0/ir: do quadops on the right texture coordinates for TXD
- nvc0/ir: use manual TXD when offsets are involved

Jordan Justen (1):

- i965: Add auxiliary surface field #defines for Broadwell.

Kenneth Graunke (9):

- i965: Don't copy propagate abs into Broadwell logic instructions.
- i965: Set execution size to 8 for instructions with force_sechalf set.
- i965/fs: Set force_uncompressed and force_sechalf on samplepos setup.
- i965/fs: Use WE_all for gl_SampleID header register munging.
- i965: Add plumbing for Broadwell's auxiliary surface support.
- i965: Drop SINT workaround for CMS layout on Broadwell.
- i965: Hook up the MCS buffers in SURFACE_STATE on Broadwell.
- i965: Add 2x MSAA support to the MCS allocation function.
- i965: Enable compressed multisample support (CMS) on Broadwell.

Marek Olšák (4):

- gallium: fix u_default_transfer_inline_write for textures
- st/mesa: fix samplerCubeShadow with bias
- radeonsi: fix samplerCubeShadow with bias
- radeonsi: add support for TXB2

Matt Turner (8):

- i965/vec4: Don't return void from a void function.
- i965/vec4: Don't fix_math_operand() on Gen >= 8.
- i965/fs: Don't fix_math_operand() on Gen >= 8.
- i965/fs: Make try_constant_propagate() static.
- i965/fs: Constant propagate into 2-src math instructions on Gen8.
- i965/vec4: Constant propagate into 2-src math instructions on Gen8.
- i965/fs: Don't use brw_imm_* unnecessarily.
- i965/fs: Set correct number of regs_written for MCS fetches.

4.200 Mesa 10.2.3 Release Notes / July 7, 2014

Mesa 10.2.3 is a bug fix release which fixes bugs found since the 10.2.2 release.

Mesa 10.2.3 implements the OpenGL 3.3 API, but the version reported by glGetString(GL_VERSION) or glGetIntegerv(GL_MAJOR_VERSION) / glGetIntegerv(GL_MINOR_VERSION) depends on the particular driver being used. Some drivers don't support all the features required in OpenGL 3.3. OpenGL 3.3 is **only** available if requested at context creation because compatibility contexts are not supported.

4.200.1 SHA256 checksums

```
e482a96170c98b17d6aba0d6e4dda4b9a2e61c39587bb64ac38cadfa4aba4aeb  MesaLib-10.2.3.tar.
↪bz2
96cffacaa1c52ae659b3b0f91be2eebf5528b748934256751261fb79ea3d6636  MesaLib-10.2.3.tar.
↪gz
82cab6ff14c8038ee39842dbdea0d447a78d119efd8d702d1497bc7c246434e9  MesaLib-10.2.3.zip
```

4.200.2 New features

None

4.200.3 Bug fixes

This list is likely incomplete.

- Bug 76223 -
- Bug 79823 -
- Bug 80015 -

4.200.4 Changes

Aaron Watry (1):

- radeon/llvm: Allocate space for kernel metadata operands

Carl Worth (2):

- docs: Add sha256 sums for the 10.2.2 release
- cherry-ignore: Add a patch that's been rejected

Ilia Mirkin (4):

- nouveau: dup fd before passing it to device
- nv50: disable dedicated ubo upload method
- nv50: do an explicit flush on draw when there are persistent buffers
- nvc0: add a memory barrier when there are persistent UBOs

Jasper St. Pierre (1):

- glxext: Send the Drawable's ID in the GLX_BufferSwapComplete event

Kenneth Graunke (3):

- i965: Don't emit SURFACE_STATEs for gather workarounds on Broadwell.
- i965: Include marketing names for Broadwell GPUs.
- i965/disasm: Fix INTEL_DEBUG=fs on Broadwell for ARB_fp applications.

Michel Dänzer (1):

- radeon/llvm: Use the llvm.rsq.clamped intrinsic for RSQ

Rob Clark (9):

- xa: fix segfault
- freedreno: use OUT_RELOCW when buffer is written
- freedreno/a3xx: fix depth/stencil GMEM positioning
- freedreno/a3xx: fix depth/stencil gmem restore
- freedreno/a3xx: fix blend opcode
- freedreno: few caps fixes
- freedreno/a3xx: texture fixes
- freedreno: fix for null textures
- freedreno/a3xx: vtx formats

Roland Scheidegger (1):

- draw: (trivial) fix clamping of viewport index

Takashi Iwai (1):

- llvmpipe: Fix zero-division in llvmpipe_texture_layout()

Thomas Hellstrom (1):

- st/xa: Don't close the drm fd on failure v2

Tobias Klausmann (1):

- nv50/ir: allow gl_ViewportIndex to work on non-provoking vertices

4.201 Mesa 10.2.2 Release Notes / June 24, 2014

Mesa 10.2.2 is a bug fix release which fixes bugs found since the 10.2.1 release.

Mesa 10.2.2 implements the OpenGL 3.3 API, but the version reported by glGetString(GL_VERSION) or glGetIntegerv(GL_MAJOR_VERSION) / glGetIntegerv(GL_MINOR_VERSION) depends on the particular driver being used. Some drivers don't support all the features required in OpenGL 3.3. OpenGL 3.3 is **only** available if requested at context creation because compatibility contexts are not supported.

4.201.1 SHA256 checksums

```
38c4a40364000f89cddaa1694f6f3cfb444981d1110238ce603093585477399c  MesaLib-10.2.2.tar.
↪bz2
2af2ec8b4db624c352e961eefbcce6c8d1f86d44c5542f6f378c50e1b958d453  MesaLib-10.2.2.tar.
↪gz
d4c0372da59367a344d62ebcdf5cf61039c9cae6925f40f2dab8f8d95cf22da9  MesaLib-10.2.2.zip
```

4.201.2 New features

None

4.201.3 Bug fixes

This list is likely incomplete.

- Bug 54372 - GLX_INTEL_swap_event crashes driver when swapping window buffers
- Bug 66452 - JUNIPER UVD accelerated playback of WMV3 streams does not work
- Bug 74005 - [i965 Bisected]Piglit/glx_glx-make-glxdrawable-current fails
- Bug 77865 - [BDW] Many Ogles3conform framebuffer_blit cases fail
- Bug 78581 - OpenCL: clBuildProgram prints error messages directly rather than storing them
- Bug 79029 - INTEL_DEBUG=shader_time is full of lies
- Bug 79729 - [i965] glClear on a multisample texture doesn't work
- Bug 79907 - Mesa 10.2.1 –enable-vdpau default=auto broken
- Bug 80115 - MESA_META_DRAW_BUFFERS induced GL_INVALID_VALUE errors

4.201.4 Changes

Adrian Negreanu (8):

- add megadriver_stub_FILES
- android: adapt to the megadriver mechanism
- android: add libloader to libGLES_mesa and libmesa_egl_dri2

- android: add src/gallium/auxiliary as include path for libmesa_dricore
- android, egl: add correct drm include for libmesa_egl_dri2
- android, egl: typo dri2_fallback_pixmap_surface -> dri2_fallback_create_pixmap_surface
- android, mesa_gen_matypes: pull in timespec POSIX definition
- android, dricore: undefined reference to _mesa_streaming_load_memcpy

Carl Worth (1):

- Update VERSION to 10.2.2

Daniel Manjarres (1):

- glx: Don't crash on swap event for a Window (non-GLXWindow)

Emil Velikov (3):

- targets/xa: limit the amount of exported symbols
- configure: error out when building opencl without LLVM
- configure: correctly autodetect xvmc/vdpau/omx

Grigori Goronzy (1):

- radeon/uvd: disable VC-1 simple/main on UVD 2.x

Iago Toral Quiroga (1):

- mesa: Copy Geom.UsesEndPrimitive when cloning a geometry program.

Ian Romanick (3):

- docs: Add initial 10.2.1 release notes
- docs: Add MD5 checksum, etc. for 10.2.1 release
- meta: Respect the driver's maximum number of draw buffers

Ilia Mirkin (7):

- gk110/ir: emit saturate flag on fadd when needed
- gk110/ir: fix emitting constbuf file index
- gk110/ir: fix bfind emission
- nv50: make sure to mark first scissor dirty after blit
- nv30: plug some memory leaks on screen destroy and shader compile
- nv30: avoid dangling references to deleted contexts
- nv30: hack to avoid errors on unexpected color/zeta combinations

Jason Ekstrand (1):

- meta_blit: properly compute texture width for the CopyTexSubImage fallback

José Fonseca (1):

- mesa/main: Prevent sefgault on glGetIntegerv(GL_ATOMIC_COUNTER_BUFFER_BINDING).

Kenneth Graunke (9):

- i965: Don't use the head sentinel as an fs_inst in Gen4 workaround code.
- i965: Invalidate live intervals when inserting Gen4 SEND workarounds.

- i965/vec4: Fix dead code elimination for VGRFs of size > 1.
- i965: Add missing MOCS setup for 3DSTATE_INDEX_BUFFER on Broadwell.
- i965: Drop Broadwell perf_debugs about missing MOCS that aren't missing.
- i965: Add missing newlines to a few perf_debug messages.
- i965/vec4: Use the sampler for pull constant loads on Broadwell.
- i965: Use 8x4 aligned rectangles for HiZ operations on Broadwell.
- i965: Save meta stencil blit programs in the context.

Kristian Høgsberg (1):

- mesa: Remove glClear optimization based on drawable size

Michel Dänzer (1):

- configure: Only check for OpenCL without LLVM when the latter is certain

Neil Roberts (1):

- i965: Set the fast clear color value for texture surfaces

Tom Stellard (2):

- clover: Prevent Clang from printing number of errors and warnings to stderr.
- clover: Don't use llvm's global context

Ville Syrjälä (1):

- i915: Fix gen2 texblend setup

4.202 Mesa 10.2.1 Release Notes / June 6, 2014

Mesa 10.2.1 is a bug fix release which fixes bugs found since the 10.2 release.

Mesa 10.2.1 implements the OpenGL 3.3 API, but the version reported by glGetString(GL_VERSION) or glGetIntegerv(GL_MAJOR_VERSION) / glGetIntegerv(GL_MINOR_VERSION) depends on the particular driver being used. Some drivers don't support all the features required in OpenGL 3.3. OpenGL 3.3 is **only** available if requested at context creation because compatibility contexts are not supported.

4.202.1 MD5 checksums

```
96f892dae2d0bb14ac9c2113f586c909  MesaLib-10.2.1.tar.gz
093f9b5d077e5f6061dcd7b01b7aa51a  MesaLib-10.2.1.tar.bz2
6ab76c1608e5deed1eb8b54c62d7a48a  MesaLib-10.2.1.zip
```

4.202.2 New features

None

4.202.3 Bug fixes

Mesa 10.2 had a build problem in the radeonsi driver due to an error resolving conflicts in a patch cherry-pick from master. The build error is fixed.

4.202.4 Changes

Ian Romanick (3):

- docs: Add MD5 checksum, etc. for 10.1 release
- radeonsi: Fix build error introduced in 5ab9a9c
- Bump version to 10.2.1

4.203 Mesa 10.2 Release Notes / June 6, 2014

Mesa 10.2 is a new development release. People who are concerned with stability and reliability should stick with a previous release or wait for Mesa 10.2.1.

Mesa 10.2 implements the OpenGL 3.3 API, but the version reported by glGetString(GL_VERSION) or glGetIntegerv(GL_MAJOR_VERSION) / glGetIntegerv(GL_MINOR_VERSION) depends on the particular driver being used. Some drivers don't support all the features required in OpenGL 3.3. OpenGL 3.3 is **only** available if requested at context creation because compatibility contexts are not supported.

4.203.1 MD5 checksums

```
c87bfb6dd5cbcf1fdef42e5ccd972581  MesaLib-10.2.0.tar.gz
7aaba90bd7169a94ae2fe83febdec963  MesaLib-10.2.0.tar.bz2
58b203aca15dadc25ab4d1126db1052b  MesaLib-10.2.0.zip
```

4.203.2 New features

Note: some of the new features are only available with certain drivers.

- GL_ARB_buffer_storage on i965, nv30, nv50, nvc0, r300, r600, and radeonsi
- GL_ARB_multi_bind on all drivers
- GL_ARB_sample_shading on nv50 (GT21x only), nvc0
- GL_ARB_separate_shader_objects (desktop OpenGL) and GL_EXT_separate_shader_objects (OpenGL ES 2.0 and 3.0) on all drivers
- GL_ARB_stencil_texturing on i965/gen8+
- GL_ARB_texture_cube_map_array on nv50 (GT21x only)
- GL_ARB_texture_gather on nv50 (GT21x only), nvc0
- GL_ARB_texture_query_lod on nv50 (GT21x only), nvc0
- GL_ARB_texture_view on i965/gen7
- GL_ARB_vertex_type_10f_11f_11f_rev on nv50, nvc0, radeonsi
- GL_ARB_viewport_array on nv50, r600

- GL_INTEL_performance_query on i965/gen5+

4.203.3 Bug fixes

TBD.

4.203.4 Changes

- Renamed *–with-llvm-shared-libs* to *–enable-llvm-shared-libs*

  The option is used to control how mesa is linked against LLVM, and now defaults to enabled (shared linking).

- Split *libxatracker.so* into a standalone library which can be used with any gallium driver.

  Previously the library was linked statically against vmware's virtual gpu driver(svga), whereas now it loads a shared pipe_*.so driver. Provide the following options during configure, if you would like support for svga driver *–enable-xa –with-gallium-drivers=svga*

  Note: The files are installed in $(libdir)/gallium-pipe/ and the interface between them and libxatracker.so is **not** stable.

- The environment variable GALLIUM_MSAA that forced a multisample GLX visual was removed.

4.204 Mesa 10.1.6 Release Notes / (June 24, 2014)

Mesa 10.1.6 is a bug fix release which fixes bugs found since the 10.1.5 release.

Mesa 10.1.6 implements the OpenGL 3.3 API, but the version reported by glGetString(GL_VERSION) or glGetIntegerv(GL_MAJOR_VERSION) / glGetIntegerv(GL_MINOR_VERSION) depends on the particular driver being used. Some drivers don't support all the features required in OpenGL 3.3. OpenGL 3.3 is **only** available if requested at context creation because compatibility contexts are not supported.

4.204.1 SHA256 checksums

```
cde60e06b340d7598802fe4a4484b3fb8befd714f9ab9caabe1f27d3149e8815  MesaLib-10.1.6.tar.
↪bz2
e4e726d7805a442f7ed07d12f71335e6126796ec85328a5989eb5348a8042d00  MesaLib-10.1.6.tar.
↪gz
bf7e3f721a7ad0c2057a034834b6fea688e64f26a66cf8d1caa2827e405e72dd  MesaLib-10.1.6.zip
```

4.204.2 New features

None

4.204.3 Bug fixes

This list is likely incomplete.

- Bug 54372 - GLX_INTEL_swap_event crashes driver when swapping window buffers
- Bug 74005 - [i965 Bisected]Piglit/glx_glx-make-glxdrawable-current fails

- Bug 78581 -
- Bug 79729 - [i965] glClear on a multisample texture doesn't work

4.204.4 Changes

Adrian Negreanu (7):

- add megadriver_stub_FILES
- android: adapt to the megadriver mechanism
- android: add libloader to libGLES_mesa and libmesa_egl_dri2
- android: add src/gallium/auxiliary as include path for libmesa_dricore
- android, egl: add correct drm include for libmesa_egl_dri2
- android, mesa_gen_matypes: pull in timespec POSIX definition
- android, dricore: undefined reference to _mesa_streaming_load_memcpy

Beren Minor (1):

- egl/main: Fix eglMakeCurrent when releasing context from current thread.

Carl Worth (3):

- docs: Add SHA256 checksums for the 10.1.5 release
- cherry-ignore: Add a patch to ignore
- Update VERSION to 10.1.6

Daniel Manjarres (1):

- glx: Don't crash on swap event for a Window (non-GLXWindow)

Emil Velikov (1):

- configure: error out when building opencl without LLVM

Iago Toral Quiroga (1):

- mesa: Copy Geom.UsesEndPrimitive when cloning a geometry program.

José Fonseca (3):

- mesa/main: Make get_hash.c values constant.
- mesa: Make glGetIntegerv(GL_*_ARRAY_SIZE) return GL_BGRA.
- mesa/main: Prevent sefgault on glGetIntegerv(GL_ATOMIC_COUNTER_BUFFER_BINDING).

Kristian Høgsberg (1):

- mesa: Remove glClear optimization based on drawable size

Michel Dänzer (1):

- configure: Only check for OpenCL without LLVM when the latter is certain

Neil Roberts (1):

- i965: Set the fast clear color value for texture surfaces

Roland Scheidegger (1):

- draw: (trivial) fix clamping of viewport index

Tobias Klausmann (1):

- nv50/ir: clear subop when folding constant expressions

Tom Stellard (2):

- clover: Prevent Clang from printing number of errors and warnings to stderr.
- clover: Don't use llvm's global context

4.205 Mesa 10.1.5 Release Notes / (June 6, 2014)

Mesa 10.1.5 is a bug fix release which fixes bugs found since the 10.1.4 release.

Mesa 10.1.5 implements the OpenGL 3.3 API, but the version reported by glGetString(GL_VERSION) or glGetIntegerv(GL_MAJOR_VERSION) / glGetIntegerv(GL_MINOR_VERSION) depends on the particular driver being used. Some drivers don't support all the features required in OpenGL 3.3. OpenGL 3.3 is **only** available if requested at context creation because compatibility contexts are not supported.

4.205.1 SHA256 checksums

```
b0aceaa75bc9a9b2d9215a113e2ad488b5cf85c99005a7624f8cf7c37c5d0eaa  MesaLib-10.1.5.tar.
↪gz
bc6c5ec7836f254a49d055a29d9aa34c97c54c038f47ad3a00fa57a5fef15bbc  MesaLib-10.1.5.tar.
↪bz2
78b7255cab0af7918945452a84de7989096ebcdd27e99b31c56c0589274cbc77  MesaLib-10.1.5.zip
```

4.205.2 New features

None

4.205.3 Bug fixes

This list is likely incomplete.

- Bug 79115 -
- Bug 79421 -

4.205.4 Changes

Brian Paul (1):

- glsl: fix use-after free bug/crash in ast_declarator_list::hir()

Carl Worth (5):

- docs: Add md5sums for 10.1.4 release
- Merge remote-tracking branch 'origin/10.1' into 10.1
- cherry-ignore: Ignore two commits.
- Ignore a patch that is not needed for the 10.1 branch.

- Update version to 10.1.5

Emil Velikov (1):

- glx: do not leak dri3Display

Ilia Mirkin (2):

- nv50/ir: fix s32 x s32 -> high s32 multiply logic
- nv50/ir: fix constant folding for OP_MUL subop HIGH

James Legg (1):

- mesa: Fix unbinding GL_DEPTH_STENCIL_ATTACHMENT

Jeremy Huddleston Sequoia (2):

- glapi: Avoid heap corruption in _glapi_table
- darwin: Fix test for kCGLPFAOpenGLProfile support at runtime

Pavel Popov (2):

- i965: Properly return *RESET* status in glGetGraphicsResetStatusARB
- i965: Fix Line Stipple enable bit in 3DSTATE_SF for Haswell.

Roland Scheidegger (1):

- llvmpipe: fix crash when not all attachments are populated in a fb

4.206 Mesa 10.1.4 Release Notes / (May 20, 2014)

Mesa 10.1.4 is a bug fix release which fixes bugs found since the 10.1.3 release.

Mesa 10.1.4 implements the OpenGL 3.3 API, but the version reported by glGetString(GL_VERSION) or glGetIntegerv(GL_MAJOR_VERSION) / glGetIntegerv(GL_MINOR_VERSION) depends on the particular driver being used. Some drivers don't support all the features required in OpenGL 3.3. OpenGL 3.3 is **only** available if requested at context creation because compatibility contexts are not supported.

4.206.1 MD5 checksums

```
e934365d77f384bfaec844999440bef8  MesaLib-10.1.4.tar.gz
6fddee101f49b7409cd29994c34ddee7  MesaLib-10.1.4.tar.bz2
ba5f48e7d5e373922c804c2651fec6c1  MesaLib-10.1.4.zip
```

4.206.2 New features

None

4.206.3 Bug fixes

This list is likely incomplete.

- Bug 78225 - Compile error due to undefined reference to 'gbm_dri_backend', fix attached
- Bug 78537 - no anisotropic filtering in a native Half-Life 2

4.206.4 Changes

Brian Paul (1):

- mesa: fix double-freeing of dispatch tables inside glBegin/End.

Carl Worth (3):

- docs: Add MD5 sums for 10.1.3
- cherry-ignore: Roland and Michel agreed to drop these patches.
- VERSION: Update to 10.1.4

Emil Velikov (1):

- configure: error out if building GBM without dri

Eric Anholt (1):

- i965/vs: Use samplers for UBOs in the VS like we do for non-UBO pulls.

Ilia Mirkin (3):

- nv50/ir: make sure to reverse cond codes on all the OP_SET variants
- nv50: fix setting of texture ms info to be per-stage
- nv50/ir: fix integer mul lowering for u32 x u32 -> high u32

Michel Dänzer (1):

- radeonsi: Fix anisotropic filtering state setup

Tom Stellard (2):

- configure.ac: Add LLVM_VERSION_PATCH to DEFINES
- radeonsi: Enable geometry shaders with LLVM 3.4.1

4.207 Mesa 10.1.3 Release Notes / (May 9, 2014)

Mesa 10.1.3 is a bug fix release which fixes bugs found since the 10.1.2 release.

Note: Mesa 10.1.3 is being released sooner than originally scheduled to make available a fix for a performance rgression that was inadvertently introduced to Mesa 10.1.2. The performance regression is reported to make vmware swapbuffers fall back to software.

Mesa 10.1.3 implements the OpenGL 3.3 API, but the version reported by glGetString(GL_VERSION) or glGetIntegerv(GL_MAJOR_VERSION) / glGetIntegerv(GL_MINOR_VERSION) depends on the particular driver being used. Some drivers don't support all the features required in OpenGL 3.3. OpenGL 3.3 is **only** available if requested at context creation because compatibility contexts are not supported.

4.207.1 MD5 checksums

```
665fe1656aaa2c37b32042068aff92cb  MesaLib-10.1.3.tar.gz
ba6dbe2b9cab0b4de840c996b9b6a3ad  MesaLib-10.1.3.tar.bz2
4e6f26330a63d3c47e62ac4bdead39e8  MesaLib-10.1.3.zip
```

4.207.2 New features

None

4.207.3 Bug fixes

This list is likely incomplete.

- Bug 77245 - Bogus GL_ARB_explicit_attrib_location layout identifier warnings

4.207.4 Changes

Carl Worth (3):

- docs: Add MD5 sums for Mesa 10.1.2
- get-pick-list.sh: Require explicit "10.1" for nominating stable patches
- VERSION: Update to 10.1.3

Kenneth Graunke (2):

- mesa: Fix MaxNumLayers for 1D array textures.
- i965: Fix depth (array slices) computation for 1D_ARRAY render targets.

Tapani Pälli (1):

- glsl: fix bogus layout qualifier warnings

Thomas Hellstrom (1):

- st/xa: Fix performance regression introduced by commit "Cache render target surface"

4.208 Mesa 10.1.2 Release Notes / (May 5, 2014)

Mesa 10.1.2 is a bug fix release which fixes bugs found since the 10.1.1 release.

Mesa 10.1.2 implements the OpenGL 3.3 API, but the version reported by glGetString(GL_VERSION) or glGetIntegerv(GL_MAJOR_VERSION) / glGetIntegerv(GL_MINOR_VERSION) depends on the particular driver being used. Some drivers don't support all the features required in OpenGL 3.3. OpenGL 3.3 is **only** available if requested at context creation because compatibility contexts are not supported.

4.208.1 MD5 checksums

```
37d79f94b1f41852a89d1fc3900bea76  MesaLib-10.1.2.tar.gz
28b60d15ac9f364da1e0155911eaf44e  MesaLib-10.1.2.tar.bz2
05300039085a65fc53c5472c4bb5747a  MesaLib-10.1.2.zip
```

4.208.2 New features

None

4.208.3 Bug fixes

This list is likely incomplete.

- Bug 27499 - [855GM i915] GL_LINE_STIPPLE displays incorrect colors
- Bug 75723 - (regression since Linux 3.14?) brw_get_graphics_reset_status: Assertion 'brw->hw_ctx != ((void *)0)' failed
- Bug 76894 - Piglit/spec/EXT_framebuffer_object/fbo-bind-renderbuffer failed
- Bug 77702 - [i965 Bisected]Piglit spec/NV_conditional_render_blitframebuffer fails

4.208.4 Changes

Ander Conselvan de Oliveira (2):

- gbm/dri: Fix out-of-memory error path in dri_device_create()
- egl: Protect use of gbm_dri with ifdef HAVE_DRM_PLATFORM

Anuj Phogat (27):

- mesa: Fix glGetVertexAttribi(GL_VERTEX_ATTRIB_ARRAY_SIZE)
- swrast: Add glBlitFramebuffer to commands affected by conditional rendering
- mesa: Fix error condition for multisample proxy texture targets
- i965: Put an assertion to check valid varying_to_slot[varying]
- i965: Fix component mask and varying_to_slot mapping for gl_Layer
- i965: Fix component mask and varying_to_slot mapping for gl_ViewportIndex
- mesa: Add helper function _mesa_is_format_integer()
- mesa: Add error condition for integer formats in glGetTexImage()
- mesa: Add an error condition in glGetFramebufferAttachmentParameteriv()
- mesa: Fix error code generation in glReadPixels()
- glsl: Allow overlapping locations for vertex input attributes
- mesa: Fix querying location of nth element of an array variable
- mesa: Use location VERT_ATTRIB_GENERIC0 for vertex attribute 0
- glsl: Compile error if fs defines conflicting qualifiers for gl_FragCoord
- glsl: Compile error if fs uses gl_FragCoord before first redeclaration
- mesa: Add entry for extension ARB_texture_stencil8
- mesa: Add error condition for format=STENCIL_INDEX in glGetTexImage()
- i965: Fix crash in do_blit_readpixels()
- mesa: Add missing types in _mesa_texstore_xx_xx() functions
- mesa: Allow srcFormat=GL_DEPTH_STENCIL in _mesa_texstore_xx_xx() functions
- mesa: Add new helper function _mesa_unpack_depth_stencil_row()
- mesa: Add support to unpack depth-stencil texture in to FLOAT_32_UNSIGNED_INT_24_8_REV
- mesa: Allow FLOAT_32_UNSIGNED_INT_24_8_REV in get_tex_depth_stencil()

- i965: Add glBlitFramebuffer to commands affected by conditional rendering
- glsl: Use switch to allow adding more shader types
- glsl: Link error if fs defines conflicting qualifiers for gl_FragCoord
- glsl: Apply the link error conditions to GL_ARB_fragment_coord_conventions

Benjamin Bellec (1):

- mesa: fix GetStringi error message with correct function name

Brian Paul (1):

- swrast: allocate swrast_texture_image::ImageSlices array if needed

Carl Worth (4):

- docs: Add the MD5 sums for the 10.1.1 release tar files.
- cherry-ignore: Ignore a patch causing a regression
- cherry-ignore: Drop an ignored patch now that piglit has been updated.
- Update VERSION to 10.1.2

Chris Forbes (1):

- glsl: Only allow ‘invariant‘ on shader in/out between stages.

Eric Anholt (1):

- i965: Fix render-to-texture in non-FinishRenderTexture cases.

Ian Romanick (1):

- dri3: Enable GLX_MESA_query_renderer on DRI3 too

Kenneth Graunke (2):

- i965: Don’t enable reset notification support on Gen4-5.
- i965: Actually emit PIPELINE_SELECT and 3DSTATE_VF_STATISTICS.

Marek Olšák (10):

- r300g: don’t crash when getting NULL colorbuffers
- st/mesa: remove trailing NULL colorbuffers
- r600g: fix edge flags and layered rendering on R600-R700
- r600g: disable async DMA on R700
- r600g: fix MSAA resolve on R6xx when the destination is 1D-tiled
- r600g: fix flushing on RV670, RS780, RS880 again
- r600g: fix buffer copying on R600-R700
- r600g: fix for broken CULL_FRONT behavior on R6xx
- r600g: fix for an MSAA hang on RV770
- r600g: fix hang on RV740 by using DX_RASTERIZATION_KILL instead of SX_MISC

Michel Dänzer (2):

- r600g: Disable LLVM by default at runtime for graphics
- st/mesa: Fix NULL pointer dereference for incomplete framebuffers

Neil Roberts (1):

- wayland: Fix the logic in disabling the prime capability

Samuel Iglesias Gonsalvez (1):

- mesa: fix check for dummy renderbuffer in _mesa_FramebufferRenderbufferEXT()

Thomas Hellstrom (1):

- st/xa: Cache render target surface

nick (1):

- swrast: Fix vertex color in _swsetup_Translate()

4.209 Mesa 10.1.1 Release Notes / April 18, 2014

Mesa 10.1.1 is a bug fix release which fixes bugs found since the 10.1 release.

Mesa 10.1.1 implements the OpenGL 3.3 API, but the version reported by glGetString(GL_VERSION) or glGetIntegerv(GL_MAJOR_VERSION) / glGetIntegerv(GL_MINOR_VERSION) depends on the particular driver being used. Some drivers don't support all the features required in OpenGL 3.3. OpenGL 3.3 is **only** available if requested at context creation because compatibility contexts are not supported.

4.209.1 MD5 checksums

```
96e63674ccfa98e7ec6eb4fee3f770c3  MesaLib-10.1.1.tar.gz
1fde7ed079df7aeb9b6a744ca033de8d  MesaLib-10.1.1.tar.bz2
e64d0a562638664b13d2edf22321df59  MesaLib-10.1.1.zip
```

4.209.2 New features

None

4.209.3 Bug fixes

- Bug 71547 - compilation failure :#error "SSE4.1 instruction set not enabled"
- Bug 74868 - r600g: Diablo III Crashes After a few minutes
- Bug 74988 - Buffer overrun (segfault) decompressing ETC2 texture in GLBenchmark 3.0 Manhattan
- Bug 75279 - XCloseDisplay() takes one minute around nouveau_dri.so, freezing Firefox startup
- Bug 75543 - OSMesa Gallium OSMesaMakeCurrent
- Bug 75660 - u_inlines.h:277:pipe_buffer_map_range: Assertion 'length' failed.
- Bug 76323 - GLSL compiler ignores layout(binding=N) on uniform blocks
- Bug 76377 - DRI3 should only be enabled on Linux due to a udev dependency
- Bug 76749 - [HSW] DOTA world lighting has no effect
- Bug 77102 - gallium nouveau has no profile in vdpau and libva
- Bug 77207 - [ivb/hsw] batch overwritten with garbage

4.209.4 Changes

Aaron Watry (1):

- gallium/util: Fix memory leak

Alexander von Gluck IV (1):

- haiku: Fix build through scons corrections and viewport fixes

Anuj Phogat (2):

- mesa: Set initial internal format of a texture to GL_RGBA
- mesa: Allow GL_DEPTH_COMPONENT and GL_DEPTH_STENCIL combinations in glTexImage{123}D()

Brian Paul (12):

- softpipe: use 64-bit arithmetic in softpipe_resource_layout()
- mesa: don't call ctx->Driver.ClearBufferSubData() if size==0
- st/osmesa: check buffer size when searching for buffers
- mesa: fix copy & paste bugs in pack_ubyte_SARGB8()
- mesa: fix copy & paste bugs in pack_ubyte_SRGB8()
- c11/threads: don't include assert.h if the assert macro is already defined
- mesa: fix unpack_Z32_FLOAT_X24S8() / unpack_Z32_FLOAT() mix-up
- st/mesa: add null pointer checking in query object functions
- mesa: fix glMultiDrawArrays inside a display list
- cso: fix sampler view count in cso_set_sampler_views()
- svga: replace sampler assertion with conditional
- svga: move LIST_INITHEAD(dirty_buffers) earlier in svga_context_create()

Carl Worth (3):

- cherry-ignore: Ignore a few patches
- glsl: Allow explicit binding on atomics again
- Update VERSION to 10.1.1

Chia-I Wu (1):

- i965/vec4: fix record clearing in copy propagation

Christian König (2):

- st/mesa: recreate sampler view on context change v3
- st/mesa: fix sampler view handling with shared textures v4

Courtney Goeltzenleuchter (1):

- mesa: add bounds checking to eliminate buffer overrun

Emil Velikov (5):

- nv50: add missing brackets when handling the samplers array
- mesa: return v.value_int64 when the requested type is TYPE_INT64
- configure: enable dri3 only for linux

- glx: drop obsolete _XUnlock_Mutex in __glXInitialize error path
- configure: cleanup libudev handling

Eric Anholt (1):

- i965: Fix buffer overruns in MSAA MCS buffer clearing.

Hans (2):

- util: don't define isfinite(), isnan() for MSVC >= 1800
- mesa: don't define c99 math functions for MSVC >= 1800

Ian Romanick (7):

- linker: Split set_uniform_binding into separate functions for blocks and samplers
- linker: Various trivial clean-ups in set_sampler_binding
- linker: Fold set_uniform_binding into call site
- linker: Clean up "unused parameter" warnings
- linker: Set block bindings based on UniformBlocks rather than UniformStorage
- linker: Set binding for all elements of UBO array
- glsl: Propagate explicit binding information from the AST all the way to the linker

Ilia Mirkin (8):

- nouveau: fix fence waiting logic in screen destroy
- nv50: adjust blit_3d handling of ms output textures
- loader: add special logic to distinguish nouveau from nouveau_vieux
- mesa/main: condition GL_DEPTH_STENCIL on ARB_depth_texture
- nouveau: add forgotten GL_COMPRESSED_INTENSITY to texture format list
- nouveau: there may not have been a texture if the fbo was incomplete
- nvc0/ir: move sample id to second source arg to fix sampler2DMS
- nouveau: fix firmware check on nvd7/nvd9

Johannes Nixdorf (1):

- configure.ac: fix the detection of expat with pkg-config

Jonathan Gray (7):

- gallium: add endian detection for OpenBSD
- loader: use 0 instead of FALSE which isn't defined
- loader: don't limit the non-udev path to only android
- megadriver_stub.c: don't use _GNU_SOURCE to gate the compat code
- egl/dri2: don't require libudev to build drm/wayland platforms
- egl/dri2: use drm macros to construct device name
- configure: don't require libudev for gbm or egl drm/wayland

José Fonseca (4):

- c11/threads: Fix nano to milisecond conversion.

- mapi/u_thread: Use GetCurrentThreadId
- c11/threads: Don't implement thrd_current on Windows.
- draw: Duplicate TGSI tokens in draw_pipe_pstipple module.

Kenneth Graunke (4):

- i965/fs: Fix register comparisons in saturate propagation.
- glsl: Fix lack of i2u in lower_ubo_reference.
- i965: Stop advertising GL_MESA_ycbcr_texture.
- glsl: Try vectorizing when seeing a repeated assignment to a channel.

Marek Olšák (13):

- r600g: fix texelFetchOffset GLSL functions
- r600g: fix blitting the last 2 mipmap levels for Evergreen
- mesa: fix the format of glEdgeFlagPointer
- r600g,radeonsi: fix MAX_TEXTURE_3D_LEVELS and MAX_TEXTURE_ARRAY_LAYERS limits
- st/mesa: fix per-vertex edge flags and GLSL support (v2)
- mesa: mark GL_RGB9_E5 as not color-renderable
- mesa: fix texture border handling for cube arrays
- mesa: allow generating mipmaps for cube arrays
- mesa: fix software fallback for generating mipmaps for cube arrays
- mesa: fix software fallback for generating mipmaps for 3D textures
- st/mesa: fix generating mipmaps for cube arrays
- st/mesa: drop the lowering of quad strips to triangle strips
- r600g: implement edge flags

Matt Turner (4):

- mesa: Wrap SSE4.1 code in #ifdef __SSE4_1__.
- i965/fs: Fix off-by-one in saturate propagation.
- i965/fs: Don't propagate saturate modifiers into partial writes.
- i965/fs: Don't propagate saturation modifiers if there are source modifiers.

Michel Dänzer (1):

- r600g: Don't leak bytecode on shader compile failure

Mike Stroyan (1):

- i965: Avoid dependency hints on math opcodes

Thomas Hellstrom (5):

- winsys/svga: Replace the query mm buffer pool with a slab pool v3
- winsys/svga: Update the vmwgfx_drm.h header to latest version from kernel
- winsys/svga: Fix prime surface references also for guest-backed surfaces
- st/xa: Bind destination before setting new state

- st/xa: Make sure unused samplers are set to NULL

Tom Stellard (1):

- configure: Use LLVM shared libraries by default

4.210 Mesa 10.1 Release Notes / March 4, 2014

Mesa 10.1 is a new development release. People who are concerned with stability and reliability should stick with a previous release or wait for Mesa 10.1.1.

Mesa 10.1 implements the OpenGL 3.3 API, but the version reported by glGetString(GL_VERSION) or glGetIntegerv(GL_MAJOR_VERSION) / glGetIntegerv(GL_MINOR_VERSION) depends on the particular driver being used. Some drivers don't support all the features required in OpenGL 3.3. OpenGL 3.3 is **only** available if requested at context creation because compatibility contexts are not supported.

4.210.1 MD5 checksums

```
3ec43f79dbcd9aa2a4a27bf1f51655b6  MesaLib-10.1.0.tar.bz2
08e796ec7122aa299d32d4f67a254315  MesaLib-10.1.0.tar.gz
bd365356543f4b38e57c1ddf7a317c40  MesaLib-10.1.0.zip
```

4.210.2 New features

Note: some of the new features are only available with certain drivers.

- GL_ARB_draw_indirect on i965.
- GL_ARB_clear_buffer_object
- GL_ARB_viewport_array on i965.
- GL_ARB_map_buffer_alignment on all drivers that did not previously support it.
- GL_AMD_shader_trinary_minmax.
- GL_EXT_framebuffer_blit on r200 and radeon.
- Reduced memory usage for display lists.
- OpenGL 3.3 support on nv50, nvc0, r600 and radeonsi

4.210.3 Bug fixes

TBD.

4.210.4 Changes

- Removed support for the GL_MESA_texture_array extension. This extension enabled the use of texture array with fixed-function and assembly fragment shaders. No applications are known to use this extension.

4.211 Mesa 10.0.5 Release Notes / April 18, 2014

Mesa 10.0.5 is a bug fix release which fixes bugs found since the 10.0.4 release.

Mesa 10.0.5 implements the OpenGL 3.3 API, but the version reported by glGetString(GL_VERSION) or glGetIntegerv(GL_MAJOR_VERSION) / glGetIntegerv(GL_MINOR_VERSION) depends on the particular driver being used. Some drivers don't support all the features required in OpenGL 3.3. OpenGL 3.3 is **only** available if requested at context creation because compatibility contexts not supported.

4.211.1 MD5 checksums

```
db606aadd0fe321f3664099677d159bc  MesaLib-10.0.5.tar.gz
e6009ccd8898d7104bb325b6af9ec354  MesaLib-10.0.5.tar.bz2
c8ab9e502542bf32299a4df85b0b704d  MesaLib-10.0.5.zip
```

4.211.2 New features

None

4.211.3 Bug fixes

This list is likely incomplete.

- Bug 58660 - CAYMAN broken with HyperZ on
- Bug 64471 - Radeon HD6570 lockup in Brütal Legend with HyperZ
- Bug 66352 - GPU lockup in L4D2 on TURKS with HyperZ
- Bug 68799 - [APITRACE] Hyper-Z lockup with Falcon BMS 4.32u6 on CAYMAN
- Bug 71547 - compilation failure :#error "SSE4.1 instruction set not enabled"
- Bug 72685 - [radeonsi hyperz] Artifacts in Unigine Sanctuary
- Bug 73088 - [HyperZ] Juniper (6770): Gone Home / Unigine Heaven 4.0 lock up system after several minutes of use
- Bug 74428 - hyperz causes gpu hang in Counter-strike: Source
- Bug 74803 - [r600g] HyperZ broken on RV630 (Cogs shadows are broken)
- Bug 74863 - [r600g] HyperZ broken on RV770 and CYPRESS (Left 4 Dead 2 trees corruption) bisected!
- Bug 74892 - HyperZ GPU lockup with radeonsi 7970M PITCAIRN and Distance Alpha game
- Bug 74988 - Buffer overrun (segfault) decompressing ETC2 texture in GLBenchmark 3.0 Manhattan
- Bug 75279 - XCloseDisplay() takes one minute around nouveau_dri.so, freezing Firefox startup
- Bug 77102 - gallium nouveau has no profile in vdpau and libva
- Bug 77207 - [ivb/hsw] batch overwritten with garbage

4.211.4 Changes

The full set of changes can be viewed by using the following git command:

```
git log mesa-10.0.4..mesa-10.0.5
```

Alex Deucher (1):

- radeon: reverse DBG_NO_HYPERZ logic

Brian Paul (9):

- mesa: add unpacking code for MESA_FORMAT_Z32_FLOAT_S8X24_UINT
- mesa: fix copy & paste bugs in pack_ubyte_SARGB8()
- mesa: fix copy & paste bugs in pack_ubyte_SRGB8()
- mesa: fix unpack_Z32_FLOAT_X24S8() / unpack_Z32_FLOAT() mix-up
- st/mesa: add null pointer checking in query object functions
- mesa: fix glMultiDrawArrays inside a display list
- cso: fix sampler view count in cso_set_sampler_views()
- svga: replace sampler assertion with conditional
- svga: move LIST_INITHEAD(dirty_buffers) earlier in svga_context_create()

Carl Worth (3):

- docs: Add md5sums for the 10.0.4 release.
- Ignore patches which don't apply.
- Update version to 10.0.5

Christian König (2):

- st/mesa: recreate sampler view on context change v3
- st/mesa: fix sampler view handling with shared textures v4

Courtney Goeltzenleuchter (1):

- mesa: add bounds checking to eliminate buffer overrun

Emil Velikov (2):

- mesa: return v.value_int64 when the requested type is TYPE_INT64
- glx: drop obsolete _XUnlock_Mutex in __glXInitialize error path

Eric Anholt (1):

- i965: Fix buffer overruns in MSAA MCS buffer clearing.

Ilia Mirkin (6):

- nouveau: fix fence waiting logic in screen destroy
- nv50: adjust blit_3d handling of ms output textures
- mesa/main: condition GL_DEPTH_STENCIL on ARB_depth_texture
- nouveau: add forgotten GL_COMPRESSED_INTENSITY to texture format list
- nouveau: there may not have been a texture if the fbo was incomplete

- nouveau: fix firmware check on nvd7/nvd9

Johannes Nixdorf (1):

- configure.ac: fix the detection of expat with pkg-config

Jonathan Gray (1):

- gallium: add endian detection for OpenBSD

José Fonseca (1):

- draw: Duplicate TGSI tokens in draw_pipe_pstipple module.

Matt Turner (1):

- mesa: Wrap SSE4.1 code in #ifdef __SSE4_1__.

Paul Berry (1):

- i965/gen7: Prefer vertical alignment of 4 when possible.

4.212 Mesa 10.0.4 Release Notes / (March 12, 2014)

Mesa 10.0.4 is a bug fix release which fixes bugs found since the 10.0.3 release.

Mesa 10.0.4 implements the OpenGL 3.3 API, but the version reported by glGetString(GL_VERSION) or glGetIntegerv(GL_MAJOR_VERSION) / glGetIntegerv(GL_MINOR_VERSION) depends on the particular driver being used. Some drivers don't support all the features required in OpenGL 3.3. OpenGL 3.3 is **only** available if requested at context creation because compatibility contexts not supported.

4.212.1 MD5 checksums

```
5a3c5b90776ec8a9fcd777c99e0607e2  MesaLib-10.0.4.tar.gz
8b148869d2620b0720c8a8d2b7eb3e38  MesaLib-10.0.4.tar.bz2
da2418d25bfbc273660af7e755fb367e  MesaLib-10.0.4.zip
```

4.212.2 New features

None

4.212.3 Bug fixes

This list is likely incomplete.

- Bug 71870 - Metro: Last Light rendering issues
- Bug 72895 - Missing trees in flightgear 2.12.1 with mesa 10.0.1
- Bug 74251 - Segfault in st_finalize_texture with Texture Buffer
- Bug 74723 - main/shaderapi.c:407: detach_shader: Assertion 'shProg->Shaders[j]->Type == 0x8B31 || shProg->Shaders[j]->Type == 0x8B30' failed.

4.212.4 Changes

The full set of changes can be viewed by using the following git command:

```
git log mesa-10.0.3..mesa-10.0.4
```

Anuj Phogat (4):

- mesa: Generate correct error code in glDrawBuffers()
- mesa: Add GL_TEXTURE_CUBE_MAP_ARRAY to legal_get_tex_level_parameter_target()
- glsl: Fix condition to generate shader link error
- i965: Fix the region's pitch condition to use blitter

Brian Paul (8):

- r200: move driContextSetFlags(ctx) call after ctx var is initialized
- radeon: move driContextSetFlags(ctx) call after ctx var is initialized
- gallium/auxiliary/indices: replace free() with FREE()
- draw: fix incorrect color of flat-shaded clipped lines
- st/mesa: avoid sw fallback for getting/decompressing textures
- mesa: update assertion in detach_shader() for geom shaders
- mesa: do depth/stencil format conversion in glGetTexImage
- softpipe: use 64-bit arithmetic in softpipe_resource_layout()

Carl Worth (4):

- docs: Add md5sums for 10.0.3 release
- main: Avoid double-free of shader Label
- get-pick-list: Update to only find patches nominated for the 10.0 branch
- Update version to 10.0.4

Chris Forbes (1):

- i965: Validate (and resolve) all the bound textures.

Christian König (1):

- radeon/uvd: fix feedback buffer handling v2

Daniel Kurtz (1):

- glsl: Add locking to builtin_builder singleton

Emil Velikov (3):

- dri/nouveau: Pass the API into _mesa_initialize_context
- nv50: correctly calculate the number of vertical blocks during transfer map
- dri/i9*5: correctly calculate the amount of system memory

Fredrik Höglund (3):

- mesa: Preserve the NewArrays state when copying a VAO
- glx: Fix the default values for GLXFBConfig attributes

- glx: Fix the GLXFBConfig attrib sort priorities

Hans (2):

- util: don't define isfinite(), isnan() for MSVC >= 1800
- mesa: don't define c99 math functions for MSVC >= 1800

Ian Romanick (6):

- meta: Release resources used by decompress_texture_image
- meta: Release resources used by _mesa_meta_DrawPixels
- meta: Fallback to software for GetTexImage of compressed GL_TEXTURE_CUBE_MAP_ARRAY
- meta: Consistenly use non-Apple VAO functions
- glcpp: Only warn for macro names containing __
- glsl: Only warn for macro names containing __

Ilia Mirkin (3):

- nv30: report 8 maximum inputs
- nouveau/video: make sure that firmware is present when checking caps
- nouveau: fix chipset checks for nv1a by using the oclass instead

Julien Cristau (1):

- glx/dri2: fix build failure on HURD

Kenneth Graunke (2):

- glsl: Don't lose precision qualifiers when encountering "centroid".
- i965: Create a hardware context before initializing state module.

Kusanagi Kouichi (1):

- targets/vdpau: Always use c++ to link

Marek Olšák (1):

- st/mesa: fix crash when a shader uses a TBO and it's not bound

Matt Turner (1):

- glsl: Initialize ubo_binding_mask flags to zero.

Paul Berry (2):

- glsl: Make condition_to_hir() callable from outside ast_iteration_statement.
- glsl: Fix continue statements in do-while loops.

Tom Stellard (1):

- r600g/compute: PIPE_CAP_COMPUTE should be false for pre-evergreen GPUs

Topi Pohjolainen (1):

- i965/blorp: do not use unnecessary hw-blending support

4.213 Mesa 10.0.3 Release Notes / (February 3, 2014)

Mesa 10.0.3 is a bug fix release which fixes bugs found since the 10.0.2 release.

Mesa 10.0.3 implements the OpenGL 3.3 API, but the version reported by glGetString(GL_VERSION) or glGetIntegerv(GL_MAJOR_VERSION) / glGetIntegerv(GL_MINOR_VERSION) depends on the particular driver being used. Some drivers don't support all the features required in OpenGL 3.3. OpenGL 3.3 is **only** available if requested at context creation because compatibility contexts not supported.

4.213.1 MD5 checksums

```
5f9f463ef08129f6762106b434910adb  MesaLib-10.0.3.tar.bz2
fb3997b6500e153bc32370cb3fc4ca9e  MesaLib-10.0.3.tar.gz
a07b4b6b9eb449b88a6cb5061e51c331  MesaLib-10.0.3.zip
```

4.213.2 New features

None

4.213.3 Bug fixes

This list is likely incomplete.

- Bug 72708 - Master fails to build with older gcc due to -msse4.1
- Bug 72926 - [REGRESSION,swrast] Memory-related crash with anti-aliasing enabled
- Bug 73096 - Query GL_RGBA_SIGNED_COMPONENTS_EXT missing
- Bug 73100 - Please use AC_PATH_TOOL instead of AC_PATH_PROG for llvm-config
- Bug 73418 - OpenCL hangs graphics on CAYMAN
- Bug 73473 - Potential crash bug in src/gallium/auxiliary/rtasm/rtasm_execmem.c
- Bug 73915 - sample shading + centroid broken since f5cfb4a
- Bug 73956 - SIGSEGV when passing GL_NONE to glReadBuffer
- Bug 74026 - Compiler rejects chained assignments involving array dereferences

4.213.4 Changes

The full set of changes can be viewed by using the following git command:

```
git log mesa-10.0.2..mesa-10.0.3
```

Aaron Watry (2):

- radeon: Move gfx/dma cs cleanup to r600_common_context_cleanup
- st/dri: prevent leak of dri option default values

Andreas Fänger (1):

- swrast: fix delayed texel buffer allocation regression for OpenMP

Anuj Phogat (3):

- glsl: Disable ARB_texture_rectangle in shader version 100.
- i965: Use sample barycentric coordinates with per sample shading
- i965: Ignore 'centroid' interpolation qualifier in case of persample shading

Brian Paul (3):

- mesa: implement missing glGet(GL_RGBA_SIGNED_COMPONENTS_EXT) query
- st/mesa: fix glReadBuffer(GL_NONE) segfault
- draw: fix incorrect vertex size computation in LLVM drawing code

Carl Worth (5):

- Add md5sums for 10.0.2. release.
- cherry-ignore: Ignore several patches not yet ready for the stable branch
- Drop another couple of patches.
- cherry-ignore: Ignore 4 patches at the request of the author, (Anuj).
- Update version to 10.0.3

Chad Versace (1):

- i965/gen6/blorp: Emit more flushes to workaround hangs

Chris Forbes (1):

- i965: fold offset into coord for textureOffset(gsampler2DRect)

Emil Velikov (5):

- mesa: use signed temporary variable to store _ColorDrawBufferIndexes
- st/mesa: use signed temporary variable to store _ColorDrawBufferIndexes
- nv50: access only the available amount of textures
- nv50: access only the available amount of constbuf
- gallium/rtasm: handle mmap failures appropriately

Eric Anholt (2):

- i965: Fix handling of MESA_pack_invert in blit (PBO) readpixels.
- i965: Don't do the temporary-and-blit-copy for INVALIDATE_RANGE maps.

Ian Romanick (2):

- mesa: Add COMPRESSED_RGBA_S3TC_DXT1_EXT to COMPRESSED_TEXTURE_FORMATS for GLES
- radeon / r200: Pass the API into _mesa_initialize_context

Ilia Mirkin (2):

- mesa: fix GL_COLOR_SUM enum for drivers without ARB_vertex_program
- st/vdpau: don't return a device if the screen doesn't support NPOT

José Fonseca (1):

- mesa: Use IROUND instead of roundf.

Kenneth Graunke (2):

- glsl: Rename "expr" to "lhs_expr" in vector_extract munging code.
- glsl: Fix chained assignments of vector channels.

Lauri Kasanen (1):

- mesa: Fix build to properly check for supported compiler flags

Marek Olšák (2):

- st/mesa: use sRGB formats for MSAA resolving if destination is sRGB
- gallium/util: util_format_srgb should not return FORMAT_NONE for sRGB formats

Matt Turner (2):

- glcpp: Define GL_EXT_shader_integer_mix in both GL and ES.
- glx: Update glxext.h to revision 24777.

Michał Górny (1):

- Use AC_PATH_TOOL instead of AC_PATH_PROG for llvm-config.

Paul Berry (1):

- i965: Ensure that all necessary state is re-emitted if we run out of aperture.

Paul Seidler (1):

- build: move ARCH_LIBS definition outside of ASM definition

Thomas Sondergaard (4):

- mesa: Preliminary support for MSVC_VERSION=12.0
- mesa: Fix compile error with MSVC 2013
- mesa: Work around internal compiler error
- mesa: Namespace qualify fma to override ambiguity with fma from math.h

Tom Stellard (1):

- r600g/compute: Emit DEALLOC_STATE on cayman after dispatching a compute shader.

4.214 Mesa 10.0.2 Release Notes / (January 9, 2014)

Mesa 10.0.2 is a bug fix release which fixes bugs found since the 10.0.1 release.

Mesa 10.0.2 implements the OpenGL 3.3 API, but the version reported by glGetString(GL_VERSION) or glGetIntegerv(GL_MAJOR_VERSION) / glGetIntegerv(GL_MINOR_VERSION) depends on the particular driver being used. Some drivers don't support all the features required in OpenGL 3.3. OpenGL 3.3 is **only** available if requested at context creation because compatibility contexts not supported.

4.214.1 MD5 checksums

```
de7d14baf0101b697c140d2f47ef27e9  MesaLib-10.0.2.tar.gz
8544c0ab3e438a08b5103421ea15b6d2  MesaLib-10.0.2.tar.bz2
181b0d6c1afca38e98a930d0e564ed90  MesaLib-10.0.2.zip
```

4.214.2 New features

None

4.214.3 Bug fixes

This list is likely incomplete.

- Bug 70740 - HiZ on SNB causes GPU hang with WebGL web app
- Bug 72026 - SIGSEGV in fs_visitor::visit(ir_dereference_variable*)
- Bug 72264 - GLSL error reporting
- Bug 72369 - glitches in serious sam 3 with the sb shader backend

4.214.4 Changes

The full set of changes can be viewed by using the following git command:

```
git log mesa-10.0.1..mesa-10.0.2
```

Aaron Watry (8):

- clover: Remove unused variable
- pipe_loader/sw: close dev->lib when initialization fails
- radeon/compute: Stop leaking LLVMContexts in radeon_llvm_parse_bitcode
- r600/compute: Free compiled kernels when deleting compute state
- r600/compute: Use the correct FREE macro when deleting compute state
- radeon/llvm: Free target data at end of optimization
- st/vdpau: Destroy context when initialization fails
- r600/pipe: Stop leaking context->start_compute_cs_cmd.buf on EG/CM

Alex Deucher (1):

- r600g: fix SUMO2 pci id

Alexander von Gluck IV (1):

- Haiku: Add in public GL kit headers

Anuj Phogat (1):

- mesa: Fix error code generation in glBeginConditionalRender()

Carl Worth (2):

- docs: Add md5sums for the 10.0.1 release.
- Update version to 10.0.2

Chad Versace (1):

- i965/gen6: Fix HiZ hang in WebGL Google Maps

Erik Faye-Lund (1):

- glcpp: error on multiple #else/#elif directives

Henri Verbeet (1):

- i915: Add support for gl_FragData[0] reads.

Ilia Mirkin (1):

- nv50: fix a small leak on context destroy

Jonathan Liu (2):

- st/mesa: use pipe_sampler_view_release()
- llvmpipe: use pipe_sampler_view_release() to avoid segfault

Kenneth Graunke (2):

- i965: Fix 3DSTATE_PUSH_CONSTANT_ALLOC_PS packet creation.
- Revert "mesa: Remove GLXContextID typedef from glx.h."

Kevin Rogovin (1):

- Use line number information from entire function expression

Kristian Høgsberg (1):

- dri_util: Don't assume __DRIcontext->driverPrivate is a gl_context

Marek Olšák (2):

- mesa: fix interpretation of glClearBuffer(drawbuffer)
- st/mesa: fix glClear with multiple colorbuffers and different formats

Paul Berry (2):

- glsl: Teach ir_variable_refcount about ir_loop::counter variables.
- glsl: Fix inconsistent assumptions about ir_loop::counter.

Vadim Girlin (1):

- r600g/sb: fix stack size computation on evergreen

4.215 Mesa 10.0.1 Release Notes / (December 12, 2013)

Mesa 10.0.1 is a bug fix release which fixes bugs found since the 10.0 release.

Mesa 10.0.1 implements the OpenGL 3.3 API, but the version reported by glGetString(GL_VERSION) or glGetIntegerv(GL_MAJOR_VERSION) / glGetIntegerv(GL_MINOR_VERSION) depends on the particular driver being used. Some drivers don't support all the features required in OpenGL 3.3. OpenGL 3.3 is **only** available if requested at context creation because compatibility contexts not supported.

4.215.1 MD5 checksums

```
0a72ca5b36046a658bf6038326ff32ed  MesaLib-10.0.1.tar.bz2
01bde35c912e504ba62caf1ef9f7022c  MesaLib-10.0.1.tar.gz
59a174a11a89e6b1b8ee9c3f7e3c388c  MesaLib-10.0.1.zip
```

4.215.2 New features

None

4.215.3 Bug fixes

This list is likely incomplete.

- Bug 64323 - Severe misrendering in Left 4 Dead 2
- Bug 68838 - GLSL: struct declarations produce a "empty declaration warning" in 9.2
- Bug 69155 - [NV50 gallium] [piglit] bin/varying-packing-simple triggers memory corruption/failures
- Bug 70250 - weston-terminal rendering corrupted with output transform 90 and 270
- Bug 70601 - [SNB Bisected]Piglit spec/ARB_texture_float/multisample-formats 2 GL_ARB_texture_float fails
- Bug 72230 - Unable to extract MesaLib-10.0.0.tar.{gz,bz2} with bsdtar
- Bug 72325 - [swrast] piglit glean fbo regression
- Bug 72327 - [swrast] piglit glean pointSprite regression

4.215.4 Changes

The full set of changes can be viewed by using the following git command:

```
git log mesa-10.0..mesa-10.0.1
```

Axel Davy (2):

- egl/wayland: Flush the wl_display at the end of SwapBuffers
- Enable throttling in SwapBuffers

Chad Versace (2):

- i965/hsw: Apply non-msrt fast color clear w/a to all HSW GTs
- i965: Add extra-alignment for non-msrt fast color clear for all hw (v2)

Dave Airlie (1):

- swrast: fix readback regression since inversion fix

Emil Velikov (1):

- automake: include only one copy VERSION in tarball

Ian Romanick (3):

- docs: Add 10.0 release md5sums
- Remove a057b83 from the pick list
- glsl: Don't emit empty declaration warning for a struct specifier

Ilia Mirkin (8):

- mesa: don't leak performance monitors on context destroy
- nv50: Fix GPU_READING/WRITING bit removal
- nouveau: avoid leaking fences while waiting

- nv50: wait on the buf's fence before sticking it into pushbuf
- nv50: enable h264 and mpeg4 for nv98+ (vp3, vp4.0)
- nouveau/video: update h264 picparm field names based on usage
- nouveau/video: update a few more h264 picparm field names
- nv50: report 15 max inputs for fragment programs

Jordan Justen (1):

- dri megadriver_stub: add compatibility for older DRI loaders

Kristian Høgsberg (2):

- egl/wayland: Damage INT32_MAX x INT32_MAX region for eglSwapBuffers
- egl/wayland: Send commit after flushing the driver context

Maarten Lankhorst (1):

- nouveau: Fix compiler warning regression

Paul Berry (1):

- i965/gen6: Fix multisample resolve blits for luminance/intensity 32F formats.

Thomas Hellstrom (1):

- st/xa: Bump major version number to 2

Tom Stellard (2):

- r300/compiler/tests: Fix segfault
- r300/compiler/tests: Fix line length check in test parser

4.216 Mesa 10.0 Release Notes / (November 30th, 2013)

Mesa 10.0 is a new development release. People who are concerned with stability and reliability should stick with a previous release or wait for Mesa 10.0.1.

Mesa 10.0 implements the OpenGL 3.3 API, but the version reported by glGetString(GL_VERSION) or glGetIntegerv(GL_MAJOR_VERSION) / glGetIntegerv(GL_MINOR_VERSION) depends on the particular driver being used. Some drivers don't support all the features required in OpenGL 3.3. OpenGL 3.3 is **only** available if requested at context creation because compatibility contexts are not supported.

4.216.1 MD5 checksums

```
b38626b96c664db67a534d7859682436  MesaLib-10.0.0.tar.gz
f3fe55d9735bea158bbe97ed9a0da819  MesaLib-10.0.0.tar.bz2
c6ee1ce51e3bf35947d2978b872daf51  MesaLib-10.0.0.zip
```

4.216.2 New features

Note: some of the new features are only available with certain drivers.

- GL_AMD_seamless_cubemap_per_texture on i965.

- GL_ARB_conservative_depth on i965.
- GL_ARB_texture_gather on i965.
- GL_ARB_texture_query_levels on i965.
- GL_ARB_texture_mirror_clamp_to_edge.
- GL_ARB_transform_feedback2, GL_ARB_transform_feedback3, and GL_ARB_transform_feedback_instanced on i965/Gen7 (with appropriate kernel support).
- GL_ARB_sample_shading on i965.
- GL_ARB_shader_atomic_counters on i965.
- GL_ARB_vertex_attrib_binding
- GL_ARB_vertex_type_10f_11f_11f_rev on i965 and r600g
- GL_KHR_debug
- GLX_MESA_query_renderer

4.216.3 Bug fixes

Attempts have been made to **not** include bugs fixed in previous 9.2 releases or bugs that were regressions during 10.0 development. This list is likely incomplete.

- Bug 47755 - [glsl-compiler] no error checking when Interpolation qualifier for built-in variable is different in vertex and fragment shader
- Bug 52171 - [gallium/r600/clover] Simple benchmarks failed to run
- Bug 53077 - [IVB] Output error with msaa when both of framebuffer and source color's alpha are not 1
- Bug 54867 - bug in r300 compiler
- Bug 60929 - [r600-llvm] mono games with opengl are blocking on start
- Bug 62142 - Mesa/demo mipmap_limits upside down with running by SOFTWARE
- Bug 62698 - [bisected] WebGL demo "Consumed": texstate.c:628: update_texture_state: Assertion „__builtin_popcount(enabledTargets) == 1" failed.
- Bug 64225 - bfgminer –scyte generates Segmentation Fault on Northern Island
- Bug 64226 - python-opencl package generate segmentation fault at pipe_r600.so
- Bug 64261 - [SNB Bisected]Ogles3conform GL3Tests_color_buffer_float_color_buffer_float_clamp_fixed.test fail
- Bug 66213 - Certain Mesa Demos Rendering Inverted (vertically)
- Bug 66806 - [softpipe] glxgears floating point exception
- Bug 67921 - [bisected commit 883987] crosscompiling fails with util/u_cpu_detect.c:247:4: error: 'asm' undeclared (first use in this function)
- Bug 68162 - [radeonsi] texture rendering is broken in Source-Engine games
- Bug 68451 - Texture flicker in native Dota2 in mesa 9.2.0rc1
- Bug 68503 - Graphical glitches in Serious Sam 3 when SB is enabled
- Bug 68792 - Problems during playback of h264 files using UVD and VLC on AMD E-350 CPU
- Bug 68845 - VDPAU/UVD regression

- Bug 69078 - Modern Warfare (1, 2 and 3) broken in Wine on SNB
- Bug 69321 - starting openCL crashes/boots system
- Bug 70042 - Major texture flickering in Dota 2 (r600g on HD 6950)
- Bug 70088 - Glamor on r600g crashes Xserver
- Bug 70123 - Freeze caused by 'winsys/radeon: remove cs_queue_empty' commit
- Bug 70327 - Casting floating point variable to integer not working properly while constant gets converted properly
- Bug 70891 - CL_INVALID_BUILD_OPTIONS results in CL_INVALID_DEVICE when asking for build log
- Bug 70913 - [PIGLIT,radeonsi] crash in "spec/EXT_framebuffer_multisample/sample-alpha-to-coverage 4 depth" (buffer overflow)
- Bug 71022 - configure: error: Expat required for DRI.
- Bug 71110 - xorg_driver.c:1030:2: error: too many arguments to function 'DamageUnregister'
- Bug 71172 - Segfault when running glxinfo. NV25GL [Quadro4 900 XGL]
- Bug 71512 - dlopen.h:54: undefined reference to 'dlopen'
- Bug 71870 - Metro: Last Light rendering issues

4.216.4 Changes

- Removed X.Org state tracker (unmaintained and broken)
- Removed the video-accel r300 targets
- Removed the video-accel softpipe targets

4.217 Mesa 9.2.5 Release Notes / (December 12, 2013)

Mesa 9.2.5 is a bug fix release which fixes bugs found since the 9.2.4 release.

Mesa 9.2 implements the OpenGL 3.1 API, but the version reported by glGetString(GL_VERSION) or glGetIntegerv(GL_MAJOR_VERSION) / glGetIntegerv(GL_MINOR_VERSION) depends on the particular driver being used. Some drivers don't support all the features required in OpenGL 3.1. OpenGL 3.1 is **only** available if requested at context creation because GL_ARB_compatibility is not supported.

4.217.1 MD5 checksums

```
9fb4de29ca1d9cfd03cbdefa123ba336  MesaLib-9.2.5.tar.bz2
1146c7c332767174f3de782b88d8e8ca  MesaLib-9.2.5.tar.gz
a9a6c46dac7ea26fd272bf14894d95f3  MesaLib-9.2.5.zip
```

4.217.2 New features

None

4.217.3 Bug fixes

This list is likely incomplete.

- Bug 62142 - Mesa/demo mipmap_limits upside down with running by SOFTWARE
- Bug 64323 - Severe misrendering in Left 4 Dead 2
- Bug 66213 - Certain Mesa Demos Rendering Inverted (vertically)
- Bug 68838 - GLSL: struct declarations produce a "empty declaration warning" in 9.2
- Bug 69155 - [NV50 gallium] [piglit] bin/varying-packing-simple triggers memory corruption/failures
- Bug 72325 - [swrast] piglit glean fbo regression
- Bug 72327 - [swrast] piglit glean pointSprite regression

4.217.4 Changes

The full set of changes can be viewed by using the following GIT command:

```
git log mesa-9.2.4..mesa-9.2.5
```

Chad Versace (2):

- i965/hsw: Apply non-msrt fast color clear w/a to all HSW GTs
- i965: Add extra-alignment for non-msrt fast color clear for all hw (v2)

Chris Forbes (4):

- i965: Gen4-5: Don't enable hardware alpha test with MRT
- i965: Gen4-5: Include alpha func/ref in program key
- i965/fs: Gen4-5: Setup discard masks for MRT alpha test
- i965/fs: Gen4-5: Implement alpha test in shader for MRT

Chí-Thanh Christopher Nguyn (1):

- st/xorg: Handle new DamageUnregister API which has only one argument

Dave Airlie (3):

- mesa/swrast: fix inverted front buffer rendering with old-school swrast
- glx: don't fail out when no configs if we have visuals
- swrast: fix readback regression since inversion fix

Ian Romanick (1):

- glsl: Don't emit empty declaration warning for a struct specifier

Ilia Mirkin (4):

- nv50: Fix GPU_READING/WRITING bit removal
- nouveau: avoid leaking fences while waiting
- nv50: wait on the buf's fence before sticking it into pushbuf
- nv50: report 15 max inputs for fragment programs

Tom Stellard (2):

- r300/compiler/tests: Fix segfault
- r300/compiler/tests: Fix line length check in test parser

4.218 Mesa 9.2.4 Release Notes / (November 27, 2013)

Mesa 9.2.4 is a bug fix release which fixes bugs found since the 9.2.3 release.

Mesa 9.2 implements the OpenGL 3.1 API, but the version reported by glGetString(GL_VERSION) or glGetIntegerv(GL_MAJOR_VERSION) / glGetIntegerv(GL_MINOR_VERSION) depends on the particular driver being used. Some drivers don't support all the features required in OpenGL 3.1. OpenGL 3.1 is **only** available if requested at context creation because GL_ARB_compatibility is not supported.

4.218.1 MD5 checksums

```
28190b831b0271d69dbc44b2686eab1c  MesaLib-9.2.4.tar.gz
e630c0a307cec4f0f70ddd029d2fe084  MesaLib-9.2.4.tar.bz2
8ef5e1e92e1d30fbedec31f716a7619e  MesaLib-9.2.4.zip
```

4.218.2 New features

None

4.218.3 Bug fixes

This list is likely incomplete.

- Bug 53077 - [IVB] Output error with msaa when both of framebuffer and source color's alpha are not 1
- Fix freedreno to compile with recent libdrm.

4.218.4 Changes

The full set of changes can be viewed by using the following GIT command:

```
git log mesa-9.2.3..mesa-9.2.4
```

Brian Paul (1):

- st/mesa: fix GL_FEEDBACK mode inverted Y coordinate bug

Paul Berry (2):

- i965: Fix vertical alignment for multisampled buffers.
- glsl: Fix lowering of direct assignment in lower_clip_distance.

Rob Clark (17):

- freedreno/a3xx: fix color inversion on mem->gmem restore
- freedreno/a3xx: fix viewport on gmem->mem resolve
- freedreno: add debug option to disable scissor optimization

- freedreno: update register headers
- freedreno/a3xx: some texture fixes
- freedreno/a3xx/compiler: fix CMP
- freedreno/a3xx/compiler: handle saturate on dst
- freedreno/a3xx/compiler: use max_reg rather than file_count
- freedreno/a3xx/compiler: cat4 cannot use const reg as src
- freedreno: fix segfault when no color buffer bound
- freedreno/a3xx/compiler: make compiler errors more useful
- freedreno/a3xx/compiler: bit of re-arrange/cleanup
- freedreno/a3xx/compiler: fix SGT/SLT/etc
- freedreno/a3xx: don't leak so much
- freedreno/a3xx/compiler: better const handling
- freedreno/a3xx/compiler: handle sync flags better
- freedreno: updates for msm drm/kms driver

Tapani Pälli (1):

- mesa: enable GL_TEXTURE_LOD_BIAS set/get

4.219 Mesa 9.2.3 Release Notes / (November 13, 2013)

Mesa 9.2.3 is a bug fix release which fixes bugs found since the 9.2.2 release.

Mesa 9.2 implements the OpenGL 3.1 API, but the version reported by glGetString(GL_VERSION) or glGetIntegerv(GL_MAJOR_VERSION) / glGetIntegerv(GL_MINOR_VERSION) depends on the particular driver being used. Some drivers don't support all the features required in OpenGL 3.1. OpenGL 3.1 is **only** available if requested at context creation because GL_ARB_compatibility is not supported.

4.219.1 MD5 checksums

```
66e9a33a414f801e1c33398bf627d56b  MesaLib-9.2.3.tar.gz
f56b6beb556e4b9072814419f7c554e3  MesaLib-9.2.3.tar.bz2
ed852dab576faac237ac4298bf55d0a1  MesaLib-9.2.3.zip
```

4.219.2 New features

None

4.219.3 Bug fixes

This list is likely incomplete.

- Bug 69437 - Composite Bypass no longer works

4.219.4 Changes

The full set of changes can be viewed by using the following GIT command:

```
git log mesa-9.2.2..mesa-9.2.3
```

Brian Paul (2):

- st/mesa: move out of memory check in st_draw_vbo()
- osmesa: fix broken triangle/line drawing when using float color buffer

Carl Worth (7):

- Remove error when calling glGenQueries/glDeleteQueries while a query is active
- Bump version to 9.2.3

Daniel Vetter (1):

- i965: CS writes/reads should use I915_GEM_INSTRUCTION

Eric Anholt (1):

- i965: Fix texture buffer rendering after a whole buffer replacement.

Kenneth Graunke (6):

- i965: Emit post-sync non-zero flush before 3DSTATE_GS_SVB_INDEX.
- i965: Emit post-sync non-zero flush before 3DSTATE_DRAWING_RECTANGLE.
- i965: Also guard 3DSTATE_DRAWING_RECTANGLE with a flush in blorp.
- i965: Move post-sync non-zero flush for 3DSTATE_MULTISAMPLE.
- i965: Also emit HIER_DEPTH and STENCIL packets when disabling depth.
- i965: Also emit HiZ and Stencil packets when disabling depth on Gen6.

Kristian Høgsberg (1):

- wayland: Don't rely on static variable for identifying wl_drm buffers

Marek Olšák (1):

- radeonsi: fix blitting the last 2 mipmap levels of compressed textures

Petr Sebor (1):

- meta: enable vertex attributes in the context of the newly created array object

Scott Graham (1):

- mesa: fixes for MSVC 2013

4.220 Mesa 9.2.2 Release Notes / (October 18, 2013)

Mesa 9.2.2 is a bug fix release which fixes bugs found since the 9.2.1 release.

Mesa 9.2 implements the OpenGL 3.1 API, but the version reported by glGetString(GL_VERSION) or glGetIntegerv(GL_MAJOR_VERSION) / glGetIntegerv(GL_MINOR_VERSION) depends on the particular driver being used. Some drivers don't support all the features required in OpenGL 3.1. OpenGL 3.1 is **only** available if requested at context creation because GL_ARB_compatibility is not supported.

4.220.1 MD5 checksums

```
df801a975045150790e10e2ccf32193f  MesaLib-9.2.2.tar.gz
20887f8020db7d1736a01ae9cd5d8c38  MesaLib-9.2.2.tar.bz2
1676f4f1b157c838d077dadd31ba6c84  MesaLib-9.2.2.zip
```

4.220.2 New features

None

4.220.3 Bug fixes

This list is likely incomplete.

- Bug 69449 - Valgrind error in program_resource_visitor::recursion
- Bug 70411 - glInvalidateFramebuffer fails with GL_INVALID_ENUM

4.220.4 Changes

The full set of changes can be viewed by using the following GIT command:

```
git log mesa-9.2.1..mesa-9.2.2
```

Brian Paul (3):

- docs: add missing <pre> tag
- svga: fix incorrect memcpy src in svga_buffer_upload_piecewise()
- mesa: consolidate cube width=height error checking

Carl Worth (3):

- docs: Add md5sums for 9.2.1 release
- Bump version to 9.2.2

Constantin Baranov (1):

- mesa: Add missing switch break in invalidate_framebuffer_storage()

Eric Anholt (3):

- i965: Don't forget the cube map padding on gen5+.
- mesa: Fix compiler warnings when ALIGN's alignment is "1 << value".
- i965: Fix 3D texture layout by more literally copying from the spec.

Francisco Jerez (1):

- glsl: Fix usage of the wrong union member in program_resource_visitor::recursion.

Tom Stellard (1):

- radeonsi: Use 'SI' as the LLVM processor for CIK on LLVM <= 3.3

4.221 Mesa 9.2.1 Release Notes / (October 4, 2013)

Mesa 9.2.1 is a bug fix release which fixes bugs found since the 9.2 release.

Mesa 9.2 implements the OpenGL 3.1 API, but the version reported by glGetString(GL_VERSION) or glGetIntegerv(GL_MAJOR_VERSION) / glGetIntegerv(GL_MINOR_VERSION) depends on the particular driver being used. Some drivers don't support all the features required in OpenGL 3.1. OpenGL 3.1 is **only** available if requested at context creation because GL_ARB_compatibility is not supported.

4.221.1 MD5 checksums

```
e6cdfa84dfddd86e3d36ec7ff4b6478a  MesaLib-9.2.1.tar.gz
dd4c82667d9c19c28a553b12eba3f8a0  MesaLib-9.2.1.tar.bz2
d9af0f5607f7d275793d293057ca9ac6  MesaLib-9.2.1.zip
```

4.221.2 New features

None

4.221.3 Bug fixes

This list is likely incomplete.

- Bug 66779 - Use of uninitialized stack variable with brw_search_cache()
- Bug 68233 - Valgrind errors in mesa
- Bug 68250 - Automatic mipmap generation with texture compression produces borders that fade to black
- Bug 68637 - [Bisected IVB/HSW]Unigine demo crash
- Bug 68753 - [regression bisected] GLSL ES: structs members can't have precision qualifiers anymore in 9.2
- Bug 69525 - [GM45, bisected] Piglit tex-shadow2drect fails

4.221.4 Changes

The full set of changes can be viewed by using the following GIT command:

```
git log mesa-9.2..mesa-9.2.1
```

Alex Deucher (1):

- radeon/winsys: pad IBs to a multiple of 8 DWs

Andreas Boll (1):

- os: First check for __GLIBC__ and then for PIPE_OS_BSD

Anuj Phogat (1):

- glsl: Allow precision qualifiers for sampler types

Brian Paul (2):

- docs: minor fixes for 9.2 release notes

- mesa: check for bufSize > 0 in _mesa_GetSynciv()

Carl Worth (3):

- cherry-ignore: Ignore a commit which appeared twice on master
- Use -Bsymbolic when linking libEGL.so
- mesa: Bump version to 9.2.1

Chris Forbes (3):

- i965/fs: Gen4: Zero out extra coordinates when using shadow compare
- i965: Fix cube array coordinate normalization
- i965: fix bogus swizzle in brw_cubemap_normalize

Christoph Bumiller (2):

- nvc0/ir: add f32 long immediate cannot saturate
- nvc0: delete compute object on screen destruction

Dave Airlie (1):

- st/mesa: don't dereference stObj->pt if NULL

Dominik Behr (1):

- glsl: propagate max_array_access through function calls

Emil Velikov (1):

- nouveau: initialise the nouveau_transfer maps

Eric Anholt (4):

- mesa: Rip out more extension checking from texformat.c.
- mesa: Don't choose S3TC for generic compression if we can't compress.
- i965/gen4: Fix fragment program rectangle texture shadow compares.
- i965: Reenable glBitmap() after the sRGB winsys enabling.

Ian Romanick (7):

- docs: Add 9.2 release md5sums
- Add .cherry-ignore file
- mesa: Note that 89a665e should not be picked
- glsl: Reallow precision qualifiers on structure members
- mesa: Support GL_MAX_VERTEX_OUTPUT_COMPONENTS query with ES3
- mesa: Remove all traces of GL_OES_matrix_get
- mesa: Don't return any data for GL_SHADER_BINARY_FORMATS

Ilia Mirkin (2):

- nv30: find first unused texcoord rather than bailing if first is used
- nv30: fix inconsistent setting of push->user_priv

Joakim Sindholt (1):

- nvc0: fix blitctx memory leak

Johannes Obermayr (1):

- st/gbm: Add $(WAYLAND_CFLAGS) for HAVE_EGL_PLATFORM_WAYLAND.

Kenneth Graunke (5):

- i965/vs: Detect GRF sources in split_virtual_grfs send-from-GRF code.
- i965/fs: Detect GRF sources in split_virtual_grfs send-from-GRF code.
- i965/vec4: Only zero out unused message components when there are any.
- i965: Fix brw_vs_prog_data_compare to actually check field members.
- meta: Set correct viewport and projection in decompress_texture_image.

Maarten Lankhorst (2):

- st/dri: do not create a new context for msaa copy
- nvc0: restore viewport after blit

Marek Olšák (2):

- r600g: fix constant buffer cache flushing
- r600g: fix texture buffer object cache flushing

Paul Berry (1):

- i965: Initialize inout_offset parameter to brw_search_cache().

Rico Schüller (1):

- glx: Initialize OpenGL version to 1.0

Tiziano Bacocco (1):

- nvc0/ir: fix use after free in texture barrier insertion pass

Torsten Duwe (1):

- wayland-egl.pc requires wayland-client.pc.

4.222 Mesa 9.2 Release Notes / (August 27, 2013)

Mesa 9.2 is a new development release. People who are concerned with stability and reliability should stick with a previous release or wait for Mesa 9.2.1.

Mesa 9.2 implements the OpenGL 3.1 API, but the version reported by glGetString(GL_VERSION) or glGetIntegerv(GL_MAJOR_VERSION) / glGetIntegerv(GL_MINOR_VERSION) depends on the particular driver being used. Some drivers don't support all the features required in OpenGL 3.1. OpenGL 3.1 is **only** available if requested at context creation because GL_ARB_compatibility is not supported.

4.222.1 MD5 checksums

```
4f93c6475ec656fc1f7b93aeffc9b6c4  MesaLib-9.2.0.tar.gz
4185b6aae890bc62a964f4b24cc1aca8  MesaLib-9.2.0.tar.bz2
3bc5339bc98b9c37777ffd14e3a8eca4  MesaLib-9.2.0.zip
```

4.222.2 New features

Note: some of the new features are only available with certain drivers.

- GL_ARB_shading_language_420pack in all drivers that support GLSL 1.30.
- GL_ARB_texture_buffer_range
- GL_ARB_texture_multisample
- GL_ARB_texture_storage_multisample
- GL_ARB_texture_query_lod
- GL_ARB_texture_storage on radeon, r200, and nouveau
- GL_EXT_discard_framebuffer in all OpenGL ES (all versions) drivers
- GL_EXT_framebuffer_multisample_blit_scaled on i965
- Added new freedreno gallium driver
- OSMesa interface for gallium llvmpipe/softpipe drivers
- Gallium Heads-Up Display (HUD) feature for performance monitoring
- Added support for UVD (2.2 and 3.0) video decoding on r600g and radeonsi through VDPAU (requires Kernel 3.10 or later)

4.222.3 Bug fixes

Attempts have been made to **not** include bugs fixed in previous 9.1 releases or bugs that were regressions during 9.2 development. This list is likely incomplete.

- Bug 41787 - [llvmpipe] stencil broken
- Bug 44618 - Cross-compilation broken by glsl builtin_compiler
- Bug 46632 - Make the alignment checks for the readpixel blit fastpath a bit more lenient
- Bug 47116 - Enemy territory freezes with rs880 and commit fbebd431ec4e2e461a0cbcd5f3a04a000b8f6bbf
- Bug 47248 - autogen missing dependency on flex and bison, causes infinite loop in glsl build
- Bug 48694 - radeonsi_pipe.c:322:7: error: 'PIPE_CAP_DUAL_SOURCE_BLEND' undeclared
- Bug 50655 - [r600g][RV670 HD3870] Ioquake games causes GPU lockup (waiting for 0x00003039 last fence id 0x00003030)
- Bug 51471 - [965gm] Corrupted graphics in corners of screen with pixel shaders enabled
- Bug 51782 - mesa-8.0.3: fails to compile against uclibc
- Bug 54240 - [swrast] piglit fbo-generatemipmap-filtering regression
- Bug 55503 - Constant vertex attributes broken
- Bug 55783 - glEnable(GL_FRAMEBUFFER_SRGB) has no effect on the backbuffer
- Bug 55825 - [Bisected i965]Oglc max_values(advanced.fragmentProgram.GL_MAX_PROGRAM_ALU_INSTRUCTIONS_ARE causes OOM-killer
- Bug 56920 - [sandybridge][uxa] graphics very glitchy and always flickering
- Bug 57753 - leak in loop_analysis
- Bug 57875 - Second Life viewer bad rendering with git-ec83535

- Bug 58666 - rv670 + llvm = errors.
- Bug 58680 - [IVB] Graphical glitches in 0 A.D
- Bug 58872 - Mac OS X configure: error: Couldn't find clock_gettime
- Bug 59322 - r300g MSAA breaks Half-Life 2 in Wine
- Bug 59364 - [bisected] Mesa build fails: clientattrib.c:33:22: fatal error: indirect.h: No such file or directory
- Bug 59439 - glCopyPixels generates no fragments (occlusion_query_meta_fragments test fails)
- Bug 59440 - glBitmap generates no fragments (occlusion_query_meta_fragments test fails)
- Bug 59494 - [Bisected]Piglit glean_depthStencil fails
- Bug 59592 - Radeon HD 5670: reproducable GPU lockups with htile enabled
- Bug 59648 - [SNB/IVB/HSW Bisected]Piglit spec/ARB_uniform_buffer/object_layout-std140-base-size-and-alignment fails
- Bug 59701 - lp_test_arit fails on non-sse41 capable machines, breaking make check
- Bug 59737 - [bisected] 0d108116bd80b757fb01a84a9f1946ef870b57b8 breaks osmesa when cross compiling
- Bug 59740 - [i965 Bisected]Oglc api-error(negative.glEvalMesh) fails
- Bug 59851 - AC_ARG_WITH misusage leading to mesa configure failure
- Bug 59873 - [swrast] piglit ext_framebuffer_multisample-interpolation 0 centroid-edges regression
- Bug 59876 - glGetTexLevelParameteriv broken for indirect rendering
- Bug 60038 - [osmesa] [git] building 32-bit mesa on 64 bit fails
- Bug 60047 - [softpipe] piglit masked-clear regression
- Bug 60052 - [Bisected]Piglit glx_extension_string_sanity fail
- Bug 60082 - [FAILED] DispatchSanity_test.GL31_CORE
- Bug 60086 - Wayland platform backend crashes if there's no back buffer during dri2_swap_buffers
- Bug 60098 - [softpipe] Unexpected PIPE_CAP 78 query
- Bug 60172 - Planeshift: triangles where grass would be
- Bug 60200 - radeon_bo with virtual address referencing mismatch
- Bug 60212 - [Bisected] Weston black output
- Bug 60524 - [softpipe] piglit depthstencil-render-miplevels 146 s=z24_s8 regression
- Bug 60527 - [softpipe] fbo-stencil GL_DEPTH24_STENCIL8 clear regression
- Bug 60633 - EXT_texture_sRGB does not work in game The Cave on IvyBridge
- Bug 60737 - In GLSL ES, a missing FS precision qualifier does not generate an error
- Bug 60866 - GLSL performance issues for uniform buffer objects
- Bug 61036 - Shader fails to build in LLVMpipe, aborts program
- Bug 61200 - insufficient linking of libxatracker.so
- Bug 61635 - glVertexAttribPointer(id, GL_UNSIGNED_BYTE, GL_FALSE,...) does not work
- Bug 62466 - r600g hyperz lockups with KSP 0.19
- Bug 62669 - HyperZ freeze when playing PrBoom-Plus demo with lots of monsters

- Bug 62721 - GPU lockup in Minecraft 1.5.1 with HyperZ
- Bug 62830 - [i965 bisected] Wrong Lightning on Freespace 2 SCP (patch attached)
- Bug 63124 - [r600g] HyperZ lockup on REDWOOD in Half Life 2 Deathmatch
- Bug 63702 - tiling2d in radeon trash vdpau UVD textures
- Bug 64935 - [swrast] s_texfetch.c:1335: set_fetch_functions: Assertion 'texImage->FetchTexel' failed.
- Bug 64959 - Cannot build against EGL without X11
- Bug 65112 - glcpp hangs parsing line continuations
- Bug 65958 - GPU Lockup on Trinity 7500G
- Bug 66450 - JUNIPER UVD accelerated playback of MPEG 1/2 streams does not work
- Bug 66606 - [i965 bisected]GLBenchmark 2.5.1/2.7.0 sometimes render error with gnome-session enabling SNA
- Bug 66713 - Team Fortress 2 crashes with r600-sb on HD4850
- Bug 67354 - glsl_parser.cpp is broken with bison 3.0
- Bug 67548 - glGetAttribLocation seems to be broken
- Bug 67927 - R600_DEBUG=sb: Celestia show 2 earths, one wrongly rendered
- Bug 67934 - [SNB/IVB/HSW 9.2 Bisected]Ogles2conform/GL2Tests/glUniform/glUniform.test fails with gnome-session enable compositing
- Bug 68162 - [radeonsi] texture rendering is broken in Source-Engine games
- Bug 68195 - piglit tests vs-struct-pad and fs-struct-pad both fail

4.222.4 Changes

- Removed d3d1x state tracker (unused, unmaintained and broken)
- Removed GL_EXT_clip_volume_hint because no driver had enabled it since 2007.
- Removed GL_MESA_resize_buffers because it was only really implemented by the (unsupported) GDI driver.
- GL_EXT_separate_shader_objects has been removed from all Gallium drivers, because it disallows a critical GLSL shader optimization. GL_ARB_separate_shader_objects doesn't have this issue.
- i965 Gen6+ requires Kernel 3.6 or later. (92d2f5a)

4.223 Mesa 9.1.7 Release Notes / October 4, 2013

Mesa 9.1.7 is a bug fix release which fixes bugs found since the 9.1.6 release.

Mesa 9.1 implements the OpenGL 3.1 API, but the version reported by glGetString(GL_VERSION) or glGetIntegerv(GL_MAJOR_VERSION) / glGetIntegerv(GL_MINOR_VERSION) depends on the particular driver being used. Some drivers don't support all the features required in OpenGL 3.1. OpenGL 3.1 is **only** available if requested at context creation because GL_ARB_compatibility is not supported.

4.223.1 MD5 checksums

```
9e4abf7b7a6db762012c3c9917a8e8c7  MesaLib-9.1.7.tar.bz2
f1d4d479d6ce12b9566fdb379960a912  MesaLib-9.1.7.tar.gz
abd612bfc5dd478d04fcc630dd7672f2  MesaLib-9.1.7.zip
```

4.223.2 New features

None.

4.223.3 Bug fixes

This list is likely incomplete.

- Bug 55503 - Constant vertex attributes broken
- Bug 61635 - glVertexAttribPointer(id, GL_UNSIGNED_BYTE, GL_FALSE,...) does not work
- Bug 65958 - GPU Lockup on Trinity 7500G
- Bug 66292 - [SNB/IVB/HSW Bisected]Ogles3conform GL3Tests_depth24_depth24_basic.test fail
- Bug 67548 - glGetAttribLocation seems to be broken
- Bug 68195 - piglit tests vs-struct-pad and fs-struct-pad both fail
- Bug 68250 - Automatic mipmap generation with texture compression produces borders that fade to black
- Bug 69525 - [GM45, bisected] Piglit tex-shadow2drect fails

4.223.4 Changes

Alex Deucher (2):

- r600g: disable GPUVM by default
- radeon/winsys: pad IBs to a multiple of 8 DWs

Andreas Boll (2):

- docs: Fix a typo in the 9.1.6 release notes
- mesa: Fix MESA_PATCH version

Anuj Phogat (1):

- meta: Fix blitting a framebuffer with renderbuffer attachment

Carl Worth (5):

- docs: Add 9.1.6 release md5sums
- Use -Bsymbolic when linking libEGL.so
- Update get-pick-list to look specifically for "9.1" in NOTE
- cherry-ignore: Ignore last two patches in current get-pick-list output
- Bump version to 9.1.7

Chris Forbes (1):

- i965/fs: Gen4: Zero out extra coordinates when using shadow compare

Emil Velikov (2):

- nv50: handle pure integer vertex attributes
- nouveau: initialise the nouveau_transfer maps

Eric Anholt (1):

- i965/gen4: Fix fragment program rectangle texture shadow compares.

Ian Romanick (11):

- mesa: Remove stray debug printfs in attachment completeness code
- mesa: Validate the layer selection of an array texture too
- mesa/vbo: Fix handling of attribute 0 in non-compatibilty contexts
- glsl: Add new overload of program_resource_visitor::visit_field method
- glsl: Use alignment of container record for its first field
- mesa: Remove all traces of GL_OES_matrix_get
- mesa: Don't call driver RenderTexture for really broken textures
- mesa: Don't call driver RenderTexture for invalid zoffset
- mesa: Generate a renderbuffer wrapper even if the texture has no image
- glsl: Move and refine test for unsized arrays in GLSL ES
- mesa: Don't return any data for GL_SHADER_BINARY_FORMATS

Ilia Mirkin (2):

- nv30: U8_USCALED only works for size 4
- nv30: remove no-longer-used formats from table

Joakim Sindholt (1):

- nvc0: fix blitctx memory leak

Johannes Obermayr (1):

- st/gbm: Add $(WAYLAND_CFLAGS) for HAVE_EGL_PLATFORM_WAYLAND.

Kenneth Graunke (1):

- meta: Set correct viewport and projection in decompress_texture_image.

Maarten Lankhorst (1):

- nvc0: restore viewport after blit

Rico Schüller (1):

- glx: Initialize OpenGL version to 1.0

Tiziano Bacocco (1):

- nvc0/ir: fix use after free in texture barrier insertion pass

Torsten Duwe (1):

- wayland-egl.pc requires wayland-client.pc.

4.224 Mesa 9.1.6 Release Notes / August 1, 2013

Mesa 9.1.6 is a bug fix release which fixes bugs found since the 9.1.5 release.

Mesa 9.1 implements the OpenGL 3.1 API, but the version reported by glGetString(GL_VERSION) or glGetIntegerv(GL_MAJOR_VERSION) / glGetIntegerv(GL_MINOR_VERSION) depends on the particular driver being used. Some drivers don't support all the features required in OpenGL 3.1. OpenGL 3.1 is **only** available if requested at context creation because GL_ARB_compatibility is not supported.

4.224.1 MD5 checksums

```
443a2a352667294b53d56cb1a74114e9  MesaLib-9.1.6.tar.bz2
08d3069cccd6821e5f33e0840bca0718  MesaLib-9.1.6.tar.gz
90aa7a6d9878cdbfcb055312f356d6b9  MesaLib-9.1.6.zip
```

4.224.2 New features

None.

4.224.3 Bug fixes

This list is likely incomplete.

- Bug 47824 - osmesa using –enable-shared-glapi depends on libgl
- Bug 62362 - Crash when using Wayland EGL platform
- Bug 63435 - [Regression since 9.0] Flickering in EGL OpenGL full-screen window with swap interval 1
- Bug 64087 - Webgl conformance shader-with-non-reserved-words crash when mesa is compiled without –enable-debug
- Bug 64330 - WebGL snake demo crash in loop_analysis.cpp:506: bool is_loop_terminator(ir_if*): assertion „inst != __null“ failed.
- Bug 65236 - [i965] Rendering artifacts in VDrift/GL2
- Bug 66558 - RS690: 3D artifacts when playing SuperTuxKart
- Bug 66847 - compilation broken with llvm 3.3
- Bug 66850 - glGenerateMipmap crashes when using GL_TEXTURE_2D_ARRAY with compressed internal format
- Bug 66921 - [r300g] Heroes of Newerth: HiZ related corruption
- Bug 67283 - VDPAU doesn't work on hybrid laptop through DRI_PRIME

4.224.4 Changes

The full set of changes can be viewed by using the following GIT command:

```
git log mesa-9.1.5..mesa-9.1.6
```

Andreas Boll (1):

- configure.ac: Require llvm-3.2 for r600g/radeonsi llvm backends

Brian Paul (4):

- mesa: handle 2D texture arrays in get_tex_rgba_compressed()
- meta: handle 2D texture arrays in decompress_texture_image()
- mesa: implement mipmap generation for compressed 2D array textures
- mesa: improve free() cleanup in generate_mipmap_compressed()

Carl Worth (7):

- docs: Add 9.1.5 release md5sums
- Merge 'origin/9.1' into stable
- cherry-ignore: Drop 13 patches from the pick list
- get-pick-list.sh: Include commits mentionining "CC: mesa-stable..." in pick list
- get-pick-list: Allow for non-whitespace between "CC:" and "mesa-stable"
- get-pick-list: Ignore commits which CC mesa-stable unless they say "9.1"
- Bump version to 9.1.6

Chris Forbes (5):

- i965/Gen4: Zero extra coordinates for ir_tex
- i965/vs: Fix flaky texture swizzling
- i965/vs: set up sampler state pointer for Gen4/5.
- i965/vs: Put lod parameter in the correct place for Gen4
- i965/vs: Gen4/5: enable front colors if back colors are written

Christoph Bumiller (1):

- nv50,nvc0: s/uint16/uint32 for constant buffer offset

Dave Airlie (1):

- gallium/vl: add prime support

Eric Anholt (1):

- egl: Restore "bogus" DRI2 invalidate event code.

Jeremy Huddleston Sequoia (1):

- Apple: glFlush() is not needed with CGLFlushDrawable()

Kenneth Graunke (1):

- glsl: Classify "layout" like other identifiers.

Kristian Høgsberg (1):

- egl-wayland: Fix left-over wl_display_roundtrip() usage

Maarten Lankhorst (2):

- osmesa: link against static libglapi library too to get the gl exports
- nvc0: force use of correct firmware file

Marek Olšák (4):

- r300g/swtcl: fix geometry corruption by uploading indices to a buffer
- r300g/swtcl: fix a lockup in MSAA resolve
- Revert "r300g: allow HiZ with a 16-bit zbuffer"
- r600g: increase array size for shader inputs and outputs

Matt Turner (2):

- i965: NULL check prog on shader compilation failure.
- i965/vs: Print error if vertex shader fails to compile.

Paul Berry (1):

- glsl: Handle empty if statement encountered during loop analysis.

4.225 Mesa 9.1.5 Release Notes / July 17, 2013

Mesa 9.1.5 is a bug fix release which fixes bugs found since the 9.1.4 release.

Mesa 9.1 implements the OpenGL 3.1 API, but the version reported by glGetString(GL_VERSION) or glGetIntegerv(GL_MAJOR_VERSION) / glGetIntegerv(GL_MINOR_VERSION) depends on the particular driver being used. Some drivers don't support all the features required in OpenGL 3.1. OpenGL 3.1 is **only** available if requested at context creation because GL_ARB_compatibility is not supported.

4.225.1 MD5 checksums

```
4ed2af5943141a85a21869053a2fc2eb  MesaLib-9.1.5.tar.bz2
47181066acf3231d74e027b2033f9455  MesaLib-9.1.5.tar.gz
4c9c6615bd99215325250f87ed34058f  MesaLib-9.1.5.zip
```

4.225.2 New features

None.

4.225.3 Bug fixes

This list is likely incomplete.

- Bug 58384 - [i965 Bisected]Oglc max_values(advanced.fragmentProgram.GL_MAX_PROGRAM_ENV_PARAMETERS_ARB) segfault
- Bug 62647 - Wrong rendering of Dota 2 on Wine (apitrace attached) - Intel IVB HD4000
- Bug 63674 - [IVB]frozen at the first frame when run Unigine-heaven 4.0
- Bug 65910 - Killing weston-launch causes segv in desktop-shell

4.225.4 Changes

The full set of changes can be viewed by using the following GIT command:

```
git log mesa-9.1.4..mesa-9.1.5
```

Anuj Phogat (1):

- mesa: Return ZeroVec/dummyReg instead of NULL pointer

Brian Paul (1):

- svga: check for NaN shader immediates

Carl Worth (3):

- cherry-ignore: Ignore previously backported patch
- cherry-ignore: Drop two patches which we've decided not to include
- mesa: Bump version to 9.1.5

Chris Forbes (1):

- i965: fix alpha test for MRT

Christoph Bumiller (1):

- r600g: x/y coordinates must be divided by block dim in dma blit

Eric Anholt (1):

- ra: Fix register spilling.

Ian Romanick (6):

- docs: Add 9.1.4 release md5sums
- glsl: Add a gl_shader_program parameter to _mesa_uniform_{merge,split}_location_offset
- glsl: Add gl_shader_program::UniformLocationBaseScale
- glsl: Generate smaller values for uniform locations
- i965: Be more careful with the interleaved user array upload optimization
- glsl: Move all var decls to the front of the IR list in reverse order

Kenneth Graunke (1):

- glsl/builtins: Fix ARB_texture_cube_map_array built-in availability.

Kristian Høgsberg (1):

- wayland: Handle global_remove event as well

Matt Turner (1):

- register_allocate: Fix the type of best_benefit.

Paul Berry (1):

- glsl ES: Fix magnitude of gl_MaxVertexUniformVectors.

Richard Sandiford (3):

- st/xlib Fix XIMage bytes-per-pixel calculation
- st/xlib: Fix XImage stride calculation

- st/dri/sw: Fix pitch calculation in drisw_update_tex_buffer

Vinson Lee (1):

- swrast: Fix memory leak.

4.226 Mesa 9.1.4 Release Notes / July 1st, 2013

Mesa 9.1.4 is a bug fix release which fixes bugs found since the 9.1.3 release.

Mesa 9.1 implements the OpenGL 3.1 API, but the version reported by glGetString(GL_VERSION) or glGetIntegerv(GL_MAJOR_VERSION) / glGetIntegerv(GL_MINOR_VERSION) depends on the particular driver being used. Some drivers don't support all the features required in OpenGL 3.1. OpenGL 3.1 is **only** available if requested at context creation because GL_ARB_compatibility is not supported.

4.226.1 MD5 checksums

```
a2c4e25d0e27918bc67f61bae04d0cb8  MesaLib-9.1.4.tar.bz2
8c7e9ce5b05cb2223f0587396dd9dc08  MesaLib-9.1.4.tar.gz
020459c5793d4279bdcb2daa1f7dd9f6  MesaLib-9.1.4.zip
```

4.226.2 New features

None.

4.226.3 Bug fixes

This list is likely incomplete.

- Bug 37871 - [bisected i965] Bus error (core dumped) on oglc texdecaltile
- Bug 42182 - egl/opengles1/tri_x11 renders wrong
- Bug 44958 - [SNB IVB HSW] mesa demo test texleak bus error
- Bug 53494 - [snb] crash in texsubimage to a large atlas in clutter
- Bug 60518 - glDrawElements segfault when compiled into display list
- Bug 61821 - src/mesa/drivers/dri/common/xmlpool.h:96:29: fatal error: xmlpool/options.h
- Bug 63520 - r300g regression (RV380): Strange rendering of light sources in Penumbra (bisected)
- Bug 63701 - [HSW] support new haswell graphics [8086:0a2e]
- Bug 64727 - [gm45, bisected] some piglit glsl 1.10 built-in-functions tests crash
- Bug 64745 - [llvmpipe] SIGSEGV src/gallium/state_trackers/glx/xlib/glx_api.c:1374
- Bug 64934 - [llvmpipe] SIGSEGV src/gallium/state_trackers/glx/xlib/glx_api.c:1363
- Bug 65173 - segfault in _mesa_get_format_datatype and _mesa_get_color_read_type when state dumping with glretrace

4.226.4 Changes

The full set of changes can be viewed by using the following GIT command:

```
git log mesa-9.1.3..mesa-9.1.4
```

Alan Coopersmith (2):

- integer overflow in XF86DRIOpenConnection() [CVE-2013-1993 1/2]
- integer overflow in XF86DRIGetClientDriverName() [CVE-2013-1993 2/2]

Alex Deucher (3):

- radeonsi: add support for hainan chips
- radeonsi: add Hainan pci ids
- winsys/radeon: add env var to disable VM on Cayman/Trinity

pp

Andreas Boll (1):

- glapi: Add some missing static_dispatch="false" annotations to es_EXT.xml

Anuj Phogat (1):

- intel: Add a null pointer check before dereferencing the pointer

Armin K (1):

- gallivm: Fix build with LLVM 3.3

Brian Paul (9):

- mesa: fix the compressed TexSubImage size checking code
- st/mesa: generate GL_OUT_OF_MEMORY if we can't create the index buffer
- mesa: fix error checking of DXT sRGB formats in _mesa_base_tex_format()
- st/glx/xlib: check for null ctx pointer in glXIsDirect()
- xlib: check for null ctx pointer in glXIsDirect()
- st/glx: add null ctx check in glXDestroyContext()
- xlib: add null ctx check in glXDestroyContext()
- meta: move vertex array enables for mipmap generation
- mesa: handle missing read buffer in _mesa_get_color_read_format/type()

Bryan Cain (1):

- nv50: initialize kick_notify callback in nv50_create

Chad Versace (3):

- egl/android: Fix error condition for EGL_ANDROID_image_native_buffer
- i965: Fix glColorPointer(GL_FIXED)
- intel: Return early if miptree allocation fails

Chia-I Wu (1):

- u_vbuf: fix index buffer leak

Chris Forbes (8):

- mesa: add accessor for effective stencil ref
- intel: Use accessor for stencil reference values
- nouveau: Use accessor for stencil reference values
- radeon: Use accessor for stencil reference values
- st: Use accessor for stencil reference values
- swrast: Use accessor for stencil reference values
- mesa: Stop clamping stencil reference value at specification time
- mesa: Use accessor for stencil reference values in glGet

Chí-Thanh Christopher Nguyn (1):

- targets/dri-i915: Force c++ linker in all cases

Daniel Martin (1):

- Fix build of swrast only without libdrm

Dave Airlie (1):

- i965: fix problem with constant out of bounds access (v3)

Eric Anholt (10):

- mesa: Make core Mesa allocate the texture renderbuffer wrapper.
- mesa: Make gl_renderbuffers backed by EGL images use FinishRenderTexture.
- i965/fs: Bake regs_written into the IR instead of recomputing it later.
- i965/vs: Fix implied_mrf_writes() for integer division pre-gen6.
- intel: Add support for writing to our linear-temporary-CPU-map case.
- intel: Do temporary CPU maps of textures that are too big to GTT map.
- intel: Avoid making tiled miptrees we won't be able to blit.
- intel: Fix MRT handling of glBitmap().
- intel: Fix format handling of blit glBitmap()
- i965: Shut up the last release build warning.

Fabian Bieler (2):

- mesa/st: Don't copy propagate from swizzles.
- mesa/program: Don't copy propagate from swizzles.

Frank Henigman (1):

- intel: initialize fs_visitor::params_remap in constructor

Ian Romanick (2):

- docs: Add 9.1.3 release md5sums
- mesa: Bump version to 9.1.4

José Fonseca (1):

- scons: Fix implicit python dependency discovery on Windows.

Kenneth Graunke (17):

- mesa: Add i965 varying index patches to .cherry-ignore.
- i965: Turn brw->urb.vs_size and gs_size into local variables.
- i965: Use a variable for the push constant size in kB.
- i965: Update URB partitioning code for Haswell's GT3 variant.
- i965: Add chipset limits for the Haswell GT3 variant.
- i965: Enable the Bay Trail platform.
- mesa: Add a reverted commit to cherry-ignore.
- vbo: Ignore PRIMITIVE_RESTART_FIXED_INDEX for glDrawArrays().
- mesa: Add a helper function for determining the restart index.
- vbo: Use the new primitive restart index helper function.
- i965: Use the correct restart index for fixed index mode on Haswell.
- mesa: Cherry-ignore a patch that got picked but squashed.
- i965: Fix can_cut_index_handle_restart_index() for byte/short types.
- st/mesa: Go back to using ctx->Array.RestartIndex, not _RestartIndex.
- mesa: Ignore fixed-index primitive restart in ArrayElement().
- mesa: Delete the ctx->Array._RestartIndex derived state.
- glsl: Bail on parsing if the #version directive is bogus.

Lauri Kasanen (1):

- r600g: Correctly initialize the shader key, v2

Maarten Lankhorst (4):

- nvc0: fix up video buffer alignment requirements
- nvc0: kill assert in ppp code
- nvc0: set rsvd_kick correctly
- nvc0: allow frame dropping in h264

Marek Olšák (7):

- radeonsi: increase array size for shader inputs and outputs
- vbo: fix possible use-after-free segfault after a VAO is deleted
- glsl: fix the value of gl_MaxFragmentUniformVectors
- st/mesa: initialize all program constants and UBO limits
- st/mesa: initialize Const.MaxColorAttachments
- st/mesa: fix a couple of issues in st_bind_ubos
- mesa: declare UniformBufferBindings as an array with a static size

Matt Turner (3):

- configure.ac: Remove redundant checks of enable_dri.
- configure.ac: Build dricommon for DRI gallium drivers

- i965: NULL check depth_mt to quiet static analysis.

Michel Dänzer (3):

- radeonsi: Fix handling of TGSI_SEMANTIC_PSIZE
- radeonsi: Fix user clip planes
- mesa: Note that two radeonsi fixes cannot be backported after all

Mike Stroyan (1):

- configure.ac: Build dricommon for gallium swrast

Naohiro Aota (1):

- xmlpool/build: Make sure to set mo properly

Paul Berry (2):

- glsl: Fix error checking on "flat" keyword to match GLSL ES 3.00, GLSL 1.50.
- i965/gen7.5: Allow HW primitive restart for all primitive types.

Paulo Zanoni (1):

- i965: make GT3 machines work as GT3 instead of GT2

Rodrigo Vivi (2):

- i965: Add missing Haswell GT3 Desktop to IS_HSW_GT3 check.
- i965: Adding more reserved PCI IDs for Haswell.

Roland Scheidegger (1):

- gallivm: fix out-of-bounds access with mirror_clamp_to_edge address mode

Stéphane Marchesin (2):

- st/xlib: Fix upside down coordinates for CopySubBuffer
- st/xlib: Flush the front buffer before doing CopySubBuffer

Sven Joachim (1):

- mesa: Fix ieee fp on Alpha

Tapani Pälli (1):

- mesa: fix type comparison errors in sub-texture error checking code

Tom Stellard (2):

- gallivm: Fix build with LLVM >= r180063
- r300g/compiler: Prevent regalloc from swizzling texture operands v2

Vinson Lee (1):

- radeon: Initialize variables in radeon_llvm_context_init.

4.227 Mesa 9.1.3 Release Notes / May 21st, 2013

Mesa 9.1.3 is a bug fix release which fixes bugs found since the 9.1.1 release.

Mesa 9.1 implements the OpenGL 3.1 API, but the version reported by glGetString(GL_VERSION) or glGetIntegerv(GL_MAJOR_VERSION) / glGetIntegerv(GL_MINOR_VERSION) depends on the particular driver being used.

Some drivers don't support all the features required in OpenGL 3.1. OpenGL 3.1 is **only** available if requested at context creation because GL_ARB_compatibility is not supported.

4.227.1 MD5 checksums

```
952ccd03547ed72333b64e1746cf8ada  MesaLib-9.1.3.tar.bz2
26d2f1aa8e9db388d51fcbd163c61fb7  MesaLib-9.1.3.tar.gz
7017b7bdf0ebfd39a5c46cee7cf6b567  MesaLib-9.1.3.zip
```

4.227.2 New features

None.

4.227.3 Bug fixes

This list is likely incomplete.

- Bug 39251 - Second Life viewers from release 2.7.4.235167 to the last 3.4.0.264911 crash on start.
- Bug 47478 - [wine] GLX_DONT_CARE does not work for GLX_DRAWABLE_TYPE or GLX_RENDER_TYPE
- Bug 56416 - [SNB bisected] SNB hang with rc6 and hiz on glxgears (and other GL apps) immediately after xinit.
- Bug 57436 - [GLSL1.40 IVB/HSW]Piglit spec/glsl-1.40/compiler_built-in-functions/inverse-mat2.frag fails
- Bug 61554 - [ivb] Mesa 9.1 performance regression on KWin's Lanczos shader
- Bug 61773 - abort is an incredibly not-smart way to handle IR validation
- Bug 62868 - solaris build broken with missing ffsll
- Bug 62999 - glXChooseFBConfig with GLX_DRAWABLE_TYPE, GLX_DONT_CARE fails
- Bug 63078 - EGL X11 Regression: Maximum swap interval is 0 (worked with 9.0)
- Bug 63447 - [i965 Bisected]Ogles1conform/Ogles2conform/Ogles3conform cases segfault
- Bug 64662 - [SNB 9.1 Bisected]Ogles2conform GL2ExtensionTests/depth_texture_cube_map/depth_texture_cube_map.test fail

4.227.4 Changes

The full set of changes can be viewed by using the following GIT command:

```
git log mesa-9.1.2..mesa-9.1.3
```

Alex Deucher (2):

- r600g: add new richland pci ids
- radeonsi: add new SI pci ids

Alexander Monakov (1):

- Honor GLX_DONT_CARE in MATCH_MASK

Andreas Boll (2):

- mesa: Add a script to generate the list of fixed bugs
- mesa: add usage examples to get-pick-list and shortlog scripts

Aras Pranckevicius (1):

- GLSL: fix lower_jumps to report progress properly

Brian Paul (3):

- mesa: remove platform checks around __builtin_ffs, __builtin_ffsll
- gallium/u_blitter: fix is_blit_generic_supported() stencil checking
- mesa: enable GL_ARB_texture_float if TEXTURE_FLOAT_ENABLED is defined

Chad Versace (2):

- egl/dri2: Fix min/max swap interval of configs
- intel: Allocate hiz in intel_renderbuffer_move_to_temp()

Chris Forbes (2):

- i965/fs: Don't try to use bogus interpolation modes pre-Gen6.
- mesa: don't memcmp() off the end of a cache key.

Dave Airlie (2):

- st/mesa: fix UBO offsets.
- ralloc: don't write to memory in case of alloc fail.

Eric Anholt (11):

- i965/fs: Remove creation of a MOV instruction that's never used.
- i965/fs: Move varying uniform offset compuation into the helper func.
- i965: Make the constant surface interface take a normal byte size.
- i965/fs: Avoid inappropriate optimization with regs_written > 1.
- i965/fs: Do CSE on gen7's varying-index pull constant loads.
- i965/fs: Clean up the setup of gen4 simd16 message destinations.
- i965/gen7: Skip resetting SOL offsets at batch start with HW contexts.
- i965/gen6: Reduce updates of transform feedback offsets with HW contexts.
- i965: Fix SNB GPU hangs when a blorp batch is the first thing to execute.
- i965: Fix hangs on HSW since the gen6 blorp fix.
- i965: Disable write masking when setting up texturing m0.

Haixia Shi (1):

- ACTIVE_UNIFORM_MAX_LENGTH should include 3 extra characters for arrays.

Ian Romanick (11):

- docs: Add 9.1.2 release md5sums
- mesa: Note that patch 0967c36 shouldn't actually get picked to the 9.1 branch
- mesa: NULL check the pointer before trying to dereference it

- egl/dri2: NULL check value returned by dri2_create_surface
- mesa: Don't leak shared state when context initialization fails
- mesa: Don't leak gl_context::BeginEnd at context destruction
- mesa/swrast: Refactor no-memory error checking in blit_linear
- mesa/swrast: Move free calls outside the attachment loop
- intel: Don't dereference a NULL pointer of calloc fails
- mesa: Note that a824692 is already back ported
- mesa: Bump version to 9.1.3

José Fonseca (1):

- winsys/sw/xlib: Prevent shared memory segment leakage.

Kenneth Graunke (9):

- mesa: Add new ctx->Stencil._WriteEnabled derived state flag.
- i965: Fix stencil write enable flag in 3DSTATE_DEPTH_BUFFER on Gen7+.
- mesa: Fix unpack function for ETC2_SRGB8_PUNCHTHROUGH_ALPHA1.
- mesa: Add an unpack function for ARGB2101010_UINT.
- mesa: Add unpack functions for R/RG/RGB [U]INT8/16/32 formats.
- mesa: Add unpack functions for A/I/L/LA [U]INT8/16/32 formats.
- glsl: Ignore redundant prototypes after a function's been defined.
- i965: Lower textureGrad() for samplerCubeShadow.
- i965/vs: Fix textureGrad() with shadow samplers on Haswell.

Maarten Lankhorst (1):

- nvc0: Fix fd leak in nvc0_create_decoder

Marek Olšák (5):

- radeonsi: add more cases for copying unsupported formats to resource_copy_region
- mesa: fix glGet queries depending on derived framebuffer state (v2)
- gallium/u_blitter: implement buffer clearing
- r600g: initialize CMASK and HTILE with the GPU using streamout
- st/mesa: depth-stencil-alpha state also depends on _NEW_BUFFERS

Martin Andersson (1):

- r600g: Fix UMAD on Cayman

Michel Dänzer (1):

- radeonsi: Handle arbitrary 2-byte formats in resource_copy_region

Paul Berry (7):

- glsl: Fix array indexing when constant folding built-in functions.
- i965: Reduce code duplication in handling of depth, stencil, and HiZ.
- glsl/linker: fix varying packing for non-flat integer varyings.

- glsl: Document lower_packed_varyings' "flat" requirement with an assert.
- glsl/linker: Adapt flat varying handling in preparation for geometry shaders.
- glsl/linker: Reduce scope of non-flat integer varying fix.
- intel: Do a depth resolve before copying images between miptrees.

Ralf Jung (1):

- egl/x11: Fix initialisation of swap_interval

Roland Scheidegger (1):

- gallivm: fix small but severe bug in handling multiple lod level strides

Vadim Girlin (1):

- gallium: handle drirc disable_glsl_line_continuations option

4.228 Mesa 9.1.2 Release Notes / April 30th, 2013

Mesa 9.1.2 is a bug fix release which fixes bugs found since the 9.1.1 release.

Mesa 9.1 implements the OpenGL 3.1 API, but the version reported by glGetString(GL_VERSION) or glGetIntegerv(GL_MAJOR_VERSION) / glGetIntegerv(GL_MINOR_VERSION) depends on the particular driver being used. Some drivers don't support all the features required in OpenGL 3.1. OpenGL 3.1 is **only** available if requested at context creation because GL_ARB_compatibility is not supported.

4.228.1 MD5 checksums

```
df2aab86ff4a510ce5b0d074caa0a59f  MesaLib-9.1.2.tar.bz2
415c2bc3a9eb571aafbfa474ebf5a2e0  MesaLib-9.1.2.tar.gz
b1ae5a4d9255953980bc9254f5323420  MesaLib-9.1.2.zip
```

4.228.2 New features

None.

4.228.3 Bug fixes

This list is likely incomplete.

- Bug 44567 - [965gm] green artifacts when using GLSL in XBMC
- Bug 59238 - many new symbols in libxatracker after recent automake work
- Bug 59445 - [SNB/IVB/HSW Bisected]Oglc draw-buffers2(advanced.blending.none) segfault
- Bug 59495 - [i965 Bisected]Oglc fbblit(advanced.blitFb-3d-cube.mirror.both) fails
- Bug 60503 - [r300g] Unigine Heaven 3.0: all objects are black
- Bug 60510 - Firefox 18.0.2 Crash On Nvidia GeForce2
- Bug 61197 - [SNB Bisected] kwin_gles screen corruption
- Bug 61317 - [IVB] corrupt rendering with UBOs

- Bug 61395 - glEdgeFlag can't be set to false
- Bug 61947 - nullpointer dereference causes xorg-server segfault when nouveau DRI driver is loaded
- Bug 62357 - llvmpipe: Fragment Shader with "return" in main causes back output
- Bug 62434 - [bisected] 3284.073] (EE) AIGLX error: dlopen of /usr/lib/xorg/modules/dri/r600_dri.so failed (/usr/lib/libllvmradeon9.2.0.so: undefined symbol: lp_build_tgsi_intrinsic)
- Debian bug #349437 - mesa - FTBFS: error: 'IEEE_ONE' undeclared
- Redhat bug #918661 - crash in routine Avogadro UI manipulation

4.228.4 Changes

The full set of changes can be viewed by using the following GIT command:

```
git log mesa-9.1.1..mesa-9.1.2
```

Adam Jackson (2):

- glx: Build with VISIBILITY_CFLAGS in automake
- linux: Don't emit a .note.ABI-tag section anymore (#26663)

Alan Hourihane (3):

- Add missing GL_TEXTURE_CUBE_MAP entry in _mesa_legal_texture_dimensions
- Unreference sampler object when it's currently bound to texture unit.
- mesa: fix glGetInteger*(GL_SAMPLER_BINDING).

Alex Deucher (1):

- r600g: disable hyperz by default on 9.1

Andreas Boll (5):

- radeon/llvm: Link against libgallium.la to fix an undefined symbol
- mesa: use ieee fp on s390 and m68k
- build: Enable x86 assembler on Hurd.
- osmesa: fix out-of-tree build
- gallium/egl: fix out-of-tree build

Anuj Phogat (1):

- mesa: Fix FB blitting in case of zero size src or dst rect

Brian Paul (4):

- mesa: flush current state when querying GL_EDGE_FLAG
- vbo: fix crash found with shared display lists
- llvmpipe: tweak CMD_BLOCK_MAX and LP_SCENE_MAX_SIZE
- llvmpipe: add some scene limit sanity check assertions

Carl Worth (1):

- i965: Avoid segfault in gen6_upload_state

Chris Forbes (1):

- i965/vs: Fix Gen4/5 VUE map inconsistency with gl_ClipVertex

Christoph Bumiller (4):

- nv50: fix 3D render target setup
- nv50,nvc0: disable DEPTH_RANGE_NEAR/FAR clipping during blit
- nv50,nvc0: fix 3d blits, restore viewport after blit
- nvc0: fix for 2d engine R source formats writing RRR1 and not R001

Eric Anholt (5):

- i965/fs: Fix register allocation for uniform pull constants in 16-wide.
- i965/fs: Fix broken rendering in large shaders with UBO loads.
- i965/fs: Also do the gen4 SEND dependency workaround against other SENDs.
- i965: Add definitions for gen7+ data cache messages.
- mesa: Disable validate_ir_tree() on release builds.

Ian Romanick (5):

- docs: Add 9.1.1 release md5sums
- mesa: Add previously picked commit to .cherry-ignore
- glsl: Add missing bool case in glsl_type::get_scalar_type
- mesa: Note that patch dbf94d1 should't actually get picked to the 9.1 branch
- mesa: Bump version to 9.1.2

Jan de Groot (1):

- dri/nouveau: fix crash in nouveau_flush

José Fonseca (3):

- autotools: Add missing top-level include dir.
- mesa,gallium,egl,mapi: One definition of C99 inline/__func__ to rule them all.
- include: Fix build with VS 11 (i.e, 2012).

Kenneth Graunke (4):

- i965: Fix INTEL_DEBUG=shader_time for Haswell.
- i965: Specialize SURFACE_STATE creation for shader time.
- i965: Make INTEL_DEBUG=shader_time use the RAW surface format.
- i965: Don't use texture swizzling to force alpha to 1.0 if unnecessary.

Maarten Lankhorst (2):

- gallium/build: Fix visibility CFLAGS in automake
- radeon/llvm: Do not link against libgallium when building statically.

Marcin Slusarz (1):

- dri/nouveau: NV17_3D class is not available for NV1a chipset

Marek Olšák (3):

- mesa: don't allocate a texture if width or height is 0 in CopyTexImage

- gallium/tgsi: fix valgrind warning
- mesa: handle HALF_FLOAT like FLOAT in get_tex_rgba

Martin Andersson (1):

- r600g: Use virtual address for PIPE_QUERY_SO* in r600_emit_query_end

Matt Turner (3):

- configure.ac: Don't check for X11 unconditionally.
- configure.ac: Remove stale comment about –x-* arguments.
- mesa: Implement TEXTURE_IMMUTABLE_LEVELS for ES 3.0.

Michel Dänzer (1):

- radeonsi: Emit pixel shader state even when only the vertex shader changed

Paul Berry (1):

- i965: Apply depthstencil alignment workaround when doing fast clears.

Roland Scheidegger (1):

- gallivm: fix return opcode handling in main function of a shader

Tapani Pälli (1):

- intel: Fix regression in intel_create_image_from_name stride handling

Tom Stellard (1):

- r300g: Fix bug in OMOD optimization

4.229 Mesa 9.1.1 Release Notes / March 19th, 2013

Mesa 9.1.1 is a bug fix release which fixes bugs found since the 9.1 release.

Mesa 9.1 implements the OpenGL 3.1 API, but the version reported by glGetString(GL_VERSION) or glGetIntegerv(GL_MAJOR_VERSION) / glGetIntegerv(GL_MINOR_VERSION) depends on the particular driver being used. Some drivers don't support all the features required in OpenGL 3.1. OpenGL 3.1 is **only** available if requested at context creation because GL_ARB_compatibility is not supported.

4.229.1 MD5 checksums

```
6508d9882d8dce7106717f365632700c  MesaLib-9.1.1.tar.gz
6ea2bdc3b7ecfb4257b39814b4182580  MesaLib-9.1.1.tar.bz2
3434c0eb47849a08c53cd32833d10d13  MesaLib-9.1.1.zip
```

4.229.2 New features

None.

4.229.3 Bug fixes

This list is likely incomplete.

- Bug 30232 - [GM45] mesa demos spriteblast render incorrectly
- Bug 32429 - [gles2] Ironlake: gl_PointCoord takes no effect for point sprites
- Bug 38086 - Mesa 7.11-devel implementation error: Unexpected program target in destroy_program_variants_cb()
- Bug 57121 - [snb] corrupted GLSL built-in function results when using Uniform Buffer contents as arguments
- Bug 58042 - [bisected] Garbled UI in Team Fortress 2 and Counter-Strike: Source
- Bug 58960 - Texture flicker with fragment shader
- Bug 59495 - [i965 Bisected]Oglc fbblit(advanced.blitFb-3d-cube.mirror.both) fails
- Bug 59783 - [IVB bisected] 3DMMES2.0 Taiji performance reduced by ~13% with gnome-session enable compositing
- Bug 60121 - build - libvdpau_softpipe fails at runtime.
- Bug 60143 - gbm_dri_bo_create fails to initialize bo->base.base.format
- Bug 60802 - Corruption with DMA ring on cayman
- Bug 60848 - [bisected] r600g: add htile support cause gpu lockup in Dishonored wine.
- Bug 60938 - [softpipe] piglit interpolation-noperspective-gl_BackColor-flat-fixed regression
- Bug 61012 - alloc_layout_array tx * ty assertion failure when making pbuffer current
- Bug 61026 - Segfault in glBitmap when called with PBO source

4.229.4 Changes

The full set of changes can be viewed by using the following GIT command:

```
git log mesa-9.1..mesa-9.1.1
```

Adam Sampson (1):

- autotools: oprofilejit should be included in the list of LLVM components required

Alex Deucher (2):

- r600g: add Richland APU pci ids
- r600g: Use blitter rather than DMA for 128bpp on cayman (v3)

Andreas Boll (2):

- docs: Add 9.1 release md5sums
- docs: add news item for 9.1 release

Anuj Phogat (1):

- meta: Allocate texture before initializing texture coordinates

Brian Paul (11):

- docs: remove stray 'date' text
- docs: insert links to the 9.0.3 release

- draw: fix non-perspective interpolation in interp()
- st/mesa: implement glBitmap unpacking from a PBO, for the cache path
- st/xlib: initialize the drawable size in create_xmesa_buffer()
- st/mesa: fix trimming of GL_QUAD_STRIP
- st/mesa: check for dummy programs in destroy_program_variants()
- st/mesa: fix polygon offset state translation logic
- draw: fix broken polygon offset stage
- llvmpipe: add missing checks for polygon offset point/line modes
- svga: always link with C++

Daniel van Vugt (1):

- gbm: Remember to init format on gbm_dri_bo_create.

Eric Anholt (7):

- i965/fs: Do a general SEND dependency workaround for the original 965.
- i965/fs: Fix copy propagation with smearing.
- i965/fs: Delay setup of uniform loads until after pre-regalloc scheduling.
- i965/fs: Only do CSE when the dst types match.
- i965/fs: Fix broken math on values loaded from uniform buffers on gen6.
- mesa: Fix setup of ctx->Point.PointSprite for GLES2.
- i965: Fix the W value of deprecated pointcoords on pre-gen6.

Frank Henigman (1):

- i965: Link i965_dri.so with C++ linker.

Ian Romanick (3):

- mesa: Add previously picked commit to .cherry-ignore
- mesa: Modify candidate search string
- egl: Allow 24-bit visuals for 32-bit RGBA8888 configs

Jakub Bogusz (1):

- vdpau-softpipe: Build correct source file - vl_winsys_xsp.c

Jerome Glisse (1):

- r600g: workaround hyperz lockup on evergreen

John Kåre Alsaker (1):

- llvmpipe: Fix creation of shared and scanout textures.

Jordan Justen (1):

- attrib: push/pop FRAGMENT_PROGRAM_ARB state

José Fonseca (3):

- scons: Allows choosing VS 10 or 11.
- scons: Define _ALLOW_KEYWORD_MACROS on MSVC builds.

- scons: Warn when using MSVS versions prior to 2012.

Keith Kriewall (1):

- scons: Fix Windows build with LLVM 3.2

Kenneth Graunke (1):

- i965: Fix Crystal Well PCI IDs.

Marek Olšák (5):

- r600g: use async DMA with a non-zero src offset
- r600g: flush and invalidate htile cache when appropriate
- gallium/util: add helper code for 1D integer range
- r600g: always map uninitialized buffer range as unsynchronized
- r600g: pad the DMA CS to a multiple of 8 dwords

Martin Andersson (1):

- winsys/radeon: Only add bo to hash table when creating flink

Matt Turner (1):

- mesa: Allow ETC2/EAC formats with ARB_ES3_compatibility.

Michel Dänzer (3):

- radeonsi: Fix up and enable flat shading.
- r600g/Cayman: Fix blending using destination alpha factor but non-alpha dest
- radeonsi: Fix off-by-one for maximum vertex element index in some cases

Tapani Pälli (2):

- mesa: add missing case in _mesa_GetTexParameterfv()
- mesa/es: NULL check in EGLImageTargetTexture2DOES

Vadim Girlin (1):

- r600g: fix check_and_set_bank_swizzle for cayman

Vincent Lejeune (2):

- r600g/llvm: Add support for UBO
- r600g: Check comp_mask before merging export instructions

4.230 Mesa 9.1 Release Notes / February 22, 2013

Mesa 9.1 is a new development release. People who are concerned with stability and reliability should stick with a previous release or wait for Mesa 9.1.1.

Mesa 9.1 implements the OpenGL 3.1 API, but the version reported by glGetString(GL_VERSION) or glGetIntegerv(GL_MAJOR_VERSION) / glGetIntegerv(GL_MINOR_VERSION) depends on the particular driver being used. Some drivers don't support all the features required in OpenGL 3.1. OpenGL 3.1 is **only** available if requested at context creation because GL_ARB_compatibility is not supported.

4.230.1 MD5 checksums

```
86d40f3056f89949368764bf84aff55e  MesaLib-9.1.tar.gz
d3891e02215422e120271d976ff1947e  MesaLib-9.1.tar.bz2
01645f28f53351c23b0beb6c688911d8  MesaLib-9.1.zip
```

4.230.2 New features

Note: some of the new features are only available with certain drivers.

- GL_ANGLE_texture_compression_dxt3
- GL_ANGLE_texture_compression_dxt5
- GL_ARB_ES3_compatibility
- GL_ARB_internalformat_query
- GL_ARB_map_buffer_alignment
- GL_ARB_shading_language_packing
- GL_ARB_texture_buffer_object_rgb32
- GL_ARB_texture_cube_map_array
- GL_EXT_color_buffer_float
- GL_OES_depth_texture_cube_map
- OpenGL 3.1 core profile support on Radeon HD2000 up to HD6000 series
- Multisample anti-aliasing support on Radeon X1000 series
- OpenGL ES 3.0 support on Intel HD Graphics 2000, 2500, 3000, and 4000

4.230.3 Bug fixes

TBD – This list is likely incomplete.

4.230.4 Changes

- Removed VAAPI state tracker (unmaintained and broken)
- Removed i965's broken hardware implementation of GL_NV_vertex_program
- Removed swrast support for GL_NV_vertex_program
- Removed swrast support for GL_NV_fragment_program
- Removed OpenVMS support (unmaintained and broken)
- Removed makedepend build dependency

4.231 Mesa 9.0.3 Release Notes / February 21th, 2013

Mesa 9.0.3 is a bug fix release which fixes bugs found since the 9.0.2 release.

Mesa 9.0 implements the OpenGL 3.1 API, but the version reported by glGetString(GL_VERSION) or glGetIntegerv(GL_MAJOR_VERSION) / glGetIntegerv(GL_MINOR_VERSION) depends on the particular driver being used. Some drivers don't support all the features required in OpenGL 3.1. OpenGL 3.1 is **only** available if requested at context creation because GL_ARB_compatibility is not supported.

4.231.1 MD5 checksums

```
168384ac0101f4600a15edd3561acdc7  MesaLib-9.0.3.tar.gz
d7515cc5116c72ac63d735655bd63689  MesaLib-9.0.3.tar.bz2
a2e1c794572440fd0d839a7d7dfea00c  MesaLib-9.0.3.zip
```

4.231.2 New features

None.

4.231.3 Bug fixes

This list is likely incomplete.

- Bug 25201 - Pink artifacts on objects in the distance in ETQW/Quake 4
- Bug 31598 - configure: Doesn't check for python libxml2
- Bug 40404 - [softpipe] piglit glsl-max-varyings regression
- Bug 47220 - [bisected] Oglc pxconv-gettex(basic.allCases) regressed
- Bug 48629 - [bisected i965]Oglc shad-compiler(advanced.TestLessThani) regressed
- Bug 54240 - [swrast] piglit fbo-generatemipmap-filtering regression
- Bug 56920 - [sandybridge][uxa] graphics very glitchy and always flickering
- Bug 57166 - [GM45] Chrome experiment "Stars" crash: brw_fs_emit.cpp:708: brw_reg brw_reg_from_fs_reg(fs_reg*): Assertion „!"not reached"" failed.
- Bug 57746 - build test failure: nouveau_fbo.c:198:3: error: too few arguments to function 'nouveau_renderbuffer_del'
- Bug 57754 - [swrast] Mesa 9.1-devel implementation error: Unable to delete renderbuffer, no context
- Bug 58680 - [IVB] Graphical glitches in 0 A.D
- Bug 58972 - [softpipe] util/u_tile.c:795:pipe_put_tile_z: Assertion '0' failed.
- Bug 59364 - [bisected] Mesa build fails: clientattrib.c:33:22: fatal error: indirect.h: No such file or directory
- Bug 59700 - [ILK/SNB/IVB Bisected]Oglc vertexshader(advanced.TestLightsTwoSided) causes GPU hung
- Bug 59873 - [swrast] piglit ext_framebuffer_multisample-interpolation 0 centroid-edges regression
- Bug 60052 - [Bisected]Piglit glx_extension_string_sanity fail
- Bug 60172 - Planeshift: triangles where grass would be

4.231.4 Changes

The full set of changes can be viewed by using the following GIT command:

```
git log mesa-9.0.2..mesa-9.0.3
```

Adam Jackson (1):

- r200: Fix probable thinko in r200EmitArrays

Andreas Boll (7):

- docs: Add 9.0.2 release md5sums
- docs: add news item for 9.0.2 release
- configure.ac: Allow OpenGL ES1 and ES2 only with enabled OpenGL
- build: require python module libxml2
- cherry-ignore: Ignore candidates for the 9.1 branch.
- mesa: Bump version to 9.0.3
- docs: Add 9.0.3 release notes

Anuj Phogat (1):

- mesa: Fix GL_LUMINANCE handling for textures in glGetTexImage

Brian Paul (29):

- st/glx: accept GLX_SAMPLE_BUFFERS/SAMPLES_ARB == 0
- draw: set precalc_flat flag for AA lines too
- softpipe: fix up FS variant unbinding / deletion
- softpipe: fix unreliable FS variant binding bug
- xlib: handle _mesa_initialize_visual()'s return value
- xlib: allow GLX_DONT_CARE for glXChooseFBConfig() attribute values
- st/glx: allow GLX_DONT_CARE for glXChooseFBConfig() attribute values
- util: fix addressing bug in pipe_put_tile_z() for PIPE_FORMAT_Z32_FLOAT
- util: add get/put_tile_z() support for PIPE_FORMAT_Z32_FLOAT_S8X24_UINT
- mesa: use GLbitfield64 when copying program inputs
- svga: add NULL pointer check in svga_create_sampler_state()
- vbo: add a null pointer check to handle OOM instead of crashing
- osmesa: use _mesa_generate_mipmap() for mipmap generation, not meta
- xlib: use _mesa_generate_mipmap() for mipmap generation, not meta
- st/mesa: set ctx->Const.MaxSamples = 0, not 1
- mesa: fix-up and use _mesa_delete_renderbuffer()
- mesa: pass context parameter to gl_renderbuffer::Delete()
- st/mesa: fix context use-after-free problem in st_renderbuffer_delete()
- dri_glx: fix use after free report

- mesa: remove warning message in _mesa_reference_renderbuffer_()
- st/mesa: add null pointer check in st_renderbuffer_delete()
- util: add some defensive coding in u_upload_alloc()
- st/mesa: do proper error checking for u_upload_alloc() calls
- util: add new error checking code in vbuf helper
- mesa: don't enable GL_EXT_framebuffer_multisample for software drivers
- st/mesa: only enable GL_EXT_framebuffer_multisample if GL_MAX_SAMPLES >= 2
- mesa: don't expose IBM_rasterpos_clip in a core context
- svga: fix sRGB rendering
- nouveau: Fix build.

Chad Versace (1):

- i965/disasm: Fix horizontal stride of dest registers

Eric Anholt (5):

- i965/fs: Fix the gen6-specific if handling for 80ecb8f15b9ad7d6edc
- i965/fs: Don't generate saturates over existing variable values.
- i965: Actually add support for GL_ANY_SAMPLES_PASSED from GL_ARB_oq2.
- i965/vs: Try again when we've successfully spilled a reg.
- i965/gen7: Set up all samplers even if samplers are sparsely used.

Frank Henigman (1):

- mesa: add bounds checking for uniform array access

Jerome Glisse (1):

- r600g: add cs memory usage accounting and limit it v3 (backport for mesa 9.0)

Jordan Justen (1):

- unpack: support unpacking MESA_FORMAT_ARGB2101010

José Fonseca (2):

- mesa/st: Don't use 4bits for GL_UNSIGNED_BYTE_3_3_2(_REV)
- draw: Properly limit vertex buffer fetches on draw arrays.

Kenneth Graunke (19):

- i965: Fix primitive restart on Haswell.
- i965: Refactor texture swizzle generation into a helper.
- i965: Do texture swizzling in hardware on Haswell.
- i965: Lower textureGrad() with samplerCubeShadow.
- i965: Use Haswell's sample_d_c for textureGrad with shadow samplers.
- i965: Add chipset limits for Haswell GT1/GT2.
- cherry-ignore: Ignore i965 guardband bug fixes.
- i965: Add missing _NEW_BUFFERS dirty bit in Gen7 SBE state.

- i965/vs: Create a 'lod_type' temporary for ir->lod_info.lod->type.
- i965/vs: Set LOD to 0 for ordinary texture() calls.
- i965/vs: Store texturing results into a vec4 temporary.
- cherry-ignore: Ignore candidates for the 9.1 branch.
- mesa: Disable GL_NV_primitive_restart extension in core contexts.
- glsl: Track UBO block names in the symbol table.
- build: Fix build on systems where /usr/bin/python isn't python 2.
- i965: Refactor Gen6+ SF attribute override code.
- i965: Compute the maximum SF source attribute.
- i965: Fix the SF Vertex URB Read Length calculation for Sandybridge.
- i965: Fix the SF Vertex URB Read Length calculation for Gen7 platforms.

Marek Olšák (3):

- r600g: fix int->bool conversion in fence_signalled
- gallium/u_upload_mgr: fix a serious memory leak
- r300g: fix blending with blend color and RGBA formats

Matt Turner (3):

- mesa: Return 0 for XFB_VARYING_MAX_LENGTH if no varyings
- mesa: Set transform feedback's default buffer mode to INTERLEAVED_ATTRIBS
- mesa/uniform_query: Don't write to *params if there is an error

Michel Dänzer (1):

- configure.ac: GLX cannot work without OpenGL

Paul Berry (1):

- mesa: Allow glReadBuffer(GL_NONE) for winsys framebuffers.

Roland Scheidegger (1):

- softpipe: fix using optimized filter function

Stefan Dösinger (3):

- meta: Disable GL_FRAGMENT_SHADER_ATI in MESA_META_SHADER
- radeon: Initialize swrast before setting limits
- r200: Initialize swrast before setting limits

Zack Rusin (2):

- glx: only advertise GLX_INTEL_swap_event if it's supported
- DRI2: Don't disable GLX_INTEL_swap_event unconditionally

4.232 Mesa 9.0.2 Release Notes / January 22th, 2013

Mesa 9.0.2 is a bug fix release which fixes bugs found since the 9.0.1 release.

Mesa 9.0 implements the OpenGL 3.1 API, but the version reported by glGetString(GL_VERSION) or glGetIntegerv(GL_MAJOR_VERSION) / glGetIntegerv(GL_MINOR_VERSION) depends on the particular driver being used. Some drivers don't support all the features required in OpenGL 3.1. OpenGL 3.1 is **only** available if requested at context creation because GL_ARB_compatibility is not supported.

4.232.1 MD5 checksums

```
5ae216ca9fecfa349f14ecb83aa3f124  MesaLib-9.0.2.tar.gz
dc45d1192203e418163e0017640e1cfc  MesaLib-9.0.2.tar.bz2
93d40ec77d656dd04b561ba203ffbb91  MesaLib-9.0.2.zip
```

4.232.2 New features

None.

4.232.3 Bug fixes

This list is likely incomplete.

- Bug 22576 - [KMS] mesa demo spectex broken on rv280
- Bug 26809 - KMS/R200: Bad shading in NWN since Mesa rewrite
- Bug 45877 - [bisected regression] Oglc fbo(negative.invalidParams3) Segmentation fault
- Bug 54402 - st_glsl_to_tgsi.cpp:4006:dst_register: Assertion 'index < VERT_RESULT_MAX' failed
- Bug 55175 - Memoryleak with glPopAttrib only on Intel GM45
- Bug 56442 - glcpp accepts junk after #else/#elif/#endif tokens
- Bug 56706 - EGL sets error to EGL_SUCCESS when DRI driver fails to create context
- Bug 57622 - Webgl conformance shader-with-non-reserved-words crash.
- Bug 57842 - r200: Culling is broken when rendering to an FBO
- Bug 57984 - r300g: blend sfactor=GL_DST_COLOR fails with FBOs
- Bug 58545 - [llvmpipe] src/gallium/auxiliary/gallivm/lp_bld_tgsi_info.c:75:analyse_src: Assertion 'src->Index < (sizeof(ctx->imm)/sizeof((ctx->imm)[0]))' failed.
- Bug 59383 - src/glsl/tests/Makefile.am missing $(top_srcdir)/include

4.232.4 Changes

The full set of changes can be viewed by using the following GIT command:

```
git log mesa-9.0.1..mesa-9.0.2
```

Abdiel Janulgue (1):

- mesa: Fix a crash in update_texture_state() for external texture type

Adam Jackson (4):

- glcpp: Fix visibility CFLAGS in automake
- glcpp: Typo fix.
- galahad, noop: Fix visibility CFLAGS in automake
- r300g: Fix visibility CFLAGS in automake

Alex Deucher (2):

- radeonsi: add some new SI pci ids
- radeonsi: add a new SI pci id

Ander Conselvan de Oliveira (2):

- egl/wayland: Don't invalidate drawable on swap buffers
- egl/wayland: Dispatch the event queue before get_buffers
- egl/wayland: Destroy the pending buffer callback with the egl surface

Andreas Boll (9):

- docs: fix release date of 9.0.1
- docs: add news item for 9.0.1 release
- Add .dirstamp to toplevel .gitignore
- build: use git ls-files for adding all Makefile.in into the release tarball
- build: Fix GLES linkage without libglapi
- Revert "r600g: try to fix streamout for the cases where BURST_COUNT > 0"
- mesa: update .cherry-ignore list
- mesa: Bump version to 9.0.2
- docs: Add 9.0.2 release notes

Anuj Phogat (2):

- mesa: Generate invalid operation in glGenerateMipMap for integer textures
- meta: Remove redundant code in _mesa_meta_GenerateMipmap

Ben Skeggs (3):

- nvc0: fix missing permanent bo reference on poly cache
- nvc0: point vertex runout at a valid address
- nv50: point vertex runout at a valid address

Brian Paul (5):

- svga: don't use uninitialized framebuffer state
- st/mesa: replace REALLOC() with realloc()
- st/mesa: free TGSI tokens with ureg_free_tokens()
- util: added pipe_surface_release() function
- gallivm: support more immediates in lp_build_tgsi_info()

Bryan Cain (1):

- glsl_to_tgsi: set correct register type for array and structure elements

Chad Versace (2):

- i965: Validate requested GLES context version in brwCreateContext
- egl/dri2: Set error code when dri2CreateContextAttribs fails

Chris Fester (1):

- util: null-out the node's prev/next pointers in list_del()

Christoph Bumiller (5):

- nv50/ir/tgsi: fix srcMask for TXP with SHADOW1D
- nvc0: add missing call to map edge flag in push_vbo
- nv50/ir: wrap assertion using typeid in #ifndef NDEBUG
- nouveau: fix undefined behaviour when testing sample_count
- nv50/ir: restore use of long immediate encodings

Dave Airlie (5):

- r600g: fix lod bias/explicit lod with cube maps.
- glsl_to_tgsi: fix dst register for texturing fetches.
- glsl: fix cut-n-paste error in error handling. (v2)
- glsl: initialise killed_all field.
- glsl: fix uninitialised variable from constructor

Eric Anholt (4):

- mesa: Fix the core GL genned-name handling for glBindBufferBase()/Range().
- mesa: Fix core GL genned-name handling for glBeginQuery().
- mesa: Fix segfault on reading from a missing color read buffer.
- i965/gen4: Fix memory leak each time compile_gs_prog() is called.

Ian Romanick (2):

- docs: Add 9.0.1 release md5sums
- glsl: Don't add structure fields to the symbol table

Johannes Obermayr (4):

- clover: Install CL headers.
- gallium/auxiliary: Add -fno-rtti to CXXFLAGS on LLVM >= 3.2.
- clover: Adapt libclc's INCLUDEDIR and LIBEXECDIR to make use of the new introduced libclc.pc.
- tests: AM_CPPFLAGS must include $(top_srcdir) instead of $(top_builddir).

Jonas Ådahl (1):

- wayland: Don't cancel a roundtrip when any event is received

José Fonseca (1):

- llvmpipe: Obey back writemask.

Kenneth Graunke (8):

- i965/vs: Fix unit mismatch in scratch base_offset parameter.
- i965/vs: Implement register spilling.
- mesa: Don't flatten IF statements by default.
- glcpp: Don't use infinite lookhead for #define differentiation.
- i965/vs: Don't lose the MRF writemask when doing compute-to-MRF.
- i965/vs: Preserve the type when copy propagating into an instruction.
- mesa: Fix glGetVertexAttribI[u]iv now that we have real integer attribs.
- i965: Fix AA Line Distance Mode in 3DSTATE_SF on Ivybridge.

Kristian Høgsberg (1):

- egl/wayland: Add invalidate back in eglSwapBuffers()

Maarten Lankhorst (2):

- makefiles: use configured name for -ldrm* where possible
- automake: strip LLVM_CXXFLAGS and LLVM_CPPFLAGS too

Marek Olšák (17):

- st/mesa: fix integer texture border color for some formats (v2)
- r300g: fix texture border color for sRGB formats
- mesa: bump MAX_VARYING to 32
- draw: fix assertion failure in draw_emit_vertex_attr
- vbo: fix glVertexAttribI* functions
- mesa: add MaxNumLevels to gl_texture_image, remove MaxLog2
- mesa: fix error checking of TexStorage(levels) for array and rect textures
- st/mesa: fix guessing the base level size
- st/mesa: fix computation of last_level during texture creation
- st/mesa: fix computation of last_level in GenerateMipmap
- r600g: fix streamout on RS780 and RS880
- r600g: advertise 32 streamout vec4 outputs
- r600g: fix broken streamout if streamout_begin caused a context flush
- mesa: fix BlitFramebuffer between linear and sRGB formats
- r600g: try to fix streamout for the cases where BURST_COUNT > 0
- r600g: always use a tiled resource as the destination of MSAA resolve
- mesa: add MaxNumLevels to gl_texture_image, remove MaxLog2

Mario Kleiner (1):

- mesa: Don't glPopAttrib() GL_POINT_SPRITE_COORD_ORIGIN on < OpenGL-2.0

Matt Turner (1):

- glcpp: Reject garbage after #else and #endif tokens

Stefan Dösinger (1):

- r300: Don't disable destination read if the src blend factor needs it

Tapani Pälli (1):

- android: generate matching remap_helper to dispatch table

Tom Stellard (1):

- r600g: Use LOOP_START_DX10 for loops

Vinson Lee (1):

- i915: Fix wrong sizeof argument in i915_update_tex_unit.

smoki (2):

- r200: fix broken tcl lighting
- radeon/r200: Fix tcl culling

4.233 Mesa 9.0.1 Release Notes / November 16th, 2012

Mesa 9.0.1 is a bug fix release which fixes bugs found since the 9.0 release.

Mesa 9.0 implements the OpenGL 3.1 API, but the version reported by glGetString(GL_VERSION) or glGetIntegerv(GL_MAJOR_VERSION) / glGetIntegerv(GL_MINOR_VERSION) depends on the particular driver being used. Some drivers don't support all the features required in OpenGL 3.1. OpenGL 3.1 is **only** available if requested at context creation because GL_ARB_compatibility is not supported.

4.233.1 MD5 checksums

```
97d6554c05ea7449398afe3a0ede7018  MesaLib-9.0.1.tar.bz2
fd0fd5a6e56bc3dd210c80e42baef975  MesaLib-9.0.1.tar.gz
c2683d957acd530a00f747f50317186f  MesaLib-9.0.1.zip
```

4.233.2 New features

None.

4.233.3 Bug fixes

This list is likely incomplete.

- Bug 44912 - [bisected] WebGL conformance/textures/texture-mips tests fails
- Bug 55856 - kwin with gles window content is not updating (gen4)
- Bug 56057 - INTEL_swap_event not correctly listed
- Bug 56211 - src/mesa/state_tracker/st_cb_texture.c:1123:copy_image_data_to_texture: Assertion 'u_minify(stImage->pt->height0, src_level) == stImage->base.Height' failed.

4.233.4 Changes

The full set of changes can be viewed by using the following GIT command:

```
git log mesa-9.0..mesa-9.0.1
```

Adam Jackson (1):

- glx: Add GLXBadProfileARB to the error string list

Andreas Boll (7):

- docs: add news item for 9.0 release
- mesa: add get-pick-list.sh script into bin/
- mesa: add initial .cherry-ignore file for the 9.0 branch
- mesa: use .cherry-ignore in the get-pick-list.sh script
- build: add config.sub and config.guess to tarballs target
- build: add missing Makefile.in files to tarballs target
- build: add missing files to tarballs target

Brian Paul (2):

- mesa: don't call TexImage driver hooks for zero-sized images
- mesa: fix error check for zero-sized compressed subtexture

Fredrik Höglund (1):

- egl_dri2/x11: Fix eglPostSubBufferNV()

Ian Romanick (5):

- docs: Add 9.0 release md5sums
- i965: Fix regression in depth texture rendering on pre-SNB
- glx: Set sRGBCapable to a default value
- docs: Add 9.0.1 release notes
- mesa: Bump version to 9.0.1

Imre Deak (7):

- mesa: glGet: fix indentation of _mesa_init_get_hash
- mesa: glGet: fix indentation of find_value
- mesa: glGet: fix indentation of print_table_stats
- mesa: glGet: fix API check for EGL_image_external enums
- glapi: rename/move GL_POLYGON_OFFSET_BIAS to its extension section
- mesa: glGet: fix parameter lookup for apps using multiple APIs
- glget: fix make check for glGet GL_POLYGON_OFFSET_BIAS

Jonas Ådahl (1):

- wayland: Destroy frame callback when destroying surface

Kenneth Graunke (1):

- glsl: Allow ir_if in the linker's move_non_declarations function.

Kristian Høgsberg (5):

- gbm: Reject buffers that are not wl_drm buffers in gbm_bo_import()
- gbm: Use the kms dumb ioctls for cursor instead of libkms
- egl/wayland: Update to Wayland 0.99 API
- wayland: Remove 0.85 compatibility #ifdefs
- wayland: Drop support for ill-defined, unused wl_egl_pixmap

Marcin Slusarz (1):

- nouveau: use pre-calculated stride for resource_get_handle

Matt Turner (4):

- egl: Return EGL_BAD_MATCH for invalid profile attributes
- Re-add HAVE_PTHREADS preprocessor macro
- build: Ship install-sh in the tarball
- ralloc: Annotate printf functions with PRINTFLIKE(...)

Michel Dänzer (2):

- st/mesa: Fix source miptree level for copying data to finalized miptree.
- st/mesa: Fix assertions for copying texture image to finalized miptree.

Owen W. Taylor (1):

- glx: Fix listing of INTEL_swap_event in glXQueryExtensionsString()

Quentin Glidic (1):

- intel: Add missing #include <time.h>

Tomeu Vizoso (1):

- mesa/es: Define GL_MAX_TEXTURE_MAX_ANISOTROPY_EXT enum for all GLs

4.234 Mesa 9.0 Release Notes / October 8, 2012

Mesa 9.0 is a new development release. People who are concerned with stability and reliability should stick with a previous release or wait for Mesa 9.0.1.

Mesa 9.0 implements the OpenGL 3.1 API, but the version reported by glGetString(GL_VERSION) or glGetIntegerv(GL_MAJOR_VERSION) / glGetIntegerv(GL_MINOR_VERSION) depends on the particular driver being used. Some drivers don't support all the features required in OpenGL 3.1. OpenGL 3.1 is **only** available if requested at context creation because GL_ARB_compatibility is not supported.

4.234.1 MD5 checksums

```
be4cd34c6599a7cb9d254b05c48bdb1f  MesaLib-9.0.tar.gz
60e557ce407be3732711da484ab3db6c  MesaLib-9.0.tar.bz2
16b128544cd3f7e237927bb9f8aab7ce  MesaLib-9.0.zip
```

4.234.2 New features

Note: some of the new features are only available with certain drivers.

- Added new Gallium3D - nv30 driver
- Added new Gallium3D - radeonsi driver
- Added OpenCL state tracker Clover
- Completed VDPAU state tracker (video decoding support is currently limited to MPEG1 and MPEG2)
- GL_ARB_base_instance
- GL_ARB_blend_func_extended
- GL_ARB_debug_output
- GL_ARB_invalidate_subdata - Currently a "no-op" implementation. This extension is always enabled in all drivers.
- GL_ARB_shader_bit_encoding
- GL_ARB_texture_buffer_object
- GL_ARB_timer_query
- GL_ARB_transform_feedback3
- GL_ARB_transform_feedback_instanced
- GL_ARB_uniform_buffer_object
- GL_EXT_unpack_subimage for ES 2.0
- GL_EXT_read_format_bgra for ES 1.1 and 2.0
- GL_EXT_texture_rg for ES 2.x
- GL_NV_read_buffer for ES 2.0
- GLX_ARB_create_context_robustness
- EGL_KHR_create_context
- EGL_KHR_surfaceless_context - This replaces the EGL_KHR_surfaceless_{gles1,gles2,opengl} extensions that were never approved by Khronos.
- EGL_EXT_create_context_robustness

4.234.3 Bug fixes

TBD – This list is likely incomplete.

4.234.4 Changes

- The legacy/static Makefile system (ex: 'make linux-dri') has been removed. The two supported build methods are now autoconf/automake and SCons.
- Removed support for GL_ARB_shadow_ambient extension
- Removed Gallium3D - nvfx driver (use nv30 instead)
- libGLU has been moved into its own repository, found at https://cgit.freedesktop.org/mesa/glu/

4.235 Mesa 8.0.5 Release Notes / October 24, 2012

Mesa 8.0.5 is a bug fix release which fixes bugs found since the 8.0.4 release.

Mesa 8.0.5 implements the OpenGL 3.0 API, but the version reported by glGetString(GL_VERSION) depends on the particular driver being used. Some drivers don't support all the features required in OpenGL 3.0.

See the *Compiling/Installing page* for prerequisites for DRI hardware acceleration.

4.235.1 MD5 checksums

```
cda5d101f43b8784fa60bdeaca4056f2  MesaLib-8.0.5.tar.gz
01305591073a76b65267f69f27d635a3  MesaLib-8.0.5.tar.bz2
97f11c00cac8fb98aa0131990086dc8e  MesaLib-8.0.5.zip
```

4.235.2 New features

None.

4.235.3 Bug fixes

This list is likely incomplete.

- Bug 44912 - [bisected] WebGL conformance/textures/texture-mips tests fails
- Bug 46644 - Sandybridge Mobile: ARBfp TXP with coords from fragment.color doesn't apply W divide
- Bug 46784 - MAD using multiply written register fails
- Bug 47375 - Blender crash on startup after upgrade to mesa 8.0.1
- Bug 48120 - GL_EXT_texture_sRGB_decode still broken
- Bug 48628 - [bisected ILK]Oglc fogexp(basic.allCases) regressed
- Bug 49772 - [SNB]Oglc depth-stencil(basic.read.ds) regressed
- Bug 52129 - [Bisected ILK]Piglit spec_ARB_shader_texture_lod_execution_glsl-fs-shadow2DGradARB-01 regressed
- Bug 52382 - [ivb gt1] Severe image corruption and GPU Hang, too many PS threads
- Bug 52563 - build failure - struct radeon_renderbuffer has no member named Base
- Bug 53311 - [Bisected IVB]Oglc transform_feedback(advanced.transformFeedback.points) Invalid argument
- Bug 53314 - [llvmpipe] src/gallium/drivers/llvmpipe/lp_texture.c:920:llvmpipe_get_texture_tile_layout: Assertion 'x < lpr->tiles_per_row[level]' failed.
- Bug 53316 - [llvmpipe] src/gallium/drivers/llvmpipe/lp_texture.c:601:llvmpipe_get_transfer: Assertion 'resource' failed.
- Bug 53317 - [llvmpipe] SIGSEGV src/gallium/auxiliary/gallivm/lp_bld_sample.c:99
- Bug 53318 - [softpipe] sp_state_shader.c:194:softpipe_delete_fs_state: Assertion 'var != softpipe->fs_variant' failed.
- Bug 53319 - [softpipe] sp_texture.c:322:softpipe_get_transfer: Assertion 'level <= resource->last_level' failed.
- Bug 53618 - [Bisected i915]Piglit glx_GLX_ARB_create_context_NULL_attribute_list Aborted

- Bug 53972 - Black Mirror III: too dark
- Bug 54183 - [Bisected ILK regression]many piglit/oglc/ogles2 cases Segmentation fault
- Bug 54193 - output_components uninitialized in fs_visitor::emit_fb_writes()
- Bug 54301 - [Bisected ILK regression]Piglit glx_GLX_ARB_create_context_forward-compatible_flag_with_3.0 Segmentation fault
- Bug 56211 - src/mesa/state_tracker/st_cb_texture.c:1123:copy_image_data_to_texture: Assertion 'u_minify(stImage->pt->height0, src_level) == stImage->base.Height' failed.

4.235.4 Changes

The full set of changes can be viewed by using the following GIT command:

```
git log mesa-8.0.4..mesa-8.0.5
```

Alex Deucher (3):

- r600g: 8.0.x support for Trinity
- r600g: add new Sumo, Palm, BTC pci ids
- r600g: add additional evergreen pci ids

Andreas Boll (4):

- docs/relnotes-8.0.4: fix html markup
- mesa: fix html in shortlog_mesa.sh script
- mesa: add get-pick-list.sh script into bin/
- mesa: Bump version number to 8.0.5

Brian Paul (18):

- mesa: use _mesa_is_user_fbo() and _mesa_is_winsys_fbo() functions
- intel: use _mesa_is_winsys/user_fbo() helpers
- st/egl: fix uninitialized pointer bug
- mesa: added Ian's shortlog_mesa.sh script in bin/
- mesa: loosen small matrix determinant check
- xlib: add X error handler around XGetImage() call
- radeon: set swrast_renderbuffer::ColorType field when mapping renderbuffers
- svga: fix invalid memory reference in needs_to_create_zero()
- meta: fix glDrawPixels fallback test, stencil drawing
- radeon: fix Base/base typo
- st/mesa: fix glCopyTexSubImage crash
- gallivm: fix crash in lp_sampler_static_state()
- st/mesa: fix renderbuffer validation bug
- softpipe: fix softpipe_delete_fs_state() failed assertion
- mesa: raise GL_INVALID_OPERATION in glGenerateMipmap for missing base image

- st/mesa: s/CALLOC/calloc/ to fix allocation bug
- mesa: do internal format error checking for glTexStorage()
- mesa: fix incorrect error for glCompressedSubTexImage

Chad Versace (3):

- mesa: Don't advertise GLES extensions in GL contexts
- i830: Fix stack corruption
- swrast: Fix implicit declaration warnings

Chris Forbes (1):

- mesa: fix dropped && in glGetStringi()

Christoph Bumiller (1):

- st/mesa: call update_renderbuffer_surface for sRGB renderbuffers, too

Eric Anholt (9):

- i965/gen7: Reduce GT1 WM thread count according to updated BSpec.
- i965/fs: Invalidate live intervals in passes that remove an instruction.
- i965: Fix bug in the old FS backend's projtex() calculation.
- i965: Add support for GL_SKIP_DECODE_EXT on other SRGB formats.
- i965/vs: Convert EdgeFlagPointer values appropriately for the VS on gen4.
- i965: Fix accumulator_contains() test to also reject swizzles of the dst.
- mesa: Fix glPopAttrib() behavior on GL_FRAMEBUFFER_SRGB.
- mesa: In conditional rendering fallback, check the query status.
- i965: Drop the confusing saturate argument to math instruction setup.

Ian Romanick (8):

- docs: Add 8.0.4 release md5sums
- Revert "i965: Avoid unnecessary recompiles for shaders that don't use dFdy()."
- i965: Fix regression in depth texture rendering on pre-SNB
- dri2: Fix bug in attribute handling for non-desktop OpenGL contexts
- mesa: Generate an error when glCopyTexImage border is invalid
- mesa/es: Validate glTexImage border in Mesa code rather than the ES wrapper
- mesa: Allow glGetTexParameter of GL_TEXTURE_SRGB_DECODE_EXT
- dri_util: Use calloc to allocate __DRIcontext

Jonas Maebe (1):

- darwin: do not create double-buffered offscreen pixel formats

Jordan Justen (1):

- intel: move error on create context to proper path

José Fonseca (1):

- mesa: disable MSVC global optimization in pack.c

Kenneth Graunke (8):

- mesa: Use GLdouble for depthMax in final unpack conversions.
- i965/fs: Initialize output_components[] by filling it with zeros.
- mesa: Prevent repeated glDeleteShader() from blowing away our refcounts.
- i965: Support MESA_FORMAT_SIGNED_RGBA_16.
- glsl: Fix #pragma invariant(all) language version check.
- i965/vs: Don't clobber sampler message MRFs with subexpressions.
- intel: Move finish_batch() call before MI_BATCH_BUFFER_END and padding.
- i965/fs: Don't use brw->fragment_program in calculate_urb_setup().

Maarten Lankhorst (1):

- winsys/radeon: Remove unnecessary pipe_thread_destroy in radeon_drm_cs_destroy

Marek Olšák (6):

- mesa: remove assertions that do not allow compressed 2D_ARRAY textures
- r300g: fix colormask with non-BGRA formats
- r600g: fix RSQ of negative value on Cayman
- r600g: fix EXP on Cayman
- r600g: fix instance divisor on Cayman
- gallium/u_blit: set dst format from pipe_resource, not pipe_surface

Michel Dänzer (2):

- st/mesa: Fix source miptree level for copying data to finalized miptree.
- st/mesa: Fix assertions for copying texture image to finalized miptree.

Niels Ole Salscheider (1):

- st/mesa: index can be negative in the PROGRAM_CONSTANT case

Paul Berry (5):

- i965: Compute dFdy() correctly for FBOs.
- mesa: Add UsesDFdy to struct gl_fragment_program.
- i965: Avoid unnecessary recompiles for shaders that don't use dFdy().
- i965/Gen6: Work around GPU hangs due to misaligned depth coordinate offsets.
- i965/Gen7: Work around GPU hangs due to misaligned depth coordinate offsets.

Stéphane Marchesin (1):

- glsl/linker: Avoid buffer over-run in parcel_out_uniform_storage::visit_field

Tapani Pälli (2):

- xmlconfig: use __progname when building for Android
- android: do not expose single buffered eglconfigs

Vadim Girlin (1):

- winsys/radeon: fix relocs caching

4.236 Mesa 8.0.4 Release Notes / July 10, 2012

Mesa 8.0.4 is a bug fix release which fixes bugs found since the 8.0.2 release.

Mesa 8.0.4 implements the OpenGL 3.0 API, but the version reported by glGetString(GL_VERSION) depends on the particular driver being used. Some drivers don't support all the features required in OpenGL 3.0.

See the *Compiling/Installing page* for prerequisites for DRI hardware acceleration.

4.236.1 MD5 checksums

```
02b96082d2f1ad72e7385f4022afafb9  MesaLib-8.0.4.tar.gz
d546f988adfdf986cff45b1efa2d8a46  MesaLib-8.0.4.tar.bz2
1f0fdabe6e8019d4de6c16e20e74d163  MesaLib-8.0.4.zip
```

4.236.2 New features

None.

4.236.3 Bug fixes

This list is likely incomplete.

- Bug 45967 - piglit getteximage-invalid-format-for-packed-type regression
- Bug 47742 - [softpipe] piglit fbo-generatemipmap-array regression
- Bug 48141 - [vmwgfx] src/gallium/auxiliary/util/u_inlines.h:256:pipe_buffer_map_range: Assertion 'offset + length <= buffer->width0' failed.
- Bug 48472 - GPU Lockup while running demo (rzr - the scene is dead) in wine
- Bug 50033 - src/mesa/state_tracker/st_cb_fbo.c:379:st_render_texture: Assertion 'strb->rtt_level <= strb->texture->last_level' failed.
- Bug 50621 - Mesa fails its test suite with a buffer overflow.
- Bug 50298 - [ILK IVB bisected]Ogles2conform GL/sin/sin_float_vert_xvary.test regressed
- Bug 51574 - ir_loop_jump constructor assigns member variable to itself

4.236.4 Changes

The full set of changes can be viewed by using the following GIT command:

```
git log mesa-8.0.3..mesa-8.0.4
```

Andreas Betz (1):

- vega: fix 565 color unpacking bug

Antoine Labour (2):

- meta: Cleanup the resources we allocate.
- mesa: Free uniforms correclty.

Brian Paul (22):

- docs: add link to 8.0.3 release notes
- mesa: fix Z32_FLOAT -> uint conversion functions
- draw: fix primitive restart bug by using the index buffer offset
- st/mesa: fix glDrawPixels(GL_DEPTH_COMPONENT) color output
- svga: fix synchronization bug between sampler views and surfaces
- mesa: new _mesa_error_check_format_and_type() function
- mesa: add missing GL_UNSIGNED_INT_10F_11F_11F_REV case
- mesa: fix missing return value in getteximage_error_check()
- st/mesa: pass GL_MAP_INVALIDATE_RANGE_BIT to gallium drivers
- svga: add 0.5 in float->int conversion of sample min/max lod
- svga: fix min/max lod clamping
- svga: change PIPE_CAPF_MAX_TEXTURE_LOD_BIAS from 16.0 to 15.0
- st/mesa: add fallback pipe formats for (compressed) R, RG formats
- st/mesa: copy num_immediates field when copying the immediates array
- svga: move svga_texture() casts/calls in svga_surface_copy()
- svga: reset vertex buffer offset in svga_release_user_upl_buffers()
- st/mesa: don't set PIPE_BIND_DISPLAY_TARGET for user-created renderbuffers
- st/mesa: use private pipe_sampler_view in decompress_with_blit()
- st/mesa: add null pointer check in st_texture_image_map()
- st/mesa: fix mipmap image size computation w.r.t. texture arrays
- draw: fix missing immediates bug in polygon stipple code
- st/mesa: fix max_offset computation for base vertex

Christoph Bumiller (1):

- nv50: handle NEG,ABS modifiers for short RCP encoding

Dylan Noblesmith (1):

- mesa: require GL_MAX_SAMPLES >= 4 for GL 3.0

Eric Anholt (1):

- i965/vs: Fix texelFetchOffset()

Ian Romanick (5):

- docs: Add 8.0.3 release md5sums
- glx/tests: Fix off-by-one error in allocating extension string buffer
- glsl: Remove spurious printf messages
- glsl: Fix pi/2 constant in acos built-in function
- mesa: Bump version number to 8.0.4

José Fonseca (2):

- mesa: Avoid void acinclude.m4 Android.common.mk Android.mk autogen.sh bin common.py configs configure.ac docs doxygen include Makefile scons SConstruct src tests arithmetic.
- draw: Ensure that prepare is always run after LLVM garbagge collection.

Kenneth Graunke (15):

- mesa: Check for a negative "size" parameter in glCopyBufferSubData().
- i965: Fix brw_swap_cmod() for LE/GE comparisons.
- glsl: Remove unused ir_loop_jump::loop pointer.
- ralloc: Fix ralloc_parent() of memory allocated out of the NULL context.
- mesa: Restore depth texture state on glPopAttrib(GL_TEXTURE_BIT).
- glsl/builtins: Fix textureGrad() for Array samplers.
- mesa: Unbind ARB_copy_buffer and transform feedback buffers on delete.
- mesa: Support BindBuffer{Base,Offset,Range} with a buffer of 0.
- mesa: Unbind ARB_transform_feedback2 binding points on Delete too.
- meta: Fix GL_RENDERBUFFER binding in decompress_texture_image().
- i965/fs: Fix texelFetchOffset() on pre-Gen7.
- i965/vs: Fix texelFetchOffset() on pre-Gen7.
- i965/fs: Fix user-defined FS outputs with less than four components.
- glsl: Hook up loop_variable_state destructor to plug a memory leak.
- glsl: Don't trust loop analysis in the presence of function calls.

Kurt Roeckx (1):

- i830: Fix crash for GL_STENCIL_TEST in i830Enable()

Lukas Rössler (1):

- glu: fix two Clang warnings

Marek Olšák (2):

- mesa: allow exposing GL3 without EXT_texture_integer
- st/mesa: don't do srgb->linear conversion in decompress_with_blit

Paul Seidler (1):

- tests: include mesa headers

Stéphane Marchesin (3):

- glx: Handle a null reply in QueryVersion.
- i915g: Don't invert signalled/unsignalled fences
- i915g: Don't avoid flushing when we have a pending fence.

Thomas Gstädtner (1):

- gallium/targets: pass ldflags parameter to MKLIB

Vadim Girlin (2):

- st/mesa: set stObj->lastLevel in guess_and_alloc_texture

- r600g: check gpr count limit

Vinson Lee (1):

- st/mesa: Fix uninitialized members in glsl_to_tgsi_visitor constructor.

4.237 Mesa 8.0.3 Release Notes / May 18, 2012

Mesa 8.0.3 is a bug fix release which fixes bugs found since the 8.0.2 release.

Mesa 8.0.3 implements the OpenGL 3.0 API, but the version reported by glGetString(GL_VERSION) depends on the particular driver being used. Some drivers don't support all the features required in OpenGL 3.0.

See the *Compiling/Installing page* for prerequisites for DRI hardware acceleration.

4.237.1 MD5 checksums

```
b7210a6d6e4584bd852ab29294ce717e  MesaLib-8.0.3.tar.gz
cc5ee15e306b8c15da6a478923797171  MesaLib-8.0.3.tar.bz2
32773634a0f7e70a680505a81426eccf  MesaLib-8.0.3.zip
```

4.237.2 New features

None.

4.237.3 Bug fixes

This list is likely incomplete.

- Bug 28138 - [G45] Regnum Online, sparkling in in-game rendering
- Bug 30102 - glean depthStencil test fails BadLength with indirect non-swrast rendering
- Bug 40361 - Glitches on X3100 after upgrade to 7.11
- Bug 41152 - [glsl] Shader backend in Regnum Online does not work
- Bug 41216 - [bisected pineview]oglc filtercubemin(basic.sizedRGBA) fails
- Bug 41372 - i830_state.c PBO crash
- Bug 41495 - i830: intel_get_vb_max / intel_batchbuffer_space mismatch.
- Bug 44701 - Regnum online textures flickering
- Bug 44961 - [bisected i965] oglc sRGB(Mipmap.1D_textures) regressed
- Bug 44970 - [i965]oglc max_values(negative.textureSize.textureCube) segfaults
- Bug 45214 - Textures disappearing or missing in RegnumOnline OpenGL game
- Bug 45558 - cannot render on a drawable of size equal the max framebuffer size
- Bug 45921 - [r300g, bisected] Multiple piglit regressions after glsl_to_tgsi changes
- Bug 46303 - [SNB] segfault in intel_miptree_release()
- Bug 46739 - [snb-m-gt2+] compiz crashed with SIGSEGV in intel_miptree_release()

- Bug 46834 - small performance when playing flightgear (swrast fallback through GTT mapping)
- Bug 47126 - tests/fbo/fbo-array.c:109: create_array_fbo: Assertion 'glGetError() == 0' failed.
- Bug 48218 - brw_fs_schedule_instructions.cpp segfault due to accessing not allocated last_mrf_write[16]
- Bug 48545 - LLVMpipe glReadPixels Firefox hits the slow path (WebGL rendering)
- Bug 49124 - swrast/s_texfetch.c:1156: set_fetch_functions: Assertion 'texImage->FetchTexel' failed.

4.237.4 Changes

The full set of changes can be viewed by using the following GIT command:

```
git log mesa-8.0.2..mesa-8.0.3
```

Alban Browaeys (1):

- dri/i915: Fix off-by-one in i830 clip region size.

Alex Deucher (2):

- r200: fix fog coordinate emit
- radeon: fix fog coordinate emit

Alexander von Gluck (4):

- llvmpipe: fix symbol conflict on Haiku
- svga: fix typedef conflicts on Haiku
- mesa: Don't use newlocale on Haiku
- glsl: Don't use newlocale on Haiku

Anuj Phogat (4):

- mesa: fix issues with texture border and array textures
- mesa: Fix valid texture target test in _mesa_GetTexLevelParameteriv()
- mesa: Fix the cause of piglit test fbo-array failure
- intel: Fix a case when mapping large texture fails

Brian Paul (17):

- mesa: add a couple fast-paths to fast_read_rgba_pixels_memcpy()
- mesa/gdi: remove wmesa_set_renderbuffer_funcs() function
- mesa/gdi: remove clear_color() function
- mesa: bump version to 8.0.2 in configs/default
- swrast: include s_fragprog.h to silence warnings
- mesa: remove LSB-first pixel packing check in glReadPixels
- mesa: fix error in _mesa_format_matches_format_and_type() for RGB888
- mesa: add BGR888 code in _mesa_format_matches_format_and_type()
- vbo: fix node_attrsz[] usage in vbo_bind_vertex_list()
- mesa: add missing texture integer test in glTexSubImage()

- mesa: add missing return after _mesa_error() in update_array()
- glsl: propagate MaxUnrollIterations to the optimizer's loop unroller
- st/mesa: set MaxUnrollIterations = 255
- st/mesa: no-op glCopyPixels if source region is out of bounds
- mesa: do more teximage error checking for generic compressed formats
- mesa: fix/add error check in _mesa_ColorMaterial()
- mesa: fix glMaterial / dlist bug

Chad Versace (3):

- glsl: Fix Android build
- main: Fix memory leak in _mesa_make_extension_string()
- intel: Disable ARB_framebuffer_object in ES contexts

Chris Wilson (1):

- i830: Compute initial number of vertices from remaining batch space

Dave Airlie (4):

- mesa/format_unpack: add LUMINANCE 8/16 UINT/INT
- glx/drisw: avoid segfaults when we fail to get visual
- drisw: fix image stride calculation for 16-bit.
- intel: fix TFP at 16-bpp

Dylan Noblesmith (7):

- intel: fix null dereference processing HiZ buffer
- util: fix undefined behavior
- util: fix uninitialized table
- egl: fix uninitialized values
- st/vega: fix uninitialized values
- egl-static: fix printf warning
- i965: fix typo

Eric Anholt (19):

- i965/fs: Jump from discard statements to the end of the program when done.
- intel: Fix rendering from textures after RenderTexture().
- mesa: Fix handling of glCopyBufferSubData() for src == dst.
- i965/fs: Move GL_CLAMP handling to coordinate setup.
- i965/fs: Implement GL_CLAMP behavior on texture rectangles on gen6+.
- mesa: Fix push/pop of multisample coverage invert.
- mesa: Include the multisample enables under GL_MULTISAMPLE_BIT attrib as well.
- mesa: Fix display list handling for GL_ARB_draw_instanced.
- mesa: Fix display lists for draw_elements_base_vertex with draw_instanced.

- mesa: Add missing error check for first < 0 in glDrawArraysInstanced().
- i915: Fix piglit fbo-nodepth-test on i830.
- intel: Return success when asked to allocate a 0-width/height renderbuffer.
- mesa: Throw error on glGetActiveUniform inside Begin/End.
- i965/vs: Fix up swizzle for dereference_array of matrices.
- glsl: Fix indentation of switch code.
- glsl: Let the constructor figure out the types of switch-related expressions.
- glsl: Reject non-scalar switch expressions.
- glsl: Fix assertion failure on handling switch on uint expressions.
- mesa: Check for framebuffer completeness before looking at the rb.

Eugeni Dodonov (1):

- intel: add PCI IDs for Ivy Bridge GT2 server variant

Han Shen() (1):

- bin/mklib: remove '-m32' for arm linux

Ian Romanick (1):

- mesa: Bump version number to 8.0.3

Jakob Bornecrantz (1):

- docs: Add 8.0.2 md5sums

Jeremy Huddleston (7):

- darwin: Eliminate a pthread mutex leak
- darwin: Fix an error message
- darwin: Make reported errors more user-friendly
- darwin: Use ASL for logging
- darwin: Unlock our mutex before destroying it
- darwin: Eliminate a possible race condition while destroying a surface
- darwin: Address a build failure on Leopard and earlier OS versions

Jon TURNEY (1):

- Have __glImageSize handle format GL_DEPTH_STENCIL_NV the same way as the server does

Jonas Maebe (2):

- glapi: Correct size of allocated _glapi_table struct
- apple: Fix a use after free

Jordan Justen (1):

- mesa: Add primitive restart support to glArrayElement

Kenneth Graunke (12):

- i965: Actually upload sampler state pointers for the VS unit on Gen6.
- i965/fs: Fix FB writes that tried to use the non-existent m16 register.

- vbo: Remove pedantic warning about 'end' beind out of bounds.
- vbo: Ignore invalid element ranges which are outside VBO bounds.
- vbo: Rework checking of 'end' against _MaxElement.
- vbo: Eliminate short-circuiting in invalid-start case.
- i965: Fix GPU hangs in the dummy fragment shader.
- i965: Make the dummy fragment shader work in SIMD16 mode.
- drirc: Add force_glsl_extensions_warn workaround for Unigine Heaven.
- i965: Avoid explicit accumulator operands in SIMD16 mode on Gen7.
- intel: Remove pointless software fallback for glBitmap on Gen6.
- glsl: Fix broken constant expression handling for <, <=, >, and >=.

Kurt Roeckx (2):

- i915: Compute maximum number of verts using the actual batchbuffer size.
- i915: Fix i830 polygon stipple from PBOs.

Marek Olšák (5):

- r300g/swtcl: don't print an error when getting ClipVertex
- r300g/swtcl: don't enter u_vbuf_mgr
- r300g/swtcl: don't expose shader subroutine support
- r300g/swtcl: fix polygon offset
- r300g/swtcl: fix crash when back color is present in vertex shader

Mathias Fröhlich (1):

- glsl: Avoid excessive loop unrolling.

Matt Turner (1):

- Remove -ffast-math from default CFLAGS

Paul Berry (1):

- i915: Initialize swrast_texture_image structure fields.

Roland Scheidegger (1):

- mesa: check_index_bounds off-by-one fix

Tom Stellard (2):

- r300/compiler: Clear loop registers in vertex shaders w/o loops
- r300/compiler: Copy all instruction attributes during local transfoms

Vinson Lee (5):

- mesa: Fix memory leak in _mesa_get_uniform_location.
- linker: Fix memory leak in count_uniform_size::visit_field.
- swrast: Fix memory leaks in blit_linear.
- ir_to_mesa: Fix uninitialized member in add_uniform_to_shader.
- mesa: Fix memory leak in generate_mipmap_compressed.

Yuanhan Liu (2):

- i915: set SPRITE_POINT_ENABLE bit correctly
- i965: fix wrong cube/3D texture layout

4.238 Mesa 8.0.2 Release Notes / March 21, 2012

Mesa 8.0.2 is a bug fix release which fixes bugs found since the 8.0.1 release.

Mesa 8.0.2 implements the OpenGL 3.0 API, but the version reported by glGetString(GL_VERSION) depends on the particular driver being used. Some drivers don't support all the features required in OpenGL 3.0.

See the *Compiling/Installing page* for prerequisites for DRI hardware acceleration.

4.238.1 MD5 checksums

```
70eb3dc74fbfcd72f6776268ee1db52e  MesaLib-8.0.2.tar.gz
a368104e5700707048dc3e8691a9a7a1  MesaLib-8.0.2.tar.bz2
d5e5cdb85d2afdbcd1c0623d3ed1c54d  MesaLib-8.0.2.zip
```

4.238.2 New features

None.

4.238.3 Bug fixes

This list is likely incomplete.

- Bug 38720 - [SNB] Trine triggers a GPU hang
- Bug 40059 - [SNB] hang in "Amnesia: The Dark Descent" demo
- Bug 45216 - [SNB] GPU hang in OilRush
- Bug 46631 - It's really hard to hit the fast path for the fallback glReadPixels code
- Bug 46679 - glReadPixels on a luminance texture returns the wrong values
- Bug 46311 - Missing support of point size in Mesa core
- Bug 46665 - [PNV] webgl conformance case max texture fails
- Bug 45975 - [Gen4 + ILK] render with pointcoord will fail to render
- Bug 46666 - [PNV] webgl conformance case NPOT case fails with TEXTURE_MIN_FILTER set to LINEAR

4.238.4 Changes

The full set of changes can be viewed by using the following GIT command:

```
git log mesa-8.0.1..mesa-8.0.2
```

Brian Paul (7):

- svga: add null vs pointer check in update_need_pipeline()

- util: add mutex lock in u_debug_memory.c code
- mesa: add _mesa_rebase_rgba_float/uint() functions
- mesa: use _mesa_rebase_rgba_float/uint() in glReadPixels code
- mesa: use _mesa_rebase_rgba_float/uint() in glGetTexImage code
- mesa: fix GL_LUMINANCE handling in glGetTexImage
- docs: add links to 8.0.1 release notes

Daniel Vetter (1):

- i965: fixup W-tile offset computation to take swizzling into account

Dylan Noblesmith (1):

- mesa: add back glGetnUniform*v() overflow error reporting

Ian Romanick (1):

- docs: Add 8.0.1 release md5sums

Jakob Bornecrantz (3):

- mesa: Include mesa ES mapi generated files
- mesa: Bump version number to 8.0.2
- docs: Add 8.0.2 release notes

Jeremy Huddleston (3):

- darwin: config file cleanups
- darwin: Build create_context.c
- darwin: Link against libxcb

José Fonseca (1):

- svga: Clamp advertised PIPE_SHADER_CAP_MAX_TEMPS to SVGA3D_TEMPREG_MAX.

Kenneth Graunke (2):

- i965: Only set Last Render Target Select on the last FB write.
- i965: Fix Gen6+ dynamic state upper bound on older kernels.

Marek Olšák (1):

- gallium/rtasm: properly detect SSE and SSE2

Neil Roberts (1):

- mesa: Don't disable fast path for normalized types

Tom Stellard (1):

- r300/compiler: Fix bug when lowering KILP on r300 cards

Yuanhan Liu (6):

- mesa: let GL3 buf obj queries not depend on opengl major version
- tnl: let _TNL_ATTRIB_POINTSIZE do not depend on ctx->VertexProgram._Enabled
- i915: fix wrong rendering of gl_PointSize on Pineview
- i915: move the FALLBACK_DRAW_OFFSET check outside the drawing rect check

- i965: handle gl_PointCoord for Gen4 and Gen5 platforms
- i915: fallback for NPOT cubemap texture

Zack Rusin (3):

- svga: fix a crash happening before setting fragment shaders.
- svga: Fix stencil op mapping
- svga: fix the rasterizer state resets

4.239 Mesa 8.0.1 Release Notes / February 16, 2012

Mesa 8.0.1 is a bug fix release which fixes bugs found since the 8.0 release.

Mesa 8.0 implements the OpenGL 3.0 API, but the version reported by glGetString(GL_VERSION) depends on the particular driver being used. Some drivers don't support all the features required in OpenGL 3.0.

See the *Compiling/Installing page* for prerequisites for DRI hardware acceleration.

4.239.1 MD5 checksums

```
4855c2d93bd2ebd43f384bdcc92c9a27  MesaLib-8.0.1.tar.gz
24eeebf66971809d8f40775a379b36c9  MesaLib-8.0.1.tar.bz2
54e745d14dac5717f7f65b4e2d5c1df2  MesaLib-8.0.1.zip
```

4.239.2 New features

None.

4.239.3 Bug fixes

This list is likely incomplete.

- Bug 28924 - [ILK] piglit tex-border-1 fail
- Bug 40864 - [bisected pineview] oglc pxconv-gettex(basic.allCases) fails on pineview
- Bug 43327 - [bisected SNB] HiZ make many oglc cases regressed
- Bug 44333 - [bisected] Color distortion with xbmc mediaplayer
- Bug 44927 - [SNB IVB regression] gl-117 abort when click
- Bug 45221 - [bisected IVB] glean/fbo regression in stencil-only case
- Bug 45877 - main/image.c:1597: _mesa_convert_colors: Assertion 'dstType == 0x1406' failed.
- Bug 45578 - main/image.c:1659: _mesa_convert_colors: Assertion 'dstType == 0x1403' failed.
- Bug 45872 - [bisected PNV] oglc mustpass(basic.stipple) regressed on pineview
- Bug 45876 - [PNV]oglc texenv(basic.allCases) regressed on pineview
- Bug 45917 - [PNV] Regression in Piglit test general/two-sided-lighting-separate-specular
- Bug 45943 - [r300g] r300_emit.c:365:r300_emit_aa_state: Assertion '(aa-d>dest)->cs_buf' failed.

4.239.4 Changes

The full set of changes can be viewed by using the following GIT command:

```
git log mesa-8.0..mesa-8.0.1
```

Alex Deucher (2):

- r600g: fix tex tile_type offset for cayman
- r600g: 128 bit formats require tile_type = 1 on cayman

Anuj Phogat (2):

- meta: Add pixel store/pack operations in decompress_texture_image
- meta: Avoid FBO resizing/reallocating in decompress_texture_image

Brian Paul (6):

- docs: add news item for 8.0 release
- docs: update info about supported systems, GPUs, APIs
- docs: add VMware link
- docs: remove link to the GLSL compiler page
- mesa: fix proxy texture target initialization
- swrast: fix span color type selection

Chad Versace (2):

- i965: Rewrite the HiZ op
- i965: Remove file i965/junk, accidentally added in 7b36c68

Dave Airlie (1):

- st/mesa: only resolve if number of samples is > 1

Eric Anholt (3):

- i965: Fix HiZ change compiler warning.
- i965: Report the failure message when failing to compile the fragment shader.
- i965/fs: Enable register spilling on gen7 too.

Ian Romanick (4):

- docs: Add 8.0 MD5 checksums
- glapi: Include GLES2 headers for ES2 extension functions
- swrast: Only avoid empty _TexEnvPrograms
- mesa: Bump version number to 8.0.1

Kenneth Graunke (4):

- i965: Fix border color on Ironlake.
- i965/fs: Add a new fs_inst::regs_written function.
- i965/fs: Take # of components into account in try_rewrite_rhs_to_dst.
- i965: Emit Ivybridge VS workaround flushes.

Mathias Fröhlich (1):

- state_stracker: Fix access to uninitialized memory.

Paul Berry (1):

- i915: Fix type of "specoffset" variable.

Simon Farnsworth (1):

- r600g: Use a fake reloc to sleep for fences

4.240 Mesa 8.0 Release Notes / February 9, 2012

Mesa 8.0 is a new development release. People who are concerned with stability and reliability should stick with a previous release or wait for Mesa 8.0.1.

Mesa 8.0 implements the OpenGL 3.0 API, but the version reported by glGetString(GL_VERSION) depends on the particular driver being used. Some drivers don't support all the features required in OpenGL 3.0.

See the *Compiling/Installing page* for prerequisites for DRI hardware acceleration.

4.240.1 MD5 checksums

```
3516fea6c28ce4a0fa9759e4894729a1  MesaLib-8.0.tar.gz
1a5668fe72651a670611164cefc703b2  MesaLib-8.0.tar.bz2
66f5a01a85530a91472a3acceb556db8  MesaLib-8.0.zip
```

4.240.2 New features

- GL_ARB_ES2_compatibility (r300g, r600g)
- GL_ARB_depth_buffer_float (r600g)
- GL_ARB_vertex_type_2_10_10_10_rev (r600g)
- GL_ARB_texture_storage (gallium drivers and swrast)
- GL_EXT_packed_float (i965)
- GL_EXT_texture_array (r600g, i965)
- GL_EXT_texture_shared_exponent (i965)
- GL_NV_fog_distance (all gallium drivers, nouveau classic)
- GL_NV_primitive_restart (r600g)
- GL_OES_EGL_image_external (gallium drivers)
- GL_OES_compressed_ETC1_RGB8_texture (softpipe, llvmpipe)
- ARB_texture_rgb10_a2ui (softpipe, r600g)
- Many updates to the VMware svga Gallium driver

4.240.3 Bug fixes

4.240.4 Changes

- Removed all DRI drivers that did not support DRI2. Specifically, i810, mach64, mga, r128, savage, sis, tdfx, and unichrome were removed.
- Removed support for BeOS.
- Removed the obsolete (and unmaintained) Windows "gldirect" and "ICD" drivers.
- Removed the linux-fbdev software driver.
- Removed all remnants of paletted texture support. As required by desktop OpenGL, `GL_COLOR_INDEX` data can still be uploaded to a color (e.g., RGBA) texture. However, the data cannot be stored internally as color-index.
- Removed support for GL_APPLE_client_storage extension.
- Removed the classic Mesa r300 and r600 drivers, which are superseded by the gallium drivers for this hardware.
- Removed the dead Gallium i965, cell and failover drivers, which were either broken and with nobody in sight to fix the situation or deprecated.

4.241 Mesa 7.11.2 Release Notes / November 27, 2011

Mesa 7.11.2 is a bug fix release which fixes bugs found since the 7.11 release.

Mesa 7.11 implements the OpenGL 2.1 API, but the version reported by glGetString(GL_VERSION) depends on the particular driver being used. Some drivers don't support all the features required in OpenGL 2.1.

See the *Compiling/Installing page* for prerequisites for DRI hardware acceleration.

4.241.1 MD5 checksums

```
b9e84efee3931c0acbccd1bb5a860554  MesaLib-7.11.2.tar.gz
0837c52698fe3252369c3fdb5195afcc  MesaLib-7.11.2.tar.bz2
141273c274d12e0d2bafb497fe937da3  MesaLib-7.11.2.zip
39ae9926794794503815ffdc069521eb  MesaGLUT-7.11.2.tar.gz
35ca3a0b54cb6f9d2e0e4eae8f6bb95e  MesaGLUT-7.11.2.tar.bz2
f8705fcff2510b6c39cd27b575c05dba  MesaGLUT-7.11.2.zip
```

4.241.2 New features

None.

4.241.3 Bug fixes

This list is likely incomplete.

- Bug 43143 - Mesa 7.11.1 fails to build at main/dlist.c:4532 with error message: "format not a string literal and no format arguments"
- Incorrect handling of CopyTexImage from RGBA window to LA texture.

4.241.4 Changes

The full set of changes can be viewed by using the following GIT command:

```
git log mesa-7.11.1..mesa-7.11.2
```

Brian Paul (4):

- mesa: stop using ctx->Driver.CopyTexImage1D/2D() hooks
- mesa: fix format selection for meta CopyTexSubImage()
- docs: update news.html and relnotes.html for 7.11.1 release
- mesa: use format string in _mesa_error() call to silence warning

Chad Versace (3):

- intel: Simplify stencil detiling arithmetic
- intel: Fix region dimensions for stencil buffers received from DDX
- intel: Fix separate stencil in builtin DRI2 backend

Ian Romanick (3):

- docs: Add 7.11.1 release md5sums
- mesa: set version string to 7.11.2-devel
- mesa: Bump version to 7.11.2 (final)

4.242 Mesa 7.11.1 Release Notes / November 17, 2011

Mesa 7.11.1 is a bug fix release which fixes bugs found since the 7.11 release.

Mesa 7.11 implements the OpenGL 2.1 API, but the version reported by glGetString(GL_VERSION) depends on the particular driver being used. Some drivers don't support all the features required in OpenGL 2.1.

See the *Compiling/Installing page* for prerequisites for DRI hardware acceleration.

4.242.1 MD5 checksums

```
ac0181a4076770fb657c1169af43aa09  MesaLib-7.11.1.tar.gz
a77307102cee844ff6544ffa8fafeac1  MesaLib-7.11.1.tar.bz2
dfcb11516c1730f3981b55a65a835623  MesaLib-7.11.1.zip
2cb2b9ecb4fb7d1a6be69346ee886952  MesaGLUT-7.11.1.tar.gz
3f54e314290d4dacbab089839197080b  MesaGLUT-7.11.1.tar.bz2
5d66c7ee8c5cc2f27e1ffb037ad4172c  MesaGLUT-7.11.1.zip
```

4.242.2 New features

None.

4.242.3 Bug fixes

This list is likely incomplete.

- Bug 3165 - texImage.IsCompressed and texImage.CompressedSize issues
- Bug 23525 - Software rendering on QEMU guests badly broken
- Bug 28125 - DRI2 prevents indirect glx
- Bug 34628 - [ilk] skybox errors in quake4
- Bug 36371 - r200: piglit readPixSanity failure
- Bug 36669 - EmitNoMainReturn set to 1 doesn't make the GLSL compiler lower all the RET opcodes
- Bug 36939 - multitexturing is messed up in quake wars (regression)
- Bug 37907 - [swrast] SIGSEGV swrast/s_depth.c:569
- Bug 38163 - Gnome Shell Display Bug
- Bug 38625 - ast_to_hir.cpp:1761: const glsl_type* process_array_type(YYLTYPE*, const glsl_type*, ast_node*, _mesa_glsl_parse_state*): Assertion 'dummy_instructions.is_empty()' failed.
- Bug 38729 - [softpipe] sp_quad_depth_test.c:215:convert_quad_stencil: Assertion '0' failed.
- Bug 38863 - [IVB]GPU hang when running 3D games like openarena
- Bug 39193 - [llvmpipe and r600g] glCheckFramebufferStatusEXT segfaults in Gallium when checking status on a framebuffer bound to a texture that's bound to a pixmap
- Bug 39651 - [glsl] Assertion failure when implicitly converting out parameters
- Bug 39991 - [regression]GL_PALETTE8_RGBA8_OES format of glCompressedTexImage2D will cause err GL_INVALID_ENUM with GLES1.x
- Bug 40022 - [i915] out-of-bounds write src/mesa/drivers/dri/i915/i915_fragprog.c:321
- Bug 40062 - in etqw the strogg radar is black (regression)
- Bug 40324 - [SNB] gpu hang in mesa 7.11
- Bug 40533 - i915: piglit glean/readPixSanity: DRI2SwapBuffers: BadDrawable (invalid Pixmap or Window parameter)
- Bug 41096 - [sandybridge-m-gt2+] GPU lockup render.IPEHR: 0x7a000002
- Bug 41969 - The Mesa meta save/restore code doesn't always save the active program
- Bug 42175 - RV730: Display errors in glxgears & WebGL
- Bug 42268 - [bisected] oglc pbo(negative.invalidOffsetValue) aborts on 7.11 branch

4.242.4 Changes

The full set of changes can be viewed by using the following GIT command:

```
git log mesa-7.11..mesa-7.11.1
```

Adam Jackson (2):

- drisw: Remove cargo culting that breaks GLX 1.3 ctors
- glx: Don't enable INTEL_swap_event unconditionally

Alex Deucher (1):

- r600g: fix up vs export handling

Ben Widawsky (1):

- intel: GetBuffer fix

Brian Paul (15):

- docs: add 7.11 md5 sums
- docs: news item for 7.11 release
- st/mesa: Convert size assertions to conditionals in st_texture_image_copy.
- softpipe: add missing stencil format case in convert_quad_stencil()
- mesa: fix texstore addressing bugs for depth/stencil formats
- mesa: add missing breaks for GL_TEXTURE_CUBE_MAP_SEAMLESS queries
- swrast: don't try to do depth testing if there's no depth buffer
- meta: fix/add checks for GL_EXT_framebuffer_sRGB
- mesa: fix PACK_COLOR_5551(), PACK_COLOR_1555() macros
- meta: fix broken sRGB mipmap generation
- mesa: add _NEW_CURRENT_ATTRIB in _mesa_program_state_flags()
- mesa: fix error handling for dlist image unpacking
- mesa: generate GL_INVALID_OPERATION in glIsEnabledIndex() between Begin/End
- mesa: fix incorrect error code in _mesa_FramebufferTexture1D/3DEXT()
- mesa: fix format/type check in unpack_image() for bitmaps

Carl Simonson (1):

- i830: Add missing vtable entry for i830 from the hiz work.

Carl Worth (5):

- glcpp: Fix two (or more) successive applications of token pasting
- glcpp: Test a non-function-like macro using the token paste operator
- glcpp: Implement token pasting for non-function-like macros
- glcpp: Raise error if defining any macro containing two consecutive underscores
- glcpp: Add a test for #elif with an undefined macro.

Chad Versace (5):

- glsl: Add method glsl_type::can_implicitly_convert_to()
- glsl: Fix implicit conversions in non-constructor function calls
- glsl: Remove ir_function.cpp:type_compare()
- glsl: Fix conversions in array constructors
- x86-64: Fix compile error with clang

Chia-I Wu (3):

- glsl: empty declarations should be valid

- intel: rename intel_extensions_es2.c to intel_extensions_es.c
- intel: fix GLESv1 support

Chris Wilson (1):

- i915: out-of-bounds write in calc_live_regs()

Christopher James Halse Rogers (1):

- glx/dri2: Paper over errors in DRI2Connect when indirect

David Reveman (1):

- i915g: Fix off-by-one in scissors.

Eric Anholt (16):

- mesa: Don't skip glGetProgramEnvParam4dvARB if there was already an error.
- mesa: Fix glGetUniform() type conversions.
- mesa: Add support for Begin/EndConditionalRender in display lists.
- mesa: Throw an error instead of asserting for condrender with query == 0.
- mesa: Throw an error when starting conditional render on an active query.
- mesa: Don't skip glGetProgramLocalParam4dvARB if there was already an error.
- glsl: Allow ir_assignment() constructor to not specify condition.
- glsl: Clarify error message about whole-array assignment in GLSL 1.10.
- glsl: When assigning to a whole array, mark the array as accessed.
- glsl: When assiging from a whole array, mark it as used.
- i965/fs: Respect ARB_color_buffer_float clamping.
- i965: Add missing _NEW_POLYGON flag to polygon stipple upload.
- i965: Fix polygon stipple offset state flagging.
- intel: Mark MESA_FORMAT_X8_Z24 as always supported.
- mesa: Don't error on glFeedbackBuffer(size = 0, buffer = NULL)
- glsl: Fix gl_NormalMatrix swizzle setup to match i965's invariants.

Henri Verbeet (6):

- mesa: Also set the remaining draw buffers to GL_NONE when updating just the first buffer in _mesa_drawbuffers().
- r600g: Support the PIPE_FORMAT_R16_FLOAT colorformat.
- mesa: Check the texture against all units in unbind_texobj_from_texunits().
- mesa: Allow sampling from units >= MAX_TEXTURE_UNITS in shaders.
- mesa: Use the Elements macro for the sampler index assert in validate_samplers().
- mesa: Fix a couple of TexEnv unit limits.

Ian Romanick (17):

- mesa: Add utility function to get base format from a GL compressed format
- mesa: Return the correct internal fmt when a generic compressed fmt was used

- mesa: Make _mesa_get_compressed_formats match the texture compression specs
- linker: Make linker_error set LinkStatus to false
- linker: Make linker_{error,warning} generally available
- mesa: Ensure that gl_shader_program::InfoLog is never NULL
- ir_to_mesa: Use Add linker_error instead of fail_link
- ir_to_mesa: Emit warnings instead of errors for IR that can't be lowered
- i915: Fail without crashing if a Mesa IR program uses too many registers
- i915: Only emit program errors when INTEL_DEBUG=wm or INTEL_DEBUG=fallbacks
- mesa: Add GL_OES_compressed_paletted_texture formats to _mesa_is_compressed_format
- mesa: Add GL_OES_compressed_paletted_texture formats to _mesa_base_tex_format
- mesa: Refactor expected texture size check in cpal_get_info
- mesa: Add _mesa_cpal_compressed_format_type
- mesa: Refactor compressed texture error checks to work with paletted textures
- mesa: Remove redundant compressed paletted texture error checks
- mesa: Advertise GL_OES_compressed_paletted_texture in OpenGL ES1.x

Jeremy Huddleston (3):

- apple: Silence some debug spew
- apple: Use the correct (OpenGL.framework) glViewport and glScissor during init
- apple: Implement applegl_unbind_context

José Fonseca (1):

- docs: Update llvmpipe docs.

Kenneth Graunke (12):

- glsl: Avoid massive ralloc_strndup overhead in S-Expression parsing.
- mesa: In validate_program(), initialize errMsg for safety.
- i965/gen5+: Fix incorrect miptree layout for non-power-of-two cubemaps.
- i965: Use proper texture alignment units for cubemaps on Gen5+.
- i965: Fix incorrect maximum PS thread count shift on Ivybridge.
- i965: Emit depth stalls and flushes before changing depth state on Gen6+.
- i965/fs: Allow SIMD16 with control flow on Ivybridge.
- i965: Allow SIMD16 color writes on Ivybridge.
- i965: Fix inconsistent indentation in brw_eu_emit.c.
- intel: Depth format fixes
- i965: Apply post-sync non-zero workaround to homebrew workaround.
- mesa/get: Move MAX_LIGHTS from GL/ES2 to GL/ES1.

Kristian Høgsberg (1):

- glx: Don't flush twice if we fallback to dri2CopySubBuffer

Marc Pignat (1):

- drisw: Fix 24bpp software rendering, take 2

Marcin Baczyński (2):

- configure: fix gcc version check
- configure: allow C{,XX}FLAGS override

Marcin Slusarz (3):

- nouveau: fix nouveau_fence leak
- nouveau: fix crash during fence emission
- nouveau: fix fence hang

Marek Olšák (19):

- vbo: do not call _mesa_max_buffer_index in debug builds
- winsys/radeon: fix space checking
- r300/compiler: fix a warning that a variable may be uninitialized
- r300/compiler: remove an unused-but-set variable and simplify the code
- u_vbuf_mgr: cleanup original vs real vertex buffer arrays
- u_vbuf_mgr: don't take per-instance attribs into acc. when computing max index
- u_vbuf_mgr: fix max_index computation for large src_offset
- u_vbuf_mgr: s/u_vbuf_mgr_/u_vbuf_
- u_vbuf_mgr: remove unused flag U_VBUF_UPLOAD_FLUSHED
- u_vbuf_mgr: rework user buffer uploads
- u_vbuf_mgr: fix uploading with a non-zero index bias
- configure.ac: fix xlib-based softpipe build
- r600g: add index_bias to index buffer bounds
- r300g: fix rendering with a non-zero index bias in draw_elements_immediate
- Revert "r300g: fix rendering with a non-zero index bias in draw_elements_immediate"
- pb_bufmgr_cache: flush cache when create_buffer fails and try again
- r300g: don't return NULL in resource_from_handle if the resource is too small
- r600g: set correct tiling flags in depth info
- r300g: don't call u_trim_pipe_prim in r300_swtcl_draw_vbo

Michel Dänzer (4):

- st/mesa: Finalize texture on render-to-texture.
- glx/dri2: Don't call X server for SwapBuffers when there's no back buffer.
- gallium/util: Add macros for converting from little endian to CPU byte order.
- r300g: Fix queries on big endian hosts.

Neil Roberts (1):

- meta: Fix saving the active program

Paul Berry (18):

- glsl: Lower unconditional return statements.
- glsl: Refactor logic for determining whether to lower return statements.
- glsl: lower unconditional returns and continues in loops.
- glsl: Use foreach_list in lower_jumps.cpp
- glsl: In lower_jumps.cpp, lower both branches of a conditional.
- glsl: Lower break instructions when necessary at the end of a loop.
- glsl: improve the accuracy of the radians() builtin function
- glsl: improve the accuracy of the atan(x,y) builtin function.
- Revert "glsl: Skip processing the first function's body in do_dead_functions()."
- glsl: Emit function signatures at toplevel, even for built-ins.
- glsl: Constant-fold built-in functions before outputting IR
- glsl: Check array size is const before asserting that no IR was generated.
- glsl: Perform implicit type conversions on function call out parameters.
- glsl: Fix type error when lowering integer divisions
- glsl: Rework oversize array check for gl_TexCoord.
- glsl: Remove field array_lvalue from ir_variable.
- glsl hierarchical visitor: Do not overwrite base_ir for parameter lists.
- glsl: improve the accuracy of the asin() builtin function.

Tobias Droste (1):

- r300/compiler: simplify code in peephole_add_presub_add

Tom Fogal (1):

- Only use gcc visibility support with gcc4+.

Tom Stellard (1):

- r300/compiler: Fix regalloc for values with multiple writers

Vadim Girlin (5):

- st/mesa: flush bitmap cache on query and conditional render boundaries
- r600g: use backend mask for occlusion queries
- r600g: take into account force_add_cf in pops
- r600g: fix check_and_set_bank_swizzle
- r600g: fix replace_gpr_with_pv_ps

Yuanhan Liu (17):

- i965: fix the constant interp bitmask for flat mode
- mesa: fix error handling for glEvalMesh1/2D
- mesa: fix error handling for some glGet* functions
- mesa: fix error handling for glTexEnv

- mesa: fix error handling for glIsEnabled
- mesa: fix error handling for glPixelZoom
- mesa: fix error handling for glSelectBuffer
- mesa: fix error handling for glMapBufferRange
- mesa: fix error handling for glMaterial*
- intel: fix the wrong code to detect null texture.
- mesa: add a function to do the image data copy stuff for save_CompressedTex(Sub)Image
- i965: setup address rounding enable bits
- mesa: generate error if pbo offset is not aligned with the size of specified type
- mesa: fix inverted pbo test error at _mesa_GetnCompressedTexImageARB
- mesa: handle the pbo case for save_Bitmap
- mesa: handle PBO access error in display list mode
- intel: don't call unmap pbo if pbo is not mapped

4.243 Mesa 7.11 Release Notes / July 31, 2011

Mesa 7.11 is a new development release. People who are concerned with stability and reliability should stick with a previous release or wait for Mesa 7.11.1.

Mesa 7.11 implements the OpenGL 2.1 API, but the version reported by glGetString(GL_VERSION) depends on the particular driver being used. Some drivers don't support all the features required in OpenGL 2.1.

See the *Compiling/Installing page* for prerequisites for DRI hardware acceleration.

4.243.1 MD5 checksums

```
fa2c7068503133fb2453244cda11cb2a  MesaLib-7.11.tar.gz
ff03aca82d0560009a076a87c888cf13  MesaLib-7.11.tar.bz2
ede1ac0976f6f05df586093fc17d63ed  MesaLib-7.11.zip
b4fb81a47c5caedaefad49af7702c23d  MesaGLUT-7.11.tar.gz
77a9a0bbd7f8bca882aa5709b88cb071  MesaGLUT-7.11.tar.bz2
c19ef0c6eb61188c96ed4ccedd70717c  MesaGLUT-7.11.zip
```

4.243.2 New features

- GL_ARB_ES2_compatibility (gallium drivers)
- GL_ARB_color_buffer_float (gallium drivers, i965)
- GL_ARB_draw_buffers_blend (gallium)
- GL_ARB_draw_instanced extension (gallium drivers, swrast)
- GL_ARB_instanced_arrays extension (gallium drivers)
- GL_ARB_occlusion_query2 (gallium drivers, swrast)
- GL_ARB_robustness (all drivers)

- GL_ARB_sampler_objects (gallium drivers)
- GL_ARB_seamless_cube_map (gallium r600)
- GL_ARB_shader_texture_lod (gallium drivers, i965)
- GL_ARB_sync (gallium drivers only, intel support was in 7.6)
- GL_ARB_texture_compression_rgtc (gallium drivers, swrast, i965)
- GL_ARB_texture_float (gallium, i965)
- GL_EXT_packed_float (gallium r600)
- GL_EXT_texture_compression_latc (gallium drivers, swrast)
- GL_EXT_texture_compression_rgtc (gallium drivers, swrast, i965)
- GL_EXT_texture_filter_anisotropic (swrast)
- GL_EXT_texture_shared_exponent (gallium drivers, swrast)
- GL_EXT_texture_sRGB_decode (gallium drivers, swrast, i965)
- GL_EXT_texture_snorm (gallium drivers)
- GL_AMD_draw_buffers_blend (alias of the ARB variant)
- GL_AMD_seamless_cubemap_per_texture (gallium r600)
- GL_AMD_shader_stencil_export (alias of the ARB variant)
- GL_ATI_draw_buffers (all drivers)
- GL_ATI_texture_compression_3dc (gallium drivers, swrast)
- GL_ATI_texture_float (gallium, i965)
- GL_NV_conditional_render (i965)
- GL_NV_texture_barrier (gallium drivers)
- Enable 16-wide fragment shader execution in i965 driver. This should improve performance in many applications.
- Initial alpha-level support for Intel "Ivybridge" chipsets in the i965 driver.

4.243.3 Bug fixes

This list is likely incomplete. This list only includes bug fixes not included in the previous release (7.10.3). Many of these are regressions that did not exist in the 7.10 release series at all.

- Bug 25871 - nearest neighbour samples wrong texel (precision/rounding problem)
- Bug 29162 - mesa/darwin is severly broken
- Bug 30080 - [i915] piglit nodepth-test fails
- Bug 30217 - Possible sources of memory leaks reported by valgrind
- Bug 30266 - Regression, segfault in libdrm_intel when calling glBitmap
- Bug 31561 - [i915] intel_regions.c:289: intel_region_release: Assertion 'region->map_refcount == 0' failed.
- Bug 31744 - [GLSL] overriding built-in function impacts another shader
- Bug 32308 - [llvmpipe] src/gallium/auxiliary/gallivm/lp_bld_init.c:319:gallivm_register_garbage_collector_callback: Assertion 'NumCallbacks < 32' failed.

- Bug 32309 - [softpipe] SIGSEGV sp_state_derived.c:204
- Bug 32459 - [softpipe] glean depthStencil regression
- Bug 32460 - [softpipe] piglit texwrap 1D regression
- Bug 32534 - [arrandale/sandybridge] Mesa swallowing GEM ioctl failures
- Bug 32560 - To fix: 64-bit-portabilty-issue state_tracker/st_program.c:427
- Bug 32634 - [r300g, bisected] Massive corruption in Unigine Sanctuary
- Bug 32768 - VBO rendering using glDrawArrays causes program termination and "cs IB too big" message
- Bug 32804 - [swrast] glean pixelFormats regression
- Bug 32814 - Build error in osmesa.c due to change in _mesa_update_framebuffer_visual() signature
- Bug 32859 - Mesa doesn't compile under NetBSD
- Bug 32912 - [bisected, r300g] Unigine Sanctuary: r300_emit.c:902:r300_emit_vertex_arrays: Assertion '&buf->b.b' failed with RADEON_HYPERZ=1
- Bug 32945 - [RADEON:KMS:R300G] HiZ: Weird behavior with 3 pipes
- Bug 33046 - [bisected]glean/pixelFormats and 3 oglc cases segfault
- Bug 33185 - [RADEON:KMS:R300G] X crashes when kwin effects are turned on
- Bug 33215 - [llvmpipe] piglit fbo-drawbuffers2-blend regression
- Bug 33247 - [swrast] tnl/t_draw.c:471: _tnl_draw_prims: Assertion 'prim[i].num_instances > 0' failed
- Bug 33284 - [llvmpipe] piglit fbo-drawbuffers-fragcolor fails
- Bug 33306 - [glsl] GLSL integer division by zero crashes GLSL compiler
- Bug 33353 - [softpipe] piglit fbo-srgb looks incorrect
- Bug 33360 - inclusion of $(TALLOC_LIBS) in src/mesa/drivers/osmesa/Makefile causes a build failure
- Bug 33374 - [bisect] FTBFS on commit 9767d3b5 (glapi: Fix OpenGL ES 1.1 and 2.0 interop)
- Bug 33508 - [glsl] GLSL compiler modulus by zero crash
- Bug 33555 - [softpipe] tgsi/tgsi_sse2.c:1527:emit_tex: Assertion '0' failed.
- Bug 33885 - [glsl] GLSL compiler allows recursion
- Bug 33823 - [glsl] ralloc.c:78: get_header: Assertion 'info->canary == 0x5A1106' failed.
- Bug 33934 - 3D blitting is orders of magnitude slower than equivalent 2D blitting.
- Bug 33946 - Crash: Mesa checks for invalid pointer, then uses it anyway.
- Bug 34008 - r600g: piglit failure (regression)
- Bug 34009 - Automatic Mipmap Generation produces very blurry image.
- Bug 34042 - Surfaceless eglMakeCurrent() fails if the supplied EGLContext is not a dummy context
- Bug 34049 - r600g: assertion failure (regression)
- Bug 34119 - [glsl] piglit glsl-texcoord-array regression
- Bug 34323 - [i915 GLSL gles2] gl_FragCoord.w not correct
- Bug 34346 - src/glsl relies on $PWD which can be unset
- Bug 34378 - st/mesa: 2a904fd6a0cb80eec6dec2bae07fd8778b04caf3 breaks sauerbraten

- Bug 34419 - Kwin crashes screensaver exits
- Bug 34463 - state_tracker/st_texture.c:370:st_texture_image_copy: Assertion 'u_minify(src->width0, srcLevel) == width' failed.
- Bug 34468 - src/glsl/Makefile fix
- Bug 34541 - [ilk, wine] massive render corruption after recent patchset
- Bug 34589 - [pineview bisected]many cases regression
- Bug 34595 - [bisected piketon]oglc half_float_vertex(misc.fillmode.wireframe) regressed
- Bug 34597 - [bisected piketon]oglc blend-constcolor and 7 draw-buffers2 subcases regressed
- Bug 34603 - [bisected piketon]oglc vbo subcase basic.bufferdata regressed
- Bug 34604 - [bisected piketon]piglit fbo/fbo-depth-sample-compare regressed
- Bug 34646 - [bisected piketon]ogles2conform GL2Tests/GL/gl_FragCoord/gl_FragCoord_w_frag.test
- Bug 34656 - i965: Crash when running WebGL Conformance Test in firefox-4 nightly build
- Bug 34691 - [GLSL] matrix array member assignment with a complex subscript fails
- Bug 34906 - [Pineview] Some WebGL conformance tests will crash firefox
- Bug 34968 - Bad fps in Lightsmark benchmark
- Bug 35025 - [Patch] Serious compiler warnings
- Bug 35312 - r600g: Automatic mipmap generation doesn't work properly
- Bug 35373 - [[GM45] OpenGL] GL_EXT_texture_sRGB_decode broken
- Bug 35434 - [RADEON:KMS:R600G] etqw: broken ground textures
- Bug 35441 - [PATCH] Mesa does not find nouveau include files with –enable-shared-dricore
- Bug 35614 - [SNB] random hang on piglit case shaders/glsl-max-varyings
- Bug 35820 - [bisected SNB] System hangs when Gnome with compiz start up
- Bug 35822 - [bisected pineview] many cases related to depth and stencil failed
- Bug 35849 - when sampling textures from both fragment and vertex shaders the vertex texture has the incorrect texture bound
- Bug 35852 - [bisected pineview] oglc case pxconv-read failed
- Bug 35961 - src/gallium/auxiliary/util/u_draw.c:77:util_draw_max_index: Assertion 'buffer_size - format_size <= buffer_size' failed.
- Bug 36032 - piglit fdo9833 regression
- Bug 36033 - main/shaderapi.c:1044: validate_samplers: Assertion '(sizeof(targetName)/sizeof(*(targetName))) == NUM_TEXTURE_TARGETS' failed.
- Bug 36086 - [wine] Segfault r300_resource_copy_region with some wine apps and RADEON_HYPERZ
- Bug 36182 - Game Trine from https://www.humblebundle.com/ needs ATI_draw_buffers
- Bug 36268 - [r300g, bisected] minor flickering in Unigine Sanctuary
- Bug 36282 - 34a5d3b9f4740601708c82093e2114356d749e65: glxgears segfaults when compiled with shared glapi
- Bug 36333 - can't build demos if mesa build with –enable-selinux
- Bug 36473 - [bisected] piglit bugs/fdo23670-depth_test failed

- Bug 36572 - [bisected]oglc case texenv segfaults
- Bug 36609 - 45920d2ecb38b14fdda5253fecce996570c22863 breaks sauerbraten on r300g
- Bug 36648 - [bisected SNB]piglit fbo/fbo-alphatest-nocolor failed
- Bug 36649 - [bisected SNB]oglc draw-buffers2 failed with 16-wide
- Bug 36753 - Some textures now rendered as completely black after register allocator rewrite.
- Bug 36821 - [bisected SNB]oglc api-texcoord causes GPU hang
- Bug 36914 - r600g: add rv670 flushing workaround. Causes games and some mesa demos to segfault.
- Bug 36917 - Rendering glitches in ETQW
- Bug 36939 - multitexturing is messed up in quake wars (regression)
- Bug 36987 - Intel GMA 4500 ARB_shader_texture_lod support
- Bug 37028 - Amnesia/HPL2 Demo: Strange graphical bugs on r600g
- Bug 37150 - sRGB textures are too bright in Starcraft 2
- Bug 37154 - main/texstore.c:4187: _mesa_texstore_rgb9_e5: Assertion 'baseInternalFormat == 0x1907' failed.
- Bug 37157 - [bisected] KDE KWin crashes on start with delayed BO mapping
- Bug 37168 - Regression: Severe memory leak when running Second Life
- Bug 37366 - [i965 bisected ILK] Fragment shader discard tests occasionally fail
- Bug 37383 - incorrect GLSL optimization
- Bug 37476 - [wine] Devil May Cry 4: TXD tgsi opcode unsupported / translation from TGSI failed / missing vertex shader
- Bug 37743 - [bisected i965]oglc GLSLlinker subcase negative.varying.beyondMaxVaryingFloats aborted
- Bug 37766 - Crash in dri2InvalidateBuffers when resizing Java window with OpenGL pipeline enabled
- Bug 37839 - main/teximage.c:2393: _mesa_choose_texture_format: Assertion 'f != MESA_FORMAT_NONE' failed.
- Bug 37934 - Corruption with topogun trace
- Bug 38015 - Some extensions enabled even when not supported by the underlying driver
- Bug 38134 - [bisected i965]piglit fbo/fbo-blit-d24s8 crashed
- Bug 38145 - r600g/evergreen: Incorrect rendering of some effects in doom3
- Bug 38440 - ETQW: Model in team select rendering too bright
- Bug 38566 - [regression] ETQW crashes with 21972c85ea734dbfcf69629c6b0b940efb42d4ba
- Bug 38584 - MESA_GLSL=dump causes SEGV in ir_print_visitor::unique_name
- Bug 38599 - THe value of WGL_PBUFFER_HEIGHT_ARB is returned as width in wglQueryPbufferARB
- Bug 38602 - [bisected] Wrong display after "prefer native texture formats when possible" commit
- Bug 38624 - program/ir_to_mesa.cpp:1440: virtual void ir_to_mesa_visitor::visit(ir_dereference_variable*): Assertion 'var->location != -1' failed.
- Bug 38626 - vbo: Don't discount stride breaks piglit on softpipe/r600g
- Bug 38649 - piglit fbo-copyteximage regression
- Bug 38762 - [IVB bisected]3D demos like glxgears abort

- Bug 38771 - [[GM45] DRI] GPU hangs with current Mesa GIT when running certain OpenGL applications
- Bug 38842 - Various valid GLX attributes are rejected by MESA glxChooseFBConfig
- Bug 38971 - [bisected]oglc glsl-autointconv subcase negative.function.ambiguousMatch failed
- Bug 38987 - sampler allowed as non-uniform / non-function parameters
- Bug 39024 - [Pineview webgl] many webgl conformance cases crash the browser
- Bug 39083 - [regression, bisected, r600g] Wrong rendering of Bubbles3D screensaver
- Bug 39119 - setting SQ_LDS_RESOURCE_MGMT register to zero in other applications muddles up font rendering permanently
- Bug 39209 - [bisected] Wrong display after "prefer native texture formats when possible" commit - part2
- Bug 39219 - libgl conflict with xbmc causes lock up on xbmc exit
- Bug 39257 - [bisected SNB]Mesa demos engine causes GPU hang
- Bug 39487 - [i965] brw_wm_surface_state.c:495: brw_update_renderbuffer_surface: Assertion 'brw->has_surface_tile_offset || (tile_x == 0 && tile_y == 0)' failed.
- Bug 39515 - FTBFS: libEGL depends on libgbm, but libEGL builds first
- Bug 39572 - Cogs: GPU hang

4.243.4 Changes

- The Windows MSVC project files have been removed. They haven't been maintained in quite a while. Building with SCons is an alternative.
- Removed GL_SGI_texture_color_table support from swrast driver - the only driver that implemented it.

The full set of changes can be viewed by using the following GIT command:

```
git log mesa-7.10..mesa-7.11
```

4.244 Mesa 7.10.3 Release Notes / June 13, 2011

Mesa 7.10.3 is a bug fix release which fixes bugs found since the 7.10.2 release.

Mesa 7.10.3 implements the OpenGL 2.1 API, but the version reported by glGetString(GL_VERSION) depends on the particular driver being used. Some drivers don't support all the features required in OpenGL 2.1.

See the *Compiling/Installing page* for prerequisites for DRI hardware acceleration.

4.244.1 MD5 checksums

```
d77b02034c11d6c2a55c07f82367d780  MesaLib-7.10.3.tar.gz
8c38fe8266be8e1ed1d84076ba5a703b  MesaLib-7.10.3.tar.bz2
614d063ecd170940d9ae7b355d365d59  MesaLib-7.10.3.zip
8768fd562ede7ed763d92b2d22232d7a  MesaGLUT-7.10.3.tar.gz
1496415b89da9549f0f3b34d9622e2e2  MesaGLUT-7.10.3.tar.bz2
1f29d0e7398fd3bf9f36f5db02941198  MesaGLUT-7.10.3.zip
```

4.244.2 New features

None.

4.244.3 Bug fixes

This list is likely incomplete.

- Bug 29162 - mesa/darwin is severly broken
- Bug 31590 - Black space between colors on mole hill example
- Bug 32395 - [glsl] Incorrect code generation for shadow2DProj() with bias
- Bug 32564 - [llvmpipe] prog: Unknown command line argument '-disable-mmx'. Try: 'prog -help' with llvm-2.9svn
- Bug 32835 - [glsl] recursive #define results in infinite stack recursion
- Bug 33303 - [glsl] ir_constant_expression.cpp:72: virtual ir_constant* ir_expression::constant_expression_value(): Assertion 'op[0]->type->base_type == op[1]->type->base_type' failed.
- Bug 33314 - [glsl] ir_constant_expression.cpp:122: virtual ir_constant* ir_expression::constant_expression_value(): Assertion 'op[0]->type->base_type == GLSL_TYPE_BOOL' failed.
- Bug 33512 - [SNB] case ogles2conform/GL/gl_FragCoord/gl_FragCoord_xy_frag.test and gl_FragCoord_w_frag.test fail
- Bug 34280 - r200 mesa-7.10 font distortion
- Bug 34321 - The ARB_fragment_program subset of ARB_draw_buffers not implemented
- Bug 35603 - GLSL compiler freezes compiling shaders
- Bug 36173 - struct renderbuffer's 'format' field never set when using FBO
- Bug 36238 - Mesa release files don't contain scons control files
- Bug 36410 - [SNB] Rendering errors in 3DMMES subtest taiji
- Bug 36527 - [wine] Wolfenstein: Failed to translate rgb instruction.
- Bug 36651 - mesa requires bison and flex to build but configure does not check for them
- Bug 36738 - Openarena crash with r300g, swrastg + llvm > 2.8
- Bug 37648 - Logic error in mesa/main/teximage.c:texsubimage
- Bug 37739 - Color clear of FBO without color buffer crashes

4.244.4 Changes

The full set of changes can be viewed by using the following GIT command:

```
git log mesa-7.10.2..mesa-7.10.3
```

Alan Hourihane (1):

- Check for out of memory when creating fence

Alex Buell (1):

- configure: bump LIBDRM_REQUIRED to 2.4.24

Alex Deucher (2):

- r600c: add new pci ids
- r600g: add new pci ids

Brian Paul (19):

- docs: add link to 7.10.2 release notes
- scons: remove dangling reference to state_trackers/python/SConscript
- Makefile: add missing Scons files
- llvmpipe: document issue with LLVM 2.8 and earlier with AVX
- docs: replace llvmpipe/README with docs/llvmpipe.html
- glsl: add static qualifier to silence warning
- glsl: add cast to silence signed/unsigned comparison warning
- mesa: s/height/depth/ in texsubimage()
- mesa: fix void pointer arithmetic warnings
- mesa: add some missing GLAPIENTRY keywords
- mesa: check that flex/bison are installed
- st/mesa: fix incorrect texture level/face/slice accesses
- draw: fix edge flag handling in clipper (for unfilled tris/quads/polygons)
- vbo: check array indexes to prevent negative indexing
- vbo: remove node->count > 0 test in vbo_save_playback_vertex_list()
- st/mesa: fix software accum buffer format bug
- mesa: add include/c99/inttypes.h include/c99/stdbool.h include/c99/stdint.h files to tarballs
- docs: 7.10.3 release notes skeleton file, links
- mesa: bump version to 7.10.3

Carl Worth (2):

- glcpp: Simplify calling convention of parser's active_list functions
- glcpp: Fix attempts to expand recursive macros infinitely (bug #32835).

Dave Airlie (1):

- st/mesa: fix compressed mipmap generation.

Eric Anholt (19):

- i965: Fix the VS thread limits for GT1, and clarify the WM limits on both.
- glsl: Avoid cascading errors when looking for a scalar boolean and failing.
- glsl: Semantically check the RHS of '&&' even when short-circuiting.
- glsl: Semantically check the RHS of '||' even when short-circuiting.
- glsl: When we've emitted a semantic error for ==, return a bool constant.
- glsl: Perform type checking on "^^" operands.

- intel: Use _mesa_base_tex_format for FBO texture attachments.
- swrast: Don't assert against glReadPixels of GL_RED and GL_RG.
- mesa: Add a gl_renderbuffer.RowStride field like textures have.
- mesa: Add a function to set up the default renderbuffer accessors.
- intel: Use Mesa core's renderbuffer accessors for depth.
- mesa: Use _mesa_get_format_bytes to refactor out the RB get_pointer_*
- mesa: Use _mesa_get_format_bytes to refactor out the RB get_row_*
- mesa: Add renderbuffer accessors for R8/RG88/R16/RG1616.
- swrast: Don't try to adjust_colors for <8bpc when handling R16, RG1616.
- intel: Use mesa core's R8, RG88, R16, RG1616 RB accessors.
- Revert "intel: Add spans code for the ARB_texture_rg support."
- mesa: Add support for the ARB_fragment_program part of ARB_draw_buffers.
- mesa: Add support for OPTION ATI_draw_buffers to ARB_fp.

Hans de Goede (1):

- texstore: fix regression stricter check for memcpy path for unorm88 and unorm1616

Henri Verbeet (3):

- mesa: Also update the color draw buffer if it's explicitly set to GL_NONE.
- glx: Destroy dri2Hash on DRI2 display destruction.
- glx: Only remove the glx_display from the list after it's destroyed.

Ian Romanick (9):

- docs: Add 7.10.2 md5sums
- glsl: Fix off-by-one error setting max_array_access for non-constant indexing
- ir_to_mesa: Handle shadow compare w/projection and LOD bias correctly
- intel: Fix ROUND_DOWN_TO macro
- glsl: Regenerate compiler and glcpp files from cherry picks
- i965: Remove hint_gs_always and resulting dead code
- mesa: Don't try to clear a NULL renderbuffer
- mesa: Ignore blits to/from missing buffers
- docs: Add list of bugs fixed in 7.10.3 release

Jeremy Huddleston (18):

- apple: Update GL specs
- apple: Rename glcontextmodes.[ch] to glxconfig.[ch]
- apple: Rename __GLcontextModes to struct glx_config
- apple: Rename GLXcontext
- apple: Re-add driContext and do_destroy
- apple: Rename _gl_context_modes_find_visual to glx_config_find_visual

- apple: Rename GLXcontext
- apple: Change from XExtDisplayInfo to struct glx_display
- apple: ifdef out come glapi-foo on darwin
- glx: Dead code removal
- apple: Build darwin using applegl rather than indirect
- apple: Fix build failures in applegl_glx.c
- darwin: Define GALLIUM_DRIVERS_DIRS in darwin config
- apple: Package applegl source into MesaLib tarball
- darwin: Set VG_LIB_{NAME,GLOB} to fix make install
- darwin: Don't link against libGL when building libOSMesa
- darwin: Fix VG_LIB_GLOB to also match the unversioned symlink
- osmesa: Fix missing symbols when GLX_INDIRECT_RENDERING is defined.

José Fonseca (13):

- llvmpipe: Update readme.
- mesa: GL_PROVOKING_VERTEX_EXT is a GLenum, not GLboolean.
- mesa: Fix GetVertexAttrib* inside display lists.
- draw: Fix draw_variant_output::format's type.
- gallivm: Tell LLVM to not assume a 16-byte aligned stack on x86.
- gallivm: Fix for dynamically linked LLVM 2.8 library.
- st/wgl: Adjust the pbuffer invisible window size.
- st/wgl: Fix debug output format specifiers of stw_framebuffer_get_size().
- st/wgl: Prevent spurious framebuffer sizes when the window is minimized.
- st/wgl: Cope with zero width/height windows.
- st/wgl: Allow to create pbuffers bigger than the desktop.
- st/wgl: Remove buggy assertion.
- wgl: Don't hold on to user supplied HDC.

Kenneth Graunke (10):

- i965/fs: Switch W and 1/W in Sandybridge interpolation setup.
- i965: Refactor Sandybridge implied move handling.
- i965: Resolve implied moves in brw_dp_READ_4_vs_relative.
- intel: Add IS_GT2 macro for recognizing Sandybridge GT2 systems.
- i965: Allocate the whole URB to the VS and fix calculations for Gen6.
- intel: Support glCopyTexImage() from ARGB8888 to XRGB8888.
- glsl: Fix memory error when creating the supported version string.
- glsl: Regenerate autogenerated file builtin_function.cpp.
- i965: Rename various gen6 #defines to match the documentation.

- i965: Never enable the GS on Gen6.

Kostas Georgiou (1):

- r600c/g: Add pci id for FirePro 2270

Marek Olšák (18):

- tgsi/ureg: bump the limit of immediates
- st/mesa: fix changing internal format via RenderbufferStorage
- st/mesa: GenerateMipmap should not be killed by conditional rendering
- swrast: BlitFramebuffer should not be killed by conditional rendering
- st/mesa: BlitFramebuffer should not be killed by conditional rendering
- st/mesa: CopyTex(Sub)Image should not be killed by conditional rendering
- st/mesa: conditional rendering should not kill texture decompression via blit
- mesa: forbid UseProgram to be called inside Begin/End
- mesa: UseShaderProgramEXT and Uniform* shouldn't be allowed inside Begin/End
- mesa: queries of non-existent FBO attachments should return INVALID_OPERATION
- r300g: fix draw_vbo splitting on r3xx-r4xx
- r300g: fix texturing with non-3D textures and wrap R mode set to sample border
- r300g: fix occlusion queries when depth test is disabled or zbuffer is missing
- r300g: clear can be killed by render condition
- st/mesa: remove asserts in st_texture_image_copy
- mesa: fix up assertion in _mesa_source_buffer_exists
- mesa: invalidate framebuffer if internal format of renderbuffer is changed
- mesa: return after invalidating renderbuffer

Matt Turner (1):

- r300/compiler: align memory allocations to 8-bytes

Tom Stellard (3):

- r300/compiler: Fix incorrect presubtract conversion
- r300/compiler: Fix dataflow analysis bug with ELSE blocks
- r300/compiler: Limit instructions to 3 source selects

Vinson Lee (1):

- gallivm: Disable MMX-disabling code on llvm-2.9.

Zou Nan hai (1):

- i965: Align interleaved URB write length to 2

pepp (1):

- st/mesa: assign renderbuffer's format field when allocating storage

4.245 Mesa 7.10.2 Release Notes / April 6, 2011

Mesa 7.10.2 is a bug fix release which fixes bugs found since the 7.10 release.

Mesa 7.10.2 implements the OpenGL 2.1 API, but the version reported by glGetString(GL_VERSION) depends on the particular driver being used. Some drivers don't support all the features required in OpenGL 2.1.

See the *Compiling/Installing page* for prerequisites for DRI hardware acceleration.

4.245.1 MD5 checksums

```
2f9f444265534a2cfd9a99d1a8291089  MesaLib-7.10.2.tar.gz
f5de82852f1243f42cc004039e10b771  MesaLib-7.10.2.tar.bz2
47836e37bab6fcafe3ac90c9544ba0e9  MesaLib-7.10.2.zip
175120325828f313621cc5bc6c504803  MesaGLUT-7.10.2.tar.gz
8c71d273f5f8d6c5eda4ffc39e0fe03e  MesaGLUT-7.10.2.tar.bz2
03036c8efe7b791a90fa0f2c41b43f43  MesaGLUT-7.10.2.zip
```

4.245.2 New features

None.

4.245.3 Bug fixes

This list is likely incomplete.

- Bug 29172 - Arrandale - Pill Popper Pops Pills
- Bug 31159 - shadow problem in 0ad game
- Bug 32688 - [RADEON:KMS:R300G] some games have a wireframe or outline visible
- Bug 32949 - [glsl wine] Need for Speed renders incorrectly with GLSL enabled
- Bug 34203 - [GLSL] fail to call long chains across shaders
- Bug 34376 - [GLSL] allowing assignment to unsized array
  - The commit message incorrectly lists bug 34367.
- Bug 34370 - [GLSL] "i<5 && i<4" in for loop fails
- Bug 34374 - [GLSL] fail to redeclare an array using initializer
- Bug 35073 - [GM45] Alpha test is broken when rendering to FBO with no color attachment
- Bug 35483 - util_blit_pixels_writemask: crash in line 322 of src/gallium/auxiliary/util/u_blit.c

4.245.4 Changes

The full set of changes can be viewed by using the following GIT command:

```
git log mesa-7.10.1..mesa-7.10.2
```

Note: Reverted commits and the reverts are not included in this list.

Alex Deucher (2):

- r600c: add new ontario pci ids
- r600g: add some additional ontario pci ids

Benjamin Franzke (1):

- st/dri: Fix surfaceless gl using contexts with previous bound surfaces

Brian Paul (9):

- docs: pull 7.9.2 release notes into 7.10 branch
- docs: update news.html with 7.10.1 and 7.9.2 releases
- docs: fill in 7.10.1 release data
- docs: add, fix release notes links
- docs: update info about Mesa packaging/contents
- docs: update prerequisites, remove old demo info
- mesa: Guard against null pointer deref in fbo validation
- st/mesa: Apply LOD bias from correct texture unit
- glsl: silence warning in printf() with a cast

Chad Versace (1):

- i965: Fix tex_swizzle when depth mode is GL_RED

Dave Airlie (1):

- r600: don't close fd on failed load

Eric Anholt (2):

- i965: Apply a workaround for the Ironlake "vertex flashing".
- i965: Fix alpha testing when there is no color buffer in the FBO.

Fabian Bieler (1):

- st/mesa: Apply LOD from texture object

Henri Verbeet (1):

- st/mesa: Validate state before doing blits.

Ian Romanick (13):

- docs: Add 7.10.1 md5sums
- glsl: Refactor AST-to-HIR code handling variable initializers
- glsl: Refactor AST-to-HIR code handling variable redeclarations
- glsl: Process redeclarations before initializers
- glsl: Function signatures cannot have NULL return type
- glsl: Add several function / call related validations
- linker: Add imported functions to the linked IR

- glsl: Use insert_before for lists instead of open coding it
- glsl: Only allow unsized array assignment in an initializer
- glcpp: Refresh autogenerated lexer files
- docs: Initial bits of 7.10.2 release notes
- mesa: set version string to 7.10.2
- mesa: Remove nonexistant files from _FILES lists

Jerome Glisse (1):

- r600g: move user fence into base radeon structure

José Fonseca (2):

- mesa: Fix typo glGet*v(GL_TEXTURE_COORD_ARRAY_*).
- mesa: More glGet* fixes.

Kenneth Graunke (4):

- glcpp: Rework lexer to use a SKIP state rather than REJECT.
- glcpp: Remove trailing contexts from #if rules.
- i965/fs: Fix linear gl_Color interpolation on pre-gen6 hardware.
- glsl: Accept precision qualifiers on sampler types, but only in ES.

Marek Olšák (15):

- st/mesa: fix crash when DrawBuffer->_ColorDrawBuffers[0] is NULL
- st/mesa: fail to alloc a renderbuffer if st_choose_renderbuffer_format fails
- r300/compiler: fix the saturate modifier when applied to TEX instructions
- r300/compiler: fix translating the src negate bits in pair_translate
- r300/compiler: Abs doesn't cancel Negate (in the conversion to native swizzles)
- r300/compiler: TEX instructions don't support negation on source arguments
- r300/compiler: do not set TEX_IGNORE_UNCOVERED on r500
- r300/compiler: saturate Z before the shadow comparison
- r300/compiler: fix equal and notequal shadow compare functions
- r300/compiler: remove unused variables
- st/mesa: fix crash when using both user and vbo buffers with the same stride
- r300g: fix alpha-test with no colorbuffer
- r300g: tell the GLSL compiler to lower the continue opcode
- r300/compiler: propagate SaturateMode down to the result of shadow comparison
- r300/compiler: apply the texture swizzle to shadow pass and fail values too

Michel Dänzer (1):

- Use proper source row stride when getting depth/stencil texels.

Tom Stellard (4):

- r300/compiler: Use a 4-bit writemask in pair instructions

- prog_optimize: Fix reallocating registers for shaders with loops
- r300/compiler: Fix vertex shader MAD instructions with constant swizzles
- r300/compiler: Don't try to convert RGB to Alpha in full instructions

4.246 Mesa 7.10.1 Release Notes / March 2, 2011

Mesa 7.10.1 is a bug fix release which fixes bugs found since the 7.10 release.

Mesa 7.10.1 implements the OpenGL 2.1 API, but the version reported by glGetString(GL_VERSION) depends on the particular driver being used. Some drivers don't support all the features required in OpenGL 2.1.

See the *Compiling/Installing page* for prerequisites for DRI hardware acceleration.

4.246.1 MD5 checksums

```
4b4cee19f3bf16eb78bd4cc278ccf812  MesaLib-7.10.1.tar.gz
efe8da4d80c2a5d32a800770b8ce5dfa  MesaLib-7.10.1.tar.bz2
0fd2b1a025934de3f8cecf9fb9b57f4c  MesaLib-7.10.1.zip
42beb0f5188d544476c19496f725fa67  MesaGLUT-7.10.1.tar.gz
637bb8a20fdad89f7382b4ea83f896e3  MesaGLUT-7.10.1.tar.bz2
bdbf3ffb2606d6aa8afabb6c6243b91b  MesaGLUT-7.10.1.zip
```

4.246.2 New features

None.

4.246.3 Bug fixes

This list is likely incomplete.

- Fix an off-by-one bug in a vsplit assertion.
- Fix incorrect handling of `layout` qualifier with `in`, `out`, `attribute`, and `varying`.
- Fix an i965 shader bug where the negative absolute value was generated instead of the absolute value of a negation.
- Fix numerous issues handling precision qualifiers in GLSL ES.
- Fixed a few GLX protocol encoder bugs (Julien Cristau)
- Assorted Gallium llvmpipe driver bug fixes
- Assorted Mesa/Gallium state tracker bug fixes
- Bug 26795 - gl_FragCoord off by one in Gallium drivers.
- Bug 29164 - [GLSL 1.20] invariant variable shouldn't be used before declaration
- Bug 29823 - GetUniform[if]v busted
- Bug 29927 - [glsl2] fail to compile shader with constructor for array of struct type
- Bug 30156 - [i965] After updating to Mesa 7.9, Civilization IV starts to show garbage
- Bug 31923 - [GLSL 1.20] allowing inconsistent centroid declaration between two vertex shaders

- Bug 31925 - [GLSL 1.20] "#pragma STDGL invariant(all)" fail
- Bug 32214 - [gles2]no link error happens when missing vertex shader or frag shader
- Bug 32375 - [gl gles2] Not able to get the attribute by function glGetVertexAttribfv
- Bug 32541 - Segmentation Fault while running an HDR (high dynamic range) rendering demo
- Bug 32569 - [gles2] glGetShaderPrecisionFormat not implemented yet
- Bug 32695 - [glsl] SIGSEGV glcpp/glcpp-parse.y:833
- Bug 32831 - [glsl] division by zero crashes GLSL compiler
- Bug 32910 - Keywords 'in' and 'out' not handled properly for GLSL 1.20 shaders
- Bug 33219 -[GLSL bisected] implicit sized array triggers segfault in ir_to_mesa_visitor::copy_propagate
- Bug 33306 - GLSL integer division by zero crashes GLSL compiler
- Bug 33308 -[glsl] ast_to_hir.cpp:3016: virtual ir_rvalue* ast_jump_statement::hir(exec_list*, _mesa_glsl_parse_state*): Assertion 'ret != __null' failed.
- Bug 33316 - uniform array will be allocate one line more and initialize it when it was freed will abort
- Bug 33386 - Dubious assembler in read_rgba_span_x86.S
- Bug 33388 - Dubious assembler in xform4.S
- Bug 33433 - Error in x86-64 API dispatch code.
- Bug 33507 - [glsl] GLSL preprocessor modulus by zero crash
- Bug 33508 - [glsl] GLSL compiler modulus by zero crash
- Bug 33916 - Compiler accepts reserved operators % and %=
- Bug 34030 - [bisected] Starcraft 2: some effects are corrupted or too big
- Bug 34047 - Assert in _tnl_import_array() when using GLfixed vertex datatypes with GLESv2
- Bug 34114 - Sun Studio build fails due to standard library functions not being in global namespace
- Bug 34179 - Nouveau 3D driver: nv50_pc_emit.c:863 assertion error kills Compiz
- Bug 34198 - [GLSL] implicit sized array with index 0 used gets assertion
- Ubuntu bug 691653 - compiz crashes when using alt-tab (the radeon driver kills it)
- Meego bug 13005 - Graphics GLSL issue lead to camera preview fail on Pinetrail

4.246.4 Changes

The full set of changes can be viewed by using the following GIT command:

```
git log mesa-7.10..mesa-7.10.1
```

Alberto Milone (1):

- r600c: add evergreen ARL support.

Brian Paul (21):

- draw: Fix an off-by-one bug in a vsplit assertion.
- docs: add links to 7.9.1 and 7.10 release notes
- docs: added news item for 7.9.1 and 7.10 release

- gallivm: work around LLVM 2.6 bug when calling C functions
- gallivm: fix copy&paste error from previous commit
- mesa: fix a few format table mistakes, assertions
- mesa: fix num_draw_buffers==0 in fixed-function fragment program generation
- mesa: don't assert in GetIntegerIndexed, etc
- mesa: check for dummy renderbuffer in _mesa_FramebufferRenderbufferEXT()
- llvmpipe: make sure binning is active when we begin/end a query
- st/mesa: fix incorrect fragcoord.x translation
- softpipe: fix off-by-one error in setup_fragcoord_coeff()
- cso: fix loop bound in cso_set_vertex_samplers()
- st/mesa: fix incorrect glCopyPixels position on fallback path
- st/mesa: set renderbuffer _BaseFormat in a few places
- st/mesa: fix the default case in st_format_datatype()
- st/mesa: need to translate clear color according to surface's base format
- docs: update 7.9.2 release notes with Brian's cherry-picks
- docs: add link to 7.10.1 release notes
- mesa: implement glGetShaderPrecisionFormat()
- docs: updated environment variable list

Bryce Harrington (1):

- r300g: Null pointer check for buffer deref in gallium winsys

Chad Versace (20):

- glsl: At link-time, check that globals have matching centroid qualifiers
- glcpp: Fix segfault when validating macro redefinitions
- glsl: Fix parser rule for type_specifier
- glsl: Change default value of ast_type_specifier::precision
- glsl: Add semantic checks for precision qualifiers
- glsl: Add support for default precision statements
- glsl: Remove redundant semantic check in parser
- glsl: Fix semantic checks on precision qualifiers
- glsl: Fix segfault due to missing printf argument
- glsl: Mark 'in' variables at global scope as read-only
- mesa: Refactor handling of extension strings
- mesa: Add/remove extensions in extension string
- mesa: Change dependencies of some OES extension strings
- mesa: Change OES_point_sprite to depend on ARB_point_sprite
- mesa: Change OES_standard_derivatives to be stand-alone extension

- i915: Disable extension OES_standard_derivatives
- glcpp: Raise error when modulus is zero
- glsl: Set operators '%' and '%=' to be reserved when GLSL < 1.30
- glsl: Reinstate constant-folding for division by zero
- tnl: Add support for datatype GL_FIXED in vertex arrays

Chia-I Wu (1):

- mesa: Add glDepthRangef and glClearDepthf to APIspec.xml.

Christoph Bumiller (1):

- nv50,nvc0: do not forget to apply sign mode to saved TGSI inputs

Cyril Brulebois (1):

- Point to bugs.freedesktop.org rather than bugzilla.freedesktop.org

Dave Airlie (3):

- radeon/r200: fix fbo-clearmipmap + gen-teximage
- radeon: calculate complete texture state inside TFP function
- radeon: avoid segfault on 3D textures.

Dimitry Andric (4):

- mesa: s/movzx/movzbl/
- mesa: s/movzxw/movzwl/ in read_rgba_span_x86.S
- glapi: adding @ char before type specifier in glapi_x86.S
- glapi: add @GOTPCREL relocation type

Eric Anholt (16):

- glsl: Fix the lowering of variable array indexing to not lose write_masks.
- i965/fs: When producing ir_unop_abs of an operand, strip negate.
- i965/vs: When MOVing to produce ABS, strip negate of the operand.
- i965/fs: Do flat shading when appropriate.
- i965: Avoid double-negation of immediate values in the VS.
- intel: Make renderbuffer tiling choice match texture tiling choice.
- i965: Fix dead pointers to fp->Parameters->ParameterValues[] after realloc.
- docs: Add a relnote for the Civ IV on i965.
- glapi: Add entrypoints and enums for GL_ARB_ES2_compatibility.
- mesa: Add extension enable bit for GL_ARB_ES2_compatibility.
- mesa: Add actual support for glReleaseShaderCompiler from ES2.
- mesa: Add support for glDepthRangef and glClearDepthf.
- mesa: Add getters for ARB_ES2_compatibility MAX_*_VECTORS.
- mesa: Add getter for GL_SHADER_COMPILER with ARB_ES2_compatibility.
- i965: Fix a bug in i965 compute-to-MRF.

- i965/fs: Add a helper function for detecting math opcodes.

Fredrik Höglund (1):

- st/mesa: fix a regression from cae2bb76

Ian Romanick (42):

- docs: Add 7.10 md5sums
- glsl: Support the 'invariant(all)' pragma
- glcpp: Generate an error for division by zero
- glsl: Add version_string containing properly formatted GLSL version
- glsl & glcpp: Refresh autogenerated lexer and parser files.
- glsl: Disallow 'in' and 'out' on globals in GLSL 1.20
- glsl: Track variable usage, use that to enforce semantics
- glsl: Allow 'in' and 'out' when 'layout' is also available
- docs: Initial bits of 7.10.1 release notes
- mesa: bump version to 7.10.1-devel
- doc: Update 7.10.1 release notes
- glsl: Emit errors or warnings when 'layout' is used with 'attribute' or 'varying'
- docs: Update 7.10.1 release notes
- glsl: Refresh autogenerated lexer and parser files.
- glsl: Don't assert when the value returned by a function has no rvalue
- linker: Set sizes for non-global arrays as well
- linker: Propagate max_array_access while linking functions
- docs: Update 7.10.1 release notes
- mesa: glGetUniform only returns a single element of an array
- linker: Generate link errors when ES shaders are missing stages
- mesa: Fix error checks in GetVertexAttrib functions
- Use C-style system headers in C++ code to avoid issues with std:: namespace
- docs: Update 7.10.1 release notes
- glapi: Regenerate for GL_ARB_ES2_compatibility.
- mesa: Connect glGetShaderPrecisionFormat into the dispatch table
- i965: Set correct values for range/precision of fragment shader types
- i915: Set correct values for range/precision of fragment shader types
- intel: Fix typeos from 3d028024 and 790ff232
- glsl: Ensure that all GLSL versions are supported in the stand-alone compiler
- glsl: Reject shader versions not supported by the implementation
- mesa: Initial size for secondary color array is 3
- glsl: Finish out the reduce/reduce error fixes

- glsl: Regenerate compiler and glcpp files from cherry picks
- linker: Fix off-by-one error implicit array sizing
- docs: update 7.10.1 release notes with Ian's recent cherry picks
- i915: Only mark a register as available if all components are written
- i915: Calculate partial result to temp register first
- i915: Force lowering of all types of indirect array accesses in the FS
- docs: Update 7.10.1 with (hopefully) the last of the cherry picks
- docs: Clean up bug fixes list
- intel: Remove driver date and related bits from renderer string
- mesa: set version string to 7.10.1 (final)

Jian Zhao (1):

- mesa: fix an error in uniform arrays in row calculating.

Julien Cristau (3):

- glx: fix request lengths
- glx: fix GLXChangeDrawableAttributesSGIX request
- glx: fix length of GLXGetFBConfigsSGIX

Keith Packard (1):

- glsl: Eliminate reduce/reduce conflicts in glsl grammar

Kenneth Graunke (20):

- glsl: Expose a public glsl_type::void_type const pointer.
- glsl: Don't bother unsetting a destructor that was never set.
- glsl, i965: Remove unnecessary talloc includes.
- glcpp: Remove use of talloc reference counting.
- ralloc: Add a fake implementation of ralloc based on talloc.
- Convert everything from the talloc API to the ralloc API.
- ralloc: a new MIT-licensed recursive memory allocator.
- Remove talloc from the make and automake build systems.
- Remove talloc from the SCons build system.
- Remove the talloc sources from the Mesa repository.
- glsl: Fix use of uninitialized values in _mesa_glsl_parse_state ctor.
- i965/fs: Apply source modifier workarounds to POW as well.
- i965: Fix shaders that write to gl_PointSize on Sandybridge.
- i965/fs: Avoid register coalescing away gen6 MATH workarounds.
- i965/fs: Correctly set up gl_FragCoord.w on Sandybridge.
- i965: Increase Sandybridge point size clamp.
- i965/fs: Refactor control flow stack handling.

- i965: Increase Sandybridge point size clamp in the clip state.
- glsl: Use reralloc instead of plain realloc.
- Revert "i965/fs: Correctly set up gl_FragCoord.w on Sandybridge."

Marek Olšák (4):

- docs: fix messed up names with special characters in relnotes-7.10
- docs: fix messed up names with special characters in relnotes-7.9.1
- mesa: fix texture3D mipmap generation for UNSIGNED_BYTE_3_3_2
- st/dri: Track drawable context bindings

Paulo Zanoni (1):

- dri_util: fail driCreateNewScreen if InitScreen is NULL

Sam Hocevar (2):

- docs: add glsl info
- docs: fix glsl_compiler name

Tom Fogal (1):

- Regenerate gl_mangle.h.

Tom Stellard (2):

- r300/compiler: Disable register rename pass on r500
- r300/compiler: Don't erase sources when converting RGB->Alpha

Vinson Lee (3):

- ralloc: Add missing va_end following va_copy.
- mesa: Move declaration before code in extensions.c.
- mesa: Move loop variable declarations outside for loop in extensions.c.

nobled (1):

- glx: Put null check before use

4.247 Mesa 7.10 Release Notes / January 7, 2011

Mesa 7.10 is a new development release. People who are concerned with stability and reliability should stick with a previous release or wait for Mesa 7.10.1.

Mesa 7.10 implements the OpenGL 2.1 API, but the version reported by glGetString(GL_VERSION) depends on the particular driver being used. Some drivers don't support all the features required in OpenGL 2.1.

See the *Compiling/Installing page* for prerequisites for DRI hardware acceleration.

4.247.1 MD5 checksums

```
0a70c15c135561824bdcae92bf232e43  MesaLib-7.10.tar.gz
33fb94eccc02cbb4d8d1365615e38e46  MesaLib-7.10.tar.bz2
5cafdc0eda0f9bf370b95c98df3338fa  MesaLib-7.10.zip
bc644be551ed585fc4f66c16b64a91c9  MesaGLUT-7.10.tar.gz
5c2677a155672352d62b177e4f0f92e8  MesaGLUT-7.10.tar.bz2
2ce5001f74496d1ba719ef74d910a5cf  MesaGLUT-7.10.zip
```

4.247.2 New features

- GL_ARB_explicit_attrib_location extension (Intel and software drivers).
- GL_ARB_texture_rg (Intel, software drivers, gallium drivers).
- GL_EXT_separate_shader_objects extension (Intel and software drivers).
- GL_NV_primitive_restart extension (Gallium softpipe, llvmpipe).
- New fragment shader back-end for i965-class hardware.
- Support for Sandybridge chipset in i965 DRI driver.

4.247.3 Bug fixes

This list is likely incomplete.

- Bug 28800 - [r300c, r300g] Texture corruption with World of Warcraft
- Bug 29420 - Amnesia / HPL2 RendererFeatTest - not rendering correctly
- Bug 29946 - [swrast] piglit valgrind glsl-array-bounds-04 fails
- Bug 30261 - [GLSL 1.20] allowing inconsistent invariant declaration between two vertex shaders
- Bug 30632 - [softpipe] state_tracker/st_manager.c:489: st_context_notify_invalid_framebuffer: Assertion 'stfb && stfb->iface == stfbi' failed.
- Bug 30694 - wincopy will crash on Gallium drivers when going to front buffer
- Bug 30771 - [r600g] vert-tex glsl demo
- Bug 30787 - Invalid asm shader does not generate draw-time error when used with GLSL shader
- Bug 30974 - [llvmpipe] SIGABRT src/gallium/drivers/llvmpipe/lp_state_fs.c:779
- Bug 30993 - getFramebufferAttachmentParameteriv wrongly generates error
- Bug 31101 - [glsl2] abort() in ir_validate::visit_enter(ir_assignment *ir)
- Bug 31193 - [regression] aa43176e break water reflections
- Bug 31194 - The mesa meta save/restore code doesn't ref the current GLSL program
- Bug 31371 - glslparsertest: ir.cpp:358: ir_constant::ir_constant(const glsl_type*, const ir_constant_data*): Assertion '(type->base_type >= 0) && (type->base_type <= 3)' failed.
- Bug 31439 - Crash in glBufferSubData() with size == 0
- Bug 31495 - [i965 gles2c bisected] OpenGL ES 2.0 conformance GL2Tests_GetBIFD_input.run regressed
- Bug 31514 - isBuffer returns true for unbound buffers
- Bug 31560 - [tdfx] tdfx_tex.c:702: error: 'const struct gl_color_table' has no member named 'Format'
- Bug 31617 - Radeon/Compiz: 'failed to attach dri2 front buffer', error case not handled

- Bug 31648 - [GLSL] array-struct-array gets assertion: '(size >= 1) && (size <= 4)' failed.
- Bug 31650 - [GLSL] varying gl_TexCoord fails to be re-declared to different size in the second shader
- Bug 31673 - GL_FRAGMENT_PRECISION_HIGH preprocessor macro undefined in GLSL ES
- Bug 31690 - i915 shader compiler fails to flatten if in Aquarium webgl demo.
- Bug 31832 - [i915] Bad renderbuffer format: 21
- Bug 31841 - [drm:radeon_cs_ioctl] *ERROR* Invalid command stream !
- Bug 31894 - Writing to gl_PointSize with GLES2 corrupts other varyings
- Bug 31909 - [i965] brw_fs.cpp:1461: void fs_visitor::emit_bool_to_cond_code(ir_rvalue*): Assertion 'expr->operands[i]->type->is_scalar()' failed.
- Bug 31934 - [gallium] Mapping empty buffer object causes SIGSEGV
- Bug 31983 - [i915 gles2] "if (expression with builtin/varying variables) discard" breaks linkage
- Bug 31985 - [GLSL 1.20] initialized uniform array considered as "unsized"
- Bug 31987 - [gles2] if input a wrong pname(GL_NONE) to glGetBoolean, it will not case GL_INVALID_ENUM
- Bug 32035 - [GLSL bisected] comparing unsized array gets segfault
- Bug 32070 - llvmpipe renders stencil demo incorrectly
- Bug 32273 - assertion fails when starting vdrift 2010 release with shaders enabled
- Bug 32287 - [bisected GLSL] float-int failure
- Bug 32311 - [965 bisected] Array look-ups broken on GM45
- Bug 32520 - [gles2] glBlendFunc(GL_ZERO, GL_DST_COLOR) will result in GL_INVALID_ENUM
- Bug 32825 - egl_glx driver completely broken in 7.9 branch [fix in master]

4.247.4 Changes

Adam Jackson (2):

- i965: Update renderer strings for sandybridge
- drivers/x11: unifdef XFree86Server

Alex Deucher (30):

- r600c: fix mipmap stride on evergreen
- r600c: add reloc for CB_COLOR0_ATTRIB
- r600c: pull over 6xx/7xx vertex fixes for evergreen
- r600c: fix segfault in evergreen stencil code
- r100: revalidate after radeon_update_renderbuffers
- r600c: add missing radeon_prepare_render() call on evergreen
- r600c: properly align mipmaps to group size
- egl_dri2: Add radeon chip ids
- r600c/evergreen: texture align is group_bytes just like 6xx/7xx
- r600g: fix buffer alignment

- r600g: All EVENT_WRITE packets need the EVENT_INDEX field
- r600g: translate ARR instruction for evergreen
- r600g: use meaningful defines for chiprev
- r600g: use full range of VS resources for vertex samplers
- r600g: fix additional EVENT_WRITE packet
- r600g: fix some winsys functions to deal properly with evergreen
- r600c: add Ontario Fusion APU support
- r600g: add support for ontario APUs
- r600c: fix VC flush on cedar and palm
- gallium/egl: fix r300 vs r600 loading
- r600c: fix some opcodes on evergreen
- r600c: bump texture limits to hw limits
- r600g: bump texture/cb limits appropriately for evergreen
- radeon: bump mip tree levels to 15
- r600g: fix rendering with a vertex attrib having a zero stride
- r600g: remove useless switch statements
- r600g: add support for NI (northern islands) asics
- r600c: add support for NI asics
- r600g: support up to 64 shader constants
- r600c: fix up SQ setup in blit code for Ontario/NI

Andre Maasikas (3):

- r600c: fix buffer height setting in dri2 case
- r600g: break alu clause earlier
- r600g: fix evergreen interpolation setup

Andrew Randrianasulu (2):

- dri/nv04: Don't expose ARB_texture_env_combine/dot3.
- dri/nv04: Enable eng3dm for A8/L8 textures.

Aras Pranckevicius (2):

- glsl: fix crash in loop analysis when some controls can't be determined
- glsl: fix matrix type check in ir_algebraic

Bas Nieuwenhuizen (3):

- r600g: set ENABLE_KILL in the shader state in the new design
- r600g: set ENABLE_KILL on evergreen too
- r600g: use dirty list to track dirty blocks

Ben Skeggs (3):

- nv50: DST

- nv50: DPH
- nv50: silence some unknown get_param warnings

Benjamin Franzke (2):

- st/egl image: multiply drm buf-stride with blocksize
- r600g: implement texture_get_handle (needed for eglExportDRMImageMESA)

Brian Paul (296):

- glx: add const qualifiers to __indirect_glMultiDrawArraysEXT()
- glsl2: fix signed/unsigned comparison warning
- llvmpipe: cast to silence warning
- llvmpipe: s/boolean/unsigned/ in bitfield to silence warning
- nv50: use unsigned int for bitfields to silence warnings
- tgsi: fix incorrect usage_mask for shadow tex instructions
- gallivm: expand AoS sampling to cover all filtering modes
- gallivm: fix incorrect vector shuffle datatype
- gallivm: move i32_vec_type inside the #ifdef
- mesa: include mfeatures.h in formats.c
- gallivm: fix wrong return value in bitwise functions
- tgsi/sse: fix aos_to_soa() loop to handle num_inputs==0
- gallivm: added missing case for PIPE_TEXTURE_RECT
- gallium: better docs for pipe_rasterizer_state::sprite_coord_enable
- gallium: rework handling of sprite_coord_enable state
- gallium/docs: added new pipeline.txt diagram
- mesa: don't call valid_texture_object() in non-debug builds
- glsl2: silence compiler warnings in printf() calls
- docs: remove old broken link
- docs: mark as obsolete, remove dead links
- llvmpipe: fix query bug when no there's no scene
- gallivm: remove debug code
- llvmpipe: maintain fragment shader state for draw module
- llvmpipe: indentation fix
- llvmpipe: reformatting, remove trailing whitespace, etc
- llvmpipe: clean-up, comments in setup_point_coefficient()
- llvmpipe: rename sprite field, add sprite_coord_origin
- llvmpipe: implement sprite coord origin modes
- draw: fix test for using the wide-point stage
- llvmpipe: check bitshift against PIPE_MAX_SHADER_OUTPUTS

- draw: check bitshift against PIPE_MAX_SHADER_OUTPUS
- Merge branch 'sprite-coord'
- draw: new draw_fs.[ch] files
- glsl2: fix typo in error msg
- gallivm: fix lp_build_sample_compare()
- softpipe: add missing calls to set draw vertex samplers/views
- mesa: don't advertise bogus GL_ARB_shading_language_120 extension
- configs: remove egl-swrast target from linux-dri config
- llvmpipe: fix sprite texcoord setup for non-projective texturing
- mesa: fix assertions to handle srgb formats
- st/mesa: add missing MESA_FORMAT_S8 case in st_mesa_format_to_pipe_format()
- st/mesa: use the wrapped renderbuffer in CopyPixels()
- llvmpipe: make min/max lod and lod bias dynamic state
- llvmpipe: make texture border_color dynamic state
- softpipe: fix repeat() function for NPOT textures
- gallivm: fix repeat() function for NPOT textures
- swrast: update comments for REMAINDER() macro
- softpipe: rename sp_state_fs.c -> sp_state_shader.c
- softpipe: make shader-related functions static
- softpipe: make blend/stencil/depth functions static
- softpipe: make sampler state functions static
- softpipe: make vertex state functions static
- softpipe: make rasterizer state functions static
- softpipe: make stream out state functions static
- softpipe: make clip state functions static
- softpipe: minor asst. clean-ups
- softpipe: allocate tile data on demand
- llvmpipe: fix swizzling of texture border color
- softpipe: fix swizzling of texture border color
- draw: pass sampler state down to llvm jit state
- gallivm: check for level=0 case in lp_build_minify()
- gallivm: added some comments
- draw: check for null sampler pointers
- swrast: fix choose_depth_texture_level() to respect mipmap filtering state
- st/mesa: replace assertion w/ conditional in framebuffer invalidation
- glsl2: fix signed/unsigned comparison warning

- st/xlib: add some comments
- ir_to_mesa: assorted clean-ups, const qualifiers, new comments
- mesa: remove assertion w/ undeclared variable texelBytes
- gallivm: remove newlines
- draw/llvmpipe: replace DRAW_MAX_TEXTURE_LEVELS with PIPE_MAX_TEXTURE_LEVELS
- mesa: reformatting, comments, code movement
- x11: fix breakage from gl_config::visualType removal
- gallivm: work-around trilinear mipmap filtering regression with LLVM 2.8
- mesa: remove post-convolution width/height vars
- gallivm: add compile-time option to emit inst addrs and/or line numbers
- llvmpipe: code to dump bytecode to file (disabled)
- gallivm: added lp_build_print_vec4()
- gallivm: added lp_build_load_volatile()
- glsl: add ir_unop_round_even case to silence unhandled enum warning
- st/mesa: fix regressions in glDrawPixels(GL_STENCIL_INDEX)
- st/mesa: reformatting in st_cb_drawpixels.c
- st/mesa: use GLuint to avoid problem w/ uint not defined on mingw32
- st/mesa: update function name, comments
- gallivm: use util_snprintf()
- llvmpipe: remove lp_setup_coef*.c files from Makefile
- mesa: fix mesa version string construction
- gallivm: fix incorrect type for zero vector in emit_kilp()
- llvmpipe/draw: always enable LLVMAddInstructionCombiningPass()
- draw: use float version of LLVM Mul/Add instructions
- draw: fix typo in comment
- mesa: add GL_RG case to _mesa_source_buffer_exists()
- mesa: add missing cases for packing red/green images
- st/mesa: added cases for GL_COMPRESSED_RED/RG in st_choose_format()
- docs: update texture red/green support in GL3.txt
- docs: add GL_ARB_texture_rg to release notes
- mesa: driver hook for primitive restart
- mesa: set/get primitive restart state
- mesa: API spec for primitive restart
- mesa: regenerated files with primitive restart
- mesa: plug in primitive restart function
- vbo: support for primitive restart

- gallium: new CAP, state for primitive restart
- st/mesa: support for primitive restart
- draw: implement primitive splitting for primitive restart
- softpipe: enable primitive restart
- llvmpipe: enable primitive restart
- docs: added GL_NV_primitive_restart extension
- Merge branch 'primitive-restart-cleanup'
- winsys/xlib: formatting fixes
- winsys/xlib: use Bool type for shm field
- winsys/xlib: fix up allocation/dealloction of XImage
- winsys/xlib: rename xm->xlib
- galahad: silence warnings
- mesa: move declaration before code
- docs: updated GL3 status for primitive restart
- mesa: 80-column wrapping
- mesa: simplify fbo format checking code
- mesa: split up the image.c file
- mesa: add pixel packing for unscaled integer types
- mesa: _mesa_ClearColorIuiEXT() and _mesa_ClearColorIiEXT()
- mesa: _mesa_is_format_integer() function
- mesa: minor reformatting, clean-ups
- mesa: added _mesa_is_fragment_shader_active() helper
- mesa: new glDrawPixels error check for integer formats
- softpipe: added some texture sample debug code (disabled)
- mesa: added new gl_extensions::EXT_gpu_shader4 field
- mesa: added new gl_framebuffer::_IntegerColor field
- mesa: added glGet query for GL_RGBA_INTEGER_MODE_EXT
- mesa: compute _IntegerColor field in _mesa_test_framebuffer_completeness()
- mesa: added cases for GL_EXT_texture_integer formats
- mesa: added cases for GL_EXT_texture_integer
- st/mesa: add format selection for signed/unsigned integer formats
- mesa: simplify target_can_be_compressed() function
- glapi: GL_EXT_texture_integer API
- glapi: include/build EXT_texture_integer.xml
- mesa: regenerated API files for GL_EXT_texture_integer
- mesa: plug in GL_EXT_texture_integer functions

- mesa: display list support for GL_EXT_texture_integer
- st/mesa: be smarter choosing texture format for glDrawPixels()
- softpipe: remove >32bpp color restriction
- mesa: silence enum comparison warning
- mesa: fix uninitialized var warning
- xlib: silence unused var warning
- util: use pointer_to_func() to silence warning
- rtasm: use pointer_to_func() to silence warning
- translate: use function typedefs, casts to silence warnings
- translate: remove unused prototypes
- mesa: additional glReadPixels error checks for GL_EXT_texture_integer
- mesa: additional switch cases for GL_EXT_texture_integer
- mesa: additional teximage error checks for GL_EXT_texture_integer
- mesa: do integer FB / shader validation check in _mesa_valid_to_render()
- mesa: call _mesa_valid_to_render() in glDrawPixels, glCopyPixels, glBitmap
- mesa: remove the unused _mesa_is_fragment_shader_active() function
- mesa: fix bug in _mesa_is_format_integer()
- mesa: rename function to _mesa_is_format_integer_color()
- mesa: remove 'normalized' parameter from _mesa_VertexAttribIPointer()
- vbo: re-indent file
- glapi: xml spec file for GL_EXT_gpu_shader4
- glapi: include EXT_gpu_shader4.xml
- glapi: regenerated API files
- mesa: plug in stubs for glBindFragDataLocation(), glGetFragDataLocation()
- mesa: add glGetUniformuiv(), plug in uint glUniform funcs
- mesa: plug in more GL_EXT_gpu_shader4 functions
- mesa: add new GLvertexformat entries for integer-valued attributes
- mesa: implement integer-valued vertex attribute functions
- mesa: add gl_client_array::Integer field and related vertex array state code
- mesa: consolidate glVertex/Color/etcPointer() code
- mesa: state/queries for GL_MIN/MAX_PROGRAM_TEXEL_OFFSET_EXT
- mesa: glArrayElement support for integer-valued arrays
- mesa: clean-up array element code
- mesa: add extension table entry for GL_EXT_gpu_shader4
- mesa: remove obsolete comment
- mesa: fix incorrect type in _mesa_texstore_rgba_int16()

- mesa: fix integer cases in _mesa_is_legal_format_and_type()
- mesa: add const qualifier to _mesa_is_legal_format_and_type()
- mesa: additional integer formats in _mesa_bytes_per_pixel()
- mesa: pixel transfer ops do not apply to integer-valued textures
- mesa: remove dead code
- osmesa: fix renderbuffer memleak in OSMesaMakeCurrent()
- mesa: use GLubyte for edge flag arrays
- mesa: move the gl_config struct declaration
- dri/util: add a bunch of comments
- mesa: remove always-false conditional in check_compatible()
- mesa: fix aux/accum comment and error message mixups
- llvmpipe: assign context's frag shader pointer before using it
- llvmpipe: add a cast
- mesa: silence new warnings in texobj.c
- egl/gdi: fix typo: xsurf->gsurf
- mesa: code to unpack RGBA as uints
- gallivm: implement scatter stores into temp register file
- gallivm: add some LLVM var labels
- gallivm: added debug code to dump temp registers
- gallivm: add pixel offsets in scatter stores
- gallivm: added lp_elem_type()
- gallivm: implement execution mask for scatter stores
- tgsi: remove unused function
- llvmpipe: added some debug assertions, but disabled
- gallivm: alloca() was called too often for temporary arrays
- gallivm: add const qualifiers, fix comment string
- softpipe: disable vertex texturing with draw/llvm
- mesa: consolidate pixel packing/unpacking code
- mesa: rename vars in pixel pack/unpack code
- mesa: implement uint texstore code
- mesa: remove stray GL_FLOAT case in _mesa_is_legal_format_and_type()
- mesa: make fixed-pt and byte-valued arrays a runtime feature
- softpipe: can't no-op depth test stage when occlusion query is enabled
- mesa: no-op glBufferSubData() on size==0
- mesa: #include mfeatures.h in enums.h
- mesa: improve error message

- mesa: add missing formats in _mesa_format_to_type_and_comps()
- mesa: handle more pixel types in mipmap generation code
- mesa: make glIsBuffer() return false for never bound buffers
- mesa: fix glDeleteBuffers() regression
- tdfx: s/Format/_BaseFormat/
- mesa: consolidate assertions in teximage code
- radeon: set gl_texture_image::TexFormat field in radeonSetTexBuffer2()
- r600: set gl_texture_image::TexFormat field in r600SetTexBuffer2()
- r200: set gl_texture_image::TexFormat field in r200SetTexBuffer2()
- r300: set gl_texture_image::TexFormat field in r300SetTexBuffer2()
- evergreen: set gl_texture_image::TexFormat field in evergreenSetTexBuffer()
- st/mesa: fix glDrawPixels(depth/stencil) bugs
- glsl: fix assorted MSVC warnings
- mesa: add more work-arounds for acoshf(), asinhf(), atahf()
- glsl: remove opt_constant_expression.cpp from SConscript
- mesa: fix error messages and minor reindenting
- mesa: whitespace cleanups
- mesa: 80-column wrapping
- mesa: reorder texture_error_check() params
- mesa: minor clean-ups in context code
- mesa: upgrade to glext.h version 66
- mesa: pass gl_format to _mesa_init_teximage_fields()
- mesa: fix error msg typo
- glapi: rename GL3.xml to GL3x.xml as it covers all GL 3.x versions
- mesa: hook up GL 3.x entrypoints
- docs: update some GL 3.0 status
- mesa: fix get_texture_dimensions() for texture array targets
- swrast: init alpha value to 1.0 in opt_sample_rgb_2d()
- glsl: fix off by one in register index assertion
- glsl: use gl_register_file in a few places
- mesa: rename, make _mesa_register_file_name() non-static
- mesa: _mesa_valid_register_index() to validate register indexes
- mesa: replace #defines with new gl_shader_type enum
- mesa: use gl_shader_type enum
- glsl: better handling of linker failures
- glsl: start restoring some geometry shader code

- mesa: add assertion and update comment in _mesa_format_image_size()
- mesa: added _mesa_format_image_size64()
- x11: remove test_proxy_teximage() function
- st/mesa: fix mapping of zero-sized buffer objects
- gallivm/llvmpipe: squash merge of the llvm-context branch
- mesa: raise max texture sizes to 16K
- softpipe: increase max texture size to 16K
- mesa: replace large/MAX_WIDTH stack allocations with heap allocations
- mesa: replace large/MAX_WIDTH stack allocations with heap allocations
- swrast: avoid large stack allocations in blend code
- swrast: avoid large stack allocations in tex combine code
- st/mesa: avoid large stack allocations in readpixels code
- mesa: replace more MAX_WIDTH stack allocations with heap allocations
- gallivm/llvmpipe: remove lp_build_context::builder
- gallivm: fix null builder pointers
- mesa: fix GL_FRAMEBUFFER_ATTACHMENT_OBJECT_NAME query
- mesa: return GL_FRAMEBUFFER_DEFAULT as FBO attachment type
- llvmpipe: fix broken stencil writemask
- mesa: consolidate some compiler -D flags
- swrast: allow GL_RG format in glDrawPixels()
- swrast: fix indentation
- swrast: accept GL_RG in glReadPixels()
- swrast: restructure some glReadPixels() code
- mesa: make glGet*(GL_NONE) generate GL_INVALID_ENUM
- mesa: remove unneeded cast
- mesa: update comments, remove dead code
- st/mesa: new comment about updating state vars
- mesa: add error margin to clip mask debug/check code
- gallium/util: minor formatting fixes
- mesa/llvm: use llvm-config –cppflags
- st/mesa: fix mipmap generation bug
- mesa: test for cube map completeness in glGenerateMipmap()
- mesa: set gl_texture_object::_Complete=FALSE in incomplete()
- mesa: consolidate glTexImage1/2/3D() code
- mesa: simplify proxy texture code in texture_error_check()
- mesa: consolidate the glTexSubImage1/2/3D() functions

- mesa: consolidate glCopyTexImage1/2D() code
- mesa: consolidate glCopyTexSubImage1/2/3D() functions
- mesa: consolidate glCompressedTexImage1/2/3D() functions
- mesa: make _mesa_test_proxy_teximage() easier to read
- configure: use llvm-config –cppflags instead of –cflags
- mesa: revamp error checking for compressed texture images
- mesa: simplify target checking for TexImage functions
- draw/llvm: don't flush in vs_llvm_delete()
- tnl: Initialize gl_program_machine memory in run_vp.
- tnl: a better way to initialize the gl_program_machine memory
- mesa, st/mesa: disable GL_ARB_geometry_shader4
- mesa/meta: fix broken assertion, rename stack depth var
- glsl: new glsl_strtod() wrapper to fix decimal point interpretation
- st/mesa: fix renderbuffer pointer check in st_Clear()

Brian Rogers (1):

- mesa: Add missing else in do_row_3D

Chad Versace (25):

- intel_extensions: Add ability to set GLSL version via environment
- glsl: Add glsl_type::uvecN_type for N=2,3
- glsl: Add lexer rules for uint and uvecN (N=2..4)
- glsl: Changes in generated file glsl_lexer.cpp
- glsl: Add lexer rules for << and >> in GLSL 1.30
- glsl: Change generated file glsl_lexer.cpp
- glsl: Implement ast-to-hir for binary shifts in GLSL 1.30
- glsl: Implement constant expr evaluation for bitwise-not
- glsl: Implement constant expr evaluation for bit-shift ops
- glsl: Implement constant expr evaluation for bitwise logic ops
- glsl: Fix ir validation for bit logic ops
- glsl: Define shift_result_type() in ast_to_hir.cpp
- glsl: Implement ast-to-hir for bit-shift-assignment
- glsl: Define bit_logic_result_type() in ast_to_hir.cpp
- glsl: Implement ast-to-hir for bit-logic ops
- glsl: Fix lexer rule for ^=
- glsl: Commit generated file glsl_lexer.cpp
- glsl: Fix ast-to-hir for ARB_fragment_coord_conventions
- mesa: Fix C++ includes in sampler.cpp

- glsl: Fix ir_expression::constant_expression_value()
- glsl: Fix erroneous cast in ast_jump_statement::hir()
- glsl: Fix Doxygen tag file in recently renamed files
- glsl: Improve usage message for glsl_compiler
- glsl: Fix linker bug in cross_validate_globals()
- glsl: In ast_to_hir, check sampler array indexing

Chia-I Wu (149):

- glapi: Fix build errors for ES.
- glapi: Fix ES build errors again.
- mesa: Update ES APIspec.xml.
- st/xlib: Notify the context when the front/back buffers are swapped.
- targets/egl: Use C++ compiler to link GL/ES state trackers.
- libgl-xlib: Remove unused st_api_create_OpenGL.
- st/egl: Split modeset code support to modeset.c.
- st/egl: Rename kms backend to drm.
- st/egl: s/kms/drm/ on the drm backend.
- egl: Enable drm platform by default.
- egl: Check extensions.
- st/egl: Skip single-buffered configs in EGL.
- mapi: Fix compiler warnings.
- st/egl: Drop context argument from egl_g3d_get_egl_image.
- targets/egl: Fix linking with libdrm.
- st/vega: Fix version check in context creation.
- egl: Use attribute names as the _EGLConfig member names.
- egl: Access config attributes directly.
- st/egl: Access _EGLConfig directly.
- st/egl: Do not finish a fence that is NULL.
- mesa: Remove unused vtxfmt_tmp.h.
- egl_dri2: Drop the use of _egl[SG]etConfigKey.
- egl_glx: Drop the use of [SG]ET_CONFIG_ATTRIB.
- egl_glx: Fix borken driver.
- egl: Move attributes in _EGLImage to _EGLImageAttribs.
- egl: Parse image attributes with _eglParseImageAttribList.
- egl: Move fallback routines to eglfallbacks.c.
- egl: Drop dpy argument from the link functions.
- egl: Minor changes to the _EGLConfig interface.

- egl: Minor changes to the _EGLScreen interface.
- egl: Fix _eglModeLookup.
- st/egl: Fix native_mode refresh mode.
- egl: Add reference count for resources.
- egl: Use reference counting to replace IsLinked or IsBound.
- egl: Fix a false negative check in _eglCheckMakeCurrent.
- st/egl: Use resource reference count for egl_g3d_sync.
- egl_dri2: Fix a typo that make glFlush be called at wrong time.
- glapi: Do not use glapidispatch.h.
- glapi: Move glapidispatch.h to core mesa.
- glapi: Do not use glapioffsets.h.
- glapi: Merge glapioffsets.h into glapidispath.h.
- vbo: Use CALL_* macros.
- mesa: Remove unnecessary glapitable.h includes.
- autoconf: Better client API selection.
- docs: Update egl and openvg docs.
- autoconf: Update configuration info.
- Merge branch 'glapi-reorg'
- targets: Add missing quotes to Makefile.xorg.
- autoconf: st/vega requires –enable-openvg.
- st/mesa: Unreference the sampler view in st_bind_surface.
- autoconf: Tidy configure output for EGL.
- targets/egl: Fix a warning with –disable-opengl build.
- egl: Rework _eglGetSearchPath.
- mesa: Select FEATURE_remap_table when multiple APIs are enabled.
- mesa: Allow contexts of different APIs to coexist.
- egl: Set up the pthread key even TLS is used.
- st/egl: Add native_surface::present callback.
- st/egl: Use native_surface::present callback.
- d3d1x: Use native_surface::present.
- st/egl: Remove flush_frontbuffer and swap_buffers.
- st/egl: Add support for swap interval and swap behavior.
- st/egl: Add support for EGL_MATCH_NATIVE_PIXMAP.
- st/egl: Add extern "C" wrapper to native.h.
- st/egl: Add native_display_buffer interface.
- st/egl: Use native_display_buffer for EGL_MESA_drm_image.

- autoconf: Add –enable-gallium-egl.
- docs: Update egl docs.
- st/dri: Add support for surfaceless current contexts.
- egl_dri2: Fix __DRI_DRI2 version 1 support.
- st/vega: Do not wait NULL fences.
- gallium: Add st_api::name.
- gallium: Add st_context_iface::share to st_api.
- st/wgl: Use st_context_iface::share for DrvShareLists.
- st/glx: Replace MESA_VERSION_STRING by xmesa_get_name.
- mesa: Clean up core.h.
- scons: Define IN_DRI_DRIVER.
- tgsi: Add STENCIL to text parser.
- st/vega: vegaLookupSingle should validate the state.
- st/vega: Set wrap_r for mask and blend samplers.
- st/vega: Fix vgReadPixels with a subrectangle.
- egl_dri2: Fix one context, multiple surfaces.
- auxiliary: util_blit_pixels_tex should restore the viewport.
- st/vega: Fix a crash with empty paths.
- st/vega: Masks and surfaces should share orientation.
- st/vega: No flipping in vg_prepare_blend_surface.
- st/vega: Fix a typo in EXTENDED_BLENDER_OVER_FUNC.
- llvmpipe: Fix build errors on x86.
- st/vega: Overhaul renderer with renderer states.
- st/vega: Add DRAWTEX renderer state.
- st/vega: Add SCISSOR renderer state.
- st/vega: Add CLEAR renderer state for vgClear.
- st/vega: Add FILTER renderer state for image filtering.
- st/vega: Use the renderer for vgMask.
- st/vega: Add POLYGON_STENCIL and POLYGON_FILL renderer state.
- st/vega: Delay fb state update to vg_validate_state.
- st/vega: Use st_framebuffer for fb width/height.
- st/vega: Move g3d states to renderer.
- st/vega: Make shader_bind call into the renderer.
- st/vega: vg_manager should care about only the color buffer.
- st/vega: Clean up vg_context fields and functions.
- st/vega: Clean up renderer fields and functions.

- st/vega: vg_copy_texture and vg_copy_surface should share code.
- st/vega: Get rid of renderer_copy_texture.
- st/vega: Update to latest headers.
- st/vega: Fix image sampler views for alpha-only formats.
- st/vega: Make path_render and path_stroke take a matrix.
- st/vega: Make image_draw take a matrix.
- st/vega: Add primitive text support.
- st/vega: Revive mask layer support.
- st/vega: More flexible shader selection.
- st/vega: Add color transformation support.
- st/vega: Bump version to 1.1.
- st/vega: Fix paint coordinates transformations.
- st/vega: Fix negated logic in image_draw.
- st/vega: Fix degenerate paints.
- st/vega: Simplify radial gradient.
- st/vega: Remove st_inlines.h.
- st/vega: Delay blend texture creation until needed.
- st/vega: Create drawing surface mask as needed.
- st/vega: Initialize pipe states with renderer.
- st/vega: Avoid unnecessary constant bufer upload.
- st/vega: Destroy the pipe context with vg_context.
- st/vega: polygon_array requires a deep free.
- st/egl: Set pipe_resource::array_size to 1.
- st/vega: Set pipe_resource::array_size to 1.
- st/vega: Move vertex transformation to shader.
- st/vega: Add a missing break.
- st/vega: Add some comments to pipeline shaders.
- st/vega: Refactor blend shaders.
- st/vega: Move masking after blending.
- st/vega: Add support for per-channel alpha.
- st/vega: Blending should use premultiplied alpha.
- st/vega: Fix VG_BLEND_MULTIPLY.
- st/vega: Add blend shaders for all blend modes.
- st/vega: Fix pipe blend state for various blend modes.
- egl: _eglFilterArray should not allocate.
- mapi: Rewrite mapi_abi.py to get rid of preprocessor magic.

- vbo: Drop second ATTR macro.
- vbo: Fix GLES2 glVertexAttrib.
- mesa: Do not advertise GL_OES_texture_3D.
- mesa: Fix GL_FIXED arrays.
- mesa: Fix glTexCoordPointer with type GL_FIXED.
- st/egl: Plug pbuffer leaks.
- st/egl: Fix eglCopyBuffers.
- st/egl: Assorted fixes for dri2_display_get_configs.
- docs/egl: Update ../egl.html.
- st/egl: Fix eglChooseConfig when configs is NULL.
- docs: Add an example for EGL_DRIVERS_PATH.
- autoconf: Fix –with-driver=xlib –enable-openvg.

Chris Wilson (2):

- i915g: Fix closure of full batch buffers
- intel: Check for unsupported texture when finishing using as a render target

Christoph Bumiller (80):

- nv50: import new compiler
- nouveau: update nouveau_class.h
- nv50: introduce the big formats table
- nv50: don't produce MOV immediate to output reg in store opt
- nv50: change back accidentally swapped UNORM,SNORM vertex type
- nv50: add/fix some license headers
- nv50: simple reload elimination and local CSE
- nv50: fix constant_operand opt mul by 2 case
- nv50: permit usage of undefined TGSI TEMPs
- nv50: add missing 2nd source for POW multiplication
- nv50: add signed RGTC1 to format table, allow 2_10_10_10 for vbufs
- nv50: fix for empty BBs
- nv50: insert MOVs also for PHI sources from dominating block
- nv50: explicitly set src type for SET ops
- nv50: fixes for nested IFs
- nv50: don't eliminate loads to dedicated values
- nv50: fix constbuf validation
- nv50: build proper phi functions in the first place
- nv50: fix reg count
- nv50: begin implementing loops

- nv50: more constant folding
- nv50: loops part 2
- nv50: flatten simple IF/ELSE/ENDIF constructs
- nv50: fix thinko in store to output reg possible check
- nv50: generate JOINs for outermost IF clauses
- nv50: more TGSI opcodes (SIN, SCS, ARL, RET, KILP)
- nv50: fix PSIZ and PRIMID mapping
- nv50: check dst compatibility in CSE
- nv50: initialize edgeflag input index
- nv50: emit predicate for interp
- Merge remote branch 'origin/master' into nv50-compiler
- nv50: DP2, fix ARL
- nv50: yet another case we need a nop.exit
- nv50: fix check for sprite/point coord enable
- nv50: handle TEXTURE_SWIZZLE and GEOMETRY_SHADER4 caps
- nv50: set the FragDepth output index
- nv50: turn off verbose debug output by default
- nv50: attempt at making more complicated loops work
- nv50: SSG
- nv50: make FrontFacing -1 or +1
- nv50: re-add proper TEXBIAS sequence
- nv50: make use of TGSI immediate type
- nv50: must join SELECT inputs before MOV inputs
- nv50: fix XPD, was negated
- nv50: fix find_dom_frontier
- nv50: fix build-predicate function
- Merge remote branch 'origin/master' into nv50-compiler
- nv50: load address register before using it, not after
- nv50: save tgsi instructions
- nv50: prepare for having multiple functions
- nv50: don't parse again in tgsi_2_nc
- nv50: use actual loads/stores if TEMPs are accessed indirectly
- nv50: create value references with the right type
- nv50: duplicate interps in load_proj_tex_coords
- nv50: address regs are 16 bit
- nv50: fix can_load check for 3rd source

- nv50: reduce bb_reachable_by runtime from pot to linear
- nv50: minor compiler fixes and cleanups
- nv50: cannot move from local mem to output reg directly
- nv50: newlines in shader bincode printing
- nv50: match TEMP limit with nv50 ir builder
- nv50: handle TGSI EXP and LOG again
- nv50: check for immediates when turning MUL ADD into MAD
- nv50: interp cannot write flags reg
- nv50: MOV TEMP[0], -CONST[0] must be float32 negation
- nv50: fix indirect CONST access with large or negative offsets
- nv50: fix TXP depth comparison value
- nv50: consider address register in reload elimination
- nv50: improve and fix modifier folding optimization
- nv50: put low limit on REG_ALLOC_TEMP and FP_RESULT_COUNT
- Merge remote branch 'origin/nv50-compiler'
- nv50: don't segfault on shaders with 0 instructions
- nv50: get shader fixups/relocations into working state
- nv50: add relocs for stack and local mem buffers
- nv50: emit constbuf relocs before uploading constants
- nv50: fix typo in fifo packet length limit
- nv50: use formats table in nv50_surface.c
- nv50: use CLEAR_BUFFERS for surface fills
- nv50: fix/handle a few more PIPE_CAPs
- nv50: fix GP state bind and validate

Corbin Simpson (8):

- r600g: Use align() instead of handrolled code.
- r600g: Trivially deobfuscate r600_hw_states.
- r600g: Deobfuscate and comment a few more functions in r600_hw_states.
- r600g: Clean up some indentation and |= vs. | usage.
- r600g: Fix false and true.
- r600g: "tmp" is such a bad name for a texture.
- r600g: Clean up PS setup.
- r600g: Cleanup viewport floats.

Daniel Lichtenberger (1):

- radeon: fix potential segfault in renderbuffer update

Daniel Vetter (21):

- r200: revalidate after radeon_update_renderbuffers
- i915g: rip out ->sw_tiled
- i915g: s/hw_tiled/tiling
- i915g: add pineview pci ids
- i915g: kill RGBA/X formats
- i915g: kill buf->map_gtt
- i915g: kill idws->pool
- i915g: drop alignment parameter from iws->buffer_create
- i915g: add winsys function to create tiled buffers
- i915g: switch to tiled allocations, kill set_fence
- i915g: prepare winsys/batchbuffer for execbuf2
- i915g: return tiling in iws->buffer_from_handle
- i915g: implement unfenced color&depth buffer using tiling bits
- i915g: implement unfenced relocs for textures using tiling bits
- i915g: postpone mipmap/face offset calculation
- i915g: don't pot-align stride for tiled buffers
- i915g: enable X-tiling for textures
- i915g: switch rendering to mipmapped textures to (x,y) offsets
- i915g: enable x-tiling for render targets
- i915g: assert(depth_surface->offset == 0)
- i915g: track TODO items

Dave Airlie (182):

- r300g: fix buffer reuse issue caused by previous commit
- r600g: pull r600_draw struct out into header
- r600g: use index min/max + index buffer offset.
- r600g: add vgt dma src defines
- r600g: fixup texture state on evergreen.
- r600g: fix texture bos and avoid doing depth blit on evergreen
- r600g: hide radeon_ctx inside winsys.
- r600g: attempt to abstract kernel bos from pipe driver.
- r600g: move constant buffer creation behind winsys abstraction.
- r600g: use malloc bufmgr for constant buffers
- r600g: add support for kernel bo
- r600g: add winsys bo caching.
- r600g: add upload manager support.
- r600g: fixup map flushing.

- r600g: use calloc for ctx bo allocations
- r600g: oops got the use_mem_constant the wrong way around.
- r600g; add uses waterfall to asm cf for r6xx.
- r600g: only emit uses waterfall on r6xx hw.
- util/r300g: split the r300 index buffer modifier functions out to util
- r600g: modify index buffers for sizes the hw can't deal with.
- r600g: send correct surface base update for multi-cbufs
- r600g: fix fbo-drawbuffers-maxtargets
- r600g: clean up valgrind issues on maxtargets test.
- r600g: drop debugging that snuck in
- r600g: fix tiling support for ddx supplied buffers
- r600g: add z16 to color setup
- r600g: add color/texture support for more depth formats.
- r600g: fix r700 cube map sizing.
- r600g: fixup r700 CB_SHADER_CONTROL register.
- r600g: add missing BC_INST wrapper for evergreen
- r600g: only flush for the correct colorbuffer, not all of them.
- r600g: deal with overflow of VTX/TEX CF clauses.
- r600g: set back to correct codepaths.
- r600g: fixup evergreen miptree setup.
- r600g: fix eg texture borders.
- r600g: fix typo in struct member name
- r600g: cleanup some of the DB blit code
- r600g: make stencil readback work
- r600g: disable dirty handling on texture from depth code.
- r600g: use floats instead of hex for blit vbo
- r600g: fix depth readback on rv610 and other quirky variants.
- r600g: fix typo in evergreen register list
- u_blitter: add a custom blitter call passing a dsa cso
- r600g: use blitter to do db->cb flushing.
- r600g: fix warnings since last commit.
- egl: fix build since 17eace581d25a626a7d75d9d1205d012cbb14a6e
- r300g: fix point sprite coord.
- r600g: add vert support for 16/16 and 16/16/16 floats.
- r600g: add some more vertex format support.
- r600g: some more vertex formats

- r600g: fix draw-elements and draw-elements-base-vertex
- r600g: drop index_offset parameter to index buffer translate.
- r600g: fixup tex wrapping.
- r600g: fixup VP->FP output->input routing.
- r600g: fix typo in r700 alu emit
- r600g: fixup sprite coord enable.
- r600g: fix polygon mode
- mesa/mipmap: fix warning since 1acadebd6270d3604b026842b8a21360968618a0
- r600g: add eg poly mode code.
- r600g: make index bias fix for evergreen
- r600g: add eg db count control register.
- r300g: fix glsl-fs-pointcoord
- r600g: add evergreen texture resource properly.
- r600g: fix db flush breaking config state
- r600g: on evergreen the centroid isn't set in this register.
- r600g: add back evergreen name.
- r600g: add evergreen texture border support to new path
- r600g: move radeon.h members around to add back map flushing.
- r600g: add initial vertex translate support.
- r600g: remove old assert from new codepath
- Revert "r600g: add initial vertex translate support."
- r600g: port r300g fix for X* formats in texformat code
- r600g: add L8A8 unorm.
- r600g: clean up some code from move to new paths.
- r600g: return string for chip family
- r600g: use Elements macro instead of manual sizeofs
- r600g: fix evergreen depth flushing.
- r600g: add winsys support for CTL constants.
- r600g: drop depth quirk on evergreen
- r600g: add reloc for evergreen color attrib
- r600g: realign evergreen code with r600 code.
- r600g: add assembler support for other vtx fetch fields.
- r600g: fixup vertex format picking.
- r600g: sync vertex/texture cache on resources on evergreen
- r600g: add cb flushing for extra buffers + depth buffer on r600/evergreen
- r600g: fix evergreen draw-buffers

- r600g: flush SH cache on constant change on evergreen
- r600g: only set the Z export if shader exports it.
- r600g: setup basic loop consts on r600 + evergreen.
- mesa/st: initial attempt at RG support for gallium drivers
- r600g: break out of search for reloc bo after finding it.
- r600g: the code to check whether a new vertex shader is needed was wrong
- r600g: fix wwarning in bo_map function
- r600g: TODO domain management
- r600g: add bo fenced list.
- pb: don't keep checking buffers after first busy
- r600g: add bo busy backoff.
- r600g: drop mman allocator
- r600g: drop use_mem_constant.
- r600g: avoid unneeded bo wait
- pb: fix numDelayed accounting
- r600g: add evergreen stencil support.
- r600g: use format from the sampler view not from the texture.
- r600g: fix Z export enable bits.
- r600g: add some RG texture format support.
- r600g: drop width/height per level storage.
- r600g: fix input/output Z export mixup for evergreen.
- r600g: evergreen has no request size bit in texture word4
- r600g: enable vertex samplers.
- r600g: add TXL opcode support.
- r600g: don't run with scissors.
- r600g: fix typo in vertex sampling on r600
- gallium/tgsi: add support for stencil writes.
- gallium/format: add support for X24S8 and S8X24 formats.
- gallium/format: add X32_S8X24_USCALED format.
- gallium/util: add S8 tile sampling support.
- mesa: add support for FRAG_RESULT_STENCIL.
- mesa: improve texstore for 8/24 formats and add texstore for S8.
- softpipe: add support for shader stencil export capability
- st/mesa: add option to choose a texture format that we won't render to.
- st/mesa: use shader stencil export to accelerate shader drawpixels.
- r600g: add support for S8, X24S8 and S8X24 sampler formats.

- r600g: add shader stencil export support.
- glsl: add support for shader stencil export
- st/mesa: enable stencil shader export extension if supported
- r600g: fix depth0 setting
- r600g: fix scissor/cliprect confusion
- r600g: store samplers/views across blit when we need to modify them
- r600g: reduce size of context structure.
- r600g: the vs/ps const arrays weren't actually being used.
- r600g: add copy into tiled texture
- r600g: split out miptree setup like r300g
- r600g: use common texture object create function
- r600g: rename pitch in texture to pitch_in_bytes
- r600g: remove bpt and start using pitch_in_bytes/pixels.
- r600g: fix transfer stride.
- r600g: drop all use of unsigned long
- r600g: use blitter for hw copy region
- r600g: evergreen add stencil export bit
- r600g: add missing eg reg definition
- r600g: fix stencil export for evergreen harder
- r600g: drop unused context members
- r600g: only pick centroid coordinate when asked.
- r600g: fixup pos/face ena/address properly
- r600g: fixup typo in macro name
- r600g: select linear interpolate if tgsi input requests it
- r300g: clean up warning due to unknown cap.
- tgsi: add scanner support for centroid inputs
- r600g: evergreen interpolation support.
- r600g: add evergreen ARL support.
- r600g: switch to a common formats.h file since they are in different regs
- r600g: add defines for tiling
- r600g: get tiling info from kernel
- r600g: set tiling bits in hw state
- r600g: do proper tracking of views/samplers.
- r600g: fix typo in tiling setup cb code.
- r600g: depth needs to bound to ds
- r600g: attempt to cleanup depth blit

- r600g: fix transfer function for tiling.
- r600g: retrieve tiling info from kernel for shared buffers.
- r600g: all non-0 mipmap levels need to be w/h aligned to POT.
- r600g: move to per-miplevel array mode.
- r600g: start adding hooks for aligning width/height for tiles.
- r600g: add r600 surface to store the aligned height.
- r600g: introduce a per-driver resource flag for transfers.
- r600g: add texture tiling alignment support.
- r600g: add texture tiling enable under a debug option.
- r600g: initial translate state support
- r600g: start splitting out common code from eg/r600.
- r600g: not fatal if we can't get tiling info from kernel
- r600g: merge more of the common r600/evergreen state handling
- r600g: drop more common state handling code
- r600g: fix magic 0x1 ->flat shade ena
- r600g: add assembler support for all the kcache fields.
- gallium/noop: report GL 2.1
- r600g: pick correct color swap for A8 fbos.
- r300g/r600g: bump cache manager timeouts to 1s
- r600g: it looks like r600 can handle dword offsets in the indices.
- r300g: try and use all of vertex constant space
- r300g: fixup rs690 tiling stride alignment calculations.
- r600g: fix evergreen segfaults.
- r600g: hack around property unknown issues.

Eric Anholt (300):

- glsl: Add definition of gl_TextureMatrix inverse/transpose builtins.
- i965: Share the KIL_NV implementation between glsl and non-glsl.
- i965: Also enable CC statistics when doing OQs.
- i965: Track the windowizer's dispatch for kill pixel, promoted, and OQ
- glsl: Rework assignments with write_masks to have LHS chan count match RHS.
- glsl: Fix copy'n'wasted ir_noop_swizzle conditions.
- ir_to_mesa: Only compare vector_elements present for any_nequal/all_equal
- i965: Fix the vector/expression splitting for the write_mask change.
- i965: When splitting vector variable assignment, ignore unset channels.
- i965: Update expression splitting for the vector-result change to compares.
- i965: Warning fix for vector result any_nequal/all_equal change.

- mesa: Remove the non-required ARB_imaging extension.
- mesa: Remove EXT_histogram.
- mesa: Remove SGI_color_table.
- mesa: Remove SGI_color_matrix.
- mesa: Remove EXT_convolution.
- intel: Remove disabled stencil drawpixels acceleration.
- intel: Remove unnecessary minimum pitch alignment to 32 bytes.
- intel: Replace my intel_texture_bitmap code with _mesa_meta_Bitmap.
- radeon: Remove copied minimum pitch alignment code.
- unichrome: Mostly revert my convolution removal changes.
- intel: Remove dead intelIsTextureResident().
- i915: Remove a dead if (0) block.
- intel: Dead comment removal.
- intel: Corresponding FinishRenderTexture debug to BeginRenderTexture.
- i965: Add support for rendering to SARGB8 FBOs.
- intel: Fix segfault on INTEL_DEBUG=fbo with unsupported framebuffers.
- intel: Add fallback debug to glGenerateMipmap.
- intel: More reverting of the sw fallback for depth texture border color.
- intel: Improve some of the miptree debugging.
- mesa: Fix type typo in glGenerateMipmap handling of GL_UNSIGNED_INT data.
- glsl: Fix broadcast_index of lower_variable_index_to_cond_assign.
- glsl: Add validation that a swizzle only references valid channels.
- i965: Fix up writemasked assignments in the new FS.
- i965: Remove swizzling of assignment to vector-splitting single-channel LHS.
- i965: Handle all_equal/any_nequal in the new FS.
- i965: Fix vector splitting RHS channel selection with sparse writemasks.
- i965: Add support for dFdx()/dFdy() to the FS backend.
- i965: Add support for attribute interpolation on Sandybridge.
- i965: Set up inputs to the fragment shader according to FP InputsRead.
- i965: Add support for POW in gen6 FS.
- i965: Fix negation in the new FS backend.
- i965: Actually track the "if" depth in loop in the new FS backend.
- i965: Apply the same set of lowering passes to new FS as to Mesa IR.
- i965: Fix valgrind complaint about base_ir for new FS debugging.
- i965: Fix up the FS backend for the variable array indexing pass.
- i965: Set the variable type when dereferencing an array.

- i965: Add support for dereferencing structs to the new FS backend.
- i965: Add support for struct, array, and matrix uniforms to FS backend.
- i965: Fix all non-snb regression in the snb attribute interpolation commit.
- i965: Fix up part of my Sandybridge attributes support patch.
- i965: Add support for gl_FrontFacing to the new FS backend.
- i965: Subtract instead of adding when computing y delta in new FS backend.
- mesa: Pull ir_to_mesa's sampler number fetcher out to shared code.
- i965: Set up sampler numbers in the FS backend.
- i965: Add support for non-color render target write data to new FS backend.
- i965: Add support for MRT to the new FS backend.
- i965: Add support for ir_loop counters to the new FS backend.
- i965: Add support for ARB_fragment_coord_conventions to the new FS backend.
- glsl: Also update implicit sizes of varyings at link time.
- i965: Do interpolation for varying matrices and arrays in the FS backend.
- i965: Don't try to emit interpolation for unused varying slots.
- i965: Fix array indexing of arrays of matrices.
- i965: Clean up obsolete FINISHME comment.
- mesa: Move the list of builtin uniform info from ir_to_mesa to shared code.
- i965: Add support for builtin uniforms to the new FS backend.
- i965: Fix use of undefined mem_ctx in vector splitting.
- i956: Make new FS discard do its work in a temp, not the null reg!
- i965: Clean up the virtual GRF handling.
- ra: First cut at a graph-coloring register allocator for mesa.
- i965: First cut at register allocation using graph coloring.
- i965: Add live interval analysis and hook it up to the register allocator.
- i965: Remove my "safety counter" code from loops.
- i965: Fix whole-structure/array assignment in new FS.
- mesa: Don't reference a W component in setting up a vec3 uniform component.
- i965: Fix new FS handling of builtin uniforms with packed scalars in structs.
- glsl: Add a lowering pass for texture projection.
- i965: Use the lowering pass for texture projection.
- i965: Split the gen4 and gen5 sampler handling apart.
- i965: Add gen6 attribute interpolation to new FS backend.
- i965: Fix the gen6 jump size for BREAK/CONT in new FS.
- i965: Also increment attribute location when skipping unused slots.
- i965: Pre-gen6, map VS outputs (not FS inputs) to URB setup in the new FS.

- i965: Add real support for pre-gen5 texture sampling to the new FS.
- i965: Fix up copy'n'pasteo from moving coordinate setup around for gen4.
- i965: Restore the forcing of aligned pairs for delta_xy on chips with PLN.
- i965: When producing a single channel swizzle, don't make a temporary.
- i965: Add a sanity check for register allocation sizes.
- i965: Fix off-by-ones in handling the last members of register classes.
- i965: Don't try to emit code if we failed register allocation.
- i965: Add support for EXT_texture_swizzle to the new FS backend.
- i965: Set up swizzling of shadow compare results for GL_DEPTH_TEXTURE_MODE.
- i965: Fix glean/texSwizzle regression in previous commit.
- i965: Be more conservative on live interval calculation.
- i965: Add trivial dead code elimination in the new FS backend.
- i965: Add initial folding of constants into operand immediate slots.
- i965: In disasm, gen6 fb writes don't put msg reg # in destreg_conditionalmod.
- i965: Add support for gen6 FB writes to the new FS.
- i965: Enable the constant propagation code.
- i965: Also do constant propagation for the second operand of CMP.
- i965: Add back gen6 headerless FB writes to the new FS backend.
- i965: Gen6 no longer has the IFF instruction; always use IF.
- i965: Fix up IF/ELSE/ENDIF for gen6.
- i965: Fix botch in the header_present case in the new FS.
- i965: Add some clarification of the WECtrl field.
- i965: Don't do 1/w multiplication in new FS for gen6
- i965: Gen6's sampler messages are the same as Ironlake.
- i965: Refactor gl_FrontFacing setup out of general variable setup.
- i965: Add support for gl_FrontFacing on gen6.
- i965: Don't assume that WPOS is always provided on gen6 in the new FS.
- i965: Fix gen6 pointsize handling to match pre-gen6.
- i965: Disable emitting if () statements on gen6 until we really fix them.
- i965: Normalize cubemap coordinates like is done in the Mesa IR path.
- mesa: Simplify a bit of _mesa_add_state_reference using memcmp.
- i965: Drop the check for duplicate _mesa_add_state_reference.
- i965: Drop the check for YUV constants in the param list.
- i965: Handle swizzles in the addition of YUV texture constants.
- i965: Fix gen6 WM push constants updates.
- i965: Fix new FS gen6 interpolation for sparsely-populated arrays.

- i965: Enable attribute swizzling (repositioning) in the gen6 SF.
- i965: Add register coalescing to the new FS backend.
- i965: Split FS_OPCODE_DISCARD into two steps.
- i965: Reduce register interference checks for changed FS_OPCODE_DISCARD.
- i965: Move FS backend structures to a header.
- i965: Give the math opcodes information on base mrf/mrf len.
- i965: Give the FB write and texture opcodes the info on base MRF, like math.
- i965: Compute to MRF in the new FS backend.
- i965: Don't consider gen6 math instructions to write to MRFs.
- i965: Add a couple of checks for gen6 math instruction limits.
- i965: Don't compute-to-MRF in gen6 math instructions.
- i965: Expand uniform args to gen6 math to full registers to get hstride == 1.
- i965: Don't compute-to-MRF in gen6 VS math.
- i965: Fix gen6 pixel_[xy] setup to avoid mixing int and float src operands.
- i965: Always use the new FS backend on gen6.
- i965: Fix missing "break;" in i2b/f2b, and missing AND of CMP result.
- intel: Allow CopyTexSubImage to InternalFormat 3/4 textures, like RGB/RGBA.
- i965: Don't rebase the index buffer to min 0 if any arrays are in VBOs.
- i965: Add support for rescaling GL_TEXTURE_RECTANGLE coords to new FS.
- i965: Set class_sizes[] for the aligned reg pair class.
- i965: Update the live interval when coalescing regs.
- i965: Add a pass to the FS to split virtual GRFs to float channels.
- i965: Add a function for handling the move of boolean values to flag regs.
- i965: Add peepholing of conditional mod generation from expressions.
- i965: Enable the new FS backend on pre-gen6 as well.
- i965: Fix texturing on pre-gen5.
- i965: Set the type of the null register to fix gen6 FS comparisons.
- i965: Disable the debug printf I added for FS disasm.
- i965: Fix a weirdness in NOT handling.
- i965: Fix assertion failure on gen6 BufferSubData to busy BO.
- i965: Assert out on gen6 VS constant buffer reads that hang the GPU for now.
- i965: Fix scissor-offscreen on gen6 like we did pre-gen6.
- i965: Avoid blits in BufferCopySubdata on gen6.
- i965: Tell the shader compiler when we expect depth writes for gen6.
- i965: Remove the gen6 emit_mi_flushes I sprinkled around the driver.
- i965: Disable thread dispatch when the FS doesn't do any work.

- i965: Add EU emit support for gen6's new IF instruction with comparison.
- i965: Set the source operand types for gen6 if/else/endif to integer.
- i965: Use the new style of IF statement with embedded comparison on gen6.
- i965: Split register allocation out of the ever-growing brw_fs.cpp.
- i965: Fix gl_FrontFacing emit on pre-gen6.
- i965: Add support for register spilling.
- i965: Don't emit register spill offsets directly into g0.
- i965: Correct scratch space allocation.
- i965: Be more aggressive in tracking live/dead intervals within loops.
- i965: Move the FS disasm/annotation printout to codegen time.
- i965: Add support for pull constants to the new FS backend.
- i965: Add EU code for dword scattered reads (constant buffer array indexing).
- i965: Clarify an XXX comment in FB writes with real info.
- i965: Use SENDC on the first render target write on gen6.
- i965: Clear some undefined fields of g0 when using them for gen6 FB writes.
- i965: Add disasm for the flag register.
- i965: Add support for discard instructions on gen6.
- i965: Handle new ir_unop_round_even in channel expression splitting.
- i965: Fix typo in comment about state flags.
- i965: Set up the constant buffer on gen6 when it's needed.
- i965: Add support for constant buffer loads on gen6.
- i965: Drop the eot argument to read messages, which can never be set.
- i965: Fix VS URB entry sizing.
- i965: Disable register spilling on gen6 until it's fixed.
- i965: Make FS uniforms be the actual type of the uniform at upload time.
- i965: Add user clip planes support to gen6.
- i965: Update gen6 SF state when point state (sprite or attenuation) changes.
- i965: Upload required gen6 VS push constants even when using pull constants.
- i965: Update the gen6 stencil ref state when stencil state changes.
- mesa: Make metaops use program refcounts instead of names.
- mesa: Don't compute an unused texture completeness debug string.
- intel: For batch, use GTT mapping instead of writing to a malloc and copying.
- intel: Annotate debug printout checks with unlikely().
- intel: Remove the magic unaligned memcpy code.
- i965: Remove dead intel_structs.h file.
- intel: Avoid taking logbase2 of several things that we max.

- intel: Remove duplicated teximage miptree to object miptree promotion.
- intel: Remove leftover dri1 locking fields in the context.
- mesa: Fix delayed state flagging for EXT_sso-related program changes.
- intel: Fix the client-side swapbuffers throttling.
- Revert "intel: Fix the client-side swapbuffers throttling."
- i965: Allow OPCODE_SWZ to put immediates in the first arg.
- i965: Add support for math on constants in gen6 brw_wm_glsl.c path.
- i965: Work around strangeness in swizzling/masking of gen6 math.
- i965: re-enable gen6 IF statements in the fragment shader.
- glsl: Free the loop state context when we free the loop state.
- i965: Fix gl_FragCoord inversion when drawing to an FBO.
- i965: Shut up spurious gcc warning about GLSL_TYPE enums.
- mesa: Don't spam the console in a debug build unless some spam is requested.
- i965: Add state dumping for sampler state.
- i965: Add dumping of the sampler default color.
- i965: Fail on loops on gen6 for now until we write the EU emit code for it.
- i965: Eliminate dead code more aggressively.
- mesa: Include C++ files in the makedepend of DRI drivers.
- i965: Fix compute_to_mrf to not move a MRF write up into another live range.
- i965: Just use memset() to clear most members in FS constructors.
- i965: Remove extra n at the end of every instruction in INTEL_DEBUG=wm.
- i965: Fold constants into the second arg of BRW_SEL as well.
- glsl: Add a helper function for determining if an rvalue could be a saturate.
- i965: Recognize saturates and turn them into a saturated mov.
- ir_to_mesa: Detect and emit MOV_SATs for saturate constructs.
- i965: Improve compute-to-mrf.
- i965: Remove duplicate MRF writes in the FS backend.
- i965: Move gen4 blend constant color to the gen4 blending file.
- i965: Don't upload polygon stipple unless required.
- i965: Don't upload line stipple pattern unless we're stippling.
- i965: Don't upload line smooth params unless we're line smoothing.
- i965: Use the new embedded compare in SEL on gen6 for VS MIN and MAX opcodes.
- i965: Fix type of gl_FragData[] dereference for FB write.
- glsl: Make the symbol table's add_function just use the function's name.
- glsl: Make the symbol table's add_variable just use the variable's name.
- glsl: Add a helper constructor for expressions that works out result type.

- glsl: Fix structure and array comparisions.
- glsl: Quiet unreachable no-return-from-function warning.
- i965: Dump the WHILE jump distance on gen6.
- i965: Add support for gen6 DO/WHILE ISA emit.
- i965: Add support for gen6 BREAK ISA emit.
- i965: Add support for gen6 CONTINUE instruction emit.
- i965: Enable IF statements in the VS.
- i965: Add support for loops in the VS.
- glsl: Mark the array access for whole-array comparisons.
- glsl: Fix flipped return of has_value() for array constants.
- mesa: Add getters for the rest of the supported draw buffers.
- mesa: Add getters for ARB_copy_buffer's attachment points.
- intel: Add an env var override to execute for a different GPU revision.
- i965: Update gen6 WM state on compiled program change, not just FP change.
- i965: Update gen6 SF state on fragment program change too.
- i965: Fix compile warning about missing opcodes.
- i965: Move payload reg setup to compile, not lookup time.
- i965: Provide delta_xy reg to gen6 non-GLSL path PINTERP.
- i965: Fix up 16-wide gen6 FB writes after various refactoring.
- i965: Don't smash a group of coordinates doing gen6 16-wide sampler headers.
- i965: Fix gen6 interpolation setup for 16-wide.
- i965: Fix up gen6 samplers for their usage by brw_wm_emit.c
- i965: Make the sampler's implied move on gen6 be a raw move.
- i965: Align gen6 push constant size to dispatch width.
- i965: Add support for the instruction compression bits on gen6.
- i965: Nuke brw_wm_glsl.c.
- i965: Remove INTEL_DEBUG=glsl_force now that there's no brw_wm_glsl.c
- i965: Fix comment about gen6_wm_constants.
- i965: Handle saturates on gen6 math instructions.
- i965: Always hand the absolute value to RSQ.
- i965: Add disabled debug code for dumping out the WM constant payload.
- i965: Work around gen6 ignoring source modifiers on math instructions.
- i965: Fix flipped value of the not-embedded-in-if on gen6.
- i965: Don't try to store gen6 (float) blend constant color in bytes.
- i965: Set up the color masking for the first drawbuffer on gen6.
- i965: Set up the per-render-target blend state on gen6.

- i965: Set the render target index in gen6 fixed-function/ARB_fp path.
- i965: Use the new pixel mask location for gen6 ARB_fp KIL instructions.
- i965: Drop KIL_NV from the ff/ARB_fp path since it was only used for GLSL.
- i965: Drop push-mode reladdr constant loading and always use constant_map.
- i965: Fix VS constants regression pre-gen6.
- i965: Clean up VS constant buffer location setup.
- i965: Set up the correct texture border color state struct for Ironlake.
- i965: Set render_cache_read_write surface state bit on gen6 constant surfs.
- i965: remove unused variable since brw_wm_glsl.c removal.
- intel: Use plain R8 and RG8 for COMPRESSED_RED and COMPRESSED_RG.
- intel: Set the swizzling for depth textures using the GL_RED depth mode.
- glsl: Correct the marking of InputsRead/OutputsWritten on in/out matrices.
- i965: Correct the dp_read message descriptor setup on g4x.
- intel: Include stdbool so we can stop using GLboolean when we want to.
- i965: Fix ARL to work on gen6.
- i956: Fix the old FP path fragment position setup on gen6.
- i965: Fix gl_FragCoord.z setup on gen6.
- i965: Add support for using the BLT ring on gen6.
- intel: Update renderbuffers before looking up CopyTexImage's read buffer.
- intel: Drop commented intel_flush from copy_teximage.
- intel: Try to sanely check that formats match for CopyTexImage.
- intel: Support glCopyTexImage() from XRGB8888 to ARGB8888.
- i965: Avoid using float type for raw moves, to work around SNB issue.
- i965: Set the alternative floating point mode on gen6 VS and WM.
- i965: Add support for gen6 constant-index constant loading.
- i965: Add support for gen6 reladdr VS constant loading.
- i965: Improve the hacks for ARB_fp scalar^scalar POW on gen6.
- i965: Factor out the ir comparision to BRW_CONDITIONAL_* code.
- i965: Fix regression in FS comparisons on original gen4 due to gen6 changes.
- i965: Do lowering of array indexing of a vector in the FS.
- intel: Only do frame throttling at glFlush time when using frontbuffer.
- intel: Handle forced swrast clears before other clear bits.
- intel: Use tri clears when we don't know how to blit clear the format.
- intel: Add spans code for the ARB_texture_rg support.
- intel: Add a couple of helper functions to reduce rb code duplication.
- intel: Fix segfaults from trying to use _ColorDrawBuffers in FBO validation.

- intel: When validating an FBO's combined depth/stencil, use the given FBO.

Fabian Bieler (2):

- r600g: set address of pop instructions to next instruction
- glsl: fix lowering conditional returns in subroutines

Francisco Jerez (51):

- dri/nv04: Fix PGRAPH_ERRORs when running OA.
- dri/nv04: Mipmapping fixes.
- dri/nv04: Align SIFM transfer dimensions.
- dri/nv04: Fix up color mask.
- dri/nv04: Fix maximum texture size.
- dri/nv04: Fix provoking vertex.
- dri/nouveau: Update nouveau_class.h.
- dri/nouveau: Add some more extensions.
- dri/nouveau: Fix glRenderbufferStorage with DEPTH_COMPONENT as internal format.
- dri/nouveau: Don't request a fake front unnecessarily.
- dri/nouveau: Don't reemit the BO state in nouveau_state_emit().
- dri/nouveau: Cleanup references to the old FBOs on glMakeCurrent().
- meta: Don't bind the created texture object in init_temp_texture().
- dri/nv10: Fix the CLAMP texture wrap mode.
- dri/nv04: Use nvgl_wrap_mode().
- dri/nouveau: Remove unnecessary assertion.
- dri/nouveau: Cleanup more references to old FBOs and VBOs.
- dri/nv10-nv20: Fix texturing in some cases after a base level change.
- dri/nouveau: Fix software mipmap generation on 1x1 textures.
- dri/nouveau: Have a smaller amount of larger scratch buffers.
- dri/nouveau: Remove unnecessary flush.
- dri/nv10: Use fast Z clears.
- dri/nouveau: Minor cleanup.
- dri/nv10: Fake fast Z clears for pre-nv17 cards.
- dri/nouveau: Initialize tile_flags when allocating a render target.
- nouveau: Get larger push buffers.
- dri/nouveau: Force a "slow" Z clear if we're getting a new depth buffer.
- dri/nv20: Clear with the 3D engine.
- dri/nouveau: Don't assert(0) on compressed internal formats.
- dri/nv25: Bind a hierarchical depth buffer.
- dri/nouveau: Call _mesa_update_state() after framebuffer invalidation.

- dri/nouveau: Honor the access flags in nouveau_bufferobj_map_range.
- dri/nouveau: Tell the vbo module we want real hardware BOs.
- dri/nouveau: Split out the scratch helpers to a separate file.
- dri/nouveau: Avoid recursion in nouveau_bo_context_reset().
- dri/nouveau: Use a macro to iterate over the bound vertex attributes.
- dri/nouveau: Split out array handling to its own file.
- dri/nouveau: Optimize VBO binding re-emission.
- dri/nouveau: Keep small DYNAMIC_DRAW vertex buffers in system ram.
- dri/nouveau: Pipeline glTexSubImage texture transfers.
- dri/nouveau: Fix type promotion issue on 32bit platforms.
- dri/nouveau: Validate the framebuffer state on read buffer changes.
- dri/nouveau: Re-emit the BO state when coming back from a software fallback.
- meta: Don't leak alpha function/reference value changes.
- meta: Fix incorrect rendering of the bitmap alpha component.
- vbo: Avoid unnecessary copy to/from current in vertex format upgrade.
- meta: Don't try to disable cube maps if the driver doesn't expose the extension.
- meta: Handle bitmaps with alpha test enabled.
- dri/nouveau: Split hardware/software TNL instantiation more cleanly.
- dri/nouveau: Fix typo.
- dri/nouveau: Kill a bunch of ternary operators.

Fredrik Höglund (2):

- r600g: Fix texture sampling with swizzled coords
- r600g: fix pow(0, 0) evaluating to NaN

Guillermo S. Romero (1):

- r300g: Do not use buf param before checking for NULL.

Henri Verbeet (19):

- r600g: Flush upload buffers before draws instead of before flushes.
- r600g: Check for other references before checking for existing mappings in radeon_bo_pb_map_internal().
- r600g: Remove a redundant flush in r600_texture_transfer_map().
- r600g: Buffer object maps imply a wait.
- r600g: Respect PB_USAGE_UNSYNCHRONIZED in radeon_bo_pb_map_internal().
- Revert "r600g: Flush upload buffers before draws instead of before flushes."
- r600g: fix exports_ps to export a number not a mask.
- r600g: Mention AMD in the renderer string.
- r600g: Cleanup the fenced_bo list in r600_context_fini().
- r600g: Evergreen has two extra frac_bits for the sampler LOD state.

- r600: Evergreen has two extra frac_bits for the sampler LOD state.
- r600g: Add PIPE_FORMAT_L8A8_UNORM for Evergreen as well.
- r600g: Swizzle vertex data only once.
- r600g: Synchronize supported color formats between Evergreen and r600/r700.
- r600g: Fix the PIPE_FORMAT_L8A8_UNORM color swaps.
- r600g: Fix the PIPE_FORMAT_A8_UNORM color swap for Evergreen as well.
- r600g: Cleanup block bo references in r600_context_fini().
- r600g: Cleanup fetch shader resources in r600_pipe_shader_destroy().
- st/mesa: Handle wrapped depth buffers in st_copy_texsubimage().

Hui Qi Tay (10):

- llvmpipe: minor changes in llvm coefficient calcs
- draw: cliptest and viewport done in a single loop in vertex shader
- draw: added viewport and cliptest flags
- draw: sanitize llvm variant key
- draw: corrections for w coordinate
- draw: corrections to allow for different cliptest cases
- llvmpipe: Moved draw pipeline twoside function to llvm setup code
- llvmpipe: added llvm offset setup code
- llvmpipe: clean up polygon offset function in lp setup code
- llvmpipe: fix such that offset/twoside function only does in-place modification

Ian Romanick (102):

- glsl2: Refactor testing for whether a deref is of a matrix or array
- glsl2: Add flags to enable variable index lowering
- glsl: Add doxygen comments
- EGL DRI2: Silence piles of 'unused variable' warnings
- EGL DRI2: Silence 'missing initializer' warnings
- egl_glx: Silence piles of 'unused variable' warnings
- egl: Fix several 'comparison between signed and unsigned integer' warnings
- dri: Ensure that DRI driver cpp files are in tarballs
- mesa: Force GL_ARB_copy_buffer to always be enabled
- mesa: Force GL_SGIS_generate_mipmap to always be enabled
- Remove GL_MESA_packed_depth_stencil
- Remove GL_EXT_cull_vertex
- Regenerate files changed by previous commit
- Remove unnescessary initializations of UpdateTexturePalette
- ARB_texture_rg: Add GLX protocol support

- ARB_texture_rg: Correct some errors in RED / RG internal format handling
- ARB_texture_rg: Add GL_TEXTURE_{RED,GREEN}_SIZE query support
- ARB_texture_rg: Add GL_RED as a valid GL_DEPTH_TEXTURE_MODE
- ARB_texture_rg: Handle RED and RG the same as RGB for tex env
- ARB_texture_rg: Add R8, R16, RG88, and RG1616 internal formats
- ARB_texture_rg: Allow RED and RG textures as FBO color buffer attachments
- mesa: Enable GL_ARB_texture_rg in software paths
- i965: Enable GL_ARB_texture_rg
- mesa: Add ARB_texture_compression_rgtc as an alias for EXT_texture_compression_rgtc
- ARB_texture_rg: Add GL_COMPRESSED_{RED,RG} cases in _mesa_is_color_format
- mesa: Fix misplaced #endif
- mesa: Trivial correction to comment
- rgtc: Detect RGTC formats as color formats and as compressed formats
- docs: Add list of bugs fixed in 7.9
- docs: Import 7.9 release notes from 7.9 branch.
- docs: Import 7.8.x release notes from 7.8 branch.
- docs: download.html does not need to be updated for each release
- docs: Update mailing lines from sf.net to freedesktop.org
- docs: Import news updates from 7.9 branch
- docs: added news item for 7.9 release
- glsl: Fail linking if assign_attribute_locations fails
- glsl: Refactor 'layout' grammar to match GLSL 1.60 spec grammar
- glsl: Slight refactor of error / warning checking for ARB_fcc layout
- glsl: Clear type_qualifier using memset
- glsl: Wrap ast_type_qualifier contents in a struct in a union
- glsl: Regenerate files modified by previous commits
- glcpp: Add the define for ARB_explicit_attrib_location when present
- glcpp: Regenerate files changes by previous commit
- glsl: Add parser support for GL_ARB_explicit_attrib_location layouts
- glsl: Regenerate files changes by previous commit
- glsl: Track explicit location in AST to IR translation
- glsl: Add linker support for explicit attribute locations
- main: Enable GL_ARB_explicit_attrib_location for swrast
- intel: Enable GL_ARB_explicit_attrib_location
- glsl: Remove const decoration from inlined function parameters
- docs: skeleton for 7.10 release notes

- docs: Update status of GL 3.x related extensions
- mesa: Validate assembly shaders when GLSL shaders are used
- glsl: Fix incorrect assertion
- linker: Reject shaders that have unresolved function calls
- mesa: Silence unused variable warning
- mesa: Refactor validation of shader targets
- mesa: Clean up two 'comparison between signed and unsigned' warnings
- mesa: Clean up various 'unused parameter' warnings in shaderapi
- glsl: Slightly change the semantic of _LinkedShaders
- linker: Trivial indention fix
- i965: Fix indentation after commit 3322fbaf
- linker: Improve handling of unread/unwritten shader inputs/outputs
- glapi: Add GL_EXT_separate_shader_objects
- glapi: Commit files changed by previous commit
- mesa: Add infrastructure to track GL_EXT_separate_shader_objects
- mesa: Skeletal support for GL_EXT_separate_shader_objects
- mesa: Add display list support for GL_EXT_separate_shader_objects functions
- mesa: Track an ActiveProgram distinct from CurrentProgram
- Track separate programs for each stage
- swrast: Enable GL_EXT_separate_shader_objects in software paths
- intel: Enable GL_EXT_separate_shader_objects in Intel drivers
- docs: add GL_EXT_separate_shader_objects to release notes
- glsl: Fix incorrect gl_type of sampler2DArray and sampler1DArrayShadow
- ir_to_mesa: Refactor code for emitting DP instructions
- mesa: Allow query of MAX_SAMPLES with EXT_framebuffer_multisample
- glsl: Refactor is_vec_{zero,one} to be methods of ir_constant
- glsl: Simplify generation of swizzle for vector constructors
- glsl: Make is_zero and is_one virtual methods of ir_rvalue
- ir_to_mesa: Generate smarter code for some conditional moves
- glsl: Add ir_unop_sin_reduced and ir_unop_cos_reduced
- glsl: Eliminate assumptions about size of ir_expression::operands
- glsl: Add ir_rvalue::is_negative_one predicate
- glsl: Add unary ir_expression constructor
- glsl: Add ir_quadop_vector expression
- glsl: Fix matrix constructors with vector parameters
- i915: Disallow alpha, red, RG, and sRGB as render targets

- glsl: Use M_LOG2E constant instead of calling log2
- glsl: Lower ir_binop_pow to a sequence of EXP2 and LOG2
- i915: Request that POW instructions be lowered
- i915: Correctly generate unconditional KIL instructions
- glsl: Ensure that equality comparisons don't return a NULL IR tree
- i965: Correctly emit constants for aggregate types (array, matrix, struct)
- glsl: Inherrit type of declared variable from initializer
- linker: Ensure that unsized arrays have a size after linking
- linker: Fix regressions caused by previous commit
- glsl: Inherrit type of declared variable from initializer after processing assignment
- linker: Allow built-in arrays to have different sizes between shader stages
- ir_to_mesa: Don't generate swizzles for record derefs of non-scalar/vectors
- Refresh autogenerated file builtin_function.cpp.
- glsl: Allow less restrictive uses of sampler array indexing in GLSL <= 1.20
- docs: Import 7.9.1 release notes from 7.9 branch

Jakob Bornecrantz (27):

- rbug: Cast opcode to corrent int size
- rbug: Add function to get opcode name string
- scons: Link against talloc in the Gallium DRI drivers
- i915g: Link with wrapper sw winsys with scons
- tgsi: Actually care what check_soa_dependencies says
- tgsi: Fix missing test before check
- llvmpipe: Move makefile include to before targets
- wrapper: Fix spelling
- wrapper: Add a way to dewrap a pipe screen without destroying it
- egl: Remove unnecessary headers
- target-helpers: Remove per target software wrapper check
- graw: Tidy graw xlib scons file a bit
- scons: Remove old pipebuffer SConscript
- scons: Detabify
- scons: Check for pkg-config before trying to use it
- scons: Check for libdrm_[intellradeon] as well
- scons: Move dependancy checks to the main gallium scons file
- scons: Unify state tracker SConscripts
- galahad: Correct the name of the scons library
- graw: Use inline sw helper instead of roll your own loader

- libgl-xlib: Use sw helper instead of roll your own
- libgl-xlib: Use inline debug helper instead of non-inline version
- graw: Use inline debug helper instead of non-inline version
- gallium: Remove redundant sw and debug target helpers
- i915g: Improve debug printing for textures
- i915g: Make sure that new vbo gets updated
- st/mesa: Unbind all constant buffers

Jerome Glisse (75):

- r600g: alternative command stream building from context
- r600g: move chip class to radeon common structure
- r600g: use pipe context for flushing inside map
- r600g: add back reference check when mapping buffer
- r600g: directly allocate bo for user buffer
- r600g: fix multi buffer rendering
- r600g: occlusion query for new design
- r600g: flush color buffer after draw command
- r600g: disable shader rebuild optimization & account cb flush packet
- r600g: fix multiple occlusion query on same id
- r600g: initial evergreen support in new path
- r600g: fix typo in evergreen define (resource are in x range)
- r600g: move use_mem_constants flags for new designs structure alignment
- r600g: evergreen fix for new design
- r600g: fix compilation after change to evergreend.h
- r600g: fixup some evergreen register definitions
- r600g: fix evergreen new path
- r600g: fix reg definition
- r600g: fix evergreen new path
- r600g: bring over fix from old path to new path
- r600g: fix vertex resource & polygon offset
- r600g: disable early cull optimization when occlusion query running
- r600g: move around variables to share depth uncompression code
- r600g: use depth decompression in new path
- r600g: fix index buffer drawing
- r600g: build packet header once
- r600g: fix pointsprite & resource unbinding
- r600g: fix routing btw vertex & pixel shader

- r600g: fix occlusion query after change to block structure
- r600g: use ptr for blit depth uncompress function
- r600g: fix remaining piglit issue in new design
- r600g: switch to new design
- r600g: suspend/resume occlusion query around clear/copy
- r600g: avoid rebuilding the vertex shader if no change to input format
- r600g: use a hash table instead of group
- r600g: delete old path
- r600g: cleanup
- r600g: more cleanup
- r600g: use constant buffer instead of register for constant
- r600g: fix constant & literal src splitting, also fix mplayer gl2 shader
- evergreeng: avoid overlapping border color btw VS & PS
- r600g: indentation fixes
- r600g: rename radeon_ws_bo to r600_bo
- r600g: allow r600_bo to be a sub allocation of a big bo
- r600g: use r600_bo for relocation argument, simplify code
- r600g: rename radeon_ws_bo to r600_bo
- r600g: remove dead label & fix indentation
- r600g: store reloc information in bo structure
- r600g: improve bo flushing
- r600g: simplify block relocation
- r600g: userspace fence to avoid kernel call for testing bo busy status
- r600g: avoid segfault due to unintialized list pointer
- r600g: fix dirty state handling
- r600g: allow driver to work without submitting cmd to GPU
- gallium/noop: no operation gallium driver
- r600g: code cleanup (indent, trailing space, empty line . . .)
- r600g: fix occlusion query on evergreen (avoid lockup)
- r600g: add fetch shader capabilities
- r600g: dump raw shader output for debugging
- r600g: update polygon offset only when rasterizer or zbuffer change
- r600g: indentation fix
- r600g: more indentation fix + warning silencing + dead code removal
- r600g: build fetch shader from vertex elements
- r600g: avoid useless shader rebuild at draw call

- r600g: remove useless flush map
- r600g: remove dead code
- r600g: fix userspace fence against lastest kernel
- r600g: avoid using pb* helper we are loosing previous cpu cycle with it
- r600g: specialized upload manager
- r600g: indentation cleanup
- r600g: fix bo size when creating bo from handle
- r600g: fix segfault when translating vertex buffer
- r600g: need to reference upload buffer as the might still live accross flush
- r600g: properly unset vertex buffer
- r600g: avoid segfault

Joakim Sindholt (3):

- util/u_blitter: fix leak
- radeong: fix leaks
- r300g: silence guard band cap errors

Johann Rudloff (3):

- radeon: Implement EGL_MESA_no_surface_extension
- radeon: Implement __DRI_IMAGE and EGL_MESA_image_drm
- radeon: Implement GL_OES_EGL_image

John Doe (3):

- r600g: misc cleanup
- r600g: don't double count dirty block
- r600g: keep a mapping around for each bo

Jon TURNEY (1):

- Ensure -L$(TOP)/$(LIB_DIR) appears in link line before any -L in $LDFLAGS

José Fonseca (128):

- gallivm: Fix address register swizzle.
- gallivm: Start collecting bitwise arithmetic helpers in a new module.
- gallivm: Clamp indirect register indices to file_max.
- util: linearized sRGB values don't fit into 8bits
- llvmpipe: Default to no threading on single processor systems.
- tgsi: Don't ignore indirect registers in tgsi_check_soa_dependencies
- llvmpipe: Describe how to profile llvmpipe.
- llvmpipe: When failing free fs shader too.
- util: Flush stdout on util_format.
- gallivm: Add unorm support to lp_build_lerp()

- llvmpipe: Special case complementary and identify blend factors in SoA.
- llvmpipe: Make rgb/alpha bland func/factors match, when there is no alpha.
- draw: Prevent clipped vertices overflow.
- draw: Fulfil the new min_lod/max_lod/lod_bias/border_color dynamic state
- gallivm: Fetch the lod from the dynamic state when min_lod == max_lod.
- gallivm: Remove dead experimental code.
- llvmpipe: Decouple sampler view and sampler state updates.
- scons: New build= option, with support for checked builds.
- scons: New build= option, with support for checked builds.
- trace: Fix set_index_buffer and draw_vbo tracing.
- python/retrace: Handle set_index_buffer and draw_vbo.
- gallivm: Use SSE4.1's ROUNDSS/ROUNDSD for scalar rounding.
- gallivm: More comprehensive border usage logic.
- retrace: Handle clear_render_target and clear_depth_stencil.
- llvmpipe: Dump a few missing shader key flags.
- llvmpipe: Fix perspective interpolation for point sprites.
- llvmpipe: Fix sprite coord perspective interpolation of Q.
- gallivm: Take the type signedness in consideration in round/ceil/floor.
- gallivm: Use a faster (and less accurate) log2 in lod computation.
- gallivm: Fast implementation of iround(log2(x))
- gallivm: Combined ifloor & fract helper.
- gallivm: Only apply min/max_lod when necessary.
- gallivm: Compute lod as integer whenever possible.
- util: Cleanup util_pack_z_stencil and friends.
- llvmpipe: Cleanup depth-stencil clears.
- gallivm: Vectorize the rho computation.
- gallivm: Do not do mipfiltering when magnifying.
- gallivm: Simplify lp_build_mipmap_level_sizes' interface.
- gallivm: Don't compute the second mipmap level when frac(lod) == 0
- gallivm: Use lp_build_ifloor_fract for lod computation.
- gallivm: Clamp mipmap level and zero mip weight simultaneously.
- gallivm: Fix copy'n'paste typo in previous commit.
- gallivm: Implement brilinear filtering.
- gallivm: Use the wrappers for SSE pack intrinsics.
- gallivm: Avoid control flow for two-sided stencil test.
- gallivm: Warn when doing inefficient integer comparisons.

- gallivm: Move into the as much of the second level code as possible.
- llvmpipe: First minify the texture size, then broadcast.
- gallivm: Help for combined extraction and broadcasting.
- gallivm: Do size computations simultanously for all dimensions (AoS).
- llvmpipe: Prevent z > 1.0
- llvmpipe: Fix MSVC build. Enable the new SSE2 code on non SSE3 systems.
- gallivm: Handle code have ret correctly.
- util: Defined M_SQRT2 when not available.
- gallivm: Less code duplication in log computation.
- gallivm: Special bri-linear computation path for unmodified rho.
- gallivm: Don't generate Phis for execution mask.
- gallivm: Use varilables instead of Phis for cubemap selection.
- gallivm: Remove support for Phi generation.
- gallivm: Factor out the SI->FP texture size conversion for SoA path too
- gallivm: Simplify if/then/else implementation.
- gallivm: Cleanup the rest of the flow module.
- gallivm: Fix a long standing bug with nested if-then-else emission.
- gallivm: Allow to disable bri-linear filtering with GALLIVM_DEBUG=no_brilinear runtime option
- gallivm: Use variables instead of Phis in loops.
- gallivm: Pass texture coords derivates as scalars.
- llvmpipe: Remove outdated comment about stencil testing.
- gallivm: Eliminate unsigned integer arithmetic from texture coordinates.
- gallium: Define C99 restrict keyword where absent.
- tgsi: Export some names for some tgsi enums.
- gallivm: More detailed analysis of tgsi shaders.
- llvmpipe: Use lp_tgsi_info.
- llvmpipe: Do not dispose the execution engine.
- llvmpipe: Fix MSVC build.
- llmvpipe: improve mm_mullo_epi32
- gallivm: Name anonymous union.
- llvmpipe: Unbreak Z32_FLOAT.
- gallivm: More accurate float -> 24bit & 32bit unorm conversion.
- llvmpipe: Generalize the x8z24 fast path to all depth formats.
- llvmpipe: Fix depth-stencil regression.
- llvmpipe: Ensure z_shift and z_width is initialized.
- gallivm: Fix SoA cubemap derivative computation.

- llvmpipe: Fix bad refactoring.
- llvmpipe: Initialize bld ctx via lp_build_context_init instead of ad-hoc and broken code.
- gallivm: Comment lp_build_insert_new_block().
- gallivm: Add a note about SSE4.1's nearest mode rounding.
- llvmpipe: Don't test rounding of x.5 numbers.
- gallium: Avoid using __doc__ in python scripts.
- gallivm: always enable LLVMAddInstructionCombiningPass()
- gallivm: Remove the EMMS opcodes.
- mesa: Fix windows build (uint -> GLuint).
- scons: Revamp how to specify targets to build.
- scons: Fix MinGW cross-compilation.
- scons: Some pipe drivers are not portable for MSVC
- scons: Restore x11 tool behavior for backwards compatability.
- scons: Disable python state tracker when swig is not present.
- r600g: List recently added files in SConscript.
- scons: Add aliases for several pipe drivers.
- scons: i915 can't build on MSVC either.
- scons: Propagate installation targets.
- xorg/vmwgfx: Add missing source file to SConscript.
- st/xorg: Add missing n to error message.
- st/xorg: Detect libkms with scons too.
- xorg/vmwgfx: Link libkms when available.
- r600g: Swap the util_blitter_destroy call order.
- gallivm: Allocate TEMP/OUT arrays only once.
- libgl-gdi: Allow to pick softpipe/llvmpipe on runtime.
- scons: Use inline wrap helpers more consistently.
- svga: Use consistent hexadecimal representation on debug output.
- scons: Alias for svga
- wgl: Stub WGL_ARB_pbuffer support.
- wgl: More complete WGL_ARB_pbuffer support.
- svga: Silence debug printf.
- scons: Move MSVS_VERSION option to common module.
- vega: Remove extraneous ;
- retrace: Some fixes.
- util: C++ safe.
- wgl: Fix double free. Remove dead code.

- util: Plug leaks in util_destroy_gen_mipmap.
- util: __builtin_frame_address() doesn't work on mingw.
- util: Don't try to use imagehlp on mingw.
- wgl: Unreference the current framebuffer after the make_current call.
- WIN32_THREADS -> WIN32
- mapi: Hack to avoid vgCreateFont being generated as vgCreateFontA.
- wgl: Fix visual's buffer_mask configuration.
- mesa: Temporary hack to prevent stack overflow on windows
- mesa: Bump the number of bits in the register index.
- llvmpipe: Plug fence leaks.

Julien Cristau (1):

- Makefile: don't include the same files twice in the tarball

Keith Whitwell (89):

- llvmpipe: brackets around macro arg
- llvmpipe: remove duplicate code
- llvmpipe: return zero from floor_pot(zero)
- gallivm: make lp_build_sample_nop public
- llvmpipe: add LP_PERF flag to disable various aspects of rasterization
- llvmpipe: add DEBUG_FS to dump variant information
- llvmpipe: use llvm for attribute interpolant calculation
- graw: add frag-face shader
- llvmpipe: fix flatshading in new line code
- draw: don't apply flatshading to clipped tris with <3 verts
- llvmpipe: handle FACING interpolants in line and point setup
- llvmpipe: handle up to 8 planes in triangle binner
- llvmpipe: make debug_fs_variant respect variant->nr_samplers
- gallivm: don't apply zero lod_bias
- llvmpipe: fail gracefully on oom in scene creation
- llvmpipe: avoid overflow in triangle culling
- gallivm: special case conversion 4x4f to 1x16ub
- gallivm: round rather than truncate in new 4x4f->1x16ub conversion path
- llvmpipe: clean up setup_tri a little
- llvmpipe: add rast_tri_4_16 for small lines and points
- llvmpipe: fix off-by-one in tri_16
- llvmpipe: defer attribute interpolation until after mask and ztest
- llvmpipe: use alloca for fs color outputs

- llvmpipe: store zero into all alloca'd values
- llvmpipe: dump fragment shader ir and asm when LP_DEBUG=fs
- gallivm: specialized x8z24 depthtest path
- gallivm: prefer blendvb for integer arguments
- gallivm: simpler uint8->float conversions
- llvmpipe: try to be sensible about whether to branch after mask updates
- llvmpipe: clean up shader pre/postamble, try to catch more early-z
- llvmpipe: simplified SSE2 swz/unswz routines
- llvmpipe: try to do more of rast_tri_3_16 with intrinsics
- llvmpipe: add debug helpers for epi32 etc
- llvmpipe: try to keep plane c values small
- llvmpipe: fix typo in last commit
- gallium: move sse intrinsics debug helpers to u_sse.h
- r600g: add missing file to sconscript
- gallivm: don't branch on KILLs near end of shader
- Revert "llvmpipe: try to keep plane c values small"
- llvmpipe: make sure intrinsics code is guarded with PIPE_ARCH_SSE
- llvmpipe: don't try to emit non-existent color outputs
- r600/drm: fix segfaults in winsys create failure path
- r600g: emit hardware linewidth
- r600g: handle absolute modifier in shader translator
- llvmpipe: reintroduce SET_STATE binner command
- llvmpipe: don't pass frontfacing as a float
- llvmpipe: slightly shrink the size of a binned triangle
- llvmpipe: don't store plane.ei value in binned data
- gallium: move some intrinsics helpers to u_sse.h
- llvmpipe: do plane calculations with intrinsics
- llvmpipe: use aligned loads/stores for plane values
- llvmpipe: fix non-sse build after recent changes
- llvmpipe: check shader outputs are non-null before using
- llvmpipe: validate color outputs against key->nr_cbufs
- llvmpipe: clean up fields in draw_llvm_variant_key
- llvmpipe: remove setup fallback path
- llvmpipe: fail cleanly on malloc failure in lp_setup_alloc_triangle
- Merge remote branch 'origin/master' into lp-setup-llvm
- llvmpipe: remove unused file

- llvmpipe: remove unused arg from jit_setup_tri function
- Merge branch 'llvm-cliptest-viewport'
- draw: make sure viewport gets updated in draw llvm shader
- llvmpipe: turn off draw offset/twoside when we can handle it
- llvmpipe: avoid generating tri_16 for tris which extend past tile bounds
- llvmpipe: guard against NULL task->query pointer
- st/mesa: unbind constant buffer when not in use
- r600g: propagate usage flags in texture transfers
- r600g: propogate resource usage flags to winsys, use to choose bo domains
- r600g: use a buffer in GTT as intermediate on texture up and downloads
- r600g: remove unused flink, domain fields from r600_resource
- r600g: set hardware pixel centers according to gl_rasterization_rules
- evergreeng: protect against null constant buffers
- r600g: don't call debug_get_bool_option for tiling more than once
- evergreeng: respect linewidth state, use integer widths only
- evergreeng: set hardware pixelcenters according to gl_rasterization_rules
- r600g: avoid recursion with staged uploads
- r600g: attempt to turn on DXTn formats
- r600g: translate ARR instruction
- r600: fix my pessimism about PIPE_TRANSFER_x flags
- ws/r600: match bo_busy shared/fence logic in bo_wait
- r600g: guard experimental s3tc code with R600_ENABLE_S3TC
- r600g: do not try to use staging resource for depth textures
- r600g: enforce minimum stride on render target texture images
- llvmpipe: fix up twoside after recent changes
- llvmpipe: twoside for specular color also
- Merge branch 'lp-offset-twoside'
- llvmpipe: raise dirty flag on transfers to bound constbuf
- llvmpipe: remove misleading debug string
- llvmpipe: shortcircuit some calls to set_scene_state

Kenneth Graunke (94):

- glsl: Change from has_builtin_signature to has_user_signature.
- glsl: Don't print blank (function . . .) headers for built-ins.
- glsl: Properly handle nested structure types.
- glsl/builtins: Fix equal and notEqual builtins.
- glsl/builtins: Switch comparison functions to just return an expression.

- glsl: Add comments to clarify the types of comparison binops.
- glsl: Fix broken handling of ir_binop_equal and ir_binop_nequal.
- glsl: "Copyright", not "Constantright"
- i965: Fix incorrect batchbuffer size in gen6 clip state command.
- i965: Use logical-not when emitting ir_unop_ceil.
- glsl: Add front-end support for the "trunc" built-in.
- glsl: Refresh autogenerated file builtin_function.cpp.
- i965: Use RNDZ for ir_unop_trunc in the new FS.
- i965: Correctly emit the RNDZ instruction.
- i965: Clean up a warning in the old fragment backend.
- glsl: Add a new ir_unop_round_even opcode for GLSL 1.30's roundEven.
- glsl: Add front-end support for GLSL 1.30's roundEven built-in.
- i965: Add support for ir_unop_round_even via the RNDE instruction.
- glsl: Add support for the 1.30 round() built-in.
- glsl: Refresh autogenerated file builtin_function.cpp.
- glsl: Don't return NULL IR for erroneous bit-shift operators.
- i965: Add missing "break" statement.
- glsl: Fix copy and paste error in ast_bit_and node creation.
- glsl: Regenerate parser files.
- i965: Remove unused variable.
- glsl: Remove useless ir_shader enumeration value.
- mesa: Remove FEATURE_ARB_shading_language_120 macro.
- glcpp: Return NEWLINE token for newlines inside multi-line comments.
- glcpp: Refresh autogenerated lexer file.
- glsl: Add support for GLSL 1.30's modf built-in.
- glsl: Refresh autogenerated file builtin_function.cpp.
- generate_builtins.py: Output large strings as arrays of characters.
- Refresh autogenerated file builtin_function.cpp.
- glsl: Fix constant component count in vector constructor emitting.
- Fix build on systems where "python" is python 3.
- i965: Add bit operation support to the fragment shader backend.
- glsl: Remove unused ARRAY_SIZE macro.
- glsl/builtins: Rename 'x' to 'y_over_x' in atan(float) implementation.
- glsl/builtins: Clean up some ugly autogenerated code in atan.
- Refresh autogenerated file builtin_function.cpp.
- glsl: Don't print a useless space at the end of an S-Expression list.

- ir_reader: Return a specific ir_dereference variant.
- ir_reader: Remove useless error check.
- ir_reader: Fix some potential NULL pointer dereferences.
- ir_dead_functions: Actually free dead functions and signatures.
- glsl: Remove unnecessary "unused variable" warning suppression.
- glsl: Remove GLSL_TYPE_FUNCTION define.
- glsl: Convert glsl_type::base_type from #define'd constants to an enum.
- glsl: Rework reserved word/keyword handling in the lexer.
- glsl: Add new keywords and reserved words for GLSL 1.30.
- glsl: Add support for the 'u' and 'U' unsigned integer suffixes.
- glsl: Refresh autogenerated lexer and parser files.
- generate_builtins.py: Fix inconsistent use of tabs and spaces warning.
- glsl: Implement the asinh, acosh, and atanh built-in functions.
- glsl: Refresh autogenerated file builtin_function.cpp.
- glsl: Add constant expression handling for asinh, acosh, and atanh.
- glsl: Remove unused and out of date Makefile.am.
- glsl: Rename various ir_* files to lower_* and opt_*.
- glcpp: Define GL_FRAGMENT_PRECISION_HIGH if GLSL version >= 1.30.
- Refresh autogenerated glcpp parser.
- glsl: Fix constant expression handling for <, >, <=, >= on vectors.
- glsl: Unconditionally define GL_FRAGMENT_PRECISION_HIGH in ES2 shaders.
- Regenerate glcpp parser.
- glsl: Reimplement the "cross" built-in without ir_binop_cross.
- Refresh autogenerated file builtin_function.cpp.
- glsl: Remove the ir_binop_cross opcode.
- glsl: Refactor get_num_operands.
- glsl: Simplify a type check by using type->is_integer().
- glsl: Combine many instruction lowering passes into one.
- mesa: Fix glGet of ES2's GL_MAX_*_VECTORS properties.
- glsl: Don't inline function prototypes.
- glsl: Use do_common_optimization in the standalone compiler.
- glsl: Add a virtual as_discard() method.
- glsl: Refactor out cloning of function prototypes.
- glsl: Lazily import built-in function prototypes.
- glsl: Remove anti-built-in hacks from the print visitor.
- glsl/linker: Free any IR discarded by optimization passes.

- glsl: Add an optimization pass to simplify discards.
- glsl: Add a lowering pass to move discards out of if-statements.
- glsl: Remove "discard" support from lower_jumps.
- glsl: Add comments to lower_jumps (from the commit message).
- ir_print_visitor: Print out constant structure values.
- glsl: Factor out code which emits a new function into the IR stream.
- symbol_table: Add support for adding a symbol at top-level/global scope.
- glsl: Properly add functions during lazy built-in prototype importing.
- glcpp: Don't emit SPACE tokens in conditional_tokens production.
- Refresh autogenerated glcpp parser.
- glsl: Clean up code by adding a new is_break() function.
- glsl: Consider the "else" branch when looking for loop breaks.
- Remove OES_compressed_paletted_texture from the ES2 extension list.
- glsl/builtins: Compute the correct value for smoothstep(vec, vec, vec).
- glsl: Support if-flattening beyond a given maximum nesting depth.
- i965: Flatten if-statements beyond depth 16 on pre-gen6.
- i965: Internally enable GL_NV_blend_square on ES2.

Kristian Høgsberg (16):

- glx: Hold on to drawables if we're just switching to another context
- intel: Fix GL_ARB_shading_language_120 commit
- dri2: Make createImageFromName() take a __DRIscreen instead of __DRIcontext
- glx: Invalidate buffers after binding a drawable
- dri: Pass the __DRIscreen and the __DRIscreen private back to image lookup
- glx: Only remove drawables from the hash when we actually delete them
- gles2: Add GL_EXT_texture_format_BGRA8888 support
- Get rid of GL/internal/glcore.h
- gl: Remove unused GLcontextModes fields
- Rename GLvisual and __GLcontextModes to struct gl_config
- Drop GLframebuffer typedef and just use struct gl_framebuffer
- Drop GLcontext typedef and use struct gl_context instead
- Drop the "neutral" tnl module
- Only install vtxfmt tables for OpenGL
- i965: Don't write mrf assignment for pointsize output
- docs: Fix MESA_drm_image typo

Krzysztof Smiechowicz (1):

- nvfx: Pair os_malloc_aligned() with os_free_aligned().

Luca Barbieri (84):

- auxiliary: fix unintended fallthrough
- glsl: add pass to lower variable array indexing to conditional assignments
- auxiliary: fix depth-only and stencil-only clears
- gallium: avoid the C++ keyword "template" in sw_winsys.h
- softpipe: make z/s test always pass if no zsbuf, instead of crashing
- tgsi: add switch/case opcodes to tgsi_opcode_tmp.h
- softpipe: fix whitespace
- d3d1x: add new Direct3D 10/11 COM state tracker for Gallium
- d3d1x: add blob and signature extraction APIs
- d3d1x: fix compilation with recent Wine versions installed
- d3d1x: add missing file
- d3d1x: actually enable and fix blob apis
- d3d1x: fix build with compilers other than GCC 4.5
- d3d1x: add template parameters to base class ctor calls for GCC 4.4
- d3d1x: fix GCC 4.1/4.2 build
- d3d1x: ignore errors while building docs
- d3d1x: attempt to fix/workaround bug #30322
- nvfx: remove gl_PointCoord hack
- glx: decouple dri2.c and GLX, fixing Gallium EGL and d3d1x build
- winsys: automatically build sw winsys needed by EGL and d3d1x
- d3d1x: don't build progs automatically
- d3d1x: add missing memory barrier
- d3d1x: link with CXXFLAGS
- d3d1x: fix cf analysis
- d3d1x: fix warning
- d3d1x: fix segfault when hashing
- d3d1x: destroy native_display on adapter destruction
- d3d1x: fix GUID declarations
- d3d1x: redesign the HWND resolver interface
- d3d1x: fix API name
- d3d1x: define GUIDs in the normal way
- d3d1x: add Wine dlls (tri, tex working, but no other testing)
- d3d1x: properly reference count the backend
- d3d1x: fix deadlocks on non-recursive mutex
- d3d1x: bind NULL CSOs before destroying default CSOs on context dtor

- d3d1x: initialize the mutex
- d3d1x: autogenerate shader enums and text from def files
- d3d1x: s/tpf/sm4/g
- d3d1x: normalize whitespace
- d3d1x: remove specstrings
- d3d1x: minifix
- d3d1x: rename context params
- d3d11: rename screen params
- d3d1x: rename params in misc and objects
- d3d1x: rename parameters in dxgi
- d3d11: obliterate IDL parameter names
- d3d1x: remove specstrings.h include
- d3d1x: flush the pipe context when presenting
- d3d1x: remove another include specstrings.h
- d3d1x: flush properly
- d3d1x: add missing guid.cpp
- d3d1x: fix build without system EGL/egl.h
- d3d1x: add autogenerated files as prerequisites, so make builds them
- d3d1x: obliterate IDL parameter names from d3d10.idl from Wine too
- d3d1x: add shader dumping
- d3d1x: add untested support for geometry shader translation
- d3d1x: don't assert on unsupported resource types
- d3d1x: fix CheckMultisampleQualityLevels
- d3d1x: draw to the correct buffer
- d3d1x: fix linking of dxbc2tgsi
- nvfx: allow setting NULL constant buffers
- nvfx: add RGB framebuffer format support in addition to BGR
- d3d1x: don't crash on drivers not supporting vertex or geometry sampling
- d3d1x: assert if X visual is not among enumerated visuals
- d3d1x: stop using GLX in demos, just use the default visual
- d3d1x: CRLF -> LF in progs
- mesa: make makedepend an hard requirement
- gallium: add $(PROGS_DEPS) as dependencies for $(PROGS)
- d3d1x: fix parallel build
- d3d1x: add private gitignore file
- d3d1x: fix progs linking if not all EGL platforms are enabled

- d3d1x: link progs with CXXFLAGS
- d3d11: advertise IDXGIDevice1, not just IDXGIDevice
- d3d11: ignore StructureByteStride
- d3d1x: link to libdrm for X11 platform too
- ureg: support centroid interpolation
- d3d1x: support centroid interpolation
- d3d1x: properly support specifying MipLevels as 0
- d3d1x: put proper calling convention in headers, fixes 64-bit builds
- d3d1x: rework DXGI for occlusion testing and default width/height
- d3d1x: fix Map
- d3d11: fix reference counting so devices get freed
- d3d1x: work around crash in widl
- glsl: Unroll loops with conditional breaks anywhere (not just the end)

Lucas Stach (1):

- nvfx: fill PIPE_CAP_PRIMITIVE_RESTART and PIPE_CAP_SHADER_STENCIL_EXPORT

Marek Olšák (100):

- r300g: prevent creating multiple winsys BOs for the same handle
- r300g/swtcl: fix CS overrun
- st/mesa: fix assertion failure in GetTexImage for cubemaps
- util: make calling remove_from_list multiple times in a row safe
- r300g: fixup long-lived BO maps being incorrectly unmapped when flushing
- r300g: make accessing map_list and buffer_handles thread-safe
- r300g: fix a copy-paste typo for logging
- r300g: fix the border color for every format other than PIPE_FORMAT_B8G8R8A8
- Build r300g by default
- util: fix util_pack_color for B4G4R4A4
- r300g: fix macrotiling on R350
- r300g: code cleanups
- r300/compiler: fix projective mapping of 2D NPOT textures
- r300/compiler: do not use copy propagation if SaturateMode is used
- r300/compiler: fix shadow sampling with swizzled coords
- r300g: add support for 3D NPOT textures without mipmapping
- r300g: fix swizzling of texture border color
- configure.ac: look for libdrm_radeon before building gallium/r300,r600
- configure.ac: do not build xorg-r300g by default
- Makefile: ensure Gallium's Makefile.xorg and SConscript.dri are in the tarball

- r300g: add support for formats beginning with X, like X8R8G8B8
- r300g: fix conditional rendering in non-wait path
- r300g: add support for R8G8 colorbuffers
- r300g: add support for L8A8 colorbuffers
- update release notes for Gallium
- r300g: fix microtiling for 16-bits-per-channel formats
- r300g: do not print get_param errors in non-debug build
- r300g: say no to PIPE_CAP_STREAM_OUTPUT and PIPE_CAP_PRIMITIVE_RESTART
- mesa: allow FBO attachments of formats LUMINANCE, LUMINANCE_ALPHA, and INTENSITY
- r300g: fix texture border for 16-bits-per-channel formats
- st/mesa: support RGBA16 and use it for RGBA12 as well
- r300g: add a default channel ordering of texture border for unhandled formats
- r300g: mention ATI in the renderer string
- r300g: rename has_hyperz -> can_hyperz
- r300g: turn magic numbers into names in the hyperz code
- gallium: add CAPs for indirect addressing and lower it in st/mesa when needed
- tgsi: fill out CAPs for indirect addressing
- i915g: fill out CAPs for indirect addressing
- i965g: fill out CAPs for indirect addressing
- nv50: fill out CAPs for indirect addressing
- nvfx: fill out CAPs for indirect addressing
- r300g: fill out CAPs for indirect addressing
- r600g: fill out CAPs for indirect addressing
- svga: fill out CAPs for indirect addressing
- r300g: fix texture border color for all texture formats
- r300g: clean up redundancy in draw functions
- r300g: return shader caps from Draw for SWTCL vertex shaders
- r300g: remove the hack with OPCODE_RET
- r300g: print FS inputs uninitialized due to hardware limits to stderr
- r300g: fix rendering with no vertex elements
- st/mesa: enable ARB_explicit_attrib_location and EXT_separate_shader_objects
- docs: add GL 4.1 status
- gallium: add PIPE_SHADER_CAP_SUBROUTINES
- st/mesa: set MaxUniformComponents
- u_blitter: use PIPE_TRANSFER_DISCARD to prevent cpu/gpu stall
- r300/compiler: fix rc_rewrite_depth_out for it to work with any instruction

- r300/compiler: remove duplicate function rc_mask_to_swz
- r300/compiler: add a function for swizzling a mask
- r300/compiler: move util functions to radeon_compiler_util
- u_blitter: interpolate clear color using a GENERIC varying instead of COLOR
- st/mesa: fix texture border color for RED and RG base formats
- util: rename u_mempool -> u_slab
- r300g: fix texture border color once again
- r300/compiler: implement and lower OPCODE_CLAMP
- ir_to_mesa: Add support for conditional discards.
- r300g: fix texture swizzling with compressed textures on r400-r500
- r300g: disable ARB_texture_swizzle if S3TC is enabled on r3xx-only
- r300g: fix up cubemap texture offset computation
- r300/compiler: disable the swizzle lowering pass in vertex shaders
- r300g: fix build
- r300g: use internal BO handle for add_buffer and write_reloc
- r300g: implement simple transfer_inline_write for buffers
- mesa, st/mesa: fix gl_FragCoord with FBOs in Gallium
- r300g: fix pointer arithmetic with void* in transfer_inline_write
- r300g: do not remove unused constants if we are not near the limit
- r300g: add capability bit index_bias_supported
- r300g: one more r500_index_bias_supported leftover
- r300g: do not use the index parameter in set_constant_buffer
- r300g: cleanup winsys
- r300g: optimize looping over atoms
- st/mesa: initialize key in st_vp_varient
- u_blitter: use util_is_format_compatible in the assert
- r300g: cache packet dwords of 3D_LOAD_VBPNTR in a command buffer if possible
- r300g: validate buffers only if any of bound buffers is changed
- r300g: also revalidate the SWTCL vertex buffer after its reallocation
- r300/compiler: don't terminate regalloc if we surpass max temps limit
- r300/compiler: add a function to query program stats (alu, tex, temps..)
- r300/compiler: cleanup rc_run_compiler
- r300/compiler: do not print pair/tex/presub program stats for vertex shaders
- r300/compiler: handle DPH and XPD in rc_compute_sources_for_writemask
- r300/compiler: make lowering passes possibly use up to two less temps
- r300/compiler: remove at least unused immediates if externals cannot be removed

- r300/compiler: fix LIT in VS
- r300/compiler: fix swizzle lowering with a presubtract source operand
- r300g: fix rendering with a vertex attrib having a zero stride
- r300g: finally fix the texture corruption on r3xx-r4xx
- r300g/swtcl: re-enable LLVM
- r300g: mark vertex arrays as dirty after a buffer_offset change
- mesa: fix texel store functions for some float formats
- r300/compiler: disable the rename_regs pass for loops

Mario Kleiner (1):

- mesa/r300classic: Fix dri2Invalidate/radeon_prepare_render for page flipping.

Mathias Fröhlich (3):

- r300g: Avoid returning values in a static array, fixing a potential race
- r600g: Only compare active vertex elements
- st/mesa: Set PIPE_TRANSFER_DISCARD for GL_MAP_INVALIDATE_RANGE/BUFFFER_BIT

Michal Krol (10):

- svga: Fix relative addressing translation for pixel shaders.
- svga: Integer constant register file has a separate namespace.
- tgsi/exec: Cleanup the remaining arithmetic instructions.
- tgsi/exec: Get rid of obsolete condition codes.
- tgsi/build: Reduce interface clutter.
- graw/gdi: Initial commit.
- scons: Hook-up graw-gdi target.
- graw/gdi: Fix window dimensions.
- os: Open file streams in binary mode.
- graw: Export graw_save_surface_to_file().

Nicolas Kaiser (26):

- swrast: remove duplicated include
- egl: remove duplicated include
- gallium/rtasm: remove duplicated include
- gallium/util: remove duplicated include
- gallium/i915: remove duplicated include
- gallium/llvmpipe: remove duplicated include
- gallium/softpipe: remove duplicated include
- gallium/st: remove duplicated includes
- gallium/winsys: remove duplicated include
- glx: remove duplicated include

- dri/common: remove duplicated include
- dri/i810: remove duplicated include
- dri/i915: remove duplicated include
- dri/i965: remove duplicated include
- dri/intel: remove duplicated include
- dri/mga: remove duplicated include
- dri/r128: remove duplicated include
- dri/r300: remove duplicated include
- dri/r600: remove duplicated include
- dri/radeon: remove duplicated includes
- dri/savage: remove duplicated include
- main: remove duplicated includes
- math: remove duplicated includes
- st: remove duplicated include
- i965g: use Elements macro instead of manual sizeofs
- nv50: fix always true conditional in shader optimization

Orion Poplawski (1):

- osmesa: link against libtalloc

Owen W. Taylor (1):

- r600g: Fix location for clip plane registers

Peter Clifton (3):

- intel: Fix emit_linear_blit to use DWORD aligned width blits
- intel: Add assert check for blitting alignment.
- meta: Mask Stencil.Clear against stencilMax in _mesa_meta_Clear

Robert Hooker (2):

- intel: Add a new B43 pci id.
- egl_dri2: Add missing intel chip ids.

Roland Scheidegger (16):

- gallivm: fix copy&paste bug
- gallivm: don't use URem/UDiv when calculating offsets for blocks
- gallivm: optimize yuv decoding
- gallivm: fix trunc/itrunc comment
- gallivm: faster iround implementation for sse2
- gallivm: replace sub/floor/ifloor combo with ifloor_fract
- gallivm: optimize some tex wrap mode calculations a bit
- gallivm: more linear tex wrap mode calculation simplification

- gallivm: avoid unnecessary URem in linear wrap repeat case
- gallivm: optimize soa linear clamp to edge wrap mode a bit
- gallivm: make use of new iround code in lp_bld_conv.
- gallivm: fix different handling of [non]normalized coords in linear soa path
- gallivm: only use lp_build_conv 4x4f -> 1x16 ub fastpath with sse2
- r200: fix r200 large points
- mesa: remove unneeded DD_POINT_SIZE and DD_LINE_WIDTH tricaps
- gallium: support for array textures and related changes

Shuang He (1):

- mesa: allow GLfixed arrays for OpenGL ES 2.0

Stephan Schmid (1):

- r600g: fix relative addressing when splitting constant accesses

Thomas Hellstrom (21):

- st/xorg: Don't try to use option values before processing options
- xorg/vmwgfx: Make vmwarectrl work also on 64-bit servers
- st/xorg: Add a customizer option to get rid of annoying cursor update flicker
- xorg/vmwgfx: Don't hide HW cursors when updating them
- st/xorg: Don't try to remove invalid fbs
- st/xorg: Fix typo
- st/xorg, xorg/vmwgfx: Be a bit more frendly towards cross-compiling environments
- st/xorg: Fix compilation errors for Xservers compiled without Composite
- st/xorg: Don't use deprecated x*alloc / xfree functions
- xorg/vmwgfx: Don't use deprecated x*alloc / xfree functions
- st/xorg: Fix compilation for Xservers >= 1.10
- mesa: Make sure we have the talloc cflags when using the talloc headers
- egl: Add an include for size_t
- mesa: Add talloc includes for gles
- st/egl: Fix build for include files in nonstandard places
- svga/drm: Optionally resolve calls to powf during link-time
- gallium/targets: Trivial crosscompiling fix
- st/xorg: Add a function to flush pending rendering and damage
- gallium/targets/xorg-vmwgfx: Xv fixes
- xorg/vmwgfx: Flush even if we don't autopaint the color key
- xorg/vmwgfx: Don't clip video to viewport

Tilman Sauerbeck (35):

- r600g: Fixed a bo leak in r600_blit_state_ps_shader().

- r600g: Use clamped math for RCP and RSQ.
- r600g: Formatting fixes.
- r600g: Added DB_SHADER_CONTROL defines.
- r600g: Only set PA_SC_EDGERULE on rv770 and greater.
- r600g: Enable PIPE_SHADER_CAP_TGSI_CONT_SUPPORTED.
- r600g: Fixed the shift in S_02880C_KILL_ENABLE.
- glsl2: Empty functions can be inlined.
- glsl2: Fixed cloning of ir_call error instructions.
- r600g: Added support for TGSI_SEMANTIC_FACE.
- gallium/docs: Fixed a typo in the SCS opcode description.
- r600g: Honour destination operand's writemask in the SCS implementation.
- r600g: Implemented the Z and W component write for the SCS opcode.
- python/tests: Fixed tri.py for API and TGSI syntax changes.
- r600g: Removed debug code.
- gallium/docs: The RET opcode may appear anywhere in a subroutine.
- r600g: Destroy the blitter.
- r600g: Fixed two texture surface leaks in r600_blit_uncompress_depth().
- r600g: Cleaned up index buffer reference handling in the draw module.
- r600g: Fixed r600_vertex_element leak.
- r600g: Added r600_pipe_shader_destroy().
- r600g: Also clear bc data when we're destroying a shader.
- r600g: In radeon_bo(), call LIST_INITHEAD early.
- r600g: Destroy the blitter.
- r600g: Removed unused 'ptr' argument from radeon_bo().
- r600g: Made radeon_bo_pb_map_internal() actually call radeon_bo_map().
- r600g: Fixed unmap condition in radeon_bo_pb_destroy().
- r600g: Made radeon_bo::map_count signed.
- r600g: We don't support PIPE_CAP_PRIMITIVE_RESTART.
- r600g: Delete custom_dsa_flush on shutdown.
- r600g: Fixed two memory leaks in winsys.
- r600g: Destroy the winsys in r600_destroy_screen().
- st/mesa: Reset the index buffer before destroying the pipe context.
- st/mesa: Reset the constant buffers before destroying the pipe context.
- r600g: Removed duplicated call to tgsi_split_literal_constant().

Timo Wiren (1):

- Fix typos in comments and debug output strings.

Tom Fogal (3):

- Implement x86_64 atomics for compilers w/o intrinsics.
- Prefer intrinsics to handrolled atomic ops.
- Revert "Prefer intrinsics to handrolled atomic ops."

Tom Stellard (32):

- r300/compiler: Refactor the pair instruction data structures
- r300g: Always try to build libr300compiler.a
- r300/compiler: Fix two mistakes in the presubtract optimization pass.
- r300/compiler: Add more helper functions for iterating through sources
- r300/compiler: Print immediate values after "dead constants" pass
- r300/compiler: radeon_remove_constants.c: fix indentation
- r300/compiler: Use rc_for_all_reads_src() in "dead constants" pass
- r300/compiler: Fix segfault in error path
- r300/compiler: Don't use rc_error() unless the error is unrecoverable
- r300/compiler: Don't merge instructions that write output regs and ALU result
- r300/compiler: Create a helper function for merging presubtract sources
- r300/compiler: Fix incorrect assumption
- r300/compiler: Clear empty registers after constant folding
- r300/compiler: Add a new function for more efficient dataflow analysis
- r300g: Add new debug option for logging vertex/fragment program stats
- r300/compiler: Use rc_get_readers_normal() for presubtract optimizations
- r300/compiler: Don't clobber presubtract sources during optimizations
- r300/compiler: Don't track readers into an IF block.
- r300/compiler: Make sure presubtract sources use supported swizzles
- r300/compiler: Fix register allocator's handling of loops
- r300/compiler: Fix instruction scheduling within IF blocks
- r300/compiler: Use zero as the register index for unused sources
- r300/compiler: Ignore alpha dest register when replicating the result
- r300/compiler: Add rc_get_readers()
- r300/compiler: Handle BREAK and CONTINUE in rc_get_readers()
- r300/compiler: Track readers through branches in rc_get_readers()
- r300/compiler: Convert RGB to alpha in the scheduler
- r300/compiler: Use presubtract operations as much as possible
- r300/compiler: Enable rename_reg pass for r500 cards
- r300/compiler: Add a more efficient version of rc_find_free_temporary()
- r300/compiler: Don't allow presubtract sources to be remapped twice

- r300/compiler: Fix black terrain in Civ4

Victor Tseng (1):

- egl/i965: include inline_wrapper_sw_helper.h

Viktor Novotný (6):

- dri/nouveau: Import headers from rules-ng-ng
- dri/nouveau: nv04: Use rules-ng-ng headers
- dri/nouveau: nv10: Use rules-ng-ng headers
- dri/nouveau nv20: Use rules-ng-ng headers
- dri/nouveau: Remove nouveau_class.h, finishing switch to rules-ng-ng headers
- dri/nouveau: Clean up magic numbers in get_rt_format

Vinson Lee (214):

- llvmpipe: Remove unnecessary header.
- r600g: Remove unnecessary headers.
- mesa: Include missing header in program.h.
- glsl: Fix 'format not a string literal and no format arguments' warning.
- r600g: Silence uninitialized variable warning.
- r600g: Silence uninitialized variable warning.
- nvfx: Silence uninitialized variable warnings.
- r600g: Silence uninitialized variable warning.
- r600g: Silence uninitialized variable warning.
- r600g: Silence uninitialized variable warning.
- r600g: Silence unused variable warning.
- nv50: Update files in SConscript to match Makefile.
- nv50: Remove unnecessary headers.
- nv50: Silence uninitialized variable warning.
- nv50: Silence uninitialized variable warning.
- nv50: Silence uninitialized variable warning.
- gallivm: Remove unnecessary headers.
- draw: Remove unnecessary header.
- nv50: Silence uninitialized variable warnings.
- nv50: Fix 'control reaches end of non-void function' warning.
- mesa/st: Silence uninitialized variable warning.
- gallivm: Remove unnecessary header.
- r600g: Remove unnecessary header.
- r600g: Remove unnecessary headers.
- r600g: Fix implicit declaration warning.

- r600g: Fix memory leak on error path.
- r600g: Silence uninitialized variable warning.
- r600g: Silence unused variable warnings.
- mesa: bump version to 7.10
- ir_to_mesa: Remove unused member array_indexed from struct statevar_element.
- mesa: Silence "'valid_texture_object' defined but not used" warning.
- x86: Silence unused variable warning on Mac OS X.
- glsl: Fix 'control reaches end of non-void function' warning.
- nvfx: Remove const qualifer from nvfx_vertprog_translate.
- nvfx: Silence uninitialized variable warnings.
- r600g: Remove unused variable.
- nv50: Silence missing initializer warning.
- nv50: Remove dead initialization.
- nv50: Remove dead initialization.
- tgsi: Remove duplicate case value.
- glut: Define markWindowHidden for non-Windows only.
- glut: Define eventParser for non-Windows only.
- r300g: Silence uninitialized variable warning.
- intel: Fix implicit declaration of function '_mesa_meta_Bitmap' warning.
- mesa: Remove unnecessary headers.
- r600g: Remove unnecessary header.
- unichrome: Remove unnecessary header.
- intel: Remove unnecessary headers.
- r600g: Remove unused variable.
- r600g: Disable unused variables.
- r600g: Remove unused variable.
- r600g: Silence 'control reaches end of non-void function' warning.
- r600g: Remove unused variable.
- r600g: Remove unused variable.
- r600g: Disable unused variables.
- intel: Remove unnecessary header.
- st/dri: Remove unnecessary header.
- r600g: Remove unused variable.
- r300g: Remove unused variable.
- r600g: Don't return a value in function returning void.
- r600g: Remove unused variables.

- r600g: Include p_compiler.h instead of malloc.h.
- r600g: Silence uninitialized variable warnings.
- scons: Add MinGW-w64 prefixes for MinGW build.
- dri: Add GET_PROGRAM_NAME definition for Mac OS X.
- scons: Add program/sampler.cpp to SCons build.
- mesa: Fix printf format warning.
- mesa: Fix printf format warning.
- mesa: Fix printf format warning.
- r300/compiler: Move declaration before code.
- r300/compiler: Move declaration before code.
- r300/compiler: Move declaration before code.
- r300/compiler: Move declaration before code.
- r600g: Update SConscript.
- r300/compiler: Move declaration before code.
- r600g: Update SConscript.
- r300/compiler: Move declaration before code.
- r600g: Update SConscript.
- r300/compiler: Move declaration before code.
- r600g: Fix SCons build.
- r300/compiler: Move declaration before code.
- r300/compiler: Move declaration before code.
- r300/compiler: Move declaration before code.
- r300/compiler: Remove declaration before code.
- r300/compiler: Move declaration before code.
- r300/compiler: Move declaration before code.
- r300/compiler: Move declaration before code.
- r300/compiler: Move declaration before code.
- glsl: Remove unnecessary header.
- savage: Remove unnecessary header.
- r600g: Remove unused variable.
- r600g: Remove unnecessary headers.
- r600g: Fix SCons build.
- r600g: Remove unnecessary header.
- gallivm: Remove unnecessary header.
- r600g: Silence uninitialized variable warning.
- r600g: Silence uninitialized variable warning.

- r600g: Silence uninitialized variable warning.
- i915: Silence unused variable warning in non-debug builds.
- i915: Silence unused variable warning in non-debug builds.
- i965: Silence unused variable warning on non-debug builds.
- i965: Silence unused variable warning on non-debug builds.
- i965: Initialize member variables.
- r300: Silence uninitialized variable warning.
- tdfx: Silence unused variable warning on non-debug builds.
- gallivm: Remove unnecessary header.
- glsl: Initialize variable in ir_derefence_array::constant_expression_value
- mesa: Add missing header to shaderobj.h.
- llvmpipe: Return non-zero exit code for lp_test_round failures.
- r300/compiler: Remove unused variable.
- st/xorg: Fix memory leak on error path.
- llvmpipe: Initialize state variable in debug_bin function.
- llvmpipe: Initialize variable.
- draw: Move loop variable declaration outside for loop.
- r600g: Ensure r600_src is initialized in tgsi_exp function.
- glsl: Add assert for unhandled ir_shader case.
- swrast: Print out format on unexpected failure in _swrast_DrawPixels.
- llvmpipe: Remove unnecessary header.
- draw: Remove unnecessary header.
- gallivm: Silence uninitialized variable warnings.
- gallivm: Silence uninitialized variable warnings.
- gallivm: Silence uninitialized variable warning.
- r300g: Silence uninitialized variable warning.
- mesa: Remove unnecessary headers.
- r600g: Silence uninitialized variable warnings.
- st/mesa: Remove unnecessary header.
- mesa: Remove unnecessary header.
- egl: Remove unnecessary headers.
- swrast: Print out format on unexpected failure in _swrast_ReadPixels.
- st/mesa: Silence uninitialized variable warning.
- savage: Remove unnecessary header.
- st/vega: Remove unnecessary headers.
- dri/nouveau: Silence uninitialized variable warning.

- r300/compiler: Move declaration before code.
- r300/compiler: Move declaration before code.
- i965: Silence uninitialized variable warning.
- i965: Silence uninitialized variable warning.
- mesa: Clean up header file inclusion in accum.h.
- mesa: Clean up header file inclusion in version.h.
- mesa: Clean up header file inclusion in api_loopback.h.
- mesa: Clean up header file inclusion in api_validate.h.
- mesa: Include mfeatures.h in api_loopback for FEATURE_beginend.
- mesa: Include mfeatures.h in api_validate.c for FEATURE_* symbols.
- mesa: Clean up header file inclusion in arrayobj.h.
- mesa: Clean up header file inclusion in atifragshader.h.
- mesa: Clean up header file inclusion in attrib.h.
- mesa: Clean up header file inclusion in blend.h.
- mesa: Clean up header file inclusion in buffers.h.
- mesa: Clean up header file inclusion in colortab.h.
- mesa: Clean up header file inclusion in convolve.h.
- mesa: Clean up header file inclusion in debug.h.
- mesa: Clean up header file inclusion in depth.h.
- mesa: Clean up header file inclusion in depthstencil.h.
- mesa: Clean up header file inclusion in drawpix.h.
- mesa: Clean up header file inclusion in drawtex.h.
- mesa: Clean up header file inclusion in enable.h.
- mesa: Clean up header file inclusion in extensions.h.
- graw: Add struct pipe_surface forward declaration.
- mesa: Clean up header file inclusion in fbobject.h.
- mesa: Clean up header file inclusion in ffvertex_prog.h.
- mesa: Clean up header file inclusion in fog.h.
- mesa: Clean up header file inclusion in framebuffer.h.
- mesa: Clean up header file inclusion in hint.h.
- mesa: Clean up header file inclusion in histogram.h.
- mesa: Clean up header file inclusion in image.h.
- mesa: Add missing header and forward declarations in dd.h.
- mesa: Clean up header file inclusion in light.h.
- mesa: Clean up header file inclusion in lines.h.
- mesa: Clean up header file inclusion in matrix.h.

- mesa: Clean up header file inclusion in multisample.h.
- mesa: Clean up header file inclusion in nvprogram.h.
- winsys/xlib: Add cygwin to SConscript.
- mesa: Clean up header file inclusion in pixel.h.
- mesa: Clean up header file inclusion in pixelstore.h.
- mesa: Fix printf format warnings.
- mesa: Clean up header file inclusion in points.h.
- i965: Silence uninitialized variable warning.
- glsl: Add ir_constant_expression.cpp to SConscript.
- mesa: Add definitions for inverse hyperbolic function on MSVC.
- glsl: Fix 'control reaches end of non-void function' warning.
- glsl: Add lower_vector.cpp to SConscript.
- glsl: Fix type of label 'default' in switch statement.
- st/mesa: Remove unnecessary headers.
- swrast: Remove unnecessary header.
- r600: Remove unnecesary header.
- intel: Remove unnecessary header.
- mesa: Clean up header file inclusion in polygon.h.
- mesa: Clean up header file inclusion in rastpos.h.
- mesa: Clean up header file inclusion in readpix.h.
- mesa: Clean up header file inclusion in renderbuffer.h.
- mesa: Clean up header file inclusion in scissor.h.
- mesa: Clean up header file inclusion in shaderapi.h.
- mesa: Clean up header file inclusion in shared.h.
- mesa: Clean up header file inclusion in stencil.h.
- r600: Remove unnecessary header.
- llvmpipe: Remove unnecessary headers.
- mesa: Clean up header file inclusion in syncobj.h.
- r300/compiler: Move declaration before code.
- r300/compiler: Move declaration before code.
- mesa: Clean up header file inclusion in texcompress.h.
- st/vega: Silence uninitialized variable warning.
- mesa: Clean up header file inclusion in texcompress_s3tc.h.
- mesa: Clean up header file inclusion in texenvprogram.h.
- mesa: Clean up header file inclusion in texformat.h.
- mesa: Clean up header file inclusion in texgetimage.h.

- mesa: Clean up header file inclusion in texobj.h.
- gallium/noop: Add prototype for noop_init_state_functions.
- mesa: Clean up header file inclusion in texrender.h.
- mesa: Clean up header file inclusion in transformfeedback.h.
- mesa: Clean up header file inclusion in varray.h.
- mesa: Clean up header file inclusion in viewport.h.
- r200: Silence uninitialized variable warning.
- r600g: Fix SCons build.
- i965: Silence uninitialized variable warning.

Xavier Chantry (8):

- nv50: fix size of outputs_written array
- nv50: apply layout_mask to tile_flags
- nvfx: only expose one rt on nv30
- nvfx: fb->nr_cbufs <= 1 on nv30
- nvfx: reset nvfx->hw_zeta
- nvfx: fixes after array textures merge
- init ps->context with util_surfaces_get and do_get
- gallium/trace: check bind_vertex_sampler_states and set_vertex_sampler_views

Xiang, Haihao (10):

- mesa: fix regression from b4bb6680200b5a898583392f4c831c02f41e63f7
- i965: add support for polygon mode on Sandybridge.
- i965: fix for flat shading on Sandybridge
- i965: set minimum/maximum Point Width on Sandybridge
- meta: allow nested meta operations
- i965: support for two-sided lighting on Sandybridge
- i965: fix register region description
- i965: use align1 access mode for instructions with execSize=1 in VS
- i965: don't spawn GS thread for LINELOOP on Sandybridge
- i965: use BLT to clear buffer if possible on Sandybridge

Zack Rusin (8):

- rbug: fix rbug when contexts are being destroyed
- llvmpipe: fix rasterization of vertical lines on pixel boundaries
- scons: build the xorg state trackers only when env includes drm
- gallivm: implement indirect addressing of the output registers
- gallivm: implement indirect addressing over inputs
- gallivm: fix storing of the addr register

- scons: add alias for identity
- gallium/util: add states relevant to geometry shaders

Zhenyu Wang (40):

- i965: disasm quarter and write enable instruction control on sandybridge
- i965: new state dump for sandybridge
- i965: enable accumulator update in PS kernel too on sandybridge
- i965: Fix color interpolation on sandybridge
- i965: force zero in clipper to ignore RTAIndex on sandybridge
- i965: fix point size setting in header on sandybridge
- i965: ff sync message change for sandybridge
- i965: ignore quads for GS kernel on sandybridge
- i965: add sandybridge viewport state bo into validation list
- i965: VS use SPF mode on sandybridge for now
- i965: fix jump count on sandybridge
- i965: Fix sampler on sandybridge
- i965: fix const register count for sandybridge
- i965: Add all device ids for sandybridge
- i965: sandybridge pipe control workaround before write cache flush
- i965: only allow SIMD8 kernel on sandybridge now
- i965: don't do calculation for delta_xy on sandybridge
- i965: fix pixel w interpolation on sandybridge
- i965: enable polygon offset on sandybridge
- i965: fix scissor state on sandybridge
- i965: fix point sprite on sandybridge
- i965: fix occlusion query on sandybridge
- i965: fallback bitmap operation on sandybridge
- i965: Always set tiling for depth buffer on sandybridge
- i965: fallback lineloop on sandybridge for now
- Revert "i965: Always set tiling for depth buffer on sandybridge"
- i965: always set tiling for fbo depth buffer on sandybridge
- i965: Fix GS hang on Sandybridge
- Revert "i965: fallback lineloop on sandybridge for now"
- i965: refresh wm push constant also for BRW_NEW_FRAMENT_PROGRAM on gen6
- i965: fix dest type of 'endif' on sandybridge
- Revert "i965: VS use SPF mode on sandybridge for now"
- i965: also using align1 mode for math2 on sandybridge

- i965: Fix GS state uploading on Sandybridge
- i965: upload WM state for _NEW_POLYGON on sandybridge
- i965: Use MI_FLUSH_DW for blt ring flush on sandybridge
- i965: explicit tell header present for fb write on sandybridge
- i965: Fix occlusion query on sandybridge
- i965: Use last vertex convention for quad provoking vertex on sandybridge
- i965: Fix provoking vertex select in clip state for sandybridge

Zou Nan hai (1):

- i965: skip too small size mipmap

delphi (2):

- draw: added userclip planes and updated variant_key
- draw: some changes to allow for runtime changes to userclip planes

nobled (3):

- r300g: Abort if atom allocations fail
- r300g: Abort if draw_create() fails
- r300g: Drop unnecessary cast

pontus lidman (1):

- mesa: check for posix_memalign() errors

richard (2):

- evergreen : fix z format setting, enable stencil.
- r600c : inline vertex format is not updated in an app, switch to use vfetch constants. For the 7.9 and 7.10 branches as well.

4.248 Mesa 7.9.2 Release Notes / March 2, 2011

Mesa 7.9.2 is a bug fix release which fixes bugs found since the 7.9.1 release.

Mesa 7.9.2 implements the OpenGL 2.1 API, but the version reported by glGetString(GL_VERSION) depends on the particular driver being used. Some drivers don't support all the features required in OpenGL 2.1.

See the *Compiling/Installing page* for prerequisites for DRI hardware acceleration.

4.248.1 MD5 checksums

```
eb4ab8c1a03386def3ea34b1358e9cda  MesaLib-7.9.2.tar.gz
8f6d1474912787ce13bd35f3bae9938a  MesaLib-7.9.2.tar.bz2
427a81dd43ac97603768dc5c6af3df26  MesaLib-7.9.2.zip
aacb8f4db997e346db40c6066942140a  MesaGLUT-7.9.2.tar.gz
18abe6cff4fad8ad4752c7b7ab548e5d  MesaGLUT-7.9.2.tar.bz2
3189e5732d636c71baf3d8bc23ce7b11  MesaGLUT-7.9.2.zip
```

4.248.2 New features

None.

4.248.3 Bug fixes

This list is likely incomplete.

- Fix an off-by-one bug in a vsplit assertion.
- Fix incorrect handling of `layout` qualifier with `in`, `out`, `attribute`, and `varying`.
- Fix an i965 GPU hang in GLSL shaders that contain an unconditional `discard` statement.
- Fix an i965 shader bug where the negative absolute value was generated instead of the absolute value of a negation.
- Fix numerous issues handling precision qualifiers in GLSL ES.
- Fixed a few GLX protocol encoder bugs (Julien Cristau)
- Assorted Gallium llvmpipe driver bug fixes
- Assorted Mesa/Gallium state tracker bug fixes
- Bug 26795 - gl_FragCoord off by one in Gallium drivers.
- Bug 29164 - [GLSL 1.20] invariant variable shouldn't be used before declaration
- Bug 29823 - GetUniform[if]v busted
- Bug 29927 - [glsl2] fail to compile shader with constructor for array of struct type
- Bug 30156 - [i965] After updating to Mesa 7.9, Civilization IV starts to show garbage
- Bug 31923 - [GLSL 1.20] allowing inconsistent centroid declaration between two vertex shaders
- Bug 31925 - [GLSL 1.20] "#pragma STDGL invariant(all)" fail
- Bug 32214 - [gles2]no link error happens when missing vertex shader or frag shader
- Bug 32375 - [gl gles2] Not able to get the attribute by function glGetVertexAttribfv
- Bug 32541 - Segmentation Fault while running an HDR (high dynamic range) rendering demo
- Bug 32569 - [gles2] glGetShaderPrecisionFormat not implemented yet
- Bug 32695 - [glsl] SIGSEGV glcpp/glcpp-parse.y:833
- Bug 32831 - [glsl] division by zero crashes GLSL compiler
- Bug 32910 - Keywords 'in' and 'out' not handled properly for GLSL 1.20 shaders
- Bug 33219 -[GLSL bisected] implicit sized array triggers segfault in ir_to_mesa_visitor::copy_propagate
- Bug 33306 - GLSL integer division by zero crashes GLSL compiler
- Bug 33308 -[glsl] ast_to_hir.cpp:3016: virtual ir_rvalue* ast_jump_statement::hir(exec_list*, _mesa_glsl_parse_state*): Assertion 'ret != __null' failed.
- Bug 33316 - uniform array will be allocate one line more and initialize it when it was freed will abort
- Bug 33386 - Dubious assembler in read_rgba_span_x86.S
- Bug 33388 - Dubious assembler in xform4.S
- Bug 33433 - Error in x86-64 API dispatch code.

- Bug 33507 - [glsl] GLSL preprocessor modulus by zero crash
- Bug 33508 - [glsl] GLSL compiler modulus by zero crash
- Bug 33916 - Compiler accepts reserved operators % and %=
- Bug 34047 - Assert in _tnl_import_array() when using GLfixed vertex datatypes with GLESv2
- Bug 34114 - Sun Studio build fails due to standard library functions not being in global namespace
- Bug 34198 - [GLSL] implicit sized array with index 0 used gets assertion
- Ubuntu bug 691653 - compiz crashes when using alt-tab (the radeon driver kills it)
- Meego bug 13005 - Graphics GLSL issue lead to camera preview fail on Pinetrail

4.248.4 Changes

The full set of changes can be viewed by using the following GIT command:

```
git log mesa-7.9.1..mesa-7.9.2
```

Alberto Milone (1):

- r600c: add evergreen ARL support.

Brian Paul (19):

- draw: Fix an off-by-one bug in a vsplit assertion.
- mesa: fix a few format table mistakes, assertions
- mesa: fix num_draw_buffers==0 in fixed-function fragment program generation
- mesa: don't assert in GetIntegerIndexed, etc
- mesa: check for dummy renderbuffer in _mesa_FramebufferRenderbufferEXT()
- llvmpipe: make sure binning is active when we begin/end a query
- st/mesa: fix incorrect fragcoord.x translation
- softpipe: fix off-by-one error in setup_fragcoord_coeff()
- cso: fix loop bound in cso_set_vertex_samplers()
- st/mesa: set renderbuffer _BaseFormat in a few places
- st/mesa: fix the default case in st_format_datatype()
- st/mesa: need to translate clear color according to surface's base format
- docs: update 7.9.2 release notes with Brian's cherry-picks
- docs: add links to 7.9.1 and 7.9.2 release notes
- mesa: include compiler.h for ASSERT macro
- glsl: add ir_shader case in switch stmt to silence warning
- glsl2: fix signed/unsigned comparison warning
- mesa: implement glGetShaderPrecisionFormat()
- docs: updated environment variable list

Bryce Harrington (1):

- r300g: Null pointer check for buffer deref in gallium winsys

Chad Versace (14):

- glsl: At link-time, check that globals have matching centroid qualifiers
- glcpp: Fix segfault when validating macro redefinitions
- glsl: Fix parser rule for type_specifier
- glsl: Change default value of ast_type_specifier::precision
- glsl: Add semantic checks for precision qualifiers
- glsl: Add support for default precision statements
- glsl: Remove redundant semantic check in parser
- glsl: Fix semantic checks on precision qualifiers
- glsl: Fix segfault due to missing printf argument
- glsl: Mark 'in' variables at global scope as read-only
- glcpp: Raise error when modulus is zero
- glsl: Set operators '%' and '%=' to be reserved when GLSL < 1.30
- glsl: Reinstate constant-folding for division by zero
- tnl: Add support for datatype GL_FIXED in vertex arrays

Chia-I Wu (1):

- mesa: Add glDepthRangef and glClearDepthf to APIspec.xml.

Chris Wilson (1):

- intel: Check for unsupported texture when finishing using as a render target

Cyril Brulebois (1):

- Point to bugs.freedesktop.org rather than bugzilla.freedesktop.org

Dave Airlie (2):

- radeon/r200: fix fbo-clearmipmap + gen-teximage
- radeon: avoid segfault on 3D textures.

Dimitry Andric (4):

- mesa: s/movzx/movzbl/
- mesa: s/movzxw/movzwl/ in read_rgba_span_x86.S
- glapi: adding @ char before type specifier in glapi_x86.S
- glapi: add @GOTPCREL relocation type

Eric Anholt (11):

- i965: Avoid double-negation of immediate values in the VS.
- docs: Add a relnote for the Civ IV on i965.
- i965/vs: When MOVing to produce ABS, strip negate of the operand.
- glsl: Fix the lowering of variable array indexing to not lose write_masks.
- intel: Make renderbuffer tiling choice match texture tiling choice.

- glapi: Add entrypoints and enums for GL_ARB_ES2_compatibility.
- mesa: Add extension enable bit for GL_ARB_ES2_compatibility.
- mesa: Add actual support for glReleaseShaderCompiler from ES2.
- mesa: Add support for glDepthRangef and glClearDepthf.
- mesa: Add getters for ARB_ES2_compatibility MAX_*_VECTORS.
- mesa: Add getter for GL_SHADER_COMPILER with ARB_ES2_compatibility.

Ian Romanick (42):

- docs: Add 7.9.1 md5sums
- glsl: Support the 'invariant(all)' pragma
- glcpp: Generate an error for division by zero
- glsl: Add version_string containing properly formatted GLSL version
- glsl & glcpp: Refresh autogenerated lexer and parser files.
- glsl: Disallow 'in' and 'out' on globals in GLSL 1.20
- glsl: Track variable usage, use that to enforce semantics
- glsl: Allow 'in' and 'out' when 'layout' is also available
- docs: Initial set of release notes for 7.9.2
- mesa: bump version to 7.9.2-devel
- docs: Update 7.9.2 release notes
- i965: Make OPCODE_KIL_NV do its work in a temp, not the null reg!
- glsl: Refresh autogenerated lexer and parser files.
- glsl: Don't assert when the value returned by a function has no rvalue
- linker: Set sizes for non-global arrays as well
- linker: Propagate max_array_access while linking functions
- docs: Update 7.9.2 release notes
- Use C-style system headers in C++ code to avoid issues with std:: namespace
- mesa: glGetUniform only returns a single element of an array
- linker: Generate link errors when ES shaders are missing stages
- mesa: Fix error checks in GetVertexAttrib functions
- docs: Update 7.9.2 release notes
- mesa: Remove unsupported OES extensions
- glapi: Regenerate for GL_ARB_ES2_compatibility.
- mesa: Connect glGetShaderPrecisionFormat into the dispatch table
- i965: Set correct values for range/precision of fragment shader types
- i915: Set correct values for range/precision of fragment shader types
- intel: Fix typeos from 3d028024 and 790ff232
- glsl: Ensure that all GLSL versions are supported in the stand-alone compiler

- glsl: Reject shader versions not supported by the implementation
- mesa: Initial size for secondary color array is 3
- glcpp: Regenerate files from recent cherry picks
- glsl: Finish out the reduce/reduce error fixes
- glsl: Regenerate compiler files from cherry picks
- linker: Fix off-by-one error implicit array sizing
- i915: Only mark a register as available if all components are written
- i915: Calculate partial result to temp register first
- i915: Force lowering of all types of indirect array accesses in the FS
- docs: Update 7.9.2 release notes for recent cherry picks
- docs: Clean up bug fixes list
- intel: Remove driver date and related bits from renderer string
- mesa: set version string to 7.9.2 (final)

Jian Zhao (1):

- mesa: fix an error in uniform arrays in row calculating.

Julien Cristau (3):

- glx: fix request lengths
- glx: fix GLXChangeDrawableAttributesSGIX request
- glx: fix length of GLXGetFBConfigsSGIX

Keith Packard (1):

- glsl: Eliminate reduce/reduce conflicts in glsl grammar

Kenneth Graunke (12):

- glsl: Expose a public glsl_type::void_type const pointer.
- glsl: Don't bother unsetting a destructor that was never set.
- glsl, i965: Remove unnecessary talloc includes.
- glcpp: Remove use of talloc reference counting.
- ralloc: Add a fake implementation of ralloc based on talloc.
- Convert everything from the talloc API to the ralloc API.
- ralloc: a new MIT-licensed recursive memory allocator.
- Remove talloc from the make and automake build systems.
- Remove talloc from the SCons build system.
- Remove the talloc sources from the Mesa repository.
- glsl: Fix use of uninitialized values in _mesa_glsl_parse_state ctor.
- glsl: Use reralloc instead of plain realloc.

Marek Olšák (3):

- docs: fix messed up names with special characters in relnotes-7.9.1

- mesa: fix texture3D mipmap generation for UNSIGNED_BYTE_3_3_2
- st/dri: Track drawable context bindings

Paulo Zanoni (1):

- dri_util: fail driCreateNewScreen if InitScreen is NULL

Sam Hocevar (2):

- docs: add glsl info
- docs: fix glsl_compiler name

Vinson Lee (1):

- ralloc: Add missing va_end following va_copy.

nobled (1):

- glx: Put null check before use

4.249 Mesa 7.9.1 Release Notes / January 7, 2011

Mesa 7.9.1 is a bug fix release which fixes bugs found since the 7.9 release.

Mesa 7.9.1 implements the OpenGL 2.1 API, but the version reported by glGetString(GL_VERSION) depends on the particular driver being used. Some drivers don't support all the features required in OpenGL 2.1.

See the *Compiling/Installing page* for prerequisites for DRI hardware acceleration.

4.249.1 MD5 checksums

```
78422843ea875ad4eac35b9b8584032b  MesaLib-7.9.1.tar.gz
07dc6cfb5928840b8b9df5bd1b3ae434  MesaLib-7.9.1.tar.bz2
c8eaea5b3c3d6dee784bd8c2db91c80f  MesaLib-7.9.1.zip
ee9ecae4ca56fbb2d14dc15e3a0a7640  MesaGLUT-7.9.1.tar.gz
41fc477d524e7dc5c84da8ef22422bea  MesaGLUT-7.9.1.tar.bz2
90b287229afdf19317aa989d19462e7a  MesaGLUT-7.9.1.zip
```

4.249.2 New features

None.

4.249.3 Bug fixes

This list is likely incomplete.

- Bug 28800 - [r300c, r300g] Texture corruption with World of Warcraft
- Bug 29420 - Amnesia / HPL2 RendererFeatTest - not rendering correctly
- Bug 29946 - [swrast] piglit valgrind glsl-array-bounds-04 fails
- Bug 30261 - [GLSL 1.20] allowing inconsistent invariant declaration between two vertex shaders
- Bug 30632 - [softpipe] state_tracker/st_manager.c:489: st_context_notify_invalid_framebuffer: Assertion 'stfb && stfb->iface == stfbi' failed.

- Bug 30694 - wincopy will crash on Gallium drivers when going to front buffer
- Bug 30787 - Invalid asm shader does not generate draw-time error when used with GLSL shader
- Bug 30993 - getFramebufferAttachmentParameteriv wrongly generates error
- Bug 31101 - [glsl2] abort() in ir_validate::visit_enter(ir_assignment *ir)
- Bug 31193 - [regression] aa43176e break water reflections
- Bug 31194 - The mesa meta save/restore code doesn't ref the current GLSL program
- Bug 31371 - glslparsertest: ir.cpp:358: ir_constant::ir_constant(const glsl_type*, const ir_constant_data*): Assertion '(type->base_type >= 0) && (type->base_type <= 3)' failed.
- Bug 31439 - Crash in glBufferSubData() with size == 0
- Bug 31495 - [i965 gles2c bisected] OpenGL ES 2.0 conformance GL2Tests_GetBIFD_input.run regressed
- Bug 31514 - isBuffer returns true for unbound buffers
- Bug 31560 - [tdfx] tdfx_tex.c:702: error: 'const struct gl_color_table' has no member named 'Format'
- Bug 31617 - Radeon/Compiz: 'failed to attach dri2 front buffer', error case not handled
- Bug 31648 - [GLSL] array-struct-array gets assertion: '(size >= 1) && (size <= 4)' failed.
- Bug 31650 - [GLSL] varying gl_TexCoord fails to be re-declared to different size in the second shader
- Bug 31673 - GL_FRAGMENT_PRECISION_HIGH preprocessor macro undefined in GLSL ES
- Bug 31690 - i915 shader compiler fails to flatten if in Aquarium webgl demo.
- Bug 31832 - [i915] Bad renderbuffer format: 21
- Bug 31841 - [drm:radeon_cs_ioctl] *ERROR* Invalid command stream !
- Bug 31894 - Writing to gl_PointSize with GLES2 corrupts other varyings
- Bug 31909 - [i965] brw_fs.cpp:1461: void fs_visitor::emit_bool_to_cond_code(ir_rvalue*): Assertion 'expr->operands[i]->type->is_scalar()' failed.
- Bug 31934 - [gallium] Mapping empty buffer object causes SIGSEGV
- Bug 31983 - [i915 gles2] "if (expression with builtin/varying variables) discard" breaks linkage
- Bug 31985 - [GLSL 1.20] initialized uniform array considered as "unsized"
- Bug 31987 - [gles2] if input a wrong pname(GL_NONE) to glGetBoolean, it will not case GL_INVALID_ENUM
- Bug 32035 - [GLSL bisected] comparing unsized array gets segfault
- Bug 32070 - llvmpipe renders stencil demo incorrectly
- Bug 32273 - assertion fails when starting vdrift 2010 release with shaders enabled
- Bug 32287 - [bisected GLSL] float-int failure
- Bug 32311 - [965 bisected] Array look-ups broken on GM45
- Bug 32520 - [gles2] glBlendFunc(GL_ZERO, GL_DST_COLOR) will result in GL_INVALID_ENUM
- Bug 32825 - egl_glx driver completely broken in 7.9 branch [fix in master]

4.249.4 Changes

The full set of changes can be viewed by using the following GIT command:

```
git log mesa-7.9..mesa-7.9.1
```

Alex Deucher (5):

- r100: revalidate after radeon_update_renderbuffers
- r600c: add missing radeon_prepare_render() call on evergreen
- r600c: properly align mipmaps to group size
- gallium/egl: fix r300 vs r600 loading
- r600c: fix some opcodes on evergreen

Aras Pranckevicius (2):

- glsl: fix crash in loop analysis when some controls can't be determined
- glsl: fix matrix type check in ir_algebraic

Brian Paul (27):

- swrast: fix choose_depth_texture_level() to respect mipmap filtering state
- st/mesa: replace assertion w/ conditional in framebuffer invalidation
- egl/i965: include inline_wrapper_sw_helper.h
- mesa: Add missing else in do_row_3D
- mesa: add missing formats in _mesa_format_to_type_and_comps()
- mesa: handle more pixel types in mipmap generation code
- mesa: make glIsBuffer() return false for never bound buffers
- mesa: fix glDeleteBuffers() regression
- swrast: init alpha value to 1.0 in opt_sample_rgb_2d()
- meta: Mask Stencil.Clear against stencilMax in _mesa_meta_Clear
- st/mesa: fix mapping of zero-sized buffer objects
- mesa: check for posix_memalign() errors
- llvmpipe: fix broken stencil writemask
- mesa: fix GL_FRAMEBUFFER_ATTACHMENT_OBJECT_NAME query
- mesa: return GL_FRAMEBUFFER_DEFAULT as FBO attachment type
- mesa: make glGet*(GL_NONE) generate GL_INVALID_ENUM
- mesa: test for cube map completeness in glGenerateMipmap()
- tnl: Initialize gl_program_machine memory in run_vp.
- tnl: a better way to initialize the gl_program_machine memory
- mesa, st/mesa: disable GL_ARB_geometry_shader4
- glsl: fix off by one in register index assertion
- st/mesa: fix mipmap generation bug

- glsl: new glsl_strtod() wrapper to fix decimal point interpretation
- mesa: no-op glBufferSubData() on size==0
- tdfx: s/Format/_BaseFormat/
- st/mesa: fix renderbuffer pointer check in st_Clear()
- mesa: Bump the number of bits in the register index.

Chad Versace (5):

- glsl: Fix lexer rule for ^=
- glsl: Fix ast-to-hir for ARB_fragment_coord_conventions
- glsl: Fix ir_expression::constant_expression_value()
- glsl: Fix erroneous cast in ast_jump_statement::hir()
- glsl: Fix linker bug in cross_validate_globals()

Chia-I Wu (10):

- targets/egl: Fix linking with libdrm.
- st/vega: Fix version check in context creation.
- st/egl: Do not finish a fence that is NULL.
- egl: Fix a false negative check in _eglCheckMakeCurrent.
- st/mesa: Unreference the sampler view in st_bind_surface.
- egl_dri2: Fix __DRI_DRI2 version 1 support.
- st/vega: Do not wait NULL fences.
- mesa: Do not advertise GL_OES_texture_3D.
- egl_glx: Fix borken driver.
- egl: Check extensions.

Daniel Lichtenberger (1):

- radeon: fix potential segfault in renderbuffer update

Daniel Vetter (1):

- r200: revalidate after radeon_update_renderbuffers

Dave Airlie (1):

- r300g: fixup rs690 tiling stride alignment calculations.

Eric Anholt (13):

- intel: Allow CopyTexSubImage to InternalFormat 3/4 textures, like RGB/RGBA.
- glsl: Free the loop state context when we free the loop state.
- i965: Allow OPCODE_SWZ to put immediates in the first arg.
- i965: Add support for rendering to SARGB8 FBOs.
- glsl: Add a helper constructor for expressions that works out result type.
- glsl: Fix structure and array comparisions.
- glsl: Quiet unreachable no-return-from-function warning.

- glsl: Mark the array access for whole-array comparisons.
- glsl: Fix flipped return of has_value() for array constants.
- mesa: Add getters for the rest of the supported draw buffers.
- mesa: Add getters for ARB_copy_buffer's attachment points.
- i965: Correct the dp_read message descriptor setup on g4x.
- glsl: Correct the marking of InputsRead/OutputsWritten on in/out matrices.

Fabian Bieler (1):

- glsl: fix lowering conditional returns in subroutines

Francisco Jerez (3):

- meta: Don't leak alpha function/reference value changes.
- meta: Fix incorrect rendering of the bitmap alpha component.
- meta: Don't try to disable cube maps if the driver doesn't expose the extension.

Henri Verbeet (2):

- r600: Evergreen has two extra frac_bits for the sampler LOD state.
- st/mesa: Handle wrapped depth buffers in st_copy_texsubimage().

Ian Romanick (33):

- Add 7.9 md5sums
- docs: Import 7.8.x release notes from 7.8 branch.
- docs: download.html does not need to be updated for each release
- docs: Update mailing lines from sf.net to freedesktop.org
- docs: added news item for 7.9 release
- mesa: Validate assembly shaders when GLSL shaders are used
- linker: Reject shaders that have unresolved function calls
- mesa: Refactor validation of shader targets
- glsl: Slightly change the semantic of _LinkedShaders
- linker: Improve handling of unread/unwritten shader inputs/outputs
- glsl: Commit lexer files changed by previous cherry picking
- mesa: Make metaops use program refcounts instead of names.
- glsl: Fix incorrect gl_type of sampler2DArray and sampler1DArrayShadow
- mesa: Allow query of MAX_SAMPLES with EXT_framebuffer_multisample
- glsl: better handling of linker failures
- mesa: Fix glGet of ES2's GL_MAX_*_VECTORS properties.
- i915: Disallow alpha, red, RG, and sRGB as render targets
- glsl/linker: Free any IR discarded by optimization passes.
- glsl: Add an optimization pass to simplify discards.
- glsl: Add a lowering pass to move discards out of if-statements.

- i915: Correctly generate unconditional KIL instructions
- glsl: Add unary ir_expression constructor
- glsl: Ensure that equality comparisons don't return a NULL IR tree
- glcpp: Commit changes in generated files cause by previous commit
- glsl: Inherrit type of declared variable from initializer
- glsl: Inherrit type of declared variable from initializer after processing assignment
- linker: Ensure that unsized arrays have a size after linking
- linker: Fix regressions caused by previous commit
- linker: Allow built-in arrays to have different sizes between shader stages
- ir_to_mesa: Don't generate swizzles for record derefs of non-scalar/vectors
- Refresh autogenerated file builtin_function.cpp.
- docs: Initial set of release notes for 7.9.1
- mesa: set version string to 7.9.1

Julien Cristau (1):

- Makefile: don't include the same files twice in the tarball

Kenneth Graunke (19):

- glcpp: Return NEWLINE token for newlines inside multi-line comments.
- generate_builtins.py: Output large strings as arrays of characters.
- glsl: Fix constant component count in vector constructor emitting.
- ir_dead_functions: Actually free dead functions and signatures.
- glcpp: Define GL_FRAGMENT_PRECISION_HIGH if GLSL version >= 1.30.
- glsl: Unconditionally define GL_FRAGMENT_PRECISION_HIGH in ES2 shaders.
- glsl: Fix constant expression handling for <, >, <=, >= on vectors.
- glsl: Use do_common_optimization in the standalone compiler.
- glsl: Don't inline function prototypes.
- glsl: Add a virtual as_discard() method.
- glsl: Remove "discard" support from lower_jumps.
- glsl: Refactor get_num_operands.
- glcpp: Don't emit SPACE tokens in conditional_tokens production.
- glsl: Clean up code by adding a new is_break() function.
- glsl: Consider the "else" branch when looking for loop breaks.
- Remove OES_compressed_paletted_texture from the ES2 extension list.
- glsl/builtins: Compute the correct value for smoothstep(vec, vec, vec).
- Fix build on systems where "python" is python 3.
- i965: Internally enable GL_NV_blend_square on ES2.

Kristian Høgsberg (1):

- i965: Don't write mrf assignment for pointsize output

Luca Barbieri (1):

- glsl: Unroll loops with conditional breaks anywhere (not just the end)

Marek Olšák (17):

- r300g: fix microtiling for 16-bits-per-channel formats
- r300g: fix texture border for 16-bits-per-channel formats
- r300g: add a default channel ordering of texture border for unhandled formats
- r300g: fix texture border color for all texture formats
- r300g: fix rendering with no vertex elements
- r300/compiler: fix rc_rewrite_depth_out for it to work with any instruction
- r300g: fix texture border color once again
- r300g: fix texture swizzling with compressed textures on r400-r500
- r300g: disable ARB_texture_swizzle if S3TC is enabled on r3xx-only
- mesa, st/mesa: fix gl_FragCoord with FBOs in Gallium
- st/mesa: initialize key in st_vp_varient
- r300/compiler: fix swizzle lowering with a presubtract source operand
- r300g: fix rendering with a vertex attrib having a zero stride
- ir_to_mesa: Add support for conditional discards.
- r300g: finally fix the texture corruption on r3xx-r4xx
- mesa: fix texel store functions for some float formats
- r300/compiler: disable the rename_regs pass for loops

Mario Kleiner (1):

- mesa/r300classic: Fix dri2Invalidate/radeon_prepare_render for page flipping.

Peter Clifton (1):

- intel: Fix emit_linear_blit to use DWORD aligned width blits

Robert Hooker (2):

- intel: Add a new B43 pci id.
- egl_dri2: Add missing intel chip ids.

Roland Scheidegger (1):

- r200: fix r200 large points

Thomas Hellstrom (17):

- st/xorg: Don't try to use option values before processing options
- xorg/vmwgfx: Make vmwarectrl work also on 64-bit servers
- st/xorg: Add a customizer option to get rid of annoying cursor update flicker
- xorg/vmwgfx: Don't hide HW cursors when updating them
- st/xorg: Don't try to remove invalid fbs

- st/xorg: Fix typo
- st/xorg, xorg/vmwgfx: Be a bit more frendly towards cross-compiling environments
- st/xorg: Fix compilation errors for Xservers compiled without Composite
- st/xorg: Don't use deprecated x*alloc / xfree functions
- xorg/vmwgfx: Don't use deprecated x*alloc / xfree functions
- st/xorg: Fix compilation for Xservers >= 1.10
- mesa: Make sure we have the talloc cflags when using the talloc headers
- egl: Add an include for size_t
- mesa: Add talloc includes for gles
- st/egl: Fix build for include files in nonstandard places
- svga/drm: Optionally resolve calls to powf during link-time
- gallium/targets: Trivial crosscompiling fix

Tom Stellard (7):

- r300/compiler: Make sure presubtract sources use supported swizzles
- r300/compiler: Fix register allocator's handling of loops
- r300/compiler: Fix instruction scheduling within IF blocks
- r300/compiler: Use zero as the register index for unused sources
- r300/compiler: Ignore alpha dest register when replicating the result
- r300/compiler: Use correct swizzles for all presubtract sources
- r300/compiler: Don't allow presubtract sources to be remapped twice

Vinson Lee (1):

- glsl: Fix 'control reaches end of non-void function' warning.

richard (1):

- r600c : inline vertex format is not updated in an app, switch to use vfetch constants. For the 7.9 and 7.10 branches as well.

4.250 Mesa 7.9 Release Notes / October 4, 2010

Mesa 7.9 is a new development release. People who are concerned with stability and reliability should stick with a previous release or wait for Mesa 7.9.1.

Mesa 7.9 implements the OpenGL 2.1 API, but the version reported by glGetString(GL_VERSION) depends on the particular driver being used. Some drivers don't support all the features required in OpenGL 2.1.

See the *Compiling/Installing page* for prerequisites for DRI hardware acceleration.

4.250.1 MD5 checksums

```
ed65ab425b25895c7f473d0a5e6e64f8  MesaLib-7.9.tar.gz
82c740c49d572baa6da2b1a1eee90bca  MesaLib-7.9.tar.bz2
cd2b6ecec759b0457475e94bbb38fedb  MesaLib-7.9.zip
7b54af9fb9b1f6a1a65db2520f50848f  MesaGLUT-7.9.tar.gz
20d07419d1929f833fdb36bced290ad5  MesaGLUT-7.9.tar.bz2
62a7edecd7c92675cd6029b05217eb0a  MesaGLUT-7.9.zip
```

4.250.2 New features

- New, improved GLSL compiler written by Intel. See the *Shading Language* page for more information.
- New, very experimental Gallium driver for R600-R700 Radeons.
- Support for AMD Evergreen-based Radeons (HD 5xxx)
- GL_EXT_timer_query extension (i965 driver and softpipe only)
- GL_EXT_framebuffer_multisample extension (intel drivers, MAX_SAMPLES = 1)
- GL_ARB_texture_swizzle extension (alias of GL_EXT_texture_swizzle)
- GL_ARB_draw_elements_base_vertex, GL_ARB_fragment_program_shadow, GL_ARB_window_pos, GL_EXT_gpu_program_parameters, GL_ATI_texture_env_combine3, GL_MESA_pack_invert, and GL_OES_EGL_image extensions in Gallium drivers
- GL_ARB_depth_clamp and GL_NV_depth_clamp extensions (in nv50 and r600 Gallium drivers)
- GL_ARB_half_float_vertex extension (in nvfx, r300, r600, softpipe, and llvmpipe Gallium drivers)
- GL_EXT_draw_buffers2 (in nv50, r600, softpipe, and llvmpipe Gallium drivers)
- GL_EXT_texture_swizzle (in nvfx, r300, r600, softpipe, and llvmpipe Gallium drivers)
- GL_ATI_texture_mirror_once (in nvfx, nv50, r300, r600, softpipe, and llvmpipe Gallium drivers)
- GL_NV_conditional_render (in r300 Gallium driver)
- Initial "signs of life" support for Sandybridge hardware in i965 DRI driver.

4.250.3 Bug fixes

This list is likely incomplete.

- Massive improvements to the Gallium driver for R300-R500 Radeons; this driver is now considered stable for use as a DRI (OpenGL) driver.
- Bug 10908 - GLSL: gl_FogParamaters gl_Fog built-in uniform not functioning
- Bug 13753 - Numerous bugs in GLSL uniform handling
- Bug 16854 - GLSL function call at global scope causes SEGV
- Bug 16856 - GLSL indexing of unsized array results in assertion failure
- Bug 18659 - Crash in shader/slang/slang_codegen.c _slang_gen_function_call_name()
- Bug 19089 - [GLSL] glsl1/shadow2D() cases fail
- Bug 22622 - [GM965 GLSL] noise*() cause GPU lockup
- Bug 23743 - For loop from 0 to 0 not optimized out
- Bug 24553 - shader compilation times explode when using more () pairs

- Bug 25664 - [GLSL] re-declaring an empty array fails to compile
- Bug 25769 - [GLSL] "float" can be implicitly converted to "int"
- Bug 25808 - [GLSL] const variable is modified successfully
- Bug 25826 - [GLSL] declaring an unsized array then re-declaring with a size fails
- Bug 25827 - [GLSL] vector constructor accepts too many arguments successfully
- Bug 25829 - [GLSL] allowing non-void function without returning value
- Bug 25830 - [GLSL] allowing non-constant-expression as const declaration initializer
- Bug 25877 - [GLSL 1.10] implicit conversion from "int" to "float" should not be allowed
- Bug 25878 - [GLSL] sampler is converted to int successfully
- Bug 25994 - [GM45][GLSL] 'return' statement in vertex shader unsupported
- Bug 25999 - [GLSL] embedded structure constructor fails to compile
- Bug 26000 - [GLSL] allowing different parameter qualifier between the function definition and declaration
- Bug 26001 - [GLSL 1.10] constructing matrix from matrix succeeds
- Bug 26224 - [GLSL] Cannot get location of a uniform struct member
- Bug 26990 - [GLSL] variable declaration in "while" fails to compile
- Bug 27004 - [GLSL] allowing macro redefinition
- Bug 27060 - [965] piglit glsl-fs-raytrace failure due to lack of function calls.
- Bug 27216 - Assignment with a function call in an if statement causes an assertion failure
- Bug 27261 - GLSL Compiler fails on the following vertex shader
- Bug 27265 - GLSL Compiler doesnt link the attached vertex shader
- Bug 27388 - [i965] piglit glsl-vs-arrays failure
- Bug 27403 - GLSL struct causing "Invalid src register file . . . " error
- Bug 27914 - GLSL compiler uses MUL+ADD where it could use MAD
- Bug 28055 - glsl-texcoord-array fails GLSL compilation
- Bug 28374 - SIGSEGV shader/slang/slang_typeinfo.c:534
- Bug 28748 - [i965] uninlined function calls support
- Bug 28833 - piglit/shaders/glsl-texcoord-array fail
- Bug 28834 - Add support for system fpclassify to GL_OES_query_matrix function for OpenBSD / NetBSD
- Bug 28837 - varying vec4 index support
- Bug 28845 - The GLU tesselator code has some warnings
- Bug 28889 - [regression] wine game crash
- Bug 28894 - slang build fails if absolute path contains spaces
- Bug 28913 - [GLSL] allowing two version statements
- Bug 28931 - Floating Point Exception in Warzone2100 Trunk version
- Bug 28966 - [r300g] Dynamic branching 3 demo does not run
- Bug 28967 - slang/slang_emit.c:350: storage_to_src_reg: Assertion 'index >= 0' failed.

- Bug 29013 - [r300g] translate_rgb_op: unknown opcode ILLEGAL OPCODE
- Bug 29020 - [r300g] Wine d3d9 tests hardlock
- Bug 29910 - Mesa advertises bogus GL_ARB_shading_language_120
- Bug 30196 - [GLSL] gl_TextureMatrix{Inverse,Transpose,InverseTranspose} unsupported

4.250.4 Changes

- The Mesa demo/test programs have been moved into a separate git repository.
- GL/glext.h file upgraded to version 64
- GL/glxext.h file upgraded to version 32
- GL/wglext.h file upgraded to version 22

4.251 Mesa 7.8.3 Release Notes / (date tbd)

Mesa 7.8.3 is a bug fix release which fixes bugs found since the 7.8.2 release.

Mesa 7.8.3 implements the OpenGL 2.1 API, but the version reported by glGetString(GL_VERSION) depends on the particular driver being used. Some drivers don't support all the features required in OpenGL 2.1.

See the *Compiling/Installing page* for prerequisites for DRI hardware acceleration.

4.251.1 MD5 checksums

```
x  MesaLib-7.8.3.tar.gz
x  MesaLib-7.8.3.tar.bz2
x  MesaLib-7.8.3.zip
x  MesaDemos-7.8.3.tar.gz
x  MesaDemos-7.8.3.tar.bz2
x  MesaDemos-7.8.3.zip
x  MesaGLUT-7.8.3.tar.gz
x  MesaGLUT-7.8.3.tar.bz2
x  MesaGLUT-7.8.3.zip
```

4.251.2 New features

None.

4.251.3 Changes

- The radeon driver should use less memory when searching for a valid mip image.

4.251.4 Bug fixes

- Fix unsupported FB with D24S8 (bug 29116)
- Fix ReadPixels crash when reading depth/stencil from an FBO

- Fixed a bug rendering to 16-bit buffers using swrast.
- Fixed a state tracker/TGSI bug that caused crashes when using Windows' memory debugging features.
- Fixed an issue rendering to 32-bit channels with swrast (bug 29487)
- GLSL: fix indirect `gl_TextureMatrix` addressing (bug 28967)
- GLSL: fix for bug 27216
- GLSL: fix zw fragcoord entries in some cases (bug 29183)
- Fix texture env generation in some cases (bug 28169)
- osmesa: a fix for calling `OSMesaMakeCurrent` twice was applied (bug 10966
- A bug was fixed which could cause Mesa to ignore the `MESA_EXTENSION_OVERRIDE` environment variable.
- A bug related to specular highlights on backfaces was fixed.
- A radeon-specific issue with `glCopyTex(Sub)Image` was corrected.
- radeon/wine: flush command stream in more cases, fixing wine d3d9 tests.
- r600: fix sin+cos normalization.
- r600: (properly) ignore `GL_COORD_REPLACE` when point sprites are disabled.
- radeon: avoid flushing when the context is not current.
- r300c: a bug affecting unaligned BOs was fixed.
- r300c: a hardlock caused by ARB_half_float_vertex incorrectly advertised on some chipsets.

4.252 Mesa 7.8.2 Release Notes / June 17, 2010

Mesa 7.8.2 is a bug fix release which fixes bugs found since the 7.8.1 release.

Mesa 7.8.2 implements the OpenGL 2.1 API, but the version reported by glGetString(GL_VERSION) depends on the particular driver being used. Some drivers don't support all the features required in OpenGL 2.1.

See the *Compiling/Installing page* for prerequisites for DRI hardware acceleration.

4.252.1 MD5 checksums

```
c89b63d253605ed40e8ac370d25a833c  MesaLib-7.8.2.tar.gz
6be2d343a0089bfd395ce02aaf8adb57  MesaLib-7.8.2.tar.bz2
a04ad3b06ac5ff3969a003fa7bbf7d5b  MesaLib-7.8.2.zip
7c213f92efeb471f0331670d5079d4c0  MesaDemos-7.8.2.tar.gz
757d9e2e06f48b1a52848be9b0307ced  MesaDemos-7.8.2.tar.bz2
8d0e5cfe68b8ebf90265d350ae2c48b1  MesaDemos-7.8.2.zip
b74482e3f44f35ed395c4aada4fd8240  MesaGLUT-7.8.2.tar.gz
a471807b65e49c325808ba4551be93ed  MesaGLUT-7.8.2.tar.bz2
9f190268c42be582ef66e47365ee61e3  MesaGLUT-7.8.2.zip
```

4.252.2 New features

None.

4.252.3 Changes

- Upgraded glext.h to version 61, and upgraded glxext.h

4.252.4 Bug fixes

- Fixed Gallium glDrawPixels(GL_DEPTH_COMPONENT).
- Fixed Gallium Cell driver to buildable, runable state
- Fixed bad error checking for glFramebufferRenderbuffer(attachment=GL_DEPTH_STENCIL_ATTACHMENT).
- Fixed incorrect Z coordinate handling in "meta" glDraw/CopyPixels. Bug #23670.
- Assorted i965 driver fixes. Including but not limited to:
  - Fix scissoring when width or height is 0. Bug #27643.
  - Fix bit allocation for number of color regions for ARB_draw_buffers.
  - Set the correct provoking vertex for clipped first-mode trifans. Bug #24470.
  - Use `R16G16B16A16_FLOAT` for 3-component half-float.
  - Fix assertion for surface tile offset usage on Ironlake.
  - Fix cube map layouts on Ironlake.
  - When an RB gets a new region, clear the old from the state cache. Bug #24119.
  - Reject shaders with uninlined function calls instead of hanging.
- Assorted i915 driver fixes. Including but not limited to:
  - Fixed texture LOD clamping in i915 driver. Bug #24846.
  - Fix off-by-one for drawing rectangle. Bug #27408.
- Fixed hangs in etracer on 830 and 845 chipsets. Bug #26557.
- Fixed tiling of small textures on all Intel drivers.
- Fixed crash in Savage driver when using `_mesa_CopyTexImage2D`. Bug #27652.
- Assorted GLX fixes. Including but not limited to:
  - Fixed `__glXInitializeVisualConfigFromTags`'s handling of unrecognized fbconfig tags.
  - Fixed regression with `GLX_USE_GL`.
  - Fixed config chooser logic for 'mask' matching.
  - Report swap events correctly in direct rendered case (DRI2)
  - Fixed build with dri2proto which doesn't define `X_DRI2SwapInterval`.
  - Get `GLX_SCREEN` first in `__glXQueryContextInfo`. Bug #14245.
- Assorted GLSL fixes. Including but not limited to:
  - Change variable declared assertion into conditional in GLSL compiler. Bug #27921.
  - Fix instruction indexing bugs. Bug #27566.
  - Updated uniform location / offset encoding to be more like other implementations.
  - Don't overwrite a driver's shader infolog with generic failure message.

- Fixed OSMesa build for 16 and 32-bit color channel depth.
- Fixed OSMesa build with hidden symbol visibility. libOSMesa no longer links to libGL. Bug #28305.
- Fixed handling of multiple render targets in fixed-function texture envrionmnent programs.
- Fixed conversion errors in `signed_rgba8888[rev]` texel fetch.
- Don't set srcLevel on `GL_TEXTURE_RECTANGLE_ARB` targets.
- Various build fixes for OpenBSD.
- Various build fixes for OS X.
- Various build fixes for GCC 3.3.

4.253 Mesa 7.8.1 Release Notes / April 5, 2010

Mesa 7.8.1 fixes a couple critical bugs in the recent Mesa 7.8 release. Even though this is a bug fix release, given its proximity to the 7.8 release, a new development release, it should also be considered new development release. People who are concerned with stability and reliability should stick with a previous release, such as 7.7.1, or wait for Mesa 7.8.2.

Mesa 7.8.1 implements the OpenGL 2.1 API, but the version reported by glGetString(GL_VERSION) depends on the particular driver being used. Some drivers don't support all the features required in OpenGL 2.1.

See the *Compiling/Installing page* for prerequisites for DRI hardware acceleration.

4.253.1 MD5 checksums

```
62e8e47cbd63741b4bbe634dcdc8a56a  MesaLib-7.8.1.tar.gz
25ec15f8e41fde6d206118cc786dbac4  MesaLib-7.8.1.tar.bz2
22b1153010ffdf513836ea9931159e80  MesaLib-7.8.1.zip
c9c0a830923d3820807a08c09d521b3e  MesaDemos-7.8.1.tar.gz
9ef47f911869657c6bf2f43ebce86b61  MesaDemos-7.8.1.tar.bz2
93720605eb3f784f9bcc289a4dd2ff52  MesaDemos-7.8.1.zip
ed1d0b1e960afe6a3768eab747cbdbd3  MesaGLUT-7.8.1.tar.gz
6bae516a44c6d26ff3152c960ab648e7  MesaGLUT-7.8.1.tar.bz2
ba306f603ea73c30ee0e7efa14dc5581  MesaGLUT-7.8.1.zip
```

4.253.2 New features

None.

4.253.3 Bug fixes

- Fix incorrect enums for GLX_INTEL_swap_event by updating glxext.h to version 27 from OpenGL.org.
- Fix compilation errors on non-GLX_DIRECT_RENDERING builds.
- Various fixes for building Mesa on OS X.
- Pass GLX drawable ID to dri2InvalidateBuffers. Fixes bug #27190.

4.253.4 Changes

None.

4.254 Mesa 7.8 Release Notes / March 28, 2010

Mesa 7.8 is a new development release. People who are concerned with stability and reliability should stick with a previous release or wait for Mesa 7.8.1.

Mesa 7.8 implements the OpenGL 2.1 API, but the version reported by glGetString(GL_VERSION) depends on the particular driver being used. Some drivers don't support all the features required in OpenGL 2.1.

See the *Compiling/Installing page* for prerequisites for DRI hardware acceleration.

4.254.1 MD5 checksums

```
5fcfde5383eccb3e9fd665f08a0ea59b  MesaLib-7.8.tar.gz
85cb891eecb89aae4fdd3499cccd934b  MesaLib-7.8.tar.bz2
754f39593006effc1c8ec3c27c2f1296  MesaLib-7.8.zip
c3869c29fa6c3dbdd763f7428d271e12  MesaDemos-7.8.tar.gz
9fe8ec184c7f78691e43c4c0a7f97d56  MesaDemos-7.8.tar.bz2
063a96947f7b83d4ad789c6cf291b184  MesaDemos-7.8.zip
5f4246756b7daaddb4fb3f970cad1e28  MesaGLUT-7.8.tar.gz
ca7048a4aa7a437dcc84cc2c7d731336  MesaGLUT-7.8.tar.bz2
b54581aeb79b585b158d6a32f94feff2  MesaGLUT-7.8.zip
```

4.254.2 New features

- GL_NV_conditional_render extension (swrast driver only)
- GL_EXT_draw_buffers2 extension (swrast and i965 driver only)
- GL_ARB_fragment_coord_conventions extension (for swrast, i965, and Gallium drivers)
- GL_EXT_texture_array extension (swrast driver only)
- GL_APPLE_object_purgeable extension (swrast and i945/i965 DRI drivers)
- Much improved support for *EGL in Mesa*
- New state trackers for *OpenGL ES 1.1 and 2.0*
- Dedicated documentation for Gallium

4.254.3 Bug fixes

- Massive improvements to the Gallium driver for R300-R500 Radeons; this driver is now moderately stable but not terribly performant.

4.254.4 Changes

- Removed support for color-index rendering
- Removed support for GCC versions earlier than 3.3.0.

4.255 Mesa 7.7.1 Release Notes / March 28, 2010

Mesa 7.7.1 is a bug-fix release.

Mesa 7.7.1 implements the OpenGL 2.1 API, but the version reported by glGetString(GL_VERSION) depends on the particular driver being used. Some drivers don't support all the features required in OpenGL 2.1.

See the *Compiling/Installing page* for prerequisites for DRI hardware acceleration.

4.255.1 MD5 checksums

```
3ab0638cfa7ce8157337a229cf0db2c4  MesaLib-7.7.1.tar.gz
46664d99e03f1e3ac078a7fea02af115  MesaLib-7.7.1.tar.bz2
4e73ba8abb59aff79485eb95d7cefff7  MesaLib-7.7.1.zip
bf1b108983995f7a712cf3343df1c918  MesaDemos-7.7.1.tar.gz
aeb39645d80d656e0adebaa09e5bcd03  MesaDemos-7.7.1.tar.bz2
01c49b7454fd292244eaf8bdc6ed8cf0  MesaDemos-7.7.1.zip
37ec6386693dcb6dc770d1efd63a7a93  MesaGLUT-7.7.1.tar.gz
1e16c85282f843791a21f7bc7b6a1ca8  MesaGLUT-7.7.1.tar.bz2
d352c9e36a8e4d1059f4abc017b131e0  MesaGLUT-7.7.1.zip
```

4.255.2 Bug fixes

- Assorted fixes to VMware SVGA gallium driver.
- Fixed broken blending to multiple color buffers in swrast driver.
- Allocate constants more tightly in GL_ARB_vertex/fragment parser.
- Fixed mipmap generation bug caused by invalid viewport state.
- Gallium SSE codegen for XPD didn't always work.
- Fixed Windows build.
- Fixed broken glMultiDrawElements().
- Silence bogus GL errors generated in glxinfo.
- Fixed several render to texture bugs.
- Assorted bug fixes in Mesa/Gallium state tracker including glCopy/DrawPixels() to FBOs.
- Assorted fixes to Gallium drivers.
- Fixed broken glPush/PopClientAttrib() for vertex arrays in GLX code.

4.256 Mesa 7.7 Release Notes / 21 December 2009

Mesa 7.7 is a new development release. People who are concerned with stability and reliability should stick with a previous release or wait for Mesa 7.7.1.

Mesa 7.7 implements the OpenGL 2.1 API, but the version reported by glGetString(GL_VERSION) depends on the particular driver being used. Some drivers don't support all the features required in OpenGL 2.1.

See the *Compiling/Installing page* for prerequisites for DRI hardware acceleration.

4.256.1 MD5 checksums

```
395c9516edf1ad54b0934d8db15557bf  MesaLib-7.7.tar.gz
e3fa64a1508bc23dd9de9dd2cea7cfb1  MesaLib-7.7.tar.bz2
e54903eb5e49c3969821fa16b32da245  MesaLib-7.7.zip
53b5b6f78e55de170d43c98cb6aaab7e  MesaDemos-7.7.tar.gz
6fd616b27b9826d0faa23e08e05d9435  MesaDemos-7.7.tar.bz2
240fe06159ad73d5e22c27033b66c80a  MesaDemos-7.7.zip
9fe11a904b2a9d8cd06cc52bc330b716  MesaGLUT-7.7.tar.gz
e8dceed05a59a2d3c2619d7d734587e3  MesaGLUT-7.7.tar.bz2
96af041d435349ee23ead4667ec36363  MesaGLUT-7.7.zip
```

4.256.2 New features

- VMware "SVGA" Gallium driver. This is a Gallium3D driver which targets the VMware virtual graphics device. It allows Linux OpenGL guest applications to utilize the 3D graphics hardware of the host operating system.
- GL_ARB_draw_elements_base_vertex (supported in Intel i965 and software drivers)
- GL_ARB_depth_clamp (supported in Intel i965 DRI and software drivers)
- GL_NV_depth_clamp (supported in Intel i965 DRI and software drivers)
- GL_ARB_provoking_vertex (same as GL_EXT_provoking_vertex)
- Wavefront .obj file loader/viewer demo (progs/demos/objviewer)

4.256.3 Bug fixes

- Many assorted i965 driver fixes.
- Many r300-gallium driver fixes; this driver is now considered unstable-quality instead of experimental-quality.

4.256.4 Changes

- New Mesa texture/surface format infrastructure
- Removed some unused Mesa device driver hooks

4.257 Mesa 7.6.1 Release Notes, 21 December 2009

Mesa 7.6.1 is a bug-fix release fixing issues since version 7.6.

Mesa 7.6.1 implements the OpenGL 2.1 API, but the version reported by glGetString(GL_VERSION) depends on the particular driver being used. Some drivers don't support all the features required in OpenGL 2.1.

See the *Compiling/Installing page* for prerequisites for DRI hardware acceleration.

4.257.1 MD5 checksums

```
e80fabad2e3eb7990adae773d6aeacba  MesaLib-7.6.1.tar.gz
7db4617e9e10ad3aca1b64339fd71b7d  MesaLib-7.6.1.tar.bz2
dd3275dbf9833480d2e92d0c69b22abd  MesaLib-7.6.1.zip
f7fdcfe3c0f363e571c60f02f74368fb  MesaDemos-7.6.1.tar.gz
a4226f06732a02556fcf6be290b86dff  MesaDemos-7.6.1.tar.bz2
849425f356bd940726cebedfa79de176  MesaDemos-7.6.1.zip
d40cc7c5e337a85b674e27a8e494f52f  MesaGLUT-7.6.1.tar.gz
ca9aecb91f05b1da9fd7d5eeb19d47d7  MesaGLUT-7.6.1.tar.bz2
23fad8398004c977a1d8953079b72ca6  MesaGLUT-7.6.1.zip
```

4.257.2 New features

- Upgraded GL/glext.h to version 56, GL/glxext.h to version 25, GL/wglext.h to version 17
- New 3D driver, r600, for Radeon R6xx, R7xx hardware

4.257.3 Bug fixes

- Fixed crash caused by glXCopyContext() and glXDestroyContext(), bug 24217
- glXQueryContext(GLX_RENDER_TYPE) returned wrong values (bug 24211)
- GLSL sqrt(0) returned unpredictable results
- Fixed default texture binding bug when a bound texture was deleted.
- r300: Work around an issue with very large fragment programs on R500.
- Fake glXQueryDrawable() didn't return good values (bug 24320)
- Fixed AUX buffer breakage (bug 24426).
- Fixed locale-dependent float parsing bug in GLSL compiler (bug 24531)
- Fixed Gallium Cell driver compilation failure.
- Fixed glGetTexLevelParameter(GL_TEXTURE_INTERNAL_FORMAT) query so that it returns the actual compressed format chosen.
- Fixed glBitmap bugs in Intel drivers.
- Fixed a number of Microsoft Visual Studio compilation problems.
- Fixed clipping / provoking vertex bugs in i965 driver.
- Assorted build fixes for AIX.
- Endianness fixes for the DRI swrast driver (bug 22767).
- Point sprite fixes for i915/945 driver.
- Fixed assorted memory leaks (usually on error paths)
- Fixed some GLSL compiler bugs (ex: 25579)
- Assorted build fixes for BlueGene

4.257.4 Changes

- Removed old VC6, VC7 project files for Windows

4.258 Mesa 7.6 Release Notes, 28 September 2009

Mesa 7.6 is a new development release. People who are concerned with stability and reliability should stick with a previous release or wait for Mesa 7.6.1.

Mesa 7.6 implements the OpenGL 2.1 API, but the version reported by glGetString(GL_VERSION) depends on the particular driver being used. Some drivers don't support all the features required in OpenGL 2.1.

See the *Compiling/Installing page* for prerequisites for DRI hardware acceleration.

4.258.1 MD5 checksums

```
5ffa7d7abf8973f57a1bc4f813e6dade  MesaLib-7.6.tar.gz
8c75f90cd0303cfac9e4b6d54f6759ca  MesaLib-7.6.tar.bz2
27fcfd69708599c978cb34ba5cd363e1  MesaLib-7.6.zip
e7befb3ae604f591806194a4da445628  MesaDemos-7.6.tar.gz
0ede7adf217951acd90dbe4551210c07  MesaDemos-7.6.tar.bz2
ed9298409cf6613bc0964525ca4afc8a  MesaDemos-7.6.zip
666955668e44ff14acf7d15dc78407d3  MesaGLUT-7.6.tar.gz
b8b59706f827d18d1b784a0ff98b4dc2  MesaGLUT-7.6.tar.bz2
c49c19c2bbef4f3b7f1389974dff25f4  MesaGLUT-7.6.zip
```

4.258.2 New features

- OpenVG front-end (state tracker for Gallium). This was written by Zack Rusin at Tungsten Graphics.
- GL_ARB_vertex_array_object and GL_APPLE_vertex_array_object extensions (supported in Gallium drivers, Intel DRI drivers, and software drivers)
- GL_ARB_copy_buffer extension (supported in Gallium drivers, Intel DRI drivers, and software drivers)
- GL_ARB_map_buffer_range extension (supported in Gallium drivers, Intel DRI drivers, and software drivers)
- GL_ARB_seamless_cube_map extension (supported in software drivers and i965 drivers)
- GL_ARB_vertex_array_bgra (ARB synonym for GL_EXT_vertex_array_bgra)
- GL_ARB_sync (supported in software drivers and Intel DRI drivers)
- GL_EXT_provoking_vertex extension (supported in Gallium, i915, i965, and software drivers)
- Rewritten radeon/r200/r300 driver using a buffer manager
- radeon/r200/r300 GL_EXT_framebuffer_object support when used with kernel memory manager
- radeon/r200/r300 support for GL_ARB_occlusion_query
- r300 driver supports OpenGL 1.5
- r300 driver support for GL_EXT_vertex_array_bgra, GL_EXT_texture_sRGB
- i915/945 driver support for GL_ARB_point_sprite, GL_EXT_stencil_two_side and GL_ATI_separate_stencil extensions
- Rewritten assembler for GL_ARB_vertex_program / GL_ARB_fragment_program.
- Added configure –with-max-width=W, –with-max-height=H options to specify max framebuffer, viewport size.
- Initial version of Gallium llvmpipe driver. This is a new driver based on LLVM which makes exensive use of run-time code generation. This is an "alpha" stage driver. See the src/gallium/drivers/llvmpipe/README file for more information.

4.258.3 Bug fixes

- i965 DRI driver fixes, including support for "unlimited" size constant buffers (GLSL uniforms)

4.259 Mesa 7.5.2 Release Notes, 28 September 2009

Mesa 7.5.2 is a bug-fix release fixing issues found since the 7.5.1 release.

The main new feature of Mesa 7.5.x is the Gallium3D infrastructure.

Mesa 7.5.2 implements the OpenGL 2.1 API, but the version reported by glGetString(GL_VERSION) depends on the particular driver being used. Some drivers don't support all the features required in OpenGL 2.1.

See the *Compiling/Installing page* for prerequisites for DRI hardware acceleration.

4.259.1 MD5 checksums

```
43a90191dd9f76cd65dcc1ac91f3be70  MesaLib-7.5.2.tar.gz
94e47a499f1226803869c2e37a6a8e3a  MesaLib-7.5.2.tar.bz2
1ecb822b567ad67a0617361d45206b67  MesaLib-7.5.2.zip
2718fdce7e075911d6147beb8f27104b  MesaDemos-7.5.2.tar.gz
4e0f5ccd58afe21eddcd94327d926e86  MesaDemos-7.5.2.tar.bz2
f621f8c223b278d7c8e49a012d56ca25  MesaDemos-7.5.2.zip
83c16c1d6bcfcc3f97aab5d2fe430b4c  MesaGLUT-7.5.2.tar.gz
e5d03bedae369ea3705783573bb33813  MesaGLUT-7.5.2.tar.bz2
e82ba28e00d653e6f437d32be8ca8481  MesaGLUT-7.5.2.zip
```

4.259.2 New features

- Detect B43 chipset in Intel driver

4.259.3 Bug fixes

- Assorted bug fixes for i965/i945 drivers
- Fixed Gallium glDrawPixels(GL_STENCIL_INDEX) failure.
- Fixed GLSL linker/preprocessor version directive issue seen in Wine (such as bug 23946)
- glUseProgram() is now compiled into display lists (bug 23746).
- glUniform functions are now compiled into display lists
- Auto mipmap generation didn't work reliably with Gallium.
- Fixed random number usage in GLX code.
- Fixed invalid GL_OUT_OF_MEMORY error sometimes raised by glTexSubImage2D when using Gallium.

4.260 Mesa 7.5.1 Release Notes, 3 September 2009

Mesa 7.5.1 is a bug-fix release fixing issues found since the 7.5 release.

The main new feature of Mesa 7.5.x is the Gallium3D infrastructure.

Mesa 7.5.1 implements the OpenGL 2.1 API, but the version reported by glGetString(GL_VERSION) depends on the particular driver being used. Some drivers don't support all the features required in OpenGL 2.1.

See the *Compiling/Installing page* for prerequisites for DRI hardware acceleration.

4.260.1 MD5 checksums

```
d7269e93bc7484430637d54ced250876  MesaLib-7.5.1.tar.gz
877d6a4b24efc2b1d02aa553f262cba8  MesaLib-7.5.1.tar.bz2
23f4fb757a05c8396425681234ae20e5  MesaLib-7.5.1.zip
5af4bd113652108f5cec5113dad813f2  MesaDemos-7.5.1.tar.gz
785402e3b9f0e335538fcc6bf19f6987  MesaDemos-7.5.1.tar.bz2
950058cc6d6106e9c7d5876a03789fe9  MesaDemos-7.5.1.zip
cb52ce2c93389c2711cbe8d857ec5303  MesaGLUT-7.5.1.tar.gz
e3a9892e056d625c5353617a7c5b7e9c  MesaGLUT-7.5.1.tar.bz2
da1de364df148c94b4994006191a1e69  MesaGLUT-7.5.1.zip
```

4.260.2 New features

- Added configure –with-max-width=W, –with-max-height=H options to specify max framebuffer, viewport size.

4.260.3 Bug fixes

- Added missing GLEW library to MesaDemos tarballs.
- Fixed swapbuffers jerkiness in Doom3/etc in Intel drivers.
- Fixed front buffer rendering bug in Intel drivers.
- Fixed minor GLX memory leaks.
- Fixed some texture env / fragment program state bugs.
- Fixed some Gallium glBlitFramebuffer() bugs
- Empty glBegin/glEnd() pair could cause divide by zero (bug 23489)
- Fixed Gallium glBitmap() Z position bug
- Setting arrays of sampler uniforms did not work
- Selection/Feedback mode didn't handle polygon culling correctly (bug 16866)
- Fixed 32/64-bit cross compilation issue in gen_matypes.c
- Fixed glXCreateGLXPixmap() for direct rendering.
- Fixed Gallium glCopyPixels(GL_STENCIL_INDEX) mispositioned image bug.

4.261 Mesa 7.5 Release Notes / 17 July 2009

Mesa 7.5 is a new development release. People who are concerned with stability and reliability should stick with the 7.4.x branch or wait for Mesa 7.5.1.

The main new feature of Mesa 7.5 is the Gallium3D infrastructure.

Mesa 7.5 implements the OpenGL 2.1 API, but the version reported by glGetString(GL_VERSION) depends on the particular driver being used. Some drivers don't support all the features required in OpenGL 2.1.

See the *Compiling/Installing page* for prerequisites for DRI hardware acceleration.

Note that the Mesa project is no longer using odd/even version numbers to indicate development/stable releases. The so-called development releases have been fairly stable. If you're especially concerned with stability you should probably look for "point" releases such as 7.5.1 which will be a bug-fix release.

4.261.1 MD5 checksums

```
553fd956e544727f30fbe249619b6286  MesaLib-7.5.tar.gz
459f332551f6ebb86f384d21dd15e1f0  MesaLib-7.5.tar.bz2
8c02c0e17a9025250d20424ae32f5163  MesaLib-7.5.zip
a188da2886fa5496ea0c2cda602b2eeb  MesaDemos-7.5.tar.gz
398ee8801814a00e47f6c2314e3dfddc  MesaDemos-7.5.tar.bz2
15a0c8ae013c54335a26335e1a98d609  MesaDemos-7.5.zip
81010147def5a644ba14f9bbb7a49a2a  MesaGLUT-7.5.tar.gz
baa7a1e850b6e39bae58868fd0684004  MesaGLUT-7.5.tar.bz2
265228418e4423fa328f2f5b7970cf08  MesaGLUT-7.5.zip
```

4.261.2 New features

- Gallium3D - this is the new architecture for OS-independent and API-independent 3D drivers. Gallium3D is intended for GPUs that fully support vertex/fragment shaders. The Gallium3D drivers currently included are:
  - softpipe - a software/reference driver
  - i915 - Intel 915/945 driver
  - Cell - IBM/Sony/Toshiba Cell processor driver
  - nouveau (for NVIDIA GPUs) and R300 for (AMD/ATI R300).

  Note: these drivers are incomplete and still under development. It's probably NOT worthwhile to report any bugs unless you have patches.

- GL_ARB_framebuffer_object extension (software drivers, i965 driver)
- Reworked two-sided stencil support. This allows a driver to support all three variations of two-sided stencil including GL_ATI_separate_stencil, GL_EXT_stencil_two_side and OpenGL 2.0
- GL_EXT_vertex_array_bgra extension (software drivers, i965 driver)
- GL_NV_texture_env_combine4 extension (software drivers, i965/i915 drivers)
- GL_EXT_texture_swizzle extension (software drivers, i965 driver)
- Updated SPARC assembly optimizations (David S. Miller)
- Initial support for separate compilation units in GLSL compiler.

- Increased max number of generic GLSL varying variables to 16 (formerly 8).
- GLSL linker now detects when too many varying variables are used.
- Optimize-out redundant glMaterial and glShadeModel calls in display lists
- Fixed gl_TextureMatrix[i][j] array indexing bug in GLSL compiler.

4.261.3 Bug fixes

- Lots of i965 driver bug fixes
- Fixed some GLSL preprocessor bugs
- GLSL: continue inside of a for-loop didn't work

4.261.4 Changes

- Remove support for GL_SGIX_shadow, GL_SGIX_shadow_ambient and GL_SGIX_depth_texture extensions. Superseded by the ARB versions.
- Omitted some old Mesa demos from the release tarballs, added some others.

4.262 Mesa 7.4.4 Release Notes / 23 June 2009

Mesa 7.4.4 is a stable development release fixing bugs since the 7.4.3 release.

Mesa 7.4.4 implements the OpenGL 2.1 API, but the version reported by glGetString(GL_VERSION) depends on the particular driver being used. Some drivers don't support all the features required in OpenGL 2.1.

See the *Compiling/Installing page* for prerequisites for DRI hardware acceleration.

4.262.1 MD5 checksums

```
0b56fe5a88ab0c3c5b2da5068f63f416  MesaLib-7.4.4.tar.gz
b66528d314c574dccbe0ed963cac5e93  MesaLib-7.4.4.tar.bz2
2818076f3ba23fa87fdfe4602a637a18  MesaLib-7.4.4.zip
3e77b208386c47b18165bce5ae317e2c  MesaDemos-7.4.4.tar.gz
628142ec9a54cd28cc027e6ce26cff47  MesaDemos-7.4.4.tar.bz2
d08a30d30ab7174859aa709cba6c726d  MesaDemos-7.4.4.zip
e6e91ba16e274d40cf3a97ad3218af01  MesaGLUT-7.4.4.tar.gz
e14bbb52517e8121b31f1387515365ab  MesaGLUT-7.4.4.tar.bz2
f10ed20469753c2b3d68c99854f80fd4  MesaGLUT-7.4.4.zip
```

4.262.2 Bug fixes

- Fixed i965/i915 segfault in screen destruction (bug 22408)

4.262.3 Driver Status

```
Driver         Status
----------------------  ----------------------
DRI drivers     varies with the driver
XMesa/GLX (on Xlib) implements OpenGL 2.1
OSMesa (off-screen) implements OpenGL 2.1
Windows/Win32       implements OpenGL 2.1
Glide (3dfx Voodoo1/2)  implements OpenGL 1.3
SVGA            unsupported
Wind River UGL      unsupported
DJGPP           unsupported
GGI         unsupported
BeOS            unsupported
Allegro         unsupported
D3D         unsupported
```

4.263 Mesa 7.4.3 Release Notes / 19 June 2009

Mesa 7.4.3 is a stable development release fixing bugs since the 7.4.2 release.

Mesa 7.4.3 implements the OpenGL 2.1 API, but the version reported by glGetString(GL_VERSION) depends on the particular driver being used. Some drivers don't support all the features required in OpenGL 2.1.

See the *Compiling/Installing page* for prerequisites for DRI hardware acceleration.

4.263.1 MD5 checksums

```
34c5a6c47ed51f31c4fa36e269831352  MesaLib-7.4.3.tar.gz
70a983ba3deaa8bd63b18bbab283f698  MesaLib-7.4.3.tar.bz2
34f21b3205b271d575030aa98a2dda51  MesaLib-7.4.3.zip
56752b7adede212e6097afb10d0c0d59  MesaDemos-7.4.3.tar.gz
8ffa51c4833b1e298300a005e2d7ca2a  MesaDemos-7.4.3.tar.bz2
0037d24d41400d6fb9800ae55b8c863f  MesaDemos-7.4.3.zip
20e24f6692c0c90e7e3b220f79c4108d  MesaGLUT-7.4.3.tar.gz
03a4beeef74fc5ef0b1d6d04710e5a8a  MesaGLUT-7.4.3.tar.bz2
273788230adbdb9d57371309adedcf5f  MesaGLUT-7.4.3.zip
```

4.263.2 Bug fixes

- Fixed texture object reference counting bug (bug 21756)
- Allow depth/stencil textures to be attached to GL_STENCIL_ATTACHMENT point (SF bug 2793846)
- Added missing glGet case for GL_VERTEX_ARRAY_BINDING_APPLE
- Fixed some OSMesa build issues
- Fixed a vertex buffer object crash
- Fixed broken glTexImage3D() when image type = GL_BITMAP
- Fixed some GLSL preprocessor bugs
- Fixed framebuffer mem leak in i945/i965 DRI drivers
- Fixed texture coordinate repeat bug in swrast (bug 21872)

- Fixed incorrect viewport clamping (lower bound is zero, not one)
- GLX fix for glean's makeCurrent test case

4.263.3 Driver Status

```
Driver         Status
----------------------  ----------------------
DRI drivers     varies with the driver
XMesa/GLX (on Xlib) implements OpenGL 2.1
OSMesa (off-screen) implements OpenGL 2.1
Windows/Win32       implements OpenGL 2.1
Glide (3dfx Voodoo1/2)  implements OpenGL 1.3
SVGA            unsupported
Wind River UGL      unsupported
DJGPP           unsupported
GGI         unsupported
BeOS            unsupported
Allegro         unsupported
D3D         unsupported
```

4.264 Mesa 7.4.2 Release Notes / May 15, 2009

Mesa 7.4.2 is a stable development release fixing bugs since the 7.4.1 release.

Mesa 7.4.2 implements the OpenGL 2.1 API, but the version reported by glGetString(GL_VERSION) depends on the particular driver being used. Some drivers don't support all the features required in OpenGL 2.1.

See the *Compiling/Installing page* for prerequisites for DRI hardware acceleration.

4.264.1 MD5 checksums

```
172f5193154dad731387f97bd44ab68f  MesaLib-7.4.2.tar.gz
b10a76e32bde4645cfc34ea0416d7d8b  MesaLib-7.4.2.tar.bz2
cc6dfc2efd424cc342b84e6bcd78ce5d  MesaLib-7.4.2.zip
182a7e78aa7a480b3650a5c956dbddd1  MesaDemos-7.4.2.tar.gz
bf559a0485667a3bfa4513a23501579b  MesaDemos-7.4.2.tar.bz2
5379e622b65e8c22022dba34aeb6f4f9  MesaDemos-7.4.2.zip
7cc43c1c35bf6a279a16e063dea3b8c5  MesaGLUT-7.4.2.tar.gz
e0dfc44d537904a030861e5b3c760c11  MesaGLUT-7.4.2.tar.bz2
4a6cf5bbbac190d6ba97448b3098b7f4  MesaGLUT-7.4.2.zip
```

4.264.2 Bug fixes

- Fixed segfault when rendering to front buffer with DRI 1.
- Fixed swrast texture rectangle bug when wrap mode = GL_CLAMP_TO_BORDER and filter mode = GL_LINEAR. (bug 21461)
- Fixed texture object mem leak during context destruction.
- Fixed a state validation bug in glCopyTex[Sub]Image()
- Fixed some i965 GLSL bugs.

- Fixed an R300 driver texture object bad memory reference.

4.264.3 Driver Status

```
Driver         Status
---------------------  ---------------------
DRI drivers     varies with the driver
XMesa/GLX (on Xlib) implements OpenGL 2.1
OSMesa (off-screen) implements OpenGL 2.1
Windows/Win32       implements OpenGL 2.1
Glide (3dfx Voodoo1/2)  implements OpenGL 1.3
SVGA            unsupported
Wind River UGL      unsupported
DJGPP           unsupported
GGI         unsupported
BeOS            unsupported
Allegro         unsupported
D3D         unsupported
```

4.265 Mesa 7.4.1 Release Notes / 18 April 2009

Mesa 7.4.1 is a stable development release fixing bugs since the 7.4 release.

Mesa 7.4.1 implements the OpenGL 2.1 API, but the version reported by glGetString(GL_VERSION) depends on the particular driver being used. Some drivers don't support all the features required in OpenGL 2.1.

See the *Compiling/Installing page* for prerequisites for DRI hardware acceleration.

4.265.1 MD5 checksums

```
0c3a72f3295a53a134c04bd7d209ea62  MesaLib-7.4.1.tar.gz
423260578b653818ba66c2fcbde6d7ad  MesaLib-7.4.1.tar.bz2
84f78b154d4bd5c3ecc42eeff2e56676  MesaLib-7.4.1.zip
aa0ad323e59d6d10ff33ac0dde462a60  MesaDemos-7.4.1.tar.gz
1e169fb6abc2b45613f1c98a82dfe690  MesaDemos-7.4.1.tar.bz2
294e42be2d74176596c994ec23322fcf  MesaDemos-7.4.1.zip
92373bfa48e7b68dddf356e86b0e5699  MesaGLUT-7.4.1.tar.gz
336f3824b578b072211e0beecf4f04f4  MesaGLUT-7.4.1.tar.bz2
20751388d8ef16b42d25d9e3d705d101  MesaGLUT-7.4.1.zip
```

4.265.2 Bug fixes

- Fixed a two-sided lighting bug in fixed-function-to-GPU code generation
- Fixed some Darwin issues (Jeremy Huddleston)
- Indexing the GLSL gl_EyePlane[] or gl_ObjectPlane[] arrays with a variable was broken, bug 20986
- Fixed incorrect texture unit bias in TXB instruction
- glTexParameter settings weren't always propogated to drivers
- Assorted vertex/fragment program bug fixes

- Fixed point rendering in software rasterizer
- Fixed potential deadlock in object hash functions
- Fix a couple bugs surrounding front-buffer rendering with DRI2, but this is not quite complete.
- Fixed glPopAttrib() bug when restoring user clip planes

4.265.3 Driver Status

```
Driver         Status
----------------------  ----------------------
DRI drivers     varies with the driver
XMesa/GLX (on Xlib) implements OpenGL 2.1
OSMesa (off-screen) implements OpenGL 2.1
Windows/Win32       implements OpenGL 2.1
Glide (3dfx Voodoo1/2)  implements OpenGL 1.3
SVGA            unsupported
Wind River UGL      unsupported
DJGPP           unsupported
GGI         unsupported
BeOS            unsupported
Allegro         unsupported
D3D         unsupported
```

4.266 Mesa 7.4 Release Notes / 27 March 2009

Mesa 7.4 is a stable development release fixing bugs since the 7.3 release.

Mesa 7.4 implements the OpenGL 2.1 API, but the version reported by glGetString(GL_VERSION) depends on the particular driver being used. Some drivers don't support all the features required in OpenGL 2.1.

See the *Compiling/Installing page* for prerequisites for DRI hardware acceleration.

4.266.1 MD5 checksums

```
ed6bd7437177307e51e16d0c7c381dfa  MesaLib-7.4.tar.gz
7ecddb341a2691e0dfdb02f697109834  MesaLib-7.4.tar.bz2
433e823f8245f9fd5f397e7b719a8e47  MesaLib-7.4.zip
656eee6128016fb237e01aa8dabbc703  MesaDemos-7.4.tar.gz
02816f10f30b1dc5e069e0f68c177c98  MesaDemos-7.4.tar.bz2
44a70d6db4aa4c64ecc47871b6aceee8  MesaDemos-7.4.zip
25f80db4f8645cd3e58e2c9af53ec341  MesaGLUT-7.4.tar.gz
04ec01caebde44f5b0d619f00716b368  MesaGLUT-7.4.tar.bz2
019dc213baecaa3cb1278847d41b8591  MesaGLUT-7.4.zip
```

4.266.2 New features

- Added MESA_GLX_FORCE_DIRECT env var for Xlib/software driver
- GLSL version 1.20 is returnd by the GL_SHADING_LANGUAGE_VERSION query

4.266.3 Bug fixes

- glGetActiveUniform() returned wrong size for some array types
- Fixed some error checking in glUniform()
- Fixed a potential glTexImage('proxy target') segfault
- Fixed bad reference counting for 1D/2D texture arrays
- Fixed VBO + glPush/PopClientAttrib() bug #19835
- Assorted i965 driver bug fixes
- Fixed a Windows compilation failure in s_triangle.c
- Fixed a GLSL array indexing bug
- Fixes for building on Haiku

4.266.4 Changes

- Updated GL/glxext.h to version 48
- Assorted updates for building on Solaris

4.266.5 Driver Status

```
Driver         Status
----------------------  ----------------------
DRI drivers     varies with the driver
XMesa/GLX (on Xlib) implements OpenGL 2.1
OSMesa (off-screen) implements OpenGL 2.1
Windows/Win32       implements OpenGL 2.1
Glide (3dfx Voodoo1/2)  implements OpenGL 1.3
SVGA            unsupported
Wind River UGL      unsupported
DJGPP           unsupported
GGI         unsupported
BeOS            unsupported
Allegro         unsupported
D3D         unsupported
```

4.267 Mesa 7.3 Release Notes / 22 January 2009

Mesa 7.3 is a new development release. Users especially concerned with stability should stick with latest stable release: version 7.2.

Mesa 7.3 implements the OpenGL 2.1 API, but the version reported by glGetString(GL_VERSION) depends on the particular driver being used. Some drivers don't support all the features required in OpenGL 2.1.

See the *Compiling/Installing page* for prerequisites for DRI hardware acceleration.

4.267.1 MD5 checksums

```
8ed03191432b22d118d88d6db497f304  MesaLib-7.3.tar.gz
781e7811a6ed5c97b2b8defefc8ffbc9  MesaLib-7.3.tar.bz2
3ccba9a1734ed6d4b3389e1535d90fbf  MesaLib-7.3.zip
d312e974b31043b13b61bac5fbf00b87  MesaDemos-7.3.tar.gz
3f0741394069bdf2329565a387396cda  MesaDemos-7.3.tar.bz2
4d0887fd4c66a824295cdd619f6d34cb  MesaDemos-7.3.zip
2d7661b66022bcb8878728f3d5bd33ab  MesaGLUT-7.3.tar.gz
abe8036a724c1a483bdad6b5a55ddc1a  MesaGLUT-7.3.tar.bz2
5f247819b47e2a7c62d07a6afe5262fb  MesaGLUT-7.3.zip
```

4.267.2 New features

- Support for GLSL 1.20
- Intel DRI drivers now use GEM and DRI2

4.267.3 Bug fixes

- Assorted GLSL bug fixes
- Assorted i965 driver fixes
- Fix for wglCreateLayerContext() in WGL/Windows driver
- Build fixes for OpenBSD and gcc 2.95
- GLSL preprocessor handles #pragma now
- Fix incorrect transformation of GL_SPOT_DIRECTION
- Fixed several bugs (#18367 and #19625) in glXMakeContextCurrent()
- Assorted Windows build fixes

4.267.4 Changes

- Deprecated the "XMesa" interface (include/GL/xmesa*.h files)
- Deprecated the "FXMesa" interface (include/GL/fxmesa.h file)
- Deprecated the "Allegro" interface (include/GL/amesa.h file)
- Removed include/GL/uglmesa.h header
- Removed include/GLView.h header for BeOS

4.267.5 Driver Status

```
Driver         Status
----------------------  ----------------------
DRI drivers     varies with the driver
XMesa/GLX (on Xlib) implements OpenGL 2.1
OSMesa (off-screen) implements OpenGL 2.1
Windows/Win32       implements OpenGL 2.1
```

(continues on next page)

(continued from previous page)

```
Glide (3dfx Voodoo1/2)  implements OpenGL 1.3
SVGA            unsupported
Wind River UGL      unsupported
DJGPP           unsupported
GGI         unsupported
BeOS            unsupported
Allegro         unsupported
D3D         unsupported
```

4.268 Mesa 7.2 Release Notes / 20 September 2008

Mesa 7.2 is a stable release fixing bugs found in 7.1, which was a new development release.

Mesa 7.2 implements the OpenGL 2.1 API, but the version reported by glGetString(GL_VERSION) depends on the particular driver being used. Some drivers don't support all the features required in OpenGL 2.1.

Note that this version of Mesa does not use the GEM memory manager. The master branch of git uses GEM. The prototype DRI2 code that was in 7.1 has also been removed.

DRM version 2.3.1 should be used with Mesa 7.2

4.268.1 MD5 checksums

```
81a2a4b7cbfce7553f7ad8d924edbe2f  MesaLib-7.2.tar.gz
04d379292e023df0b0266825cb0dbde5  MesaLib-7.2.tar.bz2
8bc497a37977a55e987a4d1fabc3d882  MesaLib-7.2.zip
10c762e39486df395838af1d7b57e69c  MesaDemos-7.2.tar.gz
22e03dc4038cd63f32c21eb60994892b  MesaDemos-7.2.tar.bz2
1197bc4eb3bf44e291c14d4eb2e19381  MesaDemos-7.2.zip
42e3c6c6d156cd9dc545dbef72407354  MesaGLUT-7.2.tar.gz
f67daf93e12c4a459703bbf3e4004e31  MesaGLUT-7.2.tar.bz2
0390567eb2c2d12fbf82e8523fd77e2b  MesaGLUT-7.2.zip
```

4.268.2 New features

- i965 driver: added support for G41 chipset (Intel)

4.268.3 Bug fixes

- Fixed display list bug involving primitives split across lists (bug 17564)
- Fixed some issues with glBindAttribLocation()
- Fixed crash in _tnl_InvalidateState() found with Amira (bug 15834)
- Assorted bug fixes for Ming build
- Fixed some vertex/pixel buffer object reference counting bugs
- Fixed depth/stencil bug in i915/945 driver
- Fixed some shader flow control bugs in i965 driver
- Fixed a few tdfx driver bugs which prevented driver from working

- Fixed multisample enable/disable bug

4.268.4 Changes

- Updated SGI header files with new license terms.

4.268.5 To Do (someday) items

- Remove the MEMCPY() and _mesa_memcpy() wrappers and just use memcpy(). Probably do the same for malloc, calloc, etc. The wrappers were useful in the past for memory debugging but now we have valgrind. Not worried about SunOS 4 support anymore either...
- Switch to freeglut
- Fix linux-glide target/driver.
- Improved lambda and derivative calculation for frag progs.

4.268.6 Driver Status

```
Driver         Status
----------------------  ----------------------
DRI drivers     varies with the driver
XMesa/GLX (on Xlib) implements OpenGL 2.1
OSMesa (off-screen) implements OpenGL 2.1
Windows/Win32       implements OpenGL 2.1
Glide (3dfx Voodoo1/2)  implements OpenGL 1.3
SVGA            unsupported
Wind River UGL      unsupported
DJGPP           unsupported
GGI         unsupported
BeOS            unsupported
Allegro         unsupported
D3D         unsupported
```

4.269 Mesa 7.1 Release Notes / August 26, 2008

Mesa 7.1 is a new development release. There have been many internal code changes since Mesa 7.0.x. It should be relatively stable, but those who are especially concerned about stability should wait for Mesa 7.2 or use Mesa 7.0.4 (the previous stable release).

Note that this version of Mesa does not use the GEM memory manager. The master branch of git uses GEM.

DRM version 2.3.1 should be used with Mesa 7.1

4.269.1 MD5 checksums

```
971c2fe6e6949dc5ba200a6f97a6dc81  MesaLib-7.1.tar.gz
6bff7f532d16f90f944a400c8bd7074d  MesaLib-7.1.tar.bz2
d48224bf9d54c3da6776adb4869ba024  MesaLib-7.1.zip
3de268420efca43e9a19ab506cdfc993  MesaDemos-7.1.tar.gz
```

(continues on next page)

(continued from previous page)

```
abfc9775e1462363af8ec160d1feb01f  MesaDemos-7.1.tar.bz2
f7b3623387c4036e9895cd9ac0dfad99  MesaDemos-7.1.zip
fdf348f78cd09304b6ff801ef8acc8eb  MesaGLUT-7.1.tar.gz
f6d88a4eeb02e98c7e92f1c895d3c76b  MesaGLUT-7.1.tar.bz2
4dc102a5ca51e1c41dde87d3f8c7b22a  MesaGLUT-7.1.zip
```

4.269.2 New features

- autoconf-based configuration (and clean-up of Makefiles)
- Assorted DRI driver enhancements
- Reduced dependencies between X server and Mesa
- GL_EXT_texture_from_pixmap extension for Xlib driver
- Support for the GL shading language with i965 driver (implemented by Intel)
- ATI R500 series support (Radeon X1300–X1950) in r300 DRI driver

4.269.3 Bug fixes

- Numerous GLSL fixes
- Fixed some error code/detection bugs in the GLSL-related API functions
- Lots of DRI driver fixes.

4.269.4 To Do (someday) items

- Remove the MEMCPY() and _mesa_memcpy() wrappers and just use memcpy(). Probably do the same for malloc, calloc, etc. The wrappers were useful in the past for memory debugging but now we have valgrind. Not worried about SunOS 4 support anymore either…
- Switch to freeglut
- Fix linux-glide target/driver.
- Improved lambda and derivative calculation for frag progs.

4.269.5 Driver Status

```
Driver         Status
----------------------  ----------------------
DRI drivers     varies with the driver
XMesa/GLX (on Xlib) implements OpenGL 2.1
OSMesa (off-screen) implements OpenGL 2.1
Windows/Win32       implements OpenGL 2.1
Glide (3dfx Voodoo1/2)  implements OpenGL 1.3
SVGA            unsupported
Wind River UGL      unsupported
DJGPP           unsupported
GGI         unsupported
BeOS            unsupported
```

(continues on next page)

(continued from previous page)

```
Allegro         unsupported
D3D         unsupported
```

4.270 Mesa 7.0.4 Release Notes / August 16, 2008

Mesa 7.0.4 is a stable release with bug fixes since version 7.0.3.

4.270.1 MD5 checksums

```
909afa3a01ae31478d363837681415ac  MesaLib-7.0.4.tar.gz
8d7bacbe0234742a5d08c8088c4619e9  MesaLib-7.0.4.tar.bz2
5e44261ef85b049a868e1785d9adc276  MesaLib-7.0.4.zip
53dcd77d37a819feaf50b5fcdd0a6e0f  MesaDemos-7.0.4.tar.gz
c1215b31c5f7b85f81eed3bfba07d556  MesaDemos-7.0.4.tar.bz2
b1825a7361f116b28d82d328077630b4  MesaDemos-7.0.4.zip
d7677d015f52602d1bf8b837fb717848  MesaGLUT-7.0.4.tar.gz
f5f8b46f7e763d9f7b7d1d115c1c44ee  MesaGLUT-7.0.4.tar.bz2
a786775271a02c62a3370b13b26bf48d  MesaGLUT-7.0.4.zip
```

4.270.2 Bug fixes

- define #extension GL_ARB_texture_rectangle in shading language
- fixed WIN32 compile problem in libGLU
- Fixed a per-vertex glMaterial bug which could cause bad lighting
- Fixed potential crash in AA/smoothed triangle rendering when using a fragment shader
- Fixed glDrawElement + VBO segfault (bug 16156)
- Fixed GLSL linker bug causing generic vertex attributes to get aliased
- Fixed stack overflow when using glPixelZoom on Windows
- Fixed broken all(bvec2) GLSL function, added misc missing bvec constructors
- ARB program "state.clip[n].plane" didn't parse correctly
- Fixed broken glGetUniformiv() (bug 13774)

4.270.3 Changes

- Including the latest glext.h and glxext.h header files from Khronos
- Added support for DragonFly OS
- Added a build config for FreeBSD static libs (Anatolij Shkodin)
- Enabled GL_EXT_multi_draw_arrays extension in R200/R300 drivers
- Enabled GL_ARB_point_sprite extension in I965 driver
- Enabled GL_EXT_texture_sRGB extension in I965 driver
- Added support for GL shading language in I965 driver

4.270.4 Driver Status

```
Driver         Status
----------------------  ----------------------
DRI drivers     varies with the driver
XMesa/GLX (on Xlib) implements OpenGL 2.1
OSMesa (off-screen) implements OpenGL 2.1
Windows/Win32       implements OpenGL 2.1
Glide (3dfx Voodoo1/2)  implements OpenGL 1.3
SVGA            unsupported
Wind River UGL      unsupported
DJGPP           unsupported
GGI         unsupported
BeOS            unsupported
Allegro         unsupported
D3D         unsupported
```

4.271 Mesa 7.0.3 Release Notes / April 4, 2008

Mesa 7.0.3 is a stable release with bug fixes since version 7.0.2.

4.271.1 MD5 checksums

```
3fd1cb76531b2515ef7db92d9a93dbf8  MesaLib-7.0.3.tar.gz
e6e6379d7793af40a6bc3ce1bace572e  MesaLib-7.0.3.tar.bz2
97882bac195229ee0b78cab82e0e3be1  MesaLib-7.0.3.zip
8abf6bbcb1661e7dd4ce73b3fbb85898  MesaDemos-7.0.3.tar.gz
47fd6863621d3c9c7dbb870ab7f0c303  MesaDemos-7.0.3.tar.bz2
99e442e14da1928f76a7297bb421a3af  MesaDemos-7.0.3.zip
2b50fe9fadc4709b57c52adef09fce3c  MesaGLUT-7.0.3.tar.gz
0ff23c4e91b238abae63a5fc9fa003e7  MesaGLUT-7.0.3.tar.bz2
70e83554a4462dad28e0d6e20f79aada  MesaGLUT-7.0.3.zip
```

4.271.2 Bug fixes

- Added missing glw.pc.in file to release tarball
- Fix GLUT/Fortran issues
- GLSL gl_FrontLightModelProduct.sceneColor variable wasn't defined
- Fix crash upon GLSL variable array indexes (not yet supported)
- Two-sided stencil test didn't work in software rendering
- Fix two-sided lighting bugs/crashes (bug 13368)
- GLSL gl_FrontFacing didn't work properly
- glGetActiveUniform returned incorrect sizes (bug 13751)
- Fix several bugs relating to uniforms and attributes in GLSL API (Bruce Merry, bug 13753)
- glTexImage3D(GL_PROXY_TEXTURE_3D) mis-set teximage depth field
- Fixed GLX indirect vertex array rendering bug (14197)

- Fixed crash when deleting framebuffer objects (bugs 13507, 14293)
- User-defined clip planes enabled for R300 (bug 9871)
- Fixed glBindTexture() crash upon bad target (bug 14514)
- Fixed potential crash in glDrawPixels(GL_DEPTH_COMPONENT) (bug 13915)
- Bad strings given to glProgramStringARB() didn't generate GL_INVALID_OPERATION
- Fixed minor point rasterization regression (bug 11016)
- state.texenv.color state var didn't work in GL_ARB_fragment_program (bug 14931)
- glBitmap from a PBO didn't always work
- glGetTexImage into a PBO didn't always work
- Comments at the end of ARB vertex/fragment programs crashed the parser

4.271.3 Changes

- Updated glext.h to version 40

4.271.4 Driver Status

```
Driver         Status
---------------------  ----------------------
DRI drivers     varies with the driver
XMesa/GLX (on Xlib) implements OpenGL 2.1
OSMesa (off-screen) implements OpenGL 2.1
Windows/Win32       implements OpenGL 2.1
Glide (3dfx Voodoo1/2)  implements OpenGL 1.3
SVGA            unsupported
Wind River UGL      unsupported
DJGPP           unsupported
GGI         unsupported
BeOS            unsupported
Allegro         unsupported
D3D         unsupported
```

4.272 Mesa 7.0.2 Release Notes / November 10, 2007

Mesa 7.0.2 is a stable release with bug fixes since version 7.0.

4.272.1 MD5 checksums

```
c9cf607f36e7e50172f5f9c7d552c34e  MesaLib-7.0.2.tar.gz
93e6ed7924ff069a4f883b4fce5349dc  MesaLib-7.0.2.tar.bz2
10c324c3613f90f059cb8429f700f300  MesaLib-7.0.2.zip
aa8b1244a5de1d23e5814bf9b67f1435  MesaDemos-7.0.2.tar.gz
11a10410bae7be85cf25bc7119966468  MesaDemos-7.0.2.tar.bz2
1dd0b5fd6d69430a2fd76a6adbfd8fff  MesaDemos-7.0.2.zip
a7dbf25c025955858bd2d89a6eb6db4c  MesaGLUT-7.0.2.tar.gz
```

(continues on next page)

(continued from previous page)

```
3a33f8efc8c58a592a854cfc7a643286  MesaGLUT-7.0.2.tar.bz2
eba4ef2aa8c362ead81b54357f1903a3  MesaGLUT-7.0.2.zip
```

4.272.2 New features

- Updated Windows VC7 project files
- Added DESTDIR variable for 'make install'
- Added pkg-config files for gl, glu, glut and glw libraries
- Added bluegene-xlc-osmesa and catamount-osmesa-pgi configs
- Support for Intel G33/Q33/Q35 graphics chipsets

4.272.3 Bug fixes

- Fixed a vertex buffer wrapping issue (bug 9962)
- Added mutex protection around texture object reference counters
- Added checking/support for additional chips in the i915/i945 family (see 11978)
- Fixed a blending/banding issue (bug 11931)
- Fixed a GLU matrix inversion bug (#6748)
- Fixed problem with large glDrawArrays calls and indirect rendering (bug 12141)
- Fixed an assortment of i965 driver bugs
- Fixed x86-64 vertex transformation bug (12216)
- Fixed X server crash caused by multiple indirect rendering clients
- Parsing of state.texgen in ARB vertex/fragment programs didn't work (bug 12313)
- Fixed a glCopyPixels/glPixelZoom bug (12417)
- Fixed a bug when using glMaterial in display lists (bug 10604)
- Fixed a few GLUT/Fortran issues (Bill Mitchell)
- Fixed Blender crash bug (12164)
- Fixed some issues preventing cross-compiling
- Fixed up broken GL_ATI_separate_stencil extension
- glDrawArrays(count=0) led to a crash
- Fix SSE code gen memory leak, possible crash
- Fixed MMX 565 rgb conversion problem (bug 12614)
- Added -fno-strict-aliasing and -fPIC flags for gcc
- Fixed Blender crash in Unichrome driver (bug 13142)

4.272.4 Driver Status

```
Driver         Status
----------------------  ----------------------
DRI drivers     varies with the driver
XMesa/GLX (on Xlib) implements OpenGL 2.1
OSMesa (off-screen) implements OpenGL 2.1
Windows/Win32       implements OpenGL 2.1
Glide (3dfx Voodoo1/2)  implements OpenGL 1.3
SVGA            unsupported
Wind River UGL      unsupported
DJGPP           unsupported
GGI         unsupported
BeOS            unsupported
Allegro         unsupported
D3D         unsupported
```

4.273 Mesa 7.0.1 Release Notes / August 3, 2007

Mesa 7.0.1 is a stable release with bug fixes since version 7.0.

4.273.1 MD5 checksums

```
db55141a44b902fcc61d9265b7862c06  MesaLib-7.0.1.tar.gz
c056abd763e899114bf745c9eedbf9ad  MesaLib-7.0.1.tar.bz2
ecc2637547fae2b38271ae362d013afa  MesaLib-7.0.1.zip
b85a4a5be4e829f4a1165e4514b13183  MesaDemos-7.0.1.tar.gz
3b66b3268df12ca8a6c4e0c4c457912c  MesaDemos-7.0.1.tar.bz2
b1c18006f16e44e80fea66774c59b391  MesaDemos-7.0.1.zip
b87a69986839ae43ce12fc8e3dc1ebb4  MesaGLUT-7.0.1.tar.gz
25f30d0c1651997b4412366ba0572f7f  MesaGLUT-7.0.1.tar.bz2
676ee6682a6ce78a5540554fd975c03e  MesaGLUT-7.0.1.zip
```

4.273.2 New features

- Added a bluegene-osmesa build config

4.273.3 Bug fixes

- Fixed some MingW build issues
- Added a few missing OpenGL 2.0 API entrypoints:
  - glVertexAttrib4bv
  - glVertexAttrib4iv
  - glVertexAttrib4ubv
  - glVertexAttrib4uiv
  - glVertexAttrib4usv

- Fixed glDrawPixels(GL_STENCIL_INDEX) pixel transfer bug 11457
- GLSL bug fix: added vec2(vec4) constructor
- GLSL bug fix: .strq and .rgba writemasks didn't always work
- Stencil pixel map didn't always work for glDrawPixels (bug 11475)
- Fixed polygon stipple bug in i915 driver
- Binding a zero-sized texture didn't disable texturing (bug 11309)
- Queries of GL_INFO_LOG_LENGTH, GL_SHADER_SOURCE_LENGTH didn't include the terminating zero (bug 11588)
- glXChooseFBConfig() in Xlib driver didn't handle GLX_STEREO flag properly
- Fixed a GLSL function call bug (#11731)
- glPointParameteriv(GL_DISTANCE_ATTENUATION_EXT) didn't work (bug 11754)
- glGetAttribLocation() always returned 1 (bug 11774)
- Fixed a few memory-related bugs in GLU library

4.273.4 Changes

- The libOSMesa library version has been reverted to 6.5.3 (soname=6) in order to avoid application linking issues. Otherwise, applications previously linked with libOSMesa.so.6 would no longer link with libOSMesa.so.7
- Dropped obsolete, unmaintained Windows project files for VC6 and VC7.

4.273.5 To Do (someday) items

- Switch to freeglut
- Fix linux-glide target/driver.
- Improved lambda and derivative calculation for frag progs.

4.273.6 Driver Status

```
Driver         Status
----------------------  ----------------------
DRI drivers     varies with the driver
XMesa/GLX (on Xlib) implements OpenGL 2.1
OSMesa (off-screen) implements OpenGL 2.1
Windows/Win32       implements OpenGL 2.1
Glide (3dfx Voodoo1/2)  implements OpenGL 1.3
SVGA            unsupported
Wind River UGL      unsupported
DJGPP           unsupported
GGI         unsupported
BeOS            unsupported
Allegro         unsupported
D3D         unsupported
```

4.274 Mesa 7.0 Release Notes / June 22, 2007

Mesa 7.0 is a stable release, featuring OpenGL 2.1 API support. A number of bugs have been fixed since the 6.5.3 release.

4.274.1 MD5 checksums

```
35a1698986f7ac8dc435624ee9256cda  MesaLib-7.0.tar.gz
50c371455fa7532c04aa0a970f9bc51f  MesaLib-7.0.tar.bz2
bcedb6f43c97c1bc49e5cc7f12835722  MesaLib-7.0.zip
9bad332c7b74f59be96556135212ca9e  MesaDemos-7.0.tar.gz
fada2bc1f29da513e015fda1e3abd0c0  MesaDemos-7.0.tar.bz2
84e3bbe470d983ae32f1f0c779faf99e  MesaDemos-7.0.zip
76c7bb54f9850c689eba844f6daed332  MesaGLUT-7.0.tar.gz
4af28296e02772ef1de00e4e79bf3d12  MesaGLUT-7.0.tar.bz2
9043cb0b54cc03d1874728d74b12188c  MesaGLUT-7.0.zip
```

4.274.2 New features

- OpenGL 2.0 and 2.1 API support.

4.274.3 Bug fixes

- Fixed a few fog-related bugs.
- Fixed broken GLSL mix() function.
- Fixed broken GLSL exp() functions.
- Fixed GLSL mod4(vec4, vec4) bug.
- Implemented GLSL asin(), acos(), atan() functions.
- Fixed an R300 driver bug that caused Xorg composite manager to crash
- Fixed R300 vertex program/matrix bug (10848)
- GLSL dFdx() and dFdy() work for fragment program inputs now (texcoords)
- Specifying an invalid texture unit as a sampler could lead to a crash
- The GLX protocol request for glXDestroyPBuffer() was incorrect (bug 10983)
- ARB vp state.light[n].half value was incorrect (bug 10987)
- Fixed a positional light source bug (bug 11009)
- Fixed point size attenuation problem (bug 11042)
- glPopAttrib didn't restore texture object's LOD bias (bug 11049)
- Fixed a TLS / TEXTREL problem (bug 7459)

4.274.4 Internal code changes

- Some texture code consolidation and simplifiction (Ian Romanick)
- R300 driver clean-ups.

4.274.5 To Do (someday) items

- Switch to freeglut
- Fix linux-glide target/driver.
- Improved lambda and derivative calculation for frag progs.

4.274.6 Driver Status

```
Driver         Status
----------------------  ----------------------
DRI drivers     varies with the driver
XMesa/GLX (on Xlib) implements OpenGL 2.1
OSMesa (off-screen) implements OpenGL 2.1
Windows/Win32       implements OpenGL 2.1
Glide (3dfx Voodoo1/2)  implements OpenGL 1.3
SVGA            unsupported
Wind River UGL      unsupported
DJGPP           unsupported
GGI         unsupported
BeOS            unsupported
Allegro         unsupported
D3D         unsupported
```

4.275 Mesa 6.5.3 Release Notes / April 27, 2007

Mesa 6.5.3 is a development release with many changes and new features. Mesa 7.0 is expected to follow shortly.

4.275.1 MD5 checksums

```
39f33ea64e34e2d5b20640b008b57649  MesaLib-6.5.3.tar.gz
46359457147c469745f24b5074a186f0  MesaLib-6.5.3.tar.bz2
a8946fa861634ce15971396f47992c41  MesaLib-6.5.3.zip
08e26948d57eaca74d02a530b2d8106e  MesaDemos-6.5.3.tar.gz
8af91773ab2653fe537499676b05f2e8  MesaDemos-6.5.3.tar.bz2
783f81b171bf89b0929abc894efd25a6  MesaDemos-6.5.3.zip
9467d415388fe1ad82991fb20704b812  MesaGLUT-6.5.3.tar.gz
360843e46b7ebb6909290b023f9b26fa  MesaGLUT-6.5.3.tar.bz2
7686065e5c15a30de08a1610860b6840  MesaGLUT-6.5.3.zip
```

4.275.2 Shared library numbering

Mesa 6.5.3 supports the OpenGL 2.0/2.1 API. However, the (unix) shared library version is still 1.5 (i.e. libGL.so.1.5.xxxxxx). Bumping the shared library version to 2.x would cause linking problems with existing OpenGL

applications. Since OpenGL 2.x is backward compatible with OpenGL 1.x the shared library version number doesn't have to be incremented (which would indicate an incompatible ABI).

Other OpenGL vendors name their OpenGL 2.x libraries libGL.so.1.0.xxxxx for the same reason.

4.275.3 New features

- OpenGL 2.0 and 2.1 API support.
- Entirely new Shading Language code generator. See the *Shading Language* page for more information.
- Much faster software execution of vertex, fragment shaders.
- New vertex buffer object (vbo) infrastructure
- Updated glext.h file (version 39)
- Updated glxext.h file (version 19)
- GL_MAX_DRAWBUFFERS is now 4 (software rendering) so "multiple render targets" are really supported.

4.275.4 Bug fixes

- Fog was errantly applied when a fragment shader was enabled (bug 9346)
- glPush/PopClientAttrib didn't handle VBO bindings correctly (bug 9445)
- With 32-bit Z buffer, the fragment Z of lines and points was sometimes wrong.
- GL_POST_CONVOLUTION_ALPHA_BIAS/SCALE was broken.
- 1D convolution state could effect 2D image transfers
- Overlapping glCopyPixels with negative Y zoom didn't work (bug 10521)
- Fixed a number of framebuffer/renderbuffer reference counting bugs
- Fixed a few bugs in software-emulated alpha planes
- Assorted minor bug fixes in glCopy/DrawPixels, glPixelZoom, etc.
- Assorted DRI driver bug fixes.
- Fixed a number of bugs that prevented "depth-peeling" rendering from working.

4.275.5 Internal code changes

- Old array_cache module replaced by new vbo module. All geometry rendering is now cast in the form of vertex buffer objects.
- Massive changes to the Shading Language compiler and related state.
- Vertex/fragment shaders are compiled into GPU instructions and programs very similar to GL_ARB_vertex/fragment_program.
- Vertex and fragment programs are executed with the same code now.
- The SSE-optimized vertex program path has been removed since it didn't support more than 12 temp registers, didn't support branching/looping, etc.

4.275.6 To Do (someday) items

- Switch to freeglut
- Fix linux-glide target/driver.
- Improved lambda and derivative calculation for frag progs.

4.275.7 Driver Status

```
Driver         Status
----------------------  ----------------------
DRI drivers     varies with the driver
XMesa/GLX (on Xlib) implements OpenGL 2.1
OSMesa (off-screen) implements OpenGL 2.1
Windows/Win32       implements OpenGL 2.1
Glide (3dfx Voodoo1/2)  implements OpenGL 1.3
SVGA            unsupported
Wind River UGL      unsupported
DJGPP           unsupported
GGI         unsupported
BeOS            unsupported
Allegro         unsupported
D3D         unsupported
```

4.276 Mesa 6.5.2 Release Notes / December 2, 2006

Mesa 6.5.2 is a 6.5 follow-on development release with a few new features but mostly consisting of bug fixes.

4.276.1 MD5 checksums

```
11a033b078e090b3caaeb467234fe299  MesaLib-6.5.2.tar.gz
e4d894181f1859651658b3704633e10d  MesaLib-6.5.2.tar.bz2
63bf1d444fa738cca52ce1043e284021  MesaLib-6.5.2.zip
2b8f1375d16bda5f5a2304174cd5bcf7  MesaDemos-6.5.2.tar.gz
e870efe98d3a50be01ab211b9b2e25d9  MesaDemos-6.5.2.tar.bz2
d92cc6f5fee5ca75af0be04f9f4908f0  MesaDemos-6.5.2.zip
8d4d77e3a7132f4217bbc7c1ab157030  MesaGLUT-6.5.2.tar.gz
e84edbb11c69c8e408dfadd2ed08e95b  MesaGLUT-6.5.2.tar.bz2
c6d7134843ed5faf11f6686ecb5d2a2e  MesaGLUT-6.5.2.zip
```

4.276.2 New features

- New DRI memory manager system. Currently used by the i915tex driver. Other DRI drivers will be updated to use the new memory manager in coming months.

  To use the new driver you'll need the most recent DRM library and drivers (version 2.2 or later) and a recent xf86-video-intel driver module from X.org.

  New features resulting from this work include:

  - EXT_framebuffer_objects, render to texture

  - ARB_pixel_buffer_objects
  - Accelerated CopyTexSubimage, DrawPixels, ReadPixels, CopyPixels
  - Accelerated texture uploads from pixel buffer objects
  - Potentially texturing directly from the pixel buffer object (zero copy texturing).
- New Intel i965 DRI driver
- New `minstall` script to replace normal install program
- Faster fragment program execution in software
- Added (or fixed) support for GLX_SGI_make_current_read to the following drivers:
  - radeon
  - savage
  - mga
  - tdfx
- Added support for ARB_occlusion_query to the tdfx driver (Ian Romanick).

4.276.3 Bug fixes

- fixed invalid memory read while rendering textured points (bug 8320)
- fixed problems with freebsd-dri configuration (bug 8344)
- Mesa's fake glxGetCurrentContext() wasn't thread-aware
- OPTION NV_position_invariant didn't work in NV vertex programs
- glDrawPixels into a user-created framebuffer object could crash Xlib driver
- Line clipping was broken in some circumstances
- fragment.fogcoord register didn't always contain the correct value
- RGBA logicops didn't work reliably in some DRI drivers
- Fixed broken RGBA LogicOps in Intel DRI drivers
- Fixed some fragment program bugs in Intel i915 DRI driver
- Fixed glGetVertexAttribfvARB bug 8883
- Implemented glGetUniform[fi]vARB() functions
- Fixed glDrawPixels(GL_COLOR_INDEX, GL_BITMAP) segfault (bug 9044)
- Fixed some gluBuild2DMipmaps() bugs (Greg McGarragh)
- Fixed broken "mgl" name mangling
- Indirect rending was broken for glMap* functions (bug 8899)

4.276.4 Internal code changes

- The device driver functions ResizeBuffers and GetBufferSize have been decprecated.
- OpenGL 2.0 and 2.1 support is nearly done. We need to do quite a bit more testing of the shading language functions.

4.276.5 To Do (someday) items

- Switch to freeglut
- Increase MAX_DRAWBUFFERS
- Fix linux-glide target/driver.
- Improved lambda and derivative calculation for frag progs.

4.276.6 Driver Status

```
Driver         Status
----------------------  ----------------------
DRI drivers     varies with the driver
XMesa/GLX (on Xlib) implements OpenGL 1.5
OSMesa (off-screen) implements OpenGL 1.5
Glide (3dfx Voodoo1/2)  implements OpenGL 1.3
SVGA            implements OpenGL 1.3
Wind River UGL      implements OpenGL 1.3
Windows/Win32       implements OpenGL 1.5
DJGPP           implements OpenGL 1.5
GGI         implements OpenGL 1.3
BeOS            implements OpenGL 1.5
Allegro         needs updating
D3D         needs updating
```

4.277 Mesa 6.5.1 Release Notes / September 15, 2006

Mesa 6.5.1 is a 6.5 follow-on development release mostly consisting of bug fixes.

4.277.1 MD5 checksums

```
d9a555297319bb932a3192952d53d073  MesaLib-6.5.1.tar.gz
c46f2c6646a270911b791dd8e1c2d977  MesaLib-6.5.1.tar.bz2
939eaaff33322bfeafac784402b45f4f  MesaLib-6.5.1.zip
9e4bbe83c007bfbaa67449a81cc3d36a  MesaDemos-6.5.1.tar.gz
0f2794baf7a9d98b22caea9f78c6942d  MesaDemos-6.5.1.tar.bz2
14c77eab9cc7a265c331abf239927c1c  MesaDemos-6.5.1.zip
c5f87c23aaf4eaf1bda0d007ea98366c  MesaGLUT-6.5.1.tar.gz
2525642fe7f454e3e1a1aad01359b406  MesaGLUT-6.5.1.tar.bz2
e33b165c22551e23b58ede8767378543  MesaGLUT-6.5.1.zip
```

4.277.2 New Features

- Intel i965 “broadwater” DRI driver
- GL_APPLE_vertex_array_object - allows encapsulation of a set of vertex arrays in an object.
- GL_EXT_texture_sRGB - non-linearly mapped texture formats
- GL_EXT_gpu_program_parameters - addes a few new functions for setting multiple vertex/fragment program parameters with one call.

- "engine" demo
- updated fbdev driver and GLUT for fbdev (Sean D'Epagnier)
- many updates to the DRI drivers

4.277.3 Changes

- The glVertexAttribARB functions no longer alias the conventional vertex attributes.
- glxinfo program prints more info with -l option
- GL_FRAGMENT_PROGRAM_NV and GL_FRAGMENT_PROGRAM_ARB are now compatible, in terms of glBindProgramARB()
- The GL_ARB_vertex_program attribute `vertex.weight` is now accepted by the parser, even though the GL_ARB_vertex_blend and GL_EXT_vertex_weighting extensions aren't supported. Allows Warcraft to run.

4.277.4 Bug fixes

- fixed broken texture border handling for depth textures (bug 6498)
- removed the test for duplicated framebuffer attachments, per version 117 of the GL_EXT_framebuffer_object specification
- fixed a few render-to-texture bugs, including render to depth texture
- clipping of lines against user-defined clip planes was broken (6512)
- assembly language dispatch for SPARC was broken (bug 6484)
- assorted compilation fixes on various Unix platforms (Dan Schikore)
- glPopAttrib could restore an invalid value for GL_DRAW_BUFFER
- assorted minor fixes for 16 and 32 bit/channel modes
- fixed assorted bugs in texture compression paths
- fixed indirect rendering vertex array crashes (bug 6863)
- glDrawPixels GL_INDEX_OFFSET didn't always work
- fixed convolution memory leak (bug 7077)
- rectangular depth textures didn't work
- invalid mode to glBegin didn't generate an error (bug 7142)
- 'normalized' parameter to glVertexAttribPointerARB didn't work
- disable bogus GLX_SGI_video_sync extension in xlib driver
- fixed R128 driver locking bug (Martijn van Oosterhout)
- using evaluators with vertex programs caused crashes (bug 7564)
- fragment.position wasn't set correctly for point/line primitives
- fixed parser bug for scalar sources for GL_NV_fragment_program
- max fragment program length was incorrectly 128, now 1024
- writes to result.depth in fragment programs weren't clamped to [0,1]
- fixed potential dangling pointer bug in glBindProgram()

- fixed some memory leaks (and potential crashes) in Xlib driver
- fixed a number of build issues on HP-UX (Christopher Bell)
- accum buffer didn't work with OSMesa interface

4.277.5 Internal code changes

A number of Mesa program-related structs were renamed. For example *struct vertex_program* is now *struct gl_vertex_program*. All the effected drivers have been updated.

Ian Romanick updated the GL API dispatch code in a number of ways. First, many old/unused extensions were removed. Second, the static entrypoints for some extensions were removed. This means GL function pointers will have to be used more often (e.g. use glXGetProcAddressARB()).

4.277.6 To Do (someday) items

- Switch to freeglut
- Increase MAX_DRAWBUFFERS
- Fix linux-glide target/driver.
- Fix lambda calculation for frag progs.

4.277.7 Driver Status

```
Driver         Status
----------------------  ----------------------
DRI drivers     varies with the driver
XMesa/GLX (on Xlib) implements OpenGL 1.5
OSMesa (off-screen) implements OpenGL 1.5
Glide (3dfx Voodoo1/2)  implements OpenGL 1.3
SVGA            implements OpenGL 1.3
Wind River UGL      implements OpenGL 1.3
Windows/Win32       implements OpenGL 1.5
DJGPP           implements OpenGL 1.5
GGI         implements OpenGL 1.3
BeOS            implements OpenGL 1.5
Allegro         needs updating
D3D         needs updating
```

4.278 Mesa 6.5 Release Notes / March 31, 2006

Mesa 6.5 is a new development release.

4.278.1 MD5 checksums

```
657be3b92f6dabc78a67ed9cb8d67813  MesaLib-6.5.tar.gz
61beda590bfc5b4a12e979d5f2d70d7a  MesaLib-6.5.tar.bz2
19d48b872d579d4f91466060804a59ac  MesaLib-6.5.zip
```

(continues on next page)

(continued from previous page)

```
694ad3a7007010c7418a9c72d1cba5b7  MesaDemos-6.5.tar.gz
ab95b590dcd640726a2d89e62068c66e  MesaDemos-6.5.tar.bz2
b792c303fefd87294488e2b7eab976e5  MesaDemos-6.5.zip
ac1d585483617db0c91e5c15cb5ec3a3  MesaGLUT-6.5.tar.gz
59f0bf2b2ffb67fe23ee479f9b044f31  MesaGLUT-6.5.tar.bz2
005decb2136718e22222ac1c4805cd15  MesaGLUT-6.5.zip
```

4.278.2 New Features

- OpenGL Shading language support

  This includes the GL_ARB_shader_objects, GL_ARB_shading_language_100, GL_ARB_vertex_shader and GL_ARB_fragment_shader extensions. Most of the work was done by Michal Krol. There's probably a fair number of bugs since this is a pretty large, complicated body of code.

  The OpenGL 2.0 interface to these features will be implemented in a future version of Mesa,

- GL_EXT_timer_query

  Used to measure the time of OpenGL operations at high precision. Only supported in the software/Xlib driver at this time.

- GL_EXT_packed_depth_stencil

  Defines a new GL_DEPTH_STENCIL_EXT pixel format.

- GL_EXT_framebuffer_blit

  A simplified glCopyPixels-like feature for copying pixel rectangles.

- GL_ARB_half_float_pixel

  Adds a new half-precision floating point format for image transfers, such as for glDrawPixels, glReadPixels, glTexImage, etc.

4.278.3 Changes

- removed GL_HP_occlusion_test (use GL_ARB_occlusion_query instead)
- removed GL_SGIX/SGIS_pixel_texture extensions

4.278.4 Bug fixes

- fixed glxcontextmodes.c datatype problem (bug 5835)
- fixed aix-gcc build/install bugs (bug 5874)
- fixed some bugs in texture env program generation
- glXCopyContext() didn't handle texture object bindings properly
- glXCopyContext() didn't copy all lighting state
- fixed FreeBSD config (Pedro Giffuni)
- fixed some minor framebuffer object bugs
- replaced dprintf() with _glu_printf() in GLU (bug 6244)
- fixed a number of thread safety bugs/regressions

- fixed a number of GLU tesselator bugs (John Shell, bug 6339)
- paletted texturing was broken w/ floating point palettes (K. Schultz)
- lots of assorted framebuffer object bug fixes

4.278.5 Known Issues

- Rendering to depth textures will not work. Rendering to GL_DEPTH_STENCIL textures should work.

4.278.6 Driver Interface Changes

- Stencil: The Driver.StencilOp/Func/Mask() functions have been replaced by the two-sided versions: Driver.Stencil*Separate().
- Render-to-texture: The functions for rendering to textures have changed.

4.278.7 To Do (someday) items

- Switch to freeglut
- Increase MAX_DRAWBUFFERS
- Fix linux-glide target/driver.
- Fix lambda calculation for frag progs.

4.278.8 Driver Status

```
Driver         Status
----------------------  ----------------------
DRI drivers     varies with the driver
XMesa/GLX (on Xlib) implements OpenGL 1.5
OSMesa (off-screen) implements OpenGL 1.5
Glide (3dfx Voodoo1/2)  implements OpenGL 1.3
SVGA            implements OpenGL 1.3
Wind River UGL      implements OpenGL 1.3
Windows/Win32       implements OpenGL 1.5
DJGPP           implements OpenGL 1.5
GGI         implements OpenGL 1.3
BeOS            implements OpenGL 1.5
Allegro         needs updating
D3D         needs updating
```

4.279 Mesa 6.4.2 / February 2, 2006

Mesa 6.4.2 is a stable, bug-fix release.

4.279.1 MD5 checksums

```
cb0d745d520fa7c2bb9178058b763544  MesaLib-6.4.2.tar.gz
7674d2c603b5834259e4e5a820cefd5b  MesaLib-6.4.2.tar.bz2
d224e1325b33ff71a0f3893fc6b4d594  MesaLib-6.4.2.zip
d4b345d4588fc750cd3d34f3ac26673e  MesaDemos-6.4.2.tar.gz
9cae1ab874af533ce356bd7dfe2e0bb0  MesaDemos-6.4.2.tar.bz2
2da6e1d1245e441d27813595c6ba50de  MesaDemos-6.4.2.zip
84427d18c3453f0ea52388eeba7169b5  MesaGLUT-6.4.2.tar.gz
b157ba8ad1ea63260cf5339132e7aac6  MesaGLUT-6.4.2.tar.bz2
fe1523744fc05edc3811dfc6a1bf4181  MesaGLUT-6.4.2.zip
```

4.279.2 New features

- added OSMesaColorClamp() function/feature
- added wglGetExtensionStringARB() function

4.279.3 Changes

- GLUT tarball: Starting with 6.4, the GLUT library sources are distributed in a separate tarball. This was done at the request of Linux distro vendors who prefer to use freeglut.

4.279.4 Bug fixes

- fixed some problems when building on Windows
- GLw header files weren't installed by installmesa script (bug 5396)
- GL/glfbdev.h file was missing from tarballs
- fixed TNL initialization bug which could lead to crash (bug 5791)

4.279.5 Driver Status

```
Driver         Status
----------------------  ----------------------
DRI drivers     varies with the driver
XMesa (Xlib)        implements OpenGL 1.5
OSMesa (off-screen) implements OpenGL 1.5
Windows/Win32       implements OpenGL 1.5
Glide (3dfx Voodoo1/2)  requires updates
SVGA            requires updates
DJGPP           requires updates
GGI         requires updates
BeOS            requires updates
Allegro         requires updates
D3D         requires updates
```

4.280 Mesa 6.4.1 / November 29, 2006

Mesa 6.4.1 is a stable, bug-fix release.

4.280.1 MD5 checksums

```
698ceb574cf882b0226761f5913c0da9  MesaLib-6.4.1.tar.gz
ea148c828ec6f645526451db1b8556f1  MesaLib-6.4.1.tar.bz2
42e93279468975ed2bf3111b8721e5d9  MesaLib-6.4.1.zip
e3b0d50807fd2bdcd1a95aaddd786f13  MesaDemos-6.4.1.tar.gz
99df1fdcb98d391666b476ca6f1dda8a  MesaDemos-6.4.1.tar.bz2
b999d2c6d92fb4b7740a3dbd889348e3  MesaDemos-6.4.1.zip
eadfe01fe5ddfb1eb8227dd567b31635  MesaGLUT-6.4.1.tar.gz
bd003bb4f981a4f91dee4c38644d4f3f  MesaGLUT-6.4.1.tar.bz2
71c401c037088bf688a88afdaeb3420f  MesaGLUT-6.4.1.zip
```

4.280.2 Bug fixes

- redefining a vertex program string didn't take effect in TNL module
- fixed occasional segfault upon vertex/fragment parsing error
- vertex program LIT instruction didn't handle 0^0=1 correctly
- fragment program fog option didn't work with glDrawPixels, glBitmap
- USE_MGL_NAMESPACE didn't work for x86-64
- OSMesa demos were missing from previous release tarballs
- fixed problem with float->ushort conversion in glClear (bug 4992)
- popping of GL_EYE_PLANE texgen state was broken (bug 4996)
- popping of GL_SPOT_DIRECTION light state was broken (bug 5005)
- fixed occasional triangle color interpolation problem on VMS
- work around invalid free() call (bug 5131)
- fixed BSD X server compilation problem by including stdint.h

4.280.3 Driver Status

```
Driver         Status
----------------------  ----------------------
DRI drivers     varies with the driver
XMesa (Xlib)        implements OpenGL 1.5
OSMesa (off-screen) implements OpenGL 1.5
Windows/Win32       implements OpenGL 1.5
Glide (3dfx Voodoo1/2)  requires updates
SVGA            requires updates
DJGPP           requires updates
GGI         requires updates
BeOS            requires updates
Allegro         requires updates
D3D         requires updates
```

4.281 Mesa 6.4 / October 24, 2005

Mesa 6.4 is a stable, bug-fix release.

4.281.1 MD5 checksums

```
1cce0c1eb4fd15e9dfe837a1ce0c9812  MesaLib-6.4.tar.gz
85a84e47a3f718f752f306b9e0954ef6  MesaLib-6.4.tar.bz2
b976fea4f3ee06354c53f91b6e3f2ffc  MesaLib-6.4.zip
d8734f2c69bcf7ef9f5ae454a85743ba  MesaDemos-6.4.tar.gz
1a8c4d4fc699233f5fdb902b8753099e  MesaDemos-6.4.tar.bz2
607ab7c7a7de0cc5febbdde2bfa03098  MesaDemos-6.4.zip
3260156f66174322a092be0767962d34  MesaGLUT-6.4.tar.gz
0465d053f83775f44a12dec4050dfd78  MesaGLUT-6.4.tar.bz2
02abfcdcdf72ba938ae00f6e3b70fbe0  MesaGLUT-6.4.zip
```

4.281.2 New

- Added a fast XOR line drawing function in Xlib driver
- Added support for GL_ARB_texture_mirrored_repeat to savage driver (supported only on Savage4 hardware).

4.281.3 Changes

- Mesa now packaged in three parts: Library, Demos and GLUT

4.281.4 Bug fixes

- GLX_X_RENDERABLE token wasn't accepted by glXChooseFBConfig
- Some files were present multiple times in the 6.3.2 tarballs
- r200_vtxtmp_x86.S file was missing from 6.3.2 tarball (bug 4207)
- glxgears_fbconfig demo didn't work (bug 4237)
- fixed bug when bilinear sampling 2d textures with borders
- glXCreatePbuffer() could segfault instead of returning 0 (bug 4235)
- fixed undefined frexp and rand in X.org libGLcore.a (bug 4242)
- fixed a few problems with proxy color tables (bug 4270)
- fixed precision problem in Z clearing (bug 4395)
- glBitmap, glDraw/CopyPixels mistakenly generated selection hits
- fixed potential segfault caused by reading pixels outside of renderbuffer bounds
- glGetTexLevelParameter didn't accept GL_TEXTURE_DEPTH_SIZE_ARB
- fixed memory corruption bug involving software alpha buffers
- glReadPixels clipped by window bounds was sometimes broken
- glDraw/CopyPixels of stencil data ignored the stencil write mask

- glReadPixels from a texture bound to a framebuffer object didn't work
- glIsRender/FramebufferEXT weren't totally correct
- fixed a number of point size attenuation/fade bugs
- fixed glFogCoord bug 4729
- GLX encoding for transpose matrix functions was broken
- fixed broken fragment program KIL and SWZ instructions
- fragment programs that wrote result.depth.z didn't work

4.281.5 Driver Status

```
Driver         Status
----------------------  ----------------------
DRI drivers     varies with the driver
XMesa (Xlib)        implements OpenGL 1.5
OSMesa (off-screen) implements OpenGL 1.5
Windows/Win32       implements OpenGL 1.5
Glide (3dfx Voodoo1/2)  requires updates
SVGA            requires updates
DJGPP           requires updates
GGI         requires updates
BeOS            requires updates
Allegro         requires updates
D3D         requires updates
```

4.282 Mesa Version History

Note: Changes for Mesa 6.4 and later are documented in the corresponding *release notes* file.

4.282.1 1.0 beta February 1995

- Initial release

4.282.2 1.1 beta March 4, 1995

Changes:

- faster point and line drawing (2x faster)
- more systems supported, better Makefiles
- Renamed lib*.a files to avoid collisions
- many small bug fixes

New:

- pseudo-GLX functions added
- new implementation of evaluators (eval2.c)

- GLUT support

4.282.3 1.1.1 beta March 7, 1995

Changes:

- Reverted from eval2.c to eval.c due to FPE on Linux
- more speed improvements
- more Makefile changes

4.282.4 1.1.2 beta March 14, 1995

New:

- implementation of SGI's blending extensions
- glXUseXFont implemented
- added MESA_DEBUG environment variable support

Changes:

- Using eval2.c again
- more FPE-prevention checks (0-length normals are OK)
- a few small bug fixes
- much faster pixel logic ops!
- faster transformation arithmetic

4.282.5 1.1.3 beta March 31, 1995

New:

- gluScaleImage() and gluBuild2DMipMaps() implemented
- Mesa widgets for Xt/Motif
- blendEXT demos
- added environment variables for selecting visuals

Changes:

- almost all GLUT demos work correctly now
- faster X device driver functions
- more bug fixes

4.282.6 1.1.4 beta April 20, 1995

Bug fixes:

- missing #define SEEK_SET in src-tk/image.c
- compile glShadeModel into display lists

- fixed pow() domain error in src/light.c
- fixed "flickering bitmaps" in double buffer mode
- fixed tk.h and aux.h for C++
- state of LIGHT_MODEL_LOCAL_VIEWER was inverted

New features:

- MUCH, MUCH nicer dithering in 8-bit RGB mode
- updated widgets and widget demos
- Implemented GLXPixmap functions
- Added GLU 1.1 and GLX 1.1 functions
- Changed the X/Mesa interface API, more versatile
- Implemented gluPartialDisk()

4.282.7 1.2 May 22, 1995

Bug fixes:

- IRIX 4.x makefile problem
- modified tk to share root colormap as needed
- gluLookAt normalization problem
- suppress Expose, NoExpose events in swapbuffers
- glBitmap() and glDrawPixels() clipping

New features:

- GL_BLEND, GL_MODULATE, GL_DECAL, and GL_REPLACE_EXT texture modes implemented
- texture maps stored more efficiently
- texture maps can be compiled into display lists
- Bogdan Sikorski's GLU polygon tesselation code
- Linas Vepstas's sweep and extrusion library
- glXCreateContext()'s shareList parameter works as it's supposed to. XMesaCreateContext() updated to accept a shareList parameter too.
- Mesa can be compiled with real OpenGL .h files
- MESA_BACK_BUFFER environment variable
- better GLX error checking

4.282.8 1.2.1 June 22, 1995

Bug fixes:

- X/Mesa double buffer window resize crash
- widgets now pass PointerMotion events
- X/Mesa incorrect default clear color and drawing color

- more robust X MIT-SHM support in X/Mesa
- glTexImage(format=GL_LUMINANCE) didn't work
- GL_LINE mode polygons with line width > 1.0 could cause a crash
- numerous feedback bugs
- glReadPixels() from depth buffer was wrong
- error prone depth and stencil buffer allocation New features:
- Preliminary Microsoft Windows driver
- Implemented a number of missing functions: glEvalCoord[12][df]v(), glGet. . . (), etc.
- Added a few missing symbols to gl.h and glu.h
- Faster rendering of smooth-shaded, RGBA, depth-buffered polygons.
- Faster rendering of lines when width=2.0
- Stencil-related functions now work in display lists

Changes:

- renamed aux.h as glaux.h (MS-DOS names can't start with aux)
- most filenames are in 8.3 format to accommodate MS-DOS
- use GLubytes to store arrays of colors instead of GLints

4.282.9 1.2.2 August 2, 1995

New features:

- texture mapped points and lines
- NURBS! (but not 100% complete)
- viewports may safely extend beyond window boundaries
- MESA_PRIVATE_CMAP environment variable
- Grayscale X display support
- two new demos: demos/gears.c and demos/shadow.c
- MachTen for Macintosh configuration

Bug fixes:

- glGet*(GL_DEPTH_BITS) returned bytes, not bits
- point, line, and bitmap rasterization suffered from roundoff errors
- fixed a division by zero error in line clipping
- occasional wrong default background color really fixed!
- glDepthFunc(GL_ALWAYS) with glDepthMask(GL_FALSE) didn't work
- gluBuild2DMipmaps malloc problem fixed
- view volume clipping of smooth shaded lines resulted in bad colors

Changes:

- new visual selection method in glXChooseVisual()

- improved GLU quadric functions
- call XSync for glFinish and XFlush for glFlush
- glVertex() calls now use a function pointer to avoid conditionals
- removed contrib directory from Mesa tar file (available on ftp site)
- AIX shared library support
- Removed GLUenum type as it's not in OpenGL

4.282.10 1.2.3 September 26, 1995

New features:

- Mesa header files now equivalent to SGI OpenGL headers
- Support for HP's Color Recovery dithering displays
- Faster vertex transformation
- Faster raster operations into X windows under certain conditions
- New configurations: HP w/ shared libs, Ultrix w/ GCC, Data General
- 4-bit visuals now supported

Bug fixes:

- glScissor bug fixed
- round-off errors in clipping lines against clip planes fixed
- byte swapping between hosts and display servers implemented
- glGetError() can be called without a current rendering context
- problem with accidentally culled polygons is fixed
- fixed some widget compilation problems

4.282.11 1.2.4 November 17, 1995

New features:

- More speed improvements (lighting, fogging, polygon drawing)
- Window system and OS-independent off-screen rendering
- Preliminary Fortran bindings
- glPolygonOffsetEXT implemented
- glColorMask and glIndexMask now fully implemented
- glPixelZoom implemented
- display lists fully implemented
- gamma correction
- dithering in 8-bit TrueColor/DirectColor visuals

Changes:

- Improved device driver interface

- tk.h renamed to gltk.h to avoid conflicts with Tcl's Tk
- Dithering support moved from core into device driver

Bug fixes:

- glEnable/Disable(GL_LIGHTING) didn't always take effect
- glReadPixels byte swapping was broken
- glMaterial with pname==GL_AMBIENT_AND_DIFFUSE was broken
- duplicate glColor4b() prototype in GL/gl.h removed
- stripes in wave -ci demo fixed
- GL_LINEAR_MIPMAP_NEAREST had wrong value
- bugs in HP Color Recovery support fixed
- fixed bug when blending lines, points, bitmaps outside of window

4.282.12 1.2.5 November 30, 1995

New Features:

- updated MS Windows driver
- new implementation of StaticGray/GrayScale visual support

Bug fixes:

- pixelzooming with gamma correction or blending didn't work
- HP color recovery visual wasn't being picked by glXChooseVisual
- glClear didn't always observe glColorMask changes
- olympic and offset demos didn't compile on some Suns
- texcoord clamping wasn't correct
- a polygon optimization introduced an occasional sampling problem

4.282.13 1.2.6 January 26, 1996

New Features:

- faster line and polygon rendering under certain conditions. See Performance Tips 9 and 10 in README
- profiling
- lighting is a bit faster
- better perspective corrected texture mapping
- Amiga AmiWin (X11) support
- preliminary Linux SVGA driver Changes:
- now using a 16-bit depth buffer, faster, smaller
- GL_NORMALIZE is disabled by default

Bug fixes:

- projective texture mapping

- fixed a memory leak in the context destroy function
- GL_POLYGON with less than 3 vertices caused a crash
- glGet*() returned wrong result for GL_INDEX_MODE
- reading pixels from an unmapped X window caused a BadMatch error

4.282.14 1.2.7 March 5, 1996

New:

- faster lighting
- faster 16-bit TrueColor rendering on Linux
- faster 32-bit TrueColor rendering on Linux, HP, IBM
- non-depth-buffered XImage polygons are faster
- vertex array extension
- software alpha planes
- updated Macintosh driver
- new NeXT driver
- GLU quadric functions generate texture coordinates
- reflect.c demo - reflective, textured surface demo

Changes:

- gamma correction code moved into the X driver for better performance

Bug fixes:

- multiple glClipPlane()'s didn't work reliably
- glPolygonMode() didn't always work
- glCullFace(GL_FRONT_AND_BACK) didn't work
- texture mapping with gamma correction was buggy
- floating point exceptions in texture coordinate interpolation
- XImage byte swapping didn't always work
- polygon edge flags weren't always used correctly

4.282.15 1.2.8 May 22, 1996

New:

- overlay planes on X servers with the SERVER_OVERLAY_VISUALS property
- better monochrome output
- more IRIX 6.x configurations
- more robust RGB mode color allocation
- added MESA_XSYNC environment variable
- GLX_MESA_pixmap_colormap and GLX_EXT_visual_info extensions

- GL_MESA_window_pos extension
- faster glReadPixels/glDrawPixels for GL_DEPTH and GL_UNSIGNED_SHORT and GL_UNSIGNED_INT
- driver for prototype Cirrus Mondello 3-D board
- updated AmigaDOS driver
- a few small speed optimizations in polygon rendering

Changes:

- internal device driver interface modified to simplify device driver implementations and to support hardware Z buffers
- several changes to the X/Mesa interface (xmesa.h)

Bug fixes:

- fixed pow(0,0) domain error triggered on some systems
- glStencilClear() in a display list caused an infinite loop
- glRasterPos*() was sometimes off by +/-0.5 in X and Y
- color masking and blending were performed in wrong order
- auxSolidCylinder() sometimes drew a wire-frame cylinder
- fixed file writing bug in osdemo.c
- pixel mapping didn't always work
- the GL_GEQUAL stencil func didn't work
- the GL_INVERT stencil op didn't work
- the stencil write mask didn't work
- glPush/PopAttrib() didn't do enough error checking
- glIsList() didn't always work correctly

4.282.16 2.0 October 10, 1996

New:

- Implements OpenGL 1.1 API functions
- all texture filtering modes supported (mipmapping)
- faster texture mapping, see Performance Tip 11 in README
- antialiased RGB points
- X support for line and polygon stippling
- glDrawBuffer(GL_FRONT_AND_BACK) works
- util/ directory of useful stuff
- demos/texobj demo of texture objects

Changes:

- major internal changes for thread-safeness
- new device driver interface

- MESA_ALPHA env variable removed
- triangle rasterizer replaces polygon rasterizer

Bug fixes:

- glPopAttrib() bug
- glDrawBuffer(GL_NONE) works now

4.282.17 2.1 December 14, 1996

New:

- VMS support
- MS-DOS driver
- OpenStep support
- updated, combined Windows 95/NT driver
- implemented glGetLighti() and glGetTexGen*()
- GLX does garbage collection of ancillary buffers

Bug fixes:

- removed unused _EXT constants from gl.h
- fixed polygon offset bugs
- Z coordinates of clipped lines were incorrect
- glEdgeFlag() in display lists didn't always work
- glLight*() in display lists didn't work
- fixed X line stipple bugs (Michael Pichler)
- glXUseXfonts XFreeFont/XFreeFontInfo bug fixed
- fixed a feedback bug
- glTexGen*() now transforms GL_EYE_PLANE by inverse modelview matrix
- polygons were sometimes culled instead of clipped
- triangle rasterizer suffered from float/int overflow exceptions
- fixed FP underflow exception in lighting (specular exponent)
- glEnable/glDisable of GL_EXT_vertex_array enums didn't work
- fixed free(NULL) in GLU tesselator code
- using 24-bit color on some X servers resulted in garbage rendering
- 32-bit per pixel mode for XFree86 now works
- glRotate(a,0,0,0) gave unpredictable results
- GL_LINE_STRIP with > 480 vertices had occasional clipping problems
- 8-bit TrueColor GLXPixmap rendering incorrectly required a colormap
- glMaterial() wasn't ignored when GL_COLOR_MATERIAL was enabled
- glEnable(GL_COLOR_MATERIAL) followed by glColor() didn't work right

- accumulation buffer was limited to positive values
- projective textures didn't work
- selection buffer overflows weren't handled correctly

Changes:

- restored the GL_EXT_polygon_offset extension
- slightly faster RGB dithering
- the SVGA driver works again
- Amiga driver now distributed separately
- NeXT driver updated for Mesa 2.x

4.282.18 2.2 March 14, 1997

New:

- better color selection when dithering
- added GL_EXT_texture_object extension
- updated MS-DOS driver for DJGPP
- added openbsd make configuration
- faster dithered flat-shaded triangles
- various compilation problems with Motif widgets fixed
- gl.h, glx.h and glu.h name mangling option
- BeOS driver
- 3D texture mapping extension
- GL_MESA_resize_buffers extension
- morph3d, stex3d and spectex demos
- 3Dfx support

Bug fixes:

- glColorMaterial should finally work right in all respects
- linear interpolation of mipmap levels was incorrectly weighted
- readpix.c didn't compile on Macintosh
- GL_INVERT and related logic ops didn't work right
- glTexImage[12]D() didn't check its parameters consistently
- fixed a memory leak in glTexImage[12]D()
- kludged around a SunOS 5.x/GCC compiler bug in the feedback code
- glReadPixels aborted instead of normally catching some errors
- a few 1.1 constants were missing or misnamed in gl.h
- glBegin(p); glBegin(q); didn't generate an error
- fixed a memory leak in GLX code

- clipping of concave polygons could cause a core dump
- 1-component alpha texture maps didn't work
- fixed a GLU polygon tesselator bug
- polygons with colinear vertices were sometimes culled
- feedback triangle colors were wrong when using smooth shading
- textures with borders didn't work correctly
- colors returned in feedback mode were wrong when using lighting
- spotlights didn't effect ambient lighting correctly
- gluPartialDisk() had a few bugs

Changes:

- device driver interface expanded to support texture mapping
- faster matrix inversion subroutine
- commented out #include "wmesa_extend.h" from src/wmesa.c
- fixed many compiler warnings in the demo programs

4.282.19 2.3 June 30, 1997

New:

- Mesa distribution divided into two pieces: library code and demos
- faster vertex transformation, clip testing, lighting
- faster line drawing
- TrueColor visuals how have dithering (for depths < 24 bits)
- added MESA_NO_DITHER environment variable
- new device driver function: NearFar(), RenderVB(), RasterSetup()
- added LynxOS configuration
- added cygnus Win32 configuration
- added texcyl.c GLUT demo
- added XMesaDitherColor() to X/Mesa interface
- new NURBS code from Bogdan Sikorski
- added demos/shape.c (non-rectangular X window!)

Bug fixes:

- glEnable/DisableClientState() were missing from GL/gl.h
- GL_SPHERE_MAP texcoord generation didn't work correctly
- glXGetConfig() returned wrong number of depth, stencil, accum bits
- glDrawPixels feedback/selection didn't examine RasterPos valid bit
- black and white were reversed on some monochrome displays
- fixed potential image memory leak (wasn't setting reference counter)

- glDrawPixels sometimes didn't recognize some GL state changes
- gluProject/UnProject() didn't check for divide by zero
- stex3d demo called random() and srandom(), not portable
- fixed memory leaks in context.c and drawpix.c
- fixed NULL dereferencing problem in gl_update_texture_state()
- glReadPixels between glBegin/glEnd didn't generate an error.
- fixed memory leak in polygon tesselator (Randy Frank)
- fixed seg fault bug drawing flat-shaded, depth-tested lines
- clipped GL_TRIANGLE_STRIPs sometimes had wrong color when flat-shaded
- glBindTexture sometimes didn't work
- fixed a bug deep in glXReleaseBuffersMESA()
- fog was mistakenly applied to alpha
- glPopMatrix didn't set "dirty matrix" flag
- glPolygonStipple pattern was sometimes wrong
- glClear wasn't disabled during feedback and selection
- fixed memory leak in glTexSubImage[123]D

Changes:

- many library source files reorganized
- faster X color allocation, colors also freed when finished with them
- new texture sampling function pointer in texture objects
- incorporated 3Dfx VooDoo driver v0.16 into main source tree
- many 3Dfx driver updates
- cygnus Makefiles now included
- updated DOS driver
- made a few changes to dosmesa.c and wmesa.c (VB->Unclipped)
- internally, colors now stored in GLubytes, not GLfixed
- optimized changing of GL_SHININESS parameter

4.282.20 2.4 September 18, 1997

New:

- updated 3Dfx Glide driver
- hacks for 3Dfx rendering into an X window or fullscreen
- added depth buffer access functions to X/Mesa and OS/Mesa interfaces

Bug fixes:

- pixel buffer could overflow with long, wide lines
- fixed FP underflow problems in lighting

- glTexSubImage1D() had an unitialized variable
- incomplete texture objects could cause a segfault
- glDrawPixels with GL_COMPILE_AND_EXECUTE caused infinite loop
- flat-shaded quads in a strip were miscolored if clipped
- mipmapped triangle lod computation now works correctly
- fixed a few under/overflow bugs in triangle rasterizer
- glArrayElement() assigned bad normal if normal array disabled
- changed argument to glXReleaseBuffersMESA()
- fixed small triangle underflow bugs in tritemp.h (hopefully)
- glBindTexture(target, 0) caused a crash
- glTexImage[123]D() with NULL image pointer caused crash
- glPixelStore parameters are now ignored during display list execution
- fixed a two-sided lighting w/ clipping bug (black vertices)
- textures with width!=height were sometimes mis-rendered
- "weird" projection matrices could cause div by 0, other fp errors

Changes:

- changed precompiled header symbol from PCH to PC_HEADER
- split api.c into api1.c and api2.c
- added hash.c source file (but not used yet)
- a few Sun and HP configuration file changes
- MESA_GLX_FX env var replaces MESA_FX_WINDOW and MESA_FX_FULLSCREEN
- fixed a few cygnus build problems (src/Makefile.cygnus, src/wmesa.c)

4.282.21 2.5 November 20, 1997

New:

- updated 3Dfx driver (v20) for GLQuake
- added GL_EXT_paletted_texture extension
- added GL_EXT_shared_texture_palette extension
- added GL_EXT_point_parameters extension
- now including Mark Kilgard's GLUT library v3.6
- new GLUT-based demos in gdemos/
- added a few more Unix config targets
- added Intel X86 assembly language vertex transformation code
- 3Dfx/Glide driver for Mesa now recognizes SST_SCREENREFRESH env var
- Windows 95 S3 Virge driver

Bug fixes:

- glCopyTexImage?D would crash due to uninitialized variable
- glColor w/ glColorMaterial in a display list caused a bug
- fixed several glDrawPixels() and ReadPixels() bugs in 3Dfx driver
- glVertex4*() vertices weren't always projected correctly
- trying to use mipmapped textured points or lines caused crash
- glColor[34][fd]() values now clamped to [0,1] before int conversion

Changes:

- new device driver functions for texture mapping
- hash tables used for display list and texture object lookup
- fixed GLX visual handling code to avoid saving redundant visuals
- 3Dfx Glide libraries automatically linked to libMesaGL.so
- dropped the Cirrus Logic Mondello code since it's obsolete
- updated Cygnus Makefiles (Stephane Rehel)
- updated Windows MSVC++ Makefiles (Oleg Letsinsky)
- procedure for making library files has changed: scripts now take a major and minor version arguments. Makeconfig changed a lot.
- new implementation of glTexSubImage2D()
- updated widgets-mesa directory to create libMesaGLwM.a (Motif widget)
- separate linux-glide and linux-386-glide configurations

4.282.22 2.6 February 12, 1998

New:

- Windows WGL functions
- updated VMS, DOS, Windows, Cygnus, BeOS, Amiga compilation support
- v0.22 of 3Dfx Glide driver
- more X86 assembly language optimizations
- faster blending for some modes
- XMesaSetFXmode() to switch between 3Dfx window and full-screen mode
- added preliminary thread support
- added GLX_MESA_copy_sub_buffer extension
- some clipping optimizations

Bug fixes:

- fixed shading/material bug when drawing long primitive strips
- fixed clipping problem in long primitive strips
- fixed clipping bug when using 3Dfx driver
- fixed a problem when trying to use X fonts w/ 3Dfx driver

- fixed a texture filter bug in 3Dfx/Glide driver
- fixed bug in 3Dfx/Glide driver involving depth mask & clearing
- glLoadMatrix to set projection matrix confused the 3Dfx driver
- non-identity texture matrices didn't work with linux-386 configs
- glGenTextures() didn't reserve the returned texture IDs
- NULL proxy image sent to glTexImageXD() caused crash
- added texture state validation optimization (Henk Kok)
- fixed colormap reuse problem when using both RGB and CI windows
- 32 BPP True/DirectColor X visuals weren't recognized
- fixed potential problem in evaluators memory allocation
- fixed assorted demo compilation bugs

Changes:

- replaced old Mesa/windows/ directory with Mesa/WIN32/ directory
- converted a few old glaux/gltk demos to GLUT
- renamed directories: demos -> xdemos, gdemos -> demos

4.282.23 3.0 September 17, 1998

New:

- OpenGL 1.2 API
- GL_EXT_abgr pixel format extension
- GL_SGIS_texture_edge_clamp extension
- GL_SGIS_multitexture extension (to be replaced by GL_ARB_multitex)
- GL_EXT_multitexture extension (to be replaced by GL_ARB_multitex)
- GL_EXT_rescale_normal extension and renormal.c demo
- GLX_SGI_video_sync extension (a no-op)
- antialiased lines
- glGetTexImage() now implemented
- glDraw/Copy/ReadPixels() optimizations
- optimized textured triangle code (Marten Stromberg)
- more optimization of dithered TrueColor triangles in X driver
- Linux GGI driver
- updated MGL driver

Bug fixes:

- lots of assorted compilation fixes
- glInitNames didn't write initial hit record
- glBitmap didn't always check for invalid raster position

- switching between GLX and OSMesa contexts caused a crash
- fixed uninitialized variable in Mesa widget code
- fixed typo in texture code which caused book/texgen to crash
- fixed texture sampling bug when filter=GL_LINEAR and wrap=GL_CLAMP
- gluDisk() in POINT or LINE mode sometimes failed
- fixed texture + fog bug
- GL_COMPILE_AND_EXECUTE mode didn't work reliably
- glMultMatrix in projection matrix mode w/ 3Dfx driver could fail
- glDrawPixels(color index pixels) weren't converted to RGBA
- fixed possible getenv() buffer overflow security bug
- glBitmap in feedback mode was offset by xOrig, yOrig params
- device driver's DrawPixels hook was never used
- glDrawPixels with zoomY!=1 and top/bottom clipping didn't work
- glDrawPixels optimized for GL_LUMINANCE, GL_LUMINANCE_ALPHA, GLubyte
- fixed MakeCurrent bug in GLwRedrawObjects() in MesaWorkstation.c
- glCopyTexSubImage2D() didn't work with 3Dfx driver
- lines with width = 2 could cause crash
- glClear with scissor rect sometimes cleared whole buffer
- glTexSubImage2D(.. GL_COLOR_INDEX ..) didn't work
- glTexImageXD(.. GL_ABGR_EXT ..) didn't work
- computation of inverse modelview matrix sometimes failed
- fixed GL_CLAMP mode texture sampling bug
- textured line interpolation was somewhat broken
- textured triangle interpolation was also somewhat broken
- glGet(MODELVIEW/PROJECTION/TEXTURE_MATRIX_STACK_DEPTH) off by one
- evaluator state wasn't fully initialized
- texture coordinate clipping was buggy
- evaluator surfaces could be mis-colored
- glAccum(GL_RETURN, s) didn't obey glColorMask() settings
- zero area polygons shouldn't be culled if polygon mode is point/line
- clipped width and height of glReadPixels was sometimes off by one
- blending with alpha = 0 or 1.0 wasn't always exact
- reading of pixels from clipped region was buggy
- minor tweaking of X visual management in GLX emulator
- glPolygonStipple now obeys pixel unpacking parameters
- glGetPolygonStipple now obeys pixel packing parameters

- interleaved vertex array texture coordinates were broken
- query of proxy texture internal format was broken
- alpha channel wasn't reliably cleared
- fixed divide by zero error in gluScaleImage if dest size = 1 x 1

Conformance bug fixes:

- GL_SELECTION_BUFFER_POINTER and GL_SELECTION_BUFFER_SIZE were missing
- GL_TEXTURE_INTERNAL_FORMAT was missing
- glGet*(GL_POLYGON_STIPPLE) was broken
- glPush/PopAttrib() didn't save/restore all texture state
- glBitmap in feedback mode didn't work
- feedback of texture coords didn't always work
- glDrawPixels w/ format=GL_DEPTH_COMPONENT, type=GLbyte was broke
- glDrawPixels w/ format=GL_DEPTH_COMPONENT, type=GLubyte was broke
- glDrawPixels w/ format=GL_STENCIL_INDEX, type=GL_BITMAP was broke

Changes:

- upgraded GLUT to version 3.7
- only GL and GLU library code included in MesaLib.tar.gz
- GLUT and all demos now in MesaDemos.tar.gz
- glaux and gltk libraries removed
- IRIX -n32 and -64 libs go in lib32/ and lib64/ directories

4.282.24 3.1 beta 1 November 19, 1998

New:

- GL_EXT_stencil_wrap extension
- GL_INGR_blend_func_separate extension
- GL_ARB_multitexture extension
- GL_NV_texgen_reflection extension
- newly optimized vertex transformation code
- updated GLUT 3.7 code
- better precision when using 32-bit Z buffer
- Allegro DJGPP driver

Bug fixes:

- glCopyPixels between front/back buffers didn't copy alpha correctly
- fixed out-of-bounds memory access in optimized 2-D texture code
- glPixelStorei didn't accept GL_PACK/UNPACK_IMAGE_HEIGHT parameter
- glGet*() didn't accept GL_MAX_3D_TEXTURE_SIZE parameter

- clipping of texture coordinates sometimes had bad R,Q values
- GL_CLAMP_TO_EDGE texture sampling was off by 0.5 texels
- glEdgeFlagPointer() now takes a GLvoid * instead of GLboolean *
- texture was sometimes applied twice with 3Dfx driver
- glPush/PopAttrib() fouled up texture object reference counts
- glDeleteLists(0, n) caused assertion failure
- bilinear texture sampling wasn't accurate enough
- glClear w/ glDepthMask(GL_FALSE) didn't work right on 3Dfx
- color components were reversed on big endian 32 BPP X visuals

Changes:

- removed GL_EXT_multitexture extension

4.282.25 3.1 beta 2 May 24, 1999

New:

- multi-textured points and lines (mjk@nvidia.com)
- optimized 24 BPP X rendering (bernd.paysan@gmx.de)
- added allegro support (bernie-t@geocities.com)
- cleaned-up Windows-related stuff (Ted Jump)
- minor stereo changes (KendallB@scitechsoft.com)
- new BeOS driver which implements BGLView class
- new Direct3D driver (see src/D3D)
- more efficient filled gluCylinder() function
- utilities: util/showbuffer.[ch] and util/glstate.[ch]
- fixed some IRIX compiler warnings
- added support for building Mesa in XFree86 with SGI's GLX (kevin@precisioninsight.com)

Bug fixes:

- a variety of Windows/Mesa bug fixes (mjk@nvidia.com)
- packed pixel images weren't unpacked correctly
- patches some win32 files in GLUT (mjk@nvidia.com)
- glTexImage[123]D() didn't accept internalFormat == GL_COLOR_INDEX
- fixed lighting bug in Keith's new shading code
- fixed texture segfault seen in Lament screensaver
- fixed miscellaneous low-memory bugs
- glClear(GL_COLOR_BUFFER_BIT) with RGBA or CI masking was broken
- GL_LINEAR sampling of 3D textures was broken
- fixed SVR4 'cc' compiler macro problem (dawes@xfree86.org)

- added GL_TEXTURE_PRIORITY fix (keithh@netcomuk.co.uk)
- fixed wide point and wide line conformance bugs (brianp)

Changes:

- some device driver changes (see src/dd.h)
- new copyright on core Mesa code

4.282.26 3.1 beta 3 September 17, 1999

New:

- optimized glAccum function
- optimized 24 BPP rendering in XMesa driver
- GLU 1.2 polygon tessellator

Bug Fixes:

- glGetTexLevelParameter wasn't fully implemented
- glXUseXFont now handles multi-byte fonts
- glIsEnabled(GL_TEXTURE_2D / 3D) returned wrong result
- alpha channel of blending points, lines was sometimes incorrect

Changes:

- New library names: "libGL" instead of "libMesaGL"
- New library numbering: libGL.so.1.2.310
- New subdirectories: docs/ and bin/
- New Makefile-system (autoconf,automake,libtool)

4.282.27 3.1 final December 14, 1999

New:

- added demos/gloss.c
- added xdemos/glxdpyinfo.c
- added GLX_ARB_get_proc_address extension
- rewritten glTexImage code paths (faster, less memory, bug fixes)

Bug Fixes:

- several vertex array bug fixes
- overlapping glCopyPixels with pixel zooming now works
- glXUseXFont() bitmaps were vertically shifted by one pixel
- glCopyPixels with pixel zooming now works

4.282.28 3.2 final April 24, 2000

Bug fixes:

- fixed memcpy bugs in span.c
- fixed missing glEnd problem in demos/tessdemo.c
- fixed bug when clearing 24 BPP Ximages
- fixed clipping problem found in Unreal Tournament
- fixed Loki's "ice bug" and "crazy triangles" seen in Heretic2
- fixed Loki's 3dfx RGB vs BGR bug
- fixed Loki's 3dfx smooth/flat shading bug in SoF

Changes:

- updated docs/README file
- use bcopy() optimizations on FreeBSD
- re-enabled the optimized persp_textured_triangle() function

4.282.29 3.2.1 July 19, 2000

Bug fixes:

- gluBuild2DMipmaps() didn't accept GL_BGRA
- Fixed compile/makefile problems on IRIX
- fixed segfault in 3dfx driver when using GL selection/feedback
- no longer cull very, very tiny triangles
- blending w/ drawbuffer==GL_FRONT_BACK caused segfault (sw rendering)
- fixed Motif detection code in widgets-mesa/configure.in
- glColorMaterial and glMaterial updates to emissive and ambient didn't always work right
- Specular highlights weren't always in the right place
- clipped GL_LINE mode polygons had interior lines appear
- blend term GL_ONE_MINUS_CONSTANT_ALPHA was broken
- GL_NICEST fog didn't always work with flat shading
- glRect commands in display lists were sometimes miscolored
- Line Z offset didn't always work
- fixed texgen normal vector problem (gloss's teapot)
- numerous GL conformance bugs fixed

Changes:

- glColorMask(false, false, false, false) handled better/faster
- reverted to old GLU polygon tessellator, GLU 1.1
- updated Win32 build files

4.282.30 3.3 July 21, 2000

New:

- antialiased triangles now implemented
- GL_EXT_texture_env_add texture mode extension
- GLX 1.3 API
- support for separate draw/read buffers (ie GL_SGI_make_current_read)
- thread-safe API dispath
- improved glxinfo program
- demos/texdown program to measure texture download performance
- glext.h header file
- demos/geartrain program
- GL_EXT_texture_lod_bias extension
- demos/lodbias program
- further optimized glRead/DrawPixels for 16-bit TrueColor X visuals
- GLX_EXT_visual_rating extension (a no-op, however)
- GL_HP_occlusion_test extension (for X and OS/Mesa drivers)
- demos/occlude program
- GL_SGIS_pixel_texture and GL_SGIX_pixel_texture extensions
- demos/pixeltex program
- GL_SGI_color_matrix extension
- GL_SGI_color_table extension
- GL_EXT_histogram extension
- GL_ARB_texture_cube_map extension
- added xdemos/glxheads and xdemos/manywin
- demos/texenv.c demo
- GL_EXT_texture_env_combine extension (by Holger Waechtler)
- Xlib driver is now thread-safe (see xdemos/glthreads)

Bug Fixes:

- various GL conformance failures fixed since 3.2.1

Changes:

- gl.h now uses #defines instead of C enums for all tokens
- glu.h now uses #defines instead of C enums for all tokens
- moved programs from 3Dfx/demos/ into demos/ directory

4.282.31 3.4 November 3, 2000

New:

- optimized glDrawPixels for glPixelZoom(1,-1) Bug Fixes:
- widgets-mesa/src/*.c files were missing from 3.3 distro
- include/GL/mesa_wgl.h file was missing from 3.3 distro
- fixed some Win32 compile problems
- texture object priorities weren't getting initialized to 1.0
- glAreTexturesResident return value was wrong when using hardware
- glXUseXFont segfaulted when using 3dfx driver (via MESA_GLX_FX)
- glReadPixels with GLushort packed types was broken
- fixed a few bugs in the GL_EXT_texture_env_combine texture code
- glPush/PopAttrib(GL_ENABLE_BIT) mishandled multi-texture enables
- fixed some typos/bugs in the VB code
- glDrawPixels(GL_COLOR_INDEX) to RGB window didn't work
- optimized glDrawPixels paths weren't being used
- per-fragment fog calculation didn't work without a Z buffer
- improved blending accuracy, fixes Glean blendFunc test failures
- glPixelStore(GL_PACK/UNPACK_SKIP_IMAGES) wasn't handled correctly
- glXGetProcAddressARB() didn't always return the right address
- gluBuild[12]DMipmaps() didn't grok the GL_BGR pixel format
- texture matrix changes weren't always detected (GLUT projtex demo)
- fixed random color problem in vertex fog code
- fixed Glide-related bug that let Quake get a 24-bit Z buffer

Changes:

- finished internal support for compressed textures for DRI

4.282.32 3.4.1 February 14, 2001

New:

- fixed some Linux build problems
- fixed some Windows build problems
- GL_EXT_texture_env_dot3 extension (Gareth Hughes)

Bug fixes:

- added RENDER_START/RENDER_FINISH macros for glCopyTexImage in DRI
- various state-update code changes needed for DRI bugs
- disabled pixel transfer ops in glColorTable commands, not needed

- fixed bugs in glCopyConvolutionFilter1D/2D, glGetConvolutionFilter
- updated sources and fixed compile problems in widgets-mesa/
- GLX_PBUFFER enum value was wrong in glx.h
- fixed a glColorMaterial lighting bug
- fixed bad args to Read/WriteStencilSpan in h/w stencil clear function
- glXCopySubBufferMESA() Y position was off by one
- Error checking of glTexSubImage3D() was broken (bug 128775)
- glPopAttrib() didn't restore all derived Mesa state correctly
- Better glReadPixels accuracy for 16 BPP color - fixes lots of OpenGL conformance problems at 16 BPP.
- clearing depth buffer with scissoring was broken, would segfault
- OSMesaGetDepthBuffer() returned bad bytesPerValue value
- fixed a line clipping bug (reported by Craig McDaniel)
- fixed RGB color over/underflow bug for very tiny triangles

Known problems:

- NURBS or evaluator surfaces inside display lists don't always work

4.282.33 3.4.2 May 17, 2001

Bug fixes:

- deleting the currently bound texture could cause bad problems
- using fog could result in random vertex alpha values
- AA triangle rendering could touch pixels outside right window bound
- fixed byteswapping problem in clear_32bit_ximage() function
- fixed bugs in wglUseFontBitmapsA(), by Frank Warmerdam
- fixed memory leak in glXUseXFont()
- fragment sampling in AA triangle function was off by 1/2 pixel
- Windows: reading pixels from framebuffer didn't always work
- glConvolutionFilter2D could segfault or cause FP exception
- fixed segfaults in FX and X drivers when using tex unit 1 but not 0
- GL_NAND logicop didn't work right in RGBA mode
- fixed a memory corruption bug in vertex buffer reset code
- clearing the software alpha buffer with scissoring was broken
- fixed a few color index mode fog bugs
- fixed some bad assertions in color index mode
- fixed FX line 'stipple' bug #420091
- fixed stencil buffer clear width/height typo
- fixed GL error glitches in gl[Client]ActiveTextureARB()

- fixed Windows compilation problem in texutil.c
- fixed 1/8-pixel AA triangle sampling error

Changes:

- optimized writing mono-colored pixel spans to X pixmaps
- increased max viewport size to 2048 x 2048

4.282.34 3.5 June 21, 2001

New:

- internals of Mesa divided into modular pieces (Keith Whitwell)
- 100% OpenGL 1.2 conformance (passes all conformance tests)
- new AA line algorithm
- GL_EXT_convolution extension
- GL_ARB_imaging subset
- OSMesaCreateContextExt() function
- GL_ARB_texture_env_add extension (same as GL_EXT_texture_env_add)
- GL_MAX_TEXTURE_UNITS_ARB now defaults to eight
- GL_EXT_fog_coord extension (Keith Whitwell)
- GL_EXT_secondary_color extension (Keith Whitwell)
- GL_ARB_texture_env_add extension (same as GL_EXT_texture_env_add)
- GL_SGIX_depth_texture extension
- GL_SGIX_shadow and GL_SGIX_shadow_ambient extensions
- demos/shadowtex.c demo of GL_SGIX_depth_texture and GL_SGIX_shadow
- GL_ARB_texture_env_combine extension
- GL_ARB_texture_env_dot3 extension
- GL_ARB_texture_border_clamp (aka GL_SGIS_texture_border_clamp)
- OSMesaCreateContextExt() function
- libOSMesa.so library, contains the OSMesa driver interface
- GL/glxext.h header file for GLX extensions
- somewhat faster software texturing, fogging, depth testing
- all color-index conformance tests now pass (only 8 BPP tested)
- SPARC assembly language TCL optimizations (David Miller)
- GL_SGIS_generate_mipmap extension

Bug Fixes:

- fbiRev and tmuRev were unitialized when using Glide3
- fixed a few color index mode conformance failures; all pass now
- now appling antialiasing coverage to alpha after texturing

- colors weren't getting clamped to [0,1] before color table lookup
- fixed RISC alignment errors caused by COPY_4UBV macro
- drawing wide, flat-shaded lines could cause a segfault
- vertices now snapped to 1/16 pixel to fix rendering of tiny triangles

Changes:

- SGI's Sample Implementation (SI) 1.3 GLU library replaces Mesa GLU
- new libOSMesa.so library, contains the OSMesa driver interface

4.282.35 4.0 October 22, 2001

New:

- Mesa 4.0 implements the OpenGL 1.3 specification
- GL_IBM_rasterpos_clip extension
- GL_EXT_texture_edge_clamp extension (aka GL_SGIS_texture_edge_clamp)
- GL_ARB_texture_mirrored_repeat extension
- WindML UGL driver (Stephane Raimbault)
- added OSMESA_MAX_WIDTH/HEIGHT queries
- attempted compiliation fixes for Solaris 5, 7 and 8
- updated glext.h and glxext.h files
- updated Windows driver (Karl Schultz)

Bug fixes:

- added some missing GLX 1.3 tokens to include/GL/glx.h
- GL_COLOR_MATRIX changes weren't recognized by teximage functions
- glCopyPixels with scale and bias was broken
- glRasterPos with lighting could segfault
- glDeleteTextures could leave a dangling pointer
- Proxy textures for cube maps didn't work
- fixed a number of 16-bit color channel bugs
- fixed a few minor memory leaks
- GLX context sharing was broken in 3.5
- fixed state-update bugs in glPopClientAttrib()
- fixed glDrawRangeElements() bug
- fixed a glPush/PopAttrib() bug related to texture binding
- flat-shaded, textured lines were broken
- fixed a dangling pointer problem in the XMesa code (Chris Burghart)
- lighting didn't always produce the correct alpha value
- fixed 3DNow! code to not read past end of arrays (Andrew Lewycky)

4.282.36 4.0.1 December 17, 2001

New:

- better sub-pixel sample positions for AA triangles (Ray Tice)
- slightly faster blending for (GL_ZERO, GL_ONE) and (GL_ONE, GL_ZERO)

Bug fixes:

- added missing break statements in glGet*() for multisample cases
- fixed uninitialized hash table mutex bug (display lists / texobjs)
- fixed bad teximage error check conditional (bug 476846)
- fixed demos readtex.c compilation problem on Windows (Karl Schultz)
- added missing glGet() query for GL_MAX_TEXTURE_LOD_BIAS_EXT
- silence some compiler warnings (gcc 2.96)
- enable the #define GL_VERSION_1_3 in GL/gl.h
- added GL 1.3 and GLX 1.4 entries to gl_mangle.h and glx_mangle.h
- fixed glu.h typedef problem found with MSDev 6.0
- build libGL.so with -Bsymbolic (fixes bug found with Chromium)
- added missing 'const' to glXGetContextIDEXT() in glxext.h
- fixed a few glXGetProcAddress() errors (texture compression, etc)
- fixed start index bug in compiled vertex arrays (Keith)
- fixed compilation problems in src/SPARC/glapi_sparc.S
- fixed triangle strip "parity" bug found in VTK medical1 demo (Keith)
- use glXGetProcAddressARB in GLUT to avoid extension linking problems
- provoking vertex of flat-shaded, color-index triangles was wrong
- fixed a few display list bugs (GLUT walker, molecule, etc) (Keith)
- glTexParameter didn't flush the vertex buffer (Ray Tice)
- feedback attributes for glDraw/CopyPixels and glBitmap were wrong
- fixed bug in normal length caching (ParaView lighting bug)
- fixed separate_specular color bug found in Chimera (18 Dec 2001)

4.282.37 4.0.2 April 2, 2002

New:

- New DOS (DJGPP) driver written by Daniel Borca
- New driver interface functions for TCL drivers (such as Radeon DRI)
- GL_RENDERER string returns "Mesa Offscreen16" or "Mesa Offscreen32" if using deep color channels
- latest GL/glext.h and GL/glxext.h headers from SGI

Bug fixes:

- GL_BLEND with non-black texture env color wasn't always correct
- GL_REPLACE with GL_RGB texture format wasn't always correct (alpha)
- glTexEnviv(pname != GL_TEXTURE_ENV_COLOR) was broken
- glReadPixels was sometimes mistakenly clipped by the scissor box
- glDraw/ReadPixels didn't catch all the errors that they should have
- Fixed 24 BPP rendering problem in Windows driver (Karl Schultz)
- 16-bit GLchan mode fixes (m_trans_tmp.h, s_triangle.c)
- Fixed 1-bit float->int conversion bug in glDrawPixels(GL_DEPTH_COMP)
- glColorMask as sometimes effecting glXSwapBuffers()
- fixed a potential bug in XMesaGarbageCollect()
- N threads rendering into one window didn't work reliably
- glCopyPixels didn't work for deep color channels
- improved 8 -> 16bit/channel texture image conversion (Gerk Huisma)
- glPopAttrib() didn't correctly restore user clip planes
- user clip planes failed for some perspective projections (Chromium)

Known bugs:

- mipmap LOD computation

4.282.38 4.0.3 June 25, 2002

New:

- updated GL/glext.h file (version 15)
- corrected MMX blend code (Jose Fonseca)
- support for software-based alpha planes in Windows driver
- updated GGI driver (Filip Spacek)

Bug fixes:

- glext.h had wrong values for GL_DOT3_RGB[A]_EXT tokens
- OSMesaMakeCurrent() didn't recognize buffer size changes
- assorted conformance fixes for 16-bit/channel rendering
- texcombine alpha subtraction mode was broken
- fixed lighting bug with non-uniform scaling and display lists
- fixed bug when deleting shared display lists
- disabled SPARC cliptest assembly code (Mesa bug 544665)
- fixed a couple Solaris compilation/link problems
- blending clipped glDrawPixels didn't always work
- glGetTexImage() didn't accept packed pixel types
- glPixelMapu[is]v() could explode given too large of pixelmap

- glGetTexParameter[if]v() didn't accept GL_TEXTURE_MAX_ANISOTROPY_EXT
- glXCopyContext() could lead to segfaults
- glCullFace(GL_FRONT_AND_BACK) didn't work (bug 572665)

Changes:

- lots of C++ (g++) code clean-ups
- lots of T&L updates for the Radeon DRI driver

Known bugs:

- mipmap LOD computation (fixed for Mesa 4.1)

4.282.39 4.0.4 October 3, 2002

New:

- GL_NV_texture_rectangle extension
- updated glext.h header (version 17)
- updated DOS driver (Daniel Borca)
- updated BeOS R5 driver (Philippe Houdoin)
- added GL_IBM_texture_mirror_repeat
- glxinfo now takes -l option to print interesting OpenGL limits info
- GL_MESA_ycbcr_texture extension
- GL_APPLE_client_storage extension (for some DRI drivers only)
- GL_MESA_pack_invert extension

Bug fixes:

- fixed GL_LINEAR fog bug by adding clamping
- fixed FP exceptions found using Alpha CPU
- 3dfx MESA_GLX_FX=window (render to window) didn't work
- fixed memory leak in wglCreateContest (Karl Schultz)
- define GLAPIENTRY and GLAPI if undefined in glu.h
- wglGetProcAddress didn't handle all API functions
- when testing for OpenGL 1.2 vs 1.3, check for GL_ARB_texture_cube_map
- removed GL_MAX_CONVOLUTION_WIDTH/HEIGHT from glGetInteger/Float/etc()
- error checking in compressed tex image functions had some glitches
- fixed AIX compile problem in src/config.c
- glGetTexImage was using pixel unpacking instead of packing params
- auto-mipmap generation for cube maps was incorrect

Changes:

- max texture units reduced to six to accommodate texture rectangles
- removed unfinished GL_MESA_sprite_point extension code

4.282.40 4.1 October 29, 2002

New:

- GL_NV_vertex_program extension
- GL_NV_vertex_program1_1 extension
- GL_ARB_window_pos extension
- GL_ARB_depth_texture extension
- GL_ARB_shadow extension
- GL_ARB_shadow_ambient extension
- GL_EXT_shadow_funcs extension
- GL_ARB_point_parameters extension
- GL_ARB_texture_env_crossbar
- GL_NV_point_sprite extension
- GL_NV_texture_rectangle extension
- GL_EXT_multi_draw_arrays extension
- GL_EXT_stencil_two_side extension
- GLX_SGIX_fbconfig and GLX_SGIX_pbuffer extensions
- GL_ATI_texture_mirror_once extension (Ian Romanick)
- massive overhaul/simplification of software rasterizer module, many contributions from Klaus Niederkrueger
- faster software texturing in some cases (i.e. trilinear filtering)
- new OSMesaGetProcAddress() function
- more blend modes implemented with MMX code (Jose Fonseca)
- added glutGetProcAddress() to GLUT
- added GLUT_FPS env var to compute frames/second in glutSwapBuffers()
- pbinfo and pbdemo PBuffer programs
- glxinfo -v prints transprent pixel info (Gerd Sussner)

Bug fixes:

- better mipmap LOD computation (prevents excessive blurriness)
- OSMesaMakeCurrent() didn't recognize buffer size changes
- assorted conformance fixes for 16-bit/channel rendering
- texcombine alpha subtraction mode was broken
- fixed some blend problems when GLchan==GLfloat (Gerk Huisma)
- clamp colors to [0,inf] in OSMesa if GLchan==GLfloat (Gerk Huisma)
- fixed divide by zero error in NURBS tessellator (Jon Perry)
- fixed GL_LINEAR fog bug by adding clamping
- fixed FP exceptions found using Alpha CPU
- 3dfx/glide driver render-to-window feature was broken

- added missing GLX_TRANSPARENT_RGB token to glx.h
- fixed error checking related to paletted textures
- fixed reference count error in glDeleteTextures (Randy Fayan)

Changes:

- New spec file and Python code to generate some GL dispatch files
- Glide driver defaults to "no" with autoconf/automake
- updated demos/stex3d with new options

4.282.41 5.0 November 13, 2002

New:

- OpenGL 1.4 support (glGetString(GL_VERSION) returns "1.4")
- removed some overlooked debugging code
- glxinfo updated to support GLX_ARB_multisample
- GLUT now support GLX_ARB_multisample
- updated DOS driver (Daniel Borca)

Bug fixes:

- GL_POINT and GL_LINE-mode polygons didn't obey cull state
- fixed potential bug in _mesa_align_malloc/calloc()
- fixed missing triangle bug when running vertex programs
- fixed a few HPUX compilation problems
- FX (Glide) driver didn't compile
- setting GL_TEXTURE_BORDER_COLOR with glTexParameteriv() didn't work
- a few EXT functions, like glGenTexturesEXT, were no-ops
- a few OpenGL 1.4 functions like glFogCoord*, glBlendFuncSeparate, glMultiDrawArrays and glMultiDrawElements were missing
- glGet*(GL_ACTIVE_STENCIL_FACE_EXT) was broken
- Pentium 4 Mobile was mistakenly identified as having 3DNow!
- fixed one-bit error in point/line fragment Z calculation
- fixed potential segfault in fakeglx code
- fixed color overflow problem in DOT3 texture env mode

4.282.42 5.0.1 March 30, 2003

New:

- DOS driver updates from Daniel Borca
- updated GL/gl_mangle.h file (Bill Hoffman) Bug fixes:
- auto mipmap generation for cube maps was broken (bug 641363)

- writing/clearing software alpha channels was unreliable
- minor compilation fixes for OS/2 (Evgeny Kotsuba)
- fixed some bad assertions found with shadowtex demo
- fixed error checking bug in glCopyTexSubImage2D (bug 659020)
- glRotate(angle, -x, 0, 0) was incorrect (bug 659677)
- fixed potential segfault in texture object validation (bug 659012)
- fixed some bogus code in _mesa_test_os_sse_exception_support (Linus)
- fix fog stride bug in tnl code for h/w drivers (Michel Danzer)
- fixed glActiveTexture / glMatrixMode(GL_TEXTURE) bug (#669080)
- glGet(GL_CURRENT_SECONDARY_COLOR) should return 4 values, not 3
- fixed compilation problem on Solaris7/x86 (bug 536406)
- fixed prefetch bug in 3DNow! code (Felix Kuhling)
- fixed NeXT build problem (FABSF macro)
- glDrawPixels Z values when glPixelZoom!=1 were invalid (bug 687811)
- zoomed glDraw/CopyPixels with clipping sometimes failed (bug 689964)
- AA line and triangle Z values are now rounded, not truncated
- fixed color interpolation bug when GLchan==GLfloat (bug 694461)
- glArePrograms/TexturesResident() wasn't 100% correct (Jose Fonseca)
- fixed a minor GL_COLOR_MATERIAL bug
- NV vertex program EXP instruction was broken
- glColorMask misbehaved with X window / pixmap rendering
- fix autoconf/libtool GLU C++ linker problem on Linux (a total hack)
- attempt to fix GGI compilation problem when MesaDemos not present
- NV vertex program ARL-relative fetches didn't work

Changes:

- use glPolygonOffset in gloss demo to avoid z-fighting artifacts
- updated winpos and pointblast demos to use ARB extensions
- disable SPARC normal transformation code (bug 673938)
- GLU fixes for OS/2 (Evgeny Kotsuba)

4.282.43 5.0.2 September 5, 2003

Bug fixes:

- fixed texgen problem causing texcoord's Q to be zero (stex3d)
- default GL_TEXTURE_COMPARE_MODE_ARB was wrong
- GL_CURRENT_MATRIX_NV query was wrong
- GL_CURRENT_MATRIX_STACK_DEPTH_NV query was off by one

- GL_LIST_MODE query wasn't correct
- GL_FOG_COORDINATE_SOURCE_EXT query wasn't supported
- GL_SECONDARY_COLOR_ARRAY_SIZE_EXT query returned wrong value
- blended, wide lines didn't always work correctly (bug 711595)
- glVertexAttrib4svNV w component was always 1
- fixed bug in GL_IBM_rasterpos_clip (missing return)
- GL_DEPTH_TEXTURE_MODE = GL_ALPHA didn't work correctly
- a few Solaris compilation fixes
- fixed glClear() problem for DRI drivers (non-existent stencil, etc)
- fixed int/REAL mixup in GLU NURBS curve evaluator (Eric Cazeaux)
- fixed delete [] bug in SI GLU (bug 721765) (Diego Santa Cruz)
- glFog() didn't clamp fog colors
- fixed bad float/int conversion for GL_TEXTURE_PRIORITY in the gl[Get]TexParameteri[v] functions
- fixed invalid memory references in glTexGen functions (bug 781602)
- integer-valued color arrays weren't handled correctly
- glDrawPixels(GL_DEPTH_COMPONENT) with glPixelZoom didn't work
- GL_EXT_texture_lod_bias is part of 1.4, overlooked in 5.0.1

Changes:

- build GLUT with -fexceptions so C++ apps propogate exceptions

4.282.44 5.1 December 17, 2003

New:

- reorganized directory tree
- GL_ARB_vertex/fragment_program extensions (Michal Krol & Karl Rasche)
- GL_ATI_texture_env_combine3 extension (Ian Romanick)
- GL_SGI_texture_color_table extension (Eric Plante)
- GL_NV_fragment_program extension
- GL_NV_light_max_exponent extension
- GL_EXT_texture_rectangle (identical to GL_NV_texture_rectangle)
- GL_ARB_occlusion_query extension
- GL_ARB_point_sprite extension
- GL_ARB_texture_non_power_of_two extension
- GL_IBM_multimode_draw_arrays extension
- GL_EXT_texture_mirror_clamp extension (Ian Romanick)
- GL_ARB_vertex_buffer_object extension
- new X86 feature detection code (Petr Sebor)

- less memory used for display lists and vertex buffers
- demo of per-pixel lighting with a fragment program (demos/fplight.c)
- new version (18) of glext.h header
- new spriteblast.c demo of GL_ARB_point_sprite
- faster glDrawPixels in X11 driver in some cases (see relnotes/5.1)
- faster glCopyPixels in X11 driver in some cases (see relnotes/5.1)

Bug fixes:

- really enable OpenGL 1.4 features in DOS driver.
- fixed issues in glDrawPixels and glCopyPixels for very wide images
- glPixelMapf/ui/usv()'s size parameter is GLsizei, not GLint
- fixed some texgen bugs reported by Daniel Borca
- fixed wglMakeCurrent(NULL, NULL) bug (#835861)
- fixed glTexSubImage3D z-offset bug (Cedric Gautier)
- fixed RGBA blend enable bug (Ville Syrjala)
- glAccum is supposed to be a no-op in selection/feedback mode
- fixed texgen bug #597589 (John Popplewell)

Changes:

- dropped API trace feature (src/Trace/)
- documentation overhaul. merged with website content. more html.
- glxgears.c demo updated to use GLX swap rate extensions
- glTexImage1/2/3D now allows width/height/depth = 0
- disable SPARC asm code on Linux (bug 852204)

4.282.45 6.0 January 16, 2004

New:

- full OpenGL 1.5 support
- updated GL/glext.h file to version 21 Changes:
- changed max framebuffer size to 4Kx4K (MAX_WIDTH/HEIGHT in config.h) Bug fixes:
- fixed bug in UNCLAMPED_FLOAT_TO_UBYTE macro; solves a color clamping issue
- updated suno5-gcc configs
- glColor3 functions sometimes resulted in undefined alpha values
- fixed FP divide by zero error seen on VMS with xlockmore, others
- fixed vertex/fragment program debug problem (bug 873011)
- building on AIX with gcc works now
- glDeleteProgramsARB failed for ARB fragment programs (bug 876160)
- glDrawRangeElements tried to modify potentially read-only storage

- updated files for building on Windows

4.282.46 6.0.1 April 2, 2004

New:

- upgraded glext.h to version 22
- new build targets (Dan Schikore)
- new linux-x86-opteron build target (Heath Feather)

Bug fixes:

- glBindProgramARB didn't update all necessary state
- fixed build problems on OpenBSD
- omit CVS directories from tarballs
- glGetTexImage(GL_COLOR_INDEX) was broken
- fixed an infinite loop in t&l module
- silenced some valgrind warnings about using unitialized memory
- fixed some compilation/link glitches on IRIX (Mike Stephens)
- glBindProgram wasn't getting compiled into display lists
- GLX_FBCONFIG_ID wasn't recognized in glXChooseFBConfig() (bug 888079)
- two-sided lighting and vertex program didn't work (bug 887330)
- stores to program parameter registers in vertex state programs didn't work.
- fixed glOrtho bug found with gcc 3.2.2 (RH9)
- glXCreateWindow() wasn't fully implemented (bug 890894)
- generic vertex attribute arrays didn't work in display lists
- vertex buffer objects' default usage and access fields were wrong
- glDrawArrays with start!=0 was broken
- fragment program PK2H, UP2H, UP4B and UP4UB instructions were broken
- linux-osmesa16-static config didn't work
- fixed a few color index rendering problems (bug 910687)
- glInterleavedArrays didn't respect GL_CLIENT_ACTIVE_TEXTURE
- OSMesa RGB and BGR modes were broken
- glProgramStringARB mistakenly required a null-terminated string
- fragment program XPD instruction was incorrect
- glGetMaterial() didn't work reliably
- ARB_fragment_program KIL instruction was incorrect

4.282.47 6.1 August 18, 2004

New:

- Revamped Makefile system
- glXUseRotatedXFont() utility (see xdemos/xuserotfont.c)
- internal driver interface changes related to texture object allocation, vertex/fragment programs, BlendEquationSeparate, etc.
- option to walk triangle edges with double-precision floats (Justin Novosad of Discreet) (see config.h file)
- support for AUX buffers in software GLX driver
- updated glext.h to version 24 and glxext.h to version 6
- new MESA_GLX_FORCE_ALPHA and MESA_GLX_DEPTH_BITS env vars
- updated BeOS support (Philippe Houdoin)

Changes:

- fragment fog interpolation is perspective corrected now
- new glTexImage code, much cleaner, may be a bit faster

Bug fixes:

- glArrayElement in display lists didn't handle generic vertex attribs
- glFogCoord didn't always work properly
- ARB_fragment_program fog options didn't work
- frag prog TEX instruction no longer incorrectly divides s,t,r by q
- ARB frag prog TEX and TEXP instructions now use LOD=0
- glTexEnviv in display lists didn't work
- glRasterPos didn't do texgen or apply texture matrix
- GL_DOUBLE-valued vertex arrays were broken in some cases
- fixed texture rectangle edge/border sampling bugs
- sampling an incomplete texture in a fragment program would segfault
- glTexImage was missing a few error checks
- fixed some minor glGetTexParameter glitches
- GL_INTENSITY was mistakenly accepted as a <format> to glTexImage
- fragment program writes to RC/HC register were broken
- fixed a few glitches in GL_HP_occlusion_test extension
- glBeginQueryARB and glEndQueryARB didn't work inside display lists
- vertex program state references were broken
- fixed triangle color interpolation bug on AIX (Shane Blackett)
- fixed a number of minor memory leaks (bug #1002030)

4.282.48 6.2 October 2, 2004

New:

- enabled GL_ARB_texture_rectangle (same as GL_NV_texture_rectangle)
- updated Doxygen support (Jose Fonseca)

Changes:

- some GGI driver updates (Christoph Egger, bug 1025977)

Bug fixes:

- Omit GL_ARB_texture_non_power_of_two from list of OpenGL 1.5 features
- fixed a few compilation issues on IRIX
- fixed a matrix classification bug (reported by Wes Bethel)
- we weren't reseting the vertex/fragment program error state before parsing (Dave Reveman)
- adjust texcoords for sampling texture rectangles (Dave Reveman)
- glGet*(GL_MAX_VERTEX_ATTRIBS_ARB) wasn't implemented
- repeated calls to glDeleteTexture(t) could lead to a crash
- fixed potential ref count bugs in VBOs and vertex/fragment programs
- spriteblast demo didn't handle window size changes correctly
- glTexSubImage didn't handle pixels=NULL correctly for PBOs
- fixed color index mode glDrawPixels bug (Karl Schultz)

4.282.49 6.2.1 December 9, 2004

Bug fixes:

- don't apply regular fog or color sum when using a fragment program
- glProgramEnvParameter4fARB always generated an error on GL_FRAGMENT_PROGRAM_ARB (fdo bug 1645)
- glVertexAttrib3svNV and glVertexAttrib3svARB were broken
- fixed width/height mix-up in glSeparableFilter2D()
- fixed regression in glCopyPixels + convolution
- glReadPixels from a clipped front color buffer didn't always work
- glTexImage didn't accept GL_RED/GREEN/BLUE as the format
- Attempting queries/accesses of VBO 0 weren't detected as errors
- paletted textures failed if the palette had fewer than 256 entries

Changes:

- fixed a bunch of compiler warnings found with gcc 3.4
- bug reports should to go bugzilla.freedesktop.org

4.282.50 6.3 July 20, 2005

New:

- GL_EXT_framebuffer_object extension
- GL_ARB_draw_buffers extension
- GL_ARB_pixel_buffer_object extension
- GL_OES_read_format extension (Ian Romanick)
- DirectFB driver (Claudio Ciccani)
- x86_64 vertex transformation code (Mikko T.)
- Updated GL/glext.h to version 29

Changes:

- added -stereo option for glxgears demo (Jacek Rosik)
- updated the PBuffer demo code in xdemos/ directory
- glDeleteTextures/Programs/Buffers() now makes the object ID available for immediate re-use
- assorted 64-bit clean-ups fixes (x86_64 and Win64)
- lots of internal changes for GL_EXT_framebuffer_object

Bug fixes:

- some functions didn't support PBO functionality
- glGetTexImage didn't convert color index images to RGBA as required
- fragment program texcoords were sometimes wrong for points and lines
- fixed problem with negative dot product in arbfplight, fplight demos
- fixed bug in perspective correction of antialiased, textured lines
- querying GL_POST_CONVOLUTION_ALPHA_BIAS_EXT returned wrong value
- fixed a couple per-pixel fog bugs (Soju Matsumoto)
- glGetBooleanv(GL_FRAGMENT_PROGRAM_BINDING_NV) was broken
- fixed float parsing bug in ARB frag/vert programs (bug 2520)
- XMesaGetDepthBuffer() returned incorrect value for bytesPerValue
- GL_COLOR_MATERIAL with glColor3 didn't properly set diffuse alpha
- glXChooseFBConfig() crashed if attribList pointer was NULL
- program state.light[n].spot.direction.w was wrong value (bug 3083)
- fragment program fog option required glEnable(GL_FOG) - wrong.
- glColorTable() could produce a Mesa implementation error (bug 3135)
- RasterPos could get corrupted by color index rendering path
- Removed bad XTranslateCoordinates call when rendering to Pixmaps
- glPopAttrib() didn't properly restore GL_TEXTURE_GEN enable state
- fixed a few Darwin compilation problems

4.282.51 6.3.1

This was an intermediate release for X.org which wasn't otherwise released.)

4.282.52 6.3.2 August 19, 2005

New:

- The distribution now includes the DRI drivers and GLX code

Changes:

- Made the DRI "new" driver interface standard, remove old code

Bug fixes:

- GL_ARB_vertex/fragment_shader were mistakenly listed in the extensions string
- negative relative addressing in vertex programs was broken
- update/fix SPARC assembly code for vertex transformation
- fixed memory leak when freeing GLX drawables/renderbuffers
- fixed display list memory leak
- the GL_PIXEL_MAP_I_TO_I table is now floating point, not integer
- wglGetProcAddress() didn't handle wgl-functions
- fixed glxext.h cross-compile issue (Colin Harrison)
- assorted DRI driver fixes

Note: Changes for Mesa 6.4 and later are documented in the corresponding *release notes* file.

CHAPTER 5

Acknowledgements

The following individuals and groups are to be acknowledged for their contributions to Mesa over the years. This list is far from complete and somewhat dated, unfortunately.

- Early Mesa development was done while Brian was part of the SSEC Visualization Project at the University of Wisconsin. He'd like to thank Bill Hibbard for letting him work on Mesa as part of that project.
- John Carmack of id Software, Inc. funded Keith Whitwell in 1999 in order to optimize Mesa's vertex transformation module. This is a very substantial piece of work.
- Precision Insight, Inc., VA Linux Systems, Inc., and most recently, Tungsten Graphics, Inc. have supported the ongoing development of Mesa.
- The Mesa website and Git repository are hosted by freedesktop.org.
- alt.software contributed the Direct3D driver.
- **Bernd Barsuhn** wrote the evaluator code for (splines, patches) in Mesa.
- **Bernhard Tschirren** wrote the Allegro DJGPP driver.
- **Bogdan Sikorski** wrote the GLU NURBS and polygon tessellator in Mesa.
- **Charlie Wallace** wrote the MS-DOS driver.
- **CJ Beyer** was the www.mesa3d.org webmaster.
- **Darren Abbott** provided the OS/2 driver.
- **David Bucciarelli** wrote and maintained the 3Dfx Glide driver. Thousands of Linux/Quake players thank David!
- **Gareth Hughes** wrote new GLU 1.2 Polygon Tessellation code (now superseded by SGI SI GLU).
- **Holger Waechtler** contributed AMD 3DNow! assembly code which accelerates vertex transformation in Mesa 3.1. Holger also implemented the GL_EXT_texture_env_combine extension.
- **Jeroen van der Zijp** and **Thorsten Ohl** contributed the Xt/Motif widget code.
- **John Stone** provided the multi-threading support in Mesa 3.0.
- **John Watson** assisted with web page design.

- **Josh Vanderhoof** contributed Intel x86 assembly code which accelerates vertex transformation in Mesa 3.x.
- **Jouk Jansen** contributed and continues to maintain the VMS support.
- **Karl Schultz** has been maintaining the Windows driver.
- **Keith Whitwell** has made extension contributions to Mesa since 1999.
- **Kendall Bennett** wrote the SciTech MGL driver.
- **Klaus Niederkrueger** contributed many improvements to Mesa's software rasterizer.
- **Mark Kilgard** contributed antialiased line improvements and several extensions.
- **Michael Pichler** contributed *many* bug fixes
- **Miklos Fazekas** wrote and maintains the Macintosh driver.
- **Pascal Thibaudeau** wrote the NeXT driver.
- **Pedro Vazquez** setup and maintains the Mesa Mailing list.
- **Randy Frank** contributed *many* bug fixes.
- **Stefan Zivkovic** wrote the Amiga driver.
- **Stephane Rehel** provided the Cygnus Win32 support
- **Ted Jump** maintained the makefiles and project files for Windows 95/98/NT compilation for some time.
- **Uwe Maurer** wrote the LibGGI driver for Mesa-3.0.
- **Victor Ng-Thow-Hing** wrote the Amiwin driver for the Amiga.

Apologies to anyone who's been omitted. Please send corrections and additions to Brian.

CHAPTER 6

Downloading and Unpacking

6.1 Downloading

You can download the released versions of Mesa via HTTPS or FTP.

Our release tarballs are GPG-signed, and the public keys are available here: release-maintainers-keys.asc.

Starting with the first release of 2017, Mesa's version scheme is year-based. Filenames are in the form `mesa-Y.N.P.tar.gz`, where `Y` is the year (two digits), `N` is an incremental number (starting at 0) and `P` is the patch number (0 for the first release, 1 for the first patch after that).

When a new release is coming, release candidates (betas) may be found in the same directory, and are recognizable by the `mesa-Y.N.P-rcX.tar.gz` filename.

6.2 Unpacking

Mesa releases are available in two formats: `.tar.xz` and `.tar.gz`.

To unpack the tarball:

```
tar xf mesa-Y.N.P.tar.xz
```

or

```
tar xf mesa-Y.N.P.tar.gz
```

6.3 Contents

Proceed to the *compilation and installation instructions*.

6.4 Demos, GLUT, and GLU

A package of SGI's GLU library is available here

A package of Mark Kilgard's GLUT library is available here

The Mesa demos collection is available here

In the past, GLUT, GLU and the Mesa demos were released in conjunction with Mesa releases. But since GLUT, GLU and the demos change infrequently, they were split off into their own Git repositories: GLUT, GLU and Demos,

CHAPTER 7

Compiling and Installing

7.1 Compilation and Installation Using Meson

7.1.1 1. Introduction

For general information about Meson see the Meson website.

Mesa's Meson build system is generally considered stable and ready for production.

Note: Mesa requires Meson >= 0.52.0 to build.

If your distribution doesn't have something recent enough in its repositories, you can try the methods suggested here to install the current version of Meson.

The Meson build of Mesa is tested on Linux, macOS, Windows, Cygwin, Haiku, FreeBSD, DragonflyBSD, NetBSD, and should work on OpenBSD.

Unix-like OSes

If Meson is not already installed on your system, you can typically install it with your package installer. For example:

```
sudo apt-get install meson   # Ubuntu
```

or

```
sudo dnf install meson   # Fedora
```

Some older versions of Meson do not check that they are too old and will error out in odd ways.

You'll also need Ninja. If it's not already installed, use apt-get or dnf to install the *ninja-build* package.

Windows

You will need to install Python 3 and Meson as a module using pip. This is because we use Python for generating code, and rely on external modules (Mako). You also need pkg-config (a hard dependency of Meson), Flex, and Bison. The easiest way to install everything you need is with Chocolatey.

```
choco install python3 winflexbison pkgconfiglite
```

You can even use Chocolatey to install MinGW and Ninja (Ninja can be used with MSVC as well)

```
choco install ninja mingw
```

Then install Meson using pip

```
py -3 -m pip install meson mako
```

You may need to add the Python 3 scripts directory to your path for Meson.

7.1.2 2. Basic Usage

The Meson program is used to configure the source directory and generates either a Ninja build file or Visual Studio® build files. The latter must be enabled via the `--backend` switch, as Ninja is the default backend on all operating systems.

Meson only supports out-of-tree builds, and must be passed a directory to put built and generated sources into. We'll call that directory "build" here. It's recommended to create a separate build directory for each configuration you might want to use.

Basic configuration is done with:

```
meson build/
```

This will create the build directory. If any dependencies are missing, you can install them, or try to remove the dependency with a Meson configuration option (see below).

To review the options which Meson chose, run:

```
meson configure build/
```

Meson does not currently support listing configuration options before running "meson build/" but this feature is being discussed upstream. For now, we have a `bin/meson-options.py` script that prints the options for you. If that script doesn't work for some reason, you can always look in the meson_options.txt file at the root of the project.

With additional arguments `meson configure` can be used to change options for a previously configured build directory. All options passed to this command are in the form `-D "option"="value"`. For example:

```
meson configure build/ -Dprefix=/tmp/install -Dglx=true
```

Note that options taking lists (such as `platforms`) are a bit more complicated, but the simplest form compatible with Mesa options is to use a comma to separate values (`-D platforms=drm,wayland`) and brackets to represent an empty list (`-D platforms=[]`).

Once you've run the initial `meson` command successfully you can use your configured backend to build the project in your build directory:

```
ninja -C build/
```

The next step is to install the Mesa libraries, drivers, etc. This also finishes up some final steps of the build process (such as creating symbolic links for drivers). To install:

```
ninja -C build/ install
```

Windows specific instructions

On Windows you have a couple of choices for compilers. If you installed MinGW with Chocolatey and want to use Ninja you should be able to open any shell and follow the instructions above. If you want to you MSVC, clang-cl, or ICL (the Intel Compiler), read on.

Both ICL and MSVC come with shell environments, the easiest way to use Meson with these it to open a shell. For clang-cl you will need to open an MSVC shell, and then override the compilers, either using a native file, or with the CC and CXX environment variables.

All of these compilers are tested and work with Ninja, but if you want Visual Studio integration or you just like msbuild, passing `--backend=vs` to Meson will generate a Visual Studio solution. If you want to use ICL or clang-cl with the vsbackend you will need Meson 0.52.0 or greater. Older versions always use the Microsoft compiler.

7.1.3 3. Advanced Usage

Installation Location

Meson default to installing libGL.so in your system's main lib/ directory and DRI drivers to a dri/ subdirectory.

Developers will often want to install Mesa to a testing directory rather than the system library directory. This can be done with the –prefix option. For example:

```
meson --prefix="${PWD}/build/install" build/
```

will put the final libraries and drivers into the build/install/ directory. Then you can set LD_LIBRARY_PATH and LIBGL_DRIVERS_PATH to that location to run/test the driver.

Meson also honors `DESTDIR` for installs.

Compiler Options

Meson supports the common CFLAGS, CXXFLAGS, etc. environment variables but their use is discouraged because of the many caveats in using them.

Instead, it is recommended to use `-D${lang}_args` and `-D${lang}_link_args`. Among the benefits of these options is that they are guaranteed to persist across rebuilds and reconfigurations.

This example sets -fmax-errors for compiling C sources and -DMAGIC=123 for C++ sources:

```
meson builddir/ -Dc_args=-fmax-errors=10 -Dcpp_args=-DMAGIC=123
```

Compiler Specification

Meson supports the standard CC and CXX environment variables for changing the default compiler. Note that Meson does not allow changing the compilers in a configured builddir so you will need to create a new build dir for a different compiler.

This is an example of specifying the clang compilers and cleaning the build directory before reconfiguring with an extra C option:

```
CC=clang CXX=clang++ meson build-clang
ninja -C build-clang
ninja -C build-clang clean
meson configure build -Dc_args="-Wno-typedef-redefinition"
ninja -C build-clang
```

The default compilers depends on your operating system. Meson supports most of the popular compilers, a complete list is available here.

LLVM

Meson includes upstream logic to wrap llvm-config using its standard dependency interface.

As of Meson 0.51.0 Meson can use CMake to find LLVM (the CMake finder was added in Meson 0.49.0, but LLVM cannot be found until 0.51) Due to the way LLVM implements its CMake finder it will only find static libraries, it will never find libllvm.so. There is also a `-Dcmake_module_path` option in this Meson version, which points to the root of an alternative installation (the prefix). For example:

```
meson builddir -Dcmake_module_path=/home/user/mycmake/prefix
```

As of Meson 0.49.0 Meson also has the concept of a "native file", these files provide information about the native build environment (as opposed to a cross build environment). They are ini formatted and can override where to find llvm-config:

custom-llvm.ini

```
[binaries]
llvm-config = '/usr/local/bin/llvm/llvm-config'
```

Then configure Meson:

```
meson builddir/ --native-file custom-llvm.ini
```

Meson < 0.49 doesn't support native files, so to specify a custom `llvm-config` you need to modify your `$PATH` (or `%PATH%` on Windows), which will be searched for `llvm-config`, `llvm-config$version`, and `llvm-config-$version`:

```
PATH=/path/to/folder/with/llvm-config:$PATH meson build
```

For selecting llvm-config for cross compiling a "cross file" should be used. It uses the same format as the native file above:

cross-llvm.ini

```
[binaries]
...
llvm-config = '/usr/lib/llvm-config-32'
cmake = '/usr/bin/cmake-for-my-arch'
```

Obviously, only cmake or llvm-config is required.

Then configure Meson:

```
meson builddir/ --cross-file cross-llvm.ini
```

See the *Cross Compilation* section for more information.

On Windows (and in other cases), using llvm-config or CMake may be either undesirable or impossible. Meson's solution for this is a wrap, in this case a "binary wrap". Follow the steps below:

- Install the binaries and headers into the `$mesa_src/subprojects/llvm`
- Add a meson.build file to that directory (more on that later)

The wrap file must define the following:

- `dep_llvm`: a `declare_dependency()` object with include_directories, dependencies, and version set)

It may also define:

- `irbuilder_h`: a `files()` object pointing to llvm/IR/IRBuilder.h (this is required for SWR)
- `has_rtti`: a `bool` that declares whether LLVM was built with RTTI. Defaults to true

such a meson.build file might look like:

```
project('llvm', ['cpp'])

cpp = meson.get_compiler('cpp')

_deps = []
_search = join_paths(meson.current_source_dir(), 'lib')
foreach d : ['libLLVMCodeGen', 'libLLVMScalarOpts', 'libLLVMAnalysis',
             'libLLVMTransformUtils', 'libLLVMCore', 'libLLVMX86CodeGen',
             'libLLVMSelectionDAG', 'libLLVMipo', 'libLLVMAsmPrinter',
             'libLLVMInstCombine', 'libLLVMInstrumentation', 'libLLVMMC',
             'libLLVMGlobalISel', 'libLLVMObjectYAML', 'libLLVMDebugInfoPDB',
             'libLLVMVectorize', 'libLLVMPasses', 'libLLVMSupport',
             'libLLVMLTO', 'libLLVMObject', 'libLLVMDebugInfoCodeView',
             'libLLVMDebugInfoDWARF', 'libLLVMOrcJIT', 'libLLVMProfileData',
             'libLLVMObjCARCOpts', 'libLLVMBitReader', 'libLLVMCoroutines',
             'libLLVMBitWriter', 'libLLVMRuntimeDyld', 'libLLVMMIRParser',
             'libLLVMX86Desc', 'libLLVMAsmParser', 'libLLVMTableGen',
             'libLLVMFuzzMutate', 'libLLVMLinker', 'libLLVMMCParser',
             'libLLVMExecutionEngine', 'libLLVMCoverage', 'libLLVMInterpreter',
             'libLLVMTarget', 'libLLVMX86AsmParser', 'libLLVMSymbolize',
             'libLLVMDebugInfoMSF', 'libLLVMMCJIT', 'libLLVMXRay',
             'libLLVMX86AsmPrinter', 'libLLVMX86Disassembler',
             'libLLVMMCDisassembler', 'libLLVMOption', 'libLLVMIRReader',
             'libLLVMLibDriver', 'libLLVMDlltoolDriver', 'libLLVMDemangle',
             'libLLVMBinaryFormat', 'libLLVMLineEditor',
             'libLLVMWindowsManifest', 'libLLVMX86Info', 'libLLVMX86Utils']
  _deps += cpp.find_library(d, dirs : _search)
endforeach

dep_llvm = declare_dependency(
  include_directories : include_directories('include'),
  dependencies : _deps,
  version : '6.0.0',
)

has_rtti = false
irbuilder_h = files('include/llvm/IR/IRBuilder.h')
```

It is very important that version is defined and is accurate, if it is not, workarounds for the wrong version of LLVM might be used resulting in build failures.

PKG_CONFIG_PATH

The `pkg-config` utility is a hard requirement for configuring and building Mesa on Unix-like systems. It is used to search for external libraries on the system. This environment variable is used to control the search path for `pkg-config`. For instance, setting `PKG_CONFIG_PATH=/usr/X11R6/lib/pkgconfig` will search for package metadata in `/usr/X11R6` before the standard directories.

Options

One of the oddities of Meson is that some options are different when passed to the `meson` than to `meson configure`. These options are passed as –option=foo to `meson`, but -Doption=foo to `meson configure`. Mesa defined options are always passed as -Doption=foo.

For those coming from autotools be aware of the following:

--buildtype/-Dbuildtype This option will set the compiler debug/optimisation levels to aid debugging the Mesa libraries.

Note that in Meson this defaults to `debugoptimized`, and not setting it to `release` will yield non-optimal performance and binary size. Not using `debug` may interfere with debugging as some code and validation will be optimized away.

For those wishing to pass their own optimization flags, use the `plain` buildtype, which causes Meson to inject no additional compiler arguments, only those in the C/CXXFLAGS and those that mesa itself defines.

-Db_ndebug This option controls assertions in Meson projects. When set to `false` (the default) assertions are enabled, when set to true they are disabled. This is unrelated to the `buildtype`; setting the latter to `release` will not turn off assertions.

7.1.4 4. Cross-compilation and 32-bit builds

Meson supports cross-compilation by specifying a number of binary paths and settings in a file and passing this file to `meson` or `meson configure` with the `--cross-file` parameter.

This file can live at any location, but you can use the bare filename (without the folder path) if you put it in $XDG_DATA_HOME/meson/cross or ~/.local/share/meson/cross

Below are a few example of cross files, but keep in mind that you will likely have to alter them for your system.

Those running on ArchLinux can use the AUR-maintained packages for some of those, as they'll have the right values for your system:

- meson-cross-x86-linux-gnu
- meson-cross-aarch64-linux-gnu

32-bit build on x86 linux:

```
[binaries]
c = '/usr/bin/gcc'
cpp = '/usr/bin/g++'
ar = '/usr/bin/gcc-ar'
strip = '/usr/bin/strip'
pkgconfig = '/usr/bin/pkg-config-32'
llvm-config = '/usr/bin/llvm-config32'

[properties]
c_args = ['-m32']
```

(continues on next page)

(continued from previous page)

```
c_link_args = ['-m32']
cpp_args = ['-m32']
cpp_link_args = ['-m32']

[host_machine]
system = 'linux'
cpu_family = 'x86'
cpu = 'i686'
endian = 'little'
```

64-bit build on ARM linux:

```
[binaries]
c = '/usr/bin/aarch64-linux-gnu-gcc'
cpp = '/usr/bin/aarch64-linux-gnu-g++'
ar = '/usr/bin/aarch64-linux-gnu-gcc-ar'
strip = '/usr/bin/aarch64-linux-gnu-strip'
pkgconfig = '/usr/bin/aarch64-linux-gnu-pkg-config'
exe_wrapper = '/usr/bin/qemu-aarch64-static'

[host_machine]
system = 'linux'
cpu_family = 'aarch64'
cpu = 'aarch64'
endian = 'little'
```

64-bit build on x86 Windows:

```
[binaries]
c = '/usr/bin/x86_64-w64-mingw32-gcc'
cpp = '/usr/bin/x86_64-w64-mingw32-g++'
ar = '/usr/bin/x86_64-w64-mingw32-ar'
strip = '/usr/bin/x86_64-w64-mingw32-strip'
pkgconfig = '/usr/bin/x86_64-w64-mingw32-pkg-config'
exe_wrapper = 'wine'

[host_machine]
system = 'windows'
cpu_family = 'x86_64'
cpu = 'i686'
endian = 'little'
```

7.2 1. Prerequisites for building

7.2.1 1.1 General

Build system

- Meson is required when building on *nix platforms and is supported on Windows.
- SCons is an alternative for building on Windows and Linux.
- Android Build system when building as native Android component. Meson is used when when building ARC.

Compiler

The following compilers are known to work, if you know of others or you're willing to maintain support for other compiler get in touch.

- GCC 4.2.0 or later (some parts of Mesa may require later versions)
- clang - exact minimum requirement is currently unknown.
- Microsoft Visual Studio 2015 or later is required, for building on Windows.

Third party/extra tools.

- Python - Python is required. When building with SCons 2.7 is required. When building with meson 3.5 or newer is required.
- Python Mako module - Python Mako module is required. Version 0.8.0 or later should work.
- lex / yacc - for building the Mesa IR and GLSL compiler.

  On Linux systems, Flex and Bison versions 2.5.35 and 2.4.1, respectively, (or later) should work. On Windows with MinGW, install Flex and Bison with:

  ```
  mingw-get install msys-flex msys-bison
  ```

  For MSVC on Windows, install Win flex-bison.

Note: Some versions can be buggy (eg. Flex 2.6.2) so do try others if things fail.

7.2.2 1.2 Requirements

The requirements depends on the features selected at configure stage. Check/install the respective -devel package as prompted by the configure error message.

Here are some common ways to retrieve most/all of the dependencies based on the packaging tool used by your distro.

```
zypper source-install --build-deps-only Mesa # openSUSE/SLED/SLES
yum-builddep mesa # yum Fedora, OpenSuse(?)
dnf builddep mesa # dnf Fedora
apt-get build-dep mesa # Debian and derivatives
... # others
```

7.3 2. Building with meson

Meson >= 0.46.0 is required

Meson is the latest build system in mesa, it is currently able to build for *nix systems like Linux and BSD, macOS, Haiku, and Windows.

The general approach is:

```
meson builddir/
ninja -C builddir/
sudo ninja -C builddir/ install
```

On Windows you can also use the Visual Studio backend

```
meson builddir --backend=vs
cd builddir
msbuild mesa.sln /m
```

Please read the *detailed meson instructions* for more information

7.4 3. Building with SCons (Windows/Linux)

To build Mesa with SCons on Linux or Windows do

```
scons
```

The build output will be placed in build/*platform-machine-debug*/..., where *platform* is for example Linux or Windows, *machine* is x86 or x86_64, optionally followed by -debug for debug builds.

To build Mesa with SCons for Windows on Linux using the MinGW crosscompiler toolchain do

```
scons platform=windows toolchain=crossmingw machine=x86 libgl-gdi
```

This will create:

- build/windows-x86-debug/gallium/targets/libgl-gdi/opengl32.dll — Mesa + Gallium + softpipe (or llvmpipe), binary compatible with Windows's opengl32.dll

Put them all in the same directory to test them. Additional information is available in README.WIN32.

7.5 4. Building with AOSP (Android)

Currently one can build Mesa for Android as part of the AOSP project, yet your experience might vary.

In order to achieve that one should update their local manifest to point to the upstream repo, set the appropriate BOARD_GPU_DRIVERS and build the libGLES_mesa library.

FINISHME: Improve on the instructions add references to Rob H repos/Jenkins, Android-x86 and/or other resources.

7.6 5. Library Information

When compilation has finished, look in the top-level `lib/` (or `lib64/`) directory. You'll see a set of library files similar to this:

```
lrwxrwxrwx    1 brian    users          10 Mar 26 07:53 libGL.so -> libGL.so.1*
lrwxrwxrwx    1 brian    users          19 Mar 26 07:53 libGL.so.1 -> libGL.so.1.5.
↪060100*
-rwxr-xr-x    1 brian    users     3375861 Mar 26 07:53 libGL.so.1.5.060100*
lrwxrwxrwx    1 brian    users          14 Mar 26 07:53 libOSMesa.so -> libOSMesa.so.
↪6*
lrwxrwxrwx    1 brian    users          23 Mar 26 07:53 libOSMesa.so.6 -> libOSMesa.
↪so.6.1.060100*
-rwxr-xr-x    1 brian    users       23871 Mar 26 07:53 libOSMesa.so.6.1.060100*
```

libGL is the main OpenGL library (i.e. Mesa), while **libOSMesa** is the OSMesa (Off-Screen) interface library.

If you built the DRI hardware drivers, you'll also see the DRI drivers:

```
-rwxr-xr-x   1 brian users 16895413 Jul 21 12:11 i915_dri.so
-rwxr-xr-x   1 brian users 16895413 Jul 21 12:11 i965_dri.so
-rwxr-xr-x   1 brian users 11849858 Jul 21 12:12 r200_dri.so
-rwxr-xr-x   1 brian users 11757388 Jul 21 12:12 radeon_dri.so
```

If you built with Gallium support, look in lib/gallium/ for Gallium-based versions of libGL and device drivers.

7.7 6. Building OpenGL programs with pkg-config

Running `ninja install` will install package configuration files for the pkg-config utility.

When compiling your OpenGL application you can use pkg-config to determine the proper compiler and linker flags.

For example, compiling and linking a GLUT application can be done with:

```
gcc `pkg-config --cflags --libs glut` mydemo.c -o mydemo
```

CHAPTER 8

Precompiled Libraries

In general, precompiled Mesa libraries are not available.

Some Linux distributions closely follow the latest Mesa releases. On others one has to use unofficial channels.

There are some general directions:

- Debian/Ubuntu based distros - PPA: xorg-edgers, oibaf and padoka
- Fedora - Corp: erp and che
- OpenSuse/SLES - OBS: X11:XOrg and pontostroy:X11
- Gentoo/Archlinux - officially provided/supported

CHAPTER 9

Mailing Lists

These are the primary Mesa 3D / DRI mailing lists:

- mesa-users - intended for end-users of Mesa and DRI drivers. Newbie questions are OK, but please try the general OpenGL resources and Mesa/DRI documentation first.
- mesa-dev - for Mesa, Gallium and DRI development discussion. Not for beginners.
- mesa-commit - relays Git check-in messages (for developers). In general, people should not post to this list.
- mesa-announce - announcements of new Mesa versions are sent to this list. Very low traffic.
- dri-devel - for the Linux/BSD Direct Rendering Module (DRM) kernel modules supporting Mesa's userspace drivers
- piglit - for Piglit (OpenGL driver testing framework) discussion.

Note: You **must** subscribe to these lists in order to post to them. If you try to post to a list and you're not a subscriber (or if you try to post from a different email address than you subscribed with) your posting will be held for an indefinite period or may be discarded entirely.

Follow the links above for list archives.

The old Mesa lists hosted at SourceForge are no longer in use. The archives are still available, however: mesa3d-announce, mesa3d-users, mesa3d-dev.

9.1 IRC

join #dri-devel channel on irc.freenode.net

9.2 OpenGL Forums

Here are some other OpenGL-related forums you might find useful:

- OpenGL discussion forums at www.opengl.org
- Usenet newsgroups:
  - comp.graphics.algorithms
  - comp.graphics.api.opengl
  - comp.os.linux.x

CHAPTER 10

Report a Bug

The Mesa bug database is hosted on freedesktop.org. The old bug database on SourceForge is no longer used.

To file a Mesa bug, go to GitLab on freedesktop.org

Please follow these bug reporting guidelines:

- Check if a new version of Mesa is available which might have fixed the problem.
- Check if your bug is already reported in the database.
- Monitor your bug report for requests for additional information, etc.
- Attach the output of running glxinfo or wglinfo. This will tell us the Mesa version, which device driver you're using, etc.
- If you're reporting a crash, try to use your debugger (gdb) to get a stack trace. Also, recompile Mesa in debug mode to get more detailed information.
- Describe in detail how to reproduce the bug, especially with games and applications that the Mesa developers might not be familiar with.
- Provide an apitrace or simple GLUT-based test program if possible.

The easier a bug is to reproduce, the sooner it will be fixed. Please do everything you can to facilitate quickly fixing bugs. If your bug report is vague or your test program doesn't compile easily, the problem may not be fixed very quickly.

CHAPTER 11

Shading Language

This page describes the features and status of Mesa's support for the OpenGL Shading Language.

11.1 Environment Variables

The **MESA_GLSL** environment variable can be set to a comma-separated list of keywords to control some aspects of the GLSL compiler and shader execution. These are generally used for debugging.

- **dump** - print GLSL shader code to stdout at link time
- **log** - log all GLSL shaders to files. The filenames will be "shader_X.vert" or "shader_X.frag" where X the shader ID.
- **cache_info** - print debug information about shader cache
- **cache_fb** - force cached shaders to be ignored and do a full recompile via the fallback path
- **uniform** - print message to stdout when glUniform is called
- **nopvert** - force vertex shaders to be a simple shader that just transforms the vertex position with ftransform() and passes through the color and texcoord[0] attributes.
- **nopfrag** - force fragment shader to be a simple shader that passes through the color attribute.
- **useprog** - log glUseProgram calls to stderr
- **errors** - GLSL compilation and link errors will be reported to stderr.

Example: export MESA_GLSL=dump,nopt

11.1.1 Experimenting with Shader Replacements

Shaders can be dumped and replaced on runtime for debugging purposes. This feature is not currently supported by SCons build. This is controlled via following environment variables:

- **MESA_SHADER_DUMP_PATH** - path where shader sources are dumped

- **MESA_SHADER_READ_PATH** - path where replacement shaders are read

Note, path set must exist before running for dumping or replacing to work. When both are set, these paths should be different so the dumped shaders do not clobber the replacement shaders. Also, the filenames of the replacement shaders should match the filenames of the corresponding dumped shaders.

11.1.2 Capturing Shaders

Setting **MESA_SHADER_CAPTURE_PATH** to a directory will cause the compiler to write `.shader_test` files for use with shader-db, a tool which compiler developers can use to gather statistics about shaders (instructions, cycles, memory accesses, and so on).

Notably, this captures linked GLSL shaders - with all stages together - as well as ARB programs.

11.2 GLSL Version

The GLSL compiler currently supports version 3.30 of the shading language.

Several GLSL extensions are also supported:

- GL_ARB_draw_buffers
- GL_ARB_fragment_coord_conventions
- GL_ARB_shader_bit_encoding

11.3 Unsupported Features

XXX update this section

The following features of the shading language are not yet fully supported in Mesa:

- Linking of multiple shaders does not always work. Currently, linking is implemented through shader concatenation and re-compiling. This doesn't always work because of some #pragma and preprocessor issues.
- The gl_Color and gl_SecondaryColor varying vars are interpolated without perspective correction

All other major features of the shading language should function.

11.4 Implementation Notes

- Shading language programs are compiled into low-level programs very similar to those of GL_ARB_vertex/fragment_program.
- All vector types (vec2, vec3, vec4, bvec2, etc) currently occupy full float[4] registers.
- Float constants and variables are packed so that up to four floats can occupy one program parameter/register.
- All function calls are inlined.
- Shaders which use too many registers will not compile.
- The quality of generated code is pretty good, register usage is fair.
- Shader error detection and reporting of errors (InfoLog) is not very good yet.
- The ftransform() function doesn't necessarily match the results of fixed-function transformation.

These issues will be addressed/resolved in the future.

11.5 Programming Hints

- Use the built-in library functions whenever possible. For example, instead of writing this:

```
float x = 1.0 / sqrt(y);
```

  Write this:

```
float x = inversesqrt(y);
```

11.6 Stand-alone GLSL Compiler

The stand-alone GLSL compiler program can be used to compile GLSL shaders into low-level GPU code.

This tool is useful for:

- Inspecting GPU code to gain insight into compilation
- Generating initial GPU code for subsequent hand-tuning
- Debugging the GLSL compiler itself

After building Mesa, the compiler can be found at src/compiler/glsl/glsl_compiler

Here's an example of using the compiler to compile a vertex shader and emit GL_ARB_vertex_program-style instructions:

```
src/compiler/glsl/glsl_compiler --version XXX --dump-ast myshader.vert
```

Options include

- **–dump-ast** - dump GPU code
- **–dump-hir** - dump high-level IR code
- **–dump-lir** - dump low-level IR code
- **–dump-builder** - dump GLSL IR code
- **–link** - link shaders
- **–just-log** - display only shader / linker info if exist, without any header or separator
- **–version** - [Mandatory] define the GLSL version to use

11.7 Compiler Implementation

The source code for Mesa's shading language compiler is in the `src/compiler/glsl/` directory.

XXX provide some info about the compiler. . . .

The final vertex and fragment programs may be interpreted in software (see prog_execute.c) or translated into a specific hardware architecture (see drivers/dri/i915/i915_fragprog.c for example).

11.8 Compiler Validation

Developers working on the GLSL compiler should test frequently to avoid regressions.

The Piglit project has many GLSL tests.

The Mesa demos repository also has some good GLSL tests.

CHAPTER 12

EGL

The current version of EGL in Mesa implements EGL 1.4. More information about EGL can be found at https://www.khronos.org/egl/.

The Mesa's implementation of EGL uses a driver architecture. The main library (`libEGL`) is window system neutral. It provides the EGL API entry points and helper functions for use by the drivers. Drivers are dynamically loaded by the main library and most of the EGL API calls are directly dispatched to the drivers.

The driver in use decides the window system to support.

12.1 Build EGL

1. Configure your build with the desired client APIs and enable the driver for your hardware. For example:

```
$ meson configure \
        -D egl=true \
        -D gles1=true \
        -D gles2=true \
        -D dri-drivers=... \
        -D gallium-drivers=...
```

   The main library and OpenGL is enabled by default. The first two options above enables *OpenGL ES 1.x and 2.x*. The last two options enables the listed classic and Gallium drivers respectively.

2. Build and install Mesa as usual.

In the given example, it will build and install `libEGL`, `libGL`, `libGLESv1_CM`, `libGLESv2`, and one or more EGL drivers.

12.1.1 Configure Options

There are several options that control the build of EGL at configuration time

`-D egl=true` By default, EGL is enabled. When disabled, the main library and the drivers will not be built.

-D platforms=... List the platforms (window systems) to support. Its argument is a comma separated string such as `-D platforms=x11,wayland`. It decides the platforms a driver may support. The first listed platform is also used by the main library to decide the native platform.

The available platforms are `x11`, `wayland`, `android`, and `haiku`. The `android` platform can either be built as a system component, part of AOSP, using `Android.mk` files, or cross-compiled using appropriate options. Unless for special needs, the build system should select the right platforms automatically.

-D gles1=true and -D gles2=true These options enable OpenGL ES support in OpenGL. The result is one big internal library that supports multiple APIs.

-D shared-glapi=true By default, `libGL` has its own copy of `libglapi`. This options makes `libGL` use the shared `libglapi`. This is required if applications mix OpenGL and OpenGL ES.

12.2 Use EGL

12.2.1 Demos

There are demos for the client APIs supported by EGL. They can be found in mesa/demos repository.

12.2.2 Environment Variables

There are several environment variables that control the behavior of EGL at runtime

EGL_PLATFORM This variable specifies the native platform. The valid values are the same as those for `-D platforms=...`. When the variable is not set, the main library uses the first platform listed in `-D platforms=...` as the native platform.

Extensions like `EGL_MESA_drm_display` define new functions to create displays for non-native platforms. These extensions are usually used by applications that support non-native platforms. Setting this variable is probably required only for some of the demos found in mesa/demo repository.

EGL_LOG_LEVEL This changes the log level of the main library and the drivers. The valid values are: `debug`, `info`, `warning`, and `fatal`.

12.3 Packaging

The ABI between the main library and its drivers are not stable. Nor is there a plan to stabilize it at the moment.

12.4 Developers

The sources of the main library and drivers can be found at `src/egl/`.

The code basically consists of two things:

1. An EGL API dispatcher. This directly routes all the `eglFooBar()` API calls into driver-specific functions.
2. Two EGL drivers (`dri2` and `haiku`), implementing the API functions handling the platforms' specifics.

Two of API functions are optional (`eglQuerySurface()` and `eglSwapInterval()`); the former provides fallback for all the platform-agnostic attributes (ie. everything except `EGL_WIDTH` & `EGL_HEIGHT`), and the latter just silently pretends the API call succeeded (as per EGL spec).

A driver _could_ implement all the other EGL API functions, but several of them are only needed for extensions, like `eglSwapBuffersWithDamageEXT()`. See `src/egl/main/egldriver.h` to see which driver hooks are only required by extensions.

12.4.1 Bootstrapping

When the apps calls `eglInitialize()`, the driver's `Initialize()` function is called. If the first driver initialization attempt fails, a second one is tried using only software components (this can be forced using the `LIBGL_ALWAYS_SOFTWARE` environment variable). Typically, this function takes care of setting up visual configs, creating EGL devices, etc.

12.4.2 Teardown

When `eglTerminate()` is called, the `driver->Terminate()` function is called. The driver should clean up after itself.

12.4.3 Subclassing

The internal libEGL data structures such as `_EGLDisplay`, `_EGLContext`, `_EGLSurface`, etc. should be considered base classes from which drivers will derive subclasses.

12.5 EGL Drivers

egl_dri2 This driver supports several platforms: `android`, `device`, `drm`, ``surfaceless`, `wayland` and `x11`. It functions as a DRI driver loader. For `x11` support, it talks to the X server directly using (XCB-)DRI3 protocol when available, and falls back to DRI2 if necessary (can be forced with `LIBGL_DRI3_DISABLE`).

This driver can share DRI drivers with `libGL`.

haiku This driver supports only the Haiku platform. It is also much less feature-complete than `egl_dri2`, supporting only part of EGL 1.4 and none of the extensions beyond it.

12.5.1 Lifetime of Display Resources

Contexts and surfaces are examples of display resources. They might live longer than the display that creates them.

In EGL, when a display is terminated through `eglTerminate`, all display resources should be destroyed. Similarly, when a thread is released through `eglReleaseThread`, all current display resources should be released. Another way to destroy or release resources is through functions such as `eglDestroySurface` or `eglMakeCurrent`.

When a resource that is current to some thread is destroyed, the resource should not be destroyed immediately. EGL requires the resource to live until it is no longer current. A driver usually calls `eglIs<Resource>Bound` to check if a resource is bound (current) to any thread in the destroy callbacks. If it is still bound, the resource is not destroyed.

The main library will mark destroyed current resources as unlinked. In a driver's `MakeCurrent` callback, `eglIs<Resource>Linked` can then be called to check if a newly released resource is linked to a display. If it is not, the last reference to the resource is removed and the driver should destroy the resource. But it should be careful here because `MakeCurrent` might be called with an uninitialized display.

This is the only mechanism provided by the main library to help manage the resources. The drivers are responsible to the correct behavior as defined by EGL.

12.5.2 EGL_RENDER_BUFFER

In EGL, the color buffer a context should try to render to is decided by the binding surface. It should try to render to the front buffer if the binding surface has EGL_RENDER_BUFFER set to EGL_SINGLE_BUFFER; If the same context is later bound to a surface with EGL_RENDER_BUFFER set to EGL_BACK_BUFFER, the context should try to render to the back buffer. However, the context is allowed to make the final decision as to which color buffer it wants to or is able to render to.

For pbuffer surfaces, the render buffer is always EGL_BACK_BUFFER. And for pixmap surfaces, the render buffer is always EGL_SINGLE_BUFFER. Unlike window surfaces, EGL spec requires their EGL_RENDER_BUFFER values to be honored. As a result, a driver should never set EGL_PIXMAP_BIT or EGL_PBUFFER_BIT bits of a config if the contexts created with the config won't be able to honor the EGL_RENDER_BUFFER of pixmap or pbuffer surfaces.

It should also be noted that pixmap and pbuffer surfaces are assumed to be single-buffered, in that eglSwapBuffers has no effect on them. It is desirable that a driver allocates a private color buffer for each pbuffer surface created. If the window system the driver supports has native pbuffers, or if the native pixmaps have more than one color buffers, the driver should carefully attach the native color buffers to the EGL surfaces, re-route them if required.

There is no defined behavior as to, for example, how glDrawBuffer interacts with EGL_RENDER_BUFFER. Right now, it is desired that the draw buffer in a client API be fixed for pixmap and pbuffer surfaces. Therefore, the driver is responsible to guarantee that the client API renders to the specified render buffer for pixmap and pbuffer surfaces.

12.5.3 EGLDisplay Mutex

The EGLDisplay will be locked before calling any of the dispatch functions (well, except for GetProcAddress which does not take an EGLDisplay). This guarantees that the same dispatch function will not be called with the same display at the same time. If a driver has access to an EGLDisplay without going through the EGL APIs, the driver should as well lock the display before using it.

CHAPTER 13

OpenGL ES

Mesa implements OpenGL ES 1.1 and OpenGL ES 2.0. More information about OpenGL ES can be found at https://www.khronos.org/opengles/.

OpenGL ES depends on a working EGL implementation. Please refer to *Mesa EGL* for more information about EGL.

13.1 Build the Libraries

1. Run `meson configure` with `-D gles1=true -D gles2=true` and enable the Gallium driver for your hardware.
2. Build and install Mesa as usual.

Alternatively, if XCB-DRI2 is installed on the system, one can use `egl_dri2` EGL driver with OpenGL|ES-enabled DRI drivers

1. Run `meson configure` with `-D gles1=true -D gles2=true`.
2. Build and install Mesa as usual.

Both methods will install libGLESv1_CM, libGLESv2, libEGL, and one or more EGL drivers for your hardware.

13.2 Run the Demos

There are some demos in `mesa/demos` repository.

13.3 Developers

13.3.1 Dispatch Table

OpenGL ES has an additional indirection when dispatching functions

```
Mesa:       glFoo() --> _mesa_Foo()
OpenGL ES:  glFoo() --> _es_Foo() --> _mesa_Foo()
```

The indirection serves several purposes

- When a function is in Mesa and the type matches, it checks the arguments and calls the Mesa function.
- When a function is in Mesa but the type mismatches, it checks and converts the arguments before calling the Mesa function.
- When a function is not available in Mesa, or accepts arguments that are not available in OpenGL, it provides its own implementation.

Other than the last case, OpenGL ES uses `APIspec.xml` to generate functions to check and/or converts the arguments.

CHAPTER 14

Environment Variables

Normally, no environment variables need to be set. Most of the environment variables used by Mesa/Gallium are for debugging purposes, but they can sometimes be useful for debugging end-user issues.

14.1 LibGL environment variables

LIBGL_DEBUG If defined debug information will be printed to stderr. If set to `verbose` additional information will be printed.

LIBGL_DRIVERS_PATH colon-separated list of paths to search for DRI drivers

LIBGL_ALWAYS_INDIRECT if set to `true`, forces an indirect rendering context/connection.

LIBGL_ALWAYS_SOFTWARE if set to `true`, always use software rendering

LIBGL_NO_DRAWARRAYS if set to `true`, do not use DrawArrays GLX protocol (for debugging)

LIBGL_SHOW_FPS print framerate to stdout based on the number of `glXSwapBuffers` calls per second.

LIBGL_DRI3_DISABLE disable DRI3 if set to `true`.

14.2 Core Mesa environment variables

MESA_NO_ASM if set, disables all assembly language optimizations

MESA_NO_MMX if set, disables Intel MMX optimizations

MESA_NO_3DNOW if set, disables AMD 3DNow! optimizations

MESA_NO_SSE if set, disables Intel SSE optimizations

MESA_NO_ERROR if set to 1, error checking is disabled as per `KHR_no_error`. This will result in undefined behavior for invalid use of the api, but can reduce CPU use for apps that are known to be error free.

MESA_DEBUG if set, error messages are printed to stderr. For example, if the application generates a `GL_INVALID_ENUM` error, a corresponding error message indicating where the error occurred, and possibly why, will be printed to stderr. For release builds, `MESA_DEBUG` defaults to off (no debug output). `MESA_DEBUG` accepts the following comma-separated list of named flags, which adds extra behavior to just set `MESA_DEBUG=1`:

silent turn off debug messages. Only useful for debug builds.

flush flush after each drawing command

incomplete_tex extra debug messages when a texture is incomplete

incomplete_fbo extra debug messages when a fbo is incomplete

context create a debug context (see `GLX_CONTEXT_DEBUG_BIT_ARB`) and print error and performance messages to stderr (or `MESA_LOG_FILE`).

MESA_LOG_FILE specifies a file name for logging all errors, warnings, etc., rather than stderr

MESA_TEX_PROG if set, implement conventional texture env modes with fragment programs (intended for developers only)

MESA_TNL_PROG if set, implement conventional vertex transformation operations with vertex programs (intended for developers only). Setting this variable automatically sets the `MESA_TEX_PROG` variable as well.

MESA_EXTENSION_OVERRIDE can be used to enable/disable extensions. A value such as `GL_EXT_foo -GL_EXT_bar` will enable the `GL_EXT_foo` extension and disable the `GL_EXT_bar` extension.

MESA_EXTENSION_MAX_YEAR The `GL_EXTENSIONS` string returned by Mesa is sorted by extension year. If this variable is set to year X, only extensions defined on or before year X will be reported. This is to work-around a bug in some games where the extension string is copied into a fixed-size buffer without truncating. If the extension string is too long, the buffer overrun can cause the game to crash. This is a work-around for that.

MESA_GL_VERSION_OVERRIDE changes the value returned by `glGetString(GL_VERSION)` and possibly the GL API type.

- The format should be `MAJOR.MINOR[FC|COMPAT]`
- `FC` is an optional suffix that indicates a forward compatible context. This is only valid for versions >= 3.0.
- `COMPAT` is an optional suffix that indicates a compatibility context or `GL_ARB_compatibility` support. This is only valid for versions >= 3.1.
- GL versions <= 3.0 are set to a compatibility (non-Core) profile
- GL versions = 3.1, depending on the driver, it may or may not have the `ARB_compatibility` extension enabled.
- GL versions >= 3.2 are set to a Core profile
- Examples:

  2.1 select a compatibility (non-Core) profile with GL version 2.1.

  3.0 select a compatibility (non-Core) profile with GL version 3.0.

  3.0FC select a Core+Forward Compatible profile with GL version 3.0.

  3.1 select GL version 3.1 with `GL_ARB_compatibility` enabled per the driver default.

  3.1FC select GL version 3.1 with forward compatibility and `GL_ARB_compatibility` disabled.

  3.1COMPAT select GL version 3.1 with `GL_ARB_compatibility` enabled.

  X.Y override GL version to X.Y without changing the profile.

  X.YFC select a Core+Forward Compatible profile with GL version X.Y.

X.YCOMPAT select a Compatibility profile with GL version X.Y.

- Mesa may not really implement all the features of the given version. (for developers only)

MESA_GLES_VERSION_OVERRIDE changes the value returned by `glGetString(GL_VERSION)` for OpenGL ES.

- The format should be `MAJOR.MINOR`
- Examples: `2.0`, `3.0`, `3.1`
- Mesa may not really implement all the features of the given version. (for developers only)

MESA_GLSL_VERSION_OVERRIDE changes the value returned by `glGetString(GL_SHADING_LANGUAGE_VERSION)`. Valid values are integers, such as `130`. Mesa will not really implement all the features of the given language version if it's higher than what's normally reported. (for developers only)

MESA_GLSL_CACHE_DISABLE if set to `true`, disables the GLSL shader cache. If set to `false`, enables the GLSL shader cache when it is disabled by default.

MESA_GLSL_CACHE_MAX_SIZE if set, determines the maximum size of the on-disk cache of compiled GLSL programs. Should be set to a number optionally followed by `K`, `M`, or `G` to specify a size in kilobytes, megabytes, or gigabytes. By default, gigabytes will be assumed. And if unset, a maximum size of 1GB will be used.

Note: A separate cache might be created for each architecture that Mesa is installed for on your system. For example under the default settings you may end up with a 1GB cache for x86_64 and another 1GB cache for i386.

MESA_GLSL_CACHE_DIR if set, determines the directory to be used for the on-disk cache of compiled GLSL programs. If this variable is not set, then the cache will be stored in `$XDG_CACHE_HOME/mesa_shader_cache` (if that variable is set), or else within `.cache/mesa_shader_cache` within the user's home directory.

MESA_GLSL *shading language compiler options*

MESA_NO_MINMAX_CACHE when set, the minmax index cache is globally disabled.

MESA_SHADER_CAPTURE_PATH see *Capturing Shaders*

MESA_SHADER_DUMP_PATH and MESA_SHADER_READ_PATH see *Experimenting with Shader Replacements*

MESA_VK_VERSION_OVERRIDE changes the Vulkan physical device version as returned in `VkPhysicalDeviceProperties::apiVersion`.

- The format should be `MAJOR.MINOR[.PATCH]`
- This will not let you force a version higher than the driver's instance version as advertised by `vkEnumerateInstanceVersion`
- This can be very useful for debugging but some features may not be implemented correctly. (For developers only)

MESA_LOADER_DRIVER_OVERRIDE chooses a different driver binary such as `etnaviv` or `zink`.

14.3 NIR passes environment variables

The following are only applicable for drivers that uses NIR, as they modify the behavior for the common NIR_PASS and NIR_PASS_V macros, that wrap calls to NIR lowering/optimizations.

NIR_PRINT If defined, the resulting NIR shader will be printed out at each successful NIR lowering/optimization call.

NIR_TEST_CLONE If defined, cloning a NIR shader would be tested at each successful NIR lowering/optimization call.

NIR_TEST_SERIALIZE If defined, serialize and deserialize a NIR shader would be tested at each successful NIR lowering/optimization call.

14.4 Mesa Xlib driver environment variables

The following are only applicable to the Mesa Xlib software driver. See the *Xlib software driver page* for details.

MESA_RGB_VISUAL specifies the X visual and depth for RGB mode

MESA_CI_VISUAL specifies the X visual and depth for CI mode

MESA_BACK_BUFFER specifies how to implement the back color buffer, either `pixmap` or `ximage`

MESA_GAMMA gamma correction coefficients for red, green, blue channels

MESA_XSYNC enable synchronous X behavior (for debugging only)

MESA_GLX_FORCE_CI if set, force GLX to treat 8 BPP visuals as CI visuals

MESA_GLX_FORCE_ALPHA if set, forces RGB windows to have an alpha channel.

MESA_GLX_DEPTH_BITS specifies default number of bits for depth buffer.

MESA_GLX_ALPHA_BITS specifies default number of bits for alpha channel.

14.5 i945/i965 driver environment variables (non-Gallium)

INTEL_NO_HW if set to 1, prevents batches from being submitted to the hardware. This is useful for debugging hangs, etc.

INTEL_DEBUG a comma-separated list of named flags, which do various things:

- **ann** annotate IR in assembly dumps
- **aub** dump batches into an AUB trace for use with simulation tools
- **bat** emit batch information
- **blit** emit messages about blit operations
- **blorp** emit messages about the blorp operations (blits & clears)
- **buf** emit messages about buffer objects
- **clip** emit messages about the clip unit (for old gens, includes the CLIP program)
- **color** use color in output
- **cs** dump shader assembly for compute shaders
- **do32** generate compute shader SIMD32 programs even if workgroup size doesn't exceed the SIMD16 limit
- **dri** emit messages about the DRI interface
- **fbo** emit messages about framebuffers
- **fs** dump shader assembly for fragment shaders

gs dump shader assembly for geometry shaders

hex print instruction hex dump with the disassembly

l3 emit messages about the new L3 state during transitions

miptree emit messages about miptrees

no8 don't generate SIMD8 fragment shader

no16 suppress generation of 16-wide fragment shaders. useful for debugging broken shaders

nocompact disable instruction compaction

nodualobj suppress generation of dual-object geometry shader code

nofc disable fast clears

norbc disable single sampled render buffer compression

optimizer dump shader assembly to files at each optimization pass and iteration that make progress

perf emit messages about performance issues

perfmon emit messages about `AMD_performance_monitor`

pix emit messages about pixel operations

prim emit messages about drawing primitives

reemit mark all state dirty on each draw call

sf emit messages about the strips & fans unit (for old gens, includes the SF program)

shader_time record how much GPU time is spent in each shader

spill_fs force spilling of all registers in the scalar backend (useful to debug spilling code)

spill_vec4 force spilling of all registers in the vec4 backend (useful to debug spilling code)

state emit messages about state flag tracking

submit emit batchbuffer usage statistics

sync after sending each batch, emit a message and wait for that batch to finish rendering

tcs dump shader assembly for tessellation control shaders

tes dump shader assembly for tessellation evaluation shaders

tex emit messages about textures.

urb emit messages about URB setup

vert emit messages about vertex assembly

vs dump shader assembly for vertex shaders

INTEL_SCALAR_VS (or TCS, TES, GS) force scalar/vec4 mode for a shader stage (Gen8-9 only)

INTEL_PRECISE_TRIG if set to 1, true or yes, then the driver prefers accuracy over performance in trig functions.

INTEL_SHADER_ASM_READ_PATH if set, determines the directory to be used for overriding shader assembly. The binaries with custom assembly should be placed in this folder and have a name formatted as `sha1_of_assembly.bin`. The sha1 of a shader assembly is printed when assembly is dumped via corresponding `INTEL_DEBUG` flag (e.g. `vs` for vertex shader). A binary could be generated from a dumped assembly by `i965_asm`. For `INTEL_SHADER_ASM_READ_PATH` to work it is necessary to enable dumping of corresponding shader stages via `INTEL_DEBUG`. It is advised to use `nocompact` flag of `INTEL_DEBUG`

when dumping and overriding shader assemblies. The success of assembly override would be signified by "Successfully overrode shader with sha1 <sha1>" in stderr replacing the original assembly.

14.6 Radeon driver environment variables (radeon, r200, and r300g)

RADEON_NO_TCL if set, disable hardware-accelerated Transform/Clip/Lighting.

14.7 EGL environment variables

Mesa EGL supports different sets of environment variables. See the *Mesa EGL* page for the details.

14.8 Gallium environment variables

GALLIUM_HUD draws various information on the screen, like framerate, CPU load, driver statistics, performance counters, etc. Set `GALLIUM_HUD=help` and run e.g. `glxgears` for more info.

GALLIUM_HUD_PERIOD sets the hud update rate in seconds (float). Use zero to update every frame. The default period is 1/2 second.

GALLIUM_HUD_VISIBLE control default visibility, defaults to true.

GALLIUM_HUD_TOGGLE_SIGNAL toggle visibility via user specified signal. Especially useful to toggle hud at specific points of application and disable for unencumbered viewing the rest of the time. For example, set `GALLIUM_HUD_VISIBLE` to `false` and `GALLIUM_HUD_TOGGLE_SIGNAL` to `10` (`SIGUSR1`). Use `kill -10 <pid>` to toggle the hud as desired.

GALLIUM_HUD_SCALE Scale hud by an integer factor, for high DPI displays. Default is 1.

GALLIUM_HUD_DUMP_DIR specifies a directory for writing the displayed hud values into files.

GALLIUM_DRIVER useful in combination with `LIBGL_ALWAYS_SOFTWARE=true` for choosing one of the software renderers `softpipe`, `llvmpipe` or `swr`.

GALLIUM_LOG_FILE specifies a file for logging all errors, warnings, etc. rather than stderr.

GALLIUM_PIPE_SEARCH_DIR specifies an alternate search directory for pipe-loader which overrides the compile-time path based on the install location.

GALLIUM_PRINT_OPTIONS if non-zero, print all the Gallium environment variables which are used, and their current values.

GALLIUM_DUMP_CPU if non-zero, print information about the CPU on start-up

TGSI_PRINT_SANITY if set, do extra sanity checking on TGSI shaders and print any errors to stderr.

DRAW_FSE ???

DRAW_NO_FSE ???

DRAW_USE_LLVM if set to zero, the draw module will not use LLVM to execute shaders, vertex fetch, etc.

ST_DEBUG controls debug output from the Mesa/Gallium state tracker. Setting to `tgsi`, for example, will print all the TGSI shaders. See `src/mesa/state_tracker/st_debug.c` for other options.

14.8.1 Clover environment variables

CLOVER_EXTRA_BUILD_OPTIONS allows specifying additional compiler and linker options. Specified options are appended after the options set by the OpenCL program in `clBuildProgram`.

CLOVER_EXTRA_COMPILE_OPTIONS allows specifying additional compiler options. Specified options are appended after the options set by the OpenCL program in `clCompileProgram`.

CLOVER_EXTRA_LINK_OPTIONS allows specifying additional linker options. Specified options are appended after the options set by the OpenCL program in `clLinkProgram`.

14.8.2 Softpipe driver environment variables

SOFTPIPE_DEBUG a comma-separated list of named flags, which do various things:

- **vs** Dump vertex shader assembly to stderr
- **fs** Dump fragment shader assembly to stderr
- **gs** Dump geometry shader assembly to stderr
- **cs** Dump compute shader assembly to stderr
- **no_rast** rasterization is no-op'd. For profiling purposes.
- **use_llvm** the softpipe driver will try to use LLVM JIT for vertex shading processing.
- **use_tgsi** if set, the softpipe driver will ask to directly consume TGSI, instead of NIR.

14.8.3 LLVMpipe driver environment variables

LP_NO_RAST if set LLVMpipe will no-op rasterization

LP_DEBUG a comma-separated list of debug options is accepted. See the source code for details.

LP_PERF a comma-separated list of options to selectively no-op various parts of the driver. See the source code for details.

LP_NUM_THREADS an integer indicating how many threads to use for rendering. Zero turns off threading completely. The default value is the number of CPU cores present.

14.8.4 VMware SVGA driver environment variables

SVGA_FORCE_SWTNL force use of software vertex transformation

SVGA_NO_SWTNL don't allow software vertex transformation fallbacks (will often result in incorrect rendering).

SVGA_DEBUG for dumping shaders, constant buffers, etc. See the code for details.

SVGA_EXTRA_LOGGING if set, enables extra logging to the `vmware.log` file, such as the OpenGL program's name and command line arguments.

SVGA_NO_LOGGING if set, disables logging to the `vmware.log` file. This is useful when using Valgrind because it otherwise crashes when initializing the host log feature.

See the driver code for other, lesser-used variables.

14.8.5 WGL environment variables

WGL_SWAP_INTERVAL to set a swap interval, equivalent to calling `wglSwapIntervalEXT()` in an application. If this environment variable is set, application calls to `wglSwapIntervalEXT()` will have no effect.

14.8.6 VA-API environment variables

VAAPI_MPEG4_ENABLED enable MPEG4 for VA-API, disabled by default.

14.8.7 VC4 driver environment variables

VC4_DEBUG a comma-separated list of named flags, which do various things:

- **cl** dump command list during creation
- **qpu** dump generated QPU instructions
- **qir** dump QPU IR during program compile
- **nir** dump NIR during program compile
- **tgsi** dump TGSI during program compile
- **shaderdb** dump program compile information for shader-db analysis
- **perf** print during performance-related events
- **norast** skip actual hardware execution of commands
- **always_flush** flush after each draw call
- **always_sync** wait for finish after each flush
- **dump** write a GPU command stream trace file (VC4 simulator only)

14.8.8 RADV driver environment variables

RADV_DEBUG a comma-separated list of named flags, which do various things:

- **llvm** enable LLVM compiler backend
- **allbos** force all allocated buffers to be referenced in submissions
- **allentrypoints** enable all device/instance entrypoints
- **checkir** validate the LLVM IR before LLVM compiles the shader
- **errors** display more info about errors
- **forcecompress** Enables DCC,FMASK,CMASK,HTILE in situations where the driver supports it but normally does not deem it beneficial.
- **info** show GPU-related information
- **metashaders** dump internal meta shaders
- **nobinning** disable primitive binning
- **nocache** disable shaders cache
- **nocompute** disable compute queue
- **nodcc** disable Delta Color Compression (DCC) on images

nodynamicbounds do not check OOB access for dynamic descriptors

nofastclears disable fast color/depthstencil clears

nohiz disable HIZ for depthstencil images

noibs disable directly recording command buffers in GPU-visible memory

nomemorycache disable memory shaders cache

nongg disable NGG for GFX10+

nooutoforder disable out-of-order rasterization

nothreadllvm disable LLVM threaded compilation

preoptir dump LLVM IR before any optimizations

shaders dump shaders

shaderstats dump shader statistics

spirv dump SPIR-V

startup display info at startup

syncshaders synchronize shaders after all draws/dispatches

vmfaults check for VM memory faults via dmesg

zerovram initialize all memory allocated in VRAM as zero

RADV_FORCE_FAMILY create a null device to compile shaders without a AMD GPU (eg. gfx900)

RADV_PERFTEST a comma-separated list of named flags, which do various things:

bolist enable the global BO list

cswave32 enable wave32 for compute shaders (GFX10+)

dccmsaa enable DCC for MSAA images

dfsm enable dfsm

gewave32 enable wave32 for vertex/tess/geometry shaders (GFX10+)

localbos enable local BOs

pswave32 enable wave32 for pixel shaders (GFX10+)

tccompatcmask enable TC-compat cmask for MSAA images

RADV_TEX_ANISO force anisotropy filter (up to 16)

RADV_TRACE_FILE generate cmdbuffer tracefiles when a GPU hang is detected

ACO_DEBUG a comma-separated list of named flags, which do various things:

validateir validate the ACO IR at various points of compilation (enabled by default for debug/debugoptimized builds)

validatera validate register assignment of ACO IR and catches many RA bugs

perfwarn abort on some suboptimal code generation

force-waitcnt force emitting waitcnt states if there is something to wait for

novn disable value numbering

noopt disable various optimizations

noscheduling disable instructions scheduling

14.8.9 radeonsi driver environment variables

AMD_DEBUG a comma-separated list of named flags, which do various things:

nodma Disable SDMA

nodmaclear Disable SDMA clears

nodmacopyimage Disable SDMA image copies

zerovram Clear VRAM allocations.

nodcc Disable DCC.

nodccclear Disable DCC fast clear.

nodccfb Disable separate DCC on the main framebuffer

nodccmsaa Disable DCC for MSAA

nodpbb Disable DPBB.

nodfsm Disable DFSM.

notiling Disable tiling

nofmask Disable MSAA compression

nohyperz Disable Hyper-Z

norbplus Disable RB+.

no2d Disable 2D tiling

info Print driver information

tex Print texture info

compute Print compute info

vm Print virtual addresses when creating resources

vs Print vertex shaders

ps Print pixel shaders

gs Print geometry shaders

tcs Print tessellation control shaders

tes Print tessellation evaluation shaders

cs Print compute shaders

noir Don't print the LLVM IR

nonir Don't print NIR when printing shaders

noasm Don't print disassembled shaders

preoptir Print the LLVM IR before initial optimizations

gisel Enable LLVM global instruction selector.

w32ge Use Wave32 for vertex, tessellation, and geometry shaders.

w32ps Use Wave32 for pixel shaders.

w32cs Use Wave32 for computes shaders.

w64ge Use Wave64 for vertex, tessellation, and geometry shaders.

w64ps Use Wave64 for pixel shaders.

w64cs Use Wave64 for computes shaders.

checkir Enable additional sanity checks on shader IR

mono Use old-style monolithic shaders compiled on demand

nooptvariant Disable compiling optimized shader variants.

forcedma Use SDMA for all operations when possible.

nowc Disable GTT write combining

check_vm Check VM faults and dump debug info.

reserve_vmid Force VMID reservation per context.

nogfx Disable graphics. Only multimedia compute paths can be used.

nongg Disable NGG and use the legacy pipeline.

nggc Always use NGG culling even when it can hurt.

nonggc Disable NGG culling.

alwayspd Always enable the primitive discard compute shader.

pd Enable the primitive discard compute shader for large draw calls.

nopd Disable the primitive discard compute shader.

switch_on_eop Program WD/IA to switch on end-of-packet.

nooutoforder Disable out-of-order rasterization

dpbb Enable DPBB.

dfsm Enable DFSM.

Other Gallium drivers have their own environment variables. These may change frequently so the source code should be consulted for details.

CHAPTER 15

Off-screen Rendering

Mesa's off-screen interface is used for rendering into user-allocated memory without any sort of window system or operating system dependencies. That is, the GL_FRONT colorbuffer is actually a buffer in main memory, rather than a window on your display.

The OSMesa API provides three basic functions for making off-screen renderings: OSMesaCreateContext(), OSMesaMakeCurrent(), and OSMesaDestroyContext(). See the Mesa/include/GL/osmesa.h header for more information about the API functions.

The OSMesa interface may be used with any of three software renderers:

1. llvmpipe - this is the high-performance Gallium LLVM driver
2. softpipe - this it the reference Gallium software driver
3. swrast - this is the legacy Mesa software rasterizer

There are several examples of OSMesa in the mesa/demos repository.

15.1 Building OSMesa

Configure and build Mesa with something like:

```
meson builddir -Dosmesa=gallium -Dgallium-drivers=swrast -Ddri-drivers=[] -Dvulkan-
↪drivers=[] -Dprefix=$PWD/builddir/install
ninja -C builddir install
```

Make sure you have LLVM installed first if you want to use the llvmpipe driver.

When the build is complete you should find:

```
$PWD/builddir/install/lib/libOSMesa.so  (swrast-based OSMesa)
$PWD/builddir/install/lib/gallium/libOSMsea.so  (Gallium-based OSMesa)
```

Set your LD_LIBRARY_PATH to point to $PWD/builddir/install to use the libraries

When you link your application, link with -lOSMesa

CHAPTER 16

Debugging Tips

Normally Mesa (and OpenGL) records but does not notify the user of errors. It is up to the application to call `glGetError` to check for errors. Mesa supports an environment variable, `MESA_DEBUG`, to help with debugging. If `MESA_DEBUG` is defined, a message will be printed to stdout whenever an error occurs.

More extensive error checking is done in DEBUG builds (`--buildtype debug` for Meson, `build=debug` for SCons).

In your debugger you can set a breakpoint in `_mesa_error()` to trap Mesa errors.

There is a display list printing/debugging facility. See the end of `src/dlist.c` for details.

CHAPTER 17

Performance Tips

Performance tips for software rendering:

1. Turn off smooth shading when you don't need it (glShadeModel)
2. Turn off depth buffering when you don't need it.
3. Turn off dithering when not needed.
4. Use double buffering as it's often faster than single buffering
5. Compile in the X Shared Memory extension option if it's supported on your system by adding -DSHM to CFLAGS and -lXext to XLIBS for your system in the Make-config file.
6. Recompile Mesa with more optimization if possible.
7. Try to maximize the amount of drawing done between glBegin/glEnd pairs.
8. Use the MESA_BACK_BUFFER variable to find best performance in double buffered mode. (X users only)
9. Optimized polygon rasterizers are employed when: rendering into back buffer which is an XImage RGB mode, not grayscale, not monochrome depth buffering is GL_LESS, or disabled flat or smooth shading dithered or non-dithered no other rasterization operations enabled (blending, stencil, etc)
10. Optimized line drawing is employed when: rendering into back buffer which is an XImage RGB mode, not grayscale, not monochrome depth buffering is GL_LESS or disabled flat shading dithered or non-dithered no other rasterization operations enabled (blending, stencil, etc)
11. Textured polygons are fastest when: using a 3-component (RGB), 2-D texture minification and magnification filters are GL_NEAREST texture coordinate wrap modes for S and T are GL_REPEAT GL_DECAL environment mode glHint(GL_PERSPECTIVE_CORRECTION_HINT, GL_FASTEST) depth buffering is GL_LESS or disabled
12. Lighting is fastest when: Two-sided lighting is disabled GL_LIGHT_MODEL_LOCAL_VIEWER is false GL_COLOR_MATERIAL is disabled No spot lights are used (all GL_SPOT_CUTOFFs are 180.0) No local lights are used (all position W's are 0.0) All material and light coefficients are >= zero
13. XFree86 users: if you want to use 24-bit color try starting your X server in 32-bit per pixel mode for better performance. That is, start your X server with startx – -bpp 32 instead of startx – -bpp 24

14. Try disabling dithering with the MESA_NO_DITHER environment variable. If this env var is defined Mesa will disable dithering and the command glEnable(GL_DITHER) will be ignored.

CHAPTER 18

Mesa Extensions

A number of extensions have been developed especially for Mesa. The specifications follow.

- MESA_agp_offset.spec
- MESA_copy_sub_buffer.spec
- MESA_drm_image.spec
- MESA_multithread_makecurrent.spec
- MESA_packed_depth_stencil.spec (obsolete)
- MESA_pack_invert.spec
- MESA_pixmap_colormap.spec
- MESA_program_debug.spec (obsolete)
- MESA_release_buffers.spec
- MESA_resize_buffers.spec (obsolete)
- MESA_set_3dfx_mode.spec
- MESA_shader_debug.spec
- MESA_sprite_point.spec (obsolete)
- MESA_swap_control.spec
- MESA_swap_frame_usage.spec
- MESA_texture_array.spec
- MESA_texture_signed_rgba.spec
- MESA_trace.spec (obsolete)
- MESA_window_pos.spec
- MESA_ycbcr_texture.spec
- WL_bind_wayland_display.spec

CHAPTER 19

VMware SVGA3D Guest Driver

This page describes how to build, install and use the VMware guest GL driver (aka the SVGA or SVGA3D driver) for Linux using the latest source code. This driver gives a Linux virtual machine access to the host's GPU for hardware-accelerated 3D. VMware Workstation running on Linux or Windows and VMware Fusion running on MacOS are all supported.

With the August 2015 Workstation 12 / Fusion 8 releases, OpenGL 3.3 is supported in the guest. This requires:

- The VM is configured for virtual hardware version 12.
- The host OS, GPU and graphics driver supports DX11 (Windows) or OpenGL 4.0 (Linux, Mac)
- On Linux, the vmwgfx kernel module must be version 2.9.0 or later.
- A recent version of Mesa with the updated svga Gallium driver.

Otherwise, OpenGL 2.1 is supported.

With the Fall 2018 Workstation 15 / Fusion 11 releases, additional features are supported in the driver:

- Multisample antialiasing (2x, 4x)
- GL_ARB/AMD_draw_buffers_blend
- GL_ARB_sample_shading
- GL_ARB_texture_cube_map_array
- GL_ARB_texture_gather
- GL_ARB_texture_query_lod
- GL_EXT/OES_draw_buffers_indexed

This requires version 2.15.0 or later of the vmwgfx kernel module and the VM must be configured for hardware version 16 or later.

OpenGL 3.3 support can be disabled by setting the environment variable SVGA_VGPU10=0. You will then have OpenGL 2.1 support. This may be useful to work around application bugs (such as incorrect use of the OpenGL 3.x core profile).

Most modern Linux distros include the SVGA3D driver so end users shouldn't be concerned with this information. But if your distro lacks the driver or you want to update to the latest code these instructions explain what to do.

For more information about the X components see these wiki pages at x.org:

- Driver Overview
- xf86-video-vmware Details

19.1 Components

The components involved in this include:

- Linux kernel module: vmwgfx
- X server 2D driver: xf86-video-vmware
- User-space libdrm library
- Mesa/Gallium OpenGL driver: "svga"

All of these components reside in the guest Linux virtual machine. On the host, all you're doing is running VMware Workstation or Fusion.

19.2 Prerequisites

- Kernel version at least 2.6.25
- Xserver version at least 1.7
- Ubuntu: For Ubuntu you need to install a number of build dependencies.

```
sudo apt-get install git-core
sudo apt-get install ninja-build meson libpthread-stubs0-dev
sudo apt-get install xserver-xorg-dev x11proto-xinerama-dev libx11-xcb-dev
sudo apt-get install libxcb-glx0-dev libxrender-dev
sudo apt-get build-dep libgl1-mesa-dri libxcb-glx0-dev
```

- Fedora: For Fedora you also need to install a number of build dependencies.

```
sudo yum install mesa-libGL-devel xorg-x11-server-devel xorg-x11-util-macros
sudo yum install libXrender-devel.i686
sudo yum install ninja-build meson gcc expat-devel kernel-devel git-core
sudo yum install makedepend flex bison
```

Depending on your Linux distro, other packages may be needed. Meson should tell you what's missing.

19.3 Getting the Latest Source Code

Begin by saving your current directory location:

```
export TOP=$PWD
```

- Mesa/Gallium master branch. This code is used to build libGL, and the direct rendering svga driver for libGL, vmwgfx_dri.so, and the X acceleration library libxatracker.so.x.x.x.

```
git clone https://gitlab.freedesktop.org/mesa/mesa.git
```

- VMware Linux guest kernel module. Note that this repo contains the complete DRM and TTM code. The vmware-specific driver is really only the files prefixed with vmwgfx.

```
git clone git://anongit.freedesktop.org/git/mesa/vmwgfx
```

- libdrm, a user-space library that interfaces with DRM. Most distros ship with this but it's safest to install a newer version. To get the latest code from Git:

```
git clone https://gitlab.freedesktop.org/mesa/drm.git
```

- xf86-video-vmware. The chainloading driver, vmware_drv.so, the legacy driver vmwlegacy_drv.so, and the vmwgfx driver vmwgfx_drv.so.

```
git clone git://anongit.freedesktop.org/git/xorg/driver/xf86-video-vmware
```

19.4 Building the Code

- Determine where the GL-related libraries reside on your system and set the LIBDIR environment variable accordingly.

  For 32-bit Ubuntu systems:

```
export LIBDIR=/usr/lib/i386-linux-gnu
```

  For 64-bit Ubuntu systems:

```
export LIBDIR=/usr/lib/x86_64-linux-gnu
```

  For 32-bit Fedora systems:

```
export LIBDIR=/usr/lib
```

  For 64-bit Fedora systems:

```
export LIBDIR=/usr/lib64
```

- Build libdrm:

```
cd $TOP/drm
meson builddir --prefix=/usr --libdir=${LIBDIR}
ninja -C builddir
sudo ninja -C builddir install
```

- Build Mesa and the vmwgfx_dri.so driver, the vmwgfx_drv.so xorg driver, the X acceleration library libxatracker. The vmwgfx_dri.so is used by the OpenGL libraries during direct rendering, and by the Xorg server during accelerated indirect GL rendering. The libxatracker library is used exclusively by the X server to do render, copy and video acceleration:

  The following configure options doesn't build the EGL system.

```
cd $TOP/mesa
meson builddir --prefix=/usr --libdir=${LIBDIR} -Dgallium-drivers=svga -Ddri-
↪drivers=swrast -Dgallium-xa=true -Ddri3=false
```

(continues on next page)

(continued from previous page)

```
ninja -C builddir
sudo ninja -C builddir install
```

Note that you may have to install other packages that Mesa depends upon if they're not installed in your system. You should be told what's missing.

- xf86-video-vmware: Now, once libxatracker is installed, we proceed with building and replacing the current Xorg driver. First check if your system is 32- or 64-bit.

```
cd $TOP/xf86-video-vmware
./autogen.sh --prefix=/usr --libdir=${LIBDIR}
make
sudo make install
```

- vmwgfx kernel module. First make sure that any old version of this kernel module is removed from the system by issuing

```
sudo rm /lib/modules/`uname -r`/kernel/drivers/gpu/drm/vmwgfx.ko*
```

Build and install:

```
cd $TOP/vmwgfx
make
sudo make install
sudo depmod -a
```

If you're using a Ubuntu OS:

```
sudo update-initramfs -u
```

If you're using a Fedora OS:

```
sudo dracut --force
```

Add 'vmwgfx' to the /etc/modules file:

```
echo vmwgfx | sudo tee -a /etc/modules
```

Note: some distros put DRM kernel drivers in different directories. For example, sometimes vmwgfx.ko might be found in `/lib/modules/{version}/extra/vmwgfx.ko` or in `/lib/modules/{version}/kernel/drivers/gpu/drm/vmwgfx/vmwgfx.ko`.

After installing vmwgfx.ko you might want to run the following command to check that the new kernel module is in the expected place:

```
find /lib/modules -name vmwgfx.ko -exec ls -l '{}' \;
```

If you see the kernel module listed in more than one place, you may need to move things around.

Finally, if you update your kernel you'll probably have to rebuild and reinstall the vmwgfx.ko module again.

Now try to load the kernel module by issuing

```
sudo modprobe vmwgfx
```

Then type

```
dmesg
```

to watch the debug output. It should contain a number of lines prefixed with "[vmwgfx]".

Then restart the Xserver (or reboot). The lines starting with "vmwlegacy" or "VMWARE" in the file /var/log/Xorg.0.log should now have been replaced with lines starting with "vmwgfx", indicating that the new Xorg driver is in use.

19.5 Running OpenGL Programs

In a shell, run 'glxinfo' and look for the following to verify that the driver is working:

```
OpenGL vendor string: VMware, Inc.
OpenGL renderer string: Gallium 0.4 on SVGA3D; build: RELEASE;
OpenGL version string: 2.1 Mesa 8.0
```

If you don't see this, try setting this environment variable:

```
export LIBGL_DEBUG=verbose
```

then rerun glxinfo and examine the output for error messages.

If OpenGL 3.3 is not working (you only get OpenGL 2.1):

- Make sure the VM uses hardware version 12.
- Make sure the vmwgfx kernel module is version 2.9.0 or later.
- Check the vmware.log file for errors.
- Run 'dmesg | grep vmwgfx' and look for "DX: yes".

CHAPTER 20

Application Issues

This page documents known issues with some OpenGL applications.

20.1 Topogun

Topogun for Linux (version 2, at least) creates a GLX visual without requesting a depth buffer. This causes bad rendering if the OpenGL driver happens to choose a visual without a depth buffer.

Mesa 9.1.2 and later (will) support a DRI configuration option to work around this issue. Using the driconf tool, set the "Create all visuals with a depth buffer" option before running Topogun. Then, all GLX visuals will be created with a depth buffer.

20.2 Old OpenGL games

Some old OpenGL games (approx. ten years or older) may crash during start-up because of an extension string buffer-overflow problem.

The problem is a modern OpenGL driver will return a very long string for the `glGetString(GL_EXTENSIONS)` query and if the application naively copies the string into a fixed-size buffer it can overflow the buffer and crash the application.

The work-around is to set the `MESA_EXTENSION_MAX_YEAR` environment variable to the approximate release year of the game. This will cause the `glGetString(GL_EXTENSIONS)` query to only report extensions older than the given year.

For example, if the game was released in 2001, do

```
export MESA_EXTENSION_MAX_YEAR=2001
```

before running the game.

20.3 Viewperf

See the *Viewperf issues* page for a detailed list of Viewperf issues.

CHAPTER 21

Viewperf Issues

This page lists known issues with SPEC Viewperf 11 and SPEC Viewperf 12 when running on Mesa-based drivers.

The Viewperf data sets are basically GL API traces that are recorded from CAD applications, then replayed in the Viewperf framework.

The primary problem with these traces is they blindly use features and OpenGL extensions that were supported by the OpenGL driver when the trace was recorded, but there's no checks to see if those features are supported by the driver when playing back the traces with Viewperf.

These issues have been reported to the SPEC organization in the hope that they'll be fixed in the future.

21.1 Viewperf 11

Some of the Viewperf 11 tests use a lot of memory. At least 2GB of RAM is recommended.

21.1.1 Catia-03 test 2

This test creates over 38000 vertex buffer objects. On some systems this can exceed the maximum number of buffer allocations. Mesa generates GL_OUT_OF_MEMORY errors in this situation, but Viewperf does no error checking and continues. When this happens, some drawing commands become no-ops. This can also eventually lead to a segfault either in Viewperf or the Mesa driver.

21.1.2 Catia-03 tests 3, 4, 8

These tests use features of the GL_NV_fragment_program2 and GL_NV_vertex_program3 extensions without checking if the driver supports them.

When Mesa tries to compile the vertex/fragment programs it generates errors (which Viewperf ignores). Subsequent drawing calls become no-ops and the rendering is incorrect.

21.1.3 sw-02 tests 1, 2, 4, 6

These tests depend on the GL_NV_primitive_restart extension.

If the Mesa driver doesn't support this extension the rendering will be incorrect and the test will fail.

Also, the color of the line drawings in test 2 seem to appear in a random color. This is probably due to some uninitialized state somewhere.

21.1.4 sw-02 test 6

The lines drawn in this test appear in a random color. That's because texture mapping is enabled when the lines are drawn, but no texture image is defined (glTexImage2D() is called with pixels=NULL). Since GL says the contents of the texture image are undefined in that situation, we get a random color.

21.1.5 Lightwave-01 test 3

This test uses a number of mipmapped textures, but the textures are incomplete because the last/smallest mipmap level (1 x 1 pixel) is never specified.

A trace captured with API trace shows this sequences of calls like this:

```
2504 glBindTexture(target = GL_TEXTURE_2D, texture = 55)
2505 glTexImage2D(target = GL_TEXTURE_2D, level = 0, internalformat = GL_RGBA, width␣
↪= 512, height = 512, border = 0, format = GL_RGB, type = GL_UNSIGNED_SHORT, pixels␣
↪= blob(1572864))
2506 glTexImage2D(target = GL_TEXTURE_2D, level = 1, internalformat = GL_RGBA, width␣
↪= 256, height = 256, border = 0, format = GL_RGB, type = GL_UNSIGNED_SHORT, pixels␣
↪= blob(393216))
2507 glTexImage2D(target = GL_TEXTURE_2D, level = 2, internalformat = GL_RGBA, width␣
↪= 128, height = 128, border = 0, format = GL_RGB, type = GL_UNSIGNED_SHORT, pixels␣
↪= blob(98304))
[...]
2512 glTexImage2D(target = GL_TEXTURE_2D, level = 7, internalformat = GL_RGBA, width␣
↪= 4, height = 4, border = 0, format = GL_RGB, type = GL_UNSIGNED_SHORT, pixels =␣
↪blob(96))
2513 glTexImage2D(target = GL_TEXTURE_2D, level = 8, internalformat = GL_RGBA, width␣
↪= 2, height = 2, border = 0, format = GL_RGB, type = GL_UNSIGNED_SHORT, pixels =␣
↪blob(24))
2514 glTexParameteri(target = GL_TEXTURE_2D, pname = GL_TEXTURE_MIN_FILTER, param =␣
↪GL_LINEAR_MIPMAP_LINEAR)
2515 glTexParameteri(target = GL_TEXTURE_2D, pname = GL_TEXTURE_WRAP_S, param = GL_
↪REPEAT)
2516 glTexParameteri(target = GL_TEXTURE_2D, pname = GL_TEXTURE_WRAP_T, param = GL_
↪REPEAT)
2517 glTexParameteri(target = GL_TEXTURE_2D, pname = GL_TEXTURE_MAG_FILTER, param =␣
↪GL_NEAREST)
```

Note that one would expect call 2514 to be glTexImage(level=9, width=1, height=1) but it's not there.

The minification filter is GL_LINEAR_MIPMAP_LINEAR and the texture's GL_TEXTURE_MAX_LEVEL is 1000 (the default) so a full mipmap is expected.

Later, these incomplete textures are bound before drawing calls. According to the GL specification, if a fragment program or fragment shader is being used, the sampler should return (0,0,0,1) ("black") when sampling from an incomplete texture. This is what Mesa does and the resulting rendering is darker than it should be.

It appears that NVIDIA's driver (and possibly AMD's driver) detects this case and returns (1,1,1,1) (white) which causes the rendering to appear brighter and match the reference image (however, AMD's rendering is *much* brighter than NVIDIA's).

If the fallback texture created in _mesa_get_fallback_texture() is initialized to be full white instead of full black the rendering appears correct. However, we have no plans to implement this work-around in Mesa.

21.1.6 Maya-03 test 2

This test makes some unusual calls to glRotate. For example:

```
glRotate(50, 50, 50, 1);
glRotate(100, 100, 100, 1);
glRotate(52, 52, 52, 1);
```

These unusual values lead to invalid modelview matrices. For example, the last glRotate command above produces this matrix with Mesa:

```
1.08536e+24 2.55321e-23 -0.000160389 0
5.96937e-25 1.08536e+24 103408 0
103408 -0.000160389 1.74755e+09 0
0 0 0 nan
```

and with NVIDIA's OpenGL:

```
1.4013e-45 0 -nan 0
0 1.4013e-45 1.4013e-45 0
1.4013e-45 -nan 1.4013e-45 0
0 0 0 1.4013e-45
```

This causes the object in question to be drawn in a strange orientation and with a semi-random color (between white and black) since GL_FOG is enabled.

21.1.7 Proe-05 test 1

This uses depth testing but there's two problems:

1. The glXChooseFBConfig() call doesn't request a depth buffer
2. The test never calls glClear(GL_DEPTH_BUFFER_BIT) to initialize the depth buffer

If the chosen visual does not have a depth buffer, you'll see the wireframe car model but it won't be rendered correctly.

If (by luck) the chosen visual has a depth buffer, its initial contents will be undefined so you may or may not see parts of the model.

Interestingly, with NVIDIA's driver most visuals happen to have a depth buffer and apparently the contents are initialized to 1.0 by default so this test just happens to work with their drivers.

Finally, even if a depth buffer was requested and the glClear(GL_COLOR_BUFFER_BIT) calls were changed to glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT) the problem still wouldn't be fixed because GL_DEPTH_WRITEMASK=GL_FALSE when glClear is called so clearing the depth buffer would be a no-op anyway.

21.1.8 Proe-05 test 6

This test draws an engine model with a two-pass algorithm. The first pass is drawn with polygon stipple enabled. The second pass is drawn without polygon stipple but with blending and GL_DEPTH_FUNC=GL_LEQUAL. If either of the two passes happen to use a software fallback of some sort, the Z values of fragments may be different between the two passes. This leads to incorrect rendering.

For example, the VMware SVGA Gallium driver uses a special semi-fallback path for drawing with polygon stipple. Since the two passes are rendered with different vertex transformation implementations, the rendering doesn't appear as expected. Setting the SVGA_FORCE_SWTNL environment variable to 1 will force the driver to use the software vertex path all the time and clears up this issue.

According to the OpenGL invariance rules, there's no guarantee that the pixels produced by these two rendering states will match. To achieve invariance, both passes should enable polygon stipple and blending with appropriate patterns/modes to ensure the same fragments are produced in both passes.

21.2 Viewperf 12

Note that Viewperf 12 only runs on 64-bit Windows 7 or later.

21.2.1 catia-04

One of the catia tests calls wglGetProcAddress() to get some GL_EXT_direct_state_access functions (such as glBindMultiTextureEXT) and some GL_NV_half_float functions (such as glMultiTexCoord3hNV). If the extension/function is not supported, wglGetProcAddress() can return NULL. Unfortunately, Viewperf doesn't check for null pointers and crashes when it later tries to use the pointer.

Another catia test uses OpenGL 3.1's primitive restart feature. But when Viewperf creates an OpenGL context, it doesn't request version 3.1 If the driver returns version 3.0 or earlier all the calls related to primitive restart generate an OpenGL error. Some of the rendering is then incorrect.

21.2.2 energy-01

This test creates a 3D luminance texture of size 1K x 1K x 1K. If the OpenGL driver/device doesn't support a texture of this size the glTexImage3D() call will fail with GL_INVALID_VALUE or GL_OUT_OF_MEMORY and all that's rendered is plain white polygons. Ideally, the test would use a proxy texture to determine the max 3D texture size. But it does not do that.

21.2.3 maya-04

This test generates many GL_INVALID_OPERATION errors in its calls to glUniform(). Causes include:

- Trying to set float uniforms with glUniformi()
- Trying to set float uniforms with glUniform3f()
- Trying to set matrix uniforms with glUniform() instead of glUniformMatrix().

Apparently, the indexes returned by glGetUniformLocation() were hard-coded into the application trace when it was created. Since different implementations of glGetUniformLocation() may return different values for any given uniform name, subsequent calls to glUniform() will be invalid since they refer to the wrong uniform variables. This causes many OpenGL errors and leads to incorrect rendering.

21.2.4 medical-01

This test uses a single GLSL fragment shader which contains a GLSL 1.20 array initializer statement, but it neglects to specify `#version 120` at the top of the shader code. So, the shader does not compile and all that's rendered is plain white polygons.

Also, the test tries to create a very large 3D texture that may exceed the device driver's limit. When this happens, the glTexImage3D call fails and all that's rendered is a white box.

21.2.5 showcase-01

This is actually a DX11 test based on Autodesk's Showcase product. As such, it won't run with Mesa.

CHAPTER 22

Xlib Software Driver

Mesa's Xlib driver provides an emulation of the GLX interface so that OpenGL programs which use the GLX API can render to any X display, even those that don't support the GLX extension. Effectively, the Xlib driver converts all OpenGL rendering into Xlib calls.

The Xlib driver is the oldest Mesa driver and the most mature of Mesa's software-only drivers.

Since the Xlib driver *emulates* the GLX extension, it's not totally conformant with a true GLX implementation. The differences are fairly obscure, however.

The unique features of the Xlib driver follows.

22.1 X Visual Selection

Mesa supports RGB(A) rendering into almost any X visual type and depth.

The glXChooseVisual function tries to choose the best X visual for the given attribute list. However, if this doesn't suit your needs you can force Mesa to use any X visual you want (any supported by your X server that is) by setting the **MESA_RGB_VISUAL** and **MESA_CI_VISUAL** environment variables. When an RGB visual is requested, glXChooseVisual will first look if the MESA_RGB_VISUAL variable is defined. If so, it will try to use the specified visual. Similarly, when a color index visual is requested, glXChooseVisual will look for the MESA_CI_VISUAL variable.

The format of accepted values is: `visual-class depth`

Here are some examples:

```
using csh:
    % setenv MESA_RGB_VISUAL "TrueColor 8"      // 8-bit TrueColor
    % setenv MESA_CI_VISUAL "PseudoColor 12"    // 12-bit PseudoColor
    % setenv MESA_RGB_VISUAL "PseudoColor 8"    // 8-bit PseudoColor

using bash:
    $ export MESA_RGB_VISUAL="TrueColor 8"
```

(continues on next page)

(continued from previous page)

```
$ export MESA_CI_VISUAL="PseudoColor 12"
$ export MESA_RGB_VISUAL="PseudoColor 8"
```

22.2 Double Buffering

Mesa can use either an X Pixmap or XImage as the back color buffer when in double-buffer mode. The default is to use an XImage. The **MESA_BACK_BUFFER** environment variable can override this. The valid values for **MESA_BACK_BUFFER** are: **Pixmap** and **XImage** (only the first letter is checked, case doesn't matter).

Using XImage is almost always faster than a Pixmap since it resides in the application's address space. When glXSwapBuffers() is called, XPutImage() or XShmPutImage() is used to transfer the XImage to the on-screen window.

A Pixmap may be faster when doing remote rendering of a simple scene. Some OpenGL features will be very slow with a Pixmap (for example, blending will require a round-trip message for pixel readback.)

Experiment with the MESA_BACK_BUFFER variable to see which is faster for your application.

22.3 Colormaps

When using Mesa directly or with GLX, it's up to the application writer to create a window with an appropriate colormap. The GLUT toolkit tries to minimize colormap *flashing* by sharing colormaps when possible. Specifically, if the visual and depth of the window matches that of the root window, the root window's colormap will be shared by the Mesa window. Otherwise, a new, private colormap will be allocated.

When sharing the root colormap, Mesa may be unable to allocate the colors it needs, resulting in poor color quality. This can happen when a large number of colorcells in the root colormap are already allocated. To prevent colormap sharing in GLUT, set the **MESA_PRIVATE_CMAP** environment variable. The value isn't significant.

22.4 Gamma Correction

To compensate for the nonlinear relationship between pixel values and displayed intensities, there is a gamma correction feature in Mesa. Some systems, such as Silicon Graphics, support gamma correction in hardware (man gamma) so you won't need to use Mesa's gamma facility. Other systems, however, may need gamma adjustment to produce images which look correct. If you believe that Mesa's images are too dim, read on.

Gamma correction is controlled with the **MESA_GAMMA** environment variable. Its value is of the form **Gr Gg Gb** or just **G** where Gr is the red gamma value, Gg is the green gamma value, Gb is the blue gamma value and G is one gamma value to use for all three channels. Each value is a positive real number typically in the range 1.0 to 2.5. The defaults are all 1.0, effectively disabling gamma correction. Examples:

```
% export MESA_GAMMA="2.3 2.2 2.4"  // separate R,G,B values
% export MESA_GAMMA="2.0"       // same gamma for R,G,B
```

The `demos/gamma.c` program in mesa/demos repository may help you to determine reasonable gamma value for your display. With correct gamma values, the color intensities displayed in the top row (drawn by dithering) should nearly match those in the bottom row (drawn as grays).

Alex De Bruyn reports that gamma values of 1.6, 1.6 and 1.9 work well on HP displays using the HP-ColorRecovery technology.

Mesa implements gamma correction with a lookup table which translates a "linear" pixel value to a gamma-corrected pixel value. There is a small performance penalty. Gamma correction only works in RGB mode. Also be aware that pixel values read back from the frame buffer will not be "un-corrected" so glReadPixels may not return the same data drawn with glDrawPixels.

For more information about gamma correction, see the Wikipedia article

22.5 Overlay Planes

Hardware overlay planes are supported by the Xlib driver. To determine if your X server has overlay support you can test for the SERVER_OVERLAY_VISUALS property:

```
xprop -root | grep SERVER_OVERLAY_VISUALS
```

22.6 HPCR Dithering

If you set the **MESA_HPCR_CLEAR** environment variable then dithering will be used when clearing the color buffer. This is only applicable to HP systems with the HPCR (Color Recovery) feature. This incurs a small performance penalty.

22.7 Extensions

The following Mesa-specific extensions are implemented in the Xlib driver.

22.7.1 GLX_MESA_pixmap_colormap

This extension adds the GLX function:

```
GLXPixmap glXCreateGLXPixmapMESA( Display *dpy, XVisualInfo *visual,
                                  Pixmap pixmap, Colormap cmap )
```

It is an alternative to the standard glXCreateGLXPixmap() function. Since Mesa supports RGB rendering into any X visual, not just True- Color or DirectColor, Mesa needs colormap information to convert RGB values into pixel values. An X window carries this information but a pixmap does not. This function associates a colormap to a GLX pixmap. See the xdemos/glxpixmap.c file for an example of how to use this extension.

GLX_MESA_pixmap_colormap specification

22.7.2 GLX_MESA_release_buffers

Mesa associates a set of ancillary (depth, accumulation, stencil and alpha) buffers with each X window it draws into. These ancillary buffers are allocated for each X window the first time the X window is passed to glXMakeCurrent(). Mesa, however, can't detect when an X window has been destroyed in order to free the ancillary buffers.

The best it can do is to check for recently destroyed windows whenever the client calls the glXCreateContext() or glXDestroyContext() functions. This may not be sufficient in all situations though.

The GLX_MESA_release_buffers extension allows a client to explicitly deallocate the ancillary buffers by calling glxReleaseBuffersMESA() just before an X window is destroyed. For example:

```
#ifdef GLX_MESA_release_buffers
   glXReleaseBuffersMESA( dpy, window );
#endif
XDestroyWindow( dpy, window );
```

GLX_MESA_release_buffers specification

This extension was added in Mesa 2.0.

22.7.3 GLX_MESA_copy_sub_buffer

This extension adds the glXCopySubBufferMESA() function. It works like glXSwapBuffers() but only copies a sub-region of the window instead of the whole window.

GLX_MESA_copy_sub_buffer specification

This extension was added in Mesa 2.6

22.8 Summary of X-related environment variables

```
MESA_RGB_VISUAL - specifies the X visual and depth for RGB mode (X only)
MESA_CI_VISUAL - specifies the X visual and depth for CI mode (X only)
MESA_BACK_BUFFER - specifies how to implement the back color buffer (X only)
MESA_PRIVATE_CMAP - force aux/tk libraries to use private colormaps (X only)
MESA_GAMMA - gamma correction coefficients (X only)
```

CHAPTER 23

Source Code Repository

Mesa uses Git as its source code management system.

The master Git repository is hosted on freedesktop.org.

You may access the repository either as an *anonymous user* (read-only) or as a *developer* (read/write).

You may also browse the main Mesa Git repository and the Mesa demos and tests Git repository.

23.1 Anonymous Git Access

To get the Mesa sources anonymously (read-only):

1. Install the Git software on your computer if needed.
2. Get an initial, local copy of the repository with:

```
git clone https://gitlab.freedesktop.org/mesa/mesa.git
```

3. Later, you can update your tree from the master repository with:

```
git pull origin
```

4. If you also want the Mesa demos/tests repository:

```
git clone https://gitlab.freedesktop.org/mesa/demos.git
```

23.2 Developer Git Access

If you wish to become a Mesa developer with GitLab merge privilege, please follow this procedure:

1. Subscribe to the mesa-dev mailing list.
2. Start contributing to the project by *submitting patches*. Specifically,

- Use GitLab to create your merge requests.
- Wait for someone to review the code and give you a `Reviewed-by` statement.
- You'll have to rely on another Mesa developer to push your initial patches after they've been reviewed.

3. After you've demonstrated the ability to write good code and have had a dozen or so patches accepted, a maintainer may use their discretion to give you access to merge your own code.

23.3 Pushing code to your GitLab account via HTTPS

Useful for people behind strict proxies

You can use personal access tokens to push over HTTPS if ssh will does not suit your needs. In this case, create a token, and put it in the URL as shown here:

```
git remote set-url --push origin https://USER:TOKEN@gitlab.freedesktop.org/your~user~
↪name/mesa.git
```

23.4 Windows Users

If you're using Git on Windows you'll want to enable automatic CR/LF conversion in your local copy of the repository:

```
git config --global core.autocrlf true
```

This will cause Git to convert all text files to CR+LF on checkout, and to LF on commit.

Unix users don't need to set this option.

23.5 Development Branches

At any given time, there may be several active branches in Mesa's repository. Generally, `master` contains the latest development (unstable) code while a branch has the latest stable code.

The command `git branch` will list all available branches.

Questions about branch status/activity should be posted to the mesa-dev mailing list.

23.6 Developer Git Tips

1. Setting up to edit the master branch

   If you try to do a pull by just saying`git pull` and Git complains that you have not specified a branch, try:

   ```
   git config branch.master.remote origin
   git config branch.master.merge master
   ```

   Otherwise, you have to say`git pull origin master` each time you do a pull.

2. Small changes to master

   If you are an experienced Git user working on substantial modifications, you are probably working on a separate branch and would rebase your branch prior to merging with master. But for small changes to the master branch itself, you also need to use the rebase feature in order to avoid an unnecessary and distracting branch in master.

   If it has been awhile since you've done the initial clone, try

   ```
   git pull
   ```

   to get the latest files before you start working.

   Make your changes and use

   ```
   git add <files to commit>
   git commit
   ```

   to get your changes ready to push back into the fd.o repository.

   It is possible (and likely) that someone has changed master since you did your last pull. Even if your changes do not conflict with their changes, Git will make a fast-forward merge branch, branching from the point in time where you did your last pull and merging it to a point after the other changes.

   To avoid this,

   ```
   git pull --rebase
   git push
   ```

   If you are familiar with CVS or similar system, this is similar to doing a `cvs update` in order to update your source tree to the current repository state, instead of the time you did the last update. (CVS doesn't work like Git in this respect, but this is easiest way to explain it.)

   In any case, your repository now looks like you made your changes after all the other changes.

   If the rebase resulted in conflicts or changes that could affect the proper operation of your changes, you'll need to investigate those before doing the push.

   If you want the rebase action to be the default action, then

   ```
   git config branch.master.rebase true
   git config --global branch.autosetuprebase=always
   ```

   See Understanding Git Conceptually for a fairly clear explanation about all of this.

CHAPTER 24

Source Code Tree

This is a brief summary of Mesa's directory tree and what's contained in each directory.

- **docs** - Documentation
- **include** - Public OpenGL header files
- **src**
  - **amd** - AMD-specific sources
    - **addrlib** - common sources for creating images
    - **common** - common code between RADV, radeonsi and ACO
    - **compiler** - ACO shader compiler
    - **llvm** - common code between RADV and radeonsi for compiling shaders using LLVM
    - **registers** - register definitions
    - **vulkan** - RADV Vulkan implementation for AMD Southern Island and newer
  - **compiler** - Common utility sources for different compilers.
    - **glsl** - the GLSL IR and compiler
    - **nir** - the NIR IR and compiler
    - **spirv** - the SPIR-V compiler
  - **egl** - EGL library sources
    - **drivers** - EGL drivers
    - **main** - main EGL library implementation. This is where all the EGL API functions are implemented, like eglCreateContext().
  - **freedreno** - Adreno-specific sources
    - **fdl** - mipmap layout manager
    - **vulkan** - Turnip is a Vulkan implementation for Qualcomm Adreno

- **gbm** - Generic Buffer Manager is a memory allocator for device buffers
- **intel** - Intel-specific sources
  - **blorp** - BLit Or Resolve Pass is a blit and HiZ resolve framework
  - **vulkan** - Anvil is a Vulkan implementation for Intel gen 7 (Ivy Bridge) and newer
- **mapi** - Mesa APIs
  - **glapi** - OpenGL API dispatch layer. This is where all the GL entrypoints like glClear, glBegin, etc. are generated, as well as the GL dispatch table. All GL function calls jump through the dispatch table to functions found in main/.
- **mesa** - Main Mesa sources
  - **main** - The core Mesa code (mainly state management)
  - **drivers** - Mesa drivers (not used with Gallium)
    - **common** - code which may be shared by all drivers
    - **dri** - Direct Rendering Infrastructure drivers
    - **common** - code shared by all DRI drivers
    - **i915** - driver for Intel i915/i945
    - **i965** - driver for Intel i965
    - **nouveau** - driver for nVidia nv04/nv10/nv20
    - **radeon** - driver for ATI R100
    - **r200** - driver for ATI R200
    - **swrast** - software rasterizer driver that uses the swrast module
    - **x11** - Xlib-based software driver
    - **osmesa** - off-screen software driver
  - **math** - vertex array translation and transformation code (not used with Gallium)
  - **program** - Vertex/fragment shader and GLSL compiler code
  - **sparc** - Assembly code/optimizations for SPARC systems (not used with Gallium)
  - **state_tracker** - Translator from Mesa to Gallium. This is basically a Mesa device driver that speaks to Gallium. This directory may be moved to src/mesa/drivers/gallium at some point.
  - **swrast** - Software rasterization module. For drawing points, lines, triangles, bitmaps, images, etc. in software. (not used with Gallium)
  - **swrast_setup** - Software primitive setup. Does things like polygon culling, glPolygonMode, polygon offset, etc. (not used with Gallium)
  - **tnl** - Software vertex Transformation 'n Lighting. (not used with Gallium)
  - **tnl_dd** - TNL code for device drivers. (not used with Gallium)
  - **vbo** - Vertex Buffer Object code. All drawing with glBegin/glEnd, glDrawArrays, display lists, etc. goes through this module. The results is a well-defined set of vertex arrays which are passed to the device driver (or tnl module) for rendering.
  - **x86** - Assembly code/optimizations for 32-bit x86 systems (not used with Gallium)
  - **x86-64** - Assembly code/optimizations for 64-bit x86 systems (not used with Gallium)

- **gallium** - Gallium3D source code
  - **include** - Gallium3D header files which define the Gallium3D interfaces
  - **drivers** - Gallium3D device drivers
    - **etnaviv** - Driver for Vivante.
    - **freedreno** - Driver for Qualcomm Adreno.
    - **i915** - Driver for Intel i915/i945.
    - **iris** - Driver for Intel gen 8 (Broadwell) and newer.
    - **lima** - Driver for ARM Mali-400 (Utgard) series.
    - **llvmpipe** - Software driver using LLVM for runtime code generation.
    - **nouveau** - Driver for NVIDIA GPUs.
    - **panfrost** - Driver for ARM Mali Txxx (Midgard) and Gxx (Bifrost) GPUs.
    - **radeon** - Shared module for the r600 and radeonsi drivers.
    - **r300** - Driver for ATI R300 - R500.
    - **r600** - Driver for ATI/AMD R600 - Northern Island (Terascale).
    - **radeonsi** - Driver for AMD Southern Island and newer (GCN, RDNA).
    - **softpipe** - Software reference driver.
    - **svga** - Driver for VMware's SVGA virtual GPU.
    - **swr** - Software driver with massively parellel vertex processing.
    - **tegra** - Driver for NVIDIA Tegra GPUs.
    - **v3d** - Driver for Broadcom VideoCore 5 and newer.
    - **vc4** - Driver for Broadcom VideoCore 4.
    - **virgl** - Driver for Virtio virtual GPU of QEMU.
    - **zink** - Driver that uses Vulkan for rendering.
  - **auxiliary** - Gallium support code
    - **cso_cache** - Constant State Objects Cache. Used to filter out redundant state changes between frontends and drivers.
    - **draw** - Software vertex processing and primitive assembly module. This includes vertex program execution, clipping, culling and optional stages for drawing wide lines, stippled lines, polygon stippling, two-sided lighting, etc. Intended for use by drivers for hardware that does not have vertex shaders. Geometry shaders will also be implemented in this module.
    - **gallivm** - LLVM module for Gallium. For LLVM-based compilation, optimization and code generation for TGSI shaders. Incomplete.
    - **hud** - Heads-Up Display, an overlay showing GPU statistics
    - **pipebuffer** - utility module for managing buffers
    - **rbug** - Gallium remote debug utility
    - **rtasm** - run-time assembly/machine code generation. Currently there's run-time code generation for x86/SSE, PowerPC and Cell SPU.
    - **tessellator**- used by software drivers to implement tessellation shaders

        - **tgsi** - TG Shader Infrastructure. Code for encoding, manipulating and interpreting GPU programs.
        - **translate** - module for translating vertex data from one format to another.
        - **util** - assorted utilities for arithmetic, hashing, surface creation, memory management, 2D blitting, simple rendering, etc.
        - **vl** - utility code for video decode/encode
        - XXX more
    - **frontends** - These implement various libraries using the device drivers
        - **clover** - OpenCL frontend
        - **dri** - Meta frontend for DRI drivers, see mesa/state_tracker
        - **glx** - Meta frontend for GLX
        - **hgl** - Haiku OpenGL
        - **nine** - D3D9 frontend, see targets/d3dadapter9
        - **omx** - OpenMAX Bellagio frontend
        - **osmesa** - Off-screen OpenGL rendering library
        - **va** - VA-API frontend
        - **vdpau** - VDPAU frontend
        - **wgl** - Windows WGL frontend
        - **xa** - XA frontend
        - **xvmc** - XvMC frontend
    - **winsys** - The device drivers are platform-independent, the winsys connects them to various platforms. There is usually one winsys per device family, and within the winsys directory there can be multiple flavors connecting to different platforms.
        - **drm** - Direct Rendering Manager on Linux
        - **gdi** - Windows
        - **xlib** - indirect rendering on X Window System
        - XXX more
- **targets** - These control how the Gallium code is compiled into different libraries. Each of these roughly corresponds to one frontend.
    - **d3dadapter9** - d3dadapter9.so for Wine
    - **dri** - libgallium_dri.so loaded by libGL.so
    - **graw** - raw Gallium interface without a frontend
    - XXX more
- **glx** - The GLX library code for building libGL.so using DRI drivers.
- **loader** - Used by libGL.so to find and load the appropriate DRI driver.
- **panfrost** - Panfrost-specific sources
    - **bifrost** - shader compiler for the Bifrost generation GPUs
    - **midgard** - shader compiler for the Midgard generation GPUs

    - **pandecode** - command stream debugger
- **util** - Various utility codes
- **vulkan** - Common code for Vulkan drivers

CHAPTER 25

Development Utilities

Mesa demos collection includes several utility routines in the `src/util/` directory.

Piglit is an open-source test suite for OpenGL implementations.

ApiTrace is a project to trace, analyze and debug graphics api's.

Valgrind is a very useful tool for tracking down memory-related problems in your code.

Coverity provides static code analysis of Mesa. If you create an account you can see the results and try to fix outstanding issues.

CHAPTER 26

Help Wanted

We can always use more help with the Mesa project. Here are some specific ideas and areas where help would be appreciated:

1. **Driver patching and testing.** Patches are often posted to the mesa-dev mailing list, but aren't immediately checked into Git because not enough people are testing them. Just applying patches, testing and reporting back is helpful.
2. **Driver debugging.** There are plenty of open bugs in the bug database.
3. **Remove aliasing warnings.** Enable gcc's `-Wstrict-aliasing=2 -fstrict-aliasing` arguments, and track down aliasing issues in the code.
4. **Contribute more tests to** Piglit.

You can find some further To-do lists here:

Common To-Do lists:

- features.txt - Status of OpenGL 3.x / 4.x features in Mesa.

Legacy Driver specific To-Do lists:

- r600g - Driver for ATI/AMD R600 - Northern Island.
- r300g - Driver for ATI R300 - R500.

If you want to do something new in Mesa, first join the Mesa developer's mailing list. Then post a message to propose what you want to do, just to make sure there's no issues.

Anyone is welcome to contribute code to the Mesa project. By doing so, it's assumed that you agree to the code's licensing terms.

Finally:

1. Try to write high-quality code that follows the existing style.
2. Use uniform indentation, write comments, use meaningful identifiers, etc.
3. Test your code thoroughly. Include test programs if appropriate.

CHAPTER 27

Development Notes

27.1 Adding Extensions

To add a new GL extension to Mesa you have to do at least the following.

- If `glext.h` doesn't define the extension, edit `include/GL/gl.h` and add code like this:

```
#ifndef GL_EXT_the_extension_name
#define GL_EXT_the_extension_name 1
/* declare the new enum tokens */
/* prototype the new functions */
/* TYPEDEFS for the new functions */
#endif
```

- In the `src/mapi/glapi/gen/` directory, add the new extension functions and enums to the `gl_API.xml` file. Then, a bunch of source files must be regenerated by executing the corresponding Python scripts.
- Add a new entry to the `gl_extensions` struct in `mtypes.h` if the extension requires driver capabilities not already exposed by another extension.
- Add a new entry to the `src/mesa/main/extensions_table.h` file.
- From this point, the best way to proceed is to find another extension, similar to the new one, that's already implemented in Mesa and use it as an example.
- If the new extension adds new GL state, the functions in `get.c`, `enable.c` and `attrib.c` will most likely require new code.
- To determine if the new extension is active in the current context, use the auto-generated `_mesa_has_##name_str()` function defined in `src/mesa/main/extensions.h`.
- The dispatch tests `check_table.cpp` and `dispatch_sanity.cpp` should be updated with details about the new extensions functions. These tests are run using `meson test`.

CHAPTER 28

Coding Style

Mesa is over 20 years old and the coding style has evolved over time. Some old parts use a style that's a bit out of date. Different sections of mesa can use different coding style as set in the local EditorConfig (.editorconfig) and/or Emacs (.dir-locals.el) file. Alternatively the following is applicable. If the guidelines below don't cover something, try following the format of existing, neighboring code.

Basic formatting guidelines

- 3-space indentation, no tabs.
- Limit lines to 78 or fewer characters. The idea is to prevent line wrapping in 80-column editors and terminals. There are exceptions, such as if you're defining a large, static table of information.
- Opening braces go on the same line as the if/for/while statement. For example:

```
if (condition) {
   foo;
} else {
   bar;
}
```

- Put a space before/after operators. For example, `a = b + c;` and not `a=b+c;`
- This GNU indent command generally does the right thing for formatting:

```
indent -br -i3 -npcs --no-tabs infile.c -o outfile.c
```

- Use comments wherever you think it would be helpful for other developers. Several specific cases and style examples follow. Note that we roughly follow Doxygen conventions.

  Single-line comments:

```
/* null-out pointer to prevent dangling reference below */
bufferObj = NULL;
```

  Or,

```
bufferObj = NULL;  /* prevent dangling reference below */
```

Multi-line comment:

```
/* If this is a new buffer object id, or one which was generated but
 * never used before, allocate a buffer object now.
 */
```

We try to quote the OpenGL specification where prudent:

```
/* Page 38 of the PDF of the OpenGL ES 3.0 spec says:
 *
 *     "An INVALID_OPERATION error is generated for any of the following
 *     conditions:
 *
 *     * <length> is zero."
 *
 * Additionally, page 94 of the PDF of the OpenGL 4.5 core spec
 * (30.10.2014) also says this, so it's no longer allowed for desktop GL,
 * either.
 */
```

Function comment example:

```
/**
 * Create and initialize a new buffer object.  Called via the
 * ctx->Driver.CreateObject() driver callback function.
 * \param  name  integer name of the object
 * \param  type  one of GL_FOO, GL_BAR, etc.
 * \return  pointer to new object or NULL if error
 */
struct gl_object *
_mesa_create_object(GLuint name, GLenum type)
{
   /* function body */
}
```

- Put the function return type and qualifiers on one line and the function name and parameters on the next, as seen above. This makes it easy to use `grep ^function_name dir/*` to find function definitions. Also, the opening brace goes on the next line by itself (see above.)
- Function names follow various conventions depending on the type of function:

```
glFooBar()        - a public GL entry point (in glapi_dispatch.c)
_mesa_FooBar()    - the internal immediate mode function
save_FooBar()     - retained mode (display list) function in dlist.c
foo_bar()         - a static (private) function
_mesa_foo_bar()   - an internal non-static Mesa function
```

- Constants, macros and enum names are `ALL_UPPERCASE`, with _ between words.
- Mesa usually uses camel case for local variables (Ex: `localVarname`) while Gallium typically uses underscores (Ex: `local_var_name`).
- Global variables are almost never used because Mesa should be thread-safe.
- Booleans. Places that are not directly visible to the GL API should prefer the use of `bool`, `true`, and `false` over `GLboolean`, `GL_TRUE`, and `GL_FALSE`. In C code, this may mean that `#include <stdbool.h>`

needs to be added. The `try_emit_*` methods in `src/mesa/program/ir_to_mesa.cpp` and `src/mesa/state_tracker/st_glsl_to_tgsi.cpp` can serve as examples.

CHAPTER 29

Submitting Patches

29.1 Basic guidelines

- Patches should not mix code changes with code formatting changes (except, perhaps, in very trivial cases.)
- Code patches should follow Mesa *coding conventions*.
- Whenever possible, patches should only affect individual Mesa/Gallium components.
- Patches should never introduce build breaks and should be bisectable (see `Git bisect`.)
- Patches should be properly *formatted*.
- Patches should be sufficiently *tested* before submitting.
- Patches should be *submitted* via a merge request for *review*.

29.2 Patch formatting

- Lines should be limited to 75 characters or less so that Git logs displayed in 80-column terminals avoid line wrapping. Note that `git log` uses 4 spaces of indentation (4 + 75 < 80).
- The first line should be a short, concise summary of the change prefixed with a module name. Examples:

```
mesa: Add support for querying GL_VERTEX_ATTRIB_ARRAY_LONG

gallium: add PIPE_CAP_DEVICE_RESET_STATUS_QUERY

i965: Fix missing type in local variable declaration.
```

- Subsequent patch comments should describe the change in more detail, if needed. For example:

```
i965: Remove end-of-thread SEND alignment code.

This was present in Eric's initial implementation of the compaction code
for Sandybridge (commit 077d01b6). There is no documentation saying this
is necessary, and removing it causes no regressions in piglit on any
platform.
```

- A "Signed-off-by:" line is not required, but not discouraged either.
- If a patch addresses an issue in GitLab, use the Closes: tag For example:

```
Closes: https://gitlab.freedesktop.org/mesa/mesa/-/issues/1
```

  Prefer the full URL to just `Closes: #1`, since the URL makes it easier to get to the bug page from `git log`

  Do not use the ''Fixes:'' tag for this! Mesa already uses `Fixes:` for something else. See *below*.

- If there have been several revisions to a patch during the review process, they should be noted such as in this example:

```
st/mesa: add ARB_texture_stencil8 support (v4)

if we support stencil texturing, enable texture_stencil8
there is no requirement to support native S8 for this,
the texture can be converted to x24s8 fine.

v2: fold fixes from Marek in:
   a) put S8 last in the list
   b) fix renderable to always test for d/s renderable
     fixup the texture case to use a stencil only format
     for picking the format for the texture view.
v3: hit fallback for getteximage
v4: put s8 back in front, it shouldn't get picked now (Ilia)
```

- If someone tested your patch, document it with a line like this:

```
Tested-by: Joe Hacker <jhacker@foo.com>
```

- If the patch was reviewed (usually the case) or acked by someone, that should be documented with:

```
Reviewed-by: Joe Hacker <jhacker@foo.com>
Acked-by: Joe Hacker <jhacker@foo.com>
```

- When updating a merge request add all the tags (`Acked-by:`, `Reviewed-by:`, `Fixes:`, `Cc: mesa-stable` and/or other) to the commit messages. This provides reviewers with quick feedback if the patch has already been reviewed.

29.3 The `Fixes:` tag

If a patch addresses a issue introduced with earlier commit, that should be noted in the commit message. For example:

```
Fixes: d7b3707c612 ("util/disk_cache: use stat() to check if entry is a directory")
```

You can produce those fixes lines by running this command once:

```
git config --global alias.fixes "show -s --pretty='format:Fixes: %h (\"%s\")'"
```

After that, using `git fixes <sha1>` will print the full line for you.

29.3.1 The stable tag

If you want a commit to be applied to a stable branch, you should add an appropriate note to the commit message.

Using a `Fixes:` tag as described in *Patch formatting* is the preferred way to nominate a commit that should be backported. There are scripts that will figure out which releases to apply the patch to automatically, so you don't need to figure it out.

Alternatively, you may use a "CC:" tag. Here are some examples of such a note:

```
Cc: mesa-stable
Cc: 20.0 <mesa-stable>
CC: 20.0 19.3 <mesa-stable>
```

Using the CC tag **should** include the stable branches you want to nominate the patch to. If you do not provide any version it is nominated to all active stable branches.

29.4 Testing Patches

It should go without saying that patches must be tested. In general, do whatever testing is prudent.

You should always run the Mesa test suite before submitting patches. The test suite can be run using the 'meson test' command. All tests must pass before patches will be accepted, this may mean you have to update the tests themselves.

Whenever possible and applicable, test the patch with Piglit and/or dEQP to check for regressions.

As mentioned at the beginning, patches should be bisectable. A good way to test this is to make use of the 'git rebase' command, to run your tests on each commit. Assuming your branch is based off `origin/master`, you can run:

```
$ git rebase --interactive --exec "meson test -C build/" origin/master
```

replacing `"meson test"` with whatever other test you want to run.

29.5 Submitting Patches

Patches are submitted to the Mesa project via a GitLab Merge Request.

Add labels to your MR to help reviewers find it. For example:

- Mesa changes affecting all drivers: mesa
- Hardware vendor specific code: amd, intel, nvidia, ...
- Driver specific code: anvil, freedreno, i965, iris, radeonsi, radv, vc4, ...
- Other tag examples: gallium, util

Tick the following when creating the MR. It allows developers to rebase your work on top of master.

```
Allow commits from members who can merge to the target branch
```

If you revise your patches based on code review and push an update to your branch, you should maintain a **clean** history in your patches. There should not be "fixup" patches in the history. The series should be buildable and functional after every commit whenever you push the branch.

It is your responsibility to keep the MR alive and making progress, as there are no guarantees that a Mesa dev will independently take interest in it.

Some other notes:

- Make changes and update your branch based on feedback
- After an update, for the feedback you handled, close the feedback discussion with the "Resolve Discussion" button. This way the reviewers know which feedback got handled and which didn't.
- Old, stale MR may be closed, but you can reopen it if you still want to pursue the changes
- You should periodically check to see if your MR needs to be rebased
- Make sure your MR is closed if your patches get pushed outside of GitLab
- Please send MRs from a personal fork rather than from the main Mesa repository, as it clutters it unnecessarily.

29.6 Reviewing Patches

To participate in code review, you can monitor the GitLab Mesa Merge Requests page, and/or register for notifications in your GitLab settings.

When you've reviewed a patch, please be unambiguous about your review. That is, state either

```
Reviewed-by: Joe Hacker <jhacker@foo.com>
```

or

```
Acked-by: Joe Hacker <jhacker@foo.com>
```

Rather than saying just "LGTM" or "Seems OK".

If small changes are suggested, it's OK to say something like:

```
With the above fixes, Reviewed-by: Joe Hacker <jhacker@foo.com>
```

which tells the patch author that the patch can be committed, as long as the issues are resolved first.

These Reviewed-by, Acked-by, and Tested-by tags should also be amended into commits in a MR before it is merged.

When providing a Reviewed-by, Acked-by, or Tested-by tag in a GitLab MR, enclose the tag in backticks:

```
`Reviewed-by: Joe Hacker <jhacker@example.com>`
```

This is the markdown format for literal, and will prevent GitLab from hiding the < and > symbols.

Review by non-experts is encouraged. Understanding how someone else goes about solving a problem is a great way to learn your way around the project. The submitter is expected to evaluate whether they have an appropriate amount of review feedback from people who also understand the code before merging their patches.

29.7 Nominating a commit for a stable branch

There are several ways to nominate a patch for inclusion in the stable branch and release. In order or preference:

- By adding the `Fixes:` tag in the commit message as described above, if you are fixing a specific commit.
- By adding the `Cc: mesa-stable` tag in the commit message as described above.
- By submitting a merge request against the `staging/year.quarter` branch on GitLab.

Please **DO NOT** send patches to mesa-stable@lists.freedesktop.org, it is not monitored actively and is a historical artifact.

If you are not the author of the original patch, please Cc: them in your nomination request.

The current patch status can be observed in the *staging branch*.

29.8 Criteria for accepting patches to the stable branch

Mesa has a designated release manager for each stable branch, and the release manager is the only developer that should be pushing changes to these branches. Everyone else should nominate patches using the mechanism described above. The following rules define which patches are accepted and which are not. The stable-release manager is also given broad discretion in rejecting patches that have been nominated.

- Patch must conform with the *Basic guidelines*
- Patch must have landed in master first. In case where the original patch is too large and/or otherwise contradicts with the rules set within, a backport is appropriate.
- It must not introduce a regression - be that build or runtime wise.

  Note: If the regression is due to faulty piglit/dEQP/CTS/other test the latter must be fixed first. A reference to the offending test(s) and respective fix(es) should be provided in the nominated patch.

- Patch cannot be larger than 100 lines.
- Patches that move code around with no functional change should be rejected.
- Patch must be a bug fix and not a new feature.

  Note: An exception to this rule, are hardware-enabling "features". For example, *backports* of new code to support a newly-developed hardware product can be accepted if they can be reasonably determined not to have effects on other hardware.

- Patch must be reviewed, For example, the commit message has Reviewed-by, Signed-off-by, or Tested-by tags from someone but the author.
- Performance patches are considered only if they provide information about the hardware, program in question and observed improvement. Use numbers to represent your measurements.

If the patch complies with the rules it will be *cherry-picked*. Alternatively the release manager will reply to the patch in question stating why the patch has been rejected or would request a backport. The stable-release manager may at times need to force-push changes to the stable branches, for example, to drop a previously-picked patch that was later identified as causing a regression). These force-pushes may cause changes to be lost from the stable branch if developers push things directly. Consider yourself warned.

29.9 Sending backports for the stable branch

By default merge conflicts are resolved by the stable-release manager. The release maintainer should resolve trivial conflicts, but for complex conflicts they should ask the original author to provide a backport or de-nominate the patch.

For patches that either need to be nominated after they've landed in master, or that are known ahead of time to not not apply cleanly to a stable branch (such as due to a rename), using a GitLab MR is most appropriate. The MR should be based on and target the staging/year.quarter branch, not on the year.quarter branch, per the stable branch policy. Assigning the MR to release maintainer for said branch or mentioning them is helpful, but not required.

29.10 Git tips

- `git rebase -i ...` is your friend. Don't be afraid to use it.
- Apply a fixup to commit FOO.

```
git add ...
git commit --fixup=FOO
git rebase -i --autosquash ...
```

- Test for build breakage between patches e.g last 8 commits.

```
git rebase -i --exec="ninja -C build/" HEAD~8
```

CHAPTER 30

Releasing Process

30.1 Overview

This document uses the convention X.Y.Z for the release number with X.Y being the stable branch name.

Mesa provides feature and bugfix releases. Former use zero as patch version (Z), while the latter have a non-zero one.

For example:

```
Mesa 10.1.0 - 10.1 branch, feature
Mesa 10.1.4 - 10.1 branch, bugfix
Mesa 12.0.0 - 12.0 branch, feature
Mesa 12.0.2 - 12.0 branch, bugfix
```

30.2 Release schedule

Releases should happen on Wednesdays. Delays can occur although those should be kept to a minimum.

See our *calendar* for information about how the release schedule is planned, and the date and other details for individual releases.

30.3 Feature releases

- Available approximately every three months.
- Initial time plan available 2-4 weeks before the planned branchpoint (rc1) on the mesa-announce@ mailing list.
- Typically, the final release will happen after 4 candidates. Additional ones may be needed in order to resolve blocking regressions, though.

30.4 Stable releases

- Normally available once every two weeks.
- Only the latest branch has releases. See note below.

Note: There is one or two releases overlap when changing branches. For example:

The final release from the 12.0 series Mesa 12.0.5 will be out around the same time (or shortly after) 13.0.1 is out.

This also involves that, as a final release may be delayed due to the need of additional candidates to solve some blocking regression(s), the release manager might have to update the *calendar* with additional bug fix releases of the current stable branch.

30.5 Cherry-picking and testing

Commits nominated for the active branch are picked as based on the *criteria* as described in the same section.

Nominations happen via special tags in the commit messages, and via GitLab merge requests against the staging branches. There are special scripts used to read the tags.

The maintainer should watch or be in contact with the Intel CI team, as well as watch the GitLab CI for regressions.

Cherry picking should be done with the '-x' switch (to automatically add "cherry picked from ..." to the commit message):

```
git cherry-pick -x abcdef12345667890
```

Developers can request, *as an exception*, patches to be applied up-to the last one hour before the actual release. This is made **only** with explicit permission/request, and the patch **must** be very well contained. Thus it cannot affect more than one driver/subsystem.

Following developers have requested permanent exception

- *Ilia Mirkin*
- *AMD team*

The GitLab CI must pass.

For Windows related changes, the main contact point is Brian Paul. Jose Fonseca can also help as a fallback contact.

For Android related changes, the main contact is Tapani Pälli. Mauro Rossi is collaborating with Android-x86 and may provide feedback about the build status in that project.

For MacOSX related changes, Jeremy Huddleston Sequoia is currently a good contact point.

Note: If a patch in the current queue needs any additional fix(es), then they should be squashed together. The commit messages and the "`cherry picked from`"-tags must be preserved.

```
git show b10859ec41d09c57663a258f43fe57c12332698e

commit b10859ec41d09c57663a258f43fe57c12332698e
Author: Jonas Pfeil <pfeiljonas@gmx.de>
Date:   Wed Mar 1 18:11:10 2017 +0100
```

(continues on next page)

(continued from previous page)

```
    ralloc: Make sure ralloc() allocations match malloc()'s alignment.

    The header of ralloc needs to be aligned, because the compiler assumes
    ...

    (cherry picked from commit cd2b55e536dc806f9358f71db438dd9c246cdb14)

    Squashed with commit:

    ralloc: don't leave out the alignment factor

    Experimentation shows that without alignment factor gcc and clang choose
    ...

    (cherry picked from commit ff494fe999510ea40e3ed5827e7818550b6de126)
```

30.6 Regression/functionality testing

- *no regressions should be observed for Piglit/dEQP/CTS/Vulkan on Intel platforms*
- *no regressions should be observed for Piglit using the swrast, softpipe and llvmpipe drivers*

30.7 Staging branch

A live branch, which contains the currently merge/rejected patches is available in the main repository under `staging/X.Y`. For example:

```
staging/18.1 - WIP branch for the 18.1 series
staging/18.2 - WIP branch for the 18.2 series
```

Notes:

- People are encouraged to test the staging branch and report regressions.
- The branch history is not stable and it **will** be rebased,

30.8 Making a branchpoint

A branchpoint is made such that new development can continue in parallel to stabilization and bugfixing.

Note: Before doing a branch ensure that basic build and `meson test` testing is done and there are little to-no issues. Ideally all of those should be tackled already.

Check if the version number is going to remain as, alternatively `git mv docs/relnotes/{current,new}.rst` as appropriate.

To setup the branchpoint:

```
git checkout master # make sure we're in master first
git tag -s X.Y-branchpoint -m "Mesa X.Y branchpoint"
git checkout -b X.Y
git checkout master
$EDITOR VERSION # bump the version number
git commit -as
truncate docs/relnotes/new_features.txt
git commit -a
git push origin X.Y-branchpoint X.Y
```

Now go to GitLab and add the new Mesa version X.Y.

Check that there are no distribution breaking changes and revert them if needed. For example: files being overwritten on install, etc. Happens extremely rarely - we had only one case so far (see commit 2ced8eb136528914e1bf4e000dea06a9d53c7e04).

30.9 Making a new release

These are the instructions for making a new Mesa release.

30.9.1 Get latest source files

Ensure the latest code is available - both in your local master and the relevant branch.

30.9.2 Perform basic testing

Most of the testing should already be done during the *cherry-pick* So we do a quick 'touch test'

- meson dist
- scons (from release tarball)
- the produced binaries work

Here is one solution:

```
__glxgears_cmd='glxgears 2>&1 | grep -v "configuration file"'
__es2info_cmd='es2_info 2>&1 | egrep "GL_VERSION|GL_RENDERER|.*dri\.so"'
__es2gears_cmd='es2gears_x11 2>&1 | grep -v "configuration file"'
test "x$LD_LIBRARY_PATH" != 'x' && __old_ld="$LD_LIBRARY_PATH"
export LD_LIBRARY_PATH=`pwd`/test/usr/local/lib/:"${__old_ld}"
export LIBGL_DRIVERS_PATH=`pwd`/test/usr/local/lib/dri/
export LIBGL_DEBUG=verbose
eval $__glxinfo_cmd
eval $__glxgears_cmd
eval $__es2info_cmd
eval $__es2gears_cmd
export LIBGL_ALWAYS_SOFTWARE=true
eval $__glxinfo_cmd
eval $__glxgears_cmd
eval $__es2info_cmd
eval $__es2gears_cmd
export LIBGL_ALWAYS_SOFTWARE=true
export GALLIUM_DRIVER=softpipe
```

(continues on next page)

(continued from previous page)

```
eval $__glxinfo_cmd
eval $__glxgears_cmd
eval $__es2info_cmd
eval $__es2gears_cmd
# Smoke test DOTA2
unset LD_LIBRARY_PATH
test "x$__old_ld" != 'x' && export LD_LIBRARY_PATH="$__old_ld" && unset __old_ld
unset LIBGL_DRIVERS_PATH
unset LIBGL_DEBUG
unset LIBGL_ALWAYS_SOFTWARE
unset GALLIUM_DRIVER
export VK_ICD_FILENAMES=`pwd`/test/usr/local/share/vulkan/icd.d/intel_icd.x86_64.json
steam steam://rungameid/570  -vconsole -vulkan
unset VK_ICD_FILENAMES
```

30.9.3 Create release notes for the new release

The release notes are completely generated by the `bin/gen_release_notes.py` script. Simply run this script **before** bumping the version. You'll need to come back to this file once the tarball is generated to add its `sha256sum`.

Increment the version contained in the file `VERSION` at Mesa's top-level, then commit this change and **push the branch** (if you forget to do this, `release.sh` below will fail).

30.9.4 Use the release.sh script from xorg util-modular

Start the release process.

```
../relative/path/to/release.sh . # append --dist if you've already done distcheck␣
↪above
```

Pay close attention to the prompts as you might be required to enter your GPG and SSH passphrase(s) to sign and upload the files, respectively.

Ensure that you do sign the tarballs, that your key is mentioned in the release notes, and is published in release-maintainers-keys.asc.

30.9.5 Add the sha256sums to the release notes

Edit `docs/relnotes/X.Y.Z.rst` to add the `sha256sum` as available in the `mesa-X.Y.Z.announce` template. Commit this change.

30.9.6 Back on mesa master, add the new release notes into the tree

Something like the following steps will do the trick:

```
git cherry-pick -x X.Y~1
git cherry-pick -x X.Y
```

Then run the

```
./bin/post_version.py X.Y.Z
```

, where X.Y.Z is the version you just made. This will update docs/relnotes.rst and docs/release-calendar.rst. It will then generate a Git commit automatically. Check that everything looks correct and push:

```
git push origin master X.Y
```

30.10 Announce the release

Use the generated template during the releasing process.

Again, pay attention to add a note to warn about a final release in a series, if that is the case.

30.11 Update GitLab issues

Parse through the bug reports as listed in the docs/relnotes/X.Y.Z.rst document. If there's outstanding action, close the bug referencing the commit ID which addresses the bug and mention the Mesa version that has the fix.

CHAPTER 31

Release Calendar

31.1 Overview

Mesa provides feature/development and stable releases.

The table below lists the date and release manager that is expected to do the specific release.

Regular updates will ensure that the schedule for the current and the next two feature releases are shown in the table.

In order to keep the whole releasing team up to date with the tools used, best practices and other details, the member in charge of the next feature release will be in constant rotation.

The way the release schedule works is explained *here*.

Take a look *here* if you'd like to nominate a patch in the next stable release.

31.2 Calendar

Branch	Expected date	Release	Release manager	Notes
20.2	2020-10-28	20.2.2	Dylan Baker	
	2020-11-11	20.2.3	Dylan Baker	
	2020-11-24	20.2.4	Dylan Baker	
20.3	2020-11-04	20.3.0-rc1	Dylan Baker	
	2020-11-11	20.3.0-rc2	Dylan Baker	
	2020-11-18	20.3.0-rc3	Dylan Baker	
	2020-11-25	20.3.0-rc4	Dylan Baker	or 20.3.0 final

CHAPTER 32

Source Documentation

Doxygen is used to automatically produce cross-referenced documentation from the Mesa source code.

The Doxygen configuration files and generated files are not included in the normal Mesa distribution (they're very large). To generate Doxygen documentation, download Mesa from Git, change to the `doxygen` directory and run `make`.

For an example of Doxygen usage in Mesa, see a recent source file such as bufferobj.c.

If you're reading this page from your local copy of Mesa, and have run the doxygen scripts, you can read the documentation here

CHAPTER 33

GL Dispatch

Several factors combine to make efficient dispatch of OpenGL functions fairly complicated. This document attempts to explain some of the issues and introduce the reader to Mesa's implementation. Readers already familiar with the issues around GL dispatch can safely skip ahead to the *overview of Mesa's implementation*.

33.1 1. Complexity of GL Dispatch

Every GL application has at least one object called a GL *context*. This object, which is an implicit parameter to every GL function, stores all of the GL related state for the application. Every texture, every buffer object, every enable, and much, much more is stored in the context. Since an application can have more than one context, the context to be used is selected by a window-system dependent function such as `glXMakeContextCurrent`.

In environments that implement OpenGL with X-Windows using GLX, every GL function, including the pointers returned by `glXGetProcAddress`, are *context independent*. This means that no matter what context is currently active, the same `glVertex3fv` function is used.

This creates the first bit of dispatch complexity. An application can have two GL contexts. One context is a direct rendering context where function calls are routed directly to a driver loaded within the application's address space. The other context is an indirect rendering context where function calls are converted to GLX protocol and sent to a server. The same `glVertex3fv` has to do the right thing depending on which context is current.

Highly optimized drivers or GLX protocol implementations may want to change the behavior of GL functions depending on current state. For example, `glFogCoordf` may operate differently depending on whether or not fog is enabled.

In multi-threaded environments, it is possible for each thread to have a different GL context current. This means that poor old `glVertex3fv` has to know which GL context is current in the thread where it is being called.

33.2 2. Overview of Mesa's Implementation

Mesa uses two per-thread pointers. The first pointer stores the address of the context current in the thread, and the second pointer stores the address of the *dispatch table* associated with that context. The dispatch table stores pointers

to functions that actually implement specific GL functions. Each time a new context is made current in a thread, these pointers a updated.

The implementation of functions such as `glVertex3fv` becomes conceptually simple:

- Fetch the current dispatch table pointer.
- Fetch the pointer to the real `glVertex3fv` function from the table.
- Call the real function.

This can be implemented in just a few lines of C code. The file `src/mesa/glapi/glapitemp.h` contains code very similar to this.

Listing 1: Sample dispatch function

```
void glVertex3f(GLfloat x, GLfloat y, GLfloat z)
{
    const struct _glapi_table * const dispatch = GET_DISPATCH();

    (*dispatch->Vertex3f)(x, y, z);
}
```

The problem with this simple implementation is the large amount of overhead that it adds to every GL function call.

In a multithreaded environment, a naive implementation of `GET_DISPATCH` involves a call to `pthread_getspecific` or a similar function. Mesa provides a wrapper function called `_glapi_get_dispatch` that is used by default.

33.3 3. Optimizations

A number of optimizations have been made over the years to diminish the performance hit imposed by GL dispatch. This section describes these optimizations. The benefits of each optimization and the situations where each can or cannot be used are listed.

33.3.1 3.1. Dual dispatch table pointers

The vast majority of OpenGL applications use the API in a single threaded manner. That is, the application has only one thread that makes calls into the GL. In these cases, not only do the calls to `pthread_getspecific` hurt performance, but they are completely unnecessary! It is possible to detect this common case and avoid these calls.

Each time a new dispatch table is set, Mesa examines and records the ID of the executing thread. If the same thread ID is always seen, Mesa knows that the application is, from OpenGL's point of view, single threaded.

As long as an application is single threaded, Mesa stores a pointer to the dispatch table in a global variable called `_glapi_Dispatch`. The pointer is also stored in a per-thread location via `pthread_setspecific`. When Mesa detects that an application has become multithreaded, `NULL` is stored in `_glapi_Dispatch`.

Using this simple mechanism the dispatch functions can detect the multithreaded case by comparing `_glapi_Dispatch` to `NULL`. The resulting implementation of `GET_DISPATCH` is slightly more complex, but it avoids the expensive `pthread_getspecific` call in the common case.

Listing 2: Improved `GET_DISPATCH` Implementation

```
#define GET_DISPATCH() \
    (_glapi_Dispatch != NULL) \
        ? _glapi_Dispatch : pthread_getspecific(&_glapi_Dispatch_key)
```

33.3.2 3.2. ELF TLS

Starting with the 2.4.20 Linux kernel, each thread is allocated an area of per-thread, global storage. Variables can be put in this area using some extensions to GCC. By storing the dispatch table pointer in this area, the expensive call to `pthread_getspecific` and the test of `_glapi_Dispatch` can be avoided.

The dispatch table pointer is stored in a new variable called `_glapi_tls_Dispatch`. A new variable name is used so that a single libGL can implement both interfaces. This allows the libGL to operate with direct rendering drivers that use either interface. Once the pointer is properly declared, `GET_DISPACH` becomes a simple variable reference.

Listing 3: TLS `GET_DISPATCH` Implementation

```
extern __thread struct _glapi_table *_glapi_tls_Dispatch
    __attribute__((tls_model("initial-exec")));

#define GET_DISPATCH() _glapi_tls_Dispatch
```

Use of this path is controlled by the preprocessor define `USE_ELF_TLS`. Any platform capable of using ELF TLS should use this as the default dispatch method.

33.3.3 3.3. Assembly Language Dispatch Stubs

Many platforms has difficulty properly optimizing the tail-call in the dispatch stubs. Platforms like x86 that pass parameters on the stack seem to have even more difficulty optimizing these routines. All of the dispatch routines are very short, and it is trivial to create optimal assembly language versions. The amount of optimization provided by using assembly stubs varies from platform to platform and application to application. However, by using the assembly stubs, many platforms can use an additional space optimization (see *below*).

The biggest hurdle to creating assembly stubs is handling the various ways that the dispatch table pointer can be accessed. There are four different methods that can be used:

1. Using `_glapi_Dispatch` directly in builds for non-multithreaded environments.
2. Using `_glapi_Dispatch` and `_glapi_get_dispatch` in multithreaded environments.
3. Using `_glapi_Dispatch` and `pthread_getspecific` in multithreaded environments.
4. Using `_glapi_tls_Dispatch` directly in TLS enabled multithreaded environments.

People wishing to implement assembly stubs for new platforms should focus on #4 if the new platform supports TLS. Otherwise, implement #2 followed by #3. Environments that do not support multithreading are uncommon and not terribly relevant.

Selection of the dispatch table pointer access method is controlled by a few preprocessor defines.

- If `USE_ELF_TLS` is defined, method #3 is used.
- If `HAVE_PTHREAD` is defined, method #2 is used.
- If none of the preceding are defined, method #1 is used.

Two different techniques are used to handle the various different cases. On x86 and SPARC, a macro called `GL_STUB` is used. In the preamble of the assembly source file different implementations of the macro are selected based on the defined preprocessor variables. The assembly code then consists of a series of invocations of the macros such as:

Listing 4: SPARC Assembly Implementation of glColor3fv

```
GL_STUB(Color3fv, _gloffset_Color3fv)
```

The benefit of this technique is that changes to the calling pattern (i.e., addition of a new dispatch table pointer access method) require fewer changed lines in the assembly code.

However, this technique can only be used on platforms where the function implementation does not change based on the parameters passed to the function. For example, since x86 passes all parameters on the stack, no additional code is needed to save and restore function parameters around a call to pthread_getspecific. Since x86-64 passes parameters in registers, varying amounts of code needs to be inserted around the call to pthread_getspecific to save and restore the GL function's parameters.

The other technique, used by platforms like x86-64 that cannot use the first technique, is to insert #ifdef within the assembly implementation of each function. This makes the assembly file considerably larger (e.g., 29,332 lines for glapi_x86-64.S versus 1,155 lines for glapi_x86.S) and causes simple changes to the function implementation to generate many lines of diffs. Since the assembly files are typically generated by scripts, this isn't a significant problem.

Once a new assembly file is created, it must be inserted in the build system. There are two steps to this. The file must first be added to src/mesa/sources. That gets the file built and linked. The second step is to add the correct #ifdef magic to src/mesa/glapi/glapi_dispatch.c to prevent the C version of the dispatch functions from being built.

33.3.4 3.4. Fixed-Length Dispatch Stubs

To implement glXGetProcAddress, Mesa stores a table that associates function names with pointers to those functions. This table is stored in src/mesa/glapi/glprocs.h. For different reasons on different platforms, storing all of those pointers is inefficient. On most platforms, including all known platforms that support TLS, we can avoid this added overhead.

If the assembly stubs are all the same size, the pointer need not be stored for every function. The location of the function can instead be calculated by multiplying the size of the dispatch stub by the offset of the function in the table. This value is then added to the address of the first dispatch stub.

This path is activated by adding the correct #ifdef magic to src/mesa/glapi/glapi.c just before glprocs.h is included.

CHAPTER 34

Gallium

Contents:

34.1 Introduction

34.1.1 What is Gallium?

Gallium is essentially an API for writing graphics drivers in a largely device-agnostic fashion. It provides several objects which encapsulate the core services of graphics hardware in a straightforward manner.

34.2 Debugging

Debugging utilities in gallium.

34.2.1 Debug Variables

All drivers respond to a set of common debug environment variables, as well as some driver-specific variables. Set them as normal environment variables for the platform or operating system you are running. For example, for Linux this can be done by typing "export var=value" into a console and then running the program from that console.

Common

GALLIUM_PRINT_OPTIONS Type: bool, **Default: false**

This option controls if the debug variables should be printed to stderr. This is probably the most useful variable, since it allows you to find which variables a driver uses.

GALLIUM_RBUG Type: bool, **Default: false**

Controls if the *Remote Debugger* should be used.

GALLIUM_TRACE Type: string, **Default: ""**

If set, this variable will cause the *Trace* output to be written to the specified file. Paths may be relative or absolute; relative paths are relative to the working directory. For example, setting it to "trace.xml" will cause the trace to be written to a file of the same name in the working directory.

GALLIUM_DUMP_CPU Type: bool, **Default: false**

Dump information about the current CPU that the driver is running on.

TGSI_PRINT_SANITY Type: bool, **Default: false**

Gallium has a built-in shader sanity checker. This option controls whether the shader sanity checker prints its warnings and errors to stderr.

DRAW_USE_LLVM Type: bool, **Default: false**

Whether the *Draw* module will attempt to use LLVM for vertex and geometry shaders.

GL State tracker-specific

ST_DEBUG Type: flags, **Default: 0x0**

Debug *Flags* for the GL state tracker.

Driver-specific

I915_DEBUG Type: flags, **Default: 0x0**

Debug *Flags* for the i915 driver.

I915_NO_HW Type: bool, **Default: false**

Stop the i915 driver from submitting commands to the hardware.

I915_DUMP_CMD Type: bool, **Default: false**

Dump all commands going to the hardware.

LP_DEBUG Type: flags, **Default: 0x0**

Debug *Flags* for the llvmpipe driver.

LP_NUM_THREADS Type: int, **Default: number of CPUs**

Number of threads that the llvmpipe driver should use.

FD_MESA_DEBUG Type: flags, **Default: 0x0**

Debug *Flags* for the freedreno driver.

Flags

The variables of type "flags" all take a string with comma-separated flags to enable different debugging for different parts of the drivers or state tracker. If set to "help", the driver will print a list of flags which the variable accepts. Order does not matter.

34.2.2 Remote Debugger

The remote debugger, commonly known as rbug, allows for runtime inspections of *Context*, *Screen*, *Resources and derived objects* and *Shader* objects; and pausing and stepping of *Draw* calls. Is used with rbug-gui which is hosted outside of the main mesa repository. rbug is can be used over a network connection, so the debugger does not need to be on the same machine.

34.3 TGSI

TGSI, Tungsten Graphics Shader Infrastructure, is an intermediate language for describing shaders. Since Gallium is inherently shaderful, shaders are an important part of the API. TGSI is the only intermediate representation used by all drivers.

34.3.1 Basics

All TGSI instructions, known as *opcodes*, operate on arbitrary-precision floating-point four-component vectors. An opcode may have up to one destination register, known as *dst*, and between zero and three source registers, called *src0* through *src2*, or simply *src* if there is only one.

Some instructions, like `I2F`, permit re-interpretation of vector components as integers. Other instructions permit using registers as two-component vectors with double precision; see *Double ISA*.

When an instruction has a scalar result, the result is usually copied into each of the components of *dst*. When this happens, the result is said to be *replicated* to *dst*. `RCP` is one such instruction.

Modifiers

TGSI supports modifiers on inputs (as well as saturate and precise modifier on instructions).

For arithmetic instruction having a precise modifier certain optimizations which may alter the result are disallowed. Example: *add(mul(a,b),c)* can't be optimized to TGSI_OPCODE_MAD, because some hardware only supports the fused MAD instruction.

For inputs which have a floating point type, both absolute value and negation modifiers are supported (with absolute value being applied first). The only source of TGSI_OPCODE_MOV and the second and third sources of TGSI_OPCODE_UCMP are considered to have float type for applying modifiers.

For inputs which have signed or unsigned type only the negate modifier is supported.

34.3.2 Instruction Set

Core ISA

These opcodes are guaranteed to be available regardless of the driver being used.

`ARL (Address Register Load)`

$$
\begin{aligned}
dst.x &= (int)\lfloor src.x \rfloor \\
dst.y &= (int)\lfloor src.y \rfloor \\
dst.z &= (int)\lfloor src.z \rfloor \\
dst.w &= (int)\lfloor src.w \rfloor
\end{aligned}
$$

MOV (Move)

$$
\begin{aligned}
dst.x &= src.x \\
dst.y &= src.y \\
dst.z &= src.z \\
dst.w &= src.w
\end{aligned}
$$

LIT (Light Coefficients)

$$
\begin{aligned}
dst.x &= 1 \\
dst.y &= max(src.x, 0) \\
dst.z &= (src.x > 0)?max(src.y, 0)^{clamp(src.w, -128, 128))} : 0 \\
dst.w &= 1
\end{aligned}
$$

RCP (Reciprocal)

This instruction replicates its result.

$$dst = \frac{1}{src.x}$$

RSQ (Reciprocal Square Root)

This instruction replicates its result. The results are undefined for src <= 0.

$$dst = \frac{1}{\sqrt{src.x}}$$

SQRT (Square Root)

This instruction replicates its result. The results are undefined for src < 0.

$$dst = \sqrt{src.x}$$

EXP (Approximate Exponential Base 2)

$$
\begin{aligned}
dst.x &= 2^{\lfloor src.x \rfloor} \\
dst.y &= src.x - \lfloor src.x \rfloor \\
dst.z &= 2^{src.x} \\
dst.w &= 1
\end{aligned}
$$

LOG (Approximate Logarithm Base 2)

$$
\begin{aligned}
dst.x &= \lfloor \log_2 |src.x| \rfloor \\
dst.y &= \frac{|src.x|}{2^{\lfloor \log_2 |src.x| \rfloor}} \\
dst.z &= \log_2 |src.x| \\
dst.w &= 1
\end{aligned}
$$

MUL (Multiply)

$$
\begin{aligned}
dst.x &= src0.x \times src1.x \\
dst.y &= src0.y \times src1.y \\
dst.z &= src0.z \times src1.z \\
dst.w &= src0.w \times src1.w
\end{aligned}
$$

ADD (Add)

$$
\begin{aligned}
dst.x &= src0.x + src1.x \\
dst.y &= src0.y + src1.y \\
dst.z &= src0.z + src1.z \\
dst.w &= src0.w + src1.w
\end{aligned}
$$

DP3 (3-component Dot Product)

This instruction replicates its result.

$$dst = src0.x \times src1.x + src0.y \times src1.y + src0.z \times src1.z$$

DP4 (4-component Dot Product)

This instruction replicates its result.

$$dst = src0.x \times src1.x + src0.y \times src1.y + src0.z \times src1.z + src0.w \times src1.w$$

DST (Distance Vector)

$$
\begin{aligned}
dst.x &= 1 \\
dst.y &= src0.y \times src1.y \\
dst.z &= src0.z \\
dst.w &= src1.w
\end{aligned}
$$

MIN (Minimum)

$$dst.x = min(src0.x, src1.x)$$
$$dst.y = min(src0.y, src1.y)$$
$$dst.z = min(src0.z, src1.z)$$
$$dst.w = min(src0.w, src1.w)$$

MAX (Maximum)

$$dst.x = max(src0.x, src1.x)$$
$$dst.y = max(src0.y, src1.y)$$
$$dst.z = max(src0.z, src1.z)$$
$$dst.w = max(src0.w, src1.w)$$

SLT (Set On Less Than)

$$dst.x = (src0.x < src1.x)?1.0F : 0.0F$$
$$dst.y = (src0.y < src1.y)?1.0F : 0.0F$$
$$dst.z = (src0.z < src1.z)?1.0F : 0.0F$$
$$dst.w = (src0.w < src1.w)?1.0F : 0.0F$$

SGE (Set On Greater Equal Than)

$$dst.x = (src0.x >= src1.x)?1.0F : 0.0F$$
$$dst.y = (src0.y >= src1.y)?1.0F : 0.0F$$
$$dst.z = (src0.z >= src1.z)?1.0F : 0.0F$$
$$dst.w = (src0.w >= src1.w)?1.0F : 0.0F$$

MAD (Multiply And Add)

Perform a * b + c. The implementation is free to decide whether there is an intermediate rounding step or not.

$$dst.x = src0.x \times src1.x + src2.x$$
$$dst.y = src0.y \times src1.y + src2.y$$
$$dst.z = src0.z \times src1.z + src2.z$$
$$dst.w = src0.w \times src1.w + src2.w$$

LRP (Linear Interpolate)

$$dst.x = src0.x \times src1.x + (1 - src0.x) \times src2.x$$
$$dst.y = src0.y \times src1.y + (1 - src0.y) \times src2.y$$
$$dst.z = src0.z \times src1.z + (1 - src0.z) \times src2.z$$
$$dst.w = src0.w \times src1.w + (1 - src0.w) \times src2.w$$

FMA (Fused Multiply-Add)

Perform a * b + c with no intermediate rounding step.

$$
\begin{aligned}
dst.x &= src0.x \times src1.x + src2.x \\
dst.y &= src0.y \times src1.y + src2.y \\
dst.z &= src0.z \times src1.z + src2.z \\
dst.w &= src0.w \times src1.w + src2.w
\end{aligned}
$$

FRC (Fraction)

$$
\begin{aligned}
dst.x &= src.x - \lfloor src.x \rfloor \\
dst.y &= src.y - \lfloor src.y \rfloor \\
dst.z &= src.z - \lfloor src.z \rfloor \\
dst.w &= src.w - \lfloor src.w \rfloor
\end{aligned}
$$

FLR (Floor)

$$
\begin{aligned}
dst.x &= \lfloor src.x \rfloor \\
dst.y &= \lfloor src.y \rfloor \\
dst.z &= \lfloor src.z \rfloor \\
dst.w &= \lfloor src.w \rfloor
\end{aligned}
$$

ROUND (Round)

$$
\begin{aligned}
dst.x &= round(src.x) \\
dst.y &= round(src.y) \\
dst.z &= round(src.z) \\
dst.w &= round(src.w)
\end{aligned}
$$

EX2 (Exponential Base 2)

This instruction replicates its result.

$$dst = 2^{src.x}$$

LG2 (Logarithm Base 2)

This instruction replicates its result.

$$dst = \log_2 src.x$$

POW (Power)

This instruction replicates its result.

$$dst = src0.x^{src1.x}$$

LDEXP (Multiply Number by Integral Power of 2)

src1 is an integer.

$$dst.x = src0.x * 2^{src1.x} dst.y = src0.y * 2^{src1.y} dst.z = src0.z * 2^{src1.z} dst.w = src0.w * 2^{src1.w}$$

COS (Cosine)

This instruction replicates its result.

$$dst = \cos src.x$$

DDX, DDX_FINE (Derivative Relative To X)

The fine variant is only used when `PIPE_CAP_TGSI_FS_FINE_DERIVATIVE` is advertised. When it is, the fine version guarantees one derivative per row while DDX is allowed to be the same for the entire 2x2 quad.

$$\begin{aligned} dst.x &= partialx(src.x) \\ dst.y &= partialx(src.y) \\ dst.z &= partialx(src.z) \\ dst.w &= partialx(src.w) \end{aligned}$$

DDY, DDY_FINE (Derivative Relative To Y)

The fine variant is only used when `PIPE_CAP_TGSI_FS_FINE_DERIVATIVE` is advertised. When it is, the fine version guarantees one derivative per column while DDY is allowed to be the same for the entire 2x2 quad.

$$\begin{aligned} dst.x &= partialy(src.x) \\ dst.y &= partialy(src.y) \\ dst.z &= partialy(src.z) \\ dst.w &= partialy(src.w) \end{aligned}$$

PK2H (Pack Two 16-bit Floats)

This instruction replicates its result.

$$dst = f32\_to\_f16(src.x)|f32\_to\_f16(src.y) << 16$$

PK2US (Pack Two Unsigned 16-bit Scalars)

This instruction replicates its result.

$$dst = f32\_to\_unorm16(src.x)|f32\_to\_unorm16(src.y) << 16$$

PK4B (Pack Four Signed 8-bit Scalars)

This instruction replicates its result.

$dst = f32\_to\_snorm8(src.x)|(f32\_to\_snorm8(src.y) << 8)|(f32\_to\_snorm8(src.z) << 16)|(f32\_to\_snorm8(src.w) << 24)$

PK4UB (Pack Four Unsigned 8-bit Scalars)

This instruction replicates its result.

$dst = f32\_to\_unorm8(src.x)|(f32\_to\_unorm8(src.y) << 8)|(f32\_to\_unorm8(src.z) << 16)|(f32\_to\_unorm8(src.w) << 24$

SEQ (Set On Equal)

$$dst.x = (src0.x == src1.x)?1.0F : 0.0F$$
$$dst.y = (src0.y == src1.y)?1.0F : 0.0F$$
$$dst.z = (src0.z == src1.z)?1.0F : 0.0F$$
$$dst.w = (src0.w == src1.w)?1.0F : 0.0F$$

SGT (Set On Greater Than)

$$dst.x = (src0.x > src1.x)?1.0F : 0.0F$$
$$dst.y = (src0.y > src1.y)?1.0F : 0.0F$$
$$dst.z = (src0.z > src1.z)?1.0F : 0.0F$$
$$dst.w = (src0.w > src1.w)?1.0F : 0.0F$$

SIN (Sine)

This instruction replicates its result.

$$dst = \sin src.x$$

SLE (Set On Less Equal Than)

$$dst.x = (src0.x <= src1.x)?1.0F : 0.0F$$
$$dst.y = (src0.y <= src1.y)?1.0F : 0.0F$$
$$dst.z = (src0.z <= src1.z)?1.0F : 0.0F$$
$$dst.w = (src0.w <= src1.w)?1.0F : 0.0F$$

SNE (Set On Not Equal)

$$dst.x = (src0.x! = src1.x)?1.0F : 0.0F$$
$$dst.y = (src0.y! = src1.y)?1.0F : 0.0F$$
$$dst.z = (src0.z! = src1.z)?1.0F : 0.0F$$
$$dst.w = (src0.w! = src1.w)?1.0F : 0.0F$$

TEX (Texture Lookup)

for array textures src0.y contains the slice for 1D, and src0.z contain the slice for 2D.

for shadow textures with no arrays (and not cube map), src0.z contains the reference value.

for shadow textures with arrays, src0.z contains the reference value for 1D arrays, and src0.w contains the reference value for 2D arrays and cube maps.

for cube map array shadow textures, the reference value cannot be passed in src0.w, and TEX2 must be used instead.

$$
\begin{aligned}
coord &= src0 \\
shadow_ref &= src0.zorsrc0.w(optional) \\
unit &= src1 \\
dst &= texture\_sample(unit, coord, shadow_ref)
\end{aligned}
$$

TEX2 (Texture Lookup (for shadow cube map arrays only))

this is the same as TEX, but uses another reg to encode the reference value.

$$
\begin{aligned}
coord &= src0 \\
shadow_ref &= src1.x \\
unit &= src2 \\
dst &= texture\_sample(unit, coord, shadow_ref)
\end{aligned}
$$

TXD (Texture Lookup with Derivatives)

$$
\begin{aligned}
coord &= src0 \\
ddx &= src1 \\
ddy &= src2 \\
unit &= src3 \\
dst &= texture\_sample\_deriv(unit, coord, ddx, ddy)
\end{aligned}
$$

TXP (Projective Texture Lookup)

$$
\begin{aligned}
coord.x &= src0.x/src0.w \\
coord.y &= src0.y/src0.w \\
coord.z &= src0.z/src0.w \\
coord.w &= src0.w \\
unit &= src1 \\
dst &= texture\_sample(unit, coord)
\end{aligned}
$$

UP2H (Unpack Two 16-Bit Floats)

$$dst.x = f16\_to\_f32(src0.x\&0xffff)$$
$$dst.y = f16\_to\_f32(src0.x >> 16)$$
$$dst.z = f16\_to\_f32(src0.x\&0xffff)$$
$$dst.w = f16\_to\_f32(src0.x >> 16)$$

Note: Considered for removal.

UP2US (Unpack Two Unsigned 16-Bit Scalars)

TBD

Note: Considered for removal.

UP4B (Unpack Four Signed 8-Bit Values)

TBD

Note: Considered for removal.

UP4UB (Unpack Four Unsigned 8-Bit Scalars)

TBD

Note: Considered for removal.

ARR (Address Register Load With Round)

$$dst.x = (int)round(src.x)$$
$$dst.y = (int)round(src.y)$$
$$dst.z = (int)round(src.z)$$
$$dst.w = (int)round(src.w)$$

SSG (Set Sign)

$$dst.x = (src.x > 0)?1 : (src.x < 0)? - 1 : 0$$
$$dst.y = (src.y > 0)?1 : (src.y < 0)? - 1 : 0$$
$$dst.z = (src.z > 0)?1 : (src.z < 0)? - 1 : 0$$
$$dst.w = (src.w > 0)?1 : (src.w < 0)? - 1 : 0$$

CMP (Compare)

$$dst.x = (src0.x < 0)?src1.x : src2.x$$
$$dst.y = (src0.y < 0)?src1.y : src2.y$$
$$dst.z = (src0.z < 0)?src1.z : src2.z$$
$$dst.w = (src0.w < 0)?src1.w : src2.w$$

KILL_IF (Conditional Discard)

Conditional discard. Allowed in fragment shaders only.

$$if(src.x < 0||src.y < 0||src.z < 0||src.w < 0)discardendif$$

KILL (Discard)

Unconditional discard. Allowed in fragment shaders only.

DEMOTE (Demote Invocation to a Helper)

This demotes the current invocation to a helper, but continues execution (while KILL may or may not terminate the invocation). After this runs, all the usual helper invocation rules apply about discarding buffer and render target writes. This is useful for having accurate derivatives in the other invocations which have not been demoted.

Allowed in fragment shaders only.

READ_HELPER (Reads Invocation Helper Status)

This is identical to `TGSI_SEMANTIC_HELPER_INVOCATION`, except this will read the current value, which might change as a result of a `DEMOTE` instruction.

Allowed in fragment shaders only.

TXB (Texture Lookup With Bias)

for cube map array textures and shadow cube maps, the bias value cannot be passed in src0.w, and TXB2 must be used instead.

if the target is a shadow texture, the reference value is always in src.z (this prevents shadow 3d and shadow 2d arrays from using this instruction, but this is not needed).

$$\begin{aligned} coord.x &= src0.x \\ coord.y &= src0.y \\ coord.z &= src0.z \\ coord.w &= none \\ bias &= src0.w \\ unit &= src1 \\ dst &= texture\_sample(unit, coord, bias) \end{aligned}$$

TXB2 (Texture Lookup With Bias (some cube maps only))

this is the same as TXB, but uses another reg to encode the lod bias value for cube map arrays and shadow cube maps. Presumably shadow 2d arrays and shadow 3d targets could use this encoding too, but this is not legal.

if the target is a shadow cube map array, the reference value is in src1.y.

$$\begin{aligned} coord &= src0 \\ bias &= src1.x \\ unit &= src2 \\ dst &= texture\_sample(unit, coord, bias) \end{aligned}$$

DIV (Divide)

$$dst.x = \frac{src0.x}{src1.x}$$
$$dst.y = \frac{src0.y}{src1.y}$$
$$dst.z = \frac{src0.z}{src1.z}$$
$$dst.w = \frac{src0.w}{src1.w}$$

DP2 (2-component Dot Product)

This instruction replicates its result.

$$dst = src0.x \times src1.x + src0.y \times src1.y$$

TEX_LZ (Texture Lookup With LOD = 0)

This is the same as TXL with LOD = 0. Like every texture opcode, it obeys pipe_sampler_view::u.tex.first_level and pipe_sampler_state::min_lod. There is no way to override those two in shaders.

$$\begin{aligned} coord.x &= src0.x \\ coord.y &= src0.y \\ coord.z &= src0.z \\ coord.w &= none \\ lod &= 0 \\ unit &= src1 \\ dst &= texture\_sample(unit, coord, lod) \end{aligned}$$

TXL (Texture Lookup With explicit LOD)

for cube map array textures, the explicit lod value cannot be passed in src0.w, and TXL2 must be used instead.

if the target is a shadow texture, the reference value is always in src.z (this prevents shadow 3d / 2d array / cube targets from using this instruction, but this is not needed).

$$\begin{aligned} coord.x &= src0.x \\ coord.y &= src0.y \\ coord.z &= src0.z \\ coord.w &= none \\ lod &= src0.w \\ unit &= src1 \\ dst &= texture\_sample(unit, coord, lod) \end{aligned}$$

TXL2 (Texture Lookup With explicit LOD (for cube map arrays only))

this is the same as TXL, but uses another reg to encode the explicit lod value. Presumably shadow 3d / 2d array / cube targets could use this encoding too, but this is not legal.

if the target is a shadow cube map array, the reference value is in src1.y.

$$
\begin{aligned}
coord &= src0 \\
lod &= src1.x \\
unit &= src2 \\
dst &= texture\_sample(unit, coord, lod)
\end{aligned}
$$

Compute ISA

These opcodes are primarily provided for special-use computational shaders. Support for these opcodes indicated by a special pipe capability bit (TBD).

XXX doesn't look like most of the opcodes really belong here.

CEIL (Ceiling)

$$
\begin{aligned}
dst.x &= \lceil src.x \rceil \\
dst.y &= \lceil src.y \rceil \\
dst.z &= \lceil src.z \rceil \\
dst.w &= \lceil src.w \rceil
\end{aligned}
$$

TRUNC (Truncate)

$$
\begin{aligned}
dst.x &= trunc(src.x) \\
dst.y &= trunc(src.y) \\
dst.z &= trunc(src.z) \\
dst.w &= trunc(src.w)
\end{aligned}
$$

MOD (Modulus)

$$
\begin{aligned}
dst.x &= src0.x \bmod src1.x \\
dst.y &= src0.y \bmod src1.y \\
dst.z &= src0.z \bmod src1.z \\
dst.w &= src0.w \bmod src1.w
\end{aligned}
$$

UARL (Integer Address Register Load)

Moves the contents of the source register, assumed to be an integer, into the destination register, which is assumed to be an address (ADDR) register.

TXF (Texel Fetch)

As per NV_gpu_shader4, extract a single texel from a specified texture image or PIPE_BUFFER resource. The source sampler may not be a CUBE or SHADOW. src 0 is a four-component signed integer vector used to identify the single texel accessed. 3 components + level. If the texture is multisampled, then the fourth component indicates the sample, not the mipmap level. Just like texture instructions, an optional offset vector is provided, which is subject to various driver restrictions (regarding range, source of offsets). This instruction ignores the sampler state.

TXF(uint_vec coord, int_vec offset).

TXQ (Texture Size Query)

As per NV_gpu_program4, retrieve the dimensions of the texture depending on the target. For 1D (width), 2D/RECT/CUBE (width, height), 3D (width, height, depth), 1D array (width, layers), 2D array (width, height, layers). Also return the number of accessible levels (last_level - first_level + 1) in W.

For components which don't return a resource dimension, their value is undefined.

$$
\begin{aligned}
lod &= src0.x \\
dst.x &= texture\_width(unit, lod) \\
dst.y &= texture\_height(unit, lod) \\
dst.z &= texture\_depth(unit, lod) \\
dst.w &= texture\_levels(unit)
\end{aligned}
$$

TXQS (Texture Samples Query)

This retrieves the number of samples in the texture, and stores it into the x component as an unsigned integer. The other components are undefined. If the texture is not multisampled, this function returns (1, undef, undef, undef).

$$dst.x = texture\_samples(unit)$$

TG4 (Texture Gather)

As per ARB_texture_gather, gathers the four texels to be used in a bi-linear filtering operation and packs them into a single register. Only works with 2D, 2D array, cubemaps, and cubemaps arrays. For 2D textures, only the addressing modes of the sampler and the top level of any mip pyramid are used. Set W to zero. It behaves like the TEX instruction, but a filtered sample is not generated. The four samples that contribute to filtering are placed into xyzw in clockwise order, starting with the (u,v) texture coordinate delta at the following locations (-, +), (+, +), (+, -), (-, -), where the magnitude of the deltas are half a texel.

PIPE_CAP_TEXTURE_SM5 enhances this instruction to support shadow per-sample depth compares, single component selection, and a non-constant offset. It doesn't allow support for the GL independent offset to get i0,j0. This would require another CAP is hw can do it natively. For now we lower that before TGSI.

PIPE_CAP_TGSI_TG4_COMPONENT_IN_SWIZZLE changes the encoding so that component is stored in the sampler source swizzle x.

$$
\begin{aligned}
coord &= src0 \\
(without TGSI_T G4_C OMPONENT_I N_S WIZZLE) component &= src1 \\
dst &= texture\_gather4(unit, coord, component) \\
(with TGSI_T G4_C OMPONENT_I N_S WIZZLE) dst &= texture\_gather4(unit, coord) component is encoded in sampler swizzle.
\end{aligned}
$$

(with SM5 - cube array shadow)

$$
\begin{aligned}
coord &= src0 \\
compare &= src1 \\
dst &= texture\_gather(uint, coord, compare)
\end{aligned}
$$

LODQ (level of detail query)

Compute the LOD information that the texture pipe would use to access the texture. The Y component contains the computed LOD lambda_prime. The X component contains the LOD that will be accessed, based on min/max lod's and mipmap filters.

$$coord = src0$$
$$dst.xy = lodq(uint, coord);$$

CLOCK (retrieve the current shader time)

Invoking this instruction multiple times in the same shader should cause monotonically increasing values to be returned. The values are implicitly 64-bit, so if fewer than 64 bits of precision are available, to provide expected wraparound semantics, the value should be shifted up so that the most significant bit of the time is the most significant bit of the 64-bit value.

$$dst.xy = clock()$$

Integer ISA

These opcodes are used for integer operations. Support for these opcodes indicated by PIPE_SHADER_CAP_INTEGERS (all of them?)

I2F (Signed Integer To Float)

Rounding is unspecified (round to nearest even suggested).

$$dst.x = (float)src.x$$
$$dst.y = (float)src.y$$
$$dst.z = (float)src.z$$
$$dst.w = (float)src.w$$

U2F (Unsigned Integer To Float)

Rounding is unspecified (round to nearest even suggested).

$$dst.x = (float)src.x$$
$$dst.y = (float)src.y$$
$$dst.z = (float)src.z$$
$$dst.w = (float)src.w$$

F2I (Float to Signed Integer)

Rounding is towards zero (truncate). Values outside signed range (including NaNs) produce undefined results.

$$dst.x = (int)src.x$$
$$dst.y = (int)src.y$$
$$dst.z = (int)src.z$$
$$dst.w = (int)src.w$$

F2U (Float to Unsigned Integer)

Rounding is towards zero (truncate). Values outside unsigned range (including NaNs) produce undefined results.

$$dst.x = (unsigned)src.x$$
$$dst.y = (unsigned)src.y$$
$$dst.z = (unsigned)src.z$$
$$dst.w = (unsigned)src.w$$

UADD (Integer Add)

This instruction works the same for signed and unsigned integers. The low 32bit of the result is returned.

$$dst.x = src0.x + src1.x$$
$$dst.y = src0.y + src1.y$$
$$dst.z = src0.z + src1.z$$
$$dst.w = src0.w + src1.w$$

UMAD (Integer Multiply And Add)

This instruction works the same for signed and unsigned integers. The multiplication returns the low 32bit (as does the result itself).

$$dst.x = src0.x \times src1.x + src2.x$$
$$dst.y = src0.y \times src1.y + src2.y$$
$$dst.z = src0.z \times src1.z + src2.z$$
$$dst.w = src0.w \times src1.w + src2.w$$

UMUL (Integer Multiply)

This instruction works the same for signed and unsigned integers. The low 32bit of the result is returned.

$$dst.x = src0.x \times src1.x$$
$$dst.y = src0.y \times src1.y$$
$$dst.z = src0.z \times src1.z$$
$$dst.w = src0.w \times src1.w$$

IMUL_HI (Signed Integer Multiply High Bits)

The high 32bits of the multiplication of 2 signed integers are returned.

$$dst.x = (src0.x \times src1.x) >> 32$$
$$dst.y = (src0.y \times src1.y) >> 32$$
$$dst.z = (src0.z \times src1.z) >> 32$$
$$dst.w = (src0.w \times src1.w) >> 32$$

UMUL_HI (Unsigned Integer Multiply High Bits)

The high 32bits of the multiplication of 2 unsigned integers are returned.

$$
\begin{aligned}
dst.x &= (src0.x \times src1.x) >> 32 \\
dst.y &= (src0.y \times src1.y) >> 32 \\
dst.z &= (src0.z \times src1.z) >> 32 \\
dst.w &= (src0.w \times src1.w) >> 32
\end{aligned}
$$

IDIV (Signed Integer Division)

TBD: behavior for division by zero.

$$
\begin{aligned}
dst.x &= \frac{src0.x}{src1.x} \\
dst.y &= \frac{src0.y}{src1.y} \\
dst.z &= \frac{src0.z}{src1.z} \\
dst.w &= \frac{src0.w}{src1.w}
\end{aligned}
$$

UDIV (Unsigned Integer Division)

For division by zero, 0xffffffff is returned.

$$
\begin{aligned}
dst.x &= \frac{src0.x}{src1.x} \\
dst.y &= \frac{src0.y}{src1.y} \\
dst.z &= \frac{src0.z}{src1.z} \\
dst.w &= \frac{src0.w}{src1.w}
\end{aligned}
$$

UMOD (Unsigned Integer Remainder)

If second arg is zero, 0xffffffff is returned.

$$
\begin{aligned}
dst.x &= src0.x \bmod src1.x \\
dst.y &= src0.y \bmod src1.y \\
dst.z &= src0.z \bmod src1.z \\
dst.w &= src0.w \bmod src1.w
\end{aligned}
$$

NOT (Bitwise Not)

$$
\begin{aligned}
dst.x &=\sim src.x \\
dst.y &=\sim src.y \\
dst.z &=\sim src.z \\
dst.w &=\sim src.w
\end{aligned}
$$

AND (Bitwise And)

$$dst.x = src0.x \& src1.x$$
$$dst.y = src0.y \& src1.y$$
$$dst.z = src0.z \& src1.z$$
$$dst.w = src0.w \& src1.w$$

OR (Bitwise Or)

$$dst.x = src0.x | src1.x$$
$$dst.y = src0.y | src1.y$$
$$dst.z = src0.z | src1.z$$
$$dst.w = src0.w | src1.w$$

XOR (Bitwise Xor)

$$dst.x = src0.x \oplus src1.x$$
$$dst.y = src0.y \oplus src1.y$$
$$dst.z = src0.z \oplus src1.z$$
$$dst.w = src0.w \oplus src1.w$$

IMAX (Maximum of Signed Integers)

$$dst.x = max(src0.x, src1.x)$$
$$dst.y = max(src0.y, src1.y)$$
$$dst.z = max(src0.z, src1.z)$$
$$dst.w = max(src0.w, src1.w)$$

UMAX (Maximum of Unsigned Integers)

$$dst.x = max(src0.x, src1.x)$$
$$dst.y = max(src0.y, src1.y)$$
$$dst.z = max(src0.z, src1.z)$$
$$dst.w = max(src0.w, src1.w)$$

IMIN (Minimum of Signed Integers)

$$dst.x = min(src0.x, src1.x)$$
$$dst.y = min(src0.y, src1.y)$$
$$dst.z = min(src0.z, src1.z)$$
$$dst.w = min(src0.w, src1.w)$$

UMIN (Minimum of Unsigned Integers)

$$dst.x = min(src0.x, src1.x)$$
$$dst.y = min(src0.y, src1.y)$$
$$dst.z = min(src0.z, src1.z)$$
$$dst.w = min(src0.w, src1.w)$$

SHL (Shift Left)

The shift count is masked with 0x1f before the shift is applied.

$$dst.x = src0.x << (0x1f \& src1.x)$$
$$dst.y = src0.y << (0x1f \& src1.y)$$
$$dst.z = src0.z << (0x1f \& src1.z)$$
$$dst.w = src0.w << (0x1f \& src1.w)$$

ISHR (Arithmetic Shift Right (of Signed Integer))

The shift count is masked with 0x1f before the shift is applied.

$$dst.x = src0.x >> (0x1f \& src1.x)$$
$$dst.y = src0.y >> (0x1f \& src1.y)$$
$$dst.z = src0.z >> (0x1f \& src1.z)$$
$$dst.w = src0.w >> (0x1f \& src1.w)$$

USHR (Logical Shift Right)

The shift count is masked with 0x1f before the shift is applied.

$$dst.x = src0.x >> (unsigned)(0x1f \& src1.x)$$
$$dst.y = src0.y >> (unsigned)(0x1f \& src1.y)$$
$$dst.z = src0.z >> (unsigned)(0x1f \& src1.z)$$
$$dst.w = src0.w >> (unsigned)(0x1f \& src1.w)$$

UCMP (Integer Conditional Move)

$$dst.x = src0.x ? src1.x : src2.x$$
$$dst.y = src0.y ? src1.y : src2.y$$
$$dst.z = src0.z ? src1.z : src2.z$$
$$dst.w = src0.w ? src1.w : src2.w$$

ISSG (Integer Set Sign)

$$dst.x = (src0.x < 0)? - 1 : (src0.x > 0)?1 : 0$$
$$dst.y = (src0.y < 0)? - 1 : (src0.y > 0)?1 : 0$$
$$dst.z = (src0.z < 0)? - 1 : (src0.z > 0)?1 : 0$$
$$dst.w = (src0.w < 0)? - 1 : (src0.w > 0)?1 : 0$$

FSLT (Float Set On Less Than (ordered))

Same comparison as SLT but returns integer instead of 1.0/0.0 float

$$dst.x = (src0.x < src1.x)? \sim 0 : 0$$
$$dst.y = (src0.y < src1.y)? \sim 0 : 0$$
$$dst.z = (src0.z < src1.z)? \sim 0 : 0$$
$$dst.w = (src0.w < src1.w)? \sim 0 : 0$$

ISLT (Signed Integer Set On Less Than)

$$dst.x = (src0.x < src1.x)? \sim 0 : 0$$
$$dst.y = (src0.y < src1.y)? \sim 0 : 0$$
$$dst.z = (src0.z < src1.z)? \sim 0 : 0$$
$$dst.w = (src0.w < src1.w)? \sim 0 : 0$$

USLT (Unsigned Integer Set On Less Than)

$$dst.x = (src0.x < src1.x)? \sim 0 : 0$$
$$dst.y = (src0.y < src1.y)? \sim 0 : 0$$
$$dst.z = (src0.z < src1.z)? \sim 0 : 0$$
$$dst.w = (src0.w < src1.w)? \sim 0 : 0$$

FSGE (Float Set On Greater Equal Than (ordered))

Same comparison as SGE but returns integer instead of 1.0/0.0 float

$$dst.x = (src0.x >= src1.x)? \sim 0 : 0$$
$$dst.y = (src0.y >= src1.y)? \sim 0 : 0$$
$$dst.z = (src0.z >= src1.z)? \sim 0 : 0$$
$$dst.w = (src0.w >= src1.w)? \sim 0 : 0$$

ISGE (Signed Integer Set On Greater Equal Than)

$$dst.x = (src0.x >= src1.x)? \sim 0 : 0$$
$$dst.y = (src0.y >= src1.y)? \sim 0 : 0$$
$$dst.z = (src0.z >= src1.z)? \sim 0 : 0$$
$$dst.w = (src0.w >= src1.w)? \sim 0 : 0$$

USGE (Unsigned Integer Set On Greater Equal Than)

$$dst.x = (src0.x >= src1.x)? \sim 0 : 0$$
$$dst.y = (src0.y >= src1.y)? \sim 0 : 0$$
$$dst.z = (src0.z >= src1.z)? \sim 0 : 0$$
$$dst.w = (src0.w >= src1.w)? \sim 0 : 0$$

FSEQ (Float Set On Equal (ordered))

Same comparison as SEQ but returns integer instead of 1.0/0.0 float

$$dst.x = (src0.x == src1.x)? \sim 0 : 0$$
$$dst.y = (src0.y == src1.y)? \sim 0 : 0$$
$$dst.z = (src0.z == src1.z)? \sim 0 : 0$$
$$dst.w = (src0.w == src1.w)? \sim 0 : 0$$

USEQ (Integer Set On Equal)

$$dst.x = (src0.x == src1.x)? \sim 0 : 0$$
$$dst.y = (src0.y == src1.y)? \sim 0 : 0$$
$$dst.z = (src0.z == src1.z)? \sim 0 : 0$$
$$dst.w = (src0.w == src1.w)? \sim 0 : 0$$

FSNE (Float Set On Not Equal (unordered))

Same comparison as SNE but returns integer instead of 1.0/0.0 float

$$dst.x = (src0.x! = src1.x)? \sim 0 : 0$$
$$dst.y = (src0.y! = src1.y)? \sim 0 : 0$$
$$dst.z = (src0.z! = src1.z)? \sim 0 : 0$$
$$dst.w = (src0.w! = src1.w)? \sim 0 : 0$$

USNE (Integer Set On Not Equal)

$$dst.x = (src0.x! = src1.x)? \sim 0 : 0$$
$$dst.y = (src0.y! = src1.y)? \sim 0 : 0$$
$$dst.z = (src0.z! = src1.z)? \sim 0 : 0$$
$$dst.w = (src0.w! = src1.w)? \sim 0 : 0$$

INEG (Integer Negate)

Two's complement.

$$dst.x = -src.x$$
$$dst.y = -src.y$$
$$dst.z = -src.z$$
$$dst.w = -src.w$$

IABS (Integer Absolute Value)

$$dst.x = |src.x|$$
$$dst.y = |src.y|$$
$$dst.z = |src.z|$$
$$dst.w = |src.w|$$

Bitwise ISA

These opcodes are used for bit-level manipulation of integers.

IBFE (Signed Bitfield Extract)
Like GLSL bitfieldExtract. Extracts a set of bits from the input, and sign-extends them if the high bit of the extracted window is set.

Pseudocode:

```
def ibfe(value, offset, bits):
  if offset < 0 or bits < 0 or offset + bits > 32:
    return undefined
  if bits == 0: return 0
  # Note: >> sign-extends
  return (value << (32 - offset - bits)) >> (32 - bits)
```

UBFE (Unsigned Bitfield Extract)
Like GLSL bitfieldExtract. Extracts a set of bits from the input, without any sign-extension.

Pseudocode:

```
def ubfe(value, offset, bits):
  if offset < 0 or bits < 0 or offset + bits > 32:
    return undefined
  if bits == 0: return 0
  # Note: >> does not sign-extend
  return (value << (32 - offset - bits)) >> (32 - bits)
```

BFI (Bitfield Insert)
Like GLSL bitfieldInsert. Replaces a bit region of 'base' with the low bits of 'insert'.

Pseudocode:

```
def bfi(base, insert, offset, bits):
  if offset < 0 or bits < 0 or offset + bits > 32:
    return undefined
  # << defined such that mask == ~0 when bits == 32, offset == 0
  mask = ((1 << bits) - 1) << offset
  return ((insert << offset) & mask) | (base & ~mask)
```

BREV (Bitfield Reverse)
See SM5 instruction BFREV. Reverses the bits of the argument.

POPC (Population Count)
See SM5 instruction COUNTBITS. Counts the number of set bits in the argument.

LSB (Index of lowest set bit)
See SM5 instruction FIRSTBIT_LO. Computes the 0-based index of the first set bit of the argument. Returns -1 if none are set.

IMSB (Index of highest non-sign bit)
See SM5 instruction FIRSTBIT_SHI. Computes the 0-based index of the highest non-sign bit of the argument (i.e. highest 0 bit for negative numbers, highest 1 bit for positive numbers). Returns -1 if all bits are the same (i.e. for inputs 0 and -1).

UMSB (Index of highest set bit)
See SM5 instruction FIRSTBIT_HI. Computes the 0-based index of the highest set bit of the argument. Returns -1 if none are set.

Geometry ISA

These opcodes are only supported in geometry shaders; they have no meaning in any other type of shader.

EMIT (Emit)
Generate a new vertex for the current primitive into the specified vertex stream using the values in the output registers.

ENDPRIM (End Primitive)
Complete the current primitive in the specified vertex stream (consisting of the emitted vertices), and start a new one.

GLSL ISA

These opcodes are part of *GLSL*'s opcode set. Support for these opcodes is determined by a special capability bit, `GLSL`. Some require glsl version 1.30 (UIF/SWITCH/CASE/DEFAULT/ENDSWITCH).

CAL (Subroutine Call)
push(pc) pc = target

RET (Subroutine Call Return)
pc = pop()

CONT (Continue)
Unconditionally moves the point of execution to the instruction after the last bgnloop. The instruction must appear within a bgnloop/endloop.

Note: Support for CONT is determined by a special capability bit, `TGSI_CONT_SUPPORTED`. See *Screen* for more information.

BGNLOOP (Begin a Loop)
Start a loop. Must have a matching endloop.

BGNSUB (Begin Subroutine)
Starts definition of a subroutine. Must have a matching endsub.

ENDLOOP (End a Loop)
End a loop started with bgnloop.

ENDSUB (End Subroutine)
Ends definition of a subroutine.

NOP (No Operation)
Do nothing.

BRK (Break)
Unconditionally moves the point of execution to the instruction after the next endloop or endswitch. The instruction must appear within a loop/endloop or switch/endswitch.

IF (Float If)
Start an IF . . . ELSE .. ENDIF block. Condition evaluates to true if

src0.x != 0.0

where src0.x is interpreted as a floating point register.

UIF (Bitwise If)
Start an UIF . . . ELSE .. ENDIF block. Condition evaluates to true if

src0.x != 0

where src0.x is interpreted as an integer register.

ELSE (Else)
Starts an else block, after an IF or UIF statement.

ENDIF (End If)
Ends an IF or UIF block.

SWITCH (Switch)
Starts a C-style switch expression. The switch consists of one or multiple CASE statements, and at most one DEFAULT statement. Execution of a statement ends when a BRK is hit, but just like in C falling through to other cases without a break is allowed. Similarly, DEFAULT label is allowed anywhere not just as last statement, and fallthrough is allowed into/from it. CASE src arguments are evaluated at bit level against the SWITCH src argument.

Example:

```
SWITCH src[0].x
CASE src[0].x
(some instructions here)
(optional BRK here)
DEFAULT
(some instructions here)
(optional BRK here)
CASE src[0].x
(some instructions here)
(optional BRK here)
ENDSWITCH
```

CASE (Switch case)
This represents a switch case label. The src arg must be an integer immediate.

DEFAULT (Switch default)
This represents the default case in the switch, which is taken if no other case matches.

ENDSWITCH (End of switch)
Ends a switch expression.

Interpolation ISA

The interpolation instructions allow an input to be interpolated in a different way than its declaration. This corresponds to the GLSL 4.00 interpolateAt* functions. The first argument of each of these must come from `TGSI_FILE_INPUT`.

`INTERP_CENTROID (Interpolate at the centroid)`
Interpolates the varying specified by src0 at the centroid

`INTERP_SAMPLE (Interpolate at the specified sample)`
Interpolates the varying specified by src0 at the sample id specified by src1.x (interpreted as an integer)

`INTERP_OFFSET (Interpolate at the specified offset)`
Interpolates the varying specified by src0 at the offset src1.xy from the pixel center (interpreted as floats)

Double ISA

The double-precision opcodes reinterpret four-component vectors into two-component vectors with doubled precision in each component.

`DABS (Absolute)`

$$dst.xy = |src0.xy|$$
$$dst.zw = |src0.zw|$$

`DADD (Add)`

$$dst.xy = src0.xy + src1.xy$$
$$dst.zw = src0.zw + src1.zw$$

`DSEQ (Set on Equal)`

$$dst.x = src0.xy == src1.xy? \sim 0 : 0$$
$$dst.z = src0.zw == src1.zw? \sim 0 : 0$$

`DSNE (Set on Not Equal)`

$$dst.x = src0.xy! = src1.xy? \sim 0 : 0$$
$$dst.z = src0.zw! = src1.zw? \sim 0 : 0$$

`DSLT (Set on Less than)`

$$dst.x = src0.xy < src1.xy? \sim 0 : 0$$
$$dst.z = src0.zw < src1.zw? \sim 0 : 0$$

DSGE (Set on Greater equal)

$$dst.x = src0.xy >= src1.xy? \sim 0 : 0$$
$$dst.z = src0.zw >= src1.zw? \sim 0 : 0$$

DFRAC (Fraction)

$$dst.xy = src.xy - \lfloor src.xy \rfloor$$
$$dst.zw = src.zw - \lfloor src.zw \rfloor$$

DTRUNC (Truncate)

$$dst.xy = trunc(src.xy)$$
$$dst.zw = trunc(src.zw)$$

DCEIL (Ceiling)

$$dst.xy = \lceil src.xy \rceil$$
$$dst.zw = \lceil src.zw \rceil$$

DFLR (Floor)

$$dst.xy = \lfloor src.xy \rfloor$$
$$dst.zw = \lfloor src.zw \rfloor$$

DROUND (Fraction)

$$dst.xy = round(src.xy)$$
$$dst.zw = round(src.zw)$$

DSSG (Set Sign)

$$dst.xy = (src.xy > 0)?1.0 : (src.xy < 0)? - 1.0 : 0.0$$
$$dst.zw = (src.zw > 0)?1.0 : (src.zw < 0)? - 1.0 : 0.0$$

DFRACEXP (Convert Number to Fractional and Integral Components)

Like the `frexp()` routine in many math libraries, this opcode stores the exponent of its source to `dst0`, and the significand to `dst1`, such that $dst1 \times 2^{dst0} = src$. The results are replicated across channels.

$$dst0.xy = dst.zw = frac(src.xy)$$
$$dst1 = frac(src.xy)$$

DLDEXP (Multiply Number by Integral Power of 2)

This opcode is the inverse of *DFRACEXP*. The second source is an integer.

$$dst.xy = src0.xy \times 2^{src1.x}$$
$$dst.zw = src0.zw \times 2^{src1.z}$$

DMIN (Minimum)

$$dst.xy = min(src0.xy, src1.xy)$$
$$dst.zw = min(src0.zw, src1.zw)$$

DMAX (Maximum)

$$dst.xy = max(src0.xy, src1.xy)$$
$$dst.zw = max(src0.zw, src1.zw)$$

DMUL (Multiply)

$$dst.xy = src0.xy \times src1.xy$$
$$dst.zw = src0.zw \times src1.zw$$

DMAD (Multiply And Add)

$$dst.xy = src0.xy \times src1.xy + src2.xy$$
$$dst.zw = src0.zw \times src1.zw + src2.zw$$

DFMA (Fused Multiply-Add)

Perform a * b + c with no intermediate rounding step.

$$dst.xy = src0.xy \times src1.xy + src2.xy$$
$$dst.zw = src0.zw \times src1.zw + src2.zw$$

DDIV (Divide)

$$dst.xy = \frac{src0.xy}{src1.xy}$$
$$dst.zw = \frac{src0.zw}{src1.zw}$$

DRCP (Reciprocal)

$$dst.xy = \frac{1}{src.xy}$$
$$dst.zw = \frac{1}{src.zw}$$

DSQRT (Square Root)

$$dst.xy = \sqrt{src.xy}$$
$$dst.zw = \sqrt{src.zw}$$

DRSQ (Reciprocal Square Root)

$$dst.xy = \frac{1}{\sqrt{src.xy}}$$
$$dst.zw = \frac{1}{\sqrt{src.zw}}$$

F2D (Float to Double)

$$dst.xy = double(src0.x)$$
$$dst.zw = double(src0.y)$$

D2F (Double to Float)

$$dst.x = float(src0.xy)$$
$$dst.y = float(src0.zw)$$

I2D (Int to Double)

$$dst.xy = double(src0.x)$$
$$dst.zw = double(src0.y)$$

D2I (Double to Int)

$$dst.x = int(src0.xy)$$
$$dst.y = int(src0.zw)$$

U2D (Unsigned Int to Double)

$$dst.xy = double(src0.x)$$
$$dst.zw = double(src0.y)$$

D2U (Double to Unsigned Int)

$$dst.x = unsigned(src0.xy)$$
$$dst.y = unsigned(src0.zw)$$

64-bit Integer ISA

The 64-bit integer opcodes reinterpret four-component vectors into two-component vectors with 64-bits in each component.

I64ABS (64-bit Integer Absolute Value)

$$dst.xy = |src0.xy|$$
$$dst.zw = |src0.zw|$$

I64NEG (64-bit Integer Negate)

Two's complement.

$$dst.xy = -src.xy$$
$$dst.zw = -src.zw$$

I64SSG (64-bit Integer Set Sign)

$$dst.xy = (src0.xy < 0)? - 1 : (src0.xy > 0)?1 : 0$$
$$dst.zw = (src0.zw < 0)? - 1 : (src0.zw > 0)?1 : 0$$

U64ADD (64-bit Integer Add)

$$dst.xy = src0.xy + src1.xy$$
$$dst.zw = src0.zw + src1.zw$$

U64MUL (64-bit Integer Multiply)

$$dst.xy = src0.xy * src1.xy$$
$$dst.zw = src0.zw * src1.zw$$

U64SEQ (64-bit Integer Set on Equal)

$$dst.x = src0.xy == src1.xy? \sim 0 : 0$$
$$dst.z = src0.zw == src1.zw? \sim 0 : 0$$

U64SNE (64-bit Integer Set on Not Equal)

$$dst.x = src0.xy! = src1.xy? \sim 0 : 0$$
$$dst.z = src0.zw! = src1.zw? \sim 0 : 0$$

U64SLT (64-bit Unsigned Integer Set on Less Than)

$$dst.x = src0.xy < src1.xy? \sim 0 : 0$$
$$dst.z = src0.zw < src1.zw? \sim 0 : 0$$

U64SGE (64-bit Unsigned Integer Set on Greater Equal)

$$dst.x = src0.xy >= src1.xy? \sim 0 : 0$$
$$dst.z = src0.zw >= src1.zw? \sim 0 : 0$$

I64SLT (64-bit Signed Integer Set on Less Than)

$$dst.x = src0.xy < src1.xy? \sim 0 : 0$$
$$dst.z = src0.zw < src1.zw? \sim 0 : 0$$

I64SGE (64-bit Signed Integer Set on Greater Equal)

$$dst.x = src0.xy >= src1.xy? \sim 0 : 0$$
$$dst.z = src0.zw >= src1.zw? \sim 0 : 0$$

I64MIN (Minimum of 64-bit Signed Integers)

$$dst.xy = min(src0.xy, src1.xy)$$
$$dst.zw = min(src0.zw, src1.zw)$$

U64MIN (Minimum of 64-bit Unsigned Integers)

$$dst.xy = min(src0.xy, src1.xy)$$
$$dst.zw = min(src0.zw, src1.zw)$$

I64MAX (Maximum of 64-bit Signed Integers)

$$dst.xy = max(src0.xy, src1.xy)$$
$$dst.zw = max(src0.zw, src1.zw)$$

U64MAX (Maximum of 64-bit Unsigned Integers)

$$dst.xy = max(src0.xy, src1.xy)$$
$$dst.zw = max(src0.zw, src1.zw)$$

U64SHL (Shift Left 64-bit Unsigned Integer)

The shift count is masked with 0x3f before the shift is applied.

$$dst.xy = src0.xy << (0x3f \& src1.x)$$
$$dst.zw = src0.zw << (0x3f \& src1.y)$$

I64SHR (Arithmetic Shift Right (of 64-bit Signed Integer))

The shift count is masked with 0x3f before the shift is applied.

$$dst.xy = src0.xy >> (0x3f \& src1.x)$$
$$dst.zw = src0.zw >> (0x3f \& src1.y)$$

U64SHR (Logical Shift Right (of 64-bit Unsigned Integer))

The shift count is masked with 0x3f before the shift is applied.

$$dst.xy = src0.xy >> (unsigned)(0x3f \& src1.x)$$
$$dst.zw = src0.zw >> (unsigned)(0x3f \& src1.y)$$

I64DIV (64-bit Signed Integer Division)

$$dst.xy = \frac{src0.xy}{src1.xy}$$
$$dst.zw = \frac{src0.zw}{src1.zw}$$

U64DIV (64-bit Unsigned Integer Division)

$$dst.xy = \frac{src0.xy}{src1.xy}$$
$$dst.zw = \frac{src0.zw}{src1.zw}$$

U64MOD (64-bit Unsigned Integer Remainder)

$$dst.xy = src0.xy \bmod src1.xy$$
$$dst.zw = src0.zw \bmod src1.zw$$

I64MOD (64-bit Signed Integer Remainder)

$$dst.xy = src0.xy \bmod src1.xy$$
$$dst.zw = src0.zw \bmod src1.zw$$

F2U64 (Float to 64-bit Unsigned Int)

$$dst.xy = (uint64_t)src0.x$$
$$dst.zw = (uint64_t)src0.y$$

F2I64 (Float to 64-bit Int)

$$dst.xy = (int64_t)src0.x$$
$$dst.zw = (int64_t)src0.y$$

U2I64 (Unsigned Integer to 64-bit Integer)

This is a zero extension.

$$dst.xy = (int64_t)src0.x$$
$$dst.zw = (int64_t)src0.y$$

I2I64 (Signed Integer to 64-bit Integer)

This is a sign extension.

$$dst.xy = (int64_t)src0.x$$
$$dst.zw = (int64_t)src0.y$$

D2U64 (Double to 64-bit Unsigned Int)

$$dst.xy = (uint64_t)src0.xy$$
$$dst.zw = (uint64_t)src0.zw$$

D2I64 (Double to 64-bit Int)

$$dst.xy = (int64_t)src0.xy$$
$$dst.zw = (int64_t)src0.zw$$

U642F (64-bit unsigned integer to float)

$$dst.x = (float)src0.xy$$
$$dst.y = (float)src0.zw$$

I642F (64-bit Int to Float)

$$dst.x = (float)src0.xy$$
$$dst.y = (float)src0.zw$$

U642D (64-bit unsigned integer to double)

$$dst.xy = (double)src0.xy$$
$$dst.zw = (double)src0.zw$$

I642D (64-bit Int to double)

$$dst.xy = (double)src0.xy$$
$$dst.zw = (double)src0.zw$$

Resource Sampling Opcodes

Those opcodes follow very closely semantics of the respective Direct3D instructions. If in doubt double check Direct3D documentation. Note that the swizzle on SVIEW (src1) determines texel swizzling after lookup.

SAMPLE

Using provided address, sample data from the specified texture using the filtering mode identified by the given sampler. The source data may come from any resource type other than buffers.

Syntax: `SAMPLE dst, address, sampler_view, sampler`

Example: `SAMPLE TEMP[0], TEMP[1], SVIEW[0], SAMP[0]`

SAMPLE_I

Simplified alternative to the SAMPLE instruction. Using the provided integer address, SAMPLE_I fetches data from the specified sampler view without any filtering. The source data may come from any resource type other than CUBE.

Syntax: `SAMPLE_I dst, address, sampler_view`

Example: `SAMPLE_I TEMP[0], TEMP[1], SVIEW[0]`

The 'address' is specified as unsigned integers. If the 'address' is out of range [0…(# texels - 1)] the result of the fetch is always 0 in all components. As such the instruction doesn't honor address wrap modes, in cases

where that behavior is desirable 'SAMPLE' instruction should be used. address.w always provides an unsigned integer mipmap level. If the value is out of the range then the instruction always returns 0 in all components. address.yz are ignored for buffers and 1d textures. address.z is ignored for 1d texture arrays and 2d textures.

For 1D texture arrays address.y provides the array index (also as unsigned integer). If the value is out of the range of available array indices [0… (array size - 1)] then the opcode always returns 0 in all components. For 2D texture arrays address.z provides the array index, otherwise it exhibits the same behavior as in the case for 1D texture arrays. The exact semantics of the source address are presented in the table below:

resource type	X	Y	Z	W
PIPE_BUFFER	x			ignored
PIPE_TEXTURE_1D	x			mpl
PIPE_TEXTURE_2D	x	y		mpl
PIPE_TEXTURE_3D	x	y	z	mpl
PIPE_TEXTURE_RECT	x	y		mpl
PIPE_TEXTURE_CUBE	not allowed as source			
PIPE_TEXTURE_1D_ARRAY	x	idx		mpl
PIPE_TEXTURE_2D_ARRAY	x	y	idx	mpl

Where 'mpl' is a mipmap level and 'idx' is the array index.

SAMPLE_I_MS

Just like SAMPLE_I but allows fetch data from multi-sampled surfaces.

Syntax: `SAMPLE_I_MS dst, address, sampler_view, sample`

SAMPLE_B

Just like the SAMPLE instruction with the exception that an additional bias is applied to the level of detail computed as part of the instruction execution.

Syntax: `SAMPLE_B dst, address, sampler_view, sampler, lod_bias`

Example: `SAMPLE_B TEMP[0], TEMP[1], SVIEW[0], SAMP[0], TEMP[2].x`

SAMPLE_C

Similar to the SAMPLE instruction but it performs a comparison filter. The operands to SAMPLE_C are identical to SAMPLE, except that there is an additional float32 operand, reference value, which must be a register with single-component, or a scalar literal. SAMPLE_C makes the hardware use the current samplers compare_func (in pipe_sampler_state) to compare reference value against the red component value for the surce resource at each texel that the currently configured texture filter covers based on the provided coordinates.

Syntax: `SAMPLE_C dst, address, sampler_view.r, sampler, ref_value`

Example: `SAMPLE_C TEMP[0], TEMP[1], SVIEW[0].r, SAMP[0], TEMP[2].x`

SAMPLE_C_LZ

Same as SAMPLE_C, but LOD is 0 and derivatives are ignored. The LZ stands for level-zero.

Syntax: `SAMPLE_C_LZ dst, address, sampler_view.r, sampler, ref_value`

Example: `SAMPLE_C_LZ TEMP[0], TEMP[1], SVIEW[0].r, SAMP[0], TEMP[2].x`

SAMPLE_D

SAMPLE_D is identical to the SAMPLE opcode except that the derivatives for the source address in the x direction and the y direction are provided by extra parameters.

Syntax: `SAMPLE_D dst, address, sampler_view, sampler, der_x, der_y`

Example: `SAMPLE_D TEMP[0], TEMP[1], SVIEW[0], SAMP[0], TEMP[2], TEMP[3]`

SAMPLE_L

SAMPLE_L is identical to the SAMPLE opcode except that the LOD is provided directly as a scalar value, representing no anisotropy.

Syntax: `SAMPLE_L dst, address, sampler_view, sampler, explicit_lod`

Example: `SAMPLE_L TEMP[0], TEMP[1], SVIEW[0], SAMP[0], TEMP[2].x`

GATHER4

Gathers the four texels to be used in a bi-linear filtering operation and packs them into a single register. Only works with 2D, 2D array, cubemaps, and cubemaps arrays. For 2D textures, only the addressing modes of the sampler and the top level of any mip pyramid are used. Set W to zero. It behaves like the SAMPLE instruction, but a filtered sample is not generated. The four samples that contribute to filtering are placed into xyzw in counter-clockwise order, starting with the (u,v) texture coordinate delta at the following locations (-, +), (+, +), (+, -), (-, -), where the magnitude of the deltas are half a texel.

SVIEWINFO

Query the dimensions of a given sampler view. dst receives width, height, depth or array size and number of mipmap levels as int4. The dst can have a writemask which will specify what info is the caller interested in.

Syntax: `SVIEWINFO dst, src_mip_level, sampler_view`

Example: `SVIEWINFO TEMP[0], TEMP[1].x, SVIEW[0]`

src_mip_level is an unsigned integer scalar. If it's out of range then returns 0 for width, height and depth/array size but the total number of mipmap is still returned correctly for the given sampler view. The returned width, height and depth values are for the mipmap level selected by the src_mip_level and are in the number of texels. For 1d texture array width is in dst.x, array size is in dst.y and dst.z is 0. The number of mipmaps is still in dst.w. In contrast to d3d10 resinfo, there's no way in the tgsi instruction encoding to specify the return type (float/rcpfloat/uint), hence always using uint. Also, unlike the SAMPLE instructions, the swizzle on src1 resinfo allowing swizzling dst values is ignored (due to the interaction with rcpfloat modifier which requires some swizzle handling in the state tracker anyway).

SAMPLE_POS

Query the position of a sample in the given resource or render target when per-sample fragment shading is in effect.

Syntax: `SAMPLE_POS dst, source, sample_index`

dst receives float4 (x, y, undef, undef) indicated where the sample is located. Sample locations are in the range [0, 1] where 0.5 is the center of the fragment.

source is either a sampler view (to indicate a shader resource) or temp register (to indicate the render target). The source register may have an optional swizzle to apply to the returned result

sample_index is an integer scalar indicating which sample position is to be queried.

If per-sample shading is not in effect or the source resource or render target is not multisampled, the result is (0.5, 0.5, undef, undef).

NOTE: no driver has implemented this opcode yet (and no gallium frontend emits it). This information is subject to change.

SAMPLE_INFO

Query the number of samples in a multisampled resource or render target.

Syntax: `SAMPLE_INFO dst, source`

dst receives int4 (n, 0, 0, 0) where n is the number of samples in a resource or the render target.

source is either a sampler view (to indicate a shader resource) or temp register (to indicate the render target). The source register may have an optional swizzle to apply to the returned result

If per-sample shading is not in effect or the source resource or render target is not multisampled, the result is (1, 0, 0, 0).

NOTE: no driver has implemented this opcode yet (and no gallium frontend emits it). This information is subject to change.

LOD (level of detail)
Same syntax as the SAMPLE opcode but instead of performing an actual texture lookup/filter, return the computed LOD information that the texture pipe would use to access the texture. The Y component contains the computed LOD lambda_prime. The X component contains the LOD that will be accessed, based on min/max lod's and mipmap filters. The Z and W components are set to 0.

Syntax: `LOD dst, address, sampler_view, sampler`

Resource Access Opcodes

For these opcodes, the resource can be a BUFFER, IMAGE, or MEMORY.

LOAD (Fetch data from a shader buffer or image)
Syntax: `LOAD dst, resource, address`

Example: `LOAD TEMP[0], BUFFER[0], TEMP[1]`

Using the provided integer address, LOAD fetches data from the specified buffer or texture without any filtering.

The 'address' is specified as a vector of unsigned integers. If the 'address' is out of range the result is unspecified.

Only the first mipmap level of a resource can be read from using this instruction.

For 1D or 2D texture arrays, the array index is provided as an unsigned integer in address.y or address.z, respectively. address.yz are ignored for buffers and 1D textures. address.z is ignored for 1D texture arrays and 2D textures. address.w is always ignored.

A swizzle suffix may be added to the resource argument this will cause the resource data to be swizzled accordingly.

STORE (Write data to a shader resource)
Syntax: `STORE resource, address, src`

Example: `STORE BUFFER[0], TEMP[0], TEMP[1]`

Using the provided integer address, STORE writes data to the specified buffer or texture.

The 'address' is specified as a vector of unsigned integers. If the 'address' is out of range the result is unspecified.

Only the first mipmap level of a resource can be written to using this instruction.

For 1D or 2D texture arrays, the array index is provided as an unsigned integer in address.y or address.z, respectively. address.yz are ignored for buffers and 1D textures. address.z is ignored for 1D texture arrays and 2D textures. address.w is always ignored.

RESQ (Query information about a resource)
Syntax: `RESQ dst, resource`

Example: `RESQ TEMP[0], BUFFER[0]`

Returns information about the buffer or image resource. For buffer resources, the size (in bytes) is returned in the x component. For image resources, .xyz will contain the width/height/layers of the image, while .w will contain the number of samples for multi-sampled images.

FBFETCH (Load data from framebuffer)
Syntax: `FBFETCH dst, output`

Example: `FBFETCH TEMP[0], OUT[0]`

This is only valid on `COLOR` semantic outputs. Returns the color of the current position in the framebuffer from before this fragment shader invocation. May return the same value from multiple calls for a particular output within a single invocation. Note that result may be undefined if a fragment is drawn multiple times without a blend barrier in between.

Bindless Opcodes

These opcodes are for working with bindless sampler or image handles and require PIPE_CAP_BINDLESS_TEXTURE.

IMG2HND (Get a bindless handle for a image)

Syntax: `IMG2HND dst, image`

Example: `IMG2HND TEMP[0], IMAGE[0]`

Sets 'dst' to a bindless handle for 'image'.

SAMP2HND (Get a bindless handle for a sampler)

Syntax: `SAMP2HND dst, sampler`

Example: `SAMP2HND TEMP[0], SAMP[0]`

Sets 'dst' to a bindless handle for 'sampler'.

Inter-thread synchronization opcodes

These opcodes are intended for communication between threads running within the same compute grid. For now they're only valid in compute programs.

BARRIER (Thread group barrier)

`BARRIER`

This opcode suspends the execution of the current thread until all the remaining threads in the working group reach the same point of the program. Results are unspecified if any of the remaining threads terminates or never reaches an executed BARRIER instruction.

MEMBAR (Memory barrier)

`MEMBAR type`

This opcode waits for the completion of all memory accesses based on the type passed in. The type is an immediate bitfield with the following meaning:

Bit 0: Shader storage buffers Bit 1: Atomic buffers Bit 2: Images Bit 3: Shared memory Bit 4: Thread group

These may be passed in in any combination. An implementation is free to not distinguish between these as it sees fit. However these map to all the possibilities made available by GLSL.

Atomic opcodes

These opcodes provide atomic variants of some common arithmetic and logical operations. In this context atomicity means that another concurrent memory access operation that affects the same memory location is guaranteed to be performed strictly before or after the entire execution of the atomic operation. The resource may be a BUFFER, IMAGE, HWATOMIC, or MEMORY. In the case of an image, the offset works the same as for `LOAD` and `STORE`, specified above. For atomic counters, the offset is an immediate index to the base hw atomic counter for this operation. These atomic operations may only be used with 32-bit integer image formats.

ATOMUADD (Atomic integer addition)

Syntax: `ATOMUADD dst, resource, offset, src`

Example: `ATOMUADD TEMP[0], BUFFER[0], TEMP[1], TEMP[2]`

The following operation is performed atomically:

$$dst_x = resource[offset]$$
$$resource[offset] = dst_x + src_x$$

ATOMFADD (Atomic floating point addition)

Syntax: `ATOMFADD dst, resource, offset, src`

Example: `ATOMFADD TEMP[0], BUFFER[0], TEMP[1], TEMP[2]`

The following operation is performed atomically:

$$dst_x = resource[offset]$$
$$resource[offset] = dst_x + src_x$$

ATOMXCHG (Atomic exchange)

Syntax: `ATOMXCHG dst, resource, offset, src`

Example: `ATOMXCHG TEMP[0], BUFFER[0], TEMP[1], TEMP[2]`

The following operation is performed atomically:

$$dst_x = resource[offset]$$
$$resource[offset] = src_x$$

ATOMCAS (Atomic compare-and-exchange)

Syntax: `ATOMCAS dst, resource, offset, cmp, src`

Example: `ATOMCAS TEMP[0], BUFFER[0], TEMP[1], TEMP[2], TEMP[3]`

The following operation is performed atomically:

$$dst_x = resource[offset]$$
$$resource[offset] = (dst_x == cmp_x ? src_x : dst_x)$$

ATOMAND (Atomic bitwise And)

Syntax: `ATOMAND dst, resource, offset, src`

Example: `ATOMAND TEMP[0], BUFFER[0], TEMP[1], TEMP[2]`

The following operation is performed atomically:

$$dst_x = resource[offset]$$
$$resource[offset] = dst_x \& src_x$$

ATOMOR (Atomic bitwise Or)

Syntax: `ATOMOR dst, resource, offset, src`

Example: `ATOMOR TEMP[0], BUFFER[0], TEMP[1], TEMP[2]`

The following operation is performed atomically:

$$dst_x = resource[offset]$$
$$resource[offset] = dst_x | src_x$$

ATOMXOR (Atomic bitwise Xor)

Syntax: `ATOMXOR dst, resource, offset, src`

Example: `ATOMXOR TEMP[0], BUFFER[0], TEMP[1], TEMP[2]`

The following operation is performed atomically:

$$dst_x = resource[offset]$$
$$resource[offset] = dst_x \oplus src_x$$

ATOMUMIN (Atomic unsigned minimum)

Syntax: `ATOMUMIN dst, resource, offset, src`

Example: `ATOMUMIN TEMP[0], BUFFER[0], TEMP[1], TEMP[2]`

The following operation is performed atomically:

$$dst_x = resource[offset]$$
$$resource[offset] = (dst_x < src_x ? dst_x : src_x)$$

ATOMUMAX (Atomic unsigned maximum)

Syntax: `ATOMUMAX dst, resource, offset, src`

Example: `ATOMUMAX TEMP[0], BUFFER[0], TEMP[1], TEMP[2]`

The following operation is performed atomically:

$$dst_x = resource[offset]$$
$$resource[offset] = (dst_x > src_x ? dst_x : src_x)$$

ATOMIMIN (Atomic signed minimum)

Syntax: `ATOMIMIN dst, resource, offset, src`

Example: `ATOMIMIN TEMP[0], BUFFER[0], TEMP[1], TEMP[2]`

The following operation is performed atomically:

$$dst_x = resource[offset]$$
$$resource[offset] = (dst_x < src_x ? dst_x : src_x)$$

ATOMIMAX (Atomic signed maximum)

Syntax: `ATOMIMAX dst, resource, offset, src`

Example: `ATOMIMAX TEMP[0], BUFFER[0], TEMP[1], TEMP[2]`

The following operation is performed atomically:

$$dst_x = resource[offset]$$
$$resource[offset] = (dst_x > src_x ? dst_x : src_x)$$

ATOMINC_WRAP (Atomic increment + wrap around)

Syntax: `ATOMINC_WRAP dst, resource, offset, src`

Example: `ATOMINC_WRAP TEMP[0], BUFFER[0], TEMP[1], TEMP[2]`

The following operation is performed atomically:

$$dst_x = resource[offset] + 1$$
$$resource[offset] = dst_x <= src_x ? dst_x : 0$$

ATOMDEC_WRAP (Atomic decrement + wrap around)

Syntax: `ATOMDEC_WRAP dst, resource, offset, src`

Example: `ATOMDEC_WRAP TEMP[0], BUFFER[0], TEMP[1], TEMP[2]`

The following operation is performed atomically:

$$dst_x = resource[offset]$$
$$resource[offset] = (dst_x > 0 \quad dst_x < src_x) ? dst_x - 1 : 0$$

Inter-lane opcodes

These opcodes reduce the given value across the shader invocations running in the current SIMD group. Every thread in the subgroup will receive the same result. The BALLOT operations accept a single-channel argument that is treated as a boolean and produce a 64-bit value.

VOTE_ANY (Value is set in any of the active invocations)

Syntax: `VOTE_ANY dst, value`

Example: `VOTE_ANY TEMP[0].x, TEMP[1].x`

VOTE_ALL (Value is set in all of the active invocations)

Syntax: `VOTE_ALL dst, value`

Example: `VOTE_ALL TEMP[0].x, TEMP[1].x`

VOTE_EQ (Value is the same in all of the active invocations)

Syntax: `VOTE_EQ dst, value`

Example: `VOTE_EQ TEMP[0].x, TEMP[1].x`

BALLOT (Lanemask of whether the value is set in each active)

invocation

Syntax: `BALLOT dst, value`

Example: `BALLOT TEMP[0].xy, TEMP[1].x`

When the argument is a constant true, this produces a bitmask of active invocations. In fragment shaders, this can include helper invocations (invocations whose outputs and writes to memory are discarded, but which are used to compute derivatives).

READ_FIRST (Broadcast the value from the first active) invocation to all active lanes

Syntax: `READ_FIRST dst, value`

Example: `READ_FIRST TEMP[0], TEMP[1]`

READ_INVOC (Retrieve the value from the given invocation) (need not be uniform)

Syntax: `READ_INVOC dst, value, invocation`

Example: `READ_INVOC TEMP[0].xy, TEMP[1].xy, TEMP[2].x`

invocation.x controls the invocation number to read from for all channels. The invocation number must be the same across all active invocations in a sub-group; otherwise, the results are undefined.

34.3.3 Explanation of symbols used

Functions

$|x|$ Absolute value of x.

$\lceil x \rceil$ Ceiling of x.

clamp(x,y,z) Clamp x between y and z. (x < y) ? y : (x > z) ? z : x

$\lfloor x \rfloor$ Floor of x.

$\log_2 x$ Logarithm of x, base 2.

max(x,y) Maximum of x and y. (x > y) ? x : y

min(x,y) Minimum of x and y. (x < y) ? x : y

partialx(x) Derivative of x relative to fragment's X.

partialy(x) Derivative of x relative to fragment's Y.

pop() Pop from stack.

x^y x to the power y.

push(x) Push x on stack.

round(x) Round x.

trunc(x) Truncate x, i.e. drop the fraction bits.

Keywords

discard Discard fragment.

pc Program counter.

target Label of target instruction.

34.3.4 Other tokens

Declaration

Declares a register that is will be referenced as an operand in Instruction tokens.

File field contains register file that is being declared and is one of TGSI_FILE.

UsageMask field specifies which of the register components can be accessed and is one of TGSI_WRITEMASK.

The Local flag specifies that a given value isn't intended for subroutine parameter passing and, as a result, the implementation isn't required to give any guarantees of it being preserved across subroutine boundaries. As it's merely a compiler hint, the implementation is free to ignore it.

If Dimension flag is set to 1, a Declaration Dimension token follows.

If Semantic flag is set to 1, a Declaration Semantic token follows.

If Interpolate flag is set to 1, a Declaration Interpolate token follows.

If file is TGSI_FILE_RESOURCE, a Declaration Resource token follows.

If Array flag is set to 1, a Declaration Array token follows.

Array Declaration

Declarations can optional have an ArrayID attribute which can be referred by indirect addressing operands. An ArrayID of zero is reserved and treated as if no ArrayID is specified.

If an indirect addressing operand refers to a specific declaration by using an ArrayID only the registers in this declaration are guaranteed to be accessed, accessing any register outside this declaration results in undefined behavior. Note that for compatibility the effective index is zero-based and not relative to the specified declaration

If no ArrayID is specified with an indirect addressing operand the whole register file might be accessed by this operand. This is strongly discouraged and will prevent packing of scalar/vec2 arrays and effective alias analysis. This is only legal for TEMP and CONST register files.

Declaration Semantic

Vertex and fragment shader input and output registers may be labeled with semantic information consisting of a name and index.

Follows Declaration token if Semantic bit is set.

Since its purpose is to link a shader with other stages of the pipeline, it is valid to follow only those Declaration tokens that declare a register either in INPUT or OUTPUT file.

SemanticName field contains the semantic name of the register being declared. There is no default value.

SemanticIndex is an optional subscript that can be used to distinguish different register declarations with the same semantic name. The default value is 0.

The meanings of the individual semantic names are explained in the following sections.

TGSI_SEMANTIC_POSITION

For vertex shaders, TGSI_SEMANTIC_POSITION indicates the vertex shader output register which contains the homogeneous vertex position in the clip space coordinate system. After clipping, the X, Y and Z components of the vertex will be divided by the W value to get normalized device coordinates.

For fragment shaders, TGSI_SEMANTIC_POSITION is used to indicate that fragment shader input (or system value, depending on which one is supported by the driver) contains the fragment's window position. The X component starts at zero and always increases from left to right. The Y component starts at zero and always increases but Y=0 may either indicate the top of the window or the bottom depending on the fragment coordinate origin convention (see TGSI_PROPERTY_FS_COORD_ORIGIN). The Z coordinate ranges from 0 to 1 to represent depth from the front to the back of the Z buffer. The W component contains the interpolated reciprocal of the vertex position W component (corresponding to gl_Fragcoord, but unlike d3d10 which interpolates the same 1/w but then gives back the reciprocal of the interpolated value).

Fragment shaders may also declare an output register with TGSI_SEMANTIC_POSITION. Only the Z component is writable. This allows the fragment shader to change the fragment's Z position.

TGSI_SEMANTIC_COLOR

For vertex shader outputs or fragment shader inputs/outputs, this label indicates that the register contains an R,G,B,A color.

Several shader inputs/outputs may contain colors so the semantic index is used to distinguish them. For example, color[0] may be the diffuse color while color[1] may be the specular color.

This label is needed so that the flat/smooth shading can be applied to the right interpolants during rasterization.

TGSI_SEMANTIC_BCOLOR

Back-facing colors are only used for back-facing polygons, and are only valid in vertex shader outputs. After rasterization, all polygons are front-facing and COLOR and BCOLOR end up occupying the same slots in the fragment shader, so all BCOLORs effectively become regular COLORs in the fragment shader.

TGSI_SEMANTIC_FOG

Vertex shader inputs and outputs and fragment shader inputs may be labeled with TGSI_SEMANTIC_FOG to indicate that the register contains a fog coordinate. Typically, the fragment shader will use the fog coordinate to compute a fog blend factor which is used to blend the normal fragment color with a constant fog color. But fog coord really is just an ordinary vec4 register like regular semantics.

TGSI_SEMANTIC_PSIZE

Vertex shader input and output registers may be labeled with TGIS_SEMANTIC_PSIZE to indicate that the register contains a point size in the form (S, 0, 0, 1). The point size controls the width or diameter of points for rasterization. This label cannot be used in fragment shaders.

When using this semantic, be sure to set the appropriate state in the *Rasterizer* first.

TGSI_SEMANTIC_TEXCOORD

Only available if PIPE_CAP_TGSI_TEXCOORD is exposed !

Vertex shader outputs and fragment shader inputs may be labeled with this semantic to make them replaceable by sprite coordinates via the sprite_coord_enable state in the *Rasterizer*. The semantic index permitted with this semantic is limited to <= 7.

If the driver does not support TEXCOORD, sprite coordinate replacement applies to inputs with the GENERIC semantic instead.

The intended use case for this semantic is gl_TexCoord.

TGSI_SEMANTIC_PCOORD

Only available if PIPE_CAP_TGSI_TEXCOORD is exposed !

Fragment shader inputs may be labeled with TGSI_SEMANTIC_PCOORD to indicate that the register contains sprite coordinates in the form (x, y, 0, 1), if the current primitive is a point and point sprites are enabled. Otherwise, the contents of the register are undefined.

The intended use case for this semantic is gl_PointCoord.

TGSI_SEMANTIC_GENERIC

All vertex/fragment shader inputs/outputs not labeled with any other semantic label can be considered to be generic attributes. Typical uses of generic inputs/outputs are texcoords and user-defined values.

TGSI_SEMANTIC_NORMAL

Indicates that a vertex shader input is a normal vector. This is typically only used for legacy graphics APIs.

TGSI_SEMANTIC_FACE

This label applies to fragment shader inputs (or system values, depending on which one is supported by the driver) and indicates that the register contains front/back-face information.

If it is an input, it will be a floating-point vector in the form (F, 0, 0, 1), where F will be positive when the fragment belongs to a front-facing polygon, and negative when the fragment belongs to a back-facing polygon.

If it is a system value, it will be an integer vector in the form (F, 0, 0, 1), where F is 0xffffffff when the fragment belongs to a front-facing polygon and 0 when the fragment belongs to a back-facing polygon.

TGSI_SEMANTIC_EDGEFLAG

For vertex shaders, this sematic label indicates that an input or output is a boolean edge flag. The register layout is [F, x, x, x] where F is 0.0 or 1.0 and x = don't care. Normally, the vertex shader simply copies the edge flag input to the edgeflag output.

Edge flags are used to control which lines or points are actually drawn when the polygon mode converts triangles/quads/polygons into points or lines.

TGSI_SEMANTIC_STENCIL

For fragment shaders, this semantic label indicates that an output is a writable stencil reference value. Only the Y component is writable. This allows the fragment shader to change the fragments stencilref value.

TGSI_SEMANTIC_VIEWPORT_INDEX

For geometry shaders, this semantic label indicates that an output contains the index of the viewport (and scissor) to use. This is an integer value, and only the X component is used.

If PIPE_CAP_TGSI_VS_LAYER_VIEWPORT or PIPE_CAP_TGSI_TES_LAYER_VIEWPORT is supported, then this semantic label can also be used in vertex or tessellation evaluation shaders, respectively. Only the value written in the last vertex processing stage is used.

TGSI_SEMANTIC_LAYER

For geometry shaders, this semantic label indicates that an output contains the layer value to use for the color and depth/stencil surfaces. This is an integer value, and only the X component is used. (Also known as rendertarget array index.)

If PIPE_CAP_TGSI_VS_LAYER_VIEWPORT or PIPE_CAP_TGSI_TES_LAYER_VIEWPORT is supported, then this semantic label can also be used in vertex or tessellation evaluation shaders, respectively. Only the value written in the last vertex processing stage is used.

TGSI_SEMANTIC_CLIPDIST

Note this covers clipping and culling distances.

When components of vertex elements are identified this way, these values are each assumed to be a float32 signed distance to a plane.

For clip distances: Primitive setup only invokes rasterization on pixels for which the interpolated plane distances are >= 0.

For cull distances: Primitives will be completely discarded if the plane distance for all of the vertices in the primitive are < 0. If a vertex has a cull distance of NaN, that vertex counts as "out" (as if its < 0);

Multiple clip/cull planes can be implemented simultaneously, by annotating multiple components of one or more vertex elements with the above specified semantic. The limits on both clip and cull distances are bound by the PIPE_MAX_CLIP_OR_CULL_DISTANCE_COUNT define which defines the maximum number of components that can be used to hold the distances and by the PIPE_MAX_CLIP_OR_CULL_DISTANCE_ELEMENT_COUNT which specifies the maximum number of registers which can be annotated with those semantics. The properties NUM_CLIPDIST_ENABLED and NUM_CULLDIST_ENABLED are used to divide up the 2 x vec4 space between clipping and culling.

TGSI_SEMANTIC_SAMPLEID

For fragment shaders, this semantic label indicates that a system value contains the current sample id (i.e. gl_SampleID) as an unsigned int. Only the X component is used. If per-sample shading is not enabled, the result is (0, undef, undef, undef).

Note that if the fragment shader uses this system value, the fragment shader is automatically executed at per sample frequency.

TGSI_SEMANTIC_SAMPLEPOS

For fragment shaders, this semantic label indicates that a system value contains the current sample's position as float4(x, y, undef, undef) in the render target (i.e. gl_SamplePosition) when per-fragment shading is in effect. Position

values are in the range [0, 1] where 0.5 is the center of the fragment.

Note that if the fragment shader uses this system value, the fragment shader is automatically executed at per sample frequency.

TGSI_SEMANTIC_SAMPLEMASK

For fragment shaders, this semantic label can be applied to either a shader system value input or output.

For a system value, the sample mask indicates the set of samples covered by the current primitive. If MSAA is not enabled, the value is (1, 0, 0, 0).

For an output, the sample mask is used to disable further sample processing.

For both, the register type is uint[4] but only the X component is used (i.e. gl_SampleMask[0]). Each bit corresponds to one sample position (up to 32x MSAA is supported).

TGSI_SEMANTIC_INVOCATIONID

For geometry shaders, this semantic label indicates that a system value contains the current invocation id (i.e. gl_InvocationID). This is an integer value, and only the X component is used.

TGSI_SEMANTIC_INSTANCEID

For vertex shaders, this semantic label indicates that a system value contains the current instance id (i.e. gl_InstanceID). It does not include the base instance. This is an integer value, and only the X component is used.

TGSI_SEMANTIC_VERTEXID

For vertex shaders, this semantic label indicates that a system value contains the current vertex id (i.e. gl_VertexID). It does (unlike in d3d10) include the base vertex. This is an integer value, and only the X component is used.

TGSI_SEMANTIC_VERTEXID_NOBASE

For vertex shaders, this semantic label indicates that a system value contains the current vertex id without including the base vertex (this corresponds to d3d10 vertex id, so TGSI_SEMANTIC_VERTEXID_NOBASE + TGSI_SEMANTIC_BASEVERTEX == TGSI_SEMANTIC_VERTEXID). This is an integer value, and only the X component is used.

TGSI_SEMANTIC_BASEVERTEX

For vertex shaders, this semantic label indicates that a system value contains the base vertex (i.e. gl_BaseVertex). Note that for non-indexed draw calls, this contains the first (or start) value instead. This is an integer value, and only the X component is used.

TGSI_SEMANTIC_PRIMID

For geometry and fragment shaders, this semantic label indicates the value contains the primitive id (i.e. gl_PrimitiveID). This is an integer value, and only the X component is used. FIXME: This right now can be either a ordinary input or a system value…

TGSI_SEMANTIC_PATCH

For tessellation evaluation/control shaders, this semantic label indicates a generic per-patch attribute. Such semantics will not implicitly be per-vertex arrays.

TGSI_SEMANTIC_TESSCOORD

For tessellation evaluation shaders, this semantic label indicates the coordinates of the vertex being processed. This is available in XYZ; W is undefined.

TGSI_SEMANTIC_TESSOUTER

For tessellation evaluation/control shaders, this semantic label indicates the outer tessellation levels of the patch. Isoline tessellation will only have XY defined, triangle will have XYZ and quads will have XYZW defined. This corresponds to gl_TessLevelOuter.

TGSI_SEMANTIC_TESSINNER

For tessellation evaluation/control shaders, this semantic label indicates the inner tessellation levels of the patch. The X value is only defined for triangle tessellation, while quads will have XY defined. This is entirely undefined for isoline tessellation.

TGSI_SEMANTIC_VERTICESIN

For tessellation evaluation/control shaders, this semantic label indicates the number of vertices provided in the input patch. Only the X value is defined.

TGSI_SEMANTIC_HELPER_INVOCATION

For fragment shaders, this semantic indicates whether the current invocation is covered or not. Helper invocations are created in order to properly compute derivatives, however it may be desirable to skip some of the logic in those cases. See `gl_HelperInvocation` documentation.

TGSI_SEMANTIC_BASEINSTANCE

For vertex shaders, the base instance argument supplied for this draw. This is an integer value, and only the X component is used.

TGSI_SEMANTIC_DRAWID

For vertex shaders, the zero-based index of the current draw in a `glMultiDraw*` invocation. This is an integer value, and only the X component is used.

TGSI_SEMANTIC_WORK_DIM

For compute shaders started via opencl this retrieves the work_dim parameter to the clEnqueueNDRangeKernel call with which the shader was started.

TGSI_SEMANTIC_GRID_SIZE

For compute shaders, this semantic indicates the maximum (x, y, z) dimensions of a grid of thread blocks.

TGSI_SEMANTIC_BLOCK_ID

For compute shaders, this semantic indicates the (x, y, z) coordinates of the current block inside of the grid.

TGSI_SEMANTIC_BLOCK_SIZE

For compute shaders, this semantic indicates the maximum (x, y, z) dimensions of a block in threads.

TGSI_SEMANTIC_THREAD_ID

For compute shaders, this semantic indicates the (x, y, z) coordinates of the current thread inside of the block.

TGSI_SEMANTIC_SUBGROUP_SIZE

This semantic indicates the subgroup size for the current invocation. This is an integer of at most 64, as it indicates the width of lanemasks. It does not depend on the number of invocations that are active.

TGSI_SEMANTIC_SUBGROUP_INVOCATION

The index of the current invocation within its subgroup.

TGSI_SEMANTIC_SUBGROUP_EQ_MASK

A bit mask of `bit index == TGSI_SEMANTIC_SUBGROUP_INVOCATION`, i.e. `1 << subgroup_invocation` in arbitrary precision arithmetic.

TGSI_SEMANTIC_SUBGROUP_GE_MASK

A bit mask of `bit index >= TGSI_SEMANTIC_SUBGROUP_INVOCATION`, i.e. `((1 << (subgroup_size - subgroup_invocation)) - 1) << subgroup_invocation` in arbitrary precision arithmetic.

TGSI_SEMANTIC_SUBGROUP_GT_MASK

A bit mask of `bit index > TGSI_SEMANTIC_SUBGROUP_INVOCATION`, i.e. `((1 << (subgroup_size - subgroup_invocation - 1)) - 1) << (subgroup_invocation + 1)` in arbitrary precision arithmetic.

TGSI_SEMANTIC_SUBGROUP_LE_MASK

A bit mask of `bit index <= TGSI_SEMANTIC_SUBGROUP_INVOCATION`, i.e. `(1 << (subgroup_invocation + 1)) - 1` in arbitrary precision arithmetic.

TGSI_SEMANTIC_SUBGROUP_LT_MASK

A bit mask of `bit index < TGSI_SEMANTIC_SUBGROUP_INVOCATION`, i.e. `(1 << subgroup_invocation) - 1` in arbitrary precision arithmetic.

TGSI_SEMANTIC_VIEWPORT_MASK

A bit mask of viewports to broadcast the current primitive to. See GL_NV_viewport_array2 for more details.

TGSI_SEMANTIC_TESS_DEFAULT_OUTER_LEVEL

A system value equal to the default_outer_level array set via set_tess_level.

TGSI_SEMANTIC_TESS_DEFAULT_INNER_LEVEL

A system value equal to the default_inner_level array set via set_tess_level.

Declaration Interpolate

This token is only valid for fragment shader INPUT declarations.

The Interpolate field specifes the way input is being interpolated by the rasteriser and is one of TGSI_INTERPOLATE_*.

The Location field specifies the location inside the pixel that the interpolation should be done at, one of `TGSI_INTERPOLATE_LOC_*`. Note that when per-sample shading is enabled, the implementation may choose to interpolate at the sample irrespective of the Location field.

The CylindricalWrap bitfield specifies which register components should be subject to cylindrical wrapping when interpolating by the rasteriser. If TGSI_CYLINDRICAL_WRAP_X is set to 1, the X component should be interpolated according to cylindrical wrapping rules.

Declaration Sampler View

Follows Declaration token if file is TGSI_FILE_SAMPLER_VIEW.

DCL SVIEW[#], resource, type(s)

Declares a shader input sampler view and assigns it to a SVIEW[#] register.

resource can be one of BUFFER, 1D, 2D, 3D, 1DArray and 2DArray.

type must be 1 or 4 entries (if specifying on a per-component level) out of UNORM, SNORM, SINT, UINT and FLOAT.

For TEX* style texture sample opcodes (as opposed to SAMPLE* opcodes which take an explicit SVIEW[#] source register), there may be optionally SVIEW[#] declarations. In this case, the SVIEW index is implied by the SAMP index, and there must be a corresponding SVIEW[#] declaration for each SAMP[#] declaration. Drivers are free to ignore this if they wish. But note in particular that some drivers need to know the sampler type (float/int/unsigned) in order to generate the correct code, so cases where integer textures are sampled, SVIEW[#] declarations should be used.

NOTE: It is NOT legal to mix SAMPLE* style opcodes and TEX* opcodes in the same shader.

Declaration Resource

Follows Declaration token if file is TGSI_FILE_RESOURCE.

DCL RES[#], resource [, WR] [, RAW]

Declares a shader input resource and assigns it to a RES[#] register.

resource can be one of BUFFER, 1D, 2D, 3D, CUBE, 1DArray and 2DArray.

If the RAW keyword is not specified, the texture data will be subject to conversion, swizzling and scaling as required to yield the specified data type from the physical data format of the bound resource.

If the RAW keyword is specified, no channel conversion will be performed: the values read for each of the channels (X,Y,Z,W) will correspond to consecutive words in the same order and format they're found in memory. No element-to-address conversion will be performed either: the value of the provided X coordinate will be interpreted in byte units instead of texel units. The result of accessing a misaligned address is undefined.

Usage of the STORE opcode is only allowed if the WR (writable) flag is set.

Hardware Atomic Register File

Hardware atomics are declared as a 2D array with an optional array id.

The first member of the dimension is the buffer resource the atomic is located in. The second member is a range into the buffer resource, either for one or multiple counters. If this is an array, the declaration will have an unique array id.

Each counter is 4 bytes in size, and index and ranges are in counters not bytes. DCL HWATOMIC[0][0] DCL HWATOMIC[0][1]

This declares two atomics, one at the start of the buffer and one in the second 4 bytes.

DCL HWATOMIC[0][0] DCL HWATOMIC[1][0] DCL HWATOMIC[1][1..3], ARRAY(1)

This declares 5 atomics, one in buffer 0 at 0, one in buffer 1 at 0, and an array of 3 atomics in the buffer 1, starting at 1.

Properties

Properties are general directives that apply to the whole TGSI program.

FS_COORD_ORIGIN

Specifies the fragment shader TGSI_SEMANTIC_POSITION coordinate origin. The default value is UPPER_LEFT.

If UPPER_LEFT, the position will be (0,0) at the upper left corner and increase downward and rightward. If LOWER_LEFT, the position will be (0,0) at the lower left corner and increase upward and rightward.

OpenGL defaults to LOWER_LEFT, and is configurable with the GL_ARB_fragment_coord_conventions extension.

DirectX 9/10 use UPPER_LEFT.

FS_COORD_PIXEL_CENTER

Specifies the fragment shader TGSI_SEMANTIC_POSITION pixel center convention. The default value is HALF_INTEGER.

If HALF_INTEGER, the fractionary part of the position will be 0.5 If INTEGER, the fractionary part of the position will be 0.0

Note that this does not affect the set of fragments generated by rasterization, which is instead controlled by half_pixel_center in the rasterizer.

OpenGL defaults to HALF_INTEGER, and is configurable with the GL_ARB_fragment_coord_conventions extension.

DirectX 9 uses INTEGER. DirectX 10 uses HALF_INTEGER.

FS_COLOR0_WRITES_ALL_CBUFS

Specifies that writes to the fragment shader color 0 are replicated to all bound cbufs. This facilitates OpenGL's fragColor output vs fragData[0] where fragData is directed to a single color buffer, but fragColor is broadcast.

VS_PROHIBIT_UCPS

If this property is set on the program bound to the shader stage before the fragment shader, user clip planes should have no effect (be disabled) even if that shader does not write to any clip distance outputs and the rasterizer's clip_plane_enable is non-zero. This property is only supported by drivers that also support shader clip distance outputs. This is useful for APIs that don't have UCPs and where clip distances written by a shader cannot be disabled.

GS_INVOCATIONS

Specifies the number of times a geometry shader should be executed for each input primitive. Each invocation will have a different TGSI_SEMANTIC_INVOCATIONID system value set. If not specified, assumed to be 1.

VS_WINDOW_SPACE_POSITION

If this property is set on the vertex shader, the TGSI_SEMANTIC_POSITION output is assumed to contain window space coordinates. Division of X,Y,Z by W and the viewport transformation are disabled, and 1/W is directly taken from the 4-th component of the shader output. Naturally, clipping is not performed on window coordinates either. The effect of this property is undefined if a geometry or tessellation shader are in use.

TCS_VERTICES_OUT

The number of vertices written by the tessellation control shader. This effectively defines the patch input size of the tessellation evaluation shader as well.

TES_PRIM_MODE

This sets the tessellation primitive mode, one of `PIPE_PRIM_TRIANGLES`, `PIPE_PRIM_QUADS`, or `PIPE_PRIM_LINES`. (Unlike in GL, there is no separate isolines settings, the regular lines is assumed to mean isolines.)

TES_SPACING

This sets the spacing mode of the tessellation generator, one of `PIPE_TESS_SPACING_*`.

TES_VERTEX_ORDER_CW

This sets the vertex order to be clockwise if the value is 1, or counter-clockwise if set to 0.

TES_POINT_MODE

If set to a non-zero value, this turns on point mode for the tessellator, which means that points will be generated instead of primitives.

NUM_CLIPDIST_ENABLED

How many clip distance scalar outputs are enabled.

NUM_CULLDIST_ENABLED

How many cull distance scalar outputs are enabled.

FS_EARLY_DEPTH_STENCIL

Whether depth test, stencil test, and occlusion query should run before the fragment shader (regardless of fragment shader side effects). Corresponds to GLSL early_fragment_tests.

NEXT_SHADER

Which shader stage will MOST LIKELY follow after this shader when the shader is bound. This is only a hint to the driver and doesn't have to be precise. Only set for VS and TES.

CS_FIXED_BLOCK_WIDTH / HEIGHT / DEPTH

Threads per block in each dimension, if known at compile time. If the block size is known all three should be at least 1. If it is unknown they should all be set to 0 or not set.

MUL_ZERO_WINS

The MUL TGSI operation (FP32 multiplication) will return 0 if either of the operands are equal to 0. That means that 0 * Inf = 0. This should be set the same way for an entire pipeline. Note that this applies not only to the literal MUL TGSI opcode, but all FP32 multiplications implied by other operations, such as MAD, FMA, DP2, DP3, DP4, DST, LOG, LRP, and possibly others. If there is a mismatch between shaders, then it is unspecified whether this behavior will be enabled.

FS_POST_DEPTH_COVERAGE

When enabled, the input for TGSI_SEMANTIC_SAMPLEMASK will exclude samples that have failed the depth/stencil tests. This is only valid when FS_EARLY_DEPTH_STENCIL is also specified.

LAYER_VIEWPORT_RELATIVE

When enabled, the TGSI_SEMATNIC_LAYER output value is relative to the current viewport. This is especially useful in conjunction with TGSI_SEMANTIC_VIEWPORT_MASK.

34.3.5 Texture Sampling and Texture Formats

This table shows how texture image components are returned as (x,y,z,w) tuples by TGSI texture instructions, such as `TEX`, `TXD`, and `TXP`. For reference, OpenGL and Direct3D conventions are shown as well.

Texture Components	Gallium	OpenGL	Direct3D 9
R	(r, 0, 0, 1)	(r, 0, 0, 1)	(r, 1, 1, 1)
RG	(r, g, 0, 1)	(r, g, 0, 1)	(r, g, 1, 1)
RGB	(r, g, b, 1)	(r, g, b, 1)	(r, g, b, 1)
RGBA	(r, g, b, a)	(r, g, b, a)	(r, g, b, a)
A	(0, 0, 0, a)	(0, 0, 0, a)	(0, 0, 0, a)
L	(l, l, l, 1)	(l, l, l, 1)	(l, l, l, 1)
LA	(l, l, l, a)	(l, l, l, a)	(l, l, l, a)
I	(i, i, i, i)	(i, i, i, i)	N/A
UV	XXX TBD	(0, 0, 0, 1)[1]	(u, v, 1, 1)
Z	XXX TBD	(z, z, z, 1)[2]	(0, z, 0, 1)
S	(s, s, s, s)	unknown	unknown

34.4 Screen

A screen is an object representing the context-independent part of a device.

[1] http://www.opengl.org/registry/specs/ATI/envmap_bumpmap.txt
[2] the default is (z, z, z, 1) but may also be (0, 0, 0, z) or (z, z, z, z) depending on the value of GL_DEPTH_TEXTURE_MODE.

34.4.1 Flags and enumerations

XXX some of these don't belong in this section.

PIPE_CAP_*

Capability queries return information about the features and limits of the driver/GPU. For floating-point values, use *get_paramf*, and for boolean or integer values, use *get_param*.

The integer capabilities:

- `PIPE_CAP_GRAPHICS`: Whether graphics is supported. If not, contexts can only be created with PIPE_CONTEXT_COMPUTE_ONLY.
- `PIPE_CAP_NPOT_TEXTURES`: Whether *NPOT* textures may have repeat modes, normalized coordinates, and mipmaps.
- `PIPE_CAP_MAX_DUAL_SOURCE_RENDER_TARGETS`: How many dual-source blend RTs are support. *Blend* for more information.
- `PIPE_CAP_ANISOTROPIC_FILTER`: Whether textures can be filtered anisotropically.
- `PIPE_CAP_POINT_SPRITE`: Whether point sprites are available.
- `PIPE_CAP_MAX_RENDER_TARGETS`: The maximum number of render targets that may be bound.
- `PIPE_CAP_OCCLUSION_QUERY`: Whether occlusion queries are available.
- `PIPE_CAP_QUERY_TIME_ELAPSED`: Whether PIPE_QUERY_TIME_ELAPSED queries are available.
- `PIPE_CAP_TEXTURE_SHADOW_MAP`: indicates whether the fragment shader hardware can do the depth texture / Z comparison operation in TEX instructions for shadow testing.
- `PIPE_CAP_TEXTURE_SWIZZLE`: Whether swizzling through sampler views is supported.
- `PIPE_CAP_MAX_TEXTURE_2D_SIZE`: The maximum size of 2D (and 1D) textures.
- `PIPE_CAP_MAX_TEXTURE_3D_LEVELS`: The maximum number of mipmap levels available for a 3D texture.
- `PIPE_CAP_MAX_TEXTURE_CUBE_LEVELS`: The maximum number of mipmap levels available for a cubemap.
- `PIPE_CAP_TEXTURE_MIRROR_CLAMP_TO_EDGE`: Whether mirrored texture coordinates are supported with the clamp-to-edge wrap mode.
- `PIPE_CAP_TEXTURE_MIRROR_CLAMP`: Whether mirrored texture coordinates are supported with clamp or clamp-to-border wrap modes.
- `PIPE_CAP_BLEND_EQUATION_SEPARATE`: Whether alpha blend equations may be different from color blend equations, in *Blend* state.
- `PIPE_CAP_MAX_STREAM_OUTPUT_BUFFERS`: The maximum number of stream buffers.
- `PIPE_CAP_PRIMITIVE_RESTART`: Whether primitive restart is supported.
- `PIPE_CAP_PRIMITIVE_RESTART_FIXED_INDEX`: Subset of PRIMITIVE_RESTART where the restart index is always the fixed maximum value for the index type.
- `PIPE_CAP_INDEP_BLEND_ENABLE`: Whether per-rendertarget blend enabling and channel masks are supported. If 0, then the first rendertarget's blend mask is replicated across all MRTs.
- `PIPE_CAP_INDEP_BLEND_FUNC`: Whether per-rendertarget blend functions are available. If 0, then the first rendertarget's blend functions affect all MRTs.

- PIPE_CAP_MAX_TEXTURE_ARRAY_LAYERS: The maximum number of texture array layers supported. If 0, the array textures are not supported at all and the ARRAY texture targets are invalid.
- PIPE_CAP_TGSI_FS_COORD_ORIGIN_UPPER_LEFT: Whether the TGSI property FS_COORD_ORIGIN with value UPPER_LEFT is supported.
- PIPE_CAP_TGSI_FS_COORD_ORIGIN_LOWER_LEFT: Whether the TGSI property FS_COORD_ORIGIN with value LOWER_LEFT is supported.
- PIPE_CAP_TGSI_FS_COORD_PIXEL_CENTER_HALF_INTEGER: Whether the TGSI property FS_COORD_PIXEL_CENTER with value HALF_INTEGER is supported.
- PIPE_CAP_TGSI_FS_COORD_PIXEL_CENTER_INTEGER: Whether the TGSI property FS_COORD_PIXEL_CENTER with value INTEGER is supported.
- PIPE_CAP_DEPTH_CLIP_DISABLE: Whether the driver is capable of disabling depth clipping (=1) (through pipe_rasterizer_state) or supports lowering depth_clamp in the client shader code (=2), for this the driver must currently use TGSI.
- PIPE_CAP_DEPTH_CLIP_DISABLE_SEPARATE: Whether the driver is capable of disabling depth clipping (through pipe_rasterizer_state) separately for the near and far plane. If not, depth_clip_near and depth_clip_far will be equal.
- PIPE_CAP_SHADER_STENCIL_EXPORT: Whether a stencil reference value can be written from a fragment shader.
- PIPE_CAP_TGSI_INSTANCEID: Whether TGSI_SEMANTIC_INSTANCEID is supported in the vertex shader.
- PIPE_CAP_VERTEX_ELEMENT_INSTANCE_DIVISOR: Whether the driver supports per-instance vertex attribs.
- PIPE_CAP_FRAGMENT_COLOR_CLAMPED: Whether fragment color clamping is supported. That is, is the pipe_rasterizer_state::clamp_fragment_color flag supported by the driver? If not, gallium frontends will insert clamping code into the fragment shaders when needed.
- PIPE_CAP_MIXED_COLORBUFFER_FORMATS: Whether mixed colorbuffer formats are supported, e.g. RGBA8 and RGBA32F as the first and second colorbuffer, resp.
- PIPE_CAP_VERTEX_COLOR_UNCLAMPED: Whether the driver is capable of outputting unclamped vertex colors from a vertex shader. If unsupported, the vertex colors are always clamped. This is the default for DX9 hardware.
- PIPE_CAP_VERTEX_COLOR_CLAMPED: Whether the driver is capable of clamping vertex colors when they come out of a vertex shader, as specified by the pipe_rasterizer_state::clamp_vertex_color flag. If unsupported, the vertex colors are never clamped. This is the default for DX10 hardware. If both clamped and unclamped CAPs are supported, the clamping can be controlled through pipe_rasterizer_state. If the driver cannot do vertex color clamping, gallium frontends may insert clamping code into the vertex shader.
- PIPE_CAP_GLSL_FEATURE_LEVEL: Whether the driver supports features equivalent to a specific GLSL version. E.g. for GLSL 1.3, report 130.
- PIPE_CAP_GLSL_FEATURE_LEVEL_COMPATIBILITY: Whether the driver supports features equivalent to a specific GLSL version including all legacy OpenGL features only present in the OpenGL compatibility profile. The only legacy features that Gallium drivers must implement are the legacy shader inputs and outputs (colors, texcoords, fog, clipvertex, edgeflag).
- PIPE_CAP_ESSL_FEATURE_LEVEL: An optional cap to allow drivers to report a higher GLSL version for GLES contexts. This is useful when a driver does not support all the required features for a higher GL version, but does support the required features for a higher GLES version. A driver is allowed to return 0 in which case PIPE_CAP_GLSL_FEATURE_LEVEL is used. Note that simply returning the same value as the GLSL

feature level cap is incorrect. For example, GLSL version 3.30 does not require `ARB_gpu_shader5`, but ESSL version 3.20 es does require `EXT_gpu_shader5`

- `PIPE_CAP_QUADS_FOLLOW_PROVOKING_VERTEX_CONVENTION`: Whether quads adhere to the flat-shade_first setting in `pipe_rasterizer_state`.
- `PIPE_CAP_USER_VERTEX_BUFFERS`: Whether the driver supports user vertex buffers. If not, gallium frontends must upload all data which is not in hw resources. If user-space buffers are supported, the driver must also still accept HW resource buffers.
- `PIPE_CAP_VERTEX_BUFFER_OFFSET_4BYTE_ALIGNED_ONLY`: This CAP describes a hw limitation. If true, pipe_vertex_buffer::buffer_offset must always be aligned to 4. If false, there are no restrictions on the offset.
- `PIPE_CAP_VERTEX_BUFFER_STRIDE_4BYTE_ALIGNED_ONLY`: This CAP describes a hw limitation. If true, pipe_vertex_buffer::stride must always be aligned to 4. If false, there are no restrictions on the stride.
- `PIPE_CAP_VERTEX_ELEMENT_SRC_OFFSET_4BYTE_ALIGNED_ONLY`: This CAP describes a hw limitation. If true, pipe_vertex_element::src_offset must always be aligned to 4. If false, there are no restrictions on src_offset.
- `PIPE_CAP_COMPUTE`: Whether the implementation supports the compute entry points defined in pipe_context and pipe_screen.
- `PIPE_CAP_CONSTANT_BUFFER_OFFSET_ALIGNMENT`: Describes the required alignment of pipe_constant_buffer::buffer_offset.
- `PIPE_CAP_START_INSTANCE`: Whether the driver supports pipe_draw_info::start_instance.
- `PIPE_CAP_QUERY_TIMESTAMP`: Whether PIPE_QUERY_TIMESTAMP and the pipe_screen::get_timestamp hook are implemented.
- `PIPE_CAP_TEXTURE_MULTISAMPLE`: Whether all MSAA resources supported for rendering are also supported for texturing.
- `PIPE_CAP_MIN_MAP_BUFFER_ALIGNMENT`: The minimum alignment that should be expected for a pointer returned by transfer_map if the resource is PIPE_BUFFER. In other words, the pointer returned by transfer_map is always aligned to this value.
- `PIPE_CAP_TEXTURE_BUFFER_OFFSET_ALIGNMENT`: Describes the required alignment for pipe_sampler_view::u.buf.offset, in bytes. If a driver does not support offset/size, it should return 0.
- `PIPE_CAP_BUFFER_SAMPLER_VIEW_RGBA_ONLY`: Whether the driver only supports R, RG, RGB and RGBA formats for PIPE_BUFFER sampler views. When this is the case it should be assumed that the swizzle parameters in the sampler view have no effect.
- `PIPE_CAP_TGSI_TEXCOORD`: This CAP describes a hw limitation. If true, the hardware cannot replace arbitrary shader inputs with sprite coordinates and hence the inputs that are desired to be replaceable must be declared with TGSI_SEMANTIC_TEXCOORD instead of TGSI_SEMANTIC_GENERIC. The rasterizer's sprite_coord_enable state therefore also applies to the TEXCOORD semantic. Also, TGSI_SEMANTIC_PCOORD becomes available, which labels a fragment shader input that will always be replaced with sprite coordinates.
- `PIPE_CAP_PREFER_BLIT_BASED_TEXTURE_TRANSFER`: Whether it is preferable to use a blit to implement a texture transfer which needs format conversions and swizzling in gallium frontends. Generally, all hardware drivers with dedicated memory should return 1 and all software rasterizers should return 0.
- `PIPE_CAP_QUERY_PIPELINE_STATISTICS`: Whether PIPE_QUERY_PIPELINE_STATISTICS is supported.
- `PIPE_CAP_TEXTURE_BORDER_COLOR_QUIRK`: Bitmask indicating whether special considerations have to be given to the interaction between the border color in the sampler object and the sampler

view used with it. If PIPE_QUIRK_TEXTURE_BORDER_COLOR_SWIZZLE_R600 is set, the border color may be affected in undefined ways for any kind of permutational swizzle (any swizzle XYZW where X/Y/Z/W are not ZERO, ONE, or R/G/B/A respectively) in the sampler view. If PIPE_QUIRK_TEXTURE_BORDER_COLOR_SWIZZLE_NV50 is set, the border color state should be swizzled manually according to the swizzle in the sampler view it is intended to be used with, or herein undefined results may occur for permutational swizzles.

- `PIPE_CAP_MAX_TEXTURE_BUFFER_SIZE`: The maximum accessible size with a buffer sampler view, in texels.
- `PIPE_CAP_MAX_VIEWPORTS`: The maximum number of viewports (and scissors since they are linked) a driver can support. Returning 0 is equivalent to returning 1 because every driver has to support at least a single viewport/scissor combination.
- `PIPE_CAP_ENDIANNESS`:: The endianness of the device. Either PIPE_ENDIAN_BIG or PIPE_ENDIAN_LITTLE.
- `PIPE_CAP_MIXED_FRAMEBUFFER_SIZES`: Whether it is allowed to have different sizes for fb color/zs attachments. This controls whether ARB_framebuffer_object is provided.
- `PIPE_CAP_TGSI_VS_LAYER_VIEWPORT`: Whether `TGSI_SEMANTIC_LAYER` and `TGSI_SEMANTIC_VIEWPORT_INDEX` are supported as vertex shader outputs. Note that the viewport will only be used if multiple viewports are exposed.
- `PIPE_CAP_MAX_GEOMETRY_OUTPUT_VERTICES`: The maximum number of vertices output by a single invocation of a geometry shader.
- `PIPE_CAP_MAX_GEOMETRY_TOTAL_OUTPUT_COMPONENTS`: The maximum number of vertex components output by a single invocation of a geometry shader. This is the product of the number of attribute components per vertex and the number of output vertices.
- `PIPE_CAP_MAX_TEXTURE_GATHER_COMPONENTS`: Max number of components in format that texture gather can operate on. 1 == RED, ALPHA etc, 4 == All formats.
- `PIPE_CAP_TEXTURE_GATHER_SM5`: Whether the texture gather hardware implements the SM5 features, component selection, shadow comparison, and run-time offsets.
- `PIPE_CAP_BUFFER_MAP_PERSISTENT_COHERENT`: Whether PIPE_MAP_PERSISTENT and PIPE_MAP_COHERENT are supported for buffers.
- `PIPE_CAP_TEXTURE_QUERY_LOD`: Whether the `LODQ` instruction is supported.
- `PIPE_CAP_MIN_TEXTURE_GATHER_OFFSET`: The minimum offset that can be used in conjunction with a texture gather opcode.
- `PIPE_CAP_MAX_TEXTURE_GATHER_OFFSET`: The maximum offset that can be used in conjunction with a texture gather opcode.
- `PIPE_CAP_SAMPLE_SHADING`: Whether there is support for per-sample shading. The context->set_min_samples function will be expected to be implemented.
- `PIPE_CAP_TEXTURE_GATHER_OFFSETS`: Whether the `TG4` instruction can accept 4 offsets.
- `PIPE_CAP_TGSI_VS_WINDOW_SPACE_POSITION`: Whether TGSI_PROPERTY_VS_WINDOW_SPACE_POSITION is supported, which disables clipping and viewport transformation.
- `PIPE_CAP_MAX_VERTEX_STREAMS`: The maximum number of vertex streams supported by the geometry shader. If stream-out is supported, this should be at least 1. If stream-out is not supported, this should be 0.
- `PIPE_CAP_DRAW_INDIRECT`: Whether the driver supports taking draw arguments { count, instance_count, start, index_bias } from a PIPE_BUFFER resource. See pipe_draw_info.

- PIPE_CAP_MULTI_DRAW_INDIRECT: Whether the driver supports pipe_draw_info::indirect_stride and ::indirect_count
- PIPE_CAP_MULTI_DRAW_INDIRECT_PARAMS: Whether the driver supports taking the number of indirect draws from a separate parameter buffer, see pipe_draw_indirect_info::indirect_draw_count.
- PIPE_CAP_TGSI_FS_FINE_DERIVATIVE: Whether the fragment shader supports the FINE versions of DDX/DDY.
- PIPE_CAP_VENDOR_ID: The vendor ID of the underlying hardware. If it's not available one should return 0xFFFFFFFF.
- PIPE_CAP_DEVICE_ID: The device ID (PCI ID) of the underlying hardware. 0xFFFFFFFF if not available.
- PIPE_CAP_ACCELERATED: Whether the renderer is hardware accelerated.
- PIPE_CAP_VIDEO_MEMORY: The amount of video memory in megabytes.
- PIPE_CAP_UMA: If the device has a unified memory architecture or on-card memory and GART.
- PIPE_CAP_CONDITIONAL_RENDER_INVERTED: Whether the driver supports inverted condition for conditional rendering.
- PIPE_CAP_MAX_VERTEX_ATTRIB_STRIDE: The maximum supported vertex stride.
- PIPE_CAP_SAMPLER_VIEW_TARGET: Whether the sampler view's target can be different than the underlying resource's, as permitted by ARB_texture_view. For example a 2d array texture may be reinterpreted as a cube (array) texture and vice-versa.
- PIPE_CAP_CLIP_HALFZ: Whether the driver supports the pipe_rasterizer_state::clip_halfz being set to true. This is required for enabling ARB_clip_control.
- PIPE_CAP_VERTEXID_NOBASE: If true, the driver only supports TGSI_SEMANTIC_VERTEXID_NOBASE (and not TGSI_SEMANTIC_VERTEXID). This means gallium frontends for APIs whose vertexIDs are offset by basevertex (such as GL) will need to lower TGSI_SEMANTIC_VERTEXID to TGSI_SEMANTIC_VERTEXID_NOBASE and TGSI_SEMANTIC_BASEVERTEX, so drivers setting this must handle both these semantics. Only relevant if geometry shaders are supported. (BASEVERTEX could be exposed separately too via PIPE_CAP_DRAW_PARAMETERS).
- PIPE_CAP_POLYGON_OFFSET_CLAMP: If true, the driver implements support for pipe_rasterizer_state::offset_clamp.
- PIPE_CAP_MULTISAMPLE_Z_RESOLVE: Whether the driver supports blitting a multisampled depth buffer into a single-sampled texture (or depth buffer). Only the first sampled should be copied.
- PIPE_CAP_RESOURCE_FROM_USER_MEMORY: Whether the driver can create a pipe_resource where an already-existing piece of (malloc'd) user memory is used as its backing storage. In other words, whether the driver can map existing user memory into the device address space for direct device access. The create function is pipe_screen::resource_from_user_memory. The address and size must be page-aligned.
- PIPE_CAP_RESOURCE_FROM_USER_MEMORY_COMPUTE_ONLY: Same as PIPE_CAP_RESOURCE_FROM_USER_MEMORY but indicates it is only supported from the compute engines.
- PIPE_CAP_DEVICE_RESET_STATUS_QUERY: Whether pipe_context::get_device_reset_status is implemented.
- PIPE_CAP_MAX_SHADER_PATCH_VARYINGS: How many per-patch outputs and inputs are supported between tessellation control and tessellation evaluation shaders, not counting in TESSINNER and TESSOUTER. The minimum allowed value for OpenGL is 30.

- PIPE_CAP_TEXTURE_FLOAT_LINEAR: Whether the linear minification and magnification filters are supported with single-precision floating-point textures.
- PIPE_CAP_TEXTURE_HALF_FLOAT_LINEAR: Whether the linear minification and magnification filters are supported with half-precision floating-point textures.
- PIPE_CAP_DEPTH_BOUNDS_TEST: Whether bounds_test, bounds_min, and bounds_max states of pipe_depth_stencil_alpha_state behave according to the GL_EXT_depth_bounds_test specification.
- PIPE_CAP_TGSI_TXQS: Whether the *TXQS* opcode is supported
- PIPE_CAP_FORCE_PERSAMPLE_INTERP: If the driver can force per-sample interpolation for all fragment shader inputs if pipe_rasterizer_state::force_persample_interp is set. This is only used by GL3-level sample shading (ARB_sample_shading). GL4-level sample shading (ARB_gpu_shader5) doesn't use this. While GL3 hardware has a state for it, GL4 hardware will likely need to emulate it with a shader variant, or by selecting the interpolation weights with a conditional assignment in the shader.
- PIPE_CAP_SHAREABLE_SHADERS: Whether shader CSOs can be used by any pipe_context.
- PIPE_CAP_COPY_BETWEEN_COMPRESSED_AND_PLAIN_FORMATS: Whether copying between compressed and plain formats is supported where a compressed block is copied to/from a plain pixel of the same size.
- PIPE_CAP_CLEAR_TEXTURE: Whether *clear_texture* will be available in contexts.
- PIPE_CAP_CLEAR_SCISSORED: Whether *clear* can accept a scissored bounding box.
- PIPE_CAP_DRAW_PARAMETERS: Whether TGSI_SEMANTIC_BASEVERTEX, TGSI_SEMANTIC_BASEINSTANCE, and TGSI_SEMANTIC_DRAWID are supported in vertex shaders.
- PIPE_CAP_TGSI_PACK_HALF_FLOAT: Whether the UP2H and PK2H TGSI opcodes are supported.
- PIPE_CAP_TGSI_FS_POSITION_IS_SYSVAL: If gallium frontends should use a system value for the POSITION fragment shader input.
- PIPE_CAP_TGSI_FS_POINT_IS_SYSVAL: If gallium frontends should use a system value for the POINT fragment shader input.
- PIPE_CAP_TGSI_FS_FACE_IS_INTEGER_SYSVAL: If gallium frontends should use a system value for the FACE fragment shader input. Also, the FACE system value is integer, not float.
- PIPE_CAP_SHADER_BUFFER_OFFSET_ALIGNMENT: Describes the required alignment for pipe_shader_buffer::buffer_offset, in bytes. Maximum value allowed is 256 (for GL conformance). 0 is only allowed if shader buffers are not supported.
- PIPE_CAP_INVALIDATE_BUFFER: Whether the use of invalidate_resource for buffers is supported.
- PIPE_CAP_GENERATE_MIPMAP: Indicates whether pipe_context::generate_mipmap is supported.
- PIPE_CAP_STRING_MARKER: Whether pipe->emit_string_marker() is supported.
- PIPE_CAP_SURFACE_REINTERPRET_BLOCKS: Indicates whether pipe_context::create_surface supports reinterpreting a texture as a surface of a format with different block width/height (but same block size in bits). For example, a compressed texture image can be interpreted as a non-compressed surface whose texels are the same number of bits as the compressed blocks, and vice versa. The width and height of the surface is adjusted appropriately.
- PIPE_CAP_QUERY_BUFFER_OBJECT: Driver supports context::get_query_result_resource callback.
- PIPE_CAP_PCI_GROUP: Return the PCI segment group number.
- PIPE_CAP_PCI_BUS: Return the PCI bus number.
- PIPE_CAP_PCI_DEVICE: Return the PCI device number.

- PIPE_CAP_PCI_FUNCTION: Return the PCI function number.
- PIPE_CAP_FRAMEBUFFER_NO_ATTACHMENT: If non-zero, rendering to framebuffers with no surface attachments is supported. The context->is_format_supported function will be expected to be implemented with PIPE_FORMAT_NONE yeilding the MSAA modes the hardware supports. N.B., The maximum number of layers supported for rasterizing a primitive on a layer is obtained from PIPE_CAP_MAX_TEXTURE_ARRAY_LAYERS even though it can be larger than the number of layers supported by either rendering or textures.
- PIPE_CAP_ROBUST_BUFFER_ACCESS_BEHAVIOR: Implementation uses bounds checking on resource accesses by shader if the context is created with PIPE_CONTEXT_ROBUST_BUFFER_ACCESS. See the ARB_robust_buffer_access_behavior extension for information on the required behavior for out of bounds accesses and accesses to unbound resources.
- PIPE_CAP_CULL_DISTANCE: Whether the driver supports the arb_cull_distance extension and thus implements proper support for culling planes.
- PIPE_CAP_PRIMITIVE_RESTART_FOR_PATCHES: Whether primitive restart is supported for patch primitives.
- PIPE_CAP_TGSI_VOTE: Whether the VOTE_* ops can be used in shaders.
- PIPE_CAP_MAX_WINDOW_RECTANGLES: The maxium number of window rectangles supported in set_window_rectangles.
- PIPE_CAP_POLYGON_OFFSET_UNITS_UNSCALED: If true, the driver implements support for pipe_rasterizer_state::offset_units_unscaled.
- PIPE_CAP_VIEWPORT_SUBPIXEL_BITS: Number of bits of subpixel precision for floating point viewport bounds.
- PIPE_CAP_RASTERIZER_SUBPIXEL_BITS: Number of bits of subpixel precision used by the rasterizer.
- PIPE_CAP_MIXED_COLOR_DEPTH_BITS: Whether there is non-fallback support for color/depth format combinations that use a different number of bits. For the purpose of this cap, Z24 is treated as 32-bit. If set to off, that means that a B5G6R5 + Z24 or RGBA8 + Z16 combination will require a driver fallback, and should not be advertised in the GLX/EGL config list.
- PIPE_CAP_TGSI_ARRAY_COMPONENTS: If true, the driver interprets the UsageMask of input and output declarations and allows declaring arrays in overlapping ranges. The components must be a contiguous range, e.g. a UsageMask of xy or yzw is allowed, but xz or yw isn't. Declarations with overlapping locations must have matching semantic names and indices, and equal interpolation qualifiers. Components may overlap, notably when the gaps in an array of dvec3 are filled in.
- PIPE_CAP_STREAM_OUTPUT_INTERLEAVE_BUFFERS: Whether interleaved stream output mode is able to interleave across buffers. This is required for ARB_transform_feedback3.
- PIPE_CAP_TGSI_CAN_READ_OUTPUTS: Whether every TGSI shader stage can read from the output file.
- PIPE_CAP_GLSL_OPTIMIZE_CONSERVATIVELY: Tell the GLSL compiler to use the minimum amount of optimizations just to be able to do all the linking and lowering.
- PIPE_CAP_FBFETCH: The number of render targets whose value in the current framebuffer can be read in the shader. 0 means framebuffer fetch is not supported. 1 means that only the first render target can be read, and a larger value would mean that multiple render targets are supported.
- PIPE_CAP_FBFETCH_COHERENT: Whether framebuffer fetches from the fragment shader can be guaranteed to be coherent with framebuffer writes.
- PIPE_CAP_TGSI_MUL_ZERO_WINS: Whether TGSI shaders support the TGSI_PROPERTY_MUL_ZERO_WINS shader property.
- PIPE_CAP_DOUBLES: Whether double precision floating-point operations are supported.

- PIPE_CAP_INT64: Whether 64-bit integer operations are supported.
- PIPE_CAP_INT64_DIVMOD: Whether 64-bit integer division/modulo operations are supported.
- PIPE_CAP_TGSI_TEX_TXF_LZ: Whether TEX_LZ and TXF_LZ opcodes are supported.
- PIPE_CAP_TGSI_CLOCK: Whether the CLOCK opcode is supported.
- PIPE_CAP_POLYGON_MODE_FILL_RECTANGLE: Whether the PIPE_POLYGON_MODE_FILL_RECTANGLE mode is supported for pipe_rasterizer_state::fill_front and pipe_rasterizer_state::fill_back.
- PIPE_CAP_SPARSE_BUFFER_PAGE_SIZE: The page size of sparse buffers in bytes, or 0 if sparse buffers are not supported. The page size must be at most 64KB.
- PIPE_CAP_TGSI_BALLOT: Whether the BALLOT and READ_* opcodes as well as the SUBGROUP_* semantics are supported.
- PIPE_CAP_TGSI_TES_LAYER_VIEWPORT: Whether TGSI_SEMANTIC_LAYER and TGSI_SEMANTIC_VIEWPORT_INDEX are supported as tessellation evaluation shader outputs.
- PIPE_CAP_CAN_BIND_CONST_BUFFER_AS_VERTEX: Whether a buffer with just PIPE_BIND_CONSTANT_BUFFER can be legally passed to set_vertex_buffers.
- PIPE_CAP_ALLOW_MAPPED_BUFFERS_DURING_EXECUTION: As the name says.
- PIPE_CAP_POST_DEPTH_COVERAGE: whether TGSI_PROPERTY_FS_POST_DEPTH_COVERAGE is supported.
- PIPE_CAP_BINDLESS_TEXTURE: Whether bindless texture operations are supported.
- PIPE_CAP_NIR_SAMPLERS_AS_DEREF: Whether NIR tex instructions should reference texture and sampler as NIR derefs instead of by indices.
- PIPE_CAP_QUERY_SO_OVERFLOW: Whether the PIPE_QUERY_SO_OVERFLOW_PREDICATE and PIPE_QUERY_SO_OVERFLOW_ANY_PREDICATE query types are supported. Note that for a driver that does not support multiple output streams (i.e., PIPE_CAP_MAX_VERTEX_STREAMS is 1), both query types are identical.
- PIPE_CAP_MEMOBJ: Whether operations on memory objects are supported.
- PIPE_CAP_LOAD_CONSTBUF: True if the driver supports TGSI_OPCODE_LOAD use with constant buffers.
- PIPE_CAP_TGSI_ANY_REG_AS_ADDRESS: Any TGSI register can be used as an address for indirect register indexing.
- PIPE_CAP_TILE_RASTER_ORDER: Whether the driver supports GL_MESA_tile_raster_order, using the tile_raster_order_* fields in pipe_rasterizer_state.
- PIPE_CAP_MAX_COMBINED_SHADER_OUTPUT_RESOURCES: Limit on combined shader output resources (images + buffers + fragment outputs). If 0 the state tracker works it out.
- PIPE_CAP_FRAMEBUFFER_MSAA_CONSTRAINTS: This determines limitations on the number of samples that framebuffer attachments can have. Possible values:

  0. color.nr_samples == zs.nr_samples == color.nr_storage_samples (standard MSAA quality)
  1. color.nr_samples >= zs.nr_samples == color.nr_storage_samples (enhanced MSAA quality)
  2. color.nr_samples >= zs.nr_samples >= color.nr_storage_samples (full flexibility in tuning MSAA quality and performance)

  All color attachments must have the same number of samples and the same number of storage samples.
- PIPE_CAP_SIGNED_VERTEX_BUFFER_OFFSET: Whether pipe_vertex_buffer::buffer_offset is treated as signed. The u_vbuf module needs this for optimal performance in workstation applications.

- PIPE_CAP_CONTEXT_PRIORITY_MASK: For drivers that support per-context priorities, this returns a bitmask of PIPE_CONTEXT_PRIORITY_x for the supported priority levels. A driver that does not support prioritized contexts can return 0.
- PIPE_CAP_FENCE_SIGNAL: True if the driver supports signaling semaphores using fence_server_signal().
- PIPE_CAP_CONSTBUF0_FLAGS: The bits of pipe_resource::flags that must be set when binding that buffer as constant buffer 0. If the buffer doesn't have those bits set, pipe_context::set_constant_buffer(.., 0, ..) is ignored by the driver, and the driver can throw assertion failures.
- PIPE_CAP_PACKED_UNIFORMS: True if the driver supports packed uniforms as opposed to padding to vec4s.
- PIPE_CAP_CONSERVATIVE_RASTER_POST_SNAP_TRIANGLES: Whether the PIPE_CONSERVATIVE_RASTER_POST_SNAP mode is supported for triangles. The post-snap mode means the conservative rasterization occurs after the conversion from floating-point to fixed-point coordinates on the subpixel grid.
- PIPE_CAP_CONSERVATIVE_RASTER_POST_SNAP_POINTS_LINES: Whether the PIPE_CONSERVATIVE_RASTER_POST_SNAP mode is supported for points and lines.
- PIPE_CAP_CONSERVATIVE_RASTER_PRE_SNAP_TRIANGLES: Whether the PIPE_CONSERVATIVE_RASTER_PRE_SNAP mode is supported for triangles. The pre-snap mode means the conservative rasterization occurs before the conversion from floating-point to fixed-point coordinates.
- PIPE_CAP_CONSERVATIVE_RASTER_PRE_SNAP_POINTS_LINES: Whether the PIPE_CONSERVATIVE_RASTER_PRE_SNAP mode is supported for points and lines.
- PIPE_CAP_CONSERVATIVE_RASTER_POST_DEPTH_COVERAGE: Whether PIPE_CAP_POST_DEPTH_COVERAGE works with conservative rasterization.
- PIPE_CAP_CONSERVATIVE_RASTER_INNER_COVERAGE: Whether inner_coverage from GL_INTEL_conservative_rasterization is supported.
- PIPE_CAP_MAX_CONSERVATIVE_RASTER_SUBPIXEL_PRECISION_BIAS: The maximum subpixel precision bias in bits during conservative rasterization.
- PIPE_CAP_PROGRAMMABLE_SAMPLE_LOCATIONS: True is the driver supports programmable sample location through `get_sample_pixel_grid` and `set_sample_locations`.
- PIPE_CAP_MAX_GS_INVOCATIONS: Maximum supported value of TGSI_PROPERTY_GS_INVOCATIONS.
- PIPE_CAP_MAX_SHADER_BUFFER_SIZE: Maximum supported size for binding with set_shader_buffers.
- PIPE_CAP_MAX_COMBINED_SHADER_BUFFERS: Maximum total number of shader buffers. A value of 0 means the sum of all per-shader stage maximums (see PIPE_SHADER_CAP_MAX_SHADER_BUFFERS).
- PIPE_CAP_MAX_COMBINED_HW_ATOMIC_COUNTERS: Maximum total number of atomic counters. A value of 0 means the default value (MAX_ATOMIC_COUNTERS = 4096).
- PIPE_CAP_MAX_COMBINED_HW_ATOMIC_COUNTER_BUFFERS: Maximum total number of atomic counter buffers. A value of 0 means the sum of all per-shader stage maximums (see PIPE_SHADER_CAP_MAX_HW_ATOMIC_COUNTER_BUFFERS).
- PIPE_CAP_MAX_TEXTURE_UPLOAD_MEMORY_BUDGET: Maximum recommend memory size for all active texture uploads combined. This is a performance hint. 0 means no limit.
- PIPE_CAP_MAX_VERTEX_ELEMENT_SRC_OFFSET: The maximum supported value for of pipe_vertex_element::src_offset.
- PIPE_CAP_SURFACE_SAMPLE_COUNT: Whether the driver supports pipe_surface overrides of resource nr_samples. If set, will enable EXT_multisampled_render_to_texture.
- PIPE_CAP_TGSI_ATOMFADD: Atomic floating point adds are supported on images, buffers, and shared memory.

- PIPE_CAP_RGB_OVERRIDE_DST_ALPHA_BLEND: True if the driver needs blend state to use zero/one instead of destination alpha for RGB/XRGB formats.
- PIPE_CAP_GLSL_TESS_LEVELS_AS_INPUTS: True if the driver wants TESSINNER and TESSOUTER to be inputs (rather than system values) for tessellation evaluation shaders.
- PIPE_CAP_DEST_SURFACE_SRGB_CONTROL: Indicates whether the drivers supports switching the format between sRGB and linear for a surface that is used as destination in draw and blit calls.
- PIPE_CAP_NIR_COMPACT_ARRAYS: True if the compiler backend supports NIR's compact array feature, for all shader stages.
- PIPE_CAP_MAX_VARYINGS: The maximum number of fragment shader varyings. This will generally correspond to PIPE_SHADER_CAP_MAX_INPUTS for the fragment shader, but in some cases may be a smaller number.
- PIPE_CAP_COMPUTE_GRID_INFO_LAST_BLOCK: Whether pipe_grid_info::last_block is implemented by the driver. See struct pipe_grid_info for more details.
- PIPE_CAP_COMPUTE_SHADER_DERIVATIVE: True if the driver supports derivatives (and texture lookups with implicit derivatives) in compute shaders.
- PIPE_CAP_TGSI_SKIP_SHRINK_IO_ARRAYS: Whether the TGSI pass to shrink IO arrays should be skipped and enforce keeping the declared array sizes instead. A driver might rely on the input mapping that was defined with the original GLSL code.
- PIPE_CAP_IMAGE_LOAD_FORMATTED: True if a format for image loads does not need to be specified in the shader IR
- PIPE_CAP_THROTTLE: Whether or not gallium frontends should throttle pipe_context execution. 0 = throttling is disabled.
- PIPE_CAP_DMABUF: Whether Linux DMABUF handles are supported by resource_from_handle and resource_get_handle.
- PIPE_CAP_PREFER_COMPUTE_FOR_MULTIMEDIA: Whether VDPAU, VAAPI, and OpenMAX should use a compute-based blit instead of pipe_context::blit and compute pipeline for compositing images.
- PIPE_CAP_FRAGMENT_SHADER_INTERLOCK: True if fragment shader interlock functionality is supported.
- PIPE_CAP_CS_DERIVED_SYSTEM_VALUES_SUPPORTED: True if driver handles gl_LocalInvocationIndex and gl_GlobalInvocationID. Otherwise, gallium frontends will lower those system values.
- PIPE_CAP_ATOMIC_FLOAT_MINMAX: Atomic float point minimum, maximum, exchange and compare-and-swap support to buffer and shared variables.
- PIPE_CAP_TGSI_DIV: Whether opcode DIV is supported
- PIPE_CAP_FRAGMENT_SHADER_TEXTURE_LOD: Whether texture lookups with explicit LOD is supported in the fragment shader.
- PIPE_CAP_FRAGMENT_SHADER_DERIVATIVES: True if the driver supports derivatives in fragment shaders.
- PIPE_CAP_VERTEX_SHADER_SATURATE: True if the driver supports saturate modifiers in the vertex shader.
- PIPE_CAP_TEXTURE_SHADOW_LOD: True if the driver supports shadow sampler types with texture functions having interaction with LOD of texture lookup.
- PIPE_CAP_SHADER_SAMPLES_IDENTICAL: True if the driver supports a shader query to tell whether all samples of a multisampled surface are definitely identical.

- PIPE_CAP_TGSI_ATOMINC_WRAP: Atomic increment/decrement + wrap around are supported.
- PIPE_CAP_PREFER_IMM_ARRAYS_AS_CONSTBUF: True if gallium frontends should turn arrays whose contents can be deduced at compile time into constant buffer loads, or false if the driver can handle such arrays itself in a more efficient manner.
- PIPE_CAP_GL_SPIRV: True if the driver supports ARB_gl_spirv extension.
- PIPE_CAP_GL_SPIRV_VARIABLE_POINTERS: True if the driver supports Variable Pointers in SPIR-V shaders.
- PIPE_CAP_DEMOTE_TO_HELPER_INVOCATION: True if driver supports demote keyword in GLSL programs.
- PIPE_CAP_TGSI_TG4_COMPONENT_IN_SWIZZLE: True if driver wants the TG4 component encoded in sampler swizzle rather than as a separate source.
- PIPE_CAP_FLATSHADE: Driver supports pipe_rasterizer_state::flatshade.
- PIPE_CAP_ALPHA_TEST: Driver supports alpha-testing.
- PIPE_CAP_POINT_SIZE_FIXED: Driver supports point-sizes that are fixed, as opposed to writing gl_PointSize for every point.
- PIPE_CAP_TWO_SIDED_COLOR: Driver supports two-sided coloring.
- PIPE_CAP_CLIP_PLANES: Driver supports user-defined clip-planes.
- PIPE_CAP_MAX_VERTEX_BUFFERS: Number of supported vertex buffers.
- PIPE_CAP_OPENCL_INTEGER_FUNCTIONS: Driver supports extended OpenCL-style integer functions. This includes averge, saturating additiong, saturating subtraction, absolute difference, count leading zeros, and count trailing zeros.
- PIPE_CAP_INTEGER_MULTIPLY_32X16: Driver supports integer multiplication between a 32-bit integer and a 16-bit integer. If the second operand is 32-bits, the upper 16-bits are ignored, and the low 16-bits are possibly sign extended as necessary.
- PIPE_CAP_NIR_IMAGES_AS_DEREF: Whether NIR image load/store intrinsics should be nir_intrinsic_image_deref_* instead of nir_intrinsic_image_*. Defaults to true.
- PIPE_CAP_PACKED_STREAM_OUTPUT: Driver supports packing optimization for stream output (e.g. GL transform feedback captured variables). Defaults to true.
- PIPE_CAP_VIEWPORT_TRANSFORM_LOWERED: Driver needs the nir_lower_viewport_transform pass to be enabled. This also means that the gl_Position value is modified and should be lowered for transform feedback, if needed. Defaults to false.
- PIPE_CAP_PSIZ_CLAMPED: Driver needs for the point size to be clamped. Additionally, the gl_PointSize has been modified and its value should be lowered for transform feedback, if needed. Defaults to false.
- PIPE_CAP_DRAW_INFO_START_WITH_USER_INDICES: pipe_draw_info::start can be non-zero with user indices.
- PIPE_CAP_GL_BEGIN_END_BUFFER_SIZE: Buffer size used to upload vertices for glBegin/glEnd.
- PIPE_CAP_VIEWPORT_SWIZZLE: Whether pipe_viewport_state::swizzle can be used to specify pre-clipping swizzling of coordinates (see GL_NV_viewport_swizzle).
- PIPE_CAP_SYSTEM_SVM: True if all application memory can be shared with the GPU without explicit mapping.
- PIPE_CAP_VIEWPORT_MASK: Whether TGSI_SEMANTIC_VIEWPORT_MASK and TGSI_PROPERTY_LAYER_VIEWPORT_RELATIVE are supported (see GL_NV_viewport_array2).

- PIPE_CAP_MAP_UNSYNCHRONIZED_THREAD_SAFE: Whether mapping a buffer as unsynchronized from any thread is safe.
- PIPE_CAP_GLSL_ZERO_INIT: Choose a default zero initialization some glsl variables. If *1*, then all glsl shader variables and gl_FragColor are initialized to zero. If *2*, then shader out variables are not initialized but function out variables are.
- PIPE_CAP_BLEND_EQUATION_ADVANCED: Driver supports blend equation advanced without necessarily supporting FBFETCH.
- PIPE_CAP_NIR_ATOMICS_AS_DEREF: Whether NIR atomics instructions should reference atomics as NIR derefs instead of by indices.
- PIPE_CAP_NO_CLIP_ON_COPY_TEX: Driver doesn't want x/y/width/height clipped based on src size when doing a copy texture operation (eg: may want out-of-bounds reads that produce 0 instead of leaving the texture content undefined)
- PIPE_CAP_MAX_TEXTURE_MB: Maximum texture size in MB (default is 1024)

PIPE_CAPF_*

The floating-point capabilities are:

- PIPE_CAPF_MAX_LINE_WIDTH: The maximum width of a regular line.
- PIPE_CAPF_MAX_LINE_WIDTH_AA: The maximum width of a smoothed line.
- PIPE_CAPF_MAX_POINT_WIDTH: The maximum width and height of a point.
- PIPE_CAPF_MAX_POINT_WIDTH_AA: The maximum width and height of a smoothed point.
- PIPE_CAPF_MAX_TEXTURE_ANISOTROPY: The maximum level of anisotropy that can be applied to anisotropically filtered textures.
- PIPE_CAPF_MAX_TEXTURE_LOD_BIAS: The maximum *LOD* bias that may be applied to filtered textures.
- PIPE_CAPF_MIN_CONSERVATIVE_RASTER_DILATE: The minimum conservative rasterization dilation.
- PIPE_CAPF_MAX_CONSERVATIVE_RASTER_DILATE: The maximum conservative rasterization dilation.
- PIPE_CAPF_CONSERVATIVE_RASTER_DILATE_GRANULARITY: The conservative rasterization dilation granularity for values relative to the minimum dilation.

PIPE_SHADER_CAP_*

These are per-shader-stage capabitity queries. Different shader stages may support different features.

- PIPE_SHADER_CAP_MAX_INSTRUCTIONS: The maximum number of instructions.
- PIPE_SHADER_CAP_MAX_ALU_INSTRUCTIONS: The maximum number of arithmetic instructions.
- PIPE_SHADER_CAP_MAX_TEX_INSTRUCTIONS: The maximum number of texture instructions.
- PIPE_SHADER_CAP_MAX_TEX_INDIRECTIONS: The maximum number of texture indirections.
- PIPE_SHADER_CAP_MAX_CONTROL_FLOW_DEPTH: The maximum nested control flow depth.
- PIPE_SHADER_CAP_MAX_INPUTS: The maximum number of input registers.
- PIPE_SHADER_CAP_MAX_OUTPUTS: The maximum number of output registers. This is valid for all shaders except the fragment shader.
- PIPE_SHADER_CAP_MAX_CONST_BUFFER_SIZE: The maximum size per constant buffer in bytes.

- PIPE_SHADER_CAP_MAX_CONST_BUFFERS: Maximum number of constant buffers that can be bound to any shader stage using set_constant_buffer. If 0 or 1, the pipe will only permit binding one constant buffer per shader.

If a value greater than 0 is returned, the driver can have multiple constant buffers bound to shader stages. The CONST register file is accessed with two-dimensional indices, like in the example below.

DCL CONST[0][0..7] # declare first 8 vectors of constbuf 0 DCL CONST[3][0] # declare first vector of constbuf 3 MOV OUT[0], CONST[0][3] # copy vector 3 of constbuf 0

- PIPE_SHADER_CAP_MAX_TEMPS: The maximum number of temporary registers.
- PIPE_SHADER_CAP_TGSI_CONT_SUPPORTED: Whether the continue opcode is supported.
- PIPE_SHADER_CAP_INDIRECT_INPUT_ADDR: Whether indirect addressing of the input file is supported.
- PIPE_SHADER_CAP_INDIRECT_OUTPUT_ADDR: Whether indirect addressing of the output file is supported.
- PIPE_SHADER_CAP_INDIRECT_TEMP_ADDR: Whether indirect addressing of the temporary file is supported.
- PIPE_SHADER_CAP_INDIRECT_CONST_ADDR: Whether indirect addressing of the constant file is supported.
- PIPE_SHADER_CAP_SUBROUTINES: Whether subroutines are supported, i.e. BGNSUB, ENDSUB, CAL, and RET, including RET in the main block.
- PIPE_SHADER_CAP_INTEGERS: Whether integer opcodes are supported. If unsupported, only float opcodes are supported.
- PIPE_SHADER_CAP_INT64_ATOMICS: Whether int64 atomic opcodes are supported. The device needs to support add, sub, swap, cmpswap, and, or, xor, min, and max.
- **PIPE_SHADER_CAP_FP16: Whether half precision floating-point opcodes are supported.** If unsupported, half precision ops need to be lowered to full precision.
- PIPE_SHADER_CAP_FP16_DERIVATIVES: Whether half precision floating-point DDX and DDY opcodes are supported.
- PIPE_SHADER_CAP_INT16: Whether 16-bit signed and unsigned integer types are supported.
- PIPE_SHADER_CAP_GLSL_16BIT_CONSTS: Lower mediump constants to 16-bit. Note that 16-bit constants are not lowered to uniforms in GLSL.
- PIPE_SHADER_CAP_MAX_TEXTURE_SAMPLERS: The maximum number of texture samplers.
- PIPE_SHADER_CAP_PREFERRED_IR: Preferred representation of the program. It should be one of the pipe_shader_ir enum values.
- PIPE_SHADER_CAP_MAX_SAMPLER_VIEWS: The maximum number of texture sampler views. Must not be lower than PIPE_SHADER_CAP_MAX_TEXTURE_SAMPLERS.
- PIPE_SHADER_CAP_TGSI_DROUND_SUPPORTED: Whether double precision rounding is supported. If it is, DTRUNC/DCEIL/DFLR/DROUND opcodes may be used.
- PIPE_SHADER_CAP_TGSI_DFRACEXP_DLDEXP_SUPPORTED: Whether DFRACEXP and DLDEXP are supported.
- PIPE_SHADER_CAP_TGSI_LDEXP_SUPPORTED: Whether LDEXP is supported.
- PIPE_SHADER_CAP_TGSI_FMA_SUPPORTED: Whether FMA and DFMA (doubles only) are supported.
- PIPE_SHADER_CAP_TGSI_ANY_INOUT_DECL_RANGE: Whether the driver doesn't ignore tgsi_declaration_range::Last for shader inputs and outputs.

- PIPE_SHADER_CAP_MAX_UNROLL_ITERATIONS_HINT: This is the maximum number of iterations that loops are allowed to have to be unrolled. It is only a hint to gallium frontends. Whether any loops will be unrolled is not guaranteed.
- PIPE_SHADER_CAP_MAX_SHADER_BUFFERS: Maximum number of memory buffers (also used to implement atomic counters). Having this be non-0 also implies support for the LOAD, STORE, and ATOM* TGSI opcodes.
- PIPE_SHADER_CAP_SUPPORTED_IRS: Supported representations of the program. It should be a mask of pipe_shader_ir bits.
- PIPE_SHADER_CAP_MAX_SHADER_IMAGES: Maximum number of image units.
- PIPE_SHADER_CAP_LOWER_IF_THRESHOLD: IF and ELSE branches with a lower cost than this value should be lowered by gallium frontends for better performance. This is a tunable for the GLSL compiler and the behavior is specific to the compiler.
- PIPE_SHADER_CAP_TGSI_SKIP_MERGE_REGISTERS: Whether the merge registers TGSI pass is skipped. This might reduce code size and register pressure if the underlying driver has a real backend compiler.
- PIPE_SHADER_CAP_MAX_HW_ATOMIC_COUNTERS: If atomic counters are separate, how many HW counters are available for this stage. (0 uses SSBO atomics).
- PIPE_SHADER_CAP_MAX_HW_ATOMIC_COUNTER_BUFFERS: If atomic counters are separate, how many atomic counter buffers are available for this stage.

PIPE_COMPUTE_CAP_*

Compute-specific capabilities. They can be queried using pipe_screen::get_compute_param.

- PIPE_COMPUTE_CAP_IR_TARGET: A description of the target of the form processor-arch-manufacturer-os that will be passed on to the compiler. This CAP is only relevant for drivers that specify PIPE_SHADER_IR_NATIVE for their preferred IR. Value type: null-terminated string. Shader IR type dependent.
- PIPE_COMPUTE_CAP_GRID_DIMENSION: Number of supported dimensions for grid and block coordinates. Value type: uint64_t. Shader IR type dependent.
- PIPE_COMPUTE_CAP_MAX_GRID_SIZE: Maximum grid size in block units. Value type: uint64_t []. Shader IR type dependent.
- PIPE_COMPUTE_CAP_MAX_BLOCK_SIZE: Maximum block size in thread units. Value type: uint64_t []. Shader IR type dependent.
- PIPE_COMPUTE_CAP_MAX_THREADS_PER_BLOCK: Maximum number of threads that a single block can contain. Value type: uint64_t. Shader IR type dependent. This may be less than the product of the components of MAX_BLOCK_SIZE and is usually limited by the number of threads that can be resident simultaneously on a compute unit.
- PIPE_COMPUTE_CAP_MAX_GLOBAL_SIZE: Maximum size of the GLOBAL resource. Value type: uint64_t. Shader IR type dependent.
- PIPE_COMPUTE_CAP_MAX_LOCAL_SIZE: Maximum size of the LOCAL resource. Value type: uint64_t. Shader IR type dependent.
- PIPE_COMPUTE_CAP_MAX_PRIVATE_SIZE: Maximum size of the PRIVATE resource. Value type: uint64_t. Shader IR type dependent.
- PIPE_COMPUTE_CAP_MAX_INPUT_SIZE: Maximum size of the INPUT resource. Value type: uint64_t. Shader IR type dependent.

- PIPE_COMPUTE_CAP_MAX_MEM_ALLOC_SIZE: Maximum size of a memory object allocation in bytes. Value type: uint64_t.
- PIPE_COMPUTE_CAP_MAX_CLOCK_FREQUENCY: Maximum frequency of the GPU clock in MHz. Value type: uint32_t
- PIPE_COMPUTE_CAP_MAX_COMPUTE_UNITS: Maximum number of compute units Value type: uint32_t
- PIPE_COMPUTE_CAP_IMAGES_SUPPORTED: Whether images are supported non-zero means yes, zero means no. Value type: uint32_t
- PIPE_COMPUTE_CAP_SUBGROUP_SIZE: The size of a basic execution unit in threads. Also known as wavefront size, warp size or SIMD width.
- PIPE_COMPUTE_CAP_ADDRESS_BITS: The default compute device address space size specified as an unsigned integer value in bits.
- PIPE_COMPUTE_CAP_MAX_VARIABLE_THREADS_PER_BLOCK: Maximum variable number of threads that a single block can contain. This is similar to PIPE_COMPUTE_CAP_MAX_THREADS_PER_BLOCK, except that the variable size is not known a compile-time but at dispatch-time.

PIPE_BIND_*

These flags indicate how a resource will be used and are specified at resource creation time. Resources may be used in different roles during their lifecycle. Bind flags are cumulative and may be combined to create a resource which can be used for multiple things. Depending on the pipe driver's memory management and these bind flags, resources might be created and handled quite differently.

- PIPE_BIND_RENDER_TARGET: A color buffer or pixel buffer which will be rendered to. Any surface/resource attached to pipe_framebuffer_state::cbufs must have this flag set.
- PIPE_BIND_DEPTH_STENCIL: A depth (Z) buffer and/or stencil buffer. Any depth/stencil surface/resource attached to pipe_framebuffer_state::zsbuf must have this flag set.
- PIPE_BIND_BLENDABLE: Used in conjunction with PIPE_BIND_RENDER_TARGET to query whether a device supports blending for a given format. If this flag is set, surface creation may fail if blending is not supported for the specified format. If it is not set, a driver may choose to ignore blending on surfaces with formats that would require emulation.
- PIPE_BIND_DISPLAY_TARGET: A surface that can be presented to screen. Arguments to pipe_screen::flush_front_buffer must have this flag set.
- PIPE_BIND_SAMPLER_VIEW: A texture that may be sampled from in a fragment or vertex shader.
- PIPE_BIND_VERTEX_BUFFER: A vertex buffer.
- PIPE_BIND_INDEX_BUFFER: An vertex index/element buffer.
- PIPE_BIND_CONSTANT_BUFFER: A buffer of shader constants.
- PIPE_BIND_STREAM_OUTPUT: A stream output buffer.
- PIPE_BIND_CUSTOM:
- PIPE_BIND_SCANOUT: A front color buffer or scanout buffer.
- PIPE_BIND_SHARED: A sharable buffer that can be given to another process.
- PIPE_BIND_GLOBAL: A buffer that can be mapped into the global address space of a compute program.
- PIPE_BIND_SHADER_BUFFER: A buffer without a format that can be bound to a shader and can be used with load, store, and atomic instructions.

- `PIPE_BIND_SHADER_IMAGE`: A buffer or texture with a format that can be bound to a shader and can be used with load, store, and atomic instructions.
- `PIPE_BIND_COMPUTE_RESOURCE`: A buffer or texture that can be bound to the compute program as a shader resource.
- `PIPE_BIND_COMMAND_ARGS_BUFFER`: A buffer that may be sourced by the GPU command processor. It can contain, for example, the arguments to indirect draw calls.

PIPE_USAGE_*

The PIPE_USAGE enums are hints about the expected usage pattern of a resource. Note that drivers must always support read and write CPU access at any time no matter which hint they got.

- `PIPE_USAGE_DEFAULT`: Optimized for fast GPU access.
- `PIPE_USAGE_IMMUTABLE`: Optimized for fast GPU access and the resource is not expected to be mapped or changed (even by the GPU) after the first upload.
- `PIPE_USAGE_DYNAMIC`: Expect frequent write-only CPU access. What is uploaded is expected to be used at least several times by the GPU.
- `PIPE_USAGE_STREAM`: Expect frequent write-only CPU access. What is uploaded is expected to be used only once by the GPU.
- `PIPE_USAGE_STAGING`: Optimized for fast CPU access.

34.4.2 Methods

XXX to-do

get_name

Returns an identifying name for the screen.

The returned string should remain valid and immutable for the lifetime of pipe_screen.

get_vendor

Returns the screen vendor.

The returned string should remain valid and immutable for the lifetime of pipe_screen.

get_device_vendor

Returns the actual vendor of the device driving the screen (as opposed to the driver vendor).

The returned string should remain valid and immutable for the lifetime of pipe_screen.

get_param

Get an integer/boolean screen parameter.

param is one of the *PIPE_CAP_** names.

get_paramf

Get a floating-point screen parameter.

param is one of the *PIPE_CAPF_** names.

context_create

Create a pipe_context.

priv is private data of the caller, which may be put to various unspecified uses, typically to do with implementing swapbuffers and/or front-buffer rendering.

is_format_supported

Determine if a resource in the given format can be used in a specific manner.

format the resource format

target one of the PIPE_TEXTURE_x flags

sample_count the number of samples. 0 and 1 mean no multisampling, the maximum allowed legal value is 32.

storage_sample_count the number of storage samples. This must be <= sample_count. See the documentation of `pipe_resource::nr_storage_samples`.

bindings is a bitmask of *PIPE_BIND_** flags.

Returns TRUE if all usages can be satisfied.

can_create_resource

Check if a resource can actually be created (but don't actually allocate any memory). This is used to implement OpenGL's proxy textures. Typically, a driver will simply check if the total size of the given resource is less than some limit.

For PIPE_TEXTURE_CUBE, the pipe_resource::array_size field should be 6.

resource_create

Create a new resource from a template. The following fields of the pipe_resource must be specified in the template:

target one of the pipe_texture_target enums. Note that PIPE_BUFFER and PIPE_TEXTURE_X are not really fundamentally different. Modern APIs allow using buffers as shader resources.

format one of the pipe_format enums.

width0 the width of the base mip level of the texture or size of the buffer.

height0 the height of the base mip level of the texture (1 for 1D or 1D array textures).

depth0 the depth of the base mip level of the texture (1 for everything else).

array_size the array size for 1D and 2D array textures. For cube maps this must be 6, for other textures 1.

last_level the last mip map level present.

nr_samples: Number of samples determining quality, driving the rasterizer, shading, and framebuffer. It is the number of samples seen by the whole graphics pipeline. 0 and 1 specify a resource which isn't multisampled.

nr_storage_samples: Only color buffers can set this lower than nr_samples. Multiple samples within a pixel can have the same color. `nr_storage_samples` determines how many slots for different colors there are per pixel. If there are not enough slots to store all sample colors, some samples will have an undefined color (called "undefined samples").

The resolve blit behavior is driver-specific, but can be one of these two:

1. Only defined samples will be averaged. Undefined samples will be ignored.
2. Undefined samples will be approximated by looking at surrounding defined samples (even in different pixels).

Blits and MSAA texturing: If the sample being fetched is undefined, one of the defined samples is returned instead.

Sample shading (`set_min_samples`) will operate at a sample frequency that is at most `nr_storage_samples`. Greater `min_samples` values will be replaced by `nr_storage_samples`.

usage one of the *PIPE_USAGE_** flags.

bind bitmask of the *PIPE_BIND_** flags.

flags bitmask of PIPE_RESOURCE_FLAG flags.

next: Pointer to the next plane for resources that consist of multiple memory planes.

As a corollary, this mean resources for an image with multiple planes have to be created starting from the highest plane.

resource_changed

Mark a resource as changed so derived internal resources will be recreated on next use.

When importing external images that can't be directly used as texture sampler source, internal copies may have to be created that the hardware can sample from. When those resources are reimported, the image data may have changed, and the previously derived internal resources must be invalidated to avoid sampling from old copies.

resource_destroy

Destroy a resource. A resource is destroyed if it has no more references.

get_timestamp

Query a timestamp in nanoseconds. The returned value should match PIPE_QUERY_TIMESTAMP. This function returns immediately and doesn't wait for rendering to complete (which cannot be achieved with queries).

get_driver_query_info

Return a driver-specific query. If the **info** parameter is NULL, the number of available queries is returned. Otherwise, the driver query at the specified **index** is returned in **info**. The function returns non-zero on success. The driver-specific query is described with the pipe_driver_query_info structure.

get_driver_query_group_info

Return a driver-specific query group. If the **info** parameter is NULL, the number of available groups is returned. Otherwise, the driver query group at the specified **index** is returned in **info**. The function returns non-zero on success. The driver-specific query group is described with the pipe_driver_query_group_info structure.

get_disk_shader_cache

Returns a pointer to a driver-specific on-disk shader cache. If the driver failed to create the cache or does not support an on-disk shader cache NULL is returned. The callback itself may also be NULL if the driver doesn't support an on-disk shader cache.

34.4.3 Thread safety

Screen methods are required to be thread safe. While gallium rendering contexts are not required to be thread safe, it is required to be safe to use different contexts created with the same screen in different threads without locks. It is also required to be safe using screen methods in a thread, while using one of its contexts in another (without locks).

34.5 Resources and derived objects

Resources represent objects that hold data: textures and buffers.

They are mostly modelled after the resources in Direct3D 10/11, but with a different transfer/update mechanism, and more features for OpenGL support.

Resources can be used in several ways, and it is required to specify all planned uses through an appropriate set of bind flags.

TODO: write much more on resources

34.5.1 Transfers

Transfers are the mechanism used to access resources with the CPU.

OpenGL: OpenGL supports mapping buffers and has inline transfer functions for both buffers and textures

D3D11: D3D11 lacks transfers, but has special resource types that are mappable to the CPU address space

TODO: write much more on transfers

34.5.2 Resource targets

Resource targets determine the type of a resource.

Note that drivers may not actually have the restrictions listed regarding coordinate normalization and wrap modes, and in fact efficient OpenCL support will probably require drivers that don't have any of them, which will probably be advertised with an appropriate cap.

TODO: document all targets. Note that both 3D and cube have restrictions that depend on the hardware generation.

PIPE_BUFFER

Buffer resource: can be used as a vertex, index, constant buffer (appropriate bind flags must be requested).

Buffers do not really have a format, it's just bytes, but they are required to have their type set to a R8 format (without a specific "just byte" format, R8_UINT would probably make the most sense, but for historic reasons R8_UNORM is ok too). (This is just to make some shared buffer/texture code easier so format size can be queried.) width0 serves as size, most other resource properties don't apply but must be set appropriately (depth0/height0/array_size must be 1, last_level 0).

They can be bound to stream output if supported. TODO: what about the restrictions lifted by the several later GL transform feedback extensions? How does one advertise that in Gallium?

They can be also be bound to a shader stage (for sampling) as usual by creating an appropriate sampler view, if the driver supports PIPE_CAP_TEXTURE_BUFFER_OBJECTS. This supports larger width than a 1d texture would (TODO limit currently unspecified, minimum must be at least 65536). Only the "direct fetch" sample opcodes are supported (TGSI_OPCODE_TXF, TGSI_OPCODE_SAMPLE_I) so the sampler state (coord wrapping etc.) is mostly ignored (with SAMPLE_I there's no sampler state at all).

They can be also be bound to the framebuffer (only as color render target, not depth buffer, also there cannot be a depth buffer bound at the same time) as usual by creating an appropriate view (this is not usable in OpenGL). TODO there's no CAP bit currently for this, there's also unspecified size etc. limits TODO: is there any chance of supporting GL pixel buffer object acceleration with this?

OpenGL: vertex buffers in GL 1.5 or GL_ARB_vertex_buffer_object

- Binding to stream out requires GL 3.0 or GL_NV_transform_feedback
- Binding as constant buffers requires GL 3.1 or GL_ARB_uniform_buffer_object
- Binding to a sampling stage requires GL 3.1 or GL_ARB_texture_buffer_object

D3D11: buffer resources - Binding to a render target requires D3D_FEATURE_LEVEL_10_0

PIPE_TEXTURE_1D / PIPE_TEXTURE_1D_ARRAY

1D surface accessed with normalized coordinates. 1D array textures are supported depending on PIPE_CAP_MAX_TEXTURE_ARRAY_LAYERS.

- **If PIPE_CAP_NPOT_TEXTURES is not supported,** width must be a power of two
- height0 must be 1
- depth0 must be 1
- array_size must be 1 for PIPE_TEXTURE_1D
- Mipmaps can be used
- Must use normalized coordinates

OpenGL: GL_TEXTURE_1D in GL 1.0

- PIPE_CAP_NPOT_TEXTURES is equivalent to GL 2.0 or GL_ARB_texture_non_power_of_two

D3D11: 1D textures in D3D_FEATURE_LEVEL_10_0

PIPE_TEXTURE_RECT

2D surface with OpenGL GL_TEXTURE_RECTANGLE semantics.

- depth0 must be 1
- array_size must be 1
- last_level must be 0
- Must use unnormalized coordinates
- Must use a clamp wrap mode

OpenGL: GL_TEXTURE_RECTANGLE in GL 3.1 or GL_ARB_texture_rectangle or GL_NV_texture_rectangle

OpenCL: can create OpenCL images based on this, that can then be sampled arbitrarily

D3D11: not supported (only PIPE_TEXTURE_2D with normalized coordinates is supported)

PIPE_TEXTURE_2D / PIPE_TEXTURE_2D_ARRAY

2D surface accessed with normalized coordinates. 2D array textures are supported depending on PIPE_CAP_MAX_TEXTURE_ARRAY_LAYERS.

- **If PIPE_CAP_NPOT_TEXTURES is not supported,** width and height must be powers of two
- depth0 must be 1
- array_size must be 1 for PIPE_TEXTURE_2D
- Mipmaps can be used
- Must use normalized coordinates
- No special restrictions on wrap modes

OpenGL: GL_TEXTURE_2D in GL 1.0

- PIPE_CAP_NPOT_TEXTURES is equivalent to GL 2.0 or GL_ARB_texture_non_power_of_two

OpenCL: can create OpenCL images based on this, that can then be sampled arbitrarily

D3D11: 2D textures

- PIPE_CAP_NPOT_TEXTURES is equivalent to D3D_FEATURE_LEVEL_9_3

PIPE_TEXTURE_3D

3-dimensional array of texels. Mipmap dimensions are reduced in all 3 coordinates.

- **If PIPE_CAP_NPOT_TEXTURES is not supported,** width, height and depth must be powers of two
- array_size must be 1
- Must use normalized coordinates

OpenGL: GL_TEXTURE_3D in GL 1.2 or GL_EXT_texture3D

- PIPE_CAP_NPOT_TEXTURES is equivalent to GL 2.0 or GL_ARB_texture_non_power_of_two

D3D11: 3D textures

- PIPE_CAP_NPOT_TEXTURES is equivalent to D3D_FEATURE_LEVEL_10_0

PIPE_TEXTURE_CUBE / PIPE_TEXTURE_CUBE_ARRAY

Cube maps consist of 6 2D faces. The 6 surfaces form an imaginary cube, and sampling happens by mapping an input 3-vector to the point of the cube surface in that direction. Cube map arrays are supported depending on PIPE_CAP_CUBE_MAP_ARRAY.

Sampling may be optionally seamless if a driver supports it (PIPE_CAP_SEAMLESS_CUBE_MAP), resulting in filtering taking samples from multiple surfaces near to the edge.

- Width and height must be equal
- depth0 must be 1

- array_size must be a multiple of 6
- **If PIPE_CAP_NPOT_TEXTURES is not supported,** width and height must be powers of two
- Must use normalized coordinates

OpenGL: GL_TEXTURE_CUBE_MAP in GL 1.3 or EXT_texture_cube_map

- PIPE_CAP_NPOT_TEXTURES is equivalent to GL 2.0 or GL_ARB_texture_non_power_of_two
- Seamless cube maps require GL 3.2 or GL_ARB_seamless_cube_map or GL_AMD_seamless_cubemap_per_texture
- Cube map arrays require GL 4.0 or GL_ARB_texture_cube_map_array

D3D11: 2D array textures with the D3D11_RESOURCE_MISC_TEXTURECUBE flag

- PIPE_CAP_NPOT_TEXTURES is equivalent to D3D_FEATURE_LEVEL_10_0
- Cube map arrays require D3D_FEATURE_LEVEL_10_1

34.5.3 Surfaces

Surfaces are views of a resource that can be bound as a framebuffer to serve as the render target or depth buffer.

TODO: write much more on surfaces

OpenGL: FBOs are collections of surfaces in GL 3.0 or GL_ARB_framebuffer_object

D3D11: render target views and depth/stencil views

34.5.4 Sampler views

Sampler views are views of a resource that can be bound to a pipeline stage to be sampled from shaders.

TODO: write much more on sampler views

OpenGL: texture objects are actually sampler view and resource in a single unit

D3D11: shader resource views

34.6 Formats in gallium

Gallium format names mostly follow D3D10 conventions, with some extensions.

Format names like XnYnZnWn have the X component in the lowest-address n bits and the W component in the highest-address n bits; for B8G8R8A8, byte 0 is blue and byte 3 is alpha. Note that platform endianness is not considered in this definition. In C:

```
struct x8y8z8w8 { uint8_t x, y, z, w; };
```

Format aliases like XYZWstrq are (s+t+r+q)-bit integers in host endianness, with the X component in the s least-significant bits of the integer. In C:

```
uint32_t xyzw8888 = (x << 0) | (y << 8) | (z << 16) | (w << 24);
```

Format suffixes affect the interpretation of the channel:

- `SINT`: N bit signed integer [-2^(N-1) . . . 2^(N-1) - 1]

- `SNORM`: N bit signed integer normalized to [-1 . . . 1]
- `SSCALED`: N bit signed integer [-2^(N-1) . . . 2^(N-1) - 1]
- `FIXED`: Signed fixed point integer, (N/2 - 1) bits of mantissa
- `FLOAT`: N bit IEEE754 float
- `NORM`: Normalized integers, signed or unsigned per channel
- `UINT`: N bit unsigned integer [0 . . . 2^N - 1]
- `UNORM`: N bit unsigned integer normalized to [0 . . . 1]
- `USCALED`: N bit unsigned integer [0 . . . 2^N - 1]

The difference between `SINT` and `SSCALED` is that the former are pure integers in shaders, while the latter are floats; likewise for `UINT` versus `USCALED`.

There are two exceptions for `FLOAT`. `R9G9B9E5_FLOAT` is nine bits each of red green and blue mantissa, with a shared five bit exponent. `R11G11B10_FLOAT` is five bits of exponent and five or six bits of mantissa for each color channel.

For the `NORM` suffix, the signedness of each channel is indicated with an S or U after the number of channel bits, as in `R5SG5SB6U_NORM`.

The `SRGB` suffix is like `UNORM` in range, but in the sRGB colorspace.

Compressed formats are named first by the compression format string (`DXT1`, `ETC1`, etc), followed by a format-specific subtype. Refer to the appropriate compression spec for details.

Formats used in video playback are named by their FOURCC code.

Format names with an embedded underscore are subsampled. `R8G8_B8G8` is a single 32-bit block of two pixels, where the R and B values are repeated in both pixels.

34.6.1 References

DirectX Graphics Infrastructure documentation on DXGI_FORMAT enum: http://msdn.microsoft.com/en-us/library/windows/desktop/bb173059%28v=vs.85%29.aspx

FOURCC codes for YUV formats: http://www.fourcc.org/yuv.php

34.7 Context

A Gallium rendering context encapsulates the state which effects 3D rendering such as blend state, depth/stencil state, texture samplers, etc.

Note that resource/texture allocation is not per-context but per-screen.

34.7.1 Methods

CSO State

All Constant State Object (CSO) state is created, bound, and destroyed, with triplets of methods that all follow a specific naming scheme. For example, `create_blend_state`, `bind_blend_state`, and `destroy_blend_state`.

CSO objects handled by the context object:

- *Blend*: `*_blend_state`
- *Sampler*: Texture sampler states are bound separately for fragment, vertex, geometry and compute shaders with the `bind_sampler_states` function. The `start` and `num_samplers` parameters indicate a range of samplers to change. NOTE: at this time, start is always zero and the CSO module will always replace all samplers at once (no sub-ranges). This may change in the future.
- *Rasterizer*: `*_rasterizer_state`
- *Depth, Stencil, & Alpha*: `*_depth_stencil_alpha_state`
- *Shader*: These are create, bind and destroy methods for vertex, fragment and geometry shaders.
- *Vertex Elements*: `*_vertex_elements_state`

Resource Binding State

This state describes how resources in various flavors (textures, buffers, surfaces) are bound to the driver.

- `set_constant_buffer` sets a constant buffer to be used for a given shader type. index is used to indicate which buffer to set (some APIs may allow multiple ones to be set, and binding a specific one later, though drivers are mostly restricted to the first one right now).
- `set_inlinable_constants` sets inlinable constants for constant buffer 0.

These are constants that the driver would like to inline in the IR of the current shader and recompile it. Drivers can determine which constants they prefer to inline in finalize_nir and store that information in shader_info::*inlinable_uniform*. When the state tracker or frontend uploads constants to a constant buffer, it can pass inlinable constants separately via this call.

Any `set_constant_buffer` call invalidates inlinable constants, so `set_inlinable_constants` must be called after it. Binding a shader also invalidates this state.

There is no `PIPE_CAP` for this. Drivers shouldn't set the shader_info fields if they don't implement `set_inlinable_constants`.

- `set_framebuffer_state`
- `set_vertex_buffers`

Non-CSO State

These pieces of state are too small, variable, and/or trivial to have CSO objects. They all follow simple, one-method binding calls, e.g. `set_blend_color`.

- `set_stencil_ref` sets the stencil front and back reference values which are used as comparison values in stencil test.
- `set_blend_color`
- `set_sample_mask` sets the per-context multisample sample mask. Note that this takes effect even if multisampling is not explicitly enabled if the framebuffer surface(s) are multisampled. Also, this mask is AND-ed with the optional fragment shader sample mask output (when emitted).
- `set_sample_locations` sets the sample locations used for rasterization. `` `get_sample_position` `` still returns the default locations. When NULL, the default locations are used.
- `set_min_samples` sets the minimum number of samples that must be run.
- `set_clip_state`
- `set_polygon_stipple`

- `set_scissor_states` sets the bounds for the scissor test, which culls pixels before blending to render targets. If the *Rasterizer* does not have the scissor test enabled, then the scissor bounds never need to be set since they will not be used. Note that scissor xmin and ymin are inclusive, but xmax and ymax are exclusive. The inclusive ranges in x and y would be [xmin..xmax-1] and [ymin..ymax-1]. The number of scissors should be the same as the number of set viewports and can be up to PIPE_MAX_VIEWPORTS.
- `set_viewport_states`
- `set_window_rectangles` sets the window rectangles to be used for rendering, as defined by GL_EXT_window_rectangles. There are two modes - include and exclude, which define whether the supplied rectangles are to be used for including fragments or excluding them. All of the rectangles are ORed together, so in exclude mode, any fragment inside any rectangle would be culled, while in include mode, any fragment outside all rectangles would be culled. xmin/ymin are inclusive, while xmax/ymax are exclusive (same as scissor states above). Note that this only applies to draws, not clears or blits. (Blits have their own way to pass the requisite rectangles in.)
- `set_tess_state` configures the default tessellation parameters:
  - `default_outer_level` is the default value for the outer tessellation levels. This corresponds to GL's `PATCH_DEFAULT_OUTER_LEVEL`.
  - `default_inner_level` is the default value for the inner tessellation levels. This corresponds to GL's `PATCH_DEFAULT_INNER_LEVEL`.
- `set_debug_callback` sets the callback to be used for reporting various debug messages, eventually reported via KHR_debug and similar mechanisms.

Samplers

pipe_sampler_state objects control how textures are sampled (coordinate wrap modes, interpolation modes, etc). Note that samplers are not used for texture buffer objects. That is, pipe_context::bind_sampler_views() will not bind a sampler if the corresponding sampler view refers to a PIPE_BUFFER resource.

Sampler Views

These are the means to bind textures to shader stages. To create one, specify its format, swizzle and LOD range in sampler view template.

If texture format is different than template format, it is said the texture is being cast to another format. Casting can be done only between compatible formats, that is formats that have matching component order and sizes.

Swizzle fields specify the way in which fetched texel components are placed in the result register. For example, `swizzle_r` specifies what is going to be placed in first component of result register.

The `first_level` and `last_level` fields of sampler view template specify the LOD range the texture is going to be constrained to. Note that these values are in addition to the respective min_lod, max_lod values in the pipe_sampler_state (that is if min_lod is 2.0, and first_level 3, the first mip level used for sampling from the resource is effectively the fifth).

The `first_layer` and `last_layer` fields specify the layer range the texture is going to be constrained to. Similar to the LOD range, this is added to the array index which is used for sampling.

- `set_sampler_views` binds an array of sampler views to a shader stage. Every binding point acquires a reference to a respective sampler view and releases a reference to the previous sampler view.

  Sampler views outside of `[start_slot, start_slot + num_views)` are unmodified. If `views` is NULL, the behavior is the same as if `views[n]` was NULL for the entire range, ie. releasing the reference for all the sampler views in the specified range.

- `create_sampler_view` creates a new sampler view. `texture` is associated with the sampler view which results in sampler view holding a reference to the texture. Format specified in template must be compatible with texture format.
- `sampler_view_destroy` destroys a sampler view and releases its reference to associated texture.

Hardware Atomic buffers

Buffers containing hw atomics are required to support the feature on some drivers.

Drivers that require this need to fill the `set_hw_atomic_buffers` method.

Shader Resources

Shader resources are textures or buffers that may be read or written from a shader without an associated sampler. This means that they have no support for floating point coordinates, address wrap modes or filtering.

There are 2 types of shader resources: buffers and images.

Buffers are specified using the `set_shader_buffers` method.

Images are specified using the `set_shader_images` method. When binding images, the `level`, `first_layer` and `last_layer` pipe_image_view fields specify the mipmap level and the range of layers the image will be constrained to.

Surfaces

These are the means to use resources as color render targets or depthstencil attachments. To create one, specify the mip level, the range of layers, and the bind flags (either PIPE_BIND_DEPTH_STENCIL or PIPE_BIND_RENDER_TARGET). Note that layer values are in addition to what is indicated by the geometry shader output variable XXX_FIXME (that is if first_layer is 3 and geometry shader indicates index 2, the 5th layer of the resource will be used). These first_layer and last_layer parameters will only be used for 1d array, 2d array, cube, and 3d textures otherwise they are 0.

- `create_surface` creates a new surface.
- `surface_destroy` destroys a surface and releases its reference to the associated resource.

Stream output targets

Stream output, also known as transform feedback, allows writing the primitives produced by the vertex pipeline to buffers. This is done after the geometry shader or vertex shader if no geometry shader is present.

The stream output targets are views into buffer resources which can be bound as stream outputs and specify a memory range where it's valid to write primitives. The pipe driver must implement memory protection such that any primitives written outside of the specified memory range are discarded.

Two stream output targets can use the same resource at the same time, but with a disjoint memory range.

Additionally, the stream output target internally maintains the offset into the buffer which is incremented every time something is written to it. The internal offset is equal to how much data has already been written. It can be stored in device memory and the CPU actually doesn't have to query it.

The stream output target can be used in a draw command to provide the vertex count. The vertex count is derived from the internal offset discussed above.

- `create_stream_output_target` create a new target.

- `stream_output_target_destroy` destroys a target. Users of this should use pipe_so_target_reference instead.
- `set_stream_output_targets` binds stream output targets. The parameter offset is an array which specifies the internal offset of the buffer. The internal offset is, besides writing, used for reading the data during the draw_auto stage, i.e. it specifies how much data there is in the buffer for the purposes of the draw_auto stage. -1 means the buffer should be appended to, and everything else sets the internal offset.

NOTE: The currently-bound vertex or geometry shader must be compiled with the properly-filled-in structure pipe_stream_output_info describing which outputs should be written to buffers and how. The structure is part of pipe_shader_state.

Clearing

Clear is one of the most difficult concepts to nail down to a single interface (due to both different requirements from APIs and also driver/hw specific differences).

`clear` initializes some or all of the surfaces currently bound to the framebuffer to particular RGBA, depth, or stencil values. Currently, this does not take into account color or stencil write masks (as used by GL), and always clears the whole surfaces (no scissoring as used by GL clear or explicit rectangles like d3d9 uses). It can, however, also clear only depth or stencil in a combined depth/stencil surface. If a surface includes several layers then all layers will be cleared.

`clear_render_target` clears a single color rendertarget with the specified color value. While it is only possible to clear one surface at a time (which can include several layers), this surface need not be bound to the framebuffer. If render_condition_enabled is false, any current rendering condition is ignored and the clear will be unconditional.

`clear_depth_stencil` clears a single depth, stencil or depth/stencil surface with the specified depth and stencil values (for combined depth/stencil buffers, it is also possible to only clear one or the other part). While it is only possible to clear one surface at a time (which can include several layers), this surface need not be bound to the framebuffer. If render_condition_enabled is false, any current rendering condition is ignored and the clear will be unconditional.

`clear_texture` clears a non-PIPE_BUFFER resource's specified level and bounding box with a clear value provided in that resource's native format.

`clear_buffer` clears a PIPE_BUFFER resource with the specified clear value (which may be multiple bytes in length). Logically this is a memset with a multi-byte element value starting at offset bytes from resource start, going for size bytes. It is guaranteed that size % clear_value_size == 0.

Evaluating Depth Buffers

`evaluate_depth_buffer` is a hint to decompress the current depth buffer assuming the current sample locations to avoid problems that could arise when using programmable sample locations.

If a depth buffer is rendered with different sample location state than what is current at the time of reading the depth buffer, the values may differ because depth buffer compression can depend the sample locations.

Uploading

For simple single-use uploads, use `pipe_context::stream_uploader` or `pipe_context::const_uploader`. The latter should be used for uploading constants, while the former should be used for uploading everything else. PIPE_USAGE_STREAM is implied in both cases, so don't use the uploaders for static allocations.

Usage:

Call u_upload_alloc or u_upload_data as many times as you want. After you are done, call u_upload_unmap. If the driver doesn't support persistent mappings, u_upload_unmap makes sure the previously mapped memory is unmapped.

Gotchas: - Always fill the memory immediately after u_upload_alloc. Any following call to u_upload_alloc and u_upload_data can unmap memory returned by previous u_upload_alloc. - Don't interleave calls using stream_uploader and const_uploader. If you use one of them, do the upload, unmap, and only then can you use the other one.

Drawing

`draw_vbo` draws a specified primitive. The primitive mode and other properties are described by `pipe_draw_info`.

The `mode`, `start`, and `count` fields of `pipe_draw_info` specify the the mode of the primitive and the vertices to be fetched, in the range between `start` to `start``+``count`-1, inclusive.

Every instance with instanceID in the range between `start_instance` and `start_instance``+``instance_count`-1, inclusive, will be drawn.

If `index_size` != 0, all vertex indices will be looked up from the index buffer.

In indexed draw, `min_index` and `max_index` respectively provide a lower and upper bound of the indices contained in the index buffer inside the range between `start` to `start``+``count`-1. This allows the driver to determine which subset of vertices will be referenced during te draw call without having to scan the index buffer. Providing a over-estimation of the the true bounds, for example, a `min_index` and `max_index` of 0 and 0xffffffff respectively, must give exactly the same rendering, albeit with less performance due to unreferenced vertex buffers being unnecessarily DMA'ed or processed. Providing a underestimation of the true bounds will result in undefined behavior, but should not result in program or system failure.

In case of non-indexed draw, `min_index` should be set to `start` and `max_index` should be set to `start``+``count`-1.

`index_bias` is a value added to every vertex index after lookup and before fetching vertex attributes.

When drawing indexed primitives, the primitive restart index can be used to draw disjoint primitive strips. For example, several separate line strips can be drawn by designating a special index value as the restart index. The `primitive_restart` flag enables/disables this feature. The `restart_index` field specifies the restart index value.

When primitive restart is in use, array indexes are compared to the restart index before adding the index_bias offset.

If a given vertex element has `instance_divisor` set to 0, it is said it contains per-vertex data and effective vertex attribute address needs to be recalculated for every index.

> attribAddr = `stride` * index + `src_offset`

If a given vertex element has `instance_divisor` set to non-zero, it is said it contains per-instance data and effective vertex attribute address needs to recalculated for every `instance_divisor`-th instance.

> attribAddr = `stride` * instanceID / `instance_divisor` + `src_offset`

In the above formulas, `src_offset` is taken from the given vertex element and `stride` is taken from a vertex buffer associated with the given vertex element.

The calculated attribAddr is used as an offset into the vertex buffer to fetch the attribute data.

The value of `instanceID` can be read in a vertex shader through a system value register declared with INSTANCEID semantic name.

Queries

Queries gather some statistic from the 3D pipeline over one or more draws. Queries may be nested, though not all gallium frontends exercise this.

Queries can be created with `create_query` and deleted with `destroy_query`. To start a query, use `begin_query`, and when finished, use `end_query` to end the query.

`create_query` takes a query type (`PIPE_QUERY_*`), as well as an index, which is the vertex stream for `PIPE_QUERY_PRIMITIVES_GENERATED` and `PIPE_QUERY_PRIMITIVES_EMITTED`, and allocates a query structure.

`begin_query` will clear/reset previous query results.

`get_query_result` is used to retrieve the results of a query. If the `wait` parameter is TRUE, then the `get_query_result` call will block until the results of the query are ready (and TRUE will be returned). Otherwise, if the `wait` parameter is FALSE, the call will not block and the return value will be TRUE if the query has completed or FALSE otherwise.

`get_query_result_resource` is used to store the result of a query into a resource without synchronizing with the CPU. This write will optionally wait for the query to complete, and will optionally write whether the value is available instead of the value itself.

`set_active_query_state` Set whether all current non-driver queries except TIME_ELAPSED are active or paused.

The interface currently includes the following types of queries:

`PIPE_QUERY_OCCLUSION_COUNTER` counts the number of fragments which are written to the framebuffer without being culled by *Depth, Stencil, & Alpha* testing or shader KILL instructions. The result is an unsigned 64-bit integer. This query can be used with `render_condition`.

In cases where a boolean result of an occlusion query is enough, `PIPE_QUERY_OCCLUSION_PREDICATE` should be used. It is just like `PIPE_QUERY_OCCLUSION_COUNTER` except that the result is a boolean value of FALSE for cases where COUNTER would result in 0 and TRUE for all other cases. This query can be used with `render_condition`.

In cases where a conservative approximation of an occlusion query is enough, `PIPE_QUERY_OCCLUSION_PREDICATE_CONSERVATIVE` should be used. It behaves like `PIPE_QUERY_OCCLUSION_PREDICATE`, except that it may return TRUE in additional, implementation-dependent cases. This query can be used with `render_condition`.

`PIPE_QUERY_TIME_ELAPSED` returns the amount of time, in nanoseconds, the context takes to perform operations. The result is an unsigned 64-bit integer.

`PIPE_QUERY_TIMESTAMP` returns a device/driver internal timestamp, scaled to nanoseconds, recorded after all commands issued prior to `end_query` have been processed. This query does not require a call to `begin_query`. The result is an unsigned 64-bit integer.

`PIPE_QUERY_TIMESTAMP_DISJOINT` can be used to check the internal timer resolution and whether the timestamp counter has become unreliable due to things like throttling etc. - only if this is FALSE a timestamp query (within the timestamp_disjoint query) should be trusted. The result is a 64-bit integer specifying the timer resolution in Hz, followed by a boolean value indicating whether the timestamp counter is discontinuous or disjoint.

`PIPE_QUERY_PRIMITIVES_GENERATED` returns a 64-bit integer indicating the number of primitives processed by the pipeline (regardless of whether stream output is active or not).

`PIPE_QUERY_PRIMITIVES_EMITTED` returns a 64-bit integer indicating the number of primitives written to stream output buffers.

`PIPE_QUERY_SO_STATISTICS` returns 2 64-bit integers corresponding to the result of `PIPE_QUERY_PRIMITIVES_EMITTED` and the number of primitives that would have been written to stream

output buffers if they had infinite space available (primitives_storage_needed), in this order. XXX the 2nd value is equivalent to `PIPE_QUERY_PRIMITIVES_GENERATED` but it is unclear if it should be increased if stream output is not active.

`PIPE_QUERY_SO_OVERFLOW_PREDICATE` returns a boolean value indicating whether a selected stream output target has overflowed as a result of the commands issued between `begin_query` and `end_query`. This query can be used with `render_condition`. The output stream is selected by the stream number passed to `create_query`.

`PIPE_QUERY_SO_OVERFLOW_ANY_PREDICATE` returns a boolean value indicating whether any stream output target has overflowed as a result of the commands issued between `begin_query` and `end_query`. This query can be used with `render_condition`, and its result is the logical OR of multiple `PIPE_QUERY_SO_OVERFLOW_PREDICATE` queries, one for each stream output target.

`PIPE_QUERY_GPU_FINISHED` returns a boolean value indicating whether all commands issued before `end_query` have completed. However, this does not imply serialization. This query does not require a call to `begin_query`.

`PIPE_QUERY_PIPELINE_STATISTICS` returns an array of the following 64-bit integers: Number of vertices read from vertex buffers. Number of primitives read from vertex buffers. Number of vertex shader threads launched. Number of geometry shader threads launched. Number of primitives generated by geometry shaders. Number of primitives forwarded to the rasterizer. Number of primitives rasterized. Number of fragment shader threads launched. Number of tessellation control shader threads launched. Number of tessellation evaluation shader threads launched. If a shader type is not supported by the device/driver, the corresponding values should be set to 0.

`PIPE_QUERY_PIPELINE_STATISTICS_SINGLE` returns a single counter from the `PIPE_QUERY_PIPELINE_STATISTICS` group. The specific counter must be selected when calling `create_query` by passing one of the `PIPE_STAT_QUERY` enums as the query's `index`.

Gallium does not guarantee the availability of any query types; one must always check the capabilities of the *Screen* first.

Conditional Rendering

A drawing command can be skipped depending on the outcome of a query (typically an occlusion query, or streamout overflow predicate). The `render_condition` function specifies the query which should be checked prior to rendering anything. Functions always honoring render_condition include (and are limited to) draw_vbo and clear. The blit, clear_render_target and clear_depth_stencil functions (but not resource_copy_region, which seems inconsistent) can also optionally honor the current render condition.

If `render_condition` is called with `query` = NULL, conditional rendering is disabled and drawing takes place normally.

If `render_condition` is called with a non-null `query` subsequent drawing commands will be predicated on the outcome of the query. Commands will be skipped if `condition` is equal to the predicate result (for non-boolean queries such as OCCLUSION_QUERY, zero counts as FALSE, non-zero as TRUE).

If `mode` is PIPE_RENDER_COND_WAIT the driver will wait for the query to complete before deciding whether to render.

If `mode` is PIPE_RENDER_COND_NO_WAIT and the query has not yet completed, the drawing command will be executed normally. If the query has completed, drawing will be predicated on the outcome of the query.

If `mode` is PIPE_RENDER_COND_BY_REGION_WAIT or PIPE_RENDER_COND_BY_REGION_NO_WAIT rendering will be predicated as above for the non-REGION modes but in the case that an occlusion query returns a non-zero result, regions which were occluded may be ommitted by subsequent drawing commands. This can result in better performance with some GPUs. Normally, if the occlusion query returned a non-zero result subsequent drawing happens normally so fragments may be generated, shaded and processed even where they're known to be obscured.

Flushing

`flush`

PIPE_FLUSH_END_OF_FRAME: Whether the flush marks the end of frame.

PIPE_FLUSH_DEFERRED: It is not required to flush right away, but it is required to return a valid fence. If fence_finish is called with the returned fence and the context is still unflushed, and the ctx parameter of fence_finish is equal to the context where the fence was created, fence_finish will flush the context.

PIPE_FLUSH_ASYNC: The flush is allowed to be asynchronous. Unlike `PIPE_FLUSH_DEFERRED`, the driver must still ensure that the returned fence will finish in finite time. However, subsequent operations in other contexts of the same screen are no longer guaranteed to happen after the flush. Drivers which use this flag must implement pipe_context::fence_server_sync.

PIPE_FLUSH_HINT_FINISH: Hints to the driver that the caller will immediately wait for the returned fence.

Additional flags may be set together with `PIPE_FLUSH_DEFERRED` for even finer-grained fences. Note that as a general rule, GPU caches may not have been flushed yet when these fences are signaled. Drivers are free to ignore these flags and create normal fences instead. At most one of the following flags can be specified:

PIPE_FLUSH_TOP_OF_PIPE: The fence should be signaled as soon as the next command is ready to start executing at the top of the pipeline, before any of its data is actually read (including indirect draw parameters).

PIPE_FLUSH_BOTTOM_OF_PIPE: The fence should be signaled as soon as the previous command has finished executing on the GPU entirely (but data written by the command may still be in caches and inaccessible to the CPU).

`flush_resource`

Flush the resource cache, so that the resource can be used by an external client. Possible usage: - flushing a resource before presenting it on the screen - flushing a resource if some other process or device wants to use it This shouldn't be used to flush caches if the resource is only managed by a single pipe_screen and is not shared with another process. (i.e. you shouldn't use it to flush caches explicitly if you want to e.g. use the resource for texturing)

Fences

`pipe_fence_handle`, and related methods, are used to synchronize execution between multiple parties. Examples include CPU <-> GPU synchronization, renderer <-> windowing system, multiple external APIs, etc.

A `pipe_fence_handle` can either be 'one time use' or 're-usable'. A 'one time use' fence behaves like a traditional GPU fence. Once it reaches the signaled state it is forever considered to be signaled.

Once a re-usable `pipe_fence_handle` becomes signaled, it can be reset back into an unsignaled state. The `pipe_fence_handle` will be reset to the unsignaled state by performing a wait operation on said object, i.e. `fence_server_sync`. As a corollary to this behavior, a re-usable `pipe_fence_handle` can only have one waiter.

This behavior is useful in producer <-> consumer chains. It helps avoid unnecessarily sharing a new `pipe_fence_handle` each time a new frame is ready. Instead, the fences are exchanged once ahead of time, and access is synchronized through GPU signaling instead of direct producer <-> consumer communication.

`fence_server_sync` inserts a wait command into the GPU's command stream.

`fence_server_signal` inserts a signal command into the GPU's command stream.

There are no guarantees that the wait/signal commands will be flushed when calling `fence_server_sync` or `fence_server_signal`. An explicit call to `flush` is required to make sure the commands are emitted to the GPU.

The Gallium implementation may implicitly `flush` the command stream during a `fence_server_sync` or `fence_server_signal` call if necessary.

Resource Busy Queries

`is_resource_referenced`

Blitting

These methods emulate classic blitter controls.

These methods operate directly on `pipe_resource` objects, and stand apart from any 3D state in the context. Blitting functionality may be moved to a separate abstraction at some point in the future.

`resource_copy_region` blits a region of a resource to a region of another resource, provided that both resources have the same format, or compatible formats, i.e., formats for which copying the bytes from the source resource unmodified to the destination resource will achieve the same effect of a textured quad blitter.. The source and destination may be the same resource, but overlapping blits are not permitted. This can be considered the equivalent of a CPU memcpy.

`blit` blits a region of a resource to a region of another resource, including scaling, format conversion, and up-/downsampling, as well as a destination clip rectangle (scissors) and window rectangles. It can also optionally honor the current render condition (but either way the blit itself never contributes anything to queries currently gathering data). As opposed to manually drawing a textured quad, this lets the pipe driver choose the optimal method for blitting (like using a special 2D engine), and usually offers, for example, accelerated stencil-only copies even where PIPE_CAP_SHADER_STENCIL_EXPORT is not available.

Transfers

These methods are used to get data to/from a resource.

`transfer_map` creates a memory mapping and the transfer object associated with it. The returned pointer points to the start of the mapped range according to the box region, not the beginning of the resource. If transfer_map fails, the returned pointer to the buffer memory is NULL, and the pointer to the transfer object remains unchanged (i.e. it can be non-NULL).

`transfer_unmap` remove the memory mapping for and destroy the transfer object. The pointer into the resource should be considered invalid and discarded.

`texture_subdata` and `buffer_subdata` perform a simplified transfer for simple writes. Basically transfer_map, data write, and transfer_unmap all in one.

The box parameter to some of these functions defines a 1D, 2D or 3D region of pixels. This is self-explanatory for 1D, 2D and 3D texture targets.

For PIPE_TEXTURE_1D_ARRAY and PIPE_TEXTURE_2D_ARRAY, the box::z and box::depth fields refer to the array dimension of the texture.

For PIPE_TEXTURE_CUBE, the box:z and box::depth fields refer to the faces of the cube map (z + depth <= 6).

For PIPE_TEXTURE_CUBE_ARRAY, the box:z and box::depth fields refer to both the face and array dimension of the texture (face = z % 6, array = z / 6).

transfer_flush_region

If a transfer was created with `FLUSH_EXPLICIT`, it will not automatically be flushed on write or unmap. Flushes must be requested with `transfer_flush_region`. Flush ranges are relative to the mapped range, not the beginning of the resource.

texture_barrier

This function flushes all pending writes to the currently-set surfaces and invalidates all read caches of the currently-set samplers. This can be used for both regular textures as well as for framebuffers read via FBFETCH.

memory_barrier

This function flushes caches according to which of the PIPE_BARRIER_* flags are set.

resource_commit

This function changes the commit state of a part of a sparse resource. Sparse resources are created by setting the `PIPE_RESOURCE_FLAG_SPARSE` flag when calling `resource_create`. Initially, sparse resources only reserve a virtual memory region that is not backed by memory (i.e., it is uncommitted). The `resource_commit` function can be called to commit or uncommit parts (or all) of a resource. The driver manages the underlying backing memory.

The contents of newly committed memory regions are undefined. Calling this function to commit an already committed memory region is allowed and leaves its content unchanged. Similarly, calling this function to uncommit an already uncommitted memory region is allowed.

For buffers, the given box must be aligned to multiples of `PIPE_CAP_SPARSE_BUFFER_PAGE_SIZE`. As an exception to this rule, if the size of the buffer is not a multiple of the page size, changing the commit state of the last (partial) page requires a box that ends at the end of the buffer (i.e., box->x + box->width == buffer->width0).

PIPE_MAP

These flags control the behavior of a transfer object.

`PIPE_MAP_READ` Resource contents read back (or accessed directly) at transfer create time.

`PIPE_MAP_WRITE` Resource contents will be written back at transfer_unmap time (or modified as a result of being accessed directly).

`PIPE_MAP_DIRECTLY` a transfer should directly map the resource. May return NULL if not supported.

`PIPE_MAP_DISCARD_RANGE` The memory within the mapped region is discarded. Cannot be used with `PIPE_MAP_READ`.

`PIPE_MAP_DISCARD_WHOLE_RESOURCE` Discards all memory backing the resource. It should not be used with `PIPE_MAP_READ`.

`PIPE_MAP_DONTBLOCK` Fail if the resource cannot be mapped immediately.

`PIPE_MAP_UNSYNCHRONIZED` Do not synchronize pending operations on the resource when mapping. The interaction of any writes to the map and any operations pending on the resource are undefined. Cannot be used with `PIPE_MAP_READ`.

`PIPE_MAP_FLUSH_EXPLICIT` Written ranges will be notified later with *transfer_flush_region*. Cannot be used with `PIPE_MAP_READ`.

`PIPE_MAP_PERSISTENT` Allows the resource to be used for rendering while mapped. PIPE_RESOURCE_FLAG_MAP_PERSISTENT must be set when creating the resource. If COHERENT is not set, memory_barrier(PIPE_BARRIER_MAPPED_BUFFER) must be called to ensure the device can see what the CPU has written.

PIPE_MAP_COHERENT If PERSISTENT is set, this ensures any writes done by the device are immediately visible to the CPU and vice versa. PIPE_RESOURCE_FLAG_MAP_COHERENT must be set when creating the resource.

Compute kernel execution

A compute program can be defined, bound or destroyed using `create_compute_state`, `bind_compute_state` or `destroy_compute_state` respectively.

Any of the subroutines contained within the compute program can be executed on the device using the `launch_grid` method. This method will execute as many instances of the program as elements in the specified N-dimensional grid, hopefully in parallel.

The compute program has access to four special resources:

- `GLOBAL` represents a memory space shared among all the threads running on the device. An arbitrary buffer created with the `PIPE_BIND_GLOBAL` flag can be mapped into it using the `set_global_binding` method.
- `LOCAL` represents a memory space shared among all the threads running in the same working group. The initial contents of this resource are undefined.
- `PRIVATE` represents a memory space local to a single thread. The initial contents of this resource are undefined.
- `INPUT` represents a read-only memory space that can be initialized at `launch_grid` time.

These resources use a byte-based addressing scheme, and they can be accessed from the compute program by means of the LOAD/STORE TGSI opcodes. Additional resources to be accessed using the same opcodes may be specified by the user with the `set_compute_resources` method.

In addition, normal texture sampling is allowed from the compute program: `bind_sampler_states` may be used to set up texture samplers for the compute stage and `set_sampler_views` may be used to bind a number of sampler views to it.

Mipmap generation

If PIPE_CAP_GENERATE_MIPMAP is true, `generate_mipmap` can be used to generate mipmaps for the specified texture resource. It replaces texel image levels base_level+1 through last_level for layers range from first_layer through last_layer. It returns TRUE if mipmap generation succeeds, otherwise it returns FALSE. Mipmap generation may fail when it is not supported for particular texture types or formats.

Device resets

Gallium frontends can query or request notifications of when the GPU is reset for whatever reason (application error, driver error). When a GPU reset happens, the context becomes unusable and all related state should be considered lost and undefined. Despite that, context notifications are single-shot, i.e. subsequent calls to `get_device_reset_status` will return PIPE_NO_RESET.

- `get_device_reset_status` queries whether a device reset has happened since the last call or since the last notification by callback.
- `set_device_reset_callback` sets a callback which will be called when a device reset is detected. The callback is only called synchronously.

Bindless

If PIPE_CAP_BINDLESS_TEXTURE is TRUE, the following `pipe_context` functions are used to create/delete bindless handles, and to make them resident in the current context when they are going to be used by shaders.

- `create_texture_handle` creates a 64-bit unsigned integer texture handle that is going to be directly used in shaders.
- `delete_texture_handle` deletes a 64-bit unsigned integer texture handle.
- `make_texture_handle_resident` makes a 64-bit unsigned texture handle resident in the current context to be accessible by shaders for texture mapping.
- `create_image_handle` creates a 64-bit unsigned integer image handle that is going to be directly used in shaders.
- `delete_image_handle` deletes a 64-bit unsigned integer image handle.
- `make_image_handle_resident` makes a 64-bit unsigned integer image handle resident in the current context to be accessible by shaders for image loads, stores and atomic operations.

34.7.2 Using several contexts

Several contexts from the same screen can be used at the same time. Objects created on one context cannot be used in another context, but the objects created by the screen methods can be used by all contexts.

Transfers

A transfer on one context is not expected to synchronize properly with rendering on other contexts, thus only areas not yet used for rendering should be locked.

A flush is required after transfer_unmap to expect other contexts to see the uploaded data, unless:

- Using persistent mapping. Associated with coherent mapping, unmapping the resource is also not required to use it in other contexts. Without coherent mapping, memory_barrier(PIPE_BARRIER_MAPPED_BUFFER) should be called on the context that has mapped the resource. No flush is required.
- Mapping the resource with PIPE_MAP_DIRECTLY.

34.8 CSO

CSO, Constant State Objects, are a core part of Gallium's API.

CSO work on the principle of reusable state; they are created by filling out a state object with the desired properties, then passing that object to a context. The context returns an opaque context-specific handle which can be bound at any time for the desired effect.

34.8.1 Blend

This state controls blending of the final fragments into the target rendering buffers.

Blend Factors

The blend factors largely follow the same pattern as their counterparts in other modern and legacy drawing APIs.

Dual source blend factors are supported for up to 1 MRT, although you can advertise > 1 MRT, the stack cannot handle them for a few reasons. There is no definition on how the 1D array of shader outputs should be mapped to something that would be a 2D array (location, index). No current hardware exposes > 1 MRT, and we should revisit this issue if anyone ever does.

Logical Operations

Logical operations, also known as logicops, LOPs, or ROPs, are supported. Only two-operand logicops are available. When logicops are enabled, all other blend state is ignored, including per-render-target state, so logicops are performed on all render targets.

Warning: The blend_enable flag is ignored for all render targets when logical operations are enabled.

For a source component s and destination component d, the logical operations are defined as taking the bits of each channel of each component, and performing one of the following operations per-channel:

- `CLEAR`: 0
- `NOR`: $\neg(s \vee d)$
- `AND_INVERTED`: $\neg s \wedge d$
- `COPY_INVERTED`: $\neg s$
- `AND_REVERSE`: $s \wedge \neg d$
- `INVERT`: $\neg d$
- `XOR`: $s \oplus d$
- `NAND`: $\neg(s \wedge d)$
- `AND`: $s \wedge d$
- `EQUIV`: $\neg(s \oplus d)$
- `NOOP`: d
- `OR_INVERTED`: $\neg s \vee d$
- `COPY`: s
- `OR_REVERSE`: $s \vee \neg d$
- `OR`: $s \vee d$
- `SET`: 1

Note: The logical operation names and definitions match those of the OpenGL API, and are similar to the ROP2 and ROP3 definitions of GDI. This is intentional, to ease transitions to Gallium.

Members

These members affect all render targets.

dither

Whether dithering is enabled.

Note: Dithering is completely implementation-dependent. It may be ignored by drivers for any reason, and some render targets may always or never be dithered depending on their format or usage flags.

logicop_enable

Whether the blender should perform a logicop instead of blending.

logicop_func

The logicop to use. One of `PIPE_LOGICOP`.

independent_blend_enable If enabled, blend state is different for each render target, and for each render target set in the respective member of the rt array. If disabled, blend state is the same for all render targets, and only the first member of the rt array contains valid data.

rt Contains the per-rendertarget blend state.

alpha_to_coverage If enabled, the fragment's alpha value is used to override the fragment's coverage mask. The coverage mask will be all zeros if the alpha value is zero. The coverage mask will be all ones if the alpha value is one. Otherwise, the number of bits set in the coverage mask will be proportional to the alpha value. Note that this step happens regardless of whether multisample is enabled or the destination buffer is multisampled.

alpha_to_one If enabled, the fragment's alpha value will be set to one. As with alpha_to_coverage, this step happens regardless of whether multisample is enabled or the destination buffer is multisampled.

max_rt The index of the max render target (irrespecitive of whether independent blend is enabled), ie. the number of MRTs minus one. This is provided so that the driver can avoid the overhead of programming unused MRTs.

Per-rendertarget Members

blend_enable If blending is enabled, perform a blend calculation according to blend functions and source/destination factors. Otherwise, the incoming fragment color gets passed unmodified (but colormask still applies).

rgb_func The blend function to use for rgb channels. One of PIPE_BLEND.

rgb_src_factor The blend source factor to use for rgb channels. One of PIPE_BLENDFACTOR.

rgb_dst_factor The blend destination factor to use for rgb channels. One of PIPE_BLENDFACTOR.

alpha_func The blend function to use for the alpha channel. One of PIPE_BLEND.

alpha_src_factor The blend source factor to use for the alpha channel. One of PIPE_BLENDFACTOR.

alpha_dst_factor The blend destination factor to use for alpha channel. One of PIPE_BLENDFACTOR.

colormask Bitmask of which channels to write. Combination of PIPE_MASK bits.

34.8.2 Depth, Stencil, & Alpha

These three states control the depth, stencil, and alpha tests, used to discard fragments that have passed through the fragment shader.

Traditionally, these three tests have been clumped together in hardware, so they are all stored in one structure.

During actual execution, the order of operations done on fragments is always:

- Alpha
- Stencil
- Depth

Depth Members

enabled Whether the depth test is enabled.

writemask Whether the depth buffer receives depth writes.

func The depth test function. One of PIPE_FUNC.

Stencil Members

enabled Whether the stencil test is enabled. For the second stencil, whether the two-sided stencil is enabled. If two-sided stencil is disabled, the other fields for the second array member are not valid.

func The stencil test function. One of PIPE_FUNC.

valuemask Stencil test value mask; this is ANDed with the value in the stencil buffer and the reference value before doing the stencil comparison test.

writemask Stencil test writemask; this controls which bits of the stencil buffer are written.

fail_op The operation to carry out if the stencil test fails. One of PIPE_STENCIL_OP.

zfail_op The operation to carry out if the stencil test passes but the depth test fails. One of PIPE_STENCIL_OP.

zpass_op The operation to carry out if the stencil test and depth test both pass. One of PIPE_STENCIL_OP.

Alpha Members

enabled Whether the alpha test is enabled.

func The alpha test function. One of PIPE_FUNC.

ref_value Alpha test reference value; used for certain functions.

34.8.3 Rasterizer

The rasterizer state controls the rendering of points, lines and triangles. Attributes include polygon culling state, line width, line stipple, multisample state, scissoring and flat/smooth shading.

Linkage

clamp_vertex_color

If set, TGSI_SEMANTIC_COLOR registers are clamped to the [0, 1] range after the execution of the vertex shader, before being passed to the geometry shader or fragment shader.

OpenGL: glClampColor(GL_CLAMP_VERTEX_COLOR) in GL 3.0 or GL_ARB_color_buffer_float

D3D11: seems always disabled

Note the PIPE_CAP_VERTEX_COLOR_CLAMPED query indicates whether or not the driver supports this control. If it's not supported, gallium frontends may have to insert extra clamping code.

clamp_fragment_color

Controls whether TGSI_SEMANTIC_COLOR outputs of the fragment shader are clamped to [0, 1].

OpenGL: glClampColor(GL_CLAMP_FRAGMENT_COLOR) in GL 3.0 or ARB_color_buffer_float

D3D11: seems always disabled

Note the PIPE_CAP_FRAGMENT_COLOR_CLAMPED query indicates whether or not the driver supports this control. If it's not supported, gallium frontends may have to insert extra clamping code.

Shading

flatshade

If set, the provoking vertex of each polygon is used to determine the color of the entire polygon. If not set, fragment colors will be interpolated between the vertex colors.

The actual interpolated shading algorithm is obviously implementation-dependent, but will usually be Gourard for most hardware.

Note: This is separate from the fragment shader input attributes CONSTANT, LINEAR and PERSPECTIVE. The flatshade state is needed at clipping time to determine how to set the color of new vertices.

Draw can implement flat shading by copying the provoking vertex color to all the other vertices in the primitive.

---

flatshade_first

Whether the first vertex should be the provoking vertex, for most primitives. If not set, the last vertex is the provoking vertex.

There are a few important exceptions to the specification of this rule.

- `PIPE_PRIMITIVE_POLYGON`: The provoking vertex is always the first vertex. If the caller wishes to change the provoking vertex, they merely need to rotate the vertices themselves.
- `PIPE_PRIMITIVE_QUAD`, `PIPE_PRIMITIVE_QUAD_STRIP`: The option only has an effect if `PIPE_CAP_QUADS_FOLLOW_PROVOKING_VERTEX_CONVENTION` is true. If it is not, the provoking vertex is always the last vertex.
- `PIPE_PRIMITIVE_TRIANGLE_FAN`: When set, the provoking vertex is the second vertex, not the first. This permits each segment of the fan to have a different color.

Polygons

light_twoside

If set, there are per-vertex back-facing colors. The hardware (perhaps assisted by *Draw*) should be set up to use this state along with the front/back information to set the final vertex colors prior to rasterization.

The frontface vertex shader color output is marked with TGSI semantic COLOR[0], and backface COLOR[1].

front_ccw Indicates whether the window order of front-facing polygons is counter-clockwise (TRUE) or clockwise (FALSE).

cull_mode Indicates which faces of polygons to cull, either PIPE_FACE_NONE (cull no polygons), PIPE_FACE_FRONT (cull front-facing polygons), PIPE_FACE_BACK (cull back-facing polygons), or PIPE_FACE_FRONT_AND_BACK (cull all polygons).

fill_front Indicates how to fill front-facing polygons, either PIPE_POLYGON_MODE_FILL, PIPE_POLYGON_MODE_LINE or PIPE_POLYGON_MODE_POINT.

fill_back Indicates how to fill back-facing polygons, either PIPE_POLYGON_MODE_FILL, PIPE_POLYGON_MODE_LINE or PIPE_POLYGON_MODE_POINT.

poly_stipple_enable Whether polygon stippling is enabled.

poly_smooth Controls OpenGL-style polygon smoothing/antialiasing

offset_point If set, point-filled polygons will have polygon offset factors applied

offset_line If set, line-filled polygons will have polygon offset factors applied

offset_tri If set, filled polygons will have polygon offset factors applied

offset_units Specifies the polygon offset bias

offset_units_unscaled Specifies the unit of the polygon offset bias. If false, use the GL/D3D1X behavior. If true, offset_units is a floating point offset which isn't scaled (D3D9). Note that GL/D3D1X behavior has different formula whether the depth buffer is unorm or float, which is not the case for D3D9.

offset_scale Specifies the polygon offset scale

offset_clamp Upper (if > 0) or lower (if < 0) bound on the polygon offset result

Lines

line_width The width of lines.

line_smooth Whether lines should be smoothed. Line smoothing is simply anti-aliasing.

line_stipple_enable Whether line stippling is enabled.

line_stipple_pattern 16-bit bitfield of on/off flags, used to pattern the line stipple.

line_stipple_factor When drawing a stippled line, each bit in the stipple pattern is repeated N times, where N = line_stipple_factor + 1.

line_last_pixel Controls whether the last pixel in a line is drawn or not. OpenGL omits the last pixel to avoid double-drawing pixels at the ends of lines when drawing connected lines.

Points

sprite_coord_enable

The effect of this state depends on PIPE_CAP_TGSI_TEXCOORD !

Controls automatic texture coordinate generation for rendering sprite points.

If PIPE_CAP_TGSI_TEXCOORD is false: When bit k in the sprite_coord_enable bitfield is set, then generic input k to the fragment shader will get an automatically computed texture coordinate.

If PIPE_CAP_TGSI_TEXCOORD is true: The bitfield refers to inputs with TEXCOORD semantic instead of generic inputs.

The texture coordinate will be of the form (s, t, 0, 1) where s varies from 0 to 1 from left to right while t varies from 0 to 1 according to the state of 'sprite_coord_mode' (see below).

If any bit is set, then point_smooth MUST be disabled (there are no round sprites) and point_quad_rasterization MUST be true (sprites are always rasterized as quads). Any mismatch between these states should be considered a bug in the gallium frontend.

This feature is implemented in the *Draw* module but may also be implemented natively by GPUs or implemented with a geometry shader.

sprite_coord_mode

Specifies how the value for each shader output should be computed when drawing point sprites. For PIPE_SPRITE_COORD_LOWER_LEFT, the lower-left vertex will have coordinates (0,0,0,1). For PIPE_SPRITE_COORD_UPPER_LEFT, the upper-left vertex will have coordinates (0,0,0,1). This state is used by *Draw* to generate texcoords.

point_quad_rasterization

Determines if points should be rasterized according to quad or point rasterization rules.

(Legacy-only) OpenGL actually has quite different rasterization rules for points and point sprites - hence this indicates if points should be rasterized as points or according to point sprite (which decomposes them into quads, basically) rules. Newer GL versions no longer support the old point rules at all.

Additionally Direct3D will always use quad rasterization rules for points, regardless of whether point sprites are enabled or not.

If this state is enabled, point smoothing and antialiasing are disabled. If it is disabled, point sprite coordinates are not generated.

Note: Some renderers always internally translate points into quads; this state still affects those renderers by overriding other rasterization state.

---

point_tri_clip Determines if clipping of points should happen after they are converted to "rectangles" (required by d3d) or before (required by OpenGL, though this rule is ignored by some IHVs). It is not valid to set this to enabled but have point_quad_rasterization disabled.

point_smooth Whether points should be smoothed. Point smoothing turns rectangular points into circles or ovals.

point_size_per_vertex Whether the vertex shader is expected to have a point size output. Undefined behavior is permitted if there is disagreement between this flag and the actual bound shader.

point_size The size of points, if not specified per-vertex.

Other Members

scissor Whether the scissor test is enabled.

multisample Whether *MSAA* is enabled.

half_pixel_center When true, the rasterizer should use (0.5, 0.5) pixel centers for determining pixel ownership (e.g, OpenGL, D3D10 and higher):

```
    0 0.5 1
 0  +-----+
    |     |
0.5 |  X  |
    |     |
 1  +-----+
```

When false, the rasterizer should use (0, 0) pixel centers for determining pixel ownership (e.g., D3D9 or ealier):

```
   -0.5 0 0.5
-0.5 +-----+
     |     |
  0  |  X  |
     |     |
 0.5 +-----+
```

bottom_edge_rule Determines what happens when a pixel sample lies precisely on a triangle edge.

When true, a pixel sample is considered to lie inside of a triangle if it lies on the *bottom edge* or *left edge* (e.g., OpenGL drawables):

```
  0                     x
0 +---------------------->
  |
  |  +-------------+
  |  |             |
  |  |             |
  |  |             |
  |  +=============+
  |
y V
```

When false, a pixel sample is considered to lie inside of a triangle if it lies on the *top edge* or *left edge* (e.g., OpenGL FBOs, D3D):

```
  0                     x
0 +---------------------->
  |
  |  +=============+
  |  |             |
  |  |             |
  |  |             |
  |  +-------------+
  |
y V
```

Where:

- a *top edge* is an edge that is horizontal and is above the other edges;
- a *bottom edge* is an edge that is horizontal and is below the other edges;
- a *left edge* is an edge that is not horizontal and is on the left side of the triangle.

Note: Actually all graphics APIs use a top-left rasterization rule for pixel ownership, but their notion of top varies with the axis origin (which can be either at y = 0 or at y = height). Gallium instead always assumes that top is always at y=0.

See also:

- http://msdn.microsoft.com/en-us/library/windows/desktop/cc627092.aspx
- http://msdn.microsoft.com/en-us/library/windows/desktop/bb147314.aspx

clip_halfz When true clip space in the z axis goes from [0..1] (D3D). When false [-1, 1] (GL)

depth_clip When false, the near and far depth clipping planes of the view volume are disabled and the depth value will be clamped at the per-pixel level, after polygon offset has been applied and before depth testing.

clip_plane_enable For each k in [0, PIPE_MAX_CLIP_PLANES), if bit k of this field is set, clipping half-space k is enabled, if it is clear, it is disabled. The clipping half-spaces are defined either by the user clip planes in `pipe_clip_state`, or by the clip distance outputs of the shader stage preceding the fragment shader. If any clip distance output is written, those half-spaces for which no clip distance is written count as disabled; i.e. user clip planes and shader clip distances cannot be mixed, and clip distances take precedence.

conservative_raster_mode The conservative rasterization mode. For PIPE_CONSERVATIVE_RASTER_OFF, conservative rasterization is disabled. For IPE_CONSERVATIVE_RASTER_POST_SNAP or PIPE_CONSERVATIVE_RASTER_PRE_SNAP, conservative rasterization is nabled. When conservative rasterization is enabled, the polygon smooth, line mooth, point smooth and line stipple settings are ignored. With the post-snap mode, unlike the pre-snap mode, fragments are never generated for degenerate primitives. Degenerate primitives, when rasterized, are considered back-facing and the vertex attributes and depth are that of the provoking vertex. If the post-snap mode is used with an unsupported primitive, the pre-snap mode is used, if supported. Behavior is similar for the pre-snap mode. If the pre-snap mode is used, fragments are generated with respect to the primitive before vertex snapping.

conservative_raster_dilate The amount of dilation during conservative rasterization.

subpixel_precision_x A bias added to the horizontal subpixel precision during conservative rasterization.

subpixel_precision_y A bias added to the vertical subpixel precision during conservative rasterization.

34.8.4 Sampler

Texture units have many options for selecting texels from loaded textures; this state controls an individual texture unit's texel-sampling settings.

Texture coordinates are always treated as four-dimensional, and referred to with the traditional (S, T, R, Q) notation.

Members

wrap_s How to wrap the S coordinate. One of PIPE_TEX_WRAP_*.

wrap_t How to wrap the T coordinate. One of PIPE_TEX_WRAP_*.

wrap_r How to wrap the R coordinate. One of PIPE_TEX_WRAP_*.

The wrap modes are:

- PIPE_TEX_WRAP_REPEAT: Standard coord repeat/wrap-around mode.
- PIPE_TEX_WRAP_CLAMP_TO_EDGE: Clamp coord to edge of texture, the border color is never sampled.
- PIPE_TEX_WRAP_CLAMP_TO_BORDER: Clamp coord to border of texture, the border color is sampled when coords go outside the range [0,1].
- PIPE_TEX_WRAP_CLAMP: The coord is clamped to the range [0,1] before scaling to the texture size. This corresponds to the legacy OpenGL GL_CLAMP texture wrap mode. Historically, this mode hasn't acted consistently across all graphics hardware. It sometimes acts like CLAMP_TO_EDGE or CLAMP_TO_BORDER. The behavior may also vary depending on linear vs. nearest sampling mode.
- PIPE_TEX_WRAP_MIRROR_REPEAT: If the integer part of the coordinate is odd, the coord becomes (1 - coord). Then, normal texture REPEAT is applied to the coord.
- PIPE_TEX_WRAP_MIRROR_CLAMP_TO_EDGE: First, the absolute value of the coordinate is computed. Then, regular CLAMP_TO_EDGE is applied to the coord.
- PIPE_TEX_WRAP_MIRROR_CLAMP_TO_BORDER: First, the absolute value of the coordinate is computed. Then, regular CLAMP_TO_BORDER is applied to the coord.
- PIPE_TEX_WRAP_MIRROR_CLAMP: First, the absolute value of the coord is computed. Then, regular CLAMP is applied to the coord.

min_img_filter The image filter to use when minifying texels. One of PIPE_TEX_FILTER_*.

mag_img_filter The image filter to use when magnifying texels. One of PIPE_TEX_FILTER_*.

The texture image filter modes are:

- PIPE_TEX_FILTER_NEAREST: One texel is fetched from the texture image at the texture coordinate.
- PIPE_TEX_FILTER_LINEAR: Two, four or eight texels (depending on the texture dimensions; 1D/2D/3D) are fetched from the texture image and linearly weighted and blended together.

min_mip_filter The filter to use when minifying mipmapped textures. One of PIPE_TEX_MIPFILTER_*.

The texture mip filter modes are:

- PIPE_TEX_MIPFILTER_NEAREST: A single mipmap level/image is selected according to the texture LOD (lambda) value.
- PIPE_TEX_MIPFILTER_LINEAR: The two mipmap levels/images above/below the texture LOD value are sampled from. The results of sampling from those two images are blended together with linear interpolation.
- PIPE_TEX_MIPFILTER_NONE: Mipmap filtering is disabled. All texels are taken from the level 0 image.

compare_mode If set to PIPE_TEX_COMPARE_R_TO_TEXTURE, the result of texture sampling is not a color but a true/false value which is the result of comparing the sampled texture value (typically a Z value from a depth texture) to the texture coordinate's R component. If set to PIPE_TEX_COMPARE_NONE, no comparison calculation is performed.

compare_func The inequality operator used when compare_mode=1. One of PIPE_FUNC_x.

normalized_coords If set, the incoming texture coordinates (nominally in the range [0,1]) will be scaled by the texture width, height, depth to compute texel addresses. Otherwise, the texture coords are used as-is (they are not scaled by the texture dimensions). When normalized_coords=0, only a subset of the texture wrap modes are allowed: PIPE_TEX_WRAP_CLAMP, PIPE_TEX_WRAP_CLAMP_TO_EDGE and PIPE_TEX_WRAP_CLAMP_TO_BORDER.

lod_bias Bias factor which is added to the computed level of detail. The normal level of detail is computed from the partial derivatives of the texture coordinates and/or the fragment shader TEX/TXB/TXL instruction.

min_lod Minimum level of detail, used to clamp LOD after bias. The LOD values correspond to mipmap levels where LOD=0 is the level 0 mipmap image.

max_lod Maximum level of detail, used to clamp LOD after bias.

border_color Color union used for texel coordinates that are outside the [0,width-1], [0, height-1] or [0, depth-1] ranges. Interpreted according to sampler view format, unless the driver reports PIPE_CAP_TEXTURE_BORDER_COLOR_QUIRK, in which case special care has to be taken (see description of the cap).

max_anisotropy Maximum anistropy ratio to use when sampling from textures. For example, if max_anistropy=4, a region of up to 1 by 4 texels will be sampled. Set to zero to disable anisotropic filtering. Any other setting enables anisotropic filtering, however it's not unexpected some drivers only will change their filtering with a setting of 2 and higher.

seamless_cube_map If set, the bilinear filter of a cube map may take samples from adjacent cube map faces when sampled near a texture border to produce a seamless look.

34.8.5 Shader

One of the two types of shaders supported by Gallium.

Members

tokens A list of tgsi_tokens.

34.8.6 Vertex Elements

This state controls the format of the input attributes contained in pipe_vertex_buffers. There is one pipe_vertex_element array member for each input attribute.

Input Formats

Gallium supports a diverse range of formats for vertex data. Drivers are guaranteed to support 32-bit floating-point vectors of one to four components. Additionally, they may support the following formats:

- Integers, signed or unsigned, normalized or non-normalized, 8, 16, or 32 bits wide
- Floating-point, 16, 32, or 64 bits wide

At this time, support for varied vertex data formats is limited by driver deficiencies. It is planned to support a single uniform set of formats for all Gallium drivers at some point.

Rather than attempt to specify every small nuance of behavior, Gallium uses a very simple set of rules for padding out unspecified components. If an input uses less than four components, it will be padded out with the constant vector `(0, 0, 0, 1)`.

Fog, point size, the facing bit, and edgeflags, all are in the standard format of `(x, 0, 0, 1)`, and so only the first component of those inputs is used.

Position

Vertex position may be specified with two to four components. Using less than two components is not allowed.

Colors

Colors, both front- and back-facing, may omit the alpha component, only using three components. Using less than three components is not allowed.

Members

src_offset The byte offset of the attribute in the buffer given by vertex_buffer_index for the first vertex.

instance_divisor The instance data rate divisor, used for instancing. 0 means this is per-vertex data, n means per-instance data used for n consecutive instances (n > 0).

vertex_buffer_index The vertex buffer this attribute lives in. Several attributes may live in the same vertex buffer.

src_format The format of the attribute data. One of the PIPE_FORMAT tokens.

34.9 Distribution

Along with the interface definitions, the following drivers, gallium frontends, and auxiliary modules are shipped in the standard Gallium distribution.

34.9.1 Drivers

Intel i915

Driver for Intel i915 and i945 chipsets.

LLVM Softpipe

A version of *Softpipe* that uses the Low-Level Virtual Machine to dynamically generate optimized rasterizing pipelines.

nVidia nv30

Driver for the nVidia nv30 and nv40 families of GPUs.

nVidia nv50

Driver for the nVidia nv50 family of GPUs.

nVidia nvc0

Driver for the nVidia nvc0 / fermi family of GPUs.

VMware SVGA

Driver for VMware virtualized guest operating system graphics processing.

ATI r300

Driver for the ATI/AMD r300, r400, and r500 families of GPUs.

ATI/AMD r600

Driver for the ATI/AMD r600, r700, Evergreen and Northern Islands families of GPUs.

AMD radeonsi

Driver for the AMD Southern Islands family of GPUs.

freedreno

Driver for Qualcomm Adreno a2xx, a3xx, and a4xx series of GPUs.

Softpipe

Reference software rasterizer. Slow but accurate.

Trace

Wrapper driver. Trace dumps an XML record of the calls made to the *Context* and *Screen* objects that it wraps.

Rbug

Wrapper driver. *Remote Debugger* driver used with stand alone rbug-gui.

34.9.2 Gallium frontends

Clover

Tracker that implements the Khronos OpenCL standard.

Direct Rendering Infrastructure

Tracker that implements the client-side DRI protocol, for providing direct acceleration services to X11 servers with the DRI extension. Supports DRI1 and DRI2. Only GL is supported.

GLX

MesaGL

The gallium frontend implementing a GL state machine. Not usable as a standalone frontend; Mesa should be built with another gallium frontend, such as *Direct Rendering Infrastructure* or EGL.

VDPAU

Tracker for Video Decode and Presentation API for Unix.

WGL

Xorg DDX

Tracker for Xorg X11 servers. Provides device-dependent modesetting and acceleration as a DDX driver.

XvMC

Tracker for X-Video Motion Compensation.

34.9.3 Auxiliary

OS

The OS module contains the abstractions for basic operating system services:

- memory allocation
- simple message logging
- obtaining run-time configuration option
- threading primitives

This is the bare minimum required to port Gallium to a new platform.

The OS module already provides the implementations of these abstractions for the most common platforms. When targeting an embedded platform no implementation will be provided – these must be provided separately.

CSO Cache

The CSO cache is used to accelerate preparation of state by saving driver-specific state structures for later use.

Draw

Draw is a software *TCL* pipeline for hardware that lacks vertex shaders or other essential parts of pre-rasterization vertex preparation.

Gallivm

Indices

Indices provides tools for translating or generating element indices for use with element-based rendering.

Pipe Buffer Managers

Each of these managers provides various services to drivers that are not fully utilizing a memory manager.

Remote Debugger

Runtime Assembly Emission

TGSI

The TGSI auxiliary module provides basic utilities for manipulating TGSI streams.

Translate

Util

34.10 Drivers

Driver specific documentation.

34.10.1 Freedreno

Freedreno driver specific docs.

IR3 NOTES

Some notes about ir3, the compiler and machine-specific IR for the shader ISA introduced with adreno a3xx. The same shader ISA is present, with some small differences, in adreno a4xx.

Compared to the previous generation a2xx ISA (ir2), the a3xx ISA is a "simple" scalar instruction set. However, the compiler is responsible, in most cases, to schedule the instructions. The hardware does not try to hide the shader core pipeline stages. For a common example, a common (cat2) ALU instruction takes four cycles, so a subsequent cat2 instruction which uses the result must have three intervening instructions (or nops). When operating on vec4's, typically the corresponding scalar instructions for operating on the remaining three components could typically fit. Although that results in a lot of edge cases where things fall over, like:

```
ADD TEMP[0], TEMP[1], TEMP[2]
MUL TEMP[0], TEMP[1], TEMP[0].wzyx
```

Here, the second instruction needs the output of the first group of scalar instructions in the wrong order, resulting in not enough instruction spots between the `add r0.w, r1.w, r2.w` and `mul r0.x, r1.x, r0.w`. Which is why the original (old) compiler which merely translated nearly literally from TGSI to ir3, had a strong tendency to fall over.

So the current compiler instead, in the frontend, generates a directed-acyclic-graph of instructions and basic blocks, which go through various additional passes to eventually schedule and do register assignment.

For additional documentation about the hardware, see wiki: a3xx ISA.

External Structure

`ir3_shader` A single vertex/fragment/etc shader from gallium perspective (ie. maps to a single TGSI shader), and manages a set of shader variants which are generated on demand based on the shader key.

`ir3_shader_key` The configuration key that identifies a shader variant. Ie. based on other GL state (two-sided-color, render-to-alpha, etc) or render stages (binning-pass vertex shader) different shader variants are generated.

ir3_shader_variant The actual hw shader generated based on input TGSI and shader key.

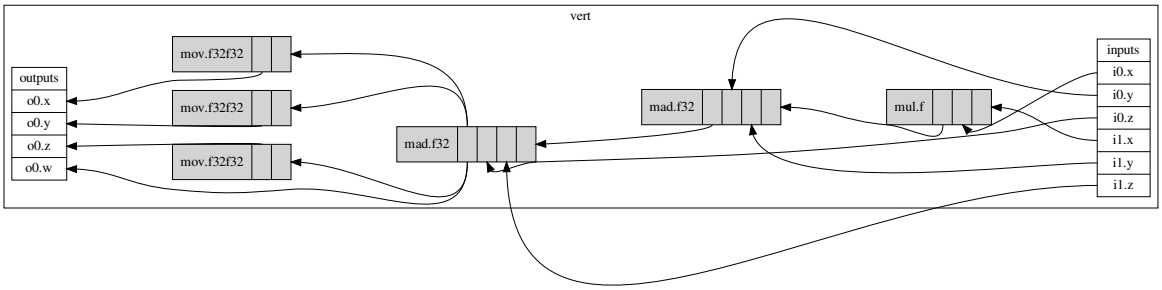
ir3_compiler Compiler frontend which generates ir3 and runs the various backend stages to schedule and do register assignment.

The IR

The ir3 IR maps quite directly to the hardware, in that instruction opcodes map directly to hardware opcodes, and that dst/src register(s) map directly to the hardware dst/src register(s). But there are a few extensions, in the form of *meta* instructions. And additionally, for normal (non-const, etc) src registers, the `IR3_REG_SSA` flag is set and `reg->instr` points to the source instruction which produced that value. So, for example, the following TGSI shader:

```
VERT
DCL IN[0]
DCL IN[1]
DCL OUT[0], POSITION
DCL TEMP[0], LOCAL
  1: DP3 TEMP[0].x, IN[0].xyzz, IN[1].xyzz
  2: MOV OUT[0], TEMP[0].xxxx
  3: END
```

eventually generates:



(after scheduling, etc, but before register assignment).

Internal Structure

ir3_block Represents a basic block.

TODO: currently blocks are nested, but I think I need to change that to a more conventional arrangement before implementing proper flow control. Currently the only flow control handles is if/else which gets flattened out and results chosen with `sel` instructions.

ir3_instruction Represents a machine instruction or *meta* instruction. Has pointers to dst register (`regs[0]`) and src register(s) (`regs[1..n]`), as needed.

ir3_register Represents a src or dst register, flags indicate const/relative/etc. If `IR3_REG_SSA` is set on a src register, the actual register number (name) has not been assigned yet, and instead the `instr` field points to src instruction.

In addition there are various util macros/functions to simplify manipulation/traversal of the graph:

foreach_src(srcreg, instr) Iterate each instruction's source `ir3_registers`

foreach_src_n(srcreg, n, instr) Like `foreach_src`, also setting `n` to the source number (starting with `0`).

foreach_ssa_src(srcinstr, instr) Iterate each instruction's SSA source `ir3_instructions`. This skips non-SSA sources (consts, etc), but includes virtual sources (such as the address register if *relative addressing* is used).

foreach_ssa_src_n(srcinstr, n, instr) Like `foreach_ssa_src`, also setting `n` to the source number.

For example:

```
foreach_ssa_src_n(src, i, instr) {
  unsigned d = delay_calc_srcn(ctx, src, instr, i);
  delay = MAX2(delay, d);
}
```

TODO probably other helper/util stuff worth mentioning here

Meta Instructions

input Used for shader inputs (registers configured in the command-stream to hold particular input values, written by the shader core before start of execution. Also used for connecting up values within a basic block to an output of a previous block.

output Used to hold outputs of a basic block.

flow TODO

phi TODO

fanin Groups registers which need to be assigned to consecutive scalar registers, for example *sam* (texture fetch) src instructions (see *register groups*) or array element dereference (see *relative addressing*).

fanout The counterpart to **fanin**, when an instruction such as *sam* writes multiple components, splits the result into individual scalar components to be consumed by other instructions.

Flow Control

TODO

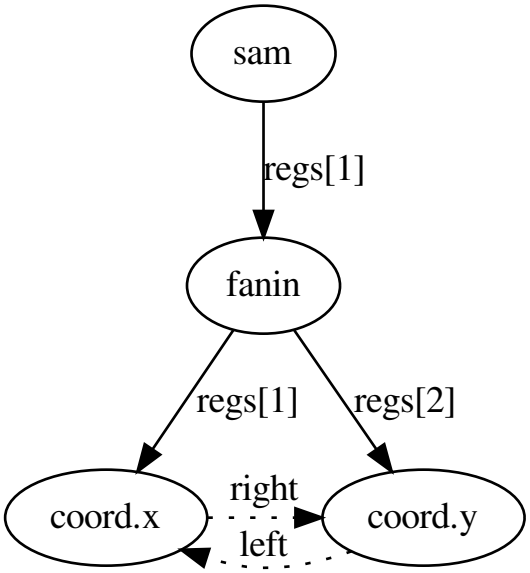
Register Groups

Certain instructions, such as texture sample instructions, consume multiple consecutive scalar registers via a single src register encoded in the instruction, and/or write multiple consecutive scalar registers. In the simplest example:

```
sam (f32)(xyz)r2.x, r0.z, s#0, t#0
```

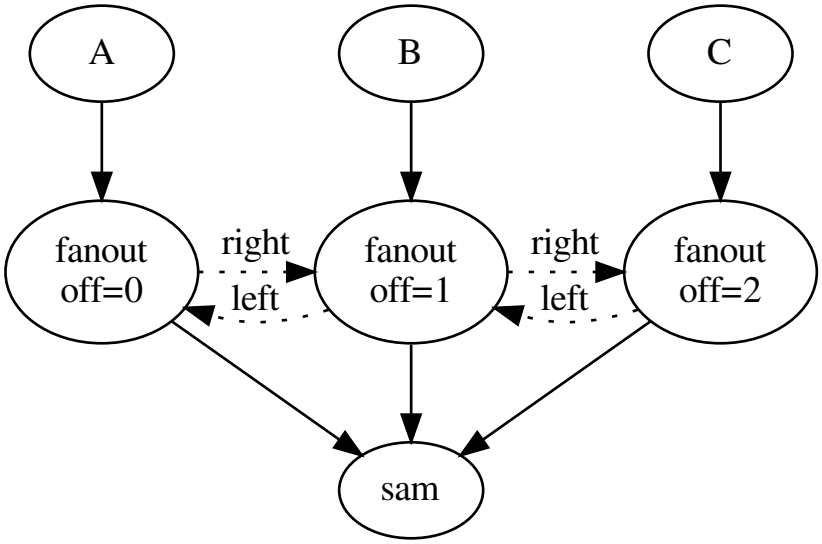
for a 2d texture, would read `r0.zw` to get the coordinate, and write `r2.xyz`.

Before register assignment, to group the two components of the texture src together:



The frontend sets up the SSA ptrs from `sam` source register to the `fanin` meta instruction, which in turn points to the instructions producing the `coord.x` and `coord.y` values. And the *grouping* pass sets up the `left` and `right` neighbor pointers to the `fanin`'s sources, used later by the *register assignment* pass to assign blocks of scalar registers.

And likewise, for the consecutive scalar registers for the destination:



Relative Addressing

Most instructions support addressing indirectly (relative to address register) into const or gpr register file in some or all of their src/dst registers. In this case the register accessed is taken from `r<a0.x + n>` or `c<a0.x + n>`, ie. address register (`a0.x`) value plus `n`, where `n` is encoded in the instruction (rather than the absolute register number).

> Note that cat5 (texture sample) instructions are the notable exception, not supporting relative addressing of src or dst.

Relative addressing of the const file (for example, a uniform array) is relatively simple. We don't do register assignment of the const file, so all that is required is to schedule things properly. Ie. the instruction that writes the address register must be scheduled first, and we cannot have two different address register values live at one time.

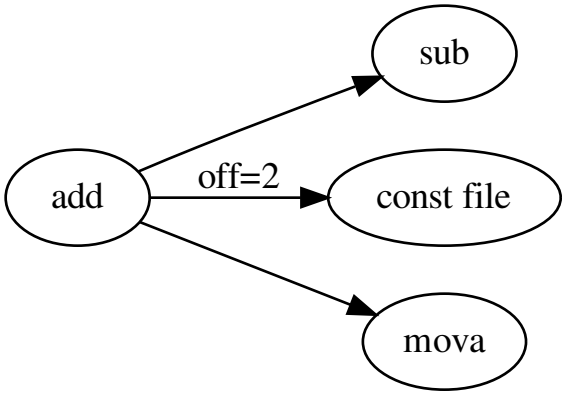
But relative addressing of gpr file (which can be as src or dst) has additional restrictions on register assignment (ie. the array elements must be assigned to consecutive scalar registers). And in the case of relative dst, subsequent instructions now depend on both the relative write, as well as the previous instruction which wrote that register, since we do not know at compile time which actual register was written.

Each instruction has an optional `address` pointer, to capture the dependency on the address register value when relative addressing is used for any of the src/dst register(s). This behaves as an additional virtual src register, ie. `foreach_ssa_src()` will also iterate the address register (last).

> Note that `nop`'s for timing constraints, type specifiers (ie. `add.f` vs `add.u`), etc, omitted for brevity in examples

```
mova a0.x, hr1.y
sub r1.y, r2.x, r3.x
add r0.x, r1.y, c<a0.x + 2>
```

results in:



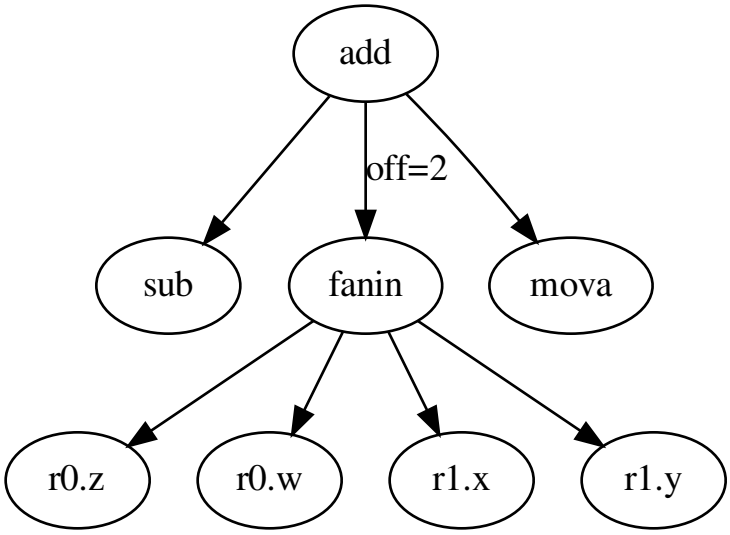
The scheduling pass has some smarts to schedule things such that only a single `a0.x` value is used at any one time.

To implement variable arrays, values are stored in consecutive scalar registers. This has some overlap with *register groups*, in that `fanin` and `fanout` are used to help group things for the *register assignment* pass.

To use a variable array as a src register, a slight variation of what is done for const array src. The instruction src is a *fanin* instruction that groups all the array members:

```
mova a0.x, hr1.y
sub r1.y, r2.x, r3.x
add r0.x, r1.y, r<a0.x + 2>
```

results in:



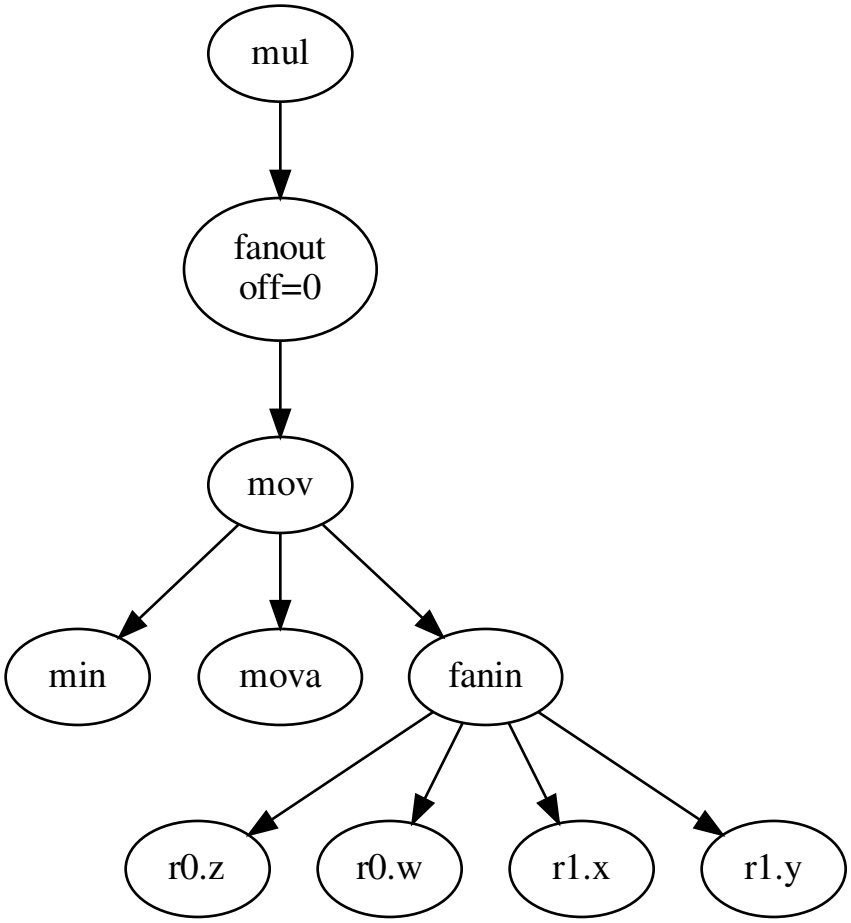
TODO better describe how actual deref offset is derived, ie. based on array base register.

To do an indirect write to a variable array, a `fanout` is used. Say the array was assigned to registers `r0.z` through `r1.y` (hence the constant offset of 2):

Note that only cat1 (mov) can do indirect write.

```
mova a0.x, hr1.y
min r2.x, r2.x, c0.x
mov r<a0.x + 2>, r2.x
mul r0.x, r0.z, c0.z
```

In this case, the `mov` instruction does not write all elements of the array (compared to usage of `fanout` for `sam` instructions in *grouping*). But the `mov` instruction does need an additional dependency (via `fanin`) on instructions that last wrote the array element members, to ensure that they get scheduled before the `mov` in *scheduling* stage (which also serves to group the array elements for the *register assignment* stage).



Note that there would in fact be `fanout` nodes generated for each array element (although only the reachable ones will be scheduled, etc).

Shader Passes

After the frontend has generated the use-def graph of instructions, they are run through various passes which include *scheduling* and *register assignment*. Because inserting `mov` instructions after scheduling would also require inserting additional `nop` instructions (since it is too late to reschedule to try and fill the bubbles), the earlier stages try to ensure that (at least given an infinite supply of registers) that *register assignment* after *scheduling* cannot fail.

> Note that we essentially have ~256 scalar registers in the architecture (although larger register usage will at some thresholds limit the number of threads which can run in parallel). And at some point we will have to deal with spilling.

Flatten

In this stage, simple if/else blocks are flattened into a single block with `phi` nodes converted into `sel` instructions. The a3xx ISA has very few predicated instructions, and we would prefer not to use branches for simple if/else.

Copy Propagation

Currently the frontend inserts `mov`s in various cases, because certain categories of instructions have limitations about const regs as sources. And the CP pass simply removes all simple `mov`s (ie. src-type is same as dst-type, no abs/neg flags, etc).

The eventual plan is to invert that, with the front-end inserting no `mov`s and CP legalize things.

Grouping

In the grouping pass, instructions which need to be grouped (for `fanin`s, etc) have their `left` / `right` neighbor pointers setup. In cases where there is a conflict (ie. one instruction cannot have two unique left or right neighbors), an additional `mov` instruction is inserted. This ensures that there is some possible valid *register assignment* at the later stages.

Depth

In the depth pass, a depth is calculated for each instruction node within it's basic block. The depth is the sum of the required cycles (delay slots needed between two instructions plus one) of each instruction plus the max depth of any of it's source instructions. (*meta* instructions don't add to the depth). As an instruction's depth is calculated, it is inserted into a per block list sorted by deepest instruction. Unreachable instructions and inputs are marked.

> TODO: we should probably calculate both hard and soft depths (?) to try to coax additional instructions to fit in places where we need to use sync bits, such as after a texture fetch or SFU.

Scheduling

After the *grouping* pass, there are no more instructions to insert or remove. Start scheduling each basic block from the deepest node in the depth sorted list created by the *depth* pass, recursively trying to schedule each instruction after it's source instructions plus delay slots. Insert `nop`s as required.

Register Assignment

TODO

34.10.2 Gallium LLVMpipe Driver

Introduction

The Gallium llvmpipe driver is a software rasterizer that uses LLVM to do runtime code generation. Shaders, point/line/triangle rasterization and vertex processing are implemented with LLVM IR which is translated to x86, x86-64, or ppc64le machine code. Also, the driver is multithreaded to take advantage of multiple CPU cores (up to 8 at this time). It's the fastest software rasterizer for Mesa.

Requirements

- For x86 or amd64 processors, 64-bit mode is recommended. Support for SSE2 is strongly encouraged. Support for SSE3 and SSE4.1 will yield the most efficient code. The fewer features the CPU has the more likely it is that you will run into underperforming, buggy, or incomplete code.

  For ppc64le processors, use of the Altivec feature (the Vector Facility) is recommended if supported; use of the VSX feature (the Vector-Scalar Facility) is recommended if supported AND Mesa is built with LLVM version 4.0 or later.

  See `/proc/cpuinfo` to know what your CPU supports.

- Unless otherwise stated, LLVM version 3.4 is recommended; 3.3 or later is required.

  For Linux, on a recent Debian based distribution do:

  ```
  aptitude install llvm-dev
  ```

  If you want development snapshot builds of LLVM for Debian and derived distributions like Ubuntu, you can use the APT repository at apt.llvm.org, which are maintained by Debian's LLVM maintainer.

  For a RPM-based distribution do:

  ```
  yum install llvm-devel
  ```

  For Windows you will need to build LLVM from source with MSVC or MINGW (either natively or through cross compilers) and CMake, and set the `LLVM` environment variable to the directory you installed it to. LLVM will be statically linked, so when building on MSVC it needs to be built with a matching CRT as Mesa, and you'll need to pass `-DLLVM_USE_CRT_xxx=yyy` as described below.

LLVM build-type	Mesa build-type	
	debug,checked	release,profile
Debug	`-DLLVM_USE_CRT_DEBUG=MTd`	`-DLLVM_USE_CRT_DEBUG=MT`
Release	`-DLLVM_USE_CRT_RELEASE=MTd`	`-DLLVM_USE_CRT_RELEASE=MT`

  You can build only the x86 target by passing `-DLLVM_TARGETS_TO_BUILD=X86` to cmake.

- scons (optional)

Building

To build everything on Linux invoke scons as:

```
scons build=debug libgl-xlib
```

Alternatively, you can build it with meson with:

```
mkdir build
cd build
meson -D glx=gallium-xlib -D gallium-drivers=swrast
ninja
```

but the rest of these instructions assume that scons is used. For Windows the procedure is similar except the target:

```
scons platform=windows build=debug libgl-gdi
```

Using

Linux

On Linux, building will create a drop-in alternative for `libGL.so` into

```
build/foo/gallium/targets/libgl-xlib/libGL.so
```

or

```
lib/gallium/libGL.so
```

To use it set the `LD_LIBRARY_PATH` environment variable accordingly.

For performance evaluation pass `build=release` to scons, and use the corresponding lib directory without the `-debug` suffix.

Windows

On Windows, building will create `build/windows-x86-debug/gallium/targets/libgl-gdi/opengl32.dll` which is a drop-in alternative for system's `opengl32.dll`. To use it put it in the same directory as your application. It can also be used by replacing the native ICD driver, but it's quite an advanced usage, so if you need to ask, don't even try it.

There is however an easy way to replace the OpenGL software renderer that comes with Microsoft Windows 7 (or later) with llvmpipe (that is, on systems without any OpenGL drivers):

- copy `build/windows-x86-debug/gallium/targets/libgl-gdi/opengl32.dll` to `C:\Windows\SysWOW64\mesadrv.dll`
- load this registry settings:

```
REGEDIT4

; https://technet.microsoft.com/en-us/library/cc749368.aspx
; https://www.msfn.org/board/topic/143241-portable-windows-7-build-from-winpe-30/
↪page-5#entry942596
[HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\Microsoft\Windows␣
↪NT\CurrentVersion\OpenGLDrivers\MSOGL]
"DLL"="mesadrv.dll"
"DriverVersion"=dword:00000001
"Flags"=dword:00000001
"Version"=dword:00000002
```

- Ditto for 64 bits drivers if you need them.

Profiling

To profile llvmpipe you should build as

```
scons build=profile <same-as-before>
```

This will ensure that frame pointers are used both in C and JIT functions, and that no tail call optimizations are done by gcc.

Linux perf integration

On Linux, it is possible to have symbol resolution of JIT code with Linux perf:

```
perf record -g /my/application
perf report
```

When run inside Linux perf, llvmpipe will create a `/tmp/perf-XXXXX.map` file with symbol address table. It also dumps assembly code to `/tmp/perf-XXXXX.map.asm`, which can be used by the `bin/perf-annotate-jit.py` script to produce disassembly of the generated code annotated with the samples.

You can obtain a call graph via Gprof2Dot.

Unit testing

Building will also create several unit tests in `build/linux-???-debug/gallium/drivers/llvmpipe`:

- `lp_test_blend`: blending
- `lp_test_conv`: SIMD vector conversion
- `lp_test_format`: pixel unpacking/packing

Some of these tests can output results and benchmarks to a tab-separated file for later analysis, e.g.:

```
build/linux-x86_64-debug/gallium/drivers/llvmpipe/lp_test_blend -o blend.tsv
```

Development Notes

- When looking at this code for the first time, start in lp_state_fs.c, and then skim through the `lp_bld_*` functions called there, and the comments at the top of the `lp_bld_*.c` functions.
- The driver-independent parts of the LLVM / Gallium code are found in `src/gallium/auxiliary/gallivm/`. The filenames and function prefixes need to be renamed from `lp_bld_` to something else though.
- We use LLVM-C bindings for now. They are not documented, but follow the C++ interfaces very closely, and appear to be complete enough for code generation. See this stand-alone example. See the `llvm-c/Core.h` file for reference.

Recommended Reading

- Rasterization
  - Triangle Scan Conversion using 2D Homogeneous Coordinates
  - Rasterization on Larrabee (DevMaster copy)
  - Rasterization using half-space functions
  - Advanced Rasterization
  - Optimizing Software Occlusion Culling
- Texture sampling
  - Perspective Texture Mapping
  - Texturing As In Unreal
  - Run-Time MIP-Map Filtering

  - Will "brilinear" filtering persist?
  - Trilinear filtering
  - Texture Swizzling
- SIMD
  - Whole-Function Vectorization
- Optimization
  - Optimizing Pixomatic For Modern x86 Processors
  - Intel 64 and IA-32 Architectures Optimization Reference Manual
  - Software optimization resources
  - Intel Intrinsics Guide
- LLVM
  - LLVM Language Reference Manual
  - The secret of LLVM C bindings
- General
  - A trip through the Graphics Pipeline
  - WARP Architecture and Performance

34.10.3 OpenSWR

The Gallium OpenSWR driver is a high performance, highly scalable software renderer targeted towards visualization workloads. For such geometry heavy workloads there is a considerable speedup over llvmpipe, which is to be expected as the geometry frontend of llvmpipe is single threaded.

This rasterizer is x86 specific and requires AVX or above. The driver fits into the gallium framework, and reuses gallivm for doing the TGSI to vectorized llvm-IR conversion of the shader kernels.

Usage

Requirements

- An x86 processor with AVX or above
- LLVM version 3.9 or later
- C++14 capable compiler

Building

To build with GNU automake, select building the swr driver at configure time, for example:

```
configure --with-gallium-drivers=swrast,swr
```

Using

On Linux, building with autotools will create a drop-in alternative for libGL.so into:

```
lib/gallium/libGL.so
lib/gallium/libswrAVX.so
lib/gallium/libswrAVX2.so
```

Alternatively, building with SCons will produce:

```
build/linux-x86_64/gallium/targets/libgl-xlib/libGL.so
build/linux-x86_64/gallium/drivers/swr/libswrAVX.so
build/linux-x86_64/gallium/drivers/swr/libswrAVX2.so
```

To use it set the LD_LIBRARY_PATH environment variable accordingly.

IMPORTANT: Mesa will default to using llvmpipe or softpipe as the default software renderer. To select the OpenSWR driver, set the GALLIUM_DRIVER environment variable appropriately:

```
GALLIUM_DRIVER=swr
```

To verify OpenSWR is being used, check to see if a message like the following is printed when the application is started:

```
SWR detected AVX2
```

FAQ

Why another software rasterizer?

Good question, given there are already three (swrast, softpipe, llvmpipe) in the Mesa tree. Two important reasons for this:

- Architecture - given our focus on scientific visualization, our workloads are much different than the typical game; we have heavy vertex load and relatively simple shaders. In addition, the core counts of machines we run on are much higher. These parameters led to design decisions much different than llvmpipe.
- Historical - Intel had developed a high performance software graphics stack for internal purposes. Later we adapted this graphics stack for use in visualization and decided to move forward with Mesa to provide a high quality API layer while at the same time benefiting from the excellent performance the software rasterizerizer gives us.

What's the architecture?

SWR is a tile based immediate mode renderer with a sort-free threading model which is arranged as a ring of queues. Each entry in the ring represents a draw context that contains all of the draw state and work queues. An API thread sets up each draw context and worker threads will execute both the frontend (vertex/geometry processing) and backend (fragment) work as required. The ring allows for backend threads to pull work in order. Large draws are split into chunks to allow vertex processing to happen in parallel, with the backend work pickup preserving draw ordering.

Our pipeline uses just-in-time compiled code for the fetch shader that does vertex attribute gathering and AOS to SOA conversions, the vertex shader and fragment shaders, streamout, and fragment blending. SWR core also supports geometry and compute shaders but we haven't exposed them through our driver yet. The fetch shader, streamout, and blend is built internally to swr core using LLVM directly, while for the vertex and pixel shaders we reuse bits of

llvmpipe from `gallium/auxiliary/gallivm` to build the kernels, which we wrap differently than llvmpipe's `auxiliary/draw` code.

What's the performance?

For the types of high-geometry workloads we're interested in, we are significantly faster than llvmpipe. This is to be expected, as llvmpipe only threads the fragment processing and not the geometry frontend. The performance advantage over llvmpipe roughly scales linearly with the number of cores available.

While our current performance is quite good, we know there is more potential in this architecture. When we switched from a prototype OpenGL driver to Mesa we regressed performance severely, some due to interface issues that need tuning, some differences in shader code generation, and some due to conformance and feature additions to the core swr. We are looking to recovering most of this performance back.

What's the conformance?

The major applications we are targeting are all based on the Visualization Toolkit (VTK), and as such our development efforts have been focused on making sure these work as best as possible. Our current code passes vtk's rendering tests with their new "OpenGL2" (really OpenGL 3.2) backend at 99%.

piglit testing shows a much lower pass rate, roughly 80% at the time of writing. Core SWR undergoes rigorous unit testing and we are quite confident in the rasterizer, and understand the areas where it currently has issues (example: line rendering is done with triangles, so doesn't match the strict line rendering rules). The majority of the piglit failures are errors in our driver layer interfacing Mesa and SWR. Fixing these issues is one of our major future development goals.

Why are you open sourcing this?

- Our customers prefer open source, and allowing them to simply download the Mesa source and enable our driver makes life much easier for them.
- The internal gallium APIs are not stable, so we'd like our driver to be visible for changes.
- It's easier to work with the Mesa community when the source we're working with can be used as reference.

What are your development plans?

- Performance - see the performance section earlier for details.
- Conformance - see the conformance section earlier for details.
- Features - core SWR has a lot of functionality we have yet to expose through our driver, such as MSAA, geometry shaders, compute shaders, and tesselation.
- AVX512 support

What is the licensing of the code?

- All code is under the normal Mesa MIT license.

Will this work on AMD?

- If using an AMD processor with AVX or AVX2, it should work though we don't have that hardware around to test. Patches if needed would be welcome.

Will this work on ARM, MIPS, POWER, <other non-x86 architecture>?

- Not without a lot of work. We make extensive use of AVX and AVX2 intrinsics in our code and the in-tree JIT creation. It is not the intention for this codebase to support non-x86 architectures.

What hardware do I need?

- Any x86 processor with at least AVX (introduced in the Intel SandyBridge and AMD Bulldozer microarchitectures in 2011) will work.
- You don't need a fire-breathing Xeon machine to work on SWR - we do day-to-day development with laptops and desktop CPUs.

Does one build work on both AVX and AVX2?

Yes. The build system creates two shared libraries, `libswrAVX.so` and `libswrAVX2.so`, and `swr_create_screen()` loads the appropriate one at runtime.

Profiling

OpenSWR contains built-in profiling which can be enabled at build time to provide insight into performance tuning.

To enable this, uncomment the following line in `rasterizer/core/knobs.h` and rebuild:

```
//#define KNOB_ENABLE_RDTSC
```

Running an application will result in a `rdtsc.txt` file being created in current working directory. This file contains profile information captured between the `KNOB_BUCKETS_START_FRAME` and `KNOB_BUCKETS_END_FRAME` (see knobs section).

The resulting file will contain sections for each thread with a hierarchical breakdown of the time spent in the various operations. For example:

```
Thread 0 (API)
 %Tot   %Par  Cycles     CPE        NumEvent   CPE2       NumEvent2  Bucket
  0.00   0.00 28370      2837       10         0          0          ␣
↪APIClearRenderTarget
  0.00  41.23 11698      1169       10         0          0          |->␣
↪APIDrawWakeAllThreads
  0.00  18.34 5202       520        10         0          0          |->␣
↪APIGetDrawContext
 98.72  98.72 12413773688 29957      414380     0          0          APIDraw
  0.36   0.36 44689364   107        414380     0          0          |->␣
↪APIDrawWakeAllThreads
 96.36  97.62 12117951562 9747       1243140    0          0          |->␣
↪APIGetDrawContext
  0.00   0.00 19904      995        20         0          0          APIStoreTiles
```

(continues on next page)

(continued from previous page)

```
  0.00   7.88 1568       78         20         0          0          |->␣
↪APIDrawWakeAllThreads
  0.00  25.28 5032       251        20         0          0          |->␣
↪APIGetDrawContext
  1.28   1.28 161344902  64         2486370    0          0          APIGetDrawContext
  0.00   0.00 50368      2518       20         0          0          APISync
  0.00   2.70 1360       68         20         0          0          |->␣
↪APIDrawWakeAllThreads
  0.00  65.27 32876      1643       20         0          0          |->␣
↪APIGetDrawContext


Thread 1 (WORKER)
 %Tot   %Par  Cycles     CPE        NumEvent   CPE2       NumEvent2  Bucket
 83.92  83.92 13198987522 96411      136902     0          0           FEProcessDraw
 24.91  29.69 3918184840 167        23410158   0          0          |-> FEFetchShader
 11.17  13.31 1756972646 75         23410158   0          0          |->␣
↪FEVertexShader
  8.89  10.59 1397902996 59         23410161   0          0          |-> FEPAAssemble
 19.06  22.71 2997794710 384        7803387    0          0          |->␣
↪FEClipTriangles
 11.67  61.21 1834958176 235        7803387    0          0              |->␣
↪FEBinTriangles
  0.00   0.00 0          0          187258     0          0                  |->␣
↪FECullZeroAreaAndBackface
  0.00   0.00 0          0          60051033   0          0                  |->␣
↪FECullBetweenCenters
  0.11   0.11 17217556   2869592    6          0          0         ␣
↪FEProcessStoreTiles
 15.97  15.97 2511392576 73665      34092      0          0         ␣
↪WorkerWorkOnFifoBE
 14.04  87.95 2208687340 9187       240408     0          0          |->␣
↪WorkerFoundWork
  0.06   0.43 9390536    13263      708        0          0              |->␣
↪BELoadTiles
  0.00   0.01 293020     182        1609       0          0              |-> BEClear
 12.63  89.94 1986508990 949        2093014    0          0              |->␣
↪BERasterizeTriangle
  2.37  18.75 372374596  177        2093014    0          0                  |->␣
↪BETriangleSetup
  0.42   3.35 66539016   31         2093014    0          0                  |->␣
↪BEStepSetup
  0.00   0.00 0          0          21766      0          0                  |->␣
↪BETrivialReject
  1.05   8.33 165410662  79         2071248    0          0                  |->␣
↪BERasterizePartial
  6.06  48.02 953847796  1260       756783     0          0                  |->␣
↪BEPixelBackend
  0.20   3.30 31521202   41         756783     0          0                      |->␣
↪BESetup
  0.16   2.69 25624304   33         756783     0          0                      |->␣
↪BEBarycentric
  0.18   2.92 27884986   36         756783     0          0                      |->␣
↪BEEarlyDepthTest
  0.19   3.20 30564174   41         744058     0          0                      |->␣
↪BEPixelShader
  0.26   4.30 41058646   55         744058     0          0                      |->␣
↪BEOutputMerger
```

(continues on next page)

(continued from previous page)

```
  1.27  20.94 199750822  32         6054264    0          0                      |->␣
↪BEEndTile
  0.33   2.34 51758160   23687      2185       0          0              |->␣
↪BEStoreTiles
  0.20  60.22 31169500   28807      1082       0          0                  |->␣
↪B8G8R8A8_UNORM
  0.00   0.00 302752     302752     1          0          0          ␣
↪WorkerWaitForThreadEvent
```

Knobs

OpenSWR has a number of environment variables which control its operation, in addition to the normal Mesa and gallium controls.

KNOB_ENABLE_ASSERT_DIALOGS Type: bool, **Default: true**

Use dialogs when asserts fire. Asserts are only enabled in debug builds

KNOB_SINGLE_THREADED Type: bool, **Default: false**

If enabled will perform all rendering on the API thread. This is useful mainly for debugging purposes.

KNOB_DUMP_SHADER_IR Type: bool, **Default: false**

Dumps shader LLVM IR at various stages of jit compilation.

KNOB_USE_GENERIC_STORETILE Type: bool, **Default: false**

Always use generic function for performing StoreTile. Will be slightly slower than using optimized (jitted) path

KNOB_FAST_CLEAR Type: bool, **Default: true**

Replace 3D primitive execute with a SWRClearRT operation and defer clear execution to first backend op on hottile, or hottile store

KNOB_MAX_NUMA_NODES Type: uint32_t, **Default: 0**

Maximum # of NUMA-nodes per system used for worker threads 0 == ALL NUMA-nodes in the system N == Use at most N NUMA-nodes for rendering

KNOB_MAX_CORES_PER_NUMA_NODE Type: uint32_t, **Default: 0**

Maximum # of cores per NUMA-node used for worker threads. 0 == ALL non-API thread cores per NUMA-node N == Use at most N cores per NUMA-node

KNOB_MAX_THREADS_PER_CORE Type: uint32_t, **Default: 1**

Maximum # of (hyper)threads per physical core used for worker threads. 0 == ALL hyper-threads per core N == Use at most N hyper-threads per physical core

KNOB_MAX_WORKER_THREADS Type: uint32_t, **Default: 0**

Maximum worker threads to spawn. IMPORTANT: If this is non-zero, no worker threads will be bound to specific HW threads. They will all be "floating" SW threads. In this case, the above 3 KNOBS will be ignored.

KNOB_BUCKETS_START_FRAME Type: uint32_t, **Default: 1200**

Frame from when to start saving buckets data. NOTE: KNOB_ENABLE_RDTSC must be enabled in core/knobs.h for this to have an effect.

KNOB_BUCKETS_END_FRAME Type: uint32_t, **Default: 1400**

Frame at which to stop saving buckets data. NOTE: KNOB_ENABLE_RDTSC must be enabled in core/knobs.h for this to have an effect.

KNOB_WORKER_SPIN_LOOP_COUNT Type: uint32_t, **Default: 5000**

Number of spin-loop iterations worker threads will perform before going to sleep when waiting for work

KNOB_MAX_DRAWS_IN_FLIGHT Type: uint32_t, **Default: 160**

Maximum number of draws outstanding before API thread blocks.

KNOB_MAX_PRIMS_PER_DRAW Type: uint32_t, **Default: 2040**

Maximum primitives in a single Draw(). Larger primitives are split into smaller Draw calls. Should be a multiple of (3 * vectorWidth).

KNOB_MAX_TESS_PRIMS_PER_DRAW Type: uint32_t, **Default: 16**

Maximum primitives in a single Draw() with tessellation enabled. Larger primitives are split into smaller Draw calls. Should be a multiple of (vectorWidth).

KNOB_MAX_FRAC_ODD_TESS_FACTOR Type: float, **Default: 63.0f**

(DEBUG) Maximum tessellation factor for fractional-odd partitioning.

KNOB_MAX_FRAC_EVEN_TESS_FACTOR Type: float, **Default: 64.0f**

(DEBUG) Maximum tessellation factor for fractional-even partitioning.

KNOB_MAX_INTEGER_TESS_FACTOR Type: uint32_t, **Default: 64**

(DEBUG) Maximum tessellation factor for integer partitioning.

KNOB_BUCKETS_ENABLE_THREADVIZ Type: bool, **Default: false**

Enable threadviz output.

KNOB_TOSS_DRAW Type: bool, **Default: false**

Disable per-draw/dispatch execution

KNOB_TOSS_QUEUE_FE Type: bool, **Default: false**

Stop per-draw execution at worker FE NOTE: Requires KNOB_ENABLE_TOSS_POINTS to be enabled in core/knobs.h

KNOB_TOSS_FETCH Type: bool, **Default: false**

Stop per-draw execution at vertex fetch NOTE: Requires KNOB_ENABLE_TOSS_POINTS to be enabled in core/knobs.h

KNOB_TOSS_IA Type: bool, **Default: false**

Stop per-draw execution at input assembler NOTE: Requires KNOB_ENABLE_TOSS_POINTS to be enabled in core/knobs.h

KNOB_TOSS_VS Type: bool, **Default: false**

Stop per-draw execution at vertex shader NOTE: Requires KNOB_ENABLE_TOSS_POINTS to be enabled in core/knobs.h

KNOB_TOSS_SETUP_TRIS Type: bool, **Default: false**

Stop per-draw execution at primitive setup NOTE: Requires KNOB_ENABLE_TOSS_POINTS to be enabled in core/knobs.h

KNOB_TOSS_BIN_TRIS Type: bool, **Default: false**

Stop per-draw execution at primitive binning NOTE: Requires KNOB_ENABLE_TOSS_POINTS to be enabled in core/knobs.h

KNOB_TOSS_RS Type: bool, **Default: false**

Stop per-draw execution at rasterizer NOTE: Requires KNOB_ENABLE_TOSS_POINTS to be enabled in core/knobs.h

34.10.4 Zink

Overview

The Zink driver is a Gallium driver that emits Vulkan API calls instead of targeting a specific GPU architecture. This can be used to get full desktop OpenGL support on devices that only support Vulkan.

Features

The feature-level of Zink depends on two things; what's implemented in Zink, as well as the features of the Vulkan driver.

OpenGL 2.1

OpenGL 2.1 is the minimum version Zink can support, and will always be exposed, given Vulkan support. There's a few features that are required for correct behavior, but not all of these are validated; instead you'll see rendering-issues and likely validation error, or even crashes.

Here's a list of those requirements:

- Vulkan 1.0
- `VkPhysicalDeviceFeatures`:
  - `logicOp`
  - `fillModeNonSolid`
  - `wideLines`
  - `largePoints`
  - `alphaToOne`
  - `shaderClipDistance`
- `VkPhysicalDeviceLimits`:
  - `maxClipDistances` 6
- Instance extensions:
  - VK_KHR_get_physical_device_properties2
  - VK_KHR_external_memory_capabilities
- Device extensions:
  - VK_KHR_maintenance1
  - VK_KHR_external_memory

OpenGL 3.0

For OpenGL 3.0 support, the following additional device extensions are required to be exposed and fully supported:

- VK_EXT_transform_feedback
- VK_EXT_conditional_rendering

Debugging

There's a few tools that are useful for debugging Zink, like this environment variable:

ZINK_DEBUG Type: flags, **Default: ""**

nir Print the NIR form of all shaders to stderr.

spirv Write the binary SPIR-V form of all compiled shaders to a file in the current directory, and print a message with the filename to stderr.

tgsi Print the TGSI form of TGSI shaders to stderr.

Vulkan Validation Layers

Another useful tool for debugging is the Vulkan Validation Layers.

The validation layers effectively insert extra checking between Zink and the Vulkan driver, pointing out incorrect usage of the Vulkan API. The layers can be enabled by setting the environment variable `VK_INSTANCE_LAYERS` to "VK_LAYER_KHRONOS_validation". You can read more about the Validation Layers in the link above.

IRC

In order to make things a bit easier to follow, we have decided to create our own IRC channel. If you're interested in contributing, or have any technical questions, don't hesitate to visit #zink on FreeNode and say hi!

34.11 Gallium Post-processing

The Gallium drivers support user-defined image post-processing. At the end of drawing a frame a post-processing filter can be applied to the rendered image. Example filters include morphological antialiasing and cell shading.

The filters can be toggled per-app via driconf, or per-session via the corresponding environment variables.

Multiple filters can be used together.

34.11.1 PP environment variables

- PP_DEBUG - If defined debug information will be printed to stderr.

34.11.2 Current filters

- pp_nored, pp_nogreen, pp_noblue - set to 1 to remove the corresponding color channel. These are basic filters for easy testing of the PP queue.

- pp_jimenezmlaa, pp_jimenezmlaa_color - Jimenez's MLAA is a morphological antialiasing filter. The two versions use depth and color data, respectively. Which works better depends on the app - depth will not blur text, but it will miss transparent textures for example. Set to a number from 2 to 32, roughly corresponding to quality. Numbers higher than 8 see minimizing gains.
- pp_celshade - set to 1 to enable cell shading (a more complex color filter).

34.12 Glossary

GLSL GL Shading Language. The official, common high-level shader language used in GL 2.0 and above.

layer This term is used as the name of the "3rd coordinate" of a resource. 3D textures have zslices, cube maps have faces, 1D and 2D array textures have array members (other resources do not have multiple layers). Since the functions only take one parameter no matter what type of resource is used, use the term "layer" instead of a resource type specific one.

LOD Level of Detail. Also spelled "LoD." The value that determines when the switches between mipmaps occur during texture sampling.

MSAA Multi-Sampled Anti-Aliasing. A basic anti-aliasing technique that takes multiple samples of the depth buffer, and uses this information to smooth the edges of polygons.

NPOT Non-power-of-two. Usually applied to textures which have at least one dimension which is not a power of two.

TCL Transform, Clipping, & Lighting. The three stages of preparation in a rasterizing pipeline prior to the actual rasterization of vertices into fragments.

34.13 Indices and tables

- genindex
- modindex
- search

CHAPTER 35

Android

Mesa hardware drivers can be built for Android one of two ways: built into the Android OS using the Android.mk build sytem on older versions of Android, or out-of-tree using the Meson build system and the Android NDK.

The Android.mk build system has proven to be hard to maintain, as one needs a built Android tree to build against, and it has never been tested in CI. The meson build system flow is frequently used by Chrome OS developers for building and testing Android drivers.

35.1 Building using the Android NDK

Download and install the NDK using whatever method you normally would. Then, create your meson cross file to use it, something like this `~/.local/share/meson/cross/android-aarch64` file:

```
[binaries]
ar = 'NDKDIR/toolchains/llvm/prebuilt/linux-x86_64/bin/aarch64-linux-android-ar'
c = ['ccache', 'NDKDIR/toolchains/llvm/prebuilt/linux-x86_64/bin/aarch64-linux-
↪android29-clang', '-fuse-ld=lld']
cpp = ['ccache', 'NDKDIR/toolchains/llvm/prebuilt/linux-x86_64/bin/aarch64-linux-
↪android29-clang++', '-fuse-ld=lld', '-fno-exceptions', '-fno-unwind-tables', '-fno-
↪asynchronous-unwind-tables', '-static-libstdc++']
strip = 'NDKDIR/toolchains/llvm/prebuilt/linux-x86_64/bin/aarch64-linux-android-strip'
# Android doesn't come with a pkg-config, but we need one for meson to be happy not
# finding all the optional deps it looks for.  Use system pkg-config pointing at a
# directory we get to populate with any .pc files we want to add for Android
pkgconfig = ['env', 'PKG_CONFIG_LIBDIR=NDKDIR/pkgconfig', '/usr/bin/pkg-config']

[host_machine]
system = 'linux'
cpu_family = 'arm'
cpu = 'armv8'
endian = 'little'
```

Now, use that cross file for your Android build directory (as in this one cross-compiling the turnip driver for a stock Pixel phone)

```
meson build-android-aarch64 \
    --cross-file android-aarch64 \
    -Dplatforms=android \
    -Dplatform-sdk-version=26 \
    -Dandroid-stub=true \
    -Dgallium-drivers= \
    -Dvulkan-drivers=freedreno \
    -Dfreedreno-kgsl=true
ninja -C build-android-aarch64
```

35.2 Replacing Android drivers on stock Android

The vendor partition with the drivers is normally mounted from a read-only disk image on `/vendor`. To be able to replace them for driver development, we need to unlock the device and remount `/vendor` read/write.

```
adb disable-verity
adb reboot
adb remount -R
```

Now you can replace drivers as in:

```
adb push build-android-aarch64/src/freedreno/vulkan/libvulkan_freedreno.so /vendor/
↪lib64/hw/vulkan.sdm710.so
```

Note this command doesn't quite work because libvulkan wants the SONAME to match. For now, in turnip we have been using a hack to the meson.build to change the SONAME.

35.3 Replacing Android drivers on Chrome OS

Chrome OS's ARC++ is an Android container with hardware drivers inside of it. The vendor partition with the drivers is normally mounted from a read-only squashfs image on disk. For doing rapid driver development, you don't want to regenerate that image. So, we'll take the existing squashfs image, copy it out on the host, and then use a bind mount instead of a loopback mount so we can update our drivers using scp from outside the container.

On your device, you'll want to make `/` read-write. ssh in as root and run:

```
crossystem dev_boot_signed_only=0
/usr/share/vboot/bin/make_dev_ssd.sh --remove_rootfs_verification --partitions 4
reboot
```

Then, we'll switch Android from using an image for `/vendor` to using a bind-mount from a directory we control.

```
cd /opt/google/containers/android/
mkdir vendor-ro
mount -o loop vendor.raw.img vendor-ro
cp -a vendor-ro vendor-rw
emacs config.json
```

In the `config.json`, you want to find the block for `/vendor` and change it to:

```
{
    "destination": "/vendor",
```

(continues on next page)

(continued from previous page)

```
    "type": "bind",
    "source": "/opt/google/containers/android/vendor-rw",
    "options": [
        "bind",
        "rw"
    ]
},
```

Now, restart the UI to do a full reload:

```
restart ui
```

At this point, your android container is restarted with your new bind-mount `/vendor`, and if you use `android-sh` to shell into it then the `mount` command should show:

```
/dev/root on /vendor type ext2 (rw,seclabel,relatime)
```

Now, replacing your DRI driver with a new one built for Android should be a matter of:

```
scp msm_dri.so $HOST:/opt/google/containers/android/vendor-rw/lib64/dri/
```

You can do your build of your DRI driver using `emerge-$BOARD arc-mesa-freedreno` (for example) if you have a source tree with ARC++, but it should also be possible to build using the NDK as described above. There are currently rough edges with this, for example the build will require that you have your arc-libdrm build available to the NDK, assuming you're building anything but the freedreno vulkan driver for KGSL. You can mostly put things in place with:

```
scp $HOST:/opt/google/containers/android/vendor-rw/lib64/libdrm.so \
    NDKDIR/sysroot/usr/lib/aarch64-linux-android/lib/

ln -s \
    /usr/include/xf86drm.h \
    /usr/include/libsync.h \
    /usr/include/libdrm \
    NDKDIR/sysroot/usr/include/
```

It seems that new invocations of an application will often reload the DRI driver, but depending on the component you're working on you may find you need to reload the whole Android container. To do so without having to log in to Chrome again every time, you can just kill the container and let it restart:

```
kill $(cat /run/containers/android-run_oci/container.pid )
```

CHAPTER 36

Conformance Testing

Mesa as a project does not get certified conformant by Khronos for the APIs it implements. Rather, individual driver teams run the conformance tests and submit their results on a set of hardware on a particular operating system. The canonical list is at Khronos's list of conformant products and you can find some reports there by searching for "Mesa", "Raspbian" and "RADV" for example.

36.1 Submitting conformance results to Khronos

If your driver team is associated with an organization that is a Khronos member and has submitted conformance for your API on another software stack (likely you're a hardware company), it will probably be easiest to submit your conformance through them.

If you are an individual developer or your organization hasn't submitted results for the given API yet, X.Org is a member through Software in the Public Interest, and they can help submit your conformance results to get added to the list of conformant products. You should probably coordinate with board@foundation.x.org for your first submission.

CHAPTER 37

Continuous Integration

37.1 GitLab CI

GitLab provides a convenient framework for running commands in response to Git pushes. We use it to test merge requests (MRs) before merging them (pre-merge testing), as well as post-merge testing, for everything that hits `master` (this is necessary because we still allow commits to be pushed outside of MRs, and even then the MR CI runs in the forked repository, which might have been modified and thus is unreliable).

The CI runs a number of tests, from trivial build-testing to complex GPU rendering:

- Build testing for a number of build systems, configurations and platforms
- Sanity checks (`meson test` & `scons check`)
- Some drivers (softpipe, llvmpipe, freedreno and panfrost) are also tested using VK-GL-CTS
- Replay of application traces

A typical run takes between 20 and 30 minutes, although it can go up very quickly if the GitLab runners are overwhelmed, which happens sometimes. When it does happen, not much can be done besides waiting it out, or cancel it.

Due to limited resources, we currently do not run the CI automatically on every push; instead, we only run it automatically once the MR has been assigned to `Marge`, our merge bot.

If you're interested in the details, the main configuration file is `.gitlab-ci.yml`, and it references a number of other files in `.gitlab-ci/`.

If the GitLab CI doesn't seem to be running on your fork (or MRs, as they run in the context of your fork), you should check the "Settings" of your fork. Under "CI / CD" → "General pipelines", make sure "Custom CI config path" is empty (or set to the default `.gitlab-ci.yml`), and that the "Public pipelines" box is checked.

If you're having issues with the GitLab CI, your best bet is to ask about it on `#freedesktop` on Freenode and tag Daniel Stone (`daniels` on IRC) or Eric Anholt (`anholt` on IRC).

The three GitLab CI systems currently integrated are:

37.1.1 Bare-metal CI

The bare-metal scripts run on a system with gitlab-runner and Docker, connected to potentially multiple bare-metal boards that run tests of Mesa. Currently only "fastboot" and "ChromeOS Servo" devices are supported.

In comparison with LAVA, this doesn't involve maintaining a separate web service with its own job scheduler and replicating jobs between the two. It also places more of the board support in Git, instead of web service configuration. On the other hand, the serial interactions and bootloader support are more primitive.

Requirements (fastboot)

This testing requires power control of the DUTs by the gitlab-runner machine, since this is what we use to reset the system and get back to a pristine state at the start of testing.

We require access to the console output from the gitlab-runner system, since that is how we get the final results back from the tests. You should probably have the console on a serial connection, so that you can see bootloader progress.

The boards need to be able to have a kernel/initramfs supplied by the gitlab-runner system, since the initramfs is what contains the Mesa testing payload.

The boards should have networking, so that we can extract the dEQP .xml results to artifacts on GitLab.

Requirements (servo)

For servo-connected boards, we can use the EC connection for power control to reboot the board. However, loading a kernel is not as easy as fastboot, so we assume your bootloader can do TFTP, and that your gitlab-runner mounts the runner's tftp directory specific to the board at /tftp in the container.

Since we're going the TFTP route, we also use NFS root. This avoids packing the rootfs and sending it to the board as a ramdisk, which means we can support larger rootfses (for piglit or tracie testing), at the cost of needing more storage on the runner.

Telling the board about where its TFTP and NFS should come from is done using dnsmasq on the runner host. For example, this snippet in the dnsmasq.conf.d in the google farm, with the gitlab-runner host we call "servo":

```
dhcp-host=1c:69:7a:0d:a3:d3,10.42.0.10,set:servo

# Fixed dhcp addresses for my sanity, and setting a tag for
# specializing other DHCP options
dhcp-host=a0:ce:c8:c8:d9:5d,10.42.0.11,set:cheza1
dhcp-host=a0:ce:c8:c8:d8:81,10.42.0.12,set:cheza2

# Specify the next server, watch out for the double ',,'.  The
# filename didn't seem to get picked up by the bootloader, so we use
# tftp-unique-root and mount directories like
# /srv/tftp/10.42.0.11/jwerner/cheza as /tftp in the job containers.
tftp-unique-root
dhcp-boot=tag:cheza1,cheza1/vmlinuz,,10.42.0.10
dhcp-boot=tag:cheza2,cheza2/vmlinuz,,10.42.0.10

dhcp-option=tag:cheza1,option:root-path,/srv/nfs/cheza1
dhcp-option=tag:cheza2,option:root-path,/srv/nfs/cheza2
```

Setup

Each board will be registered in fd.o GitLab. You'll want something like this to register a fastboot board:

```
sudo gitlab-runner register \
     --url https://gitlab.freedesktop.org \
     --registration-token $1 \
     --name MY_BOARD_NAME \
     --tag-list MY_BOARD_TAG \
     --executor docker \
     --docker-image "alpine:latest" \
     --docker-volumes "/dev:/dev" \
     --docker-network-mode "host" \
     --docker-privileged \
     --non-interactive
```

For a servo board, you'll need to also volume mount the board's NFS root dir at /nfs and TFTP kernel directory at /tftp.

The registration token has to come from a fd.o GitLab admin going to https://gitlab.freedesktop.org/admin/runners

The name scheme for Google's lab is google-freedreno-boardname-n, and our tag is something like google-freedreno-db410c. The tag is what identifies a board type so that board-specific jobs can be dispatched into that pool.

We need privileged mode and the /dev bind mount in order to get at the serial console and fastboot USB devices (–device arguments don't apply to devices that show up after container start, which is the case with fastboot, and the servo serial devices are actually links to /dev/pts). We use host network mode so that we can spin up a nginx server to collect XML results for fastboot.

Once you've added your boards, you're going to need to add a little more customization in `/etc/gitlab-runner/config.toml`. First, add `concurrent = <number of boards>` at the top ("we should have up to this many jobs running managed by this gitlab-runner"). Then for each board's runner, set `limit = 1` ("only 1 job served by this board at a time"). Finally, add the board-specific environment variables required by your bare-metal script, something like:

```
[[runners]]
  name = "google-freedreno-db410c-1"
  environment = ["BM_SERIAL=/dev/ttyDB410c8", "BM_POWERUP=google-power-up.sh 8", "BM_
↪FASTBOOT_SERIAL=15e9e390"]
```

If you want to collect the results for fastboot you need to add the following two board-specific environment variables `BM_WEBDAV_IP` and `BM_WEBDAV_PORT`. These represent the IP address of the Docker host and the board specific port number that gets used to start a nginx server.

Once you've updated your runners' configs, restart with `sudo service gitlab-runner restart`

37.1.2 LAVA CI

LAVA is a system for functional testing of boards including deploying custom bootloaders and kernels. This is particularly relevant to testing Mesa because we often need to change kernels for UAPI changes (and this lets us do full testing of a new kernel during development), and our workloads can easily take down boards when mistakes are made (kernel oopses, OOMs that take out critical system services).

Mesa-LAVA software architecture

The gitlab-runner will run on some host that has access to the LAVA lab, with tags like "lava-mesa-boardname" to control only taking in jobs for the hardware that the LAVA lab contains. The gitlab-runner spawns a Docker container with lava-cli in it, and connects to the LAVA lab using a predefined token to submit jobs under a specific device type.

The LAVA instance manages scheduling those jobs to the boards present. For a job, it will deploy the kernel, device tree, and the ramdisk containing the CTS.

Deploying a new Mesa-LAVA lab

You'll want to start with setting up your LAVA instance and getting some boards booting using test jobs. Start with the stock QEMU examples to make sure your instance works at all. Then, you'll need to define your actual boards.

The device type in lava-gitlab-ci.yml is the device type you create in your LAVA instance, which doesn't have to match the board's name in `/etc/lava-dispatcher/device-types`. You create your boards under that device type and the Mesa jobs will be scheduled to any of them. Instantiate your boards by creating them in the UI or at the command line attached to that device type, then populate their dictionary (using an "extends" line probably referencing the board's template in `/etc/lava-dispatcher/device-types`). Now, go find a relevant healthcheck job for your board as a test job definition, or cobble something together from a board that boots using the same boot_method and some public images, and figure out how to get your boards booting.

Once you can boot your board using a custom job definition, it's time to connect Mesa CI to it. Install gitlab-runner and register as a shared runner (you'll need a GitLab admin for help with this). The runner *must* have a tag (like "mesa-lava-db410c") to restrict the jobs it takes or it will grab random jobs from tasks across fd.o, and your runner isn't ready for that.

The runner will be running an ARM Docker image (we haven't done any x86 LAVA yet, so that isn't documented). If your host for the gitlab-runner is x86, then you'll need to install qemu-user-static and the binfmt support.

The Docker image will need access to the lava instance. If it's on a public network it should be fine. If you're running the LAVA instance on localhost, you'll need to set `network_mode="host"` in `/etc/gitlab-runner/config.toml` so it can access localhost. Create a gitlab-runner user in your LAVA instance, log in under that user on the web interface, and create an API token. Copy that into a `lavacli.yaml`:

```
default:
  token: <token contents>
  uri: <URL to the instance>
  username: gitlab-runner
```

Add a volume mount of that `lavacli.yaml` to `/etc/gitlab-runner/config.toml` so that the Docker container can access it. You probably have a `volumes = ["/cache"]` already, so now it would be:

```
volumes = ["/home/anholt/lava-config/lavacli.yaml:/root/.config/lavacli.yaml", "/cache
↪"]
```

Note that this token is visible to anybody that can submit MRs to Mesa! It is not an actual secret. We could just bake it into the GitLab CI yml, but this way the current method of connecting to the LAVA instance is separated from the Mesa branches (particularly relevant as we have many stable branches all using CI).

Now it's time to define your test runner in `.gitlab-ci/lava-gitlab-ci.yml`.

37.1.3 Docker CI

For llvmpipe and swrast CI, we run tests in a container containing VK-GL-CTS, on the shared GitLab runners provided by freedesktop

Software architecture

The Docker containers are rebuilt from the debian-install.sh script when DEBIAN_TAG is changed in .gitlab-ci.yml, and debian-test-install.sh when DEBIAN_ARM64_TAG is changed in .gitlab-ci.yml. The resulting images are around

500MB, and are expected to change approximately weekly (though an individual developer working on them may produce many more images while trying to come up with a working MR!).

gitlab-runner is a client that polls gitlab.freedesktop.org for available jobs, with no inbound networking requirements. Jobs can have tags, so we can have DUT-specific jobs that only run on runners with that tag marked in the GitLab UI.

Since dEQP takes a long time to run, we mark the job as "parallel" at some level, which spawns multiple jobs from one definition, and then deqp-runner.sh takes the corresponding fraction of the test list for that job.

To reduce dEQP runtime (or avoid tests with unreliable results), a deqp-runner.sh invocation can provide a list of tests to skip. If your driver is not yet conformant, you can pass a list of expected failures, and the job will only fail on tests that aren't listed (look at the job's log for which specific tests failed).

DUT requirements

In addition to the general *CI farm expectations*, using Docker requires:

- DUTs must have a stable kernel and GPU reset (if applicable).

If the system goes down during a test run, that job will eventually time out and fail (default 1 hour). However, if the kernel can't reliably reset the GPU on failure, bugs in one MR may leak into spurious failures in another MR. This would be an unacceptable impact on Mesa developers working on other drivers.

- DUTs must be able to run Docker

The Mesa gitlab-runner based test architecture is built around Docker, so that we can cache the Debian package installation and CTS build step across multiple test runs. Since the images are large and change approximately weekly, the DUTs also need to be running some script to prune stale Docker images periodically in order to not run out of disk space as we rev those containers (perhaps this script).

Note that Docker doesn't allow containers to be stored on NFS, and doesn't allow multiple Docker daemons to interact with the same network block device, so you will probably need some sort of physical storage on your DUTs.

- DUTs must be public

By including your device in .gitlab-ci.yml, you're effectively letting anyone on the internet run code on your device. Docker containers may provide some limited protection, but how much you trust that and what you do to mitigate hostile access is up to you.

- DUTs must expose the dri device nodes to the containers.

Obviously, to get access to the HW, we need to pass the render node through. This is done by adding `devices = ["/dev/dri"]` to the `runners.docker` section of /etc/gitlab-runner/config.toml.

37.2 Intel CI

The Intel CI is not yet integrated into the GitLab CI. For now, special access must be manually given (file a issue in the Intel CI configuration repo if you think you or Mesa would benefit from you having access to the Intel CI). Results can be seen on mesa-ci.01.org if you are *not* an Intel employee, but if you are you can access a better interface on mesa-ci-results.jf.intel.com.

The Intel CI runs a much larger array of tests, on a number of generations of Intel hardware and on multiple platforms (X11, Wayland, DRM & Android), with the purpose of detecting regressions. Tests include Crucible, VK-GL-CTS, dEQP, Piglit, Skia, VkRunner, WebGL, and a few other tools. A typical run takes between 30 minutes and an hour.

If you're having issues with the Intel CI, your best bet is to ask about it on `#dri-devel` on Freenode and tag Clayton Craft (`craftyguy` on IRC) or Nico Cortes (`ngcortes` on IRC).

37.3 CI farm expectations

To make sure that testing of one vendor's drivers doesn't block unrelated work by other vendors, we require that a given driver's test farm produces a spurious failure no more than once a week. If every driver had CI and failed once a week, we would be seeing someone's code getting blocked on a spurious failure daily, which is an unacceptable cost to the project.

Additionally, the test farm needs to be able to provide a short enough turnaround time that we can get our MRs through marge-bot without the pipeline backing up. As a result, we require that the test farm be able to handle a whole pipeline's worth of jobs in less than 15 minutes (to compare, the build stage is about 10 minutes).

If a test farm is short the HW to provide these guarantees, consider dropping tests to reduce runtime. `VK-GL-CTS/scripts/log/bottleneck_report.py` can help you find what tests were slow in a `results.qpa` file. Or, you can have a job with no `parallel` field set and:

```
variables:
  CI_NODE_INDEX: 1
  CI_NODE_TOTAL: 10
```

to just run 1/10th of the test list.

If a HW CI farm goes offline (network dies and all CI pipelines end up stalled) or its runners are consistently spuriously failing (disk full?), and the maintainer is not immediately available to fix the issue, please push through an MR disabling that farm's jobs by adding '.' to the front of the jobs names until the maintainer can bring things back up. If this happens, the farm maintainer should provide a report to mesa-dev@lists.freedesktop.org after the fact explaining what happened and what the mitigation plan is for that failure next time.

37.4 Personal runners

Mesa's CI is currently run primarily on packet.net's m1xlarge nodes (2.2Ghz Sandy Bridge), with each job getting 8 cores allocated. You can speed up your personal CI builds (and marge-bot merges) by using a faster personal machine as a runner. You can find the gitlab-runner package in Debian, or use GitLab's own builds.

To do so, follow GitLab's instructions to register your personal GitLab runner in your Mesa fork. Then, tell Mesa how many jobs it should serve (`concurrent=`) and how many cores those jobs should use (`FDO_CI_CONCURRENT=`) by editing these lines in `/etc/gitlab-runner/config.toml`, for example:

```
concurrent = 2

[[runners]]
  environment = ["FDO_CI_CONCURRENT=16"]
```

37.5 Docker caching

The CI system uses Docker images extensively to cache infrequently-updated build content like the CTS. The freedesktop.org CI templates help us manage the building of the images to reduce how frequently rebuilds happen, and trim down the images (stripping out manpages, cleaning the apt cache, and other such common pitfalls of building Docker images).

When running a container job, the templates will look for an existing build of that image in the container registry under `FDO_DISTRIBUTION_TAG`. If it's found it will be reused, and if not, the associated *.gitlab-ci/containers/<jobname>.sh*' will be run to build it. So, when developing any change to container build scripts, you need to update the associated `FDO_DISTRIBUTION_TAG` to a new unique string. We recommend using the

current date plus some string related to your branch (so that if you rebase on someone else's container update from the same day, you will get a Git conflict instead of silently reusing their container)

When developing a given change to your Docker image, you would have to bump the tag on each `git commit --amend` to your development branch, which can get tedious. Instead, you can navigate to the container registry for your repository and delete the tag to force a rebuild. When your code is eventually merged to master, a full image rebuild will occur again (forks inherit images from the main repo, but MRs don't propagate images from the fork into the main repo's registry).

37.6 Building locally using CI docker images

It can be frustrating to debug build failures on an environment you don't personally have. If you're experiencing this with the CI builds, you can use Docker to use their build environment locally. Go to your job log, and at the top you'll see a line like:

```
Pulling docker image registry.freedesktop.org/anholt/mesa/debian/android_build:2020-
↪09-11
```

We'll use a volume mount to make our current Mesa tree be what the Docker container uses, so they'll share everything (their build will go in _build, according to `meson-build.sh`). We're going to be using the image non-interactively so we use `run --rm $IMAGE command` instead of `run -it $IMAGE bash` (which you may also find useful for debug). Extract your build setup variables from .gitlab-ci.yml and run the CI meson build script:

```
IMAGE=registry.freedesktop.org/anholt/mesa/debian/android_build:2020-09-11
sudo docker pull $IMAGE
sudo docker run --rm -v `pwd`:/mesa -w /mesa $IMAGE env PKG_CONFIG_PATH=/usr/local/
↪lib/aarch64-linux-android/pkgconfig/:/android-ndk-r21d/toolchains/llvm/prebuilt/
↪linux-x86_64/sysroot/usr/lib/aarch64-linux-android/pkgconfig/ GALLIUM_
↪DRIVERS=freedreno UNWIND=disabled EXTRA_OPTION="-D android-stub=true -D␣
↪llvm=disabled" DRI_LOADERS="-D glx=disabled -D gbm=disabled -D egl=enabled -D␣
↪platforms=android" CROSS=aarch64-linux-android ./.gitlab-ci/meson-build.sh
```

All you have left over from the build is its output, and a _build directory. You can hack on mesa and iterate testing the build with:

```
sudo docker run --rm -v `pwd`:/mesa $IMAGE ninja -C /mesa/_build
```

CHAPTER 38

Introduction

The Mesa project began as an open-source implementation of the OpenGL specification - a system for rendering interactive 3D graphics.

Over the years the project has grown to implement more graphics APIs, including OpenGL ES (versions 1, 2, 3), OpenCL, OpenMAX, VDPAU, VA API, XvMC and Vulkan.

A variety of device drivers allows the Mesa libraries to be used in many different environments ranging from software emulation to complete hardware acceleration for modern GPUs.

Mesa ties into several other open-source projects: the Direct Rendering Infrastructure and X.org to provide OpenGL support on Linux, FreeBSD and other operating systems.

38.1 Project History

The Mesa project was originally started by Brian Paul. Here's a short history of the project.

August, 1993: I begin working on Mesa in my spare time. The project has no name at that point. I was simply interested in writing a simple 3D graphics library that used the then-new OpenGL API. I was partially inspired by the *VOGL* library which emulated a subset of IRIS GL. I had been programming with IRIS GL since 1991.

November 1994: I contact SGI to ask permission to distribute my OpenGL-like graphics library on the internet. SGI was generally receptive to the idea and after negotiations with SGI's legal department, I get permission to release it.

February 1995: Mesa 1.0 is released on the internet. I expected that a few people would be interested in it, but not thousands. I was soon receiving patches, new features and thank-you notes on a daily basis. That encouraged me to continue working on Mesa. The name Mesa just popped into my head one day. SGI had asked me not to use the terms *"Open"* or *"GL"* in the project name and I didn't want to make up a new acronym. Later, I heard of the Mesa programming language and the Mesa spreadsheet for NeXTStep.

In the early days, OpenGL wasn't available on too many systems. It even took a while for SGI to support it across their product line. Mesa filled a big hole during that time. For a lot of people, Mesa was their first introduction to OpenGL. I think SGI recognized that Mesa actually helped to promote the OpenGL API, so they didn't feel threatened by the project.

1995-1996: I continue working on Mesa both during my spare time and during my work hours at the Space Science and Engineering Center at the University of Wisconsin in Madison. My supervisor, Bill Hibbard, lets me do this because Mesa is now being using for the Vis5D project.

October 1996: Mesa 2.0 is released. It implements the OpenGL 1.1 specification.

March 1997: Mesa 2.2 is released. It supports the new 3dfx Voodoo graphics card via the Glide library. It's the first really popular hardware OpenGL implementation for Linux.

September 1998: Mesa 3.0 is released. It's the first publicly-available implementation of the OpenGL 1.2 API.

March 1999: I attend my first OpenGL ARB meeting. I contribute to the development of several official OpenGL extensions over the years.

September 1999: I'm hired by Precision Insight, Inc. Mesa is a key component of 3D hardware acceleration in the new DRI project for XFree86. Drivers for 3dfx, 3dLabs, Intel, Matrox and ATI hardware soon follow.

October 2001: Mesa 4.0 is released. It implements the OpenGL 1.3 specification.

November 2001: I cofounded Tungsten Graphics, Inc. with Keith Whitwell, Jens Owen, David Dawes and Frank LaMonica. Tungsten Graphics was acquired by VMware in December 2008.

November 2002: Mesa 5.0 is released. It implements the OpenGL 1.4 specification.

January 2003: Mesa 6.0 is released. It implements the OpenGL 1.5 specification as well as the GL_ARB_vertex_program and GL_ARB_fragment_program extensions.

June 2007: Mesa 7.0 is released, implementing the OpenGL 2.1 specification and OpenGL Shading Language.

2008: Keith Whitwell and other Tungsten Graphics employees develop Gallium - a new GPU abstraction layer. The latest Mesa drivers are based on Gallium and other APIs such as OpenVG are implemented on top of Gallium.

February 2012: Mesa 8.0 is released, implementing the OpenGL 3.0 specification and version 1.30 of the OpenGL Shading Language.

July 2016: Mesa 12.0 is released, including OpenGL 4.3 support and initial support for Vulkan for Intel GPUs. Plus, there's another Gallium software driver ("swr") based on LLVM and developed by Intel.

Ongoing: Mesa is the OpenGL implementation for devices designed by Intel, AMD, NVIDIA, Qualcomm, Broadcom, Vivante, plus the VMware and VirGL virtual GPUs. There's also several software-based renderers: swrast (the legacy Mesa rasterizer), softpipe (a Gallium reference driver), llvmpipe (LLVM/JIT-based high-speed rasterizer) and swr (another LLVM-based driver).

Work continues on the drivers and core Mesa to implement newer versions of the OpenGL, OpenGL ES and Vulkan specifications.

38.2 Major Versions

This is a summary of the major versions of Mesa. Mesa's major version number has been incremented whenever a new version of the OpenGL specification is implemented.

38.2.1 Version 12.x features

Version 12.x of Mesa implements the OpenGL 4.3 API, but not all drivers support OpenGL 4.3.

Initial support for Vulkan is also included.

38.2.2 Version 11.x features

Version 11.x of Mesa implements the OpenGL 4.1 API, but not all drivers support OpenGL 4.1.

38.2.3 Version 10.x features

Version 10.x of Mesa implements the OpenGL 3.3 API, but not all drivers support OpenGL 3.3.

38.2.4 Version 9.x features

Version 9.x of Mesa implements the OpenGL 3.1 API. While the driver for Intel Sandy Bridge and Ivy Bridge is the only driver to support OpenGL 3.1, many developers across the open-source community contributed features required for OpenGL 3.1. The primary features added since the Mesa 8.0 release are GL_ARB_texture_buffer_object and GL_ARB_uniform_buffer_object.

Version 9.0 of Mesa also included the first release of the Clover state tracker for OpenCL.

38.2.5 Version 8.x features

Version 8.x of Mesa implements the OpenGL 3.0 API. The developers at Intel deserve a lot of credit for implementing most of the OpenGL 3.0 features in core Mesa, the GLSL compiler as well as the i965 driver.

38.2.6 Version 7.x features

Version 7.x of Mesa implements the OpenGL 2.1 API. The main feature of OpenGL 2.x is the OpenGL Shading Language.

38.2.7 Version 6.x features

Version 6.x of Mesa implements the OpenGL 1.5 API with the following extensions incorporated as standard features:

- GL_ARB_occlusion_query
- GL_ARB_vertex_buffer_object
- GL_EXT_shadow_funcs

Also note that several OpenGL tokens were renamed in OpenGL 1.5 for the sake of consistency. The old tokens are still available.

```
New Token                   Old Token
------------------------------------------------------------
GL_FOG_COORD_SRC            GL_FOG_COORDINATE_SOURCE
GL_FOG_COORD                GL_FOG_COORDINATE
GL_CURRENT_FOG_COORD        GL_CURRENT_FOG_COORDINATE
GL_FOG_COORD_ARRAY_TYPE     GL_FOG_COORDINATE_ARRAY_TYPE
GL_FOG_COORD_ARRAY_STRIDE   GL_FOG_COORDINATE_ARRAY_STRIDE
GL_FOG_COORD_ARRAY_POINTER  GL_FOG_COORDINATE_ARRAY_POINTER
GL_FOG_COORD_ARRAY          GL_FOG_COORDINATE_ARRAY
GL_SRC0_RGB                 GL_SOURCE0_RGB
GL_SRC1_RGB                 GL_SOURCE1_RGB
GL_SRC2_RGB                 GL_SOURCE2_RGB
```

(continues on next page)

(continued from previous page)

```
GL_SRC0_ALPHA                GL_SOURCE0_ALPHA
GL_SRC1_ALPHA                GL_SOURCE1_ALPHA
GL_SRC2_ALPHA                GL_SOURCE2_ALPHA
```

See the OpenGL specification for more details.

38.2.8 Version 5.x features

Version 5.x of Mesa implements the OpenGL 1.4 API with the following extensions incorporated as standard features:

- GL_ARB_depth_texture
- GL_ARB_shadow
- GL_ARB_texture_env_crossbar
- GL_ARB_texture_mirror_repeat
- GL_ARB_window_pos
- GL_EXT_blend_color
- GL_EXT_blend_func_separate
- GL_EXT_blend_logic_op
- GL_EXT_blend_minmax
- GL_EXT_blend_subtract
- GL_EXT_fog_coord
- GL_EXT_multi_draw_arrays
- GL_EXT_point_parameters
- GL_EXT_secondary_color
- GL_EXT_stencil_wrap
- GL_EXT_texture_lod_bias (plus, a per-texture LOD bias parameter)
- GL_SGIS_generate_mipmap

38.2.9 Version 4.x features

Version 4.x of Mesa implements the OpenGL 1.3 API with the following extensions incorporated as standard features:

- GL_ARB_multisample
- GL_ARB_multitexture
- GL_ARB_texture_border_clamp
- GL_ARB_texture_compression
- GL_ARB_texture_cube_map
- GL_ARB_texture_env_add
- GL_ARB_texture_env_combine
- GL_ARB_texture_env_dot3

- GL_ARB_transpose_matrix

38.2.10 Version 3.x features

Version 3.x of Mesa implements the OpenGL 1.2 API with the following features:

- BGR, BGRA and packed pixel formats
- New texture border clamp mode
- glDrawRangeElements()
- standard 3-D texturing
- advanced MIPMAP control
- separate specular color interpolation

38.2.11 Version 2.x features

Version 2.x of Mesa implements the OpenGL 1.1 API with the following features.

- Texture mapping:
  - glAreTexturesResident
  - glBindTexture
  - glCopyTexImage1D
  - glCopyTexImage2D
  - glCopyTexSubImage1D
  - glCopyTexSubImage2D
  - glDeleteTextures
  - glGenTextures
  - glIsTexture
  - glPrioritizeTextures
  - glTexSubImage1D
  - glTexSubImage2D
- Vertex Arrays:
  - glArrayElement
  - glColorPointer
  - glDrawElements
  - glEdgeFlagPointer
  - glIndexPointer
  - glInterleavedArrays
  - glNormalPointer
  - glTexCoordPointer
  - glVertexPointer

- Client state management:
  - glDisableClientState
  - glEnableClientState
  - glPopClientAttrib
  - glPushClientAttrib
- Misc:
  - glGetPointer
  - glIndexub
  - glIndexubv
  - glPolygonOffset

Index

F

G

I

K

L

M

N

O

P

R

S

T

U

V

X

Z