symbiflow-arch-defs Documentation

Release 0.1

Various

Oct 25, 2018

Contents

Warning: This project is a work in progress and many items may be broken.

This project contains documentation of various FPGA architectures, it is currently concentrating on;

- Lattice iCE40
- Artix 7

The aim is to include useful documentation (both human and machine readable) on the primitives and routing infrastructure for these architectures. We hope this enables growth in the open source FPGA tools space.

The project includes;

- Black box part definitions
- Verilog simulations
- Verilog To Routing architecture definitions
- Documentation for humans

CHAPTER 1

Contents

1.1 Getting Started

Checkout the project from:

> https://github.com/SymbiFlow/symbiflow-arch-defs

Make sure git submodules are cloned:

```
git submodule init
git submodule update
```

Run the full suite:

```
make
```

1.2 Development Practices

These documents outline the development practices for the project.

1.2.1 Structure

Directories

- *XXX/device/* - Full architecture definitions of a given device for [Verilog To Routing](https://verilogtorouting.org/)
  - *XXX/device/YYYY-virt* - Verilog to Routing architecture definitions generally are not able to able to generate the **exact** model of many FPGA routing interconnects, but this is a pretty close.
- *XXX/primitives/* - The primitives that make up the architecture. These are generally used inside the tiles.

- *XXX/tiles/* - The tiles found in the architecture.
- *XXX/tests/* - Tests for making sure the architecture specific features works with VPR.
- [*vpr*](vpr) - Common defines used by multiple architectures.

Files

- ***pb_type.xml* - The Verilog to Routing** [Complex Block](https://docs.verilogtorouting.org/en/latest/arch/reference/#complex-blocks) defintinition.
  - Inside *primitives* directory they should be intermediate or primitive <*pb_type*> and thus allow setting the *num_pb* attribute.
  - **Inside *tiles* directory they should be top level <*pb_type*> and thus have,**
    - *capacity* (if a pin type),
    - *width* & *height* (and maybe *area*)
- ***model.xml* - The Verilog to Routing** [Recognized BLIF Models](https://docs.verilogtorouting.org/en/latest/arch/reference/#recognized-blif-models-models) defintinition.
- ***sim.v* - A Verilog definition of the object. It should;**
  - **[ ] Match the definition in *model.xml* (should be one *module* in** *sim.v* for every *model* in *model.xml*)
  - **[ ] Include a *ifndef BLACKBOX* section which actually defines how the** Verilog works.
- *macro.v* - A Verilog definition of the object which a user might instantiate in their own code when specifying a primitive. This should match the definition provided by a manufacturer. Examples would be the definitions in;
  - [Lattice iCE Technology Library](http://www.latticesemi.com/~/media/LatticeSemi/Documents/TechnicalBriefs/SBTICETechnologyLibrary201504.pdf)
  - [UG953: Vivado Design Suite 7 Series FPGA and Zynq-7000 All Programmable SoC Libraries Guide](https://www.xilinx.com/support/documentation/sw_manuals/xilinx2017_3/ug953-vivado-7series-libraries.pdf)

Names

- *BLK_MB-block_1_name-block_2_name* - *BLOCK* which is a "**m**ega **b**lock". A "mega block" is a top level block which is made up of other blocks.
- *BLK_XX-name* - *BLOCK* which is the hierarchy. Maps to *BLK_SI* -> *SITE* and *BLK_TI* -> *TILE* in Xilinx terminology.
- *BLK_IG-name* - *BLOCK* which is ignored. They don't appear in the output hierarchy and are normally used when something is needed in the description which doesn't match actual architecture.
- *BEL_RX-mux_name* - *BEL* which is a **r**outing mu**x**. Routing muxes are statically configured at PnR time.
- *BEL_MX-mux_name* - *BEL* which is a **m**u**x** .
- *BEL_LT-lut_name* - *BEL* which is a **l**ook up **t**able.
- *BEL_MM-mem_name* - *BEL* which is a **m**e**m**ory.
- *BEL_FF-ff_name* - *BEL* which is a **f**lip **f**lop (*FF*).
- *BEL_LL-latch_name* - *BEL* which is a **l**atch (*LL*).
- *BEL_BB-name* - *BEL* which is a **b**lack **b**ox (*BB*).

- *PAD_IN-name* - A signal input location.
- *PAD_OT-name* - A signal output location.

Notes

- Unless there is a good reason otherwise, all muxes should be generated via [*mux_gen.py*](utils/mux_gen.py)
- DRY (Don't repeat yourself) - Uses [XML XIncludes](https://en.wikipedia.org/wiki/XInclude) to reuse stuff!

1.2.2 Verilog To Routing Notes

We have to do some kind of weird things to make VPR work for real architectures, here are some tips;

- VPR doesn't have channels right or above tiles on the right most / left most edge. To get these channels, pad the left most / right most edges with EMPTY tiles.
- Generally we use the [*vpr/pad*](vpr/pad) object for the **actual** *.input* and *.output* BLIF definitions. These are then connected to the tile which has internal IO logic.

CHAPTER 2

Indices and tables

- genindex
- modindex
- search