
minnesota Documentation

Release 0.1pre

Christopher L. Felton

October 28, 2015

1	Introduction	3
1.1	System	3
1.2	Models	3
1.3	Cores	3
2	Cores	5
2.1	FIFO	5
2.2	fpgalink	5
2.3	usbp	5
2.4	spi	5
2.5	vga	6
3	Xess Xula(2)	7
4	Digilent Nexys	9
5	Digilent Atlys	11
6	Digilent Zybo	13
7	Open-Source UFO-400	15
8	DSPtronics Signa-X1 (sx1)	17
9	Indices and tables	19

Contents:

Introduction

The `mn` package is a collection of HDL cores written in `myhdl`. The `mn` package is more than just a collection of cores it is also a framework for creating complex digital circuits. The `mn` package also includes a complete test suite.

The `mn` package is divided into three main subpackages:

- `system`
- `models`
- `cores`

1.1 System

The `system` subpackage contains the `[interfaces]()` and other useful tools to assist in building complex digital designs.

1.1.1 regfile

The register file objects provide simple methods to define registers and collections of registers. The registers files can be easily be connected to memory-mapped bus (e.g `wishbone`, `avalon`, etc.).

Defining a Register File

The following is a example defining a couple registers:

1.2 Models

This subpackage contains various models used for development and verification.

1.3 Cores

This subpackage contains the core implementations.

The following is a list of currently implemented cores.

2.1 FIFO

Various synchronous and asynchronous FIFO implementations.

2.2 fpgalink

This is a MyHDL implementation of the HDL for the *fpgalink* project. The *fpgalink* HDL core can be instantiated into a design:

For simulation and verification the *fpgalink* interface can be stimulated using the FX2 model and high-level access functions:

The following is a pictorial of the verification environment .

For more information on the [fpgalink]() software, firmware, and general design information see [makestuff]() .

2.3 usbp

USB Peripheral, this is another Cypress FX2 controller interface, this has two interfaces a “control” interface and a “streaming” interface. This FX2 interface is intended to work with the [fx2 firmware]() that configures the controller as a USB CDC/ACM device (virtual serial port). The [fx2 firmware]() also has a couple vendor unique commands that can be sent using the *pyusb* (or other low-level USB interfaces like *libusb*). The Python version of the host software (including firmware) can be retrieved via *pip*

```
>> pip install usbp
>>> import usbp
>>> import serial
```

One of the tricky items with USB devices is setting the permissions correctly. On a linux system to set the ...

2.4 spi

This is a generic SPI controller.

2.5 vga

VGA controller.

The following are the examples available in the examples directory.

Xess Xula(2)

- binary hello (blinky)
- VGA (TBC)

Digilent Nexys

- binary hello (blinky)
- fpgalink
- usbp

Digilent Atlys

- binary hello (blinky)
- fpgalink (TBC)
- usbp (TBC)

Digilent Zybo

- binary hello (blinky)

Open-Source UFO-400

- binary hello (blinky)
- usbp

DSPtronics Signa-X1 (sx1)

- binary hello (blinky)
- fpgalink
- usbp
- **audio examples**
 - audio echo (TBC)
 - audio streaming (TBC)

Indices and tables

- `genindex`
- `modindex`
- `search`