GitHub Reference for Prospective Publication Writing

Noa’s Given Reference URLs:

<https://github.com/ox-computing/Summer23-SmartNIC>

<https://github.com/NetFPGA/NetFPGA-plus>

<https://github.com/Xilinx/open-nic>

<https://github.com/Xilinx/open-nic-shell>

Some online Verilog compilers:

<https://www.edaplayground.com/x/9>

<https://www.jdoodle.com/execute-Verilog-online>

https://github.com/eugene-tarassov/vivado-risc-v

NetFPGA\_Plus:

Open NIC Shell Code Part:

图形用户界面, 图示

描述已自动生成

Shell and user logic boxes communicates through 3 types of interfaces.

* AXI-lite interface running at 125MHz for register access.
* AXI4-stream interface running at either 250MHz or 322MHz for data path.
* Synchronous reset interface running at 125MHz.

Currently, cocotb (<https://www.cocotb.org/>) and modelsim (<https://eda.sw.siemens.com/en-US/>) are used for simulation. cocotb allows writing testbenches in python.

* Quick tutorial on modelsim: <https://users.ece.cmu.edu/~jhoe/doku/doku.php?id=a_short_intro_to_modelsim_verilog_simulator>
* Reference testbenches using cocotb: <https://github.com/alexforencich/verilog-axis/blob/master/tb/axis_switch/test_axis_switch.py>

User Plugin Integration

图形用户界面, 文本

描述已自动生成

图形用户界面, 文本, 应用程序

描述已自动生成

箱线图

低可信度描述已自动生成

There are one or more holes in the OpenNIC shell, into which designers can place hardware functional IP blocks, dubbed “roles,” which tailor the SmartNIC design for specific networking needs.

The 250MHz box will contain the unique SmartNIC data path that sits between the PCIe (QDMA) and Ethernet (CMAC) interfaces. The other user logic box, called the “322MHz box”, is used to customize the CMAC subsystem for specialized networking needs.

The Universities of Cambridge and Oxford have developed the next-generation NetFPGA, called NetFPGA PLUS, which is a line-rate, flexible, and open SmartNIC platform for research and classroom experimentation.

文本, Word

描述已自动生成

**NetFPGA-Plus**

**Implementation Precautions:**

~/NetFPGA-PLUS\_final\_version $ vim tools/settings.sh

### User defined

export NFPLUS\_FOLDER=${HOME}/NetFPGA-PLUS

export BOARD\_NAME=au280

export NF\_PROJECT\_NAME=reference switch

export PYTHON\_BNRY=/usr/bin/python3

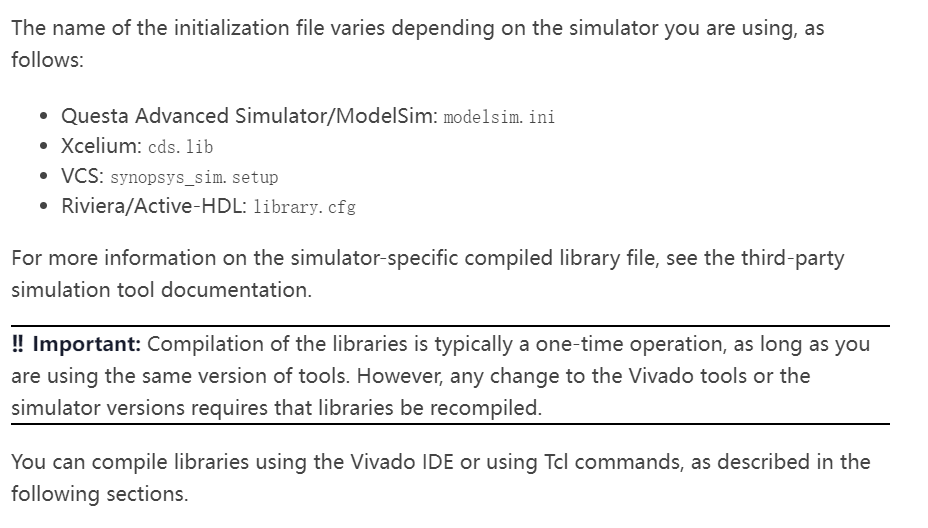
**用FPGA实现NIC（network interface controller，网络接口控制器），从该项目出发可以实现数据可控制化，进而实现DPU功能。**

Vivado Surrounding

Behavioral Simulation at the Register Transfer Level

Register Transfer Level (RTL) behavioural simulation can include:

* RTL Code
* Instantiated UNISIM library components
* Instantiated UNIMACRO components
* UNISIM gate-level model (for the Vivado logic analyser)
* SECUREIP Library



两种模式下都可以运行Tcl命令，但是需要注意的是有一些命令是指定在工程模式下或非工程模式下，这些命令就不能进行混用。如果在创建脚本的时候，在工程模式下使用了非工程模式的命令，那么数据库不会更新状态信息，并且也不会自动生成报告。但可以看出有一些报告命令既可以用于工程模式也可以用于非工程模式下，如上图标记的报告命令：report\_timing\_summary。

非工程模式下的命令只能在Vivado Tcl Shell中运行，但这并不意味着无法与图形界面方式交互使用。比如可以在Vivado Tcl Shell中执行start\_gui命令就可以打开工具回到图形界面方式，利用GUI操作对设计进行分析，分析完成后可通过stop\_gui命令回到Vivado Tcl Shell。

**表格

描述已自动生成RISC-V**

\*\*

Implement FPGA by **Pulpissimo & Pulp Project** 时会对enum type报错：

vim /home/yuzhejin/pulp/.bender/git/checkouts/pulp\_soc-bd81fb9bb29f56c3/rtl/pulp\_soc/soc\_interconnect.sv

Vim进原文档后原代码为：

LatencyMode: axi\_pkg::CUT\_MST\_AX | axi\_pkg::MuxW,

文本

描述已自动生成

**https://github.com/pulp-platform/pulpissimo/issues/354**

修改后的参考链接如上

**Vivado RISC-V Project**

Linux kernel and U-Boot use device tree, which is stored in RISC-V bootrom in FPGA. So, same SD card should boot OK on any board or RISC-V configuration.

Riscv-2022.2.tcl

set list\_projs [get\_projects -quiet]

if { $list\_projs eq "" } {

create\_project project\_1 myproj -part xcvu9p-fsgd2104-2L-e

set\_property BOARD\_PART xilinx.com:vcu1525:part0:1.3 [current\_project]

}

riscv-2022.2.tcl：

文本

描述已自动生成

473行改成：

default\_300mhz\_clk0

C:\Desktop\vivado-risc-v\_version4\workspace\rocket64b2\vivado-u280-riscv\u280-riscv.cache\ip\2022.2\2\2\222272cc29ff625c

图形用户界面, 文本, 应用程序

描述已自动生成

C:\Desktop\vivado-risc-v\_version4\workspace\rocket64b2\vivado-u280-riscv\u280-riscv.gen\sources\_1

图形用户界面, 应用程序, 电子邮件

描述已自动生成

图形用户界面, 文本, 应用程序

描述已自动生成

**Integration**

**Block Diagrams:**

Rocket Scala:

/\*----------------- 64-bit RocketChip ---------------\*/

/\*

\* WithExtMemSize(0x380000000L) = 14GB (16GB minus 2GB for IO) is max supported by the base config.

\* Actual memory size depends on the target board.

\* The Makefile changes the size to correct value during build.

\* It also sets right core clock frequency.

\*/

图形用户界面, 文本, 应用程序

描述已自动生成

**图形用户界面, 文本, 应用程序, 电子邮件

描述已自动生成**

**Conversation with 小叔：**

In NetFPGA top module,

.NetFPGA\_IO(Global Signal 想替换的信号)

Top\_tb

Wire Global\_qdma\_data

Netfpga u\_netfpga(.qdma\_data\_netfpga(global\_qdma\_data))

Rsic\_v u\_rsic\_v(.qdma\_data\_rsic\_v(global\_qdma\_data))

Second

Wire qdma\_riscv\_data 注释掉

Risi\_v

Qdma u\_qmda注释掉 //yuzhe0823，for reason

/home/yuzhejin/NetFPGA-PLUS/hw/lib/common/hdl/top.v

/home/yuzhejin/NetFPGA-PLUS/hw/projects/reference\_nic/hw/hdl/nf\_datapath.v

/home/yuzhejin/NetFPGA-PLUS/hw/lib/common/hdl/top\_wrapper.sv

图形用户界面, 文本, 应用程序, 电子邮件

描述已自动生成

图形用户界面, 文本, 应用程序

描述已自动生成

**图形用户界面, 文本, 应用程序, 电子邮件

描述已自动生成**

**图形用户界面, 文本, 应用程序, 电子邮件

描述已自动生成**