P51 - Lab 3
Infrastructure

Noa Zilberman, Andrew W Moore

Lent, 2019/20

The goal of this lab is to introduce you to the NetFPGA register infrastructure, as well as to the test environment, providing hands on experience in NetFPGA development.

1 Development Machines

This week you will continue and use the machines assigned to you last week. All the machines are located in the Practical Classroom (SW02).

1. On a computer in the Practical Classroom, log in using your own UIS credentials.
2. From Moodle, download the private key.
3. Limit the permissions of the private key: `chmod 600 p51_key`
4. `ssh -X root@<hostname>.nf.cl.cam.ac.uk -i p51_key`. Hosts ending in .cl.cam.ac.uk are permitted to ssh into these machines. `-X` enables X11 forwarding, allowing you to run graphical applications. `-i` is the private ssh authentication key.

To ssh to the machines from outside the lab, follow the instructions on `https://www.cl.cam.ac.uk/local/sys/ssh/`.

Hostname	IP Address
nf-test103	128.232.82.63
nf-test104	128.232.82.64
nf-test102	128.232.82.62
nf-test108	128.232.82.68
nf-test110	128.232.82.70
nf-test111	128.232.82.71

Important: The IP addresses noted above should not be used for anything except for communication with the machines. The network interfaces assigned for the tests use different IP addresses.

2 Practical Instructions

This section provides step-by-step instructions how to add a new register to your design. To this end, we will be using the (verilog) Reference Switch design studied in class.

2.1 Accessing the board

1. Login to the development machine:

   ```
   ssh root@<hostname>.nf.cl.cam.ac.uk
   ```

2. Pull the latest NetFPGA release:

   ```
   cd ~/NetFPGA-SUME-live/
   git pull
   ```

3. ```
   cd tools
   vim settings.sh
   ```

4. Make sure that NF_PROJECT_NAME is set to "reference_switch"

5. Load the environment settings:

   ```
   source settings.sh
   ```

6. Compile all cores:

   ```
   cd $SUME_FOLDER
   make
   ```

2.2 Adding a new register

The goal of this exercise is to add a register that counts the number of packets of size 64B. Once you complete this exercise, try adding a second register that sets the packet size to monitor.

The following instructions use the spreadsheet.

You can alternatively use the csv file, combined with the script *csv_gen.py*. The file is located under $SUME\_FOLDER/tools/infrastructure$. Then skip to step 7.

1. In Libreoffice, set security to medium.

2. Open $IP_FOLDER/switch_output_port_lookup_v1_0_1/data/module_generation.xls

3. If needed, change block name to match your module name (for sub-module this is optional)
   ```

4. Delete all indirect registers (and others you don't want) (note potential issues in some releases)
5. Change OS to Linux
6. Press "Generate Registers". You may need to copy locally *SUME_FOLDER/tools/infrastructure/regs_template.txt*.
7. From console, run:

```
python regs_gen.py
cp *.v $IP_FOLDER/switch_output_port_lookup_v1_0_1/hdl
```

8. Copy data files to the data folder:

```
cp <*.tcl,*.h,*.txt> $IP_FOLDER/switch_output_port_lookup_v1_0_1/data
```

9. Go to the core's folder:

```
cd $IP_FOLDER/switch\output_port_lookup_v1_0_1/
```

10. Open the HDL code of the core:

```
vim hdl/switch_output_port_lookup.v
```

11. Add the following lines to the code, starting line 370 (this is a hint, you can also use your own code):

```
reg next_is_new;
always @(posedge axis_aclk) next_is_new <= #1 ~resetn_sync ? 1'b0 :
     (s_axis_tvalid && s_axis_tready) ? s_axis_tlast : next_is_new;
```

12. Copy the lines from the template file to switch_output_port_lookup.v
13. Add in switch_output_port.v support for the register functionality
14. Compiling your IP core:

```
cd $IP_FOLDER/switch_output_port_lookup_v1_0_1/
make
```

15. Update the functional test to read (or write) your register:

```
cd $NF_DESIGN_DIR/test/both_learning_sw
vim run.py
```

16. Run a simulation of your design and test that it works:

```
cd $SUME_FOLDER/tools/scripts
./nf_test.py sim --major learning --minor sw
```

3 Useful links

- NetFPGA Repository: `https://github.com/NetFPGA/NetFPGA-SUME-live/`
- NetFPGA Wiki: `https://github.com/NetFPGA/NetFPGA-SUME-public/wiki`
- NetFPGA registration page: `https://netfpga.org/site/#/SUME_reg_form/`
- P4-NetFPGA Repository: `https://github.com/NetFPGA/P4-NetFPGA-live`
- P4-NetFPGA Wiki: `https://github.com/NetFPGA/P4-NetFPGA-public/wiki`
- P4-NetFPGA registration page: `https://goo.gl/forms/h7RbYmKZL7H4EaUf1`
- P4-NetFPGA online tutorial: `https://github.com/NetFPGA/P4-NetFPGA-public/wiki/Tutorial-Assignments`