EE272 Project
F20

Your team will design a simple switch box to allow connecting 4 NOC perm blocks to a single test bench. The test bench will be released in stages. The first stage works like the current test bench with a few minor changes. This test bench is called ‘baby’. The changes are mostly in addressing the perm device over the NOC bus. The second test bench called ‘child’ talks to two devices, one device at a time. The third test bench called teen talks to to devices mixing up commands to the two units. The fourth test bench talks to 4 devices with mixed transfers. You need to pass all 4 test benches

The switch box has 5 ports. 4 ports are for the perm/NOC interfaces, and the fifth port is connected to the test bench. The perm devices have ids 8’h40-h’h43. The switch should only send a packet to a device if addressed to the device. This implies you will have to delay a packed until you see the device address and know which device to send the data to. This can be done with a simple fifo, and or pipeline. The destination address is the second byte of each packet.

The devices shall be connected using system verilog interfaces. (See chapter 3 section 3.5 to better understand interfaces)

```
interface NOCI(reg clk, reg reset);
        logic noc_to_dev_ctl;
        logic [7:0] noc_to_dev_data;
        logic noc_from_dev_ctl;
        logic [7:0] noc_from_dev_data;

        modport FI(input clk, input reset, input noc_from_dev_ctl, input noc_from_dev_ctl) ;
        modport FO(input clk, input reset, output noc_from_dev_ctl, output noc_from_dev_ctl) ;
        modport TI(input clk, input reset,input noc_to_dev_ctl,input noc_to_dev_data);
        modport TO(input clk, input reset,output noc_to_dev_ctl,output noc_to_dev_data);
endinterface
```

Both directions of both buses are provided. The test bench uses the TO, and the switch uses TI.
The perm block NOC engine uses the TI, and the switch uses the TO with the devices.

Your design will be a single module (You may include the memories, they are not in the test bench). It will have a TO and FI interface. Your module must be called PS.

```
Module PS(NOCI.TO t, NOCI.FI f);
        ...
endmodule
```