LUCID VLSI

Learning Unlimited. Coding InDelible.

8

SystemVerilog LAB8

After completing this lab, you should be able to:

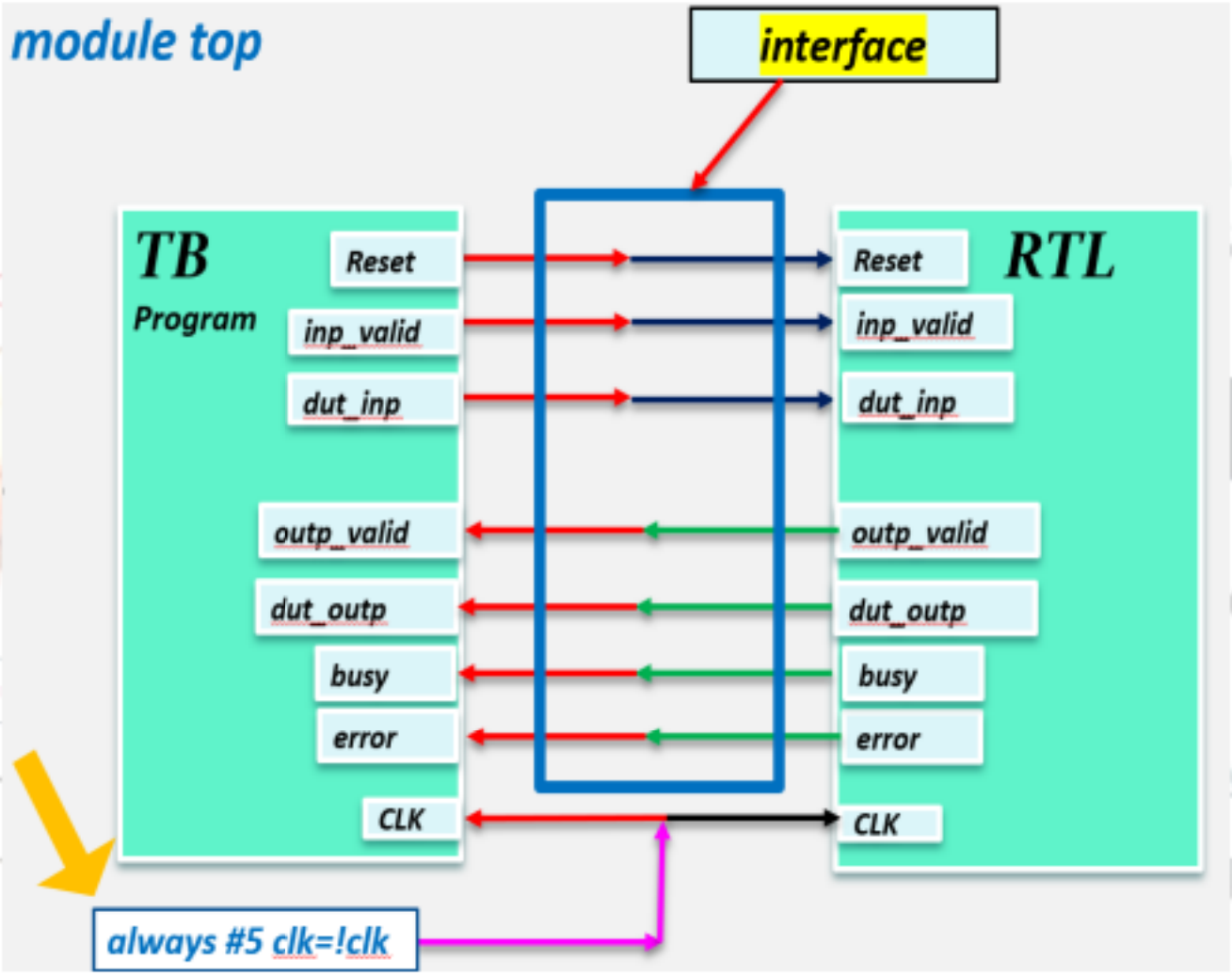
Implement modports in interface.

Understand how to use modports as port in TB.

Understand how to drive/sample signals using interface port in TB.

Understand how to connect interface instance to interface port of program block.

Verify DUT behavior with the help of self-checking mechanism in program block(testbench).


module top
interface
TB
Program
Reset
inp_valid
dut_inp
outp_valid
dut_outp
busy
error
CLK
RTL
Reset
inp_valid
dut_inp
outp_valid
dut_outp
busy
error
CLK
always #5 clk=!clk

LAB8: Copy lab7 directory as lab8

Step 1: Define modport in interface with required signals.

1) Open file top.sv and define modport in interface.
2) Define modport to specify signal directions for TB.
   Add this inside interface code in top.sv
   Ex: Directions are specified w.r.t TB.

```
modport tb_mod_port (output reset,dut_inp,inp_valid,
                     input outp_valid,dut_outp,busy,error);
```

Step 2: Connect modport to vif port.

1) Open top.sv and connect modport of interface to **vif** in TB port connection.
   We will have to drive/sample signals which are part of interface.

Ex:

Old code: **Interface instance connected in port list**

```
testbench  tb_inst(
                .clk(clk),
                .vif(router_if_inst)
                );
```

New Code: **modport connected in port list.**

```
testbench  tb_inst(
                .clk(clk),
                .vif(router_if_inst.tb_mod_port)
                );
```

Step 4: Run the simulation and validate the output of DUT with the results printed by self-checking mechanism.

1. Run the simulation and check the test Passed or Failed and debug the if there are any failures.