LUCID VLSI

Learning Unlimited. Coding InDelible.

7

SystemVerilog LAB7

After completing this lab, you should be able to:

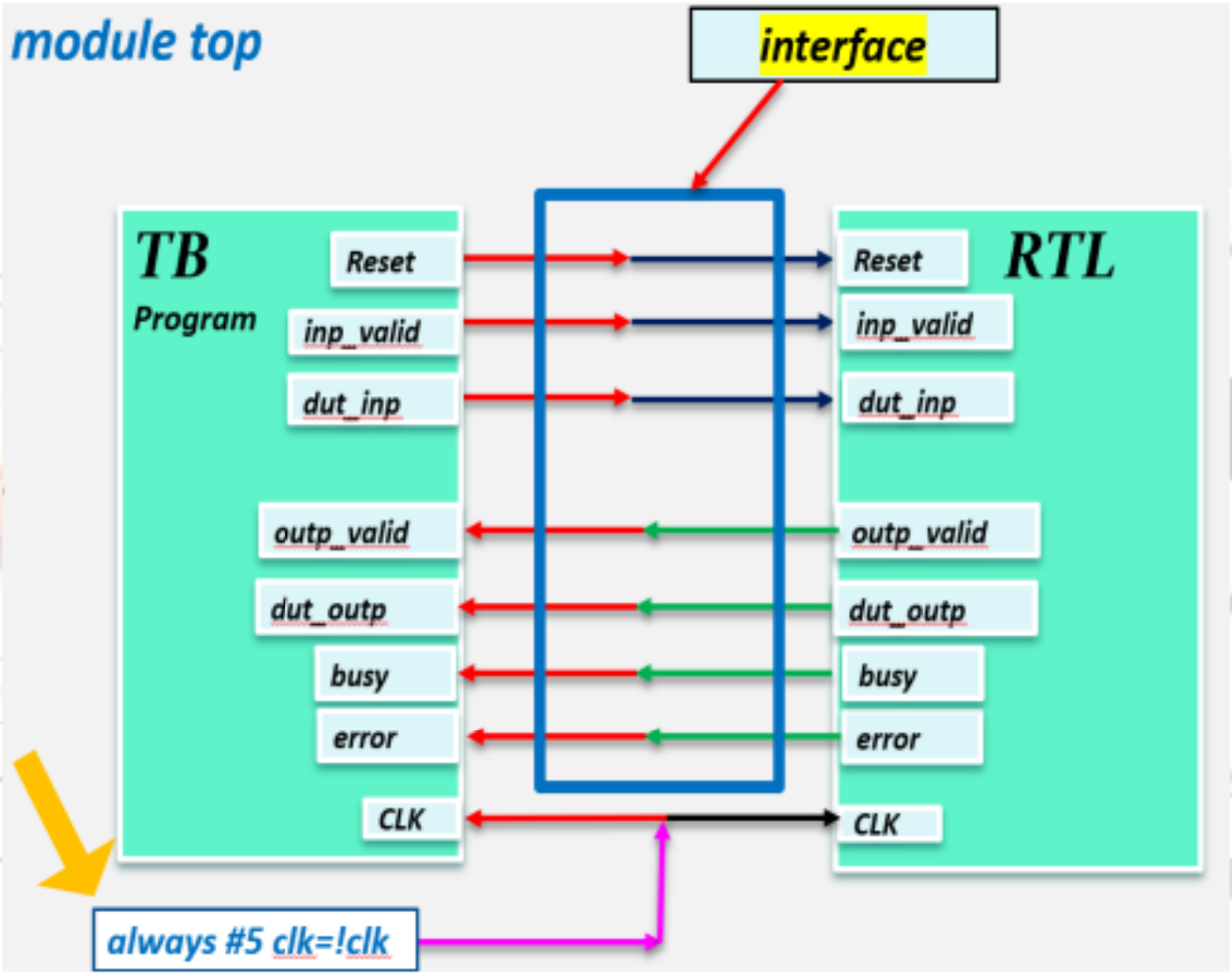
Implement interface as a port to program block(TB).

Understand how to use interface 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

LAB7: Copy lab6 directory as lab7

Step 1: Define interface as port in program block port list.

1) Open file testbench.sv and define interface as port.

   Ex:

   program testbench(input clk, router_if vif);

Step 2: Access all dut (ports) signals using vif port.

1) Open testbench.sv and replace all dut signals access with vif

   We will have to drive/sample signals which are part of interface.

Ex:

Old code: Signals are part of testbench I/O Port list

```
reset <=1;
inp_valid <=1;
dut_inp  <=1;
wait(busy==0);
```


New Code: Signals are part of interface Port vif.

```
vif.reset  <= 1;
vif.inp_valid <=1;
vif.dut_inp  <=1;
wait(vif.busy==0);
```

Step 3: Connect interface instance to program block(TB).

1) *Open top.sv and modify section 4 (Program Block (TB) instantiation).*
2) *Connect interface instance to port **vif** in port connection. Use name based port mapping.*
   Add this code in Section4 in top.sv

Ex:

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


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

1. *run the simulation.*
2. *Check the test Passed or Failed and debug the if there are any failures.*