

**United** International **University** (UIU)

Dept. of Electrical and Electronic Engineering (EEE)

# **Course**: VLSI Design Lab (EEE 442)

**Experiment 3: Synthesis of Counter using Verilog**

Introduction:

1. Switch to csh, this will also setup your cadence tools environment variables

* csh

1. Create the lab directory and few other sub-directories under your work area:

* mkdir lab3
* cd lab3
* mkdir simulation synthesis rtl lib constraints
* cd simulation

1. Use an editor (vi, nedit, gedit etc.) to write behavioral model of an 8-bit counter

* vi cntr8bit.v

1. Use an editor to write a test bench Verilog code for your counter

* vi cntr8bit\_tb.v

1. Use Incisive Enterprise Simulator (IES) to verify that the counter is working properly. Follow the steps from previous labs to complete the simulation.
2. When you are satisfied with your simulation, copy the counter module into the rtl sub-directory.

* cp cntr8bit.v ../rtl/

1. Go to lib/ and copy the slow.lib and fast.lib files from ~sahmed/eee442/lab3/lib/

* cd ../lib
* cp ~sahmed/eee442/lab3/lib/\*.lib . (don’t forget the dot(.) at the end)

1. Go to synthesis/ and create the RC synthesis script (rc\_script.tcl)

* cd ../synthesis
* vi rc\_script.tcl

1. Sample contents of rc\_script.tcl file:

set\_attr lib\_search\_path ../lib/

set\_attr hdl\_search\_path ../rtl/

set\_attr library slow.lib

read\_hdl cntr8bit.v

elaborate

read\_sdc ../constraints/cntr8bit.g

synthesize -to\_mapped -effort medium

write\_hdl > cntr8bit\_netlist.v

write\_sdc > cntr8bit.sdc

1. Go to constraints/ and create the design constraint file (cntr8bit.g)

* cd ../constraints
* vi cntr8bit.g

1. Sample contents of cntr8bit.g file:

create\_clock -name clk -period 10 -waveform {0 5} [get\_ports "clk"]

set\_clock\_transition -rise 0.1 [get\_clocks "clk"]

set\_clock\_transition -fall 0.1 [get\_clocks "clk"]

set\_clock\_uncertainty 0.1 [get\_ports "clk"]

set\_input\_delay -max 1.0 [get\_ports "rst"] -clock [get\_clocks "clk"]

set\_output\_delay -max 1.0 [get\_ports "count"] -clock [get\_clocks "clk"]

1. Move back to the synthesis/ and invoke RTL Compiler along with the rc\_script.tcl file to synthesize the netlist.

* cd ../synthesis
* rc –f rc\_script.tcl

1. While performing synthesis, always check the RC terminal whether the tool is reporting any error.
2. After synthesis, launch the GUI to see the schematic in the ‘rc’ terminal.

* gui\_show

1. After checking the schematic, if you are satisfied, close the GUI in the ‘rc’ terminal.

* gui\_hide

1. Use ‘report’ command to write out the results

* report timing (reports the timing details)
* report power (write the power report)
* report (to know different reports you can dump out from RC)

1. After completing synthesis, write out gate level netlist and SDC constraints for Physical Design.

* write\_hdl > cntr8bit\_netlist.v
* write\_sdc > cntr8bit.sdc

(Both the above commands are part of the rc\_script.tcl file, so they are already executed)

1. Close the RTL Compiler

* exit

1. Open the Cadence Help gui, expand the RTL Compiler and double click on any sub-menu you would like to explore.

* cdnshelp &

1. …