ELEN 603/COEN 303

Due : 12/11/2018 11:59PM
Final Project

1. Objective

In this assignment, you will be designing a 8x8 internet switch.

Note: This project is intended to be ***individually***. I don't have a problem if you wish to discuss the project with other students; however, your final work ***must*** be your own.

2. Functional Specification

Your circuit must implement a simple internet switch with 8 channels. Each channel has an input and output interface. A packet of data can be sent to any of the 8 output channels. When a packet is sent to a channel, the user must specify the destination channel along with the 32-bit payload. The switch will route the payload to the destination channel.

3. Interface Specification

Clock	Clock
Reset_n	Active low reset
Frame_n[7:0]	Frame in vector
Vald_n [7:0]	Valid in vector
Di[7:0]	Data in vector
Frameo_n[7:0]	Frame out vector
Valdo_n[7:0]	Valid out vector
Dout[7:0]	Data out vector

```
module router(
   input reset_n, clock,
   input [7:0] frame_n, vald_n, di,
  output [7:0] dout, valdo_n, frameo_n);
```

4. Protocol Specification

In order to send a packet of data to a specific port i, one would use frame_n[i], vald_n[i], di[i]

Payload = 32'b P_{31} … P_0
Destination address = 4'b$A_0A_1A_2A_3$

Cycle	Frame_n[i]	Vldn[i]	Di[i]
1	0	1	A_0
2	0	1	A_1
3	0	1	A_2
4	0	1	A_3
5	0	1	X
6	0	1	X
N	0	0	P_0
n+1	0	0	P_1
n+31	1	0	P_{31}
n+32	1	1	X

5. RTL Modeling

Write the synthesizable RTL model of your chip, simulate it against the functional model, and then synthesize it against the library "lsi_10k.db". Ensure your design is free of any coding style errors.

Your project will have points deducted if it contains any of the following:

- Latches
- Synthesis errors (ie: bad 3rd state drivers)
- Blocking assignments to registers inside clocked processes
- Race conditions
- Functional bugs
- Gate-level simulation mismatches

Ensure your design meets the following design goals:

Timing: clock period = 30 ns

Area : 50k gates maximum (report_area)

6. **Testbench**

   A testbench will be provided for you as a starting point. You are expected to enhance the testbench to create different input scenarios

7. **Grading**

All final projects must be demonstrated-live in order to receive credit. You are encouraged to send email/code if there is any doubt about anything.

RTL Simulations:

- smoke test : apply 1-packet to port0 -> port7
- 1-port test: serially apply packet to port0 -> port[0-7]
- 4-port test : apply 10 packets each (40 in total) in parallel to port[0,1,2,3] -> port[3,2,1,0]
  - [10 pts] RTL, [20 pts] gate-level
- Concurrent test : apply 10 packets (30 in total) to port0,port1,port7 -> port7
  - [10 pts] RTL, [20 pts] gate-level

To run any of the tests, do the following:
./simv +SMOKE
./simv +1PORT
./simv +4PORT
./simv +CONCURRENT

8. Synthesis script

Setup path for Design Compiler:
Source ~bgreene/setvcs

Invoke Design Compiler:
dc_shell-t

Commands to enter into Design Compiler (you can also put in a file, and source that file):

- set link_library {* lsi_10k.db dw_foundation.sldb}
- set target_library {lsi_10k.db}
- analyze –format sverilog { fifo.v router.v port.v }
- elaborate router
- create_clock clk -period 25
- set_max_area 25000
- set_input_delay -clock clk 0 [all_inputs]
- set_output_delay -clock clk 0 [all_outputs]

- compile
- report_area
- report_timing
- write -f verilog -hier -out gate.v

To simulation your RT-Level design :

```
vcs -sverilog -debug_access testbench.sv mod1.v mod2.v -y
$SYNOPSYS/dw/sim_ver +libext+.v
./simv +SMOKE
```

To simulation your gate-level design:

```
vcs -sverilog -debug_access testbench.sv my_gate_level_design.v -
y $SYNOPSYS/dw/sim_ver +libext+.v -v
/home/bgreene/VERILOG/lsi_10k.v
./simv +SMOKE
```

Note: lsi_10k.v is a simulation library and should only be used for simulating your gate-level design
You can find lsi_10k.v here : /home/bgreene/VERILOG/lsi_10k.v
You can find the testbench.v here : /home/bgreene/testbench.sv