CMPE 125 Lab 3 Assignments

Dr. Donald Hung
Computer Engineering Department, San Jose State University

Combinational Building Blocks

Purpose:

(1) To be familiar with the basic combinational building blocks
(2) To be familiar with the design/verification/validation flow and the EDA tools

Preparation:

1) Study the lecture slides
2) Study the relevant part of the *Verilog Coding Examples* posted on class Canvas (*CMPE125 Lecture Materials Section 3 – Basic Building Blocks and Verilog Fundamentals*).

Tasks:

1. The **priority encoder** (refer to *Verilog Coding Examples* on class Canvas):

   a) Follow the Verilog 2001 standard discussed in class, rewrite Verilog design code for the 8-3 priority encoder using ***if***, ***casez*** statements and ***for*** loop, respectively (3 versions).
   b) Write a **self-testing testbench** for the 8-3 priority encoder and use it to verify all three versions of your design. Use waveform viewer to confirm the correctness of your testbench report.
   c) Validate the 8-3 priority encoder (choose one version of your design) using the Nexys4 DDR FPGA board. Use the individual on-board LEDs to display outputs.

2. The **4-bit right shifter/rotator** with the functional table shown below:

Ctrl [2:0]	Operation	Input	Output
000	Pass	a b c d	a b c d
001	Shift right 1 bit	a b c d	0 a b c
010	Shift right 2 bits	a b c d	0 0 a b
011	Shift right 3 bits	a b c d	0 0 0 a
100	Shift right 4 bits	a b c d	0 0 0 0
101	Rotate right 1 bit	a b c d	d a b c
110	Rotate right 2 bits	a b c d	c d a b
111	Rotate right 3 bits	a b c d	b c d a

   Do the following:

   a) Follow the Verilog 2001 standard discussed in class, write Verilog design code for the *right shifter/rotator* specified above.
   b) Write a **self-testing Verilog testbench** and functionally verify the *right shifter/rotator* designed. Use waveform viewer to confirm the correctness of your testbench report.

c) Validate the designed *right shifter/rotator* using the Nexys4 DDR FPGA board. Use the individual on-board LEDs to display outputs.

3. The **4-bit ALU** (refer to the *Verilog Coding Examples* on class Canvas):
   a) Follow the Verilog 2001 standard discussed in class, rewrite Verilog design code to enhance the 4-bit ALU by adding two output signals, one as the "zero" flag, the other as the "overflow" flag.
   b) Write a **self-testing Verilog testbench** and functionally verify the enhanced ALU. Use waveform viewer to confirm the correctness of your testbench report.
   c) Validate the enhanced 4-bit ALU using the Nexys4 DDR FPGA board. Use the individual on-board LEDs to display outputs.

Contents of Report:

1) Cover page (use photo copy of the cover page signed by the lab TA or the course instructor).
2) A list of successfully accomplished tasks. For **each task** you should include the following:
   a) A function table of the design and a test plan which describes how you will functionally verify the design
   b) Commented source code (for both design and verification)
   c) Verification results reported by the self-checking testbench, as well as captured from the waveform viewer
   d) A diagram that shows the hardware validation (FPGA board prototyping) environment setup, as well as the top-level Verilog code and constraint (.xdc) file for this validation
3) Detailed descriptions of the task(s) that you were not able to accomplish, including reason(s) and/or your analysis.

Lab Checkup Schedule:

Week	Due to be Checked by TA
Week #1 (9/18 & 9/20)	1. Task 1 (Part a, b, c) 2. Task 2 (Part a)
Week #2 (9/25 & 9/27)	1. Task 2 (Part b, c) 2. Task 3 (Part a, b, c)