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: (*see checklist at end of document for more details)*

1) Cover page (use the provided template).
2) A list of successfully accomplished tasks. For each task you should include the following:
   a) A test plan (i.e., how to verify the design)
   b) Commented source code (for both design and verification)
   c) Captured verification results (waveforms and simulation reports)
   d) A diagram that shows the validation (hardware prototyping) environment setup
   e) Descriptions and analysis on your observations, problems encountered, lessons learned, etc.
3) Detailed descriptions of the task(s) that you were not able to accomplish, including reason(s) and/or your analysis.

Report checklist

Follow the "CMPE 125 Lab Report Guidelines" document on Canvas.

Your report should include these sections:

- Cover page *(with name, SID, and the Lab Record filled out with your name, TA's name, and lab demo score)*

- Introduction *(1 paragraph)*

- Design methodology *(1 paragraph for <u>each</u> module: priority encoder, shifter/rotator, ALU)*
    - Block diagrams:
        - Priority encoder (FOR version)
        - Priority encoder (IF version)
        - Priority encoder (CASE version)
        - Shifter/rotator
        - ALU

- Test plan for simulation *(1 paragraph for <u>each</u> module: priority encoder, shifter/rotator, ALU)*
    - **No waveforms** as we are using self-checking test benches.
    - Console output from simulator (as a picture), one for <u>each</u> module.

- Test plan for FPGA board *(1 paragraph for <u>each</u> module: priority encoder, shifter/rotator, ALU)*
    - Include photograph of FPGA board showing results with a caption that clearly labels the state of the switches and LEDs, one for <u>each</u> module.

- Conclusion *(1 paragraph)*
    - In addition if you could not complete any part of the lab explain clearly what went wrong and how you could have remedied the problem.

- Appendix
    - Any tables or other diagrams go here that don't fit in the other sections.

- Source code
    - Use a fixed-width font such as Courier New for the source code.
    - Clearly label the file name of each source file you include.
    - Include **all** source code you used in the project, including constraints.