CMPE 125 Lab 6 Assignments

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

Storage Building Blocks

Purpose:

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

Preparation:

1) Review Lecture Slides 7
2) Read Sections 3.20 and 3.21 of the *CMPE125 Verilog Starter* on class Canvas site.

Tasks:

1. Design and functionally verify a 32x32 register file (module name *regfile2*) with 2 read ports and 2 write ports, following the I/O signal naming given below:

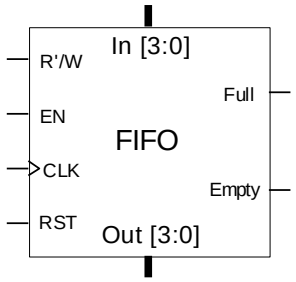
Signal Name	Definition
rd1[31:0]	Read data port 1
ra1[4:0]	Read address for read data port 1
rd2[31:0]	Read data port 2
ra2[4:0]	Read address for read data port 2
wd1[31:0]	Write data port 1
wa1[4:0]	Write address for write data port 1
we1	Write enable for write data port 1
wd2[31:0]	Write data port 2
wa2[4:0]	Write address for write data port 2
we2	Write enable for write data port 2
clk	Clock

Use a **self-checking testbench** for functional verification of the register file. Please make sure that all write and read ports operate properly. Also test that all locations in the register file can be written to and read from. You do not have to go through all possible inputs for each register file location.

Specific Requirement:

Your testbench should have a decent coverage, particularly, you need to test the condition when a write occurs to the register file and the write address is the same as the read addresses (e.g., wa1 = ra1 = ra2 or wa2 = ra1 = ra2) and observe and explain the results. Note: **The two write addresses cannot be the same**, so make sure that during testing, wa1 and wa2 are always set to different values.

2. Design, functionally verify, and FPGA validate an 8x4 (8-deep, 4-bit wide) FIFO queue with the top-level schematics and function table given below (rename the signal *R'/W* as *WNR*).



FIFO Control R'/W	EN	FIFO Operation
x	0	nop
0	1	read
1	1	write

Use a **self-checking testbench** for functional verification of the FIFO. In particular, test the state of the FIFO flags for all possible cases (empty, partially empty or partially full, full) as you fill up and empty out the FIFO.

Specific Requirements:

For FPGA validation, use the Nexys4 DDR's on-board dip switches to generate inputs, and the individual LEDs to display outputs. The system clock can be manually generated by a push button (must be debounced). Do **NOT** use the 4-second clock to control your FIFO.

Contents of Report:

1) Cover page.
2) A list of successfully accomplished tasks.
3) Your test plans (i.e., how to verify the register file and the FIFO)
4) Commented source code (design and verification)
5) Captured verification results (waveforms/simulation log files)
6) Validation (hardware prototyping) environment setup (for the FIFO)
7) Descriptions and analysis on your observations
8) Detailed descriptions of the task(s) you cannot, or did not 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 each module: register file, FIFO)*
    - Block diagrams:
        - Register file
        - FIFO
    - Note: the block diagrams must adhere closely to the lab report guidelines.

- Test plan for simulation *(1 paragraph for each module: register file, FIFO)*
    - TCL console output is required for both modules.
    - No waveforms are required for the FIFO.
    - For the register file you must include a waveform for testing the special consideration outlined above.
    - Console output from simulator (as a picture), one for each module.

- Test plan for FPGA board *(1 paragraph)*
    - Include photograph of FPGA board showing results with a caption that clearly labels the state of the switches and LEDs.

- 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.