

Name: _____ Date: _____

CSSE232 Practical 4 Worksheet

Read and perform the lab guide posted on the course website. Answer the questions in the practical guide. Numbers in square brackets (e.g. [1]) indicate the point value of each question. The annotations in parentheses refer to the list of "general practical requirements" from the practical page. The answers to these questions should be as concise as possible. Maintain the spacing provided by this template. A word doc and pdf version of this are also available.

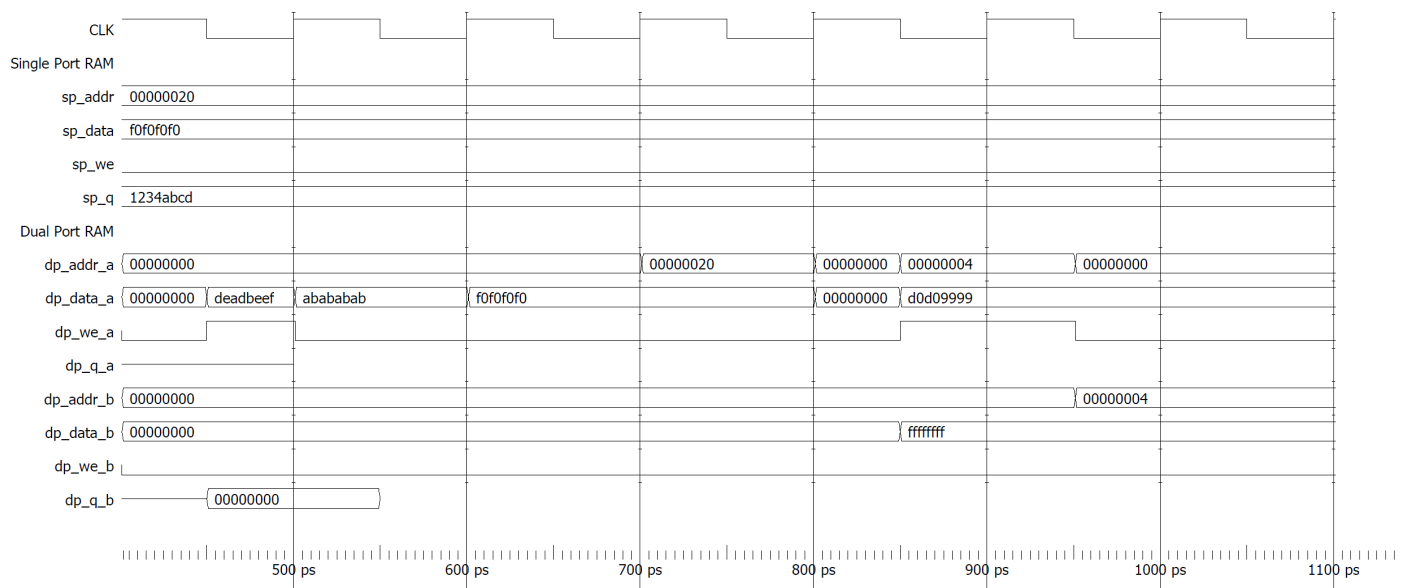
ImmGen

- [8] (Correctness) Describe your strategy for constructing tests for `tb_ImmGen`. Explain how these tests substantially check enough variations for you to be confident that your ImmGen module will work for all input variation.
- [4] Include a screenshot of your ModelSim running `tb_ImmGen`. Make sure the screenshot includes both the waveform and the transcript with the `END_REPORT`.

- [8] (Correctness) The following is a partially-complete waveform from `tb_Memory.v`. Complete the missing signals for `dp_q_a` and `dp_q_b` (use textboxes with borders on word or Acrobat). Assume that all displayed values are hexadecimal and that both memory blocks are initialized to have the following values.

Address:	Value:
0x0000 0000 ->	0x0000 0000
0x0000 0001 ->	0x0000 0000
0x0000 0002 ->	0x0000 0000
0x0000 0000 ->	0x0000 0000
0x0000 0020 ->	0x1234 abcd

Do not reference the instructions nor your ModelSim waveform. Complete this from your understanding of how DP_Memory behaves.



Once you have completed the waveform above, check the correctness with your ModelSim waveform. Are there any differences? Don't change your waveform above, but note down any corrections.

- [8] (Need) Explain what purpose the RegFile will serve when you implement a processor. Where is the source of values connected to RegFile's regnum_a, regnum_b, and write_regnum inputs? Where will the outputs go?
- [8] (Correctness) Describe your strategy for constructing tests for tb_RegFile. Explain how these tests substantially check enough variations for you to be confident that your RegFile module will work for all input variation.

- [4] Include a screenshot of your ModelSim running `tb_RegFile`. Make sure the screenshot includes both the waveform and the transcript with the `END_REPORT`.

- [8] (Need) I asked ChatGPT the following prompt: "Can you give me an implementation of a RISC-V register file in Verilog?" ChatGPT obliged by providing a Verilog implementation that is uncomfortably accurate.

It also summarizes the features of the implemented RegFile with the following items:

- 32 Registers: Indexed from x0 to x31.
- Read Ports: Two read ports (rd1, rd2) for simultaneous access.
- Write Port: One write port (rd).
- x0 Hardwired to Zero: Ensured by ignoring writes to register 0.
- Synchronous Write: Registers update on the rising edge of clk.
- Combinational Read: Outputs update immediately based on inputs.

Is ChatGPT's list of features complete? What is missing from the specifications you had to follow in the instructions? Why do you think ChatGPT missed these specification items? Reflect up on ChatGPT's usefulness as a tool to implement RISC- V modules.

- [8] (Iteration) Briefly describe any early attempts or efforts you abandoned for one of the modules or tests you implemented. If you committed "in progress" work to git, then changed your mind and re-implemented it differently, that is a good thing to explain here.
- [8] What was the biggest challenge in implementing and testing the RegFile? Explain in 100 words or less.
- [10] What is the single biggest thing you learned from writing tests for this practical? Explain in 100 words or less.
- [0] What is the git commit ID for your final commit of your code. This is required to pass the assignment. Check Practical 1 for instructions on how to get the correct commit ID.