LSU EE 4720 **Homework 5** **Due: 8 April 2026**

Formatted 9:14, 2 April 2026

Collaboration Rules

Each student is expected to complete his or her own assignment. It is okay to work with other students and to ask questions in order to get ideas on how to solve the problems or how to overcome some obstacle (be it a question of MIPS, RISC-V, or assembler syntax, interpreting error messages, how a part of the problem might be solved, etc.) It is also acceptable to seek out resources for help on MIPS, RISC-V, etc. It is okay to make use of AI LLM tools such as ChatGPT, Claude, and Copilot to generate sample code. (Do not assume LLM output is correct. Treat LLM output the same way one might treat legal advice given by a lawyer character in a movie: it may sound impressive, but it can range from sage advice to utter nonsense.)

After availing oneself to these resources **each student is expected to be able to complete the assignment alone**. Test questions will be based on homework questions and **the assumed time needed to complete the question will be for a student who had solved the homework assignment on which it was based**.

Student Expectations

To solve this assignment you are expected to avail yourself of references provided in class and on the Web site, and to learn how to handle references that are at first hard to understand, and to keep looking (and asking) when the answer isn't in the first place you look. Some of the problems require thought, and you are expected to persevere until you find a solution. It is each student's duty to him or herself to resolve frustrations and roadblocks, helped along by the satisfaction of making progress. There are plenty of old problems and solutions to look at. One way to resolve issues is to ask Dr. Koppelman or others for help.

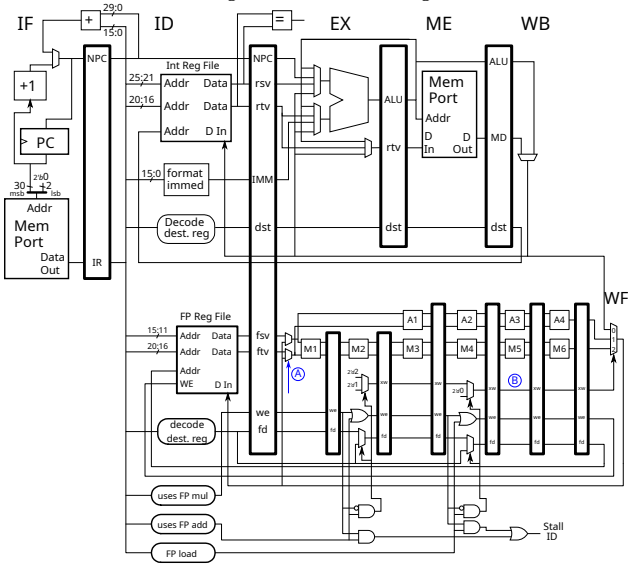
Resources

Many past final exams and homework assignments have problems on FP pipeline variations and control logic.

Problems start on the next page.

Problem 1: *This problem is based on one that appeared on a recent exam, with some enhancements.* Appearing to the right and much larger on the next page is our pipelined MIPS implementation with floating point hardware. This hardware includes multiplexors to bypass values to the FP multiply and addition units. Two wires in the diagram are labeled (in blue) A and B.

Larger on Next Page



Larger on Next Page

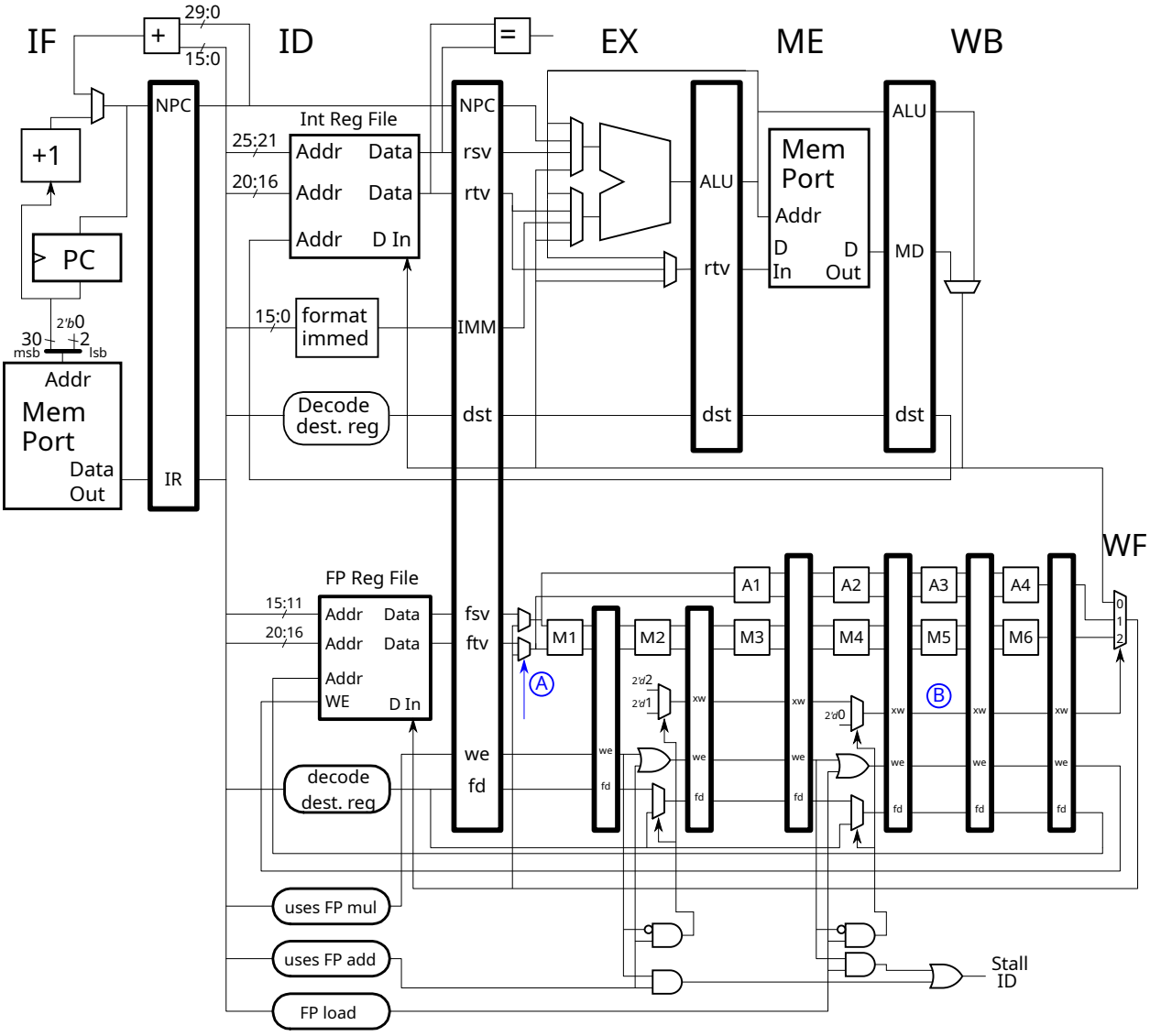
☐ Show the execution of the code below on the implementation to the right. ☐ Check for dependencies. ☐ Don't forget to avoid `WF` structural hazards.

☐ Show the value on the wires labeled A and B **only** for the cycles in which the respective values are used. ☐ Leave a cycle blank when it doesn't matter what the value is (execution would be correct for any value).

```
# Cycle            0    1    2    3    4    5    6    7    8    9    10   11   12 ...
mul.s f1, f2, f3

mul.s f4, f5, f6

add.s f8, f15, f16

add.s f7, f9, f4

lwc1 f11, 0(r1)

add.s f10, f12, f11
```

```
# Cycle            0    1    2    3    4    5    6    7    8    9    10   11   12...
A

B

# Cycle            0    1    2    3    4    5    6    7    8    9    10   11   12...
```

Use the diagram below for Problem 1:



There's another problem on the next page.

Problem 2: Problem 2 from the 2025 Final Exam asks for completing the design of a MIPS FP pipeline in which FP add instructions pass though the same six stages as the FP multiply, but only perform computation in the first four.

SVG versions of the diagrams from the exam problem can be found at `https://www.ece.lsu.edu/ee4720/2025/fe-h-part-a.svg` and `https://www.ece.lsu.edu/ee4720/2025/fe-h-part-b.svg`, a collection of logic gates is at `https://www.ece.lsu.edu/ee4720/2026/c.svg`. All of these can be edited with your favorite SVG editor, mine is Inkscape.

(*a*) Solve 2025 Final Exam Problem 2 (both parts). Though this problem was based on a homework assignment and past exam questions, it can be solved without referring to them. That said, there's nothing wrong with looking at those problems and their solutions.

(*b*) Part b of the exam problem mentions a possible 2026 homework problem in which stall logic is to be designed. This is **not** that problem. Notice that in the illustrated hardware for the exam problem there is no path from the data memory port (in `ME`) to the FP register file, which would be needed for `lwc1`. (Such a path does exist in the FP hardware for Problem 1 of this homework assignment, but that path won't work for this problem.) Add such hardware for FP load instructions, and do it in a way that avoids structural hazard stalls in the same way that the hardware already avoids structural hazards with `add.s` and `mul.s`. That is, a `lwc1` instruction should go through the same number of stages as a FP add and multiply.

- ☐ Add connections so that `lwc1` can execute without risk of a structural hazard at `WF` ☐ by going through the same number of stages ad `add.s` and `mul.s`.
- ☐ Connections should include the value to write back, control signals, and the register number.
- ☐ Pay attention to cost and critical path. ▷ Assume that per-bit, pipeline latches cost twice as much a 2-input multiplexor.