

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.

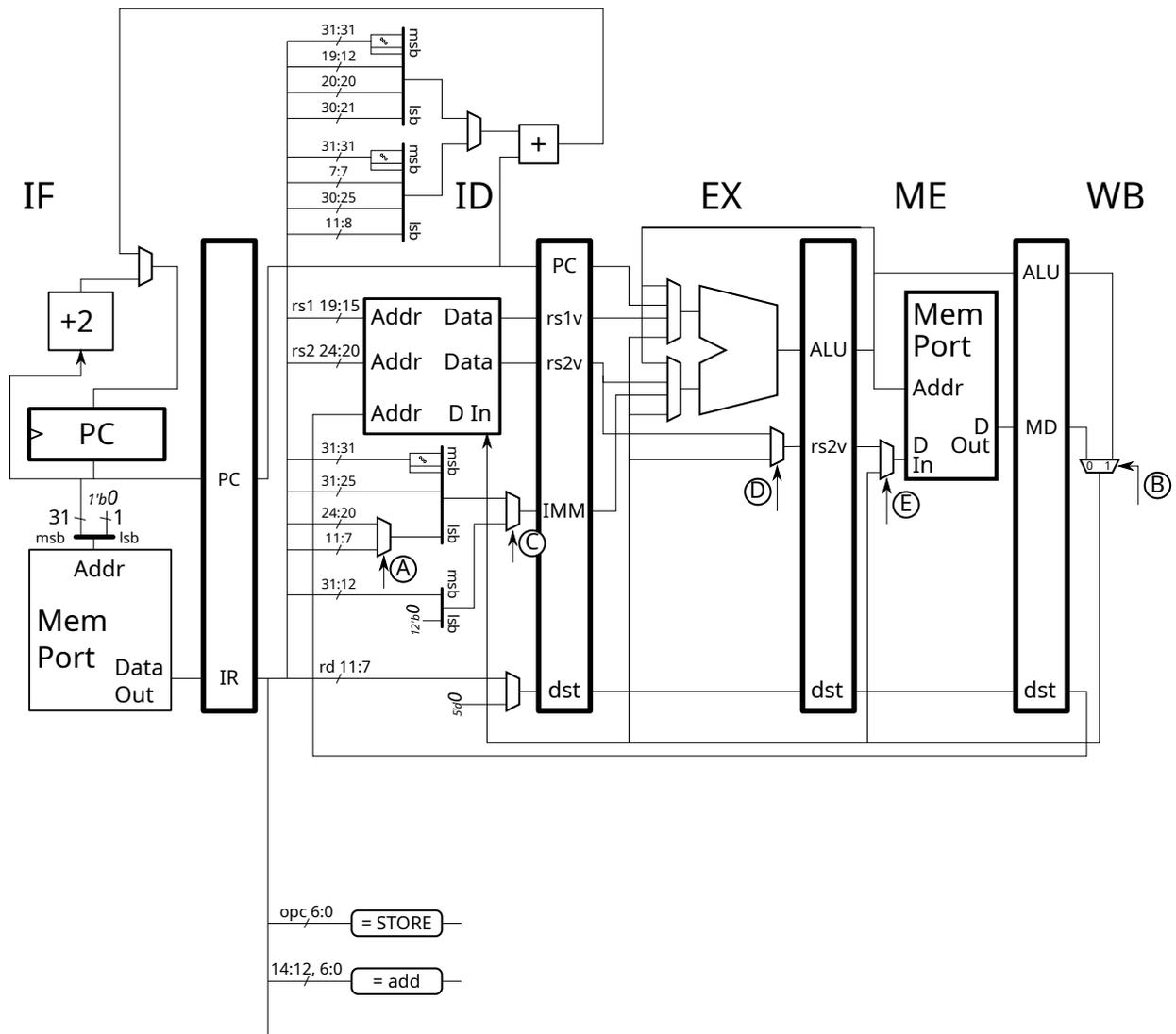
Resources

Many past exams have multiplexor-select-signal design problems, the type of question asked in Problem 2. See, for example, 2024 midterm exam Problem 2. The RISC-V specification is linked to the course site at <https://www.ece.lsu.edu/ee4720/doc/riscv-spec-20250508.pdf> and can be found from the RISC-V site starting at <https://riscv.org/specifications/ratified/>.

Problem 1: Solve 2023 Homework 3 Problem 2, which asks for the select signal values on some multiplexors in a MIPS implementation. Try to solve this problem without looking at the solution. There may be a similar problem on this semester's midterm. Solve the 2023 problem before attempting the next question on this assignment.

Another problem starts on the next page.

Problem 2: Appearing below (and on the last page) is a five-stage implementation of RISC-V RV-32i in which some multiplexor select inputs are labeled. The labels range from A to E. In the subproblems below design control logic for the indicated select signals. An SVG of the illustration can be found at <https://www.ece.lsu.edu/ee4720/2026/hw04-rv-1.svg>, you can use your favorite SVG editor, even if it's not Inkscape, to prepare solutions.



The control logic needs to account for the type of instruction or the exact instruction that's present. There is already control logic that detects store instructions, that's `=STORE`, and that detects an `add` instruction, that's `=add`. Note that instruction type `STORE` refers to many store instructions such as `sw` and `sb`, while `add` just refers to the 3-register `add` instruction. To solve the problems you may need to add additional instruction detection logic. Don't make up instruction types, instead look up the types in the RISC-V specification.

All of the multiplexors for which logic is to be designed have two inputs, making the logic design easier. Because there are two inputs, all one needs to do is detect when one of those inputs is needed and so one only needs to design logic for the easier of the two. For example, for select signal B the 0 input is only used for load instructions, while the 1 input is used for other instructions

that write an integer register. So, use the 0 input if there's a load in WB and the 1 input if not. There is no need to consider all of the instructions that could use the 1 input.

Regardless of which stage the select signals are in, the control signals must be computed in ID and if necessary sent through the pipeline.

(a) Look at Chapter 2 of the RISC-V specification to figure out which instructions affect signals A and C, and of course take into account the illustrated hardware.

- Provide an example of an instruction for which the A signal must be 0 (upper input):
- Provide an example of an instruction for which the A signal must be 1 (lower input):
- Provide an example of an instruction for which the C signal must be 1 (lower input):

(b) Design control logic for select signals A, B, and C.

- Logic for signals A, B, and C.
- The logic must be in ID, pass the select signals through the pipeline when necessary.
- Don't forget that the implementation is pipelined and that each stage will be occupied by a different instruction.

(c) Signals D and E are used for bypassing store values. These bypasses should enable the executions shown below (that is, without stalls):

```
# Cycle      0  1  2  3  4  5
add r1, r3, r4 IF ID EX ME WB
sw r1, 0(r2)   IF ID EX ME WB
```

```
# Cycle      0  1  2  3  4  5
lw r1, 4(r3)  IF ID EX ME WB
sw r1, 8(r2)   IF ID EX ME WB
```

```
# Cycle      0  1  2  3  4  5  6
xor r1, r3, r4 IF ID EX ME WB
addi r2, r2, 4  IF ID EX ME WB
sw r1, 8(r2)   IF ID EX ME WB
```

- Design control logic for select signals D and E.
- The logic must be in ID, pass the select signals through the pipeline when necessary.
- Because these multiplexors are only used for stores and have just two inputs the logic will be simpler than the examples shown in the class notes for the ALU multiplexor.
- Don't forget that the implementation is pipelined and that each stage will be occupied by a different instruction.

An SVG of the illustration can be found at <https://www.ece.lsu.edu/ee4720/2026/hw04-rv-1.svg> and a collection of logic gates (and other symbols) at <https://www.ece.lsu.edu/ee4720/2026/c.svg>. Use Inkscape or any other SVG editor to modify.

