CS4100 Computer Architecture

Spring 2022 Homework 5

Department of Computer Science

National Tsing Hua University

Deadline: 2022/05/26 10:00

- Please use HackMD to view the file.
- In this assignment, we will use a RISC-V pipeline simulator (Ripes) to observe the working of the pipeline. Please set up the Ripes before doing this assignment. After finishing your homework, please submit your assembly code and report by following the requested format on EECLASS.

Environment Setup

We will use a RISC-V pipeline simulator (Ripes) to observe the working of the pipeline. Before starting, let's install the tool and test it. There are different versions of the setup guide for different operating systems (Linux / macOS / Windows).

Assignment

All the work, including source code and report, should be done by yourself. It is not allowed to use any automatic tool or compiler to generate your source code. And plagiarism is strictly forbidden.

1. Assembly Coding (50%)

Please refer to `cmul.S` to write your own assembly code and draw a flow chart of your code. The code function assigned to you is based on **the last digit of your student ID**. Please implement your code base on the template in the individual folder. There are also reference links or reference CPP files in each folder.

- KMP algorithm with `KMP_template.S`: the last digits of 0, 1, 2
- LCS algorithm with `LCS_template.S`: the last digits of 3, 4
- Polynomial Multiplication with `Poly_template.S`: the last digits of 5, 6, 7
- Matrix chain algorithm with `Matrix_template.S`: the last digits of 8, 9
    - Please **run your program with Ripes.** Try to familiarize yourself with the Ripes simulator before doing this part.
    - Please make sure your code can **pass the two test patterns** we provide (listed in the comments of each template).
    - In your report, you must show the evidence of correct results, and provide the **flow chart** of your

implementation.

2. Hazards in Your Code (50%)

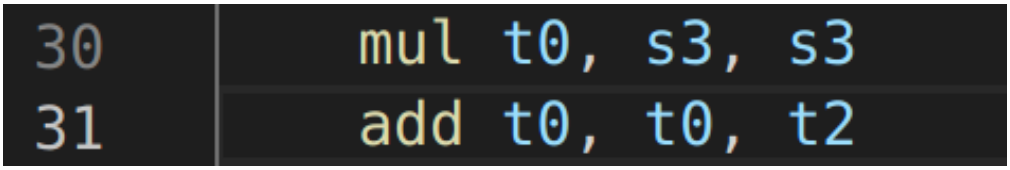
Please find in your code the following dependency types and control hazard:

Type (1): R-types RAW (read after write) at the following 1st instruction.
Type (2): R-types RAW at the following 2nd instruction.
Type (3): Load RAW at the following 1st instruction.
Type (4): Load RAW at the following 2nd instruction.
Type (5): Branch instruction (control hazard).

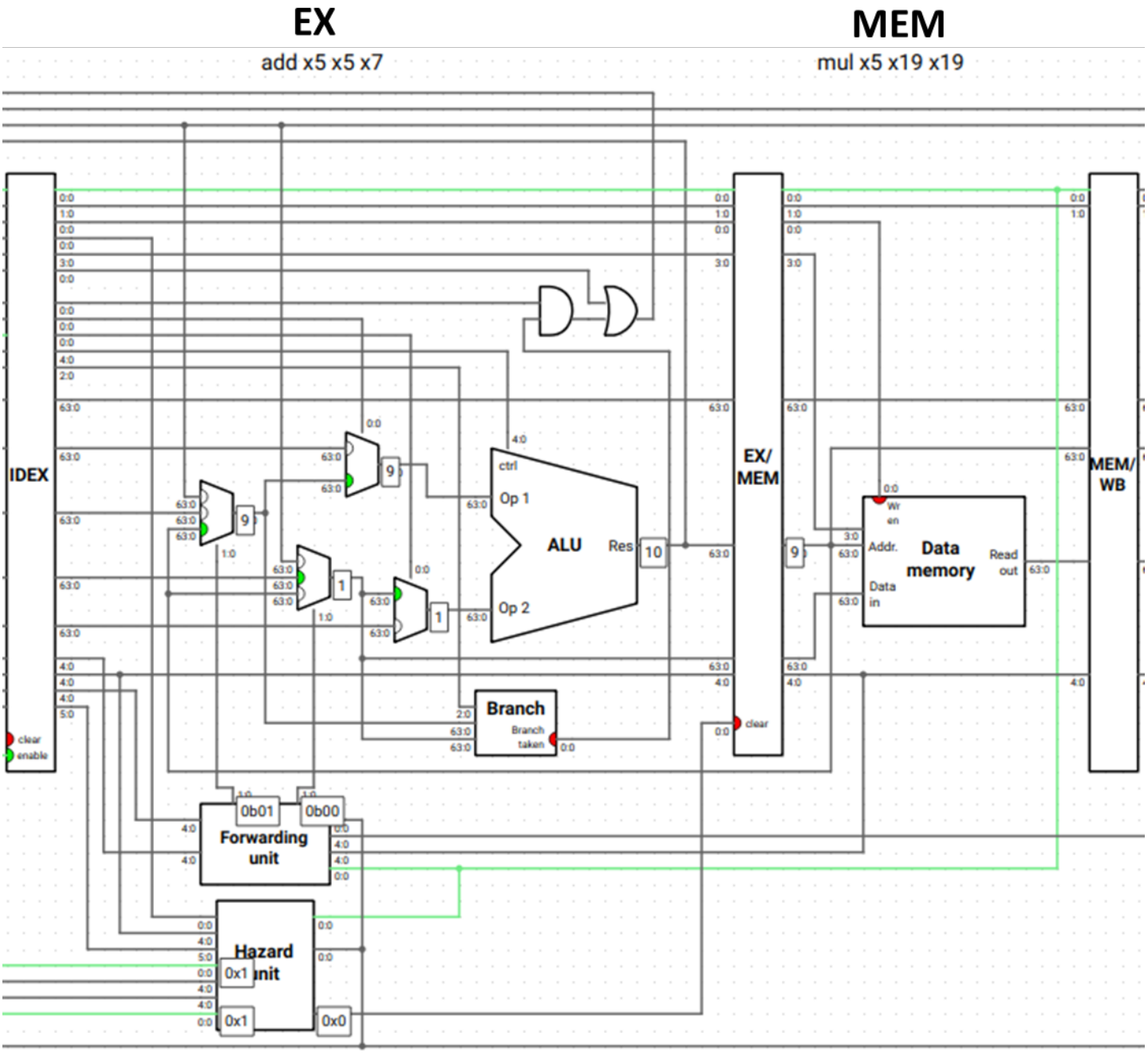
Run your code cycle by cycle with the 5-stage pipeline processor in Ripes. If these dependencies happen in your code, record the corresponded pipeline stages when the dependency happens. Observe and explain how Ripes handles it by using the screenshot.

Note: There can be many cases of the same dependency type in your code. Only one example of each type should be noted. If there is no such type of dependency (which is rare), write a simple code example instead. Remember that you will not get the credit if you fail to locate a specific dependency type in your code, even if you write a simple code example instead.

For example, we report a `Type (1)` dependency (i.e., R-types RAW at the following 1st instruction) as follows:

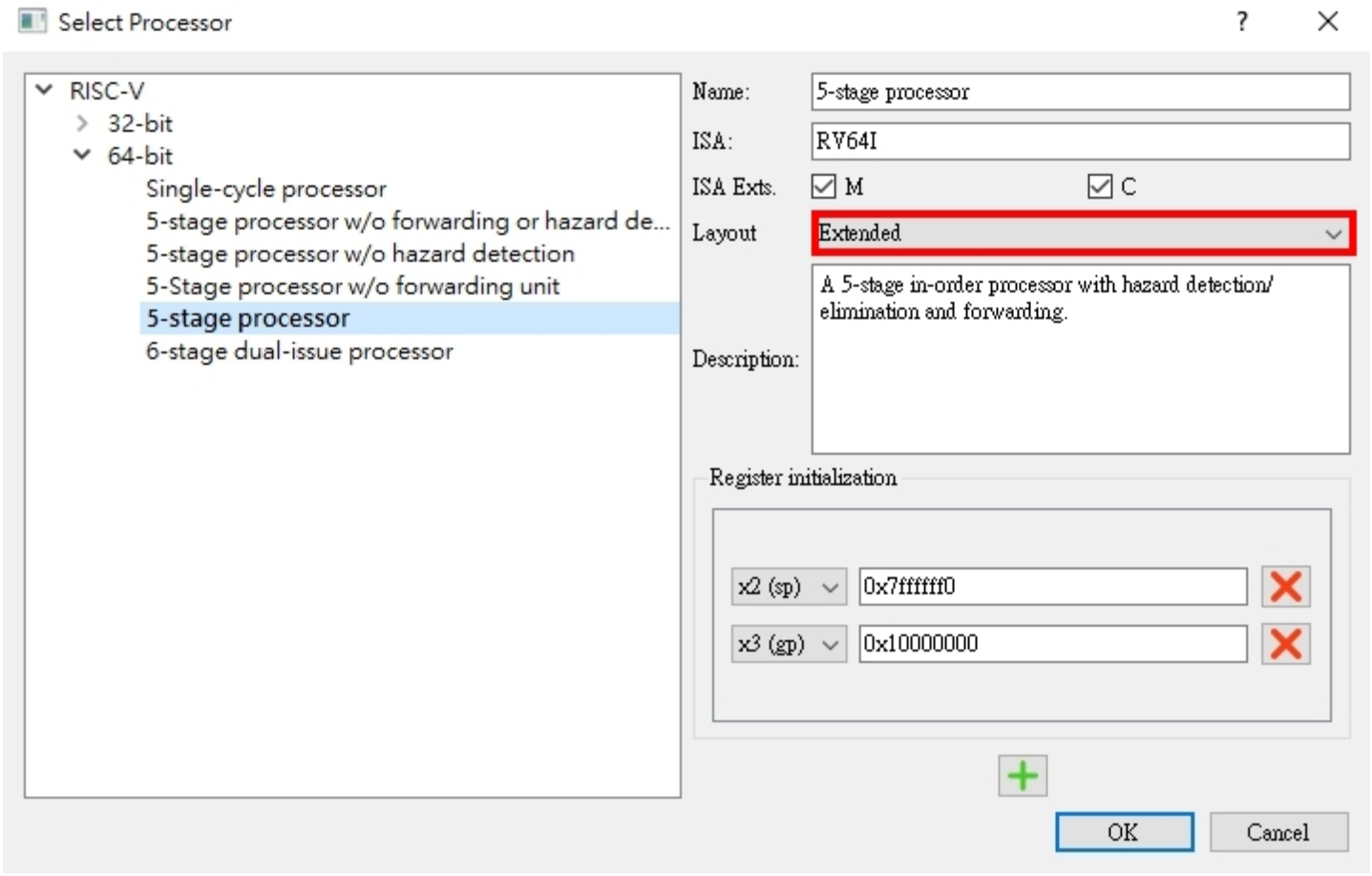


Show the code segment with line numbers in the report. The register t0 is used to store the result of multiplying s3 and s3 in the first instruction. However, t0 is also used in the next instruction `add` as rs1.



Assume that in our example, s3 = 3 and t2 = 0. In Ripes, the processor detects dependency on MEM and EX stages for `Type (1)`. Then, the Forwarding unit sets the control signal (0b01) of MUX before rs1 to select the t0 value forwarded from the MEM stage to the EX stage. **(Note that the MUX control of (0b01) is different from that in the lecture notes and textbook (i.e., ForwardA=10 for the EX hazard). Please follow Ripes' definition in this homework.).** As a result, the rs1 of ALU will be the result of the preceding instruction. You should take a screenshot of your observation and explain the details.

Hint : You can change the processor's layout into the extended mode to see more details of the components of the processor



Submission Rules

- Please submit your report in PDF format and your assembly code.
- The **pdf** file should be named as **HW5_{your student ID}.pdf**
  - (e.g. HW5_123456789.pdf)
- The **source code** should be named as **HW5*{KMP/LCS/Poly/Matrix}*{your student ID}.s**
  - (e.g. HW5_KMP_123456789.s)
- Please upload each file separately.