LSU EE 4720 **Homework 5** Solution **Due: 24 April 2024**

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 or assembler syntax, interpreting error messages, how a part of the problem might be solved, etc.) It is also acceptable to seek out assembly language resources for help on MIPS, etc. It is okay to make use of AI LLM tools such as ChatGPT 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

Some of the problems require thought, and students are expected to persevere until they find a solution. A very good strategy for those who are completely lost is to solve simpler problems on the same topic. It is each student's duty to himself or herself to resolve frustrations and roadblocks quickly, perhaps just by first solving easier problems, perhaps by asking for help. There are plenty of old problems and solutions to look at.

For EE 4720 exams, homework assignments, and their solutions visit
`https://www.ece.lsu.edu/ee4720/prev.html`.

Problem 1: Solve 2017 Final Exam Problem 2 (a) and (b). (The solution is available. For maximum pedagogical benefit make an earnest attempt to solve it. You'll need the practice for the next problem, not to mention the final exam.)

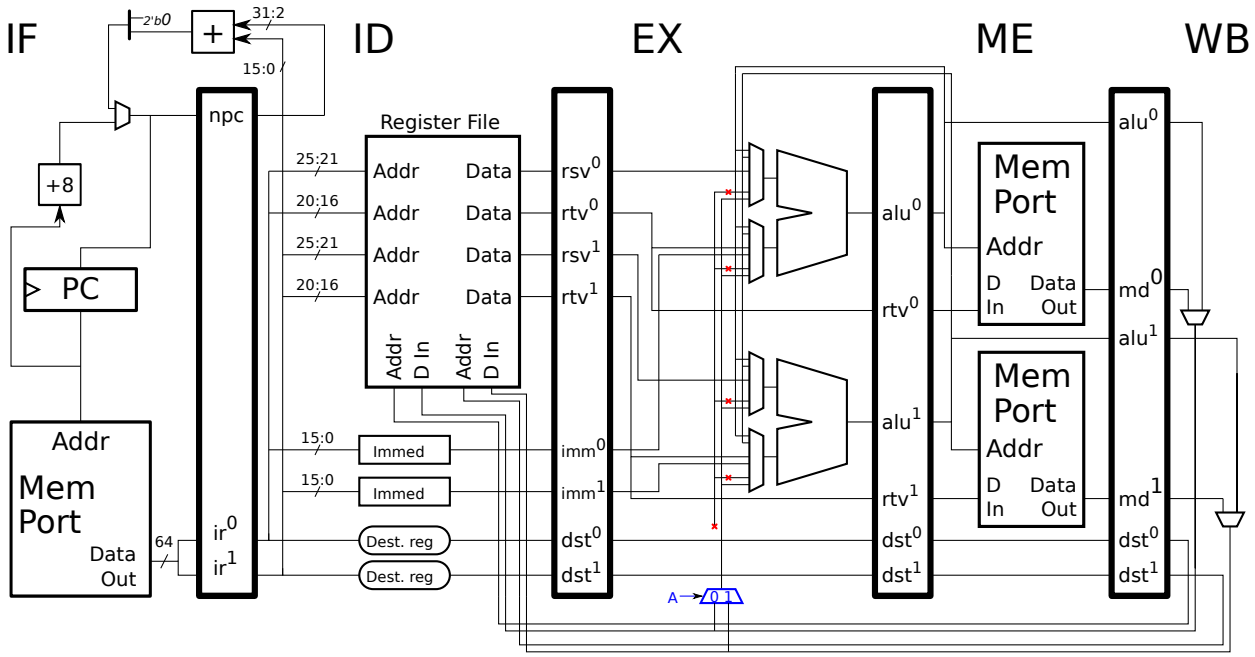
See solution at `https://www.ece.lsu.edu/ee4720/2017/fe_sol.pdf`.

Problem 2: Solve 2023 Final Exam Problem 1c, in which the execution of code on a 4-way MIPS implementation is to be found.

See solution at `https://www.ece.lsu.edu/ee4720/2023/fe_sol.pdf`.

There is another problem on the next page.

Problem 3: The two-way superscalar MIPS implementation below is a reduced cost version of the two-way implementation usually shown in class. Red exes show where bypass connections are removed, and a new multiplexor appears in blue (in the bottom of the `EX` stage).



(*a*) Show a code fragment that would stall on this implementation but would not stall if the exed-out bypass connections were not removed.

The solution appears to the right. The `xor` stalls because it would need to bypass two results from the `WB` stage in cycle 4, one through `r1` and the other through `r4`. If could only bypass one result, and so it stalls in `ID` until cycle 4, which is when the values it needs are written to the register file.

```
#   Cycle          0  1  2  3  4  5  6  7
add R1, r2, r3     IF ID EX ME WB
sub R4, r5, r6     IF ID EX ME WB
and r7, r8, r9        IF ID EX ME WB
or r10, r11, r12      IF ID EX ME WB
xor r13, R1, R4          IF ID -> EX ME WB
slt r15, r16, r17        IF ID -> EX ME WB
#   Cycle          0  1  2  3  4  5  6  7
```

(*b*) Write a code fragment in which the new mux select signal, labeled A, must be 0 in one cycle and 1 in another cycle. Show the value of the select signal in a pipeline execution diagram, leaving the value blank where its value does not matter.

The solution appears to the right. The value of select signal A is 0 if a bypass is needed from the instruction in `WB` slot 0, and 1 if a bypass is needed from `WB` slot 1. In cycle 4 the `xor` instruction needs the value of `r1` bypassed from slot 0 in `WB`. In cycle 5 the `slt` instruction needs the value of `r10` bypassed from slot 1 in `WB`.

```
#   Cycle          0  1  2  3  4  5  6  7
add R1, r2, r3     IF ID EX ME WB
sub r4, r5, r6     IF ID EX ME WB
and r7, r8, r9        IF ID EX ME WB
or R10, r11, r12      IF ID EX ME WB
xor r13, R1, r14         IF ID EX ME WB
lw  r18, 0(r19)          IF ID EX ME WB
slt r15, R10, r16           IF ID EX ME WB
lui  r17, 0xffff            IF ID EX ME WB
A                              0  1
#   Cycle          0  1  2  3  4  5  6  7
```