LSU EE 4720 Homework 5 Due: 12 April 2021

☣ Do not hand in paper copies. Instead, E-mail your solution to koppel@ece.lsu.edu. The preferred format is a PDF file.

Problem 1: Look over SPECcpu2017 run and reporting rules, available at `http://www.spec.org/cpu2017/Docs/runrules.html`. Start with Sections 1.1 to 1.5 and read other sections as needed to answer the questions below.

(*a*) Section 1.2.3 of the run and reporting rules list several assumptions about the tester.

Consider the following testing scenario: The SUT (system being benchmarked) is a new product and that the tester works for the company that developed it. The company spent lots of money developing the product and their potential customers will use SPECcpu2017 when making buying decisions.

☐ Explain why assumptions b and c seem reasonable given the testing scenario above.

☐ Explain why assumption d also seems reasonable, given other stipulations set forth in the run and reporting rules (and discussed in class).

(*b*) The SPECcpu benchmarks can be prepared at base and peak tuning levels (or builds). These are described in Section 1.5. Section 2.3.1 stipulates that base optimizations are expected to be safe.

☐ What is an unsafe optimization? (Points deducted for irrelevant or lengthy answers, especially if they appear copied.)

Does that mean peak optimizations are unsafe? Does that mean peak results can be obtained with unsafe, don't-try-this-at-home optimizations?

☐ Why would it be bad if peak results were obtained with unsafe optimizations?

☐ What rules ensure that optimizations used to obtain peak results aren't too unsafe?

Problem 2: The illustration below is our familiar 5-stage MIPS implementation with the destination register mux and an immediate mux shown. Modify it so that it is consistent with the RISC-V RV32I version as described below. The modifications should include datapath and labels, but not control logic. For this problem use RISC-V specification 20191213 available at
`https://github.com/riscv/riscv-isa-manual/releases/download/Ratified-IMAFDQC/riscv-spec-20191213.pdf`.

The Inkscape SVG source for the image is at
`https://www.ece.lsu.edu/ee4720/2021/hw05-mips-id-mux.svg`. It can be edited with your favorite SVG or plain text editor.

In some ways RISC-V is similar to MIPS, but there are differences. Pay attention to the encoding of the store instructions. Also pay attention to how branch and jump targets are computed.

Be sure to change the following:

☐ Bit ranges at the register file inputs.

☐ The bit ranges used to extract the immediate.

☐ The bit ranges used for the offsets of branch and jump instructions and the hardware used to compute branch and jump targets.

☐ The inputs to the destination register mux (which connects to the `dst` pipeline latch).

☐ The names used in the pipeline latches.

☐ Add or remove unneeded pipeline latches. (Such changes will be needed for branches and jumps.)

Consider the following instructions:

☐ Two-register and immediate arithmetic instructions, such as `add` and `addi`.

☐ The `lui` instruction (which is similar but not identical to MIPS' lui).

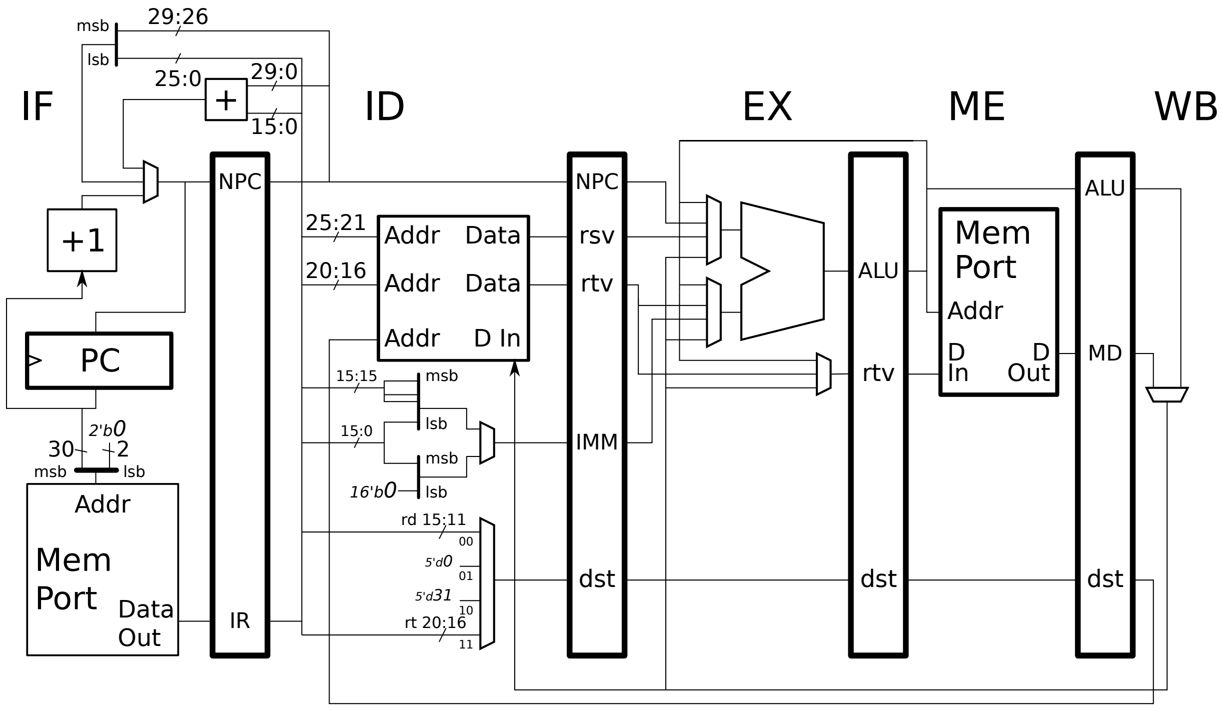
☐ Branch instructions as well as `jal` and `jalr`.

☐ The load and store instructions. (Only the store instructions will require a change beyond what is required for arithmetic instructions.)

Note:

- Do not show control logic such as logic driving mux select inputs.
- Do not show the logic to decide whether a branch is taken.

SVG source at `https://www.ece.lsu.edu/ee4720/2021/hw05-mips-id-mux.svg`.



IF
ID
EX
ME
WB
msb
lsb
29:26
25:0
+
29:0
15:0
NPC
+1
PC
2'b0
30
2
msb
lsb
Addr
Mem
Port
Data
Out
IR
25:21
20:16
Addr
Data
Addr
Data
Addr
D In
15:15
msb
lsb
15:0
msb
16'b0
lsb
rd 15:11
00
5'd0
01
5'd31
10
rt 20:16
11
NPC
rsv
rtv
IMM
dst
ALU
rtv
dst
Mem
Port
Addr
D
In
D
Out
ALU
MD
dst