CSCI 2321 (Computer Design), Spring 2020
Homework 8

Credit: 15 points.

1   Reading

Be sure you have read, or at least skimmed, Chapter 4 up through section 4.4.

2   Problems

Answer the following questions. You may write out your answers by hand or using a word processor or other program, but please submit hard copy, either in class or in one of my mailboxes (outside my office or in the ASO).

1. For this problem your mission is to describe what changes, if any, would be needed to the single-cycle implementation sketched in Figure 4.24 to allow it to execute additional instructions:

   - Would you need additional logic blocks or state elements? If so, what? Please be as detailed as you can; you may want to make a copy of Figure 4.24 and mark it up. (If you can't easily get such a copy let me know and I will figure out some way to do so myself.)
   - Would you need additional control signals?
   - What values would be needed for the existing control signals and any new ones? ("Existing control signals" here refers to the outputs of the logic blocks labeled `Control` and `ALU control`.)

   Hint: It might be helpful to think about what had to be added to Figure 4.17 to have it support `j` (Figure 4.24).

   Instructions to add:

   (a) (7.5 points) The existing instruction `bne`.
   (b) (7.5 points) A hypothetical instruction `lwi` (for "load word indexed") that loads a word from a memory location obtained by adding the contents of two registers. This would be an R-format instruction that in assembly language would look like

       lwi rd,rs(rt)

   where `rs`, `rt`, and `rd` are register numbers, and the result of executing the instruction would be to load into register `rd` the word obtained from data memory at the address given by adding the contents of registers `rs` and `rt`. *Note*: Although syntactically this is an R-format instruction, I'm not sure it's possible to make this work with the normal R-format opcode of `0x000000` without a fairly major change to the design, so assume that it has its own opcode, distinct from opcodes for other instructions. (For extra credit, explain why this seems necessary or at least advisable.) (Credit where credit is due: This question was inspired by problem 4.2 in the textbook.)

3   Honor Code Statement

Include the Honor Code pledge or just the word "pledged", plus *at least one of the following* about collaboration and help (as many as apply).[1] Text *in italics* is explanatory or something for you to fill in. For programming assignments, this should go in the body of the e-mail or in a plain-text file `honor-code.txt` (no word-processor files please).

- This assignment is entirely my own work. *(Here, "entirely my own work" means that it's your own work except for anything you got from the assignment itself — some programming assignments include "starter code", for example — or from the course Web site. In particular, for programming assignments you can copy freely from anything on the "sample programs page".)*

- I worked with *names of other students* on this assignment.

- I got help with this assignment from *source of help — ACM tutoring, another student in the course, the instructor, etc. (Here, "help" means significant help, beyond a little assistance with tools or compiler errors.)*

- I got help from *outside source — a book other than the textbook (give title and author), a Web site (give its URL), etc.. (Here too, you only need to mention significant help — you don't need to tell me that you looked up an error message on the Web, but if you found an algorithm or a code sketch, tell me about that.)*

- I provided help to *names of students* on this assignment. *(And here too, you only need to tell me about significant help.)*

4   Essay

Include a brief essay (a sentence or two is fine, though you can write as much as you like) telling me what about the assignment you found interesting, difficult, or otherwise noteworthy. For programming assignments, it should go in the body of the e-mail or in a plain-text file `essay.txt` (no word-processor files please).

---

[1] Credit where credit is due: I based the wording of this list on a posting to a SIGCSE mailing list. SIGCSE is the ACM's Special Interest Group on CS Education.