

CSCI 2321 (Principles of Computer Design), Spring 2014

Homework 5

Credit: 15 points.

1 Reading

Be sure you have read all assigned sections of chapter 4.

2 Problems

Do the following problems. 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 my mailbox in the department office.

1. (5 points) Do problems 4.1.1, 4.1.2, and 4.1.3 from the textbook, for the following instructions:

(a) `and $t1, $t0, $t0`

(b) `sw $t0, 4($s0)`

Problem 4.1.1 wants you to give values for the following signals: `RegWrite`, `MemRead`, `ALUMux`, `MemWrite`, `ALUop` (can be the name of the operation, e.g., `AND`), `RegMux`, `Branch`.

In problems 4.1.2 and 4.1.3, “resources” means the labeled blocks in Figure 4.2 — Instruction Memory, Registers, etc.

2. (10 points) Do problem 4.7 from the textbook, all parts except that for 4.7.3 you can ignore the sentence “Highlight the path ...”.
3. (Optional: Up to 5 extra-credit points — but I recommend that you at least attempt the problem, since partial credit will be given if appropriate, and I think you will learn something from thinking about it.)

(In the style of the textbook: This problem explores changes to the single-cycle implementation needed to support the `jr` and `jal` instructions.)

Section 4.4 of the textbook discusses how information flows through the datapath of Figure 4.17 for three instructions (`R-format`, `lw`, and `beq` — this is the discussion illustrated by Figures 4.19 through 4.24)). There is also a discussion of what must be added to support the `j` instruction, culminating in Figure 4.24.

For each of the specified instructions (`jr` and `jal`), first describe (as a numbered list of steps) what execution of the instruction needs to involve and what changes (if any) you would need to make to the design shown in Figure 4.24 to make it work. (If you need to make changes to the figure, it might be clearest to print/photocopy the figure and mark it up.)

4. (Optional: Up to 5 extra-credit points — but I recommend that you at least attempt the problem, since partial credit will be given if appropriate, and I think you will learn something from thinking about it.)

(In the style of the textbook: This problem explores changes to the single-cycle implementation needed to support exceptions.)

Section 4.9 of the textbook discusses changes to the pipelined implementation needed to support exceptions. Describe how you could adapt this approach to the single-cycle implementation — that is, describe what changes you would need to make to the design shown in Figure 4.24. (If you need to make changes to the figure, it might be clearest to print/photocopy the figure and mark it up.)