CSCI 2321 (Principles of Computer Design), Spring 2015

Homework 4

Credit: 30 points.

1 Reading

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

2 Notes

- If the assignment asks you to do one or more problems from the textbook, *be sure* you get them from the edition specified in the syllabus; these sets of problems change from edition to edition.
- If a question requires you to do calculations, your odds of getting partial credit are much better if you show enough work to make it clear how you arrived at your answer.

3 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 one of my mailboxes (outside my office or in the ASO).

1. (20 points) In this problem your mission is to trace through what happens during execution of different instructions using the single-cycle implementation of the MIPS ISA as represented in Figure 4.17 in the textbook. You are to assume that at the beginning of a clock cycle the following is true. (`0xN` denotes a base-16 value of `N`, e.g., `0x10` denotes 16 in base 10.)

   - The program counter (PC) has a value of `0x4`.
   - Location `0x4` in the instruction memory contains the binary representation of the MIPS assembler instruction in question.
   - Register and data-memory contents are as described.

   At the point at which values are written into state elements, what values will the following have? Give each in binary form with the appropriate number of bits, or say "not used" if the value does not make any difference to what is saved into the state elements. (E.g., if `RegWrite` is zero, the values of `Write register` and `Write data` are not used.) (Unless otherwise noted, "inputs" or "outputs" here are the ones shown in black — so you don't have to repeat the values of the control signals.)

   - All control signals output from the logic block labeled `Control` (`RegDst`, `Branch`, etc.).
   - Inputs of register file (`Read register 1`, etc.)
   - Inputs and output of the two adders at the top of the diagram and the main ALU, including the control signals `ALU control` and `Zero`.

- Input and output of the block labeled `PC`.
- Inputs and output of data memory.

Instructions to trace:

- `sub $t2, $t0, $t1`, if `$t0` contains the value `0x6` and `$t1` contains the value `0x2`.
- `lw $t0, 4($s0)`, if `$s0` contains the value `0x10000000`, and the data memory starting at `0x10000000` contains the values `0x1`, `0x2`, `0x3`, and `0x4` (with each value occupying 4 bytes).
- `sw $t0, -4($s0)`, if `$t0` contains the value `0x1` and `$s0` contains the value `0x10000008`.
- `beq $t0, $t1, LBL`, if `$t0` contains the value `0x2`, `$t1` contains the value `0x1`, and `LBL` corresponds to the instruction at location `0x18` in instruction memory. Also say what if anything would change if `$t1` contained the value `0x2` instead of `0x1`.

2. (10 points) 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? 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`.)

   Instructions to add:

   - The existing instruction `bne`.
   - 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`. (Credit where credit is due: This question was inspired by problem 4.2 in the textbook.)