Rose-Hulman Institute of Technology
Computer Science and Software Engineering
Prof. Archana Chidanandan

Homework 3
(Machine Language Representation, Exceptions, Register Transfer Language and Datapaths)
Maximum points : 55 points

Directions

This assignment is due Tuesday, Jan. 13th for Sections 1 and 2. Submit your solutions on a separate sheet of paper.

Comment: by 5:00 PM on

Learning Objectives

In the process of completing this homework assignment, students will develop their abilities to

- Interpret machine language instruction formats and fields.
- Describe the implementation of machine language instructions using Register Transfer Language.
- Identify the components required to implement machine language instructions, including their input, output, and control signals, as well as their high-level behavior.
- Design datapaths to support machine language instruction sets by specifying the interconnections between components and the behavior of the associated control units.
- Determine how to handle exceptions, including interrupts.

Problems

1. A computer has a 24-bit word length, and all the instructions are one word in length. Every instruction has a two bit field which specifies the instruction type. The number of registers (the register file) in the architecture of the computer is 26.

   a. [2 pts] For a format that has an opcode field to determine the function of the instruction and three register fields, what is the maximum number of opcodes possible?

   b. [3 pts] For a format with two register fields, one immediate/address field, and a maximum of 64 opcodes, what is the maximum number of bits available for the immediate/address field? How many instructions, of this format, will be required, at a minimum, to copy a 24-bit immediate value into a 24-bit register?

2. Figure 5.33 of Patterson & Hennessy, shows a complete datapath for a multicycle implementation of most R-Type MIPS assembly language instructions, as well as `lw`, `sw`, `beq`, and `j`. For this question, you are required to modify the datapath to include the multicycle implementation of the `addi` instruction. Specifically, you must:

   a. [5 points] Write a multicycle RTL description of an implementation of the `addi` instruction that uses as few cycles as possible without extending the clock cycle of your design.

b. [5 points] List all new and modified components required for the implementation of the `addi` instruction. Also, list the input, output, and control signals for each of those components. Indicate the number of bits in each signal.

c. [5 points] Add any necessary datapaths and control signals to the multicycle datapath. You can photocopy existing figures or download figures from www.mkp.com/cod2e.htm to make it easier to show your modifications. A pdf version of the figure is also available on the class website(Homework).

3. [5 points] For the previous question, you wrote the RTL description for the `addi` instruction. For this question, you must modify the RTL description for `addi`, to handle an ~~interrupt~~ exception i.e. an unexpected event that ~~occurs outside the processor, and~~ causes a change in the flow of execution of the program. *Note:* You do not need to list the new and modified components, nor do you need to show the modified datapath.

Comment: Changed from interrupt to exception

4. [15 points] Repeat Steps 1a, 1b and 1c for the `mfc0` and `mtc0` MIPS instructions (combine the component lists and the datapath modifications). *Hint:* Page A69 of Hennessey and Patterson, has a description for these two instructions.

5. [15 points] Repeats Steps 1a, 1b and 1c for an "undefined" instruction in MIPS. *Hint:* Save PC-4 in EPC, modify the PC, and update the Status Register. Read through pages 410-413 and pages A32-A35 of Patterson and Hennessy for more information.