CPSC 418	Advanced Computer Architecture	Spring 1996

Homework 4: Due 30 Jan

Problem 1 (3 Points) **Function Calls and Parameters**

The function `bigSub` has the following parameters:

input params:

name	constant or variable	type
I1	constant	integer
F1	constant	float
I2	variable	integer
AI	constant	array of integers
AF	variable	array of floats
S	constant	string

result: R : float

where "constant or variable" says whether assignments to the parameter within `bigSub` also effect the parameter in `main` (*e.g.* variable = `var` in Pascal or `&` in C).

Using the F&R conventions, describe how each parameter is passed (*e.g.* value of `X` in `R5`) and whether a copy of the parameter must be made before calling `bigSub`.

Problem 2 (4 Points) **Word Addressing**

Feldman & Retter 2.7. Note: your code fragment should end with the `Z` flag set to true if the two addresses point to the same byte.

Problem 3 (2 Points) **CALL Inst**

In the F&R instruction set architecture (ISA), is the `CALL` instruction necessary? In other words, can you write a sequence of one or more instructions that has *exactly* the same behaviour as a `CALL`? If so, write the code sequence, otherwise say precisely why you can not write the sequence.

Problem 4 (2 Points) **LODSET Inst**

In the F&R instruction set architecture (ISA), is the `LODSET` instruction necessary? In other words, can you write a sequence of one or more instructions that has *exactly* the same behaviour as a `LODSET`? If so, write the code sequence, otherwise say precisely why you can not write the sequence.

Problem 5 (3 Points) Special Registers: PC, SP, RA

In Figure 2.4 F&R give the symbolic register definitions:

PC = R31
SP = R30
RA = R29

For each of the definitions, in which one of the following places does it make the most sense for it to occur?

- in the *micro-architecture* (*e.g.* different CPUs for the same ISA can have different definitions)
- in the *ISA* (*e.g.* all CPUs must use the same definition)
- in the *compiler* (*e.g.* different compilers for the same ISA can have different definitions)

Be sure to take into account how the CPU handles traps and interupts when answering the questions. Justify your answers.

Problem 6 (5 Points) **Traps and Interupt Levels**

The keyboard (`KBD`) and memory allocation (`MEM`) routines use the following sequence of interupt levels (IL):

`KBD`	100 clock cycles at IL1
`MEM`	50 clock cycles at IL0
	50 clock cycles at IL1
	50 clock cycles at IL2
	50 clock cycles at IL1
	50 clock cycles at IL0

It takes the CPU 10 cycles to respond to an interupt signal.

A user's process (`P1`) wants to run for 100 cycles at IL0, request more memory via the `MEM` routine, and then run for another 100 cycles.

Draw a graph of interupt level vs. time for time=(0 to 550) describing what happens if `P1` is running and at clock cycle 250 a user hits (CTL-C) for process `P2`. Be sure to clearly label what happens at each time, why the IL changes, and whether the CPU switches processes because of a CALL, TRAP, RFI, or interupt.

Problem 7 (1 Points)

How much time did you spend total on the assignment and on each of these problems? Which problems were useful or useless? How well did the lectures and text book prepare you for this homework? Which concepts needed better explanation?