CSC 230 - Assignment 2 - Part 1 (Written)

Due Wednesday June 25 by 5:00 p.m.

Total Marks = 50

Question 1. [6] Sometimes software optimization can dramatically improve the performance of a computer system. Assume that a CPU can perform a multiplication operation in *10 ns*, and an addition or subtraction operation in *1 ns*. Assume that the operands (*x* in the question below) are already loaded into registers.

(a) How long will it take for the CPU to calculate the result of $x^2 + 2x$?
(b) Could you optimize (change/factorize) the equation so that it will take less time? Show all your steps and comment your reasoning in order to get full marks.

Question 2. [8] For each feature listed below, indicate with an "X" in the appropriate box(es) whether it *usually* appears in a RISC-based system, in a CISC-based system, or in neither. It is OK to tick both the RISC and CISC boxes if the feature applies to both. (Note the emphasis on the word '*usually*' – no common computers can be described as being pure-RISC or pure-CISC in nature.)

Feature	RISC	CISC	Neither
Has a large number of general-purpose registers			
Operands must be in registers for arithmetic instructions			
Has a simple instruction set			
Has a single instruction size (all instructions occupy the same number of bytes)			
Has a single data operand size (all memory operands occupy the same number of bytes)			
An instruction is fetched in each clock cycle			
Every instruction is 4 bytes in size			
The typical instruction allows many addressing modes for its operands			

Question 3. [2] The following is *not* an ARM instruction, but it could be an assembly language instruction in some other processor.

```
LOAD      R3,[[R4],#8]          @ R3 = *((*R4)+8)
```

Its semantics is shown using a C-like expression, as it appears in the comment, assuming that R4 is a pointer containing a valid address in memory and R3 is an integer variable.

Show how you would implement this LOAD task using only 2 ARM instruction(s).

Question 4. [2] Consider the following C statement and the declaration of variables in:

```
int   R3;
int   *R4;
R3 = (*(*R4))+4;
```

Show how the same functionality can be implemented using only 2 ARM instructions.

Question 5. [14] Figure 1 shows the main components of a CPU, its interface to memory via buses and its internal datapath. In the "Fetch/Decode/Execute" phases of an instruction, the "Fetch" phase is the same for every instruction. (The Fetch phase includes the operation of incrementing PC by 4; all instructions are 4 bytes in size.)

The sequence of steps for the Fetch phase is shown in Table 1.

TABLE 1. Steps Performed during the Fetch Phase of an Instruction

Step	Action
1.	PC → MAR
2.	MAR→ address bus
3.	Read signal → control lines
4.	PC → B input of ALU
5.	0 → MUX to select 4 as input

Step	Action
6.	Add signal → ALU control lines
7.	Z → PC
8.	Wait for memory to finish reading
9.	Data bus → MDR
10.	MDR → IR

Imitate the style and level of detail when answering parts (a) and (b) below.

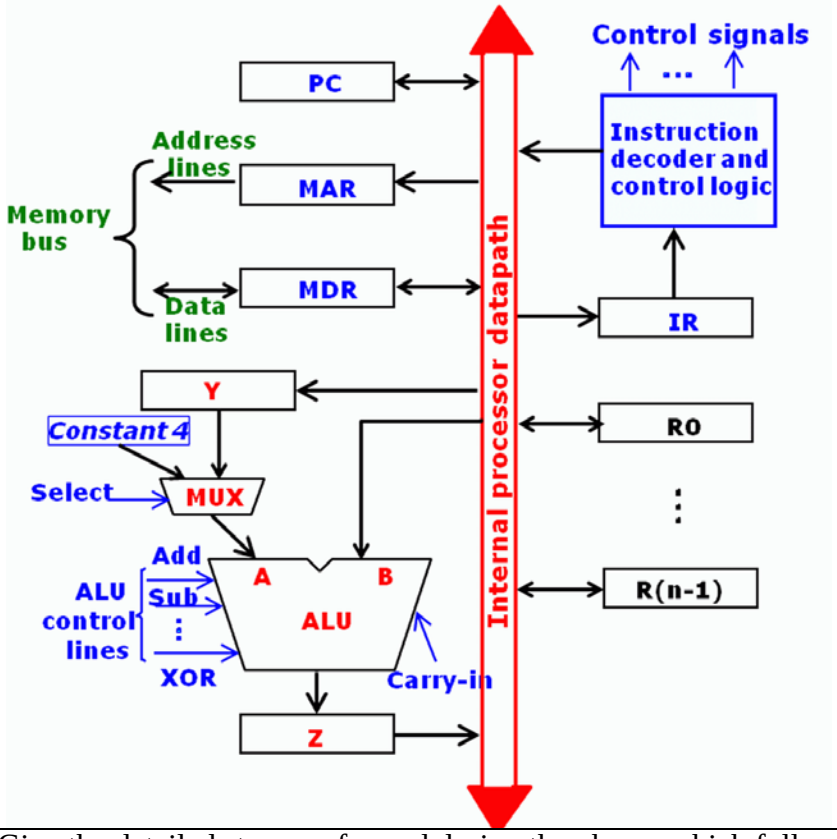

Figure 1: Possible internal organization of a processor

(a) [4] Give the detailed steps performed during the phases which follow Fetch for the instruction:

```
LDIND  R1, [R2]
```

where `LDIND` is the Load-Indirect instruction (not a regular ARM instruction). In this example, register R2 holds the address of a word of memory, and that word in turn holds the address of another word of memory. It is the word at that second address which is copied into register R1. The semantics can be expressed as:

```
R1 = Memory[ Memory[R2] ]
```

(b) [4] Give the detailed steps performed during the phases which follow Fetch for the instruction:

```
BLX  R1
```

where BLX is the Branch-with-Link-and-Exchange instruction. It causes the address of the instruction following the BLX to be copied into register R14 and an immediate transfer of control to the address held in register R1.

(c) [2] BLX is an ARM instruction, however LDIND is not. What ARM instruction(s) would you have to write to achieve the same effect as `LDIND R1,[R2]`?

(d) [4] Repeat the process in (a) and give the detailed steps performed during the phases which follow Fetch for the ARM instruction:

```
STR  R1, [R2,R3,LSL #2]!
```

Question 6. **[12]** Fill in the table below for each ARM instruction.

Instruction	Explain what the instruction does in precise english
`LDR R6,[R3,#8]!`	Explain precisely what the instruction does
	State what is the addressing mode of **`[R3,#8]!`**
`STRB R3,[R6],R7`	Explain precisely what the instruction does
	State what is the addressing mode of **`[R6],R7`**

`LDR R5,=Temp`
`where`
`Temp: .word 4`
	Explain precisely what the instruction does

	State what is the addressing mode of **`=Temp`**

Instruction	Explain what the instruction does in precise english
`EOR R1,R2,R2,ROR #4`	Explain precisely what the instruction does
	State what is the addressing mode of `R2,R2,ROR #4`

Question 7. [6] Check your web page on Connex for the assignments and follow the links to the 2-part (fun) articles on the history of the ARM processor. A PDF printout of the articles is available but you may want to look at the original links at:
http://www.reghardware.com/2012/05/02/
unsung_heroes_of_tech_arm_creators_sophie_wilson_and_steve_furber/
and
http://www.reghardware.com/2012/05/03/
unsung_heroes_of_tech_arm_creators_sophie_wilson_and_steve_furber/

Read the story (it is interesting in many aspects). Answer the following questions on a separate piece of paper (only 1 piece at most):

(a) Does ARM manufacture ARM processors? If yes, state where the main production facilities are located (you may want to explore beyond the articles); if not, state what it is they actually produce and sell.
(b) Where did ARM get the idea of being a RISC-type processor?
(c) Where is the main location/base for ARM nowadays?
(d) Who bought the first prototype of the processor?
(e) What was the biggest flaw that non-RISC 16-bit processors (e.g. from Motorola) had in common in the early 19080's such that Wilson and Furber needed new ideas?
(f) What is the most significant feature in ARM family of processors?
(g) What is the connection between the Italian heritage of your instructor and ARM?
(h) Which american company was the first to own a large percentage of ARM?
(i) Has ARM sold more or fewer processors than Intel? Either way, how many more or fewer?
(j) Which would you say, in your opinion, is the most striking human-interest story embedded in the history of ARM?
(k) How many transistors were used in the first ARM?