NAME: ______________________ STUDENT NUMBER: V00____________

UNIVERSITY OF VICTORIA EXAMINATIONS AUGUST 2011

C SC 230 - Introduction to Computer Architecture and Assembly Language CRN 30178

TIME: 3 hours
INSTRUCTOR: Julie Beeston
TOTAL MARKS: 117
TO BE ANSWERED ON THE PAPER

Question No.	Value	Mark	Question No.	Value	Mark
1	15		8	14	
2	2		9	3	
3	3		10	3	
4	12		11	10	
5	8		12	10	
6	4		13	10	
7	6		Total	100	

INSTRUCTIONS:

1. STUDENTS MUST COUNT THE NUMBER OF PAGES IN THIS EXAMINATION PAPER BEFORE BEGINNING TO WRITE, AND REPORT ANY DISCREPANCY IMMEDIATELY TO THE INVIGILATOR
2. This examination paper consists of 16 pages including this cover page plus 2 pages of Appendix (1 page double sided).
3. No aids are permitted. However, an Appendix describing the ARM instruction set is provided for your use.
4. The marks assigned to each question are shown within square brackets. Partial marks are available for all questions.
5. Please be precise but brief, and use point form where appropriate.
6. It is strongly recommended that you read the entire exam through from beginning to end before beginning to answer the questions.

Question 1. [15] Consider the following ARM code segment. Assume that all registers hold 8 bits and r1 has an initial value of 10110000 in binary

```
[1]           mov    r0, #0
[2]   loop:   movs   r1, r1, LSL #1
[3]           addcs  r0, r0, #1
[4]           bne    loop
```

a) [8] What are the vales of R0 and R1 each time the code reaches line [4]? (Note there may be more lines than needed.

Loop	R0	R1
Initial Value	0	10110000_2
1		
2		
3		
4		
5		

/10

b) [2] Describe in a single sentence what this function does.

c) [5] 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 the PC by 4; assume all instructions are 4 bytes in size.)
The sequence of steps for the Fetch phase is shown in Table 1 and it follows the type of CPU organization seen in the lecture notes. Imitate the style and level of detail when answering parts the question below.

Step	Action
1.	PC → MAR
2.	MAR → Address Bus
3.	Read Signal → CTRL
4.	PC → ALU
5.	#4 → ALU input

Step	Action
6.	Add Signal → ALU CTRL
7.	ALU Output → PC
8.	Wait for mem read to finish
9.	Data bus → MDR
10.	MDR → IR

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

Give the detailed steps performed during the phases which follow Fetch for the instruction:

movs r1, r1, LSL #1

/5

Question 2. **[2]** What are the two main differences between a subroutine and an interrupt-service routine?

/2

Question 3. **[3]** Assume that dynamic random access memory (DRAM) must be refreshed every 64ms. Suppose that we have a DRAM chip which has 2000 rows of 8 bytes each, and that each row can be refreshed in 4 clock cycles. The clock speed is 1MHz (1,000,000 cycles per second).

(a) [1] How many clock cycles are needed to refresh the entire DRAM chip?

(b) [1] Given the number of clock cycles from above, how much time, in ms, are they equivalent to?

/3

(c) [1] Let your answer from above be N ms. What fraction of time is spent refreshing the DRAM chip? (Give you answer as a simple expression using N – you do not need to evaluate it.)

Question 4. [12] Fill in the table below with the appropriate information about the instructions.

Instruction	Addressing Modes (ALL Operands)	What it does (Use appropriate notation).
LDR R0,=X		
CMP R1,#0		
STR R1,[R2,#4]!		

/12

Question 5. [8] Figure 2 shows the possible organization of a 4-stage pipeline.

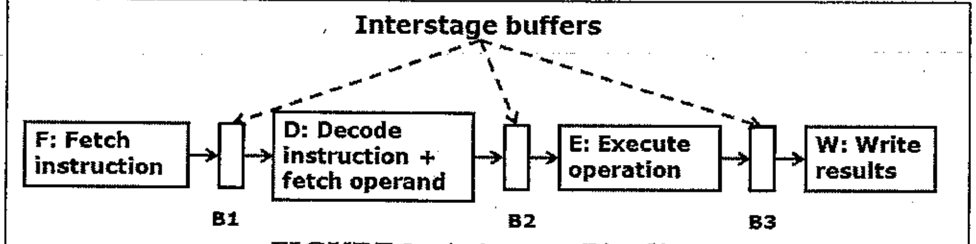

FIGURE 2. A 4-stage Pipeline

a) Explain what happens during the first 6 clock cycles given this structure for a sequence of instructions labeled I_1, I_2, I_3, I_4, and I_5. That is, show in which phase of the pipeline they are.

	Time →					
\ Clock \cycle Instruction	1	2	3	4	5	6
I_1						
I_2						
I_3						
I_4						
I_5						

/3

b) Define the speedup which could possibly be obtained, in the best case, by using a pipeline with m stages executing N instructions. Using your table above, give a good explanation of your answer (do not just state a formula).

/3

c) In the example above, which time slots are the most efficient/effective in terms of CPU usage? Why?

/1

d) Given this structure, what is the penalty incurred from a branch?

/1

Question 6. [4] Consider the sequence of ARM instructions listed below. Where in this sequence might pipelining hazard(s) occur? Give the line number of the instruction, and state which kind of hazard it could be. Assume that the branch in line 5 is taken.

```
[1]       ...
[2]       LDR   r1,=myData
[3]       LDR   r1,[r1]
[4]       SUBS  r2,r2,#1
[5]       BEQ   L1              @taken!!!
[6]       STR   r1, [r5]
[7] L1:   STR   r1, [r4]
[8]       ...
```

/4

Question 7. [6] Amdhal's law was discussed, stated as:

$$\text{Speedup} = \frac{1}{f + \frac{-f}{p}}$$

(a) [2] Define f and p.

/6

(b) [4] What is the meaning of this law, or what is its impact? What does it say about increasing the number of processing elements and the effect on performance?

Question 8. [14] Consider a computer which has a virtual memory system with the following characteristics. The computer has a 32-bit address space with byte addressable memory; this implies a virtual address space which is 2^{32} bytes or 4 GB in size. The main memory (RAM memory) is 1 GB in size. Pages are 64 KB in size.

A) [1] How many entries does the page table have?

B) [1] How many bits are necessary to represent the page table numbers (given the number of pages stated above)? Also state the number of hexadecimal digits required.

C) [6] Suppose that the CPU generates the sequence of memory addresses shown below (written in hexadecimal) and that the page table contains the entries shown on the left. Invalid (unmapped) entries in the page table are shown as empty. Show the corresponding sequence of addresses in main memory which the virtual addresses are mapped to. If any address cannot be mapped, show your answer as the words 'page fault'.

/14

0	0x1234
1	0x2257
2	0x8048
3	-
4	0x1387
5	0x0001
6	-
	...

Page Table

0x00051255	→	
0x00024788	→	
0x0003851	→	
0x00041121	→	
0x00066666	→	
0x00001111	→	

Virtual Table **Corresponding real address**

D) [6] Suppose that a program is running and incurs a page fault. All frames in the main memory are currently in use. What does the operating system have to do to allow the program to continue execution? Your answer should be written as a sequence of steps taken by the operating system. (Note assume write back.)

Question 9. [3] Suppose that execution time for a program is directly proportional only to instruction access time. Access time to an instruction is 2 ns from the cache and 18 ns from memory. The probability of a cache hit is 98%. In the case of a cache miss, the instruction is fetched from main memory and copied into the cache, and then a second access must take place to copy the instruction from the cache (this time it will be a hit).

(a) [1] Compute the execution time of a program with 100 instructions without the cache (an expression is fine).

(b) [1] Compute the execution time of a program with 100 instructions with the cache (an expression is fine).

/3

(c) [1] If the cache size is doubled, the probability of not finding an instruction is cut in half. Compute the execution time of a program with 100 instructions with the larger cache (an expression is fine).

Question 10. [3] Give 3 general differences between L1 (primary) cache and L2 (secondary) cache.

/3

Question 11. [10] Fill in the missing portion of the program below that determines the largest integer in an unordered non-empty array.

```
@This program determines the largest integer in an unordered
@ non-empty array
@Pseudo- code:
```

A)[3] Fill in the Pseudo code here. NOTE: C CODE is acceptable.

/3

```
@ Register use: r1 <->array size; r2<->array address;
@ r3<->tempmax; r0<->index count and r4<-> array element

        .text
        .global _start
        .equ    EXIT,0x11
_start: LDR     r1,=size            @get array size
        LDR     r1,[r1]
        LDR     r2,=array       @get start of array
```

B) [3] Fill in the main section of the code here

/3

```
exit:   LDR     r2,=max             @store tempmax in variable max
        STR     r3,[r2]
        SWI     EXIT
        .data
size:   .word   10
array:  .word   10,-2,12, 13,-5,6,9,11,13,-2
max:    .skip   4
        .end
```

C) Write the equivalent of the following C code in ARM Assembler

```
#define SEED 3
int random = SEED;
for (int i = 1; i < 20; i++)
{
    print('*');
    random = random + 3
    random = random & 0x3     // Bitwise And
    if (random == 0)
        break;
}
```

```
.equ    SWI_PrStr, 0x69
.equ    Console, 1
.equ    SEED, 0x3  @A seed for a pseudo random number
```

/3

```
.data
    Star: .asciz  "* "
```

b) Would it be more efficient to restructure the loop so it counts down from 19 to 0 instead? If so, why?

/1

Question 12. **[10]** Some first year students ask you questions and you need to answer them clearly and precisely, with the main goal being of complete understanding on their part, without writing a long essay. Show what a great *Teaching Assistant* you could be!

(a) **[3]** Can you state at least 3 main characteristics which differentiate between static and dynamic RAM? Explain how they apply to each.

(b) **[2]** Where is static RAM used mostly and where is dynamic RAM used mostly?

(c) **[1]** What is volatile and non-volatile memory?

/10

(d) **[1]** What is an example (or application) of non-volatile memory?

(e) **[3]** Access to data on disk is much slower than to data in main memory. What are the factors in computing disk access time? Just state, do not need to describe

Question 13. [10]

You are involved in the design of a system which needs to have a 32-bit address bus and a 16-bit data bus. The system is expected to be byte-addressable for everything and a word is defined to be 16 bits (2 bytes). There are both peripherals and memory units to be connected within it and it is expected that the whole address space will be used. The diagram on the next page may help you visualize the required answers to the questions. State the answers in the appropriate boxes. Leave the values in the answers as powers of 2 or simple approximations to them or simple factors - there is no need to give the actual number.

j) [1] What is the total number of addressable locations for this system?

A)

h) [1] ½ of the addressable space is to be used for the peripherals requirements. What is the total number of available addresses for peripherals?

B)

The addressable space for peripherals is further subdivided as shown below.

g) [1] ½ of the addressable space is to be used for the memory requirements of RAM and ROM. What is the total number of addresses available for memory?

C)

e) [1] The ROM has already been bought and its size is 32 KB. How many addresses does it need?

H

d) [1] How many addresses are left for RAM?

I)

a) [1] ¼ of the addresses are to be used for input components. How many?

D)

b) [1] ½ of the addresses are to be used for disk addressing. How many?

E)

c) [1] . ¼ of the addresses are to be used for output components. How many?

F)

i) [1] The chosen disk drives require 256MB of address space. How many can one fit into the system??

G)

f) [1] The cheapest RAM chips contain 16 K **words**. How many should you buy to fit into this system?

J)

/10

This page is intentionally blank. Use it for scrap paper if needed.

The End