CSC 230 - Assignment 1 - Part 1 (Written)

Due Wednesday, June 4, by 5:00 p.m. Total Marks = 50

Answer the following questions directly on this paper and hand in your answers in the CSC 230 box in the ECS corridor on the 2nd floor (by 5 pm on Wednesday June 4). No electronic submissions are accepted. Your pages should be stapled together, with identification *on every page* (student number at least). A neat and professional appearance is expected and marks will be deducted otherwise.

General notes.

- A word-addressable system (or portion of) implies that the smallest location which can be addressed and whose content can be transferred is exactly one word, whatever size the word is. In this system one needs an address for every word.
- A byte-addressable system (or portion of) implies that the smallest location which can be addressed and whose content can be transferred is exactly one byte. In this system one needs an address for every byte.
- Reminder: 1K=1,024=2^{10}, 1M=1,024K=2^{20} and 1G=1,024M=2^{30}
- Most answers should be computed and given as power of 2, or as power of 2 with a factor.

Question 1. [1] Is the JVM (Java Virtual Machine) a little-endian or a big-endian machine? (Is this a precise enough question?)

Question 2. [3] The C language provides two different ways to process the elements of an array, as illustrated by the following two versions of a function for summing the elements of an array called "M", whose size is K elements.

```
int Sum1( int *M, int K ) {
     int i, ss;
     i = ss = 0;
     while( i<K ) {
          ss += M[i]; i++;
     }
     return ss;
}
```

```
int Sum2( int *M, int K ) {
     int *MEnd, ss;
     ss = 0;
     MEnd = M+K;
     while( M < MEnd ) {
          ss += *M; M++;
     }
     return ss;
}
```

(a) [2] What is the C datatype of M in *Sum1* and *Sum2*? Tick the most appropriate answer below:

□ **int** □ **array of int** □ **pointer to int** □ **pointer to array of int**

(b) [1] Suppose that *Sum2* is called with the values `0xA000` for `M` and `7` for `K`. What value, in hexadecimal, is stored in variable `MEnd` by the statement which initializes it? Tick the correct answer.

□ `0xA01C` □ `0xA028` □ `0xA018` □ `0xA007`

Question 3. [2] A program compiled for a SPARC ISA writes the 32-bit unsigned integer 0xABCDEF01 to a file, and reads it back correctly. The same program compiled for a Pentium ISA also works correctly. However, when the file is transferred between machines, the program incorrectly reads the integer from the file as `0x 01EFCDAB.` What is going wrong? Can you identify a possible reason for the behaviour (without going into deep explanations)?

Question 4. [4] Figure 1 shows the bus organization when there is a single bus for memory and peripherals. This is called a "Memory-Mapped" topology.

(a) [2] State at least one advantage and at least one disadvantage of such a bus topology.

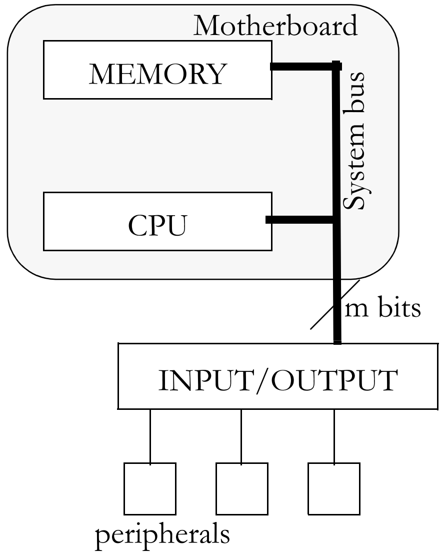
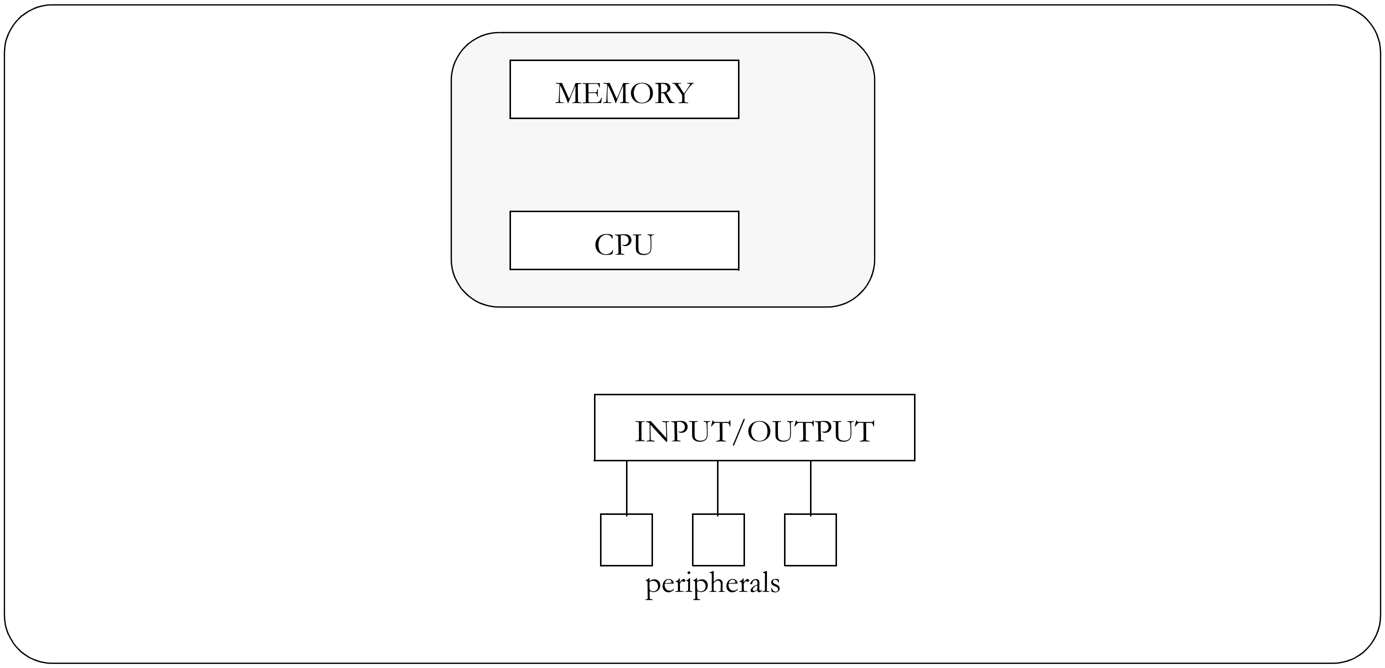


FIGURE 1. Memory-Mapped Bus Topology

(b) [2] Given the same elements as in Figure 1 draw the appropriate connection to show what a "Port I/O" or "Isolated I/O" bus organization looks like. Label every element precisely



Question 5. [8] Consider a system that has a byte-addressable memory organized in 32-bit words according to the big-endian scheme. A program reads two integers into an array and stores them in successive locations, starting at location at address 0x00001000. The two integers are: 1,025 and 287,454,020.

(a) [4] Show the contents of the two memory words at locations 0x00001000 and 0x00001004 after the two integers have been stored.

content in hex								
byte address (hex)	00001000	00001001	00001002	00001003	00001004	00001005	00001006	00001007

WORD @ 0x00001000 WORD @ 0x00001004

(b) [4] Repeat the question above for the little-endian scheme.

content in hex								
byte address (hex)	00001000	00001001	00001002	00001003	00001004	00001005	00001006	00001007

WORD @ 0x00001000 WORD @ 0x00001004

Question 6. [4] (a) Write the *pseudo code steps* to evaluate the expression $K \times B + C \times D$ in a single accumulator processor architecture. This implies that your processor has only one register, the *Accumulator* (call it “A”). Assume that the processor has only the Load, Store, Multiply, and Add instructions, where one of the operands can be accessed from memory (see below). Leave the final result in Accumulator A. Assume you have the four variables K, B, C, D declared as memory storage and you are allowed to declare others if necessary. Use *only* the 4 instructions listed below with their same semantics.

The pseudo instructions available are:

```
Load     A,Loc    ; Accumulator A ← memory content at Loc
Store    A,Loc    ; memory at Loc ← value in Acc. A
Mul      A,Loc    ; A ← A * (memory content at Loc)
Add      A,Loc    ; A ← A + (memory content at Loc)
```

(b) Rewrite the pseudo code for $K \times B + C \times D$ from (a) assuming a stack architecture. In such an architecture, a stack, call it “S”, is available as your *only* accumulator, together with the instructions:

- Push *Op* – to copy operand “*Op*” onto the Stack;
- Pop *Op* – to remove a value from the Stack and store it in operand “*Op*”;
- Add – to pop the top two operands from the stack, add them, and push the result on the stack;
- Mult – to pop the top two operands from the stack, multiply them, and push the result on the stack.

Question 7. [7] You are analyzing with your manager a new microprocessor which is byte addressable with a 20-bit address bus and a 16-bit data bus, and one word contains 2 bytes. Your manager wants to purchase RAM chips each containing 512K words (0.5 Megabytes words), where each word is 2 bytes, for a total of 2^{19} words.

(a) [1] What is the maximum number of these RAM chips you could place in your system and be able to address?	

(b) [3] Having decided how much RAM could be bought, your manager realizes that the whole address space has been used up for it, leaving no room for peripherals to make the system useful after all. The manager tells you to buy half as much RAM as what was calculated above and use the rest of the address space to install peripherals. They would like to have at least 32 peripherals each of which is byte addressable and requires an address space of 64KB. Your manager asks you to check that it is all feasible..

Is it? YES NO (circle one)

Show a complete logical explanation of the reasoning for your answer, as you might do at work

(c) If the answer above stated that 32 peripherals is too many, state how many you could handle in your system, without reducing further the RAM.

(d) If the answer above stated that 32 peripherals fit exactly into the address space, then you are finished.

(e) If the answer above stated that you could fit 32 peripherals and you had spare address space, state how much more RAM you could have in your system (that is, how many more chips you could buy.

Choose one of the three cases (c), (d) or (e) and give an answer with explanation. [3]

(Note for curiosity: the specifications above might refer to an old Intel 8086 processor).

Question 8. [8] In the lectures you saw some lines of ARM code with their semantics and the effect they had after execution on some registers and on a portion of memory. Repeat the same exercise with the four lines of code below, given the initial status of the registers and of memory as shown Show the effect after the execution of each instruction.

The instructions to be executed are:

```
[1]  LOAD R2,[R2]   @ R2 = MEM[R2], i.e. copy into R2 the content of the
                    @ location in memory whose address is in R2
[2]  LOAD R4,[R4]   @ R4 = MEM[R4], i.e. copy into R4 the content of the
                    @ location in memory whose address is in R4
[3]  ADD R4,R4,R2   @ R4 = R4 + R2
[4]  STR R4,[R3]    @ MEM[R3] = R4, i.e. copy the value in R4 into the
                    @ location in memory whose address is in R3
```

.

Registers	**Content**	**After [1]**	**After [2]**	**After [3]**	**After [4]**
R2	000090C4				
R3	000090D0				
R4	000090C8				

Memory					
Before		**After [1]**	**After [2]**	**After [3]**	**After [4]**
Address	**Content**	**Content**	**Content**	**Content**	**Content**
C00090C0	028F101C				
C00090C4	06112000				
C00090C8	028F3018				
C00090CC	06134000				
C00090D0	00825004				
C00090D4	028F6010				
C00090D8	02165000				
C00090DC	00000005				
C00090E0	00000006				
C00090E4	00000000				

Question 9. [7] Consider a 2MB memory which is word-addressable, that is, it is assumed that the smallest location in memory which can be addressed and transferred is exactly one word – here we define one word as being 4 bytes (reminder: Mega = 2^{20}). It implies that one needs an address for every word (but it is not necessary to have an address for every byte).

(a) How many bytes does the memory contain?	
(b) How many bits does the memory contain?	
(c) How many addressable locations are there?	
(d) How many address lines would you need in the address bus?	

Suppose instead that it were byte-addressable, that is, it is assumed that the smallest location in memory which can be addressed and transferred is exactly one byte. It implies that one needs an address for every byte..

(e) How many bytes does the memory contain?	
(f) How many addressable locations are there?	
(g) How many address lines would you need?	

Question 10. [1] Consider having 24 address lines to address memory, a byte-addressable system, and having RAM chips each containing 256K words, where each word is 4 bytes, for a total of 2^{18}words (equivalent to 2^{20}bytes).

(a) What is the maximum number of these RAM chips you could place in your system and be able to address?	

Question 11. [5] Find two consumer products, currently on the market, which use some ARM processor. For each, give a short description of the product, the type of ARM processor being used, and any other information of its architecture you think is relevant or interesting. The top 5 most intriguing examples will be posted. Use no more than 1 page for this answer and state your sources. Imagine having to give this answer to one of your managers in a group meeting, in a new job, when it is important to make a good impression. Neatness, clarity, legibility and good grammar are essential. (Add one sheet of paper for your answer if needed and make sure to label it with name and student number).