

COEN 311 Section W (Computer Organization and Software)

Assignment 3

Due Friday October 27, 2023

Note: Please submit only one PDF file through Moodle.

Question 1:

- a) Hand assemble the following SIMARM program. Give your answer in **Hexadecimal** and compute all memory addresses

Line #	Mem-Address	Machine Instruction	Assembly Instruction		Size of each instruction (in bytes)	Number of Memory Accesses for each instruction
			.thumb			
			.globl _start			
1			_start:	ADD R0, R0, #1		
2				LDR R3, =n		
3				LDR R1, [R3]		
4				ADD R2, R2, #1		
5			loop:	CMP R2, R1		
6				BGT endloop		
7				MUL R0, R2		
8				ADD R2, R2, #1		
9				B loop		
10			endloop	LDR R3, =result		
11				STR R0, [R3]		
12				B .		
13			n:	.word 5		
14			result:	.space 4		

- b) Indicate what will be the contents of location “result” after the execution of the above program. Assume R0 and R2 are initialized to “0” at the beginning of the program.
- c) Also provide the size of each instruction to obtain the total size of the program, and number of memory accesses required by the program.

Question 2:

Write a SIMARM assembly program to implement “**mults**” instruction based on *successive addition* that multiplies two positive (non-zero) 32-bit values “x” and “y” stored in memory locations X and Y respectively (Y being the result to be stored back in memory).

Use the following *successive addition* algorithm to multiply x and y (y is destination)

```
y0 = y;
i:=1;
while i < x
do {y:= y0 + y;
    i:= i + 1
}
```

Question 3:

Given the following array of 9 integers stored in memory:

[67, 98, 34, 9, 51, 15, 73, 22, 6]

Write a SIMARM program that

- (i) Reads these values as 32-bit integers.
- (ii) Determines the largest of the numbers.
- (iii) Stores the largest number at a memory location L.

Use the following code as possible algorithm

```
counter = 0
largest = first number
while ( ++counter < 10 )
{
    read the next number
    if (number > largest)
        largest = number;
}
```

Question 4)

In each of the following instructions identify the addressing mode used for the source operand:

- a) MOV R0, R1
- b) MOV R2, #1
- c) MOV R3, [R1]
- d) MOV R2, [R1, #8]
- e) MOV R1, [R2, R0, #8]
- f) MOV R1, [\$1234]