NAME: ______________________________ **STUDENT :**________________________

Lecture Section:_____________ **Tutorial Section:**_______________

Question 1

a) Assume a three-stage pipeline (fetch, decode & execute). Draw a timing diagram to show how many units are needed for four instructions. **[2 marks]**

	Time 1	2	3	4	5	6
Instruction 1	Fetch	Decode	Execute			
Instruction 2		Fetch	Decode	Execute		
Instruction 3			Fetch	Decode	Execute	
Instruction 4				Fetch	Decode	Execute

b) Write ARM instructions to do the following**:** **[2 marks]**

i) Load the number FFFFFFFF_H in R1 and FFFFFFFE_H in R3

ii) Add with update of condition flags, FFFFFFFF_H and FFFFFFFE_H using R1 and R3,

iii) Store the result in memory address 6000_H

Write down the contents in the condition flags in the Program Status Register (CPSR) after the execution of all the three instructions. **[2 marks]**

```
 MOV R1,
#0xFFFF FFFF

 MOV R3,
#0xFFFF FFFE

 ADDs R5, R1,
R3

 MOV R4,
#0x6000
```

STR R5, [R4]

(0.4 mark x 5 = 2 marks)

N (Negative): 1 , Z (Zero): 0 , C (Carry): 1, V (Overflow): 0

(0.5 mark x 4 = 2 marks)

c. Assume that a hypothetical computer contains a single data register, called an accumulator

(A). Both instructions and data are 16 bits long. Memory is byte addressable. Instruction format provides 4 bits for the opcode and remaining bits for the memory direct address. Integer data uses 2's complement method. Partial list of assembly instructions includes the following:

Opcode	**Operand**	**Machine Code**	**Meaning**
LDA	Direct Memory address	**0010_2**	Load the data from memory to accumulator
SUB	Direct Memory address	**0011_2**	Subtract the memory contents from accumulator and store the result in accumulator.
STA	Direct Memory address	**0111_2**	Store the data from accumulator to memory

This hypothetical computer uses program counter to keep track of program execution and instruction register for holding the instructions temporarily before decoding.

Answer the following:

i) Identify the instruction format. [1 mark]

ii) Identify the range of direct memory addresses. [1 mark]

iii) By assume that program is stored starting from memory address 002_H and the instructions are executed in sequence. Write the program fragment to subtract the contents of memory word at address $27A_H$ from the contents of memory word at address 279_H and store the result in memory location, $27B_H$. [2 marks]

Solution:

(i) Instruction Format:

0 3	4 15
Opcode (4 bits)	Operand (memory direct address) (12 bits)

(0.5 * 2 = 1 mark)

(ii) Range of Direct Memory Addresses:

2^{12} = 4096 memory addresses.

Range: 0 to 4095 (Decimal) or 0000 0000 0000 to 1111 1111 1111 (binary)

(0.5 * 2 = 1 mark)

Memory address	Mnemonics
002_H	LDA 279 [0.5 *mark]*
003_H	SUB 27A [*1 mark]*
004_H	STA 27B [0.5 *mark]*

(0.667 * 3= 2 marks)

Question 2.

a) The **basic functions of a processor** (also called a CPU – Central Processing Unit) can be summarized into five main categories. List and describe any **THREE** of the five main categories. [3 marks]

i) Fetch instruction The processor reads an instruction from memory (register, cache, main memory)

ii) Interpret instruction -- The instruction is decoded to determine what action is required

iii) Fetch data-- The execution of an instruction may require reading data from memory or an I/O module

iv) Process data -- The execution of an instruction may require performing some arithmetic or logical operation on data

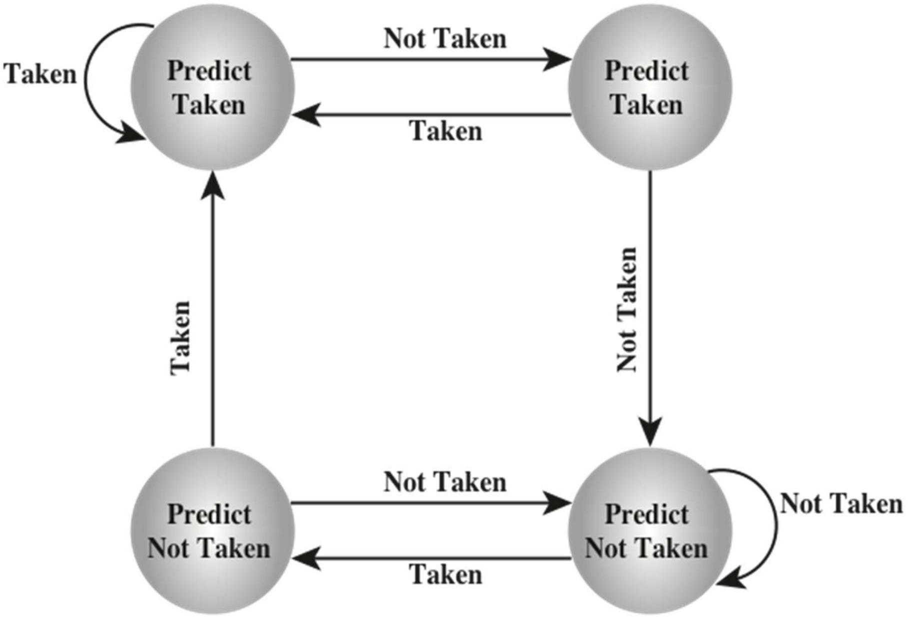
v) Write data -- The results of an execution may require writing data to memory or an I/O module

(any three correct answers, 3 x 1 = 3 marks)

b. The Branch Prediction State Diagram for pipeline branch hazards is shown below. Based on the history of instruction branches given, state whether the next branch prediction will be taken or not:

i. Taken, Not Taken, Not Taken, Taken [1.5 marks]

ii. Not Taken, Not Taken , Not Taken, Taken, Not Taken [1.5 marks]



i) Not Taken [1.5 marks]
ii) Not Taken [1.5 marks]

c. Write a program to evaluate the arithmetic expression T = [(B-C) / D)] * E, using one address instructions and two address instructions. The instructions available for use are as follows: [2 x 2 = 4 marks]

One address	**Two address**
LOAD X	MOVE X ,Y
STORE X	ADD X, Y
ADD X	SUB X, Y
SUB X	MUL X ,Y
MUL X	DIV X, Y
DIV X	

One Instructions	**Two Instructions**
LOAD B SUB C DIV D MUL E STORE T [2 marks]	MOVE T, B SUB T, C DIV T, D MUL T, E **OR** MOVE R0, B SUB R0, C DIV R0, D MUL R0, E MOVE T, R0 WHERE R0 IS A TEMPORARY REGISTER [2 marks]

The End