The Open University of Sri Lanka
Faculty of Engineering Technology
Department of Electrical and Computer Engineering

Study Programme	**: Bachelor of Software Engineering Honours**
Name of the Examination	**: Final Examination**
Course Code and Title	**: EEX 3373 Communication and Computer Technology**
Academic Year	**: 2023/24**
Date	**: 25 th January 2025**
Time	**: 0930 - 1230hrs**
Duration	**: 3 hours**

General Instructions

1. Answer all questions.
2. This is a closed book type examination.
3. You can use non-programmable calculators.
4. Write all relevant intermediate steps when answering.
5. Answers should be in clear handwriting.
6. Do not use red color pens for answering questions.

Question 1:

I.

a) Draw a diagram of the accumulator-based processor architecture and explain the function of each component. **[05 Marks]**

b) Explain the step-by-step process of the fetch-execute cycle when executing a direct addressing mode instruction and an indirect addressing mode instruction. **[04 Marks]**

II. Assume you must perform an AND operation and store the result in memory. Referring to the diagram you drew in Question 1 - I. a), write down the steps involved in the fetch and execution process for this operation in an accumulator-based microprocessor. **[06 Marks]**

III. Draw the general instruction format and describe its components. **[04 Marks]**

IV. Consider an array of five positive integers: Arr[] = {A, B, C, D, E}. You have to calculate the average of the five elements in the array Arr[]. The first number starts from Arr[0]. Consider the following tasks,

a) Store each element of Arr[] in memory and choose appropriate memory locations for the array elements.
b) Calculate the average, avg_Arr of the Arr[] and store it in the 0x80 memory location.
c) If Arr[0] = avg_Arr, Store integer 1 in 0x90 memory location.
d) If Arr[0] < avg_Arr, Store integer 2 in 0x91 memory location.
e) If Arr[0] > avg_Arr, Store integer 3 in 0x92 memory location.

Draw a single flowchart to show the algorithm of the assembly program (accumulator-based computer architecture) to perform the above given tasks in **Question 1 - IV**. Clearly state if you made any assumptions. Important: Use the standard symbols / shapes to draw the flowchart.

[13 Marks]

V. Write an assembly program for the algorithm described in **Question 1 - IV**, considering the array Arr[] = {20, 6, 27, 34, 13}. Use the Instruction Set Architecture (ISA) provided in **Appendix A.** **[08 Marks]**

Question 2:

I. Explain the Von Neumann architecture with the help of a diagram. **[03Marks]**

II. Convert the following binary numbers. Show the intermediate calculation steps.
a) Convert 101101.101_2 to its decimal equivalent.
b) Convert 265.75_{10} to its binary equivalent.
c) Convert $1101111101101011100110 10_2$ to its hexadecimal equivalent.
d) Convert 11110110101110001_2 to its octal equivalent.

[06 Marks]

III. Perform the following operations.
a) 1010.101 * 101
b) Use the 2's complement method to perform the following calculation considering **8-bit** binary representation. 31 - 86 = ?
c) Convert the numbers A = 74 and B = 38 into 8421 BCD format. Refer to **Table 01** provided in **Appendix A.**

[07 Marks]

IV. Convert the number -187.75_{10} into IEEE 754 Single precision format. **[04 Marks]**

Question 3:

I. Modulation and demodulation are very important in communication systems.

a) Briefly explain the terms modulation and demodulation. **[02 marks]**

b) State two reasons why modulation is necessary in communication systems. **[02 marks]**

c) Define amplitude modulation (AM). **[02 marks]**

d) Discuss how variations in the modulation index affect the AM signal. **[02 marks]**

e) Calculate the modulation index when the maximum peak-to-peak voltage of an AM wave is 16 mV, and the minimum peak-to-peak voltage is 4 mV. **[02 marks]**

II. The frequency modulated signal is given as,

$$V_m(t) = 14\cos\left(8 \times 10^8 t + 10\sin(1000t)\cos\left(1000t\right)\right)$$

Find the following

a) Modulating Frequency

b) Carrier frequency

c) Modulation Index

d) Frequency deviation

10 marks]

Question 4:

I. The OSI model standardizes network functions across seven layers, enabling seamless communication between systems.

Explain the seven layers of the OSI reference model, describing the main function of each layer.

[04 marks]

II. A Research Lab plans to connect its network with its five affiliated research centers located in different regions. The number of hosts required in each center is as follows: Center A (200), Center B (150), Center C (300), Center D (75), and Center E (25). The main IP block assigned for these connections is 192.168.100.0/22.

a) Find the network address of the main IP block. **[04 marks]**

b) Determine the maximum number of usable hosts that can be accommodated within the main IP block without subnetting. **[02 marks]**

c) Perform subnetting for the given network to meet the requirements of each center. Calculate the network address, subnet mask, starting host address, ending host address, and broadcast address for each subnet. State your assumptions used in the calculation.

[10marks]

Appendix A

DECIMAL	8421 BCD	GRAY	BINARY
0	0000	0000	0000
1	0001	0001	0001
2	0010	0011	0010
3	0011	0010	0011
4	0100	0110	0100
5	0101	0111	0101
6	0110	0101	0110
7	0111	0100	0111
8	1000	1100	1000
9	1001	1101	1001
10	0001 0000	1111	1010
11	0001 0001	1110	1011
12	0001 0010	1010	1100
13	0001 0011	1011	1101
14	0001 0100	1001	1110
15	0001 0101	1000	1111

Table 01- Special Binary codes

Accumulator Architecture

Mnemonic	Syntax	Operation	Description	Add Mode	Flags affected
Arithmetic instructions					
ADD	ADD AM x	Addition	Acc ← Acc + op	#,&,default	ZF, OF, SF, PF =1
SUB	SUB AM x	Subtraction	Acc ← Acc - op	#,&,default	ZF, OF, SF, CF, PF =1
MUL	MUL AM x	Signed multiplication	Acc (16 bit) ← Acc(8 LSBs) * op(8 LSBs)	#,&,default	ZF, SF, PF =1. OF, CF=0
DIV	DIV AM x	Unsigned division	Acc ← Acc / op2	#,&,default	ZF, SF, PF =1. OF, CF =0
INC	INC	Increment by 1	Acc ← Acc + 1	#,&,default	OF, SF, ZF, PF=1
Logical instructions					
AND	AND AM x	Bit-wise And	Acc ← Acc AND op	#,&,default	OF, CF=0; SF, ZF, PF =1
OR	OR AM x	Bit-wise OR	Acc ← Acc OR op	#,&,default	OF, CF =0; SF, ZF, PF =1
XOR	XOR AM x	Bit-wise XOR	Acc ← Acc XOR op	#,&,default	OF, CF=0; SF, ZF, PF =1
SHL	SHL	Shift left by 1-bit	CF ← Acc (MSB), op← Acc (14 down to 0) & 0	Implied (A)	CF= bit shifted by Acc. SF, ZF, PF=1.
SHR	SHR	Shift right by 1-bit	CF ← Acc (LSB), Acc ← 0 & Acc (15 down to 1)	Implied (A)	CF= bit shifted by Acc. SF, ZF, PF=1.
ROL	ROL	rotate left by 1-bit	CF ← Acc (MSB), op← Acc (14 down to 0) & CF	Implied (A)	CF= bit shifted by Acc. SF, ZF, PF=1.
ROR	ROR	rotate right by 1-bit	CF ← Acc (LSB), Acc ← CF & Acc (15 down to 1)	Implied (A)	CF= bit shifted by Acc. SF, ZF, PF=1.
NOT	NOT	One's compliment negation	Acc ← NOT Acc	Implied (A)	none
Control Transfer instructions					
Conditional Branches					
JC	JC d	Jump if carry	If CF =1 then PC ← PC + Operand	PC relative	none
JOF	JOF d	Jump if over-flow	If OF =1 then PC ← PC + Operand	PC relative	none
JS	JS d	Jump if Sign	If SF =1 then PC ← PC + Operand	PC relative	none
JP	JP d	Jump if parity	If PF =1 then PC ← PC + Operand	PC relative	none
JZ	JZ d	Jump if result is zero	If ZF =1 then PC ← PC + Operand	PC relative	none
Unconditional branch					
JUMP	JUMP d	Jump	PC ← PC + Operand	PC relative	none
Loops					
LOOZ	LOOZ d	Loop until zero	Count ← Count – 1, IF Count =0; Loop termination ELSE; PC ← PC + operand	PC relative	none
Calls and Returns					
CALL	CALL d	Procedure call	implied return address ← PC, PC ← immediate address	*	none
RETURN	RETURN	Return from procedure	PC ← Contents of implied return address	**	none
Miscellaneous instructions					
NOP	NOP	No operation		none	none
Data Movement instructions					
LOADacc	LOADacc AM d	Copy the operand to the accumulator	Immediate: Acc ← op; Direct: Acc ← memory (op)	#, default	none
STOREacc	STOREacc AM d	Copies the accumulator to the memory address	Direct: Memory (op) ← Acc Indirect: Memory {memory (op)} ← Acc	default, &	none

*Immediate operand is used as the jumping location.
**Contents of the implied return address is used as the jumping location
x- Memory Address or immediate value, d- Displacement

Addressing Modes -AM

Immediate	#
Direct	default (blank)
Indirect	&

Table 02- Instruction Set Architecture

-- End --