

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



0516

Study Programme : Bachelor of Software Engineering Honours
Name of the Examination : Final Examination
Course Code and Title : EEX3373 – Communication and Computer Technology
Academic Year : 2024/25
Date : 17th February 2026
Time : 0930-1230h
Duration : 3 hours

General instructions

1. Read all instructions carefully before answering the questions.
2. This question paper consists of **Four (4)** questions in **Six (6)** pages.
3. Answer **all** questions.
4. You can use non-programmable calculators.
5. Write all relevant intermediate steps when answering
6. Answer for each question should commence from a new page.
7. This is a Closed Book Test (CBT).

Question 01

[40 Marks]

- a. Describe the fundamental components of a computer system. Using a block diagram, illustrate how these components are interconnected and describe how data is communicated among them during processing. [3 marks]
- b. Describe, step-by-step, how the fetch-execute cycle operates in a typical accumulator-based microprocessor during the execution of an instruction stored in main memory. [5 marks]
- c. If A and B are inputs, draw the truth table of the operation "A XOR B" and explain how it executes in an accumulator-based microprocessor. [6 marks]
- d. Describe three different addressing modes in which an instruction can refer to its operand in an accumulator-based processor and provide one example instruction for each addressing mode. [6 marks]
- e. In modern educational institutions, automated lighting systems are increasingly used to improve energy efficiency and create comfortable learning environments. These systems monitor ambient light levels in classrooms and automatically control artificial lighting without manual intervention. [12 marks]

The main objective is to ensure adequate illumination while minimizing electricity consumption and promoting sustainable energy use.

A microprocessor can be used as the core controller of such a system to process light sensor data, make logical decisions based on predefined thresholds, and control outputs such as classroom lights. In this context, a microprocessor-based lighting controller reads light intensity values from three different classroom zones, processes the information, and determines whether additional lighting is required.

You are required to design a microprocessor-based application for an automated classroom lighting system. The system must monitor light intensity levels in three zones using sensor values. Let the light sensor readings be denoted as L1, L2, and L3, and assume that these values are stored in memory locations m1, m2, and m3 respectively. These values, ranging from 0 to 100, represent the percentage of natural light detected in each zone.

To determine the overall lighting condition, the system must calculate the average light level, avg_L using the formula:

$$\text{Average Light Level}(\text{avg_L}) = \frac{L1 + L2 + L3}{3}$$

The calculated average should be stored in memory location **m4**. Once the average is computed, the system should perform the following conditional operations:

- If avg_L = **60**, increment the value in memory location **m5**, indicating a balanced lighting condition.
- If avg_L < **60**, increment the value in memory location **m6**, indicating that artificial lighting should be switched ON.
- If avg_L > **60**, increment the value in memory location **m7**, indicating that artificial lighting is not required.

For simulation purposes, the sensor values are assigned as,

$$L1 = 58, L2 = 62, \text{ and } L3 = 75.$$

Draw a single flowchart to show the algorithm of the assembly program based on accumulator-based computer architecture to perform the above given tasks in **Question 1- e**. The allocation of memory locations for m1 to m7 may be determined based on your proposed system design. Clearly state if you made any assumptions. **Important:** Use the standard symbols and shapes to draw the flowchart.

- f. Write an assembly program for the algorithm described in **Question 1 - e**. [8 marks]
Use the Instruction Set Architecture (ISA) provided in **Appendix A**.

Question 02

- a. Convert the following numbers. **Show all intermediate calculation steps clearly.** [6 marks]
- (i) Convert 110011.011_2 into its **decimal equivalent**.
- (ii) Convert $101101111010011011100101_2$ into its **hexadecimal equivalent**.
- (iii) Convert 11101101011010110_2 into its **octal equivalent**.
- b. Perform the following operations. [4 marks]
- (i) $101010101 \div 101$
- (ii) Find the 8-bit binary, 2's complement form of -85.
- c. Convert the number -132.625_{10} into IEEE 754 Single precision format. [6 marks]
- d. Convert the decimal numbers $A = 87$ and $B = 43$ into 8421 BCD format. Refer to **Table 01** provided below. [4 marks]

Table 01- Special Binary codes

DECIMAL	8421 BCD
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111
8	1000
9	1001
10	0001 0000
11	0001 0001
12	0001 0010
13	0001 0011
14	0001 0100
15	0001 0101

Question 03

- a. Define analog modulation and list the main types of analog modulation types. [2 marks]
- b. With the aid of mathematical equations, explain how the modulation index is defined for [4 marks]

- (i) Amplitude Modulation (AM) and
(ii) Frequency Modulation (FM).

Clearly identify all symbols used in the equations.

- c. An amplitude-modulated signal is applied to a linear envelope detector. The measured envelope voltages are,

$$V_{max} = 14 V \text{ and } V_{min} = 6 V$$

- (i) Calculate the modulation index of the AM signal. [2 marks]
- (ii) Determine whether the signal is under-modulated, perfectly modulated, or over-modulated, and justify your answer. [2 marks]
- (iii) If the carrier amplitude is kept constant and the modulation index is increased beyond one, explain why the envelope detector output becomes distorted, even though the transmitted power increases. [4 marks]
- d. A frequency-modulated signal is given by:

$$v_{FM}(t) = 20 \sin [2\pi(80 \times 10^6)t + \beta \sin (2\pi \times 4 \times 10^3 t)]$$

The maximum frequency deviation of the carrier is known to be 24 kHz.

- (i) Calculate the FM modulation index β . [2 marks]
- (ii) Determine the modulating frequency. [2 marks]
- (iii) Estimate the bandwidth of the FM signal. [2 marks]

Question 04**0516**
[20 marks]

- a. Explain the concept of the World Wide Web (WWW), including the role of hypertext and how a web browser is used to access information on the Web. [4 marks]
- b. A **Large Software Services Company** is building a new enterprise network for its main office that hosts application development, cloud services, and client systems. For security, performance, and management reasons, the company must separate its network into five functional segments using subnetting. The number of hosts required in each segment is as follows,

Name	No. Host
Segment A	310
Segment B	160
Segment C	240
Segment D	90
Segment E	40

The main IP block assigned for these connections is **192.168.60.0/21**.

- (i) Find the network address of the main IP block. [4 marks]
- (ii) Determine the maximum number of usable hosts that can be accommodated within the main IP block without subnetting. [2 marks]
- (iii) Perform subnetting for the given network to meet the requirements of each segment. Determine the network address, subnet mask, starting host address, ending host address, and broadcast address for each subnet. State all your assumptions used. [10marks]

Accumulator Architecture

Mnemonic	Syntax	Operation	Description	Add Mode	Flags affected
Arithmetic instructions					
ADD	AM x	Addition	Acc ← Acc + op	#, & default	ZF, OF, SF, PF = 1
SUB	AM x	Subtraction	Acc ← Acc - op	#, & default	ZF, OF, SF, PF = 1
MUL	AM x	Signed multiplication	Acc (16 bit) ← Acc(8 LSBs) * op(8 LSBs)	#, & default	ZF, SF, PF = 1, OF, CF = 0
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	AM x	Bit-wise And	Acc ← Acc AND op	#, & default	OF, CF = 0, SF, ZF, PF = 1
OR	AM x	Bit-wise OR	Acc ← Acc OR op	#, & default	OF, CF = 0, SF, ZF, PF = 1
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 complement 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					
LOOP	LOOP 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
Indirect

default (blank)
&