Model Question (IT 233: Digital Logic)

Group-A

Brief Answer Questions (10x1 = 10)

Attempt all the questions.

1) What do you mean by digital computer?

⇒ A digital computer is an electronic device that processes information in binary form, using digital logic circuits (such as AND, OR, NOT gates) to perform operations like computation, data storage, and control. These computers work with discrete signals (0s and 1s) rather than continuous signals.

2) Classify the binary number.

⇒ A binary number can be classified in following ways:

- Unsigned Binary Number
- Signed Binary Number
- Binary Coded Decimal (BCD)
- Gray code

3) What do you mean by 2's complement?

⇒ Two's complement is a binary number system used to represent both positive and negative integers, where negative values are obtained by inverting the bits of the positive number and adding 1 to the result.

4) NAND gate is universal gate. Justify.

⇒ NAND gate is considered a universal gate because it can be used to implement any other basic logic gate (AND, OR, NOT) and, therefore, any digital circuit. By combining multiple NAND gates, we can construct all other logic gates and perform complex logical operations.

5) What is Boolean algebra?

⇒ Boolean algebra is a mathematical system for designing and analyzing digital circuits. It has binary variables, boolean operators, AND (.), OR (+) and other symbols.

6) What do you mean by SOP?

⇒ Sum of Products (SOP) is a canonical form where a Boolean expression is written as a sum (OR) of product terms (AND). Each product terms is a conjunction (AND) of literals.

7) Define PLA.

⇒ Programmable Logic Array (PLA) is a combinational circuit and it is similar to ROM in a concept. However, it doesn't provide full decoding as ROM. A situation where don't care condition is excessive, it is more economical and convinent to use PLA as compared to ROM. In PLA, programmable AND gates and OR gates are used.

8) What do you mean by flip flop?

⇒ Latch with clock enabled is called flip flop. It is a bi-stable memory device that store 1 bit of binary data. It is a basic building block of any sequential circuit. It's types are: SR flip flop, D-FlipFlop, J-K flipflop, T-FlipFlop.

9) Define registers.

⇒ Registers are small, fast storage locations within a CPU used to hold data, instructions, or addresses temporarily during processing. They are quick access to data required for operations and control tasks within the processor.

10) What do you mean by ripple counter?

⇒ A ripple counter is a type of binary counter where each flip-flop's output serves as the clock input for the next flip-flop in the sequence, causing a "ripple" effect through the stages.

Group-B

Short Answers Questions (5x3 = 15)

Attempt any five questions.

11) Describe the octal and hexadecimal numbers.

⇒

- **Octal Numbers**

  Octal Numbers are base-8 numbers using digits 0 through 7. Each octal digit represents three binary bits. Each digit's place value is a power of 8 (eg: $8^0, 8^1, 8^2$, etc)

- **Hexadecimal Numbers**

  Hexadecimal Numbers are base-16 numbers using digits 0 through 9 and letters A through F (where A=10, B=11, C=12, D=13, E=14, F=15). Each hexadecimal digit represents four binary bits. Each digit's place value is a power of 16 (eg: $16^0, 16^1, 16^2$, etc)

12) Discuss the basic theorems and properties of Boolean algebra.

→

· **Theorems**

Commutative Law

$A + B = B + A$ or $A \cdot B = B \cdot A$

Associative Law

$A + (B + C) = (A + B) + C$ or $(A \cdot B) \cdot C = A \cdot (B \cdot C)$

Distributive Law

$A + (B \cdot C) = (A + B) \cdot (A + C)$

De-Morgan's Law

$\overline{A + B} = \overline{A} \cdot \overline{B}$ or $\overline{A \cdot B} = \overline{A} + \overline{B}$

· **Properties**

1) $A + A = A$
2) $A \cdot A = A$
3) $A + 1 = 1$
4) $A + 0 = A$
5) $A + \overline{A} = 1$
6) $A \cdot \overline{A} = 0$
7) $A \cdot 1 = A$
8) $A \cdot 0 = 0$
9) $\overline{\overline{A}} = A$

13) **Explain the don't care condition with example.**

⇒ Don't care condition is a situation that arises in the K-Map. In K-Map each min term is represented with 1 and max term with 0. A situation where we cannot differentiate 1 and 0. Such, situation is called dont care condition.

For example!

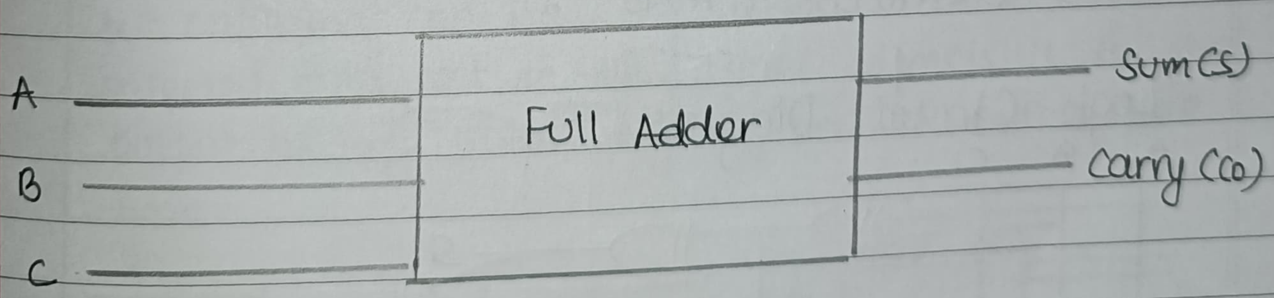
*) Simplify a Boolean Function $F(w,x,y,z) = \Sigma(0,1,2,3,4,8,9,10,13)$ and don't care condition $D(w,x,y,z) = \Sigma(5,7,11)$ by using K-Map.

	$\bar{y}\bar{z}$	$\bar{y}z$	yz	$y\bar{z}$
$\bar{w}\bar{x}$	1	1	1	1
$\bar{w}x$	1	X	X	
wx		1		
$w\bar{x}$	1	1	X	1

$F = \bar{x} + \bar{w}.\bar{y} + \bar{y}z$

14) **Design full adder with truth table.**

⇒



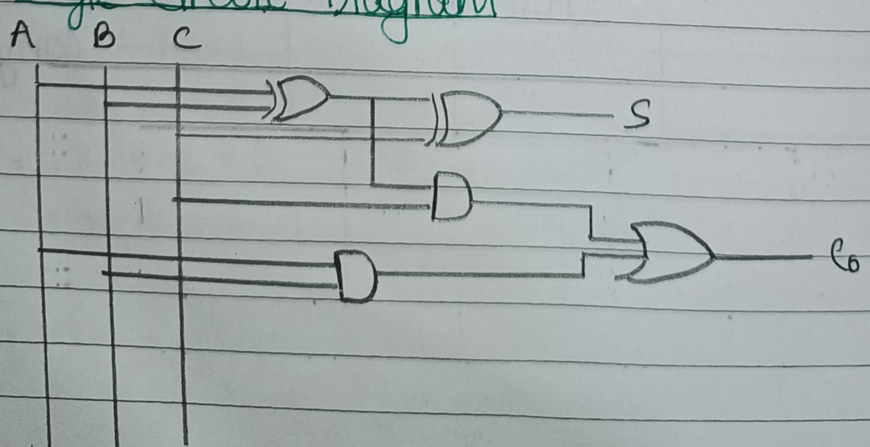
Truth Table

Input			Output	
A	B	C	S	C_o
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

$$S = \bar{A}\bar{B}C + \bar{A}B\bar{C} + A\bar{B}\bar{C} + ABC$$
$$= \bar{A}(\bar{B}C + B\bar{C}) + A(\bar{B}C + BC)$$
$$= \bar{A}(B \oplus C) + A(B \odot C)$$
$$= \bar{A}(B \oplus C) + A(\overline{B \oplus C})$$
$$= A \oplus B \oplus C$$

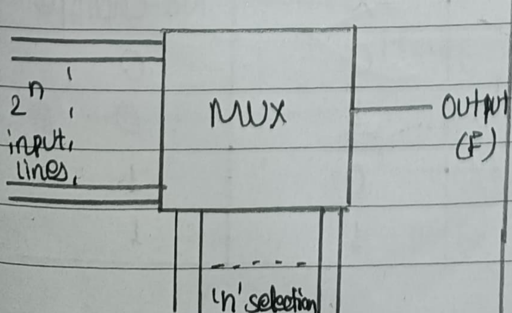
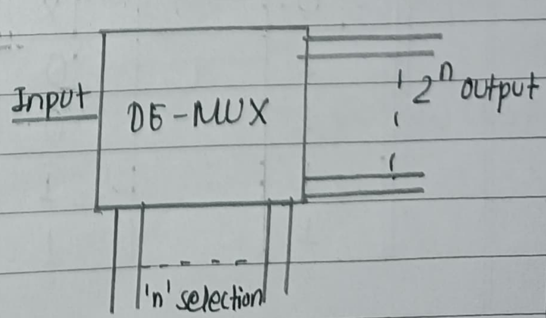
$$C_o = \bar{A}BC + A\bar{B}C + AB\bar{C} + ABC$$
$$= C(\bar{A}B + A\bar{B}) + AB(\bar{C} + C)$$
$$= C(A \oplus B) + A \cdot B$$

Logic Circuit Diagram



15) Differentiate between multiplexers and demultiplexers.

⇒ The differences between multiplexers and demultiplexers are:

Multiplexers	Demultiplexers
• It is a device that combine multiple channel into single composite channel.	• It is a device that takes single input line and provide multiple output lines, based on select lines.
• A MUX has '2^n' input line, 'n' selection line and single output line.	• A DE-MUX has single input line, '2^n' output line and 'n' selection line.
• It is also known as Data selector.	• It is also known as Data Distributor.
• It works on many to one operational principle.	• It works on one to many operational principle.
• It is used at, the transmitter end.	• It is used at the receiver end.
• In MUX, multiple data sources can be connected, but only one can be selected and sent to the output at any given time.	• In DE-MUX, it takes single input signal and based on the select lines, directs it to the appropriate output line.
 Block diagram of multiplexer	 Block diagram of demultiplexer

16) Explain D-flip flop.

⇒ D stands for Data. D-flip flop is constructed by cross coupling of NAND or NOR gates. When clock is equal to zero, it doesn't function its operation and remain unchange state. When clock is equal to one, it performs its function.

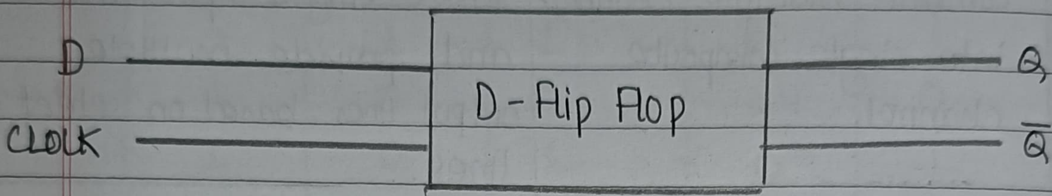
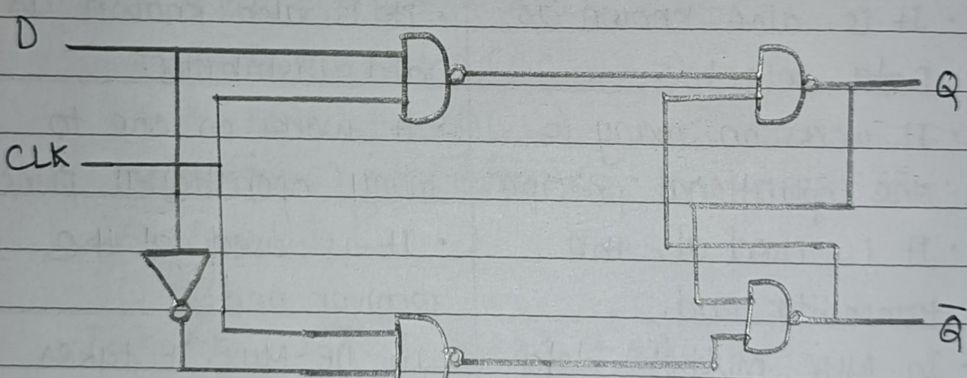


Fig: Block Diagram

Logic Circuit Diagram



Characteristics Table

Clock	D	Qp	Qn
0	X	X	No change
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	1

Characteristics Equation

Q_n \ Q_P	$\bar{Q}_P$	Q_P
$\bar{D}$		
D	1	1

$\therefore Q_n = D$

Excitation Table

Present state	Next state	F.F input
Q_P	Q_n	D
0	0	0
0	1	1
1	0	0
1	1	1

Group-C

Long Answer Questions (3×5 = 15)

Attempt any three questions

17) Design a half subtractors logic circuit using only NOR gate.

⇒

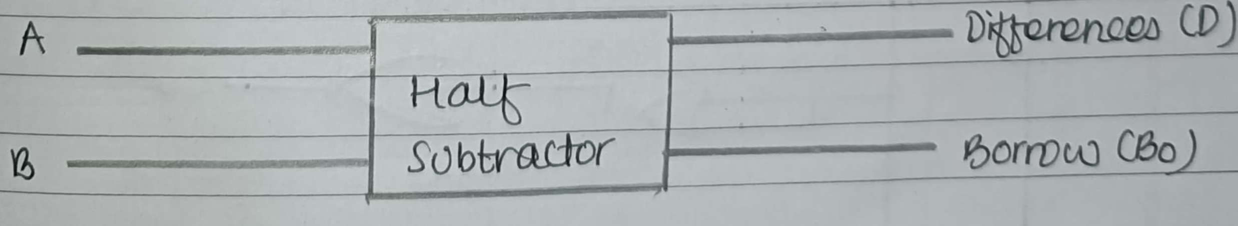


Fig: Block Diagram

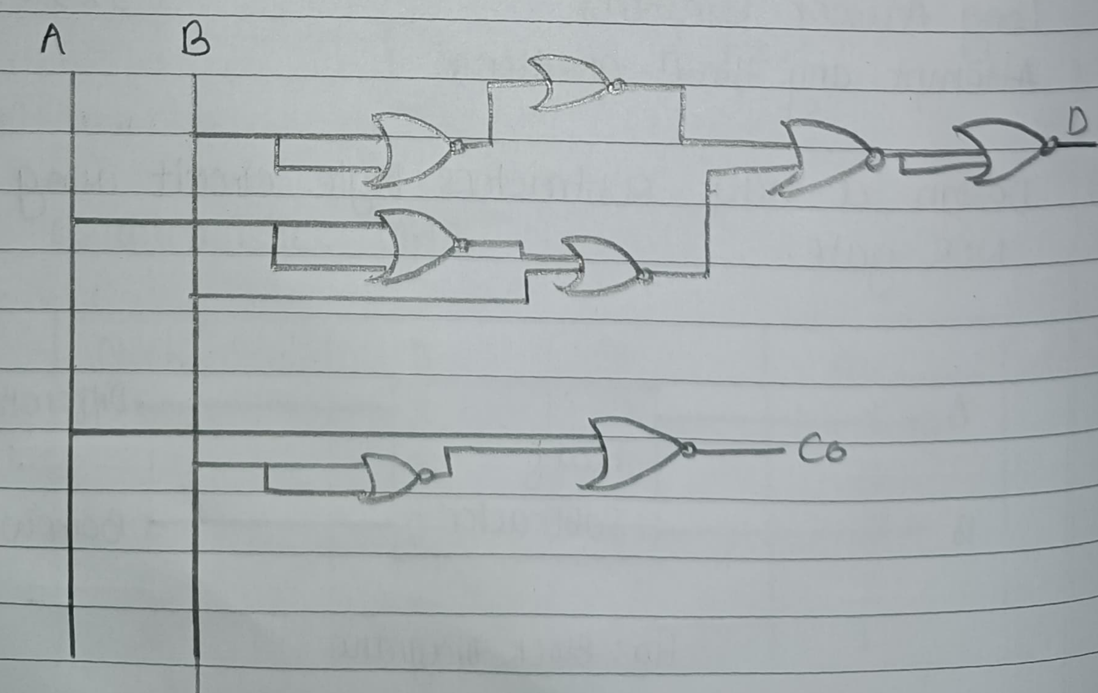
Truth Table

Input		Output	
A	B	D	Bo
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0

$$D = \bar{A}B + A.\bar{B}$$
$$= \overline{\overline{\bar{A}.B}} + \overline{\overline{A.\bar{B}}}$$
$$= \overline{A+\bar{B}} + \overline{\bar{A}+B}$$

$$Bo = \bar{A}.B$$
$$= \overline{\overline{\bar{A}.B}}$$
$$= \overline{A+\bar{B}}$$

Logic Circuit Diagram



18) Convert the following decimal numbers into hexadecimal and octal number.

a) 305

Decimal to Hexadecimal

$(305)_{10} \rightarrow (?)_{16}$

16	305	1
16	19	3
	1	

$\therefore (305)_{10} \rightarrow (131)_{16}$

Decimal to Octal

$(305)_{10} \rightarrow (?)_{8}$

8	305	1
8	38	6
	4	

$\therefore (305)_{10} \rightarrow (461)_{16}$

b) 225

Decimal to Hexadecimal

$(225)_{10} \rightarrow (?)_{16}$

16	225	1
	14	

$\therefore (225)_{10} \rightarrow (E1)_{16}$

Decimal to Octal

$(225)_{10} \rightarrow (?)_8$

8	225	1
8	28	4
	3	

$\therefore (225)_{10} \rightarrow (341)_8$

---

19) Describe 3-variable K-map with example.

⇒ A Karnaugh-map (K-map) is a visual tool used in digital logic design to simplify boolean expressions. A 3-variable K-map helps in minimizing boolean functions involving three variables.

A 3-variable K-map has 8 cells. Each cell represent a minterm or its corresponding decimal number from 0 to 7. The K-Map is arranged in a way that any two adjacent cells differ by only one bit in their binary representation.

m_0	m_1	m_3	m_2
m_4	m_5	m_7	m_6

F

A \ BC	$\bar{B}\bar{C}$	$\bar{B}C$	BC	$B\bar{C}$
$\bar{A}$	$\bar{A}\bar{B}\bar{C}$	$\bar{A}\bar{B}C$	$\bar{A}BC$	$\bar{A}B\bar{C}$
A	$A\bar{B}\bar{C}$	$A\bar{B}C$	ABC	$AB\bar{C}$

for example:

$F(A,B,C) = \Sigma(0,1,2,4,5,6)$

A \ BC	$\bar{B}\bar{C}$	$\bar{B}C$	BC	$B\bar{C}$
$\bar{A}$	1	1		1
A	1	1		1

$$\therefore F = \bar{B} + \bar{C}$$

20) Explain the triggering of flip-flop with example.

Group-D

Comprehensive Questions (2×10 = 20)
Attempt all questions.

21) What do you mean by ripple counter? Design the 4 bit ripple counter with timing diagram.

⇒ In Ripple Counter, first flip-flop is triggered by external clock and after that output of previous flip-flop triggered the next flip-flop. Ripple counter has slow operation. The propagation delay of the counter is commulative delay of all of the flip-flop.

4-bit ripple counter

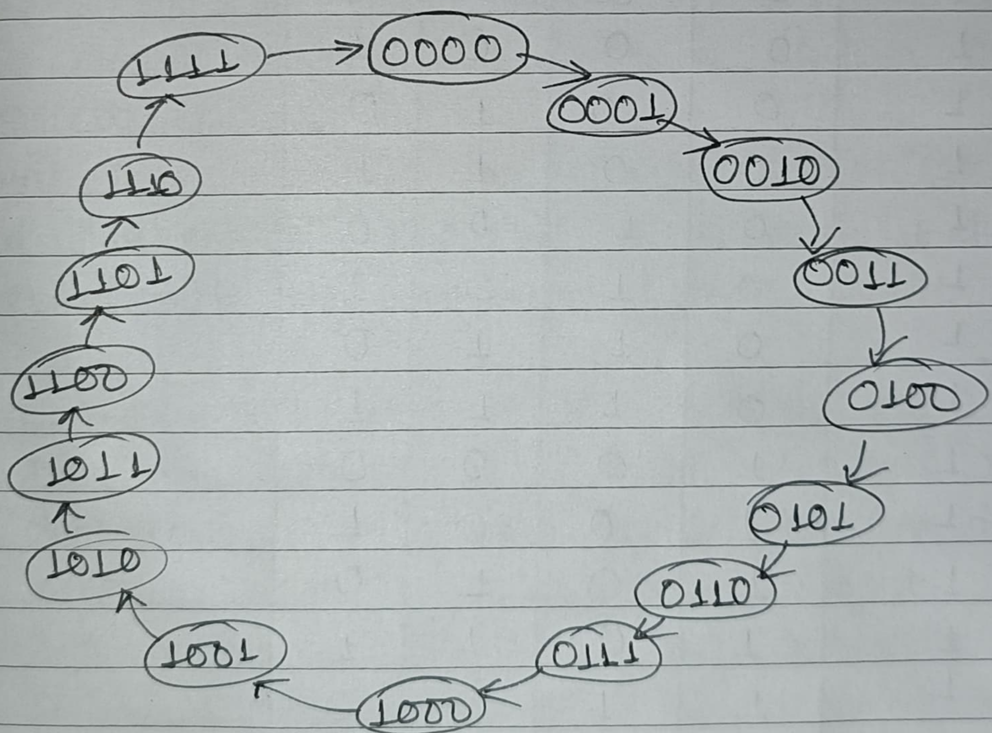
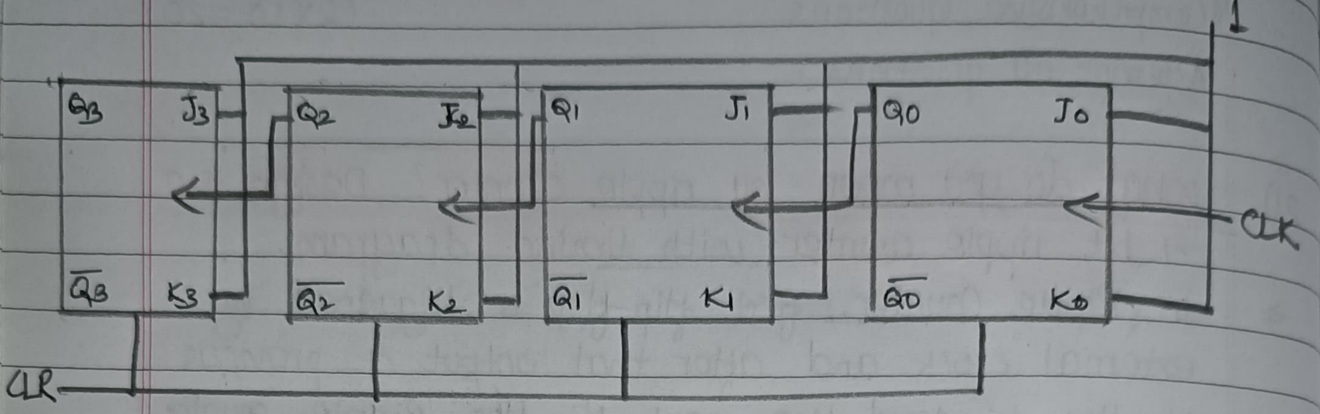


Fig: State Diagram

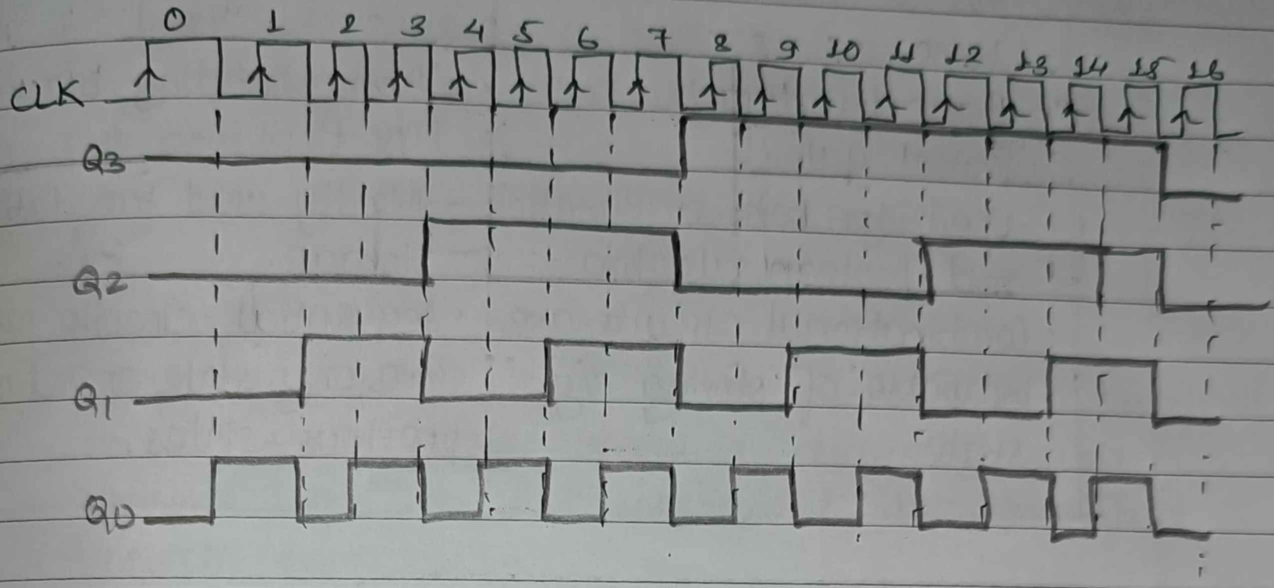
Logic Circuit Diagram



Operational Table

Clock	Q3	Q2	Q1	Q0
0	0	0	0	0
1	0	0	0	1
1	0	0	1	0
1	0	0	1	1
1	0	1	0	0
1	0	1	0	1
1	0	1	1	0
1	0	1	1	1
1	1	0	0	0
1	1	0	0	1
1	1	0	1	0
1	1	0	1	1
1	1	1	0	0
1	1	1	0	1
1	1	1	1	0
1	1	1	1	1
1	0	0	0	0

Timing Diagram



22) Differentiate between sequential and combinational logic circuits. Explain the steps to design of combinational logic.

⇒ Digital logic circuits are divided into "combinational" and "sequential" types. The differences between combinational and sequential logic circuit are:

Combinational Logic Circuit	Sequential Logic Circuit.
• This output is solely dependent on the current input	• This output is affected by both current and previous input.
• The process is quick.	• The process is slow.
• It is intended to be simple.	• When compared to combinational circuits, it is more effectively designed.
• There is no feedback from input to output.	• A feedback path exists between input and output.

Combinational Logic Circuit	Sequential Logic Circuit
• This is not dependent on time.	• This is time sensitive.
• Basic building blocks: logical gates.	• Basic building blocks: Flip flops.
• Used for both arithmetic and boolean algebra.	• Mostly used for data storage.
• Combinational circuits are incapable of storing any state.	• Sequential circuits can store any state or retain previous states.

Steps to design combinational circuits are:

- Understand the problem statement
  Clearly define the desired output functions based on given input variables. This could involve creating a truth table or analyzing a problem statement to determine what the circuit needs to do.

- Define inputs and outputs and assign with binary variables
  Identify how many Input variables and output variables are needed. This will help in creating the truth table or the boolean expressions.

- Derive input and output relationship using truth table
  Construct a truth table that lists all possible combinations of input values and their corresponding output values. The truth table is a crucial step as it directly reflects the desired operation of the circuit.

- **Derive output equations and simplify it if possible**
From the truth table, derive the boolean expressions for each output. Simplify the boolean expression using boolean algebra rules, K-map method etc. The goal is used to reduce the number of logic gates required, minimizing the circuit's complexity.

- **Draw logic circuit diagram**
Select appropriate logic gates based on the simplified boolean expressions. Create a diagram using choosen logic gates to represent the simplified boolean expressions.

March - April 2023

Group-A

Brief Answer Questions (10 x 1 = 10)

1) What is digital system?
⇒ Digital system is an electronic system that works with discrete values represented by 0 (low or off) and 1 (high or on) and store data in the form of binary bit.

2) How many different combination can be represented by 5-bit binary data?
⇒ $2^5 = 32$.
In 32 different combination, 5-bit binary data can be represented.