

CTEVT, DIPLOMA, QUESTION & SOLUTION

2078 to 2080

Digital Logic

(For Diploma I Yrs. II Part)

(Second & Third) Semester



Arjun Chaudhary

Published in : www.arjun00.com.np

www.facebook.com/arjun00.com.np

Authors © Arjun Chaudhary

Email:- info@arjun00.com.np

****Dont Remove these page, Otherwise You Got Copyright**

S.No Exam Year, Month

1. 2078 Magh Regular/Back
2. 2079 Falgun Regular/Back
3. 2080 Bhadra Regular/Back

**** "I have created this content for students' benefit, so please don't copy it or remove my logos without permission. "**



**** This Solution is for Educational Purpose Only.**

SCAN ME
To Visit My Website.



Digital Logic --- (DCOM) 2ndSem

(2078) Question Paper Solution.

Compile by © Arjun Chy

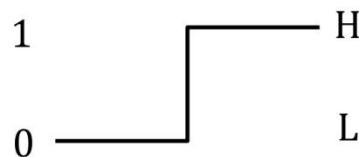
Website :- www.arjun00.com.np

Facebook :- www.facebook.com/Arjun00.com.np

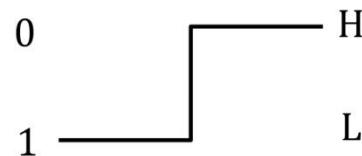
1. a) Differentiate analog and digital signals. What do you mean by positive and negative logic.

Analog Signals	Digital Signals
An analog signal is a continuous signal that represents physical measurements.	Digital signals are time separated signals which are generated using digital modulation.
It is denoted by sine waves	It is denoted by square waves
It uses a continuous range of values that help you to represent information.	Digital signal uses discrete 0 and 1 to represent information.
The analog signal bandwidth is low	The digital signal bandwidth is high.
Analog hardware never offers flexible implementation.	Digital hardware offers flexibility in implementation.
It is suited for audio and video transmission.	It is suited for Computing and digital electronics.
Analog signal doesn't offer any fixed range.	Digital signal has a finite number, i.e., 0 and 1.
Examples :- Temperature sensors, FM radio signals, Photocells, Light sensor, Resistive touch screen etc.	Examples :- Computers, CDs, DVDs etc.

- In Positive logic, the most positive voltages level represents logic 1 state or high level (H) and lowest voltage level represent logic 0 state or low level (L).



- In Negative logic, the most positive voltages level represents logic 0 state and lowest voltage level represent logic 1 state.



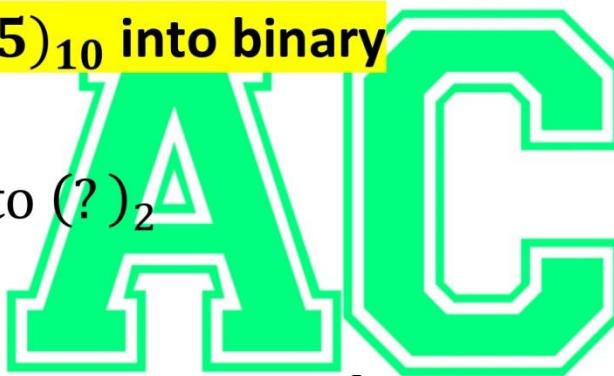
b) Perform following task:

i) Convert $(41.6875)_{10}$ into binary

➤ Solⁿ

$(41.6875)_{10}$ into $(?)_2$

Remainder		
2	41	- 1
2	20	- 0
2	10	- 0
2	5	- 1
2	2	- 0
		1



$$\begin{aligned}
 0.6875 \times 2 &= 1.375 \\
 0.375 \times 2 &= 0.75 \\
 0.75 \times 2 &= 1.5 \\
 0.5 \times 2 &= 1
 \end{aligned}$$

∴ Answer format : $(\text{Rem} \uparrow. \text{integer} \downarrow)_2$

$$\therefore (41.6875)_{10} = (101001.1011)_2$$

ii) Convert $(1001)_2$ into BCD

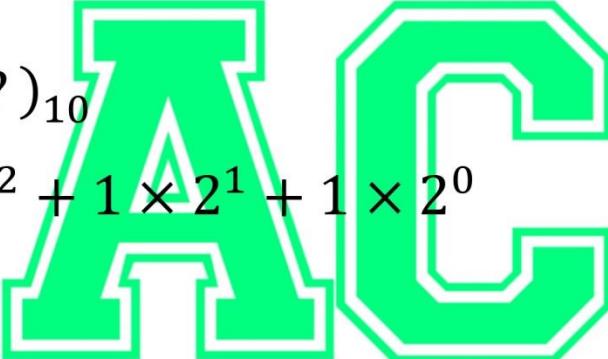
➤ Solⁿ

BN:	1	0	0	1
BCD	0001	0000	0000	0001

$$\therefore (1001)^2 = (00010000000001) \text{BCD}$$

iii) Convert $(1011)_2$ into decimal

➤ Solⁿ

$$\begin{aligned} & (1011)_2 \text{ into } (?)_{10} \\ &= 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 \\ &= 8 + 0 + 2 + 1 \\ &= 11 \end{aligned}$$


$$\therefore (1011)_2 \rightarrow (11)_{10}$$

IV) Convert $(12AB)_{16}$ into decimal

➤ Solⁿ

First we converted into Binary then, it is converted to decimal.

HD : 1 2 A(10) B(11)

BN : 0001 0010 1010 1011

(4th bit pairs)

$$BN = (0001001010101011)_2$$

(Binary to Decimal)

$$= 1 \times 2^{12} + 0 \times 2^{11} + 0 \times 2^{10} + 1 \times 2^9 + 0 \times 2^8 + 1 \times 2^7 + \\ 0 \times 2^6 + 1 \times 2^5 + 0 \times 2^4 + 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + \\ 1 \times 2^0$$

$$= 4096 + 0 + 0 + 512 + 0 + 128 + 0 + 32 + 0 + 8 + 0 + 2 \\ + 1$$

$$= (4779)_{10}$$

$$\therefore (12AB)_{16} = (4779)_{10}$$

2. a) Explain about basic gates with necessary truth table and logical expression.

➤ The basic gates with necessary truth table and logical expression are :-

❖ AND Gate

➤ AND gate is an electronic circuit, which produces true output (1) only when all the inputs are true and produces false output (0) when at least one input is false.

Logical expression: $X = A \cdot B$, where A and B are inputs, X is output and represents AND operation.

Gate symbol:



Truth table:

Input		Output
A	B	$X = A \cdot B$
0	0	0
0	1	0
1	0	0
1	1	1

❖OR Gate

➤ OR gate is an electronic circuit, which produces true output (1) when at least one input is true and produces false output (0) only when all the inputs are false.

Logical expression: $X = A + B$, where A and B are inputs, X is output and + represents OR operation.

Gate symbol:



Truth table:

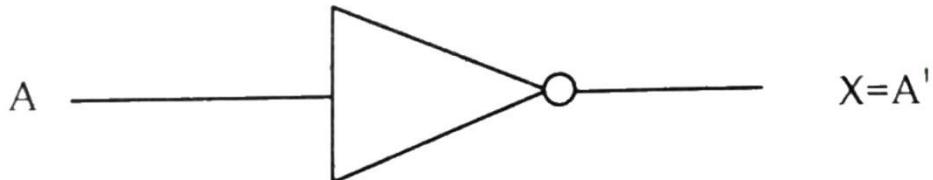
Input		Output
A	B	$X = A + B$
0	0	0
0	1	1
1	0	1
1	1	1

❖ NOT Gate

- NOT gate is an electronic circuit, which produces true output (1) when the input is false (0) and vice-versa. It inverts the input.

Logical expression: $X = A'$, where A is inputs, X is output.

Gate symbol:



Truth table:

Input	Output
A	$X = A'$
0	1
1	0

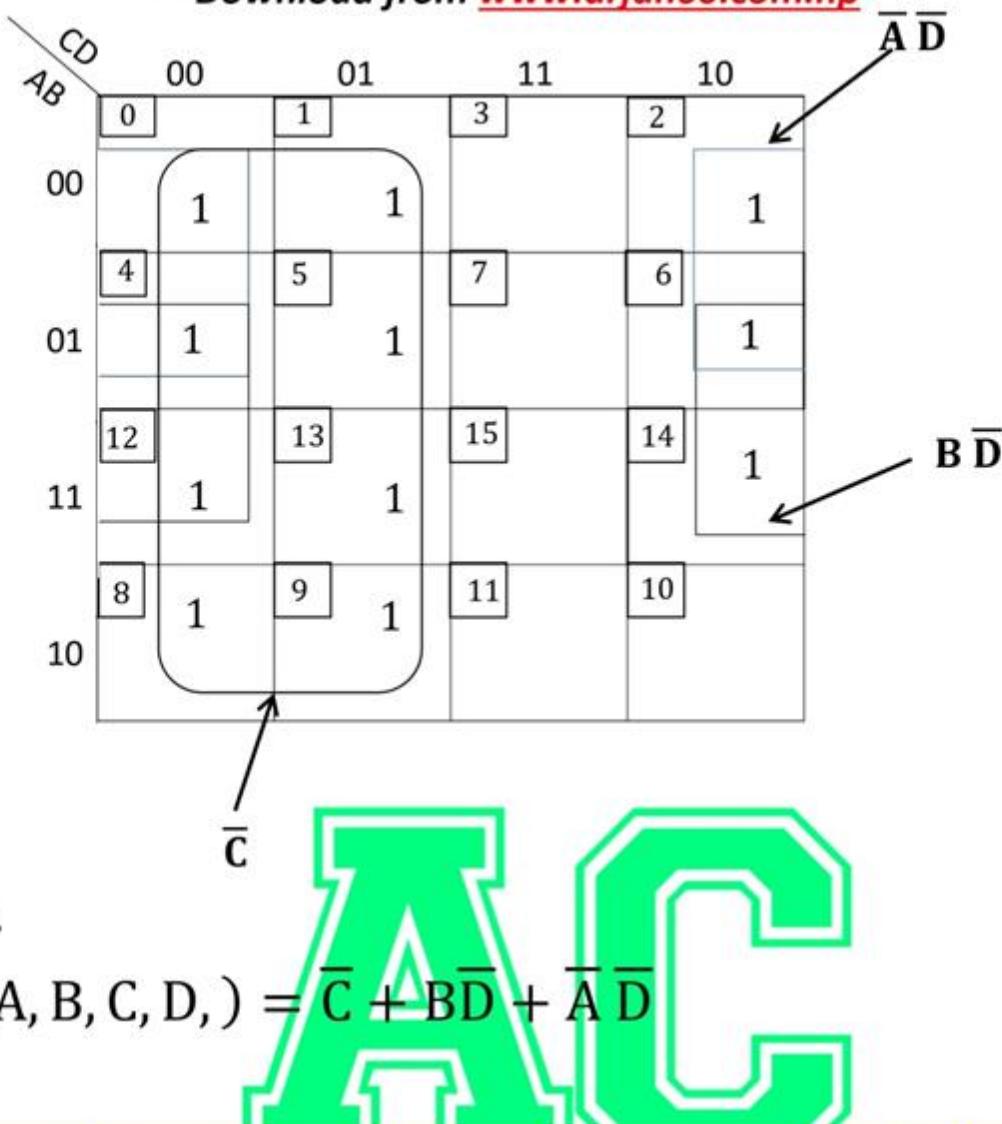
b) Explain universal gates and why are they called so? Reduce the following expression using k-map.

$$F(A, B, C, D) = \sum(0, 1, 2, 4, 5, 6, 8, 9, 12, 13, 14)$$

- NAND gates and NOR gates are called universal gates because any type gates or logic functions can be implemented by these gates. They are small in size and easy for fabrication.

$$\rightarrow F(A, B, C, D) = \sum(0, 1, 2, 4, 5, 6, 8, 9, 12, 13, 14)$$

Σ represent sum of product terms.



Simplified,

$$\therefore F(A, B, C, D) = \bar{C} + BD + \bar{A}D$$



3. a) State and prove De-Morgan theorem using truth table and logic diagram. Subtract $(11101010)_2$ from $(11111000)_2$ using 2's complement.

➤ First Law:

The De Morgan's first law states that, "The complement of a sum is equal to the product of the complements".

$$i.e. (A + B)' = A' \cdot B'$$

Proof:

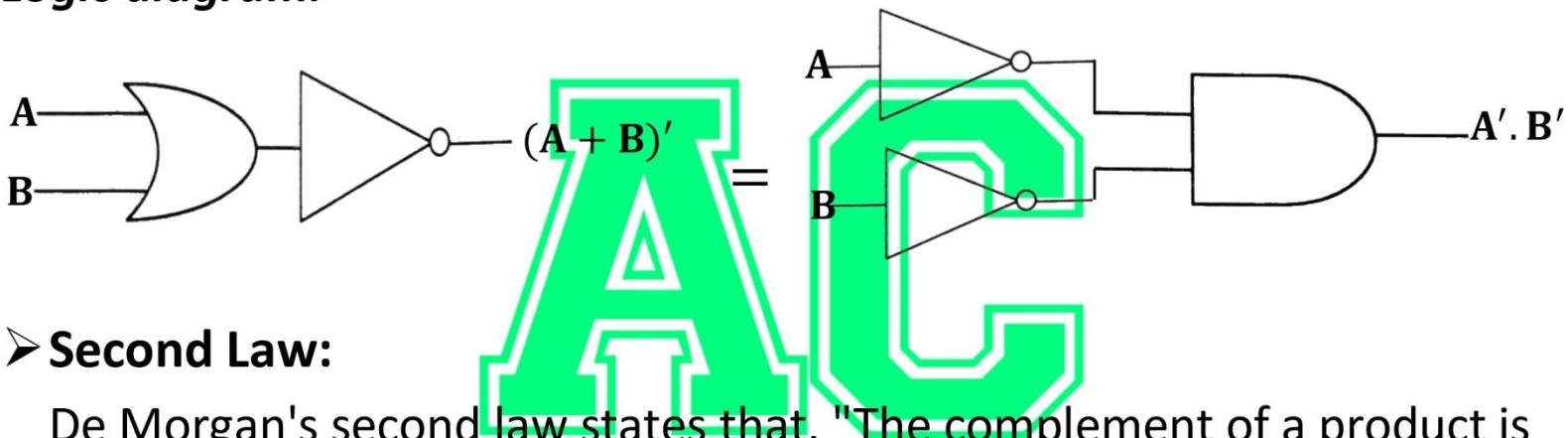
Truth table:

Input			Output 1			Output 2
A	B	$A + B$	$(A + B)'$	A'	B'	$A' \cdot B'$
0	0	0	1	1	1	1
0	1	1	0	1	0	0
1	0	1	0	0	1	0
1	1	1	0	0	0	0

Conclusion:

Comparing the values of output 1 of $(A + B)'$ and output 2 $A' \cdot B'$ from the truth table, both are equal, hence proved.

Logic diagram:



➤ Second Law:

De Morgan's second law states that, "The complement of a product is equal to the sum of the complements".

$$i.e. (A \cdot B)' = A' + B'$$

Proof:

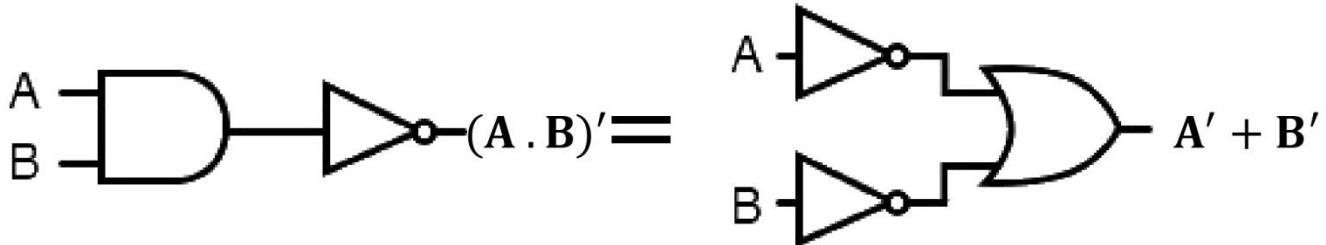
Truth table:

Input			Output 1			Output 2
A	B	$A \cdot B$	$(A \cdot B)'$	A'	B'	$A' + B'$
0	0	0	1	1	1	1
0	1	0	1	1	0	1
1	0	0	1	0	1	1
1	1	1	0	0	0	0

Conclusion:

Comparing the values of output 1 of $(A \cdot B)'$ and output 2 $A' + B'$ from the truth table, both are equal, hence proved.

Logic diagram:



❖ Subtract $(11101010)_2$ from $(11111000)_2$ using 2's complement.

➤ Solⁿ

$$X: 11111000$$

$$Y: 11101010$$

2's complement of Y:

$$\begin{array}{r}
 11101010 \\
 00010101 \\
 \hline
 +1 \\
 \hline
 00010110
 \end{array}$$

In 2's complement

The diagram shows the binary numbers 11101010 and 00010101 above a horizontal line. Below the line is the symbol '+1'. Below '+1' is the result 00010110. To the right of the result, the text 'In 2's complement' is written. Above the first two numbers, there are large green letters 'A' and 'C' with thick outlines, partially overlapping the digits. An arrow points from the text 'In 2's complement' towards the result 00010110.

Adding,

$$\begin{array}{r}
 11111000 \\
 +00010110 \\
 \hline
 100001110
 \end{array}$$

A horizontal line with a circle at its end, labeled 'Carry', points to the circled '1' in the result 100001110.

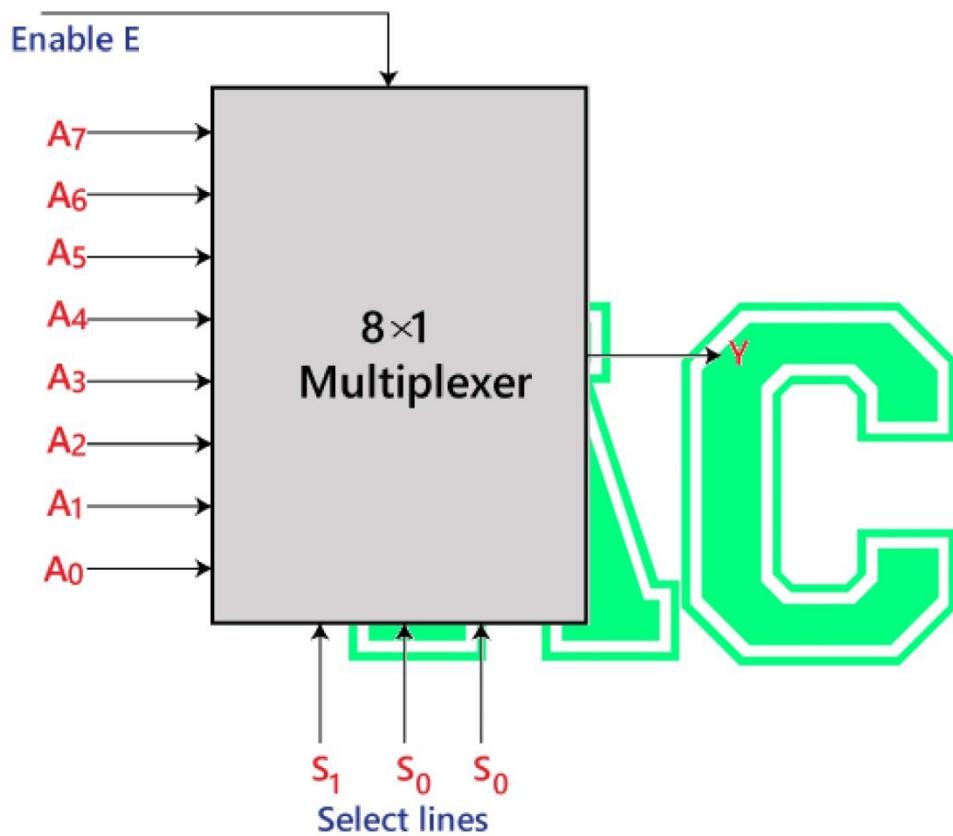
Since, there is additional carry so, discard it.

$$\text{Result} = 00001110 \Rightarrow (1110)_2$$

$$\therefore (11101010)_2 - (11111000)_2 = (1110)_2$$

b) Explain the working of 8 to 1 multiplexer with necessary diagram and truth table. What do you mean by combinational logic circuit?

➤ The block diagram and the truth table of the 8×1 multiplexer are given below:-



In the 8 to 1 multiplexer, there are total eight inputs, i.e., $A_0, A_1, A_2, A_3, A_4, A_5, A_6$, and A_7 , 3 selection lines, i.e., S_0, S_1 and S_2 and single output, i.e., Y . On the basis of the combination of inputs that are present at the selection lines S_0, S_1 and S_2 , one of these 8 inputs are connected to the output.

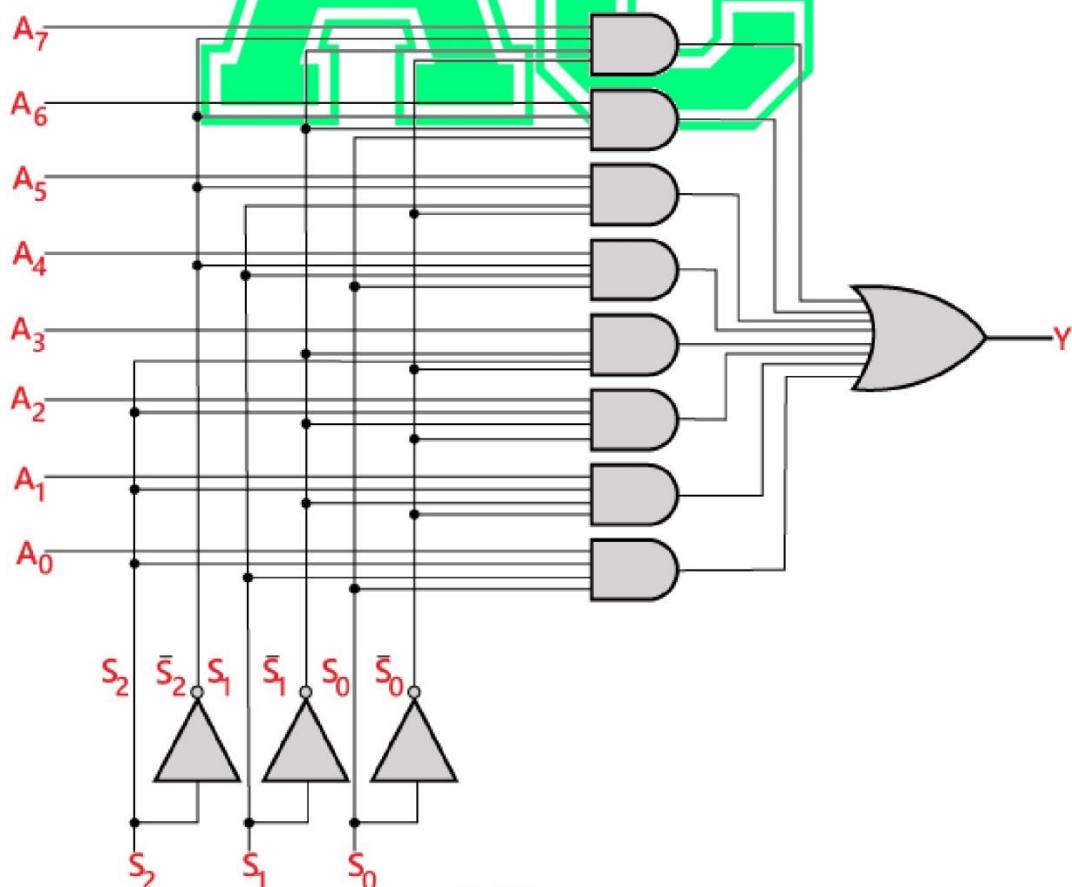
Truth table :

INPUTS			Output
S_2	S_1	S_0	Y
0	0	0	A_0
0	0	1	A_1
0	1	0	A_2
0	1	1	A_3
1	0	0	A_4
1	0	1	A_5
1	1	0	A_6
1	1	1	A_7

The logical expression of the term Y is as follows:-

$$\begin{aligned}
 Y = & S'_0 \cdot S'_1 \cdot S'_2 \cdot A_0 + S_0 \cdot S'_1 \cdot S'_2 \cdot A_1 + S'_0 \cdot S_1 \cdot S'_2 \cdot A_2 + S_0 \cdot S_1 \cdot S'_2 \cdot A_3 \\
 & + S'_0 \cdot S_1 \cdot S_2 \cdot A_4 + S_0 \cdot S'_1 \cdot S_2 \cdot A_5 + S'_0 \cdot S_1 \cdot S_2 \cdot A_6 + S_0 \cdot S_1 \cdot S_3 \cdot A_7
 \end{aligned}$$

Logical circuit of the above expression is given below:-



➤ A combinational logic circuit consists of logic gates whose output at any time are determined directly from the present combination of input without regard to previous inputs. A combinational circuit performs a specific information - processing operation fully specified logically by the set of Boolean function. A combinational circuit consists of input variables, logic gates, and output variables. The logic gates accept signals from the inputs and generates signals to the outputs. This process transforms binary information from the given input data to the required output data.

4. a) Define Boolean Algebra. State and explain basic properties of Boolean Algebra.

➤ Boolean algebra is the algebra of logic. It is a two valued system that represents Boolean variables and Boolean operations. It is introduced by George Boole. It is used for designing of digital electronic circuits.

➤ The basic properties/law of Boolean Algebra:-

- **Commutative law :-** This law allows change in the position of the input variables in OR and AND expression.

$$A + B = B + A$$

$$A \cdot B = B \cdot A$$

- **Associative law :-** This law states that it makes no difference in what order the variables are grouped when ORing or ANDing more than two variables.

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

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

- **Distributive law** :- This law permits factoring or multiplying an expression.

$$A(B + C) = AB + AC$$

$$AB + AC = A(B + C)$$

- **Inversion law** :- In Boolean algebra, the inversion law states that double inversion of variable results in the original variable itself.

$$\overline{\overline{A}} = A$$

- **AND law** :- These laws use the AND operation. Therefore they are called AND laws.

$$A \cdot 0 = 0$$

$$A \cdot 1 = A$$

$$A \cdot A = A$$

$$A \cdot \overline{A} = 0$$



- **OR law** :- These laws use the OR operation. Therefore they are called OR laws.

$$A + 0 = A$$

$$A + 1 = 1$$

$$A + A = A$$

$$A + \overline{A} = 1$$

b) Design BCD to Decimal decoder with necessary diagram and truth table and mention different types of decoder IC package.

➤ The BCD to decimal decoder converts each BCD code to its decimal equivalent. The technique employed is very similar to the one used in developing the 3-line-to-8 line decoder. It is referred as 4-line-to 10-line or 1 of 10 decodes.

Truth Table:-

Decimal digit	Binary input					Logic function
0	0	0	0	0	0	$\bar{A} \bar{B} \bar{C} \bar{D}$
1	0	0	0	0	1	$\bar{A} \bar{B} \bar{C} D$
2	0	0	1	0	0	$\bar{A} \bar{B} C \bar{D}$
3	0	0	1	1	0	$\bar{A} B \bar{C} D$
4	0	1	0	0	0	$\bar{A} B \bar{C} \bar{D}$
5	0	1	0	0	1	$\bar{A} B C \bar{D}$
6	0	1	1	0	0	$\bar{A} B C \bar{D}$
7	0	1	1	1	0	$\bar{A} B C D$
8	1	0	0	0	0	$A \bar{B} \bar{C} \bar{D}$
9	1	0	0	0	1	$A \bar{B} C \bar{D}$

Now we can develop a decoder based on each logic function and implement the SOP logic circuits.

This is Shown in figure below:-

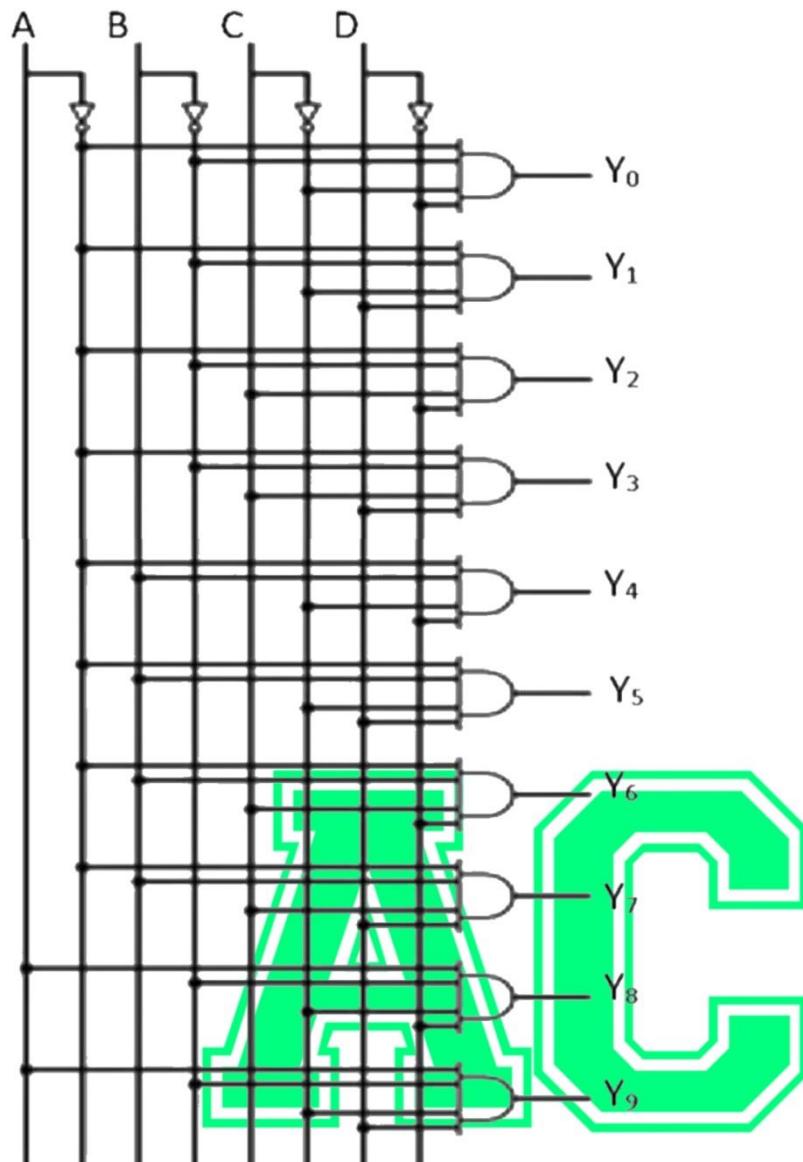


Fig: circuit diagram of BCD to decimal decoder

➤ The types of decoder IC package:-

- Dual In-Line Package
- Surface Mount Device Packages
- Ceramic Dual In-Line Package
- Quad Flat Package
- Leadless Chip Carrier
- Pin Grid Array

5. a) Define latch and flip-flop. Explain JK flip-flop with all necessary diagram; symbol, truth table.

- The **Latch** is a types of temporary storage device that has two stable states (bistable) and is normally placed in a category separate from that of flip-flops.
- The storage elements in clocked sequential circuits are called **flip-flops**. A flip-flop is defined as a circuit that alternates between two possible states when a pulse is received at the input.
- **JK flip-flop** has no invalid state as does the SR flip-flop. A JK flip-flop is a refinement of the SR flip-flop in that the indeterminate state of the SR type is defined in the JK type. Inputs J and K behave like inputs S and R to set and reset the flip-flop respectively. When $J = K = 1$, the flip-flop output toggles i.e., switches to its complements state; if $Q = 0$ it switches to $Q = 1$ and vice-versa.

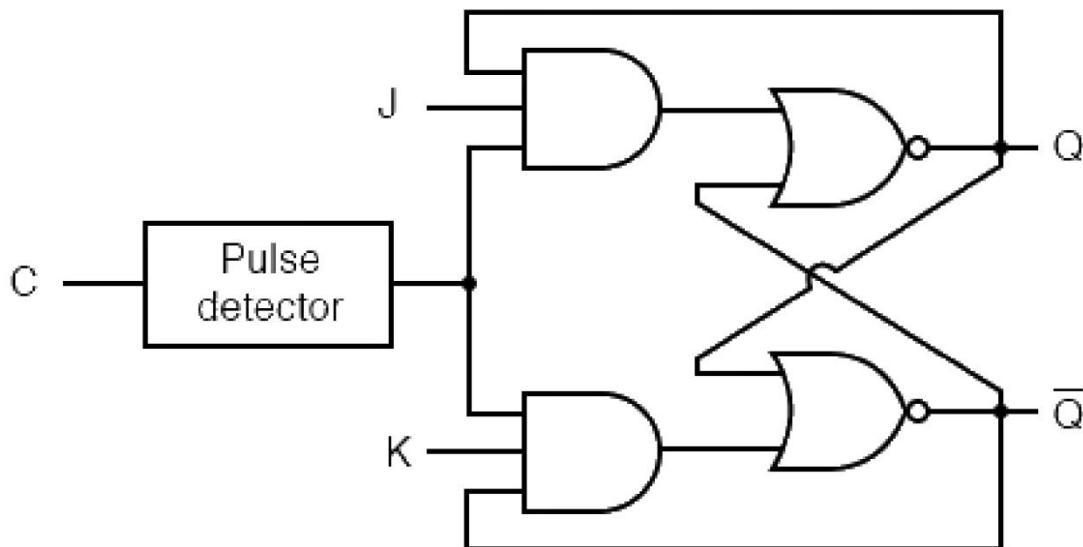
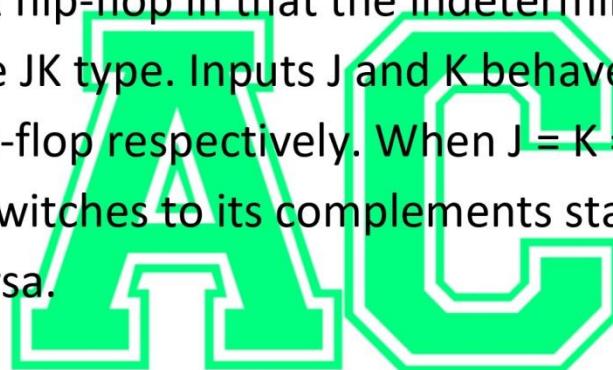


Figure : Logic Diagram

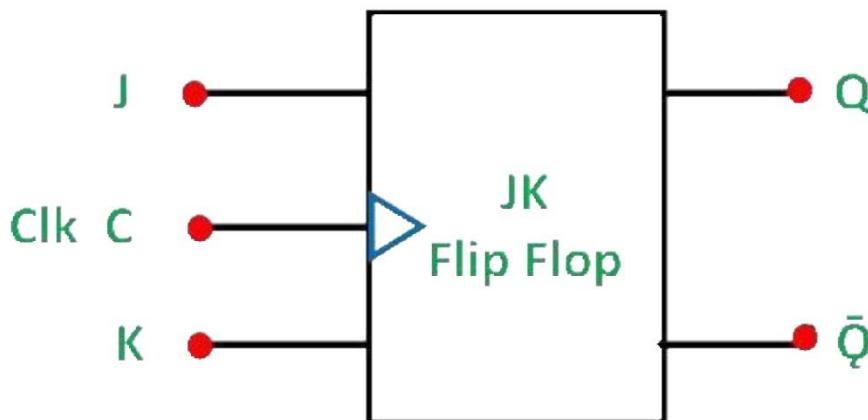


Figure : Logic Symbol

Truth Table:-

Input			Output		Comments
J	K	CLK	Q	\bar{Q}	
0	0	↑	Q_0	\bar{Q}_0	No change
0	1	↑	0	1	RESET
1	0	↑	1	0	SET
1	1	↑	\bar{Q}_0	Q_0	Toggle

b) Explain Ripple up counter with timing diagram.

➤ Ripple counter is a special type of Asynchronous counter in which the clock pulse ripples through the circuit. The n-MOD ripple counter forms by combining n number of flip-flops. The n-MOD ripple counter can count 2^n states, and then the counter resets to its initial value.

Features of the Ripple Counter:-

- Different types of flip flops with different clock pulse are used.
- It is an example of an asynchronous counter.
- The flip flops are used in toggle mode.

- The external clock pulse is applied to only one flip flop. The output of this flip flop is treated as a clock pulse for the next flip flop.
- In counting sequence, the flip flop in which external clock pulse is passed, act as LSB.

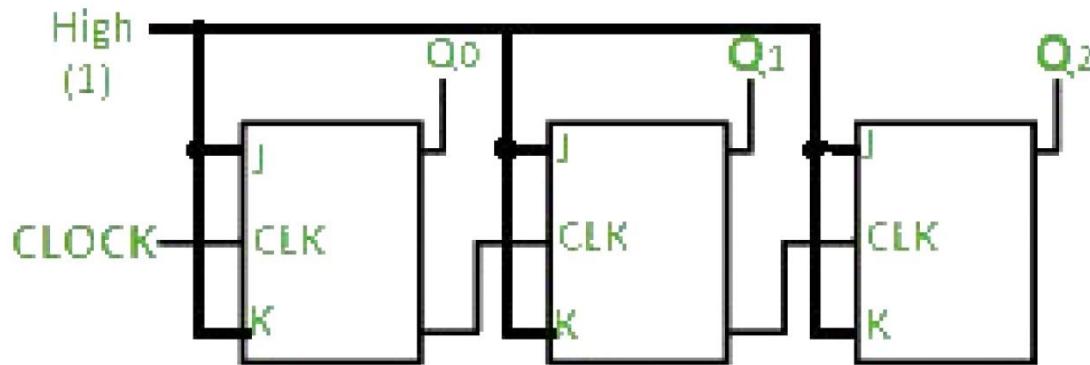
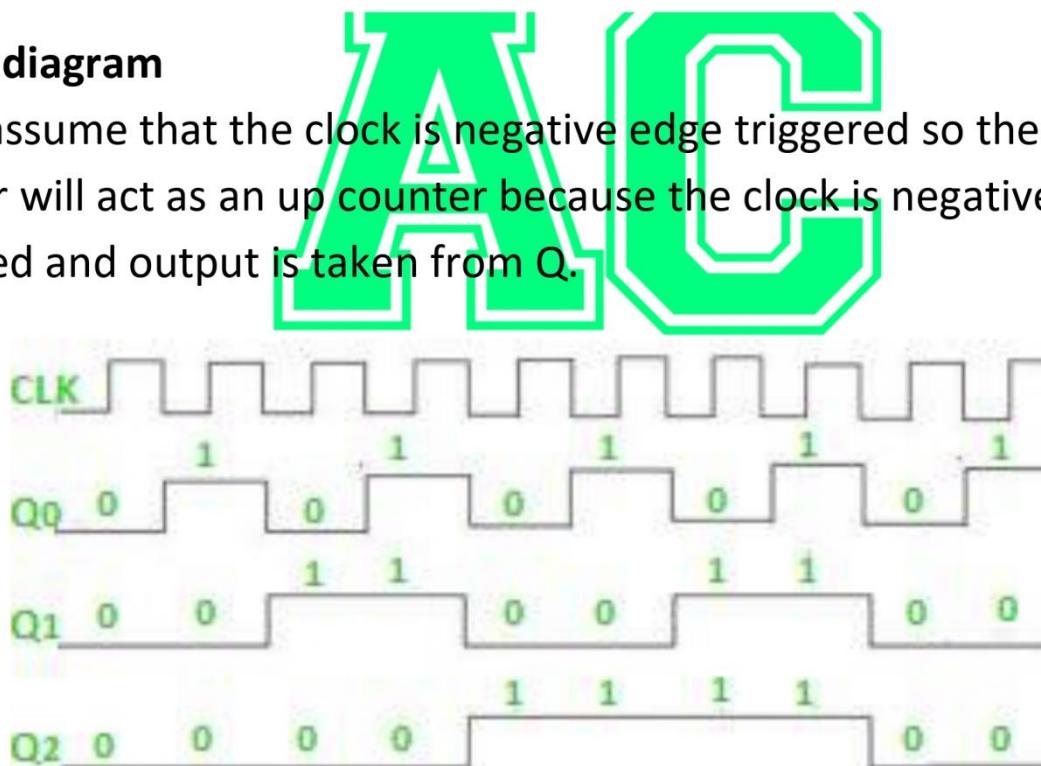


Fig:- 3-bit Ripple counter

➤ Timing diagram

Let us assume that the clock is negative edge triggered so the above the counter will act as an up counter because the clock is negative edge triggered and output is taken from Q.



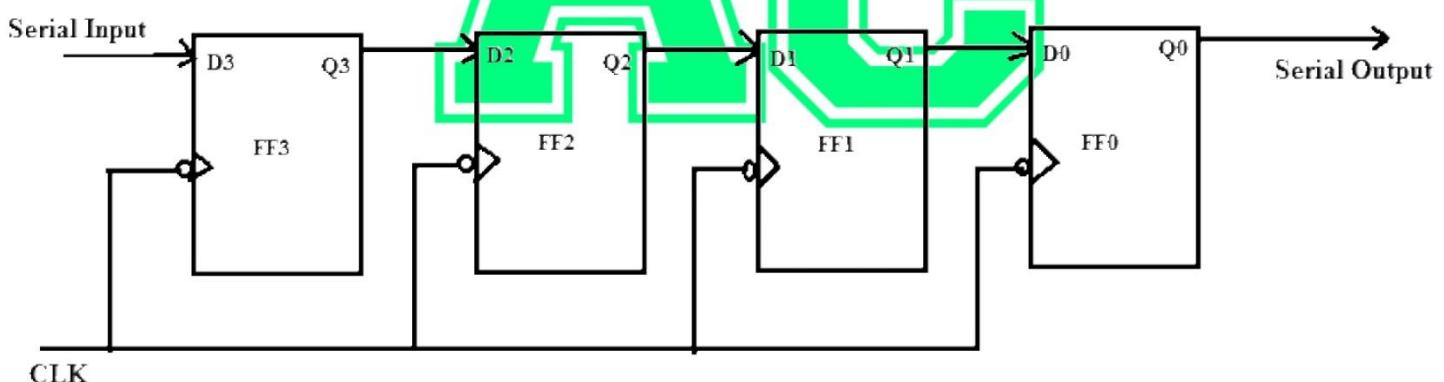
Counters are used very frequently to divide clock frequencies and their uses mainly involve digital clocks and in multiplexing. The widely known example of the counter is parallel to serial data conversion logic.

6. Write short notes on: (Any Four)

i) SISO shift register

➤ The term “SISO” stands for “Serial-In Serial-Out”. The SISO shift register circuit accepts serial data on its input pin and shifts it out serially on its output pin. The number of bits that can be shifted out before the next bit arrives depends on the speed of the clock signal that controls the operation of the shift register. This type of shift register can be used as a buffer between two asynchronous devices that communicate with each other using signals with different frequencies or phases.

The SISO shift register block diagram is shown below which includes 3-D flip-flops. The connection of these FFs can be done by connecting the one Flip Flops output to the next flip flop input. So these FFs are synchronous through each other because the equal CLK signal is applied in each Flip Flop.

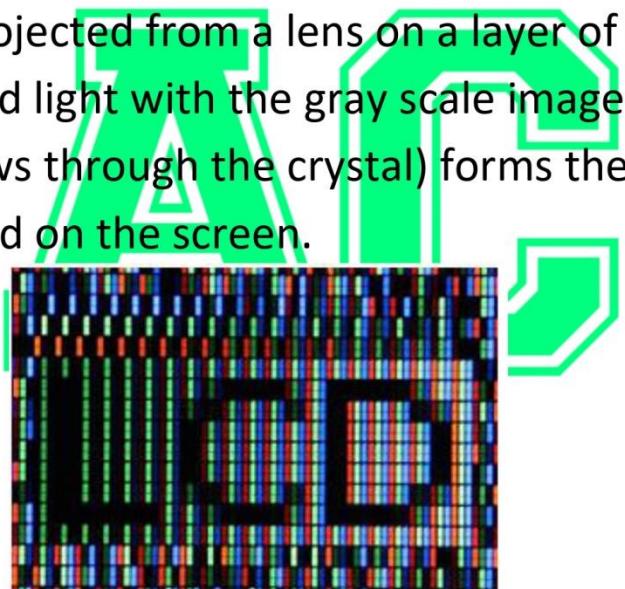


This shift register includes simply three connections the SI (serial input), SO (serial output), and CLK (clock signal). Here, the SI determines the input enters into the left-hand side flip-flop, the SO is the output taken from the right-hand side flip-flop & the sequencing CLK signal.

ii) LCD display

➤ A liquid crystal display or LCD draws its definition from its name itself. It is a combination of two states of matter, the solid and the liquid. LCD uses a liquid crystal to produce a visible image. Liquid crystal displays are super-thin technology display screens that are generally used in laptop computer screens, TVs, cell phones, and portable video games. LCD's technologies allow displays to be much thinner when compared to a cathode ray tube (CRT) technology.

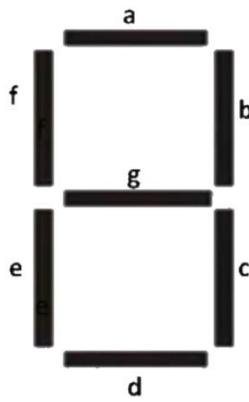
Liquid crystal display is composed of several layers which include two polarized panel filters and electrodes. LCD technology is used for displaying the image in a notebook or some other electronic devices like mini computers. Light is projected from a lens on a layer of liquid crystal. This combination of colored light with the gray scale image of the crystal (formed as electric current flows through the crystal) forms the colored image. This image is then displayed on the screen.



An LCD is either made up of an active matrix display grid or a passive display grid. Most of the Smartphone's with LCD technology uses active matrix display, but some of the older displays still make use of the passive display grid designs. Most of the electronic devices mainly depend on liquid crystal display technology for their display. The liquid has a unique advantage of having low power consumption than the LED or cathode ray tube.

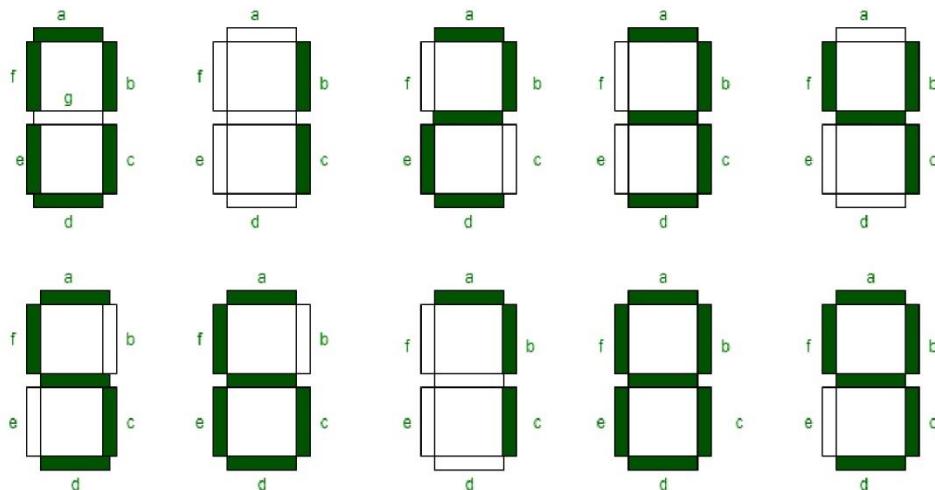
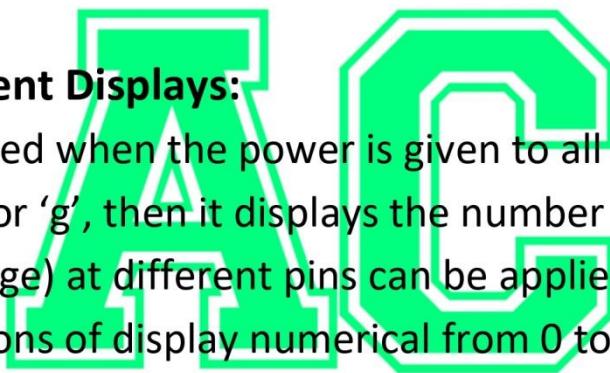
iii) 7-Segment Display

➤ Seven segment displays are the output display device that provides a way to display information in the form of images or text or decimal numbers which is an alternative to the more complex dot matrix displays. It is widely used in digital clocks, basic calculators, electronic meters, and other electronic devices that display numerical information. It consists of seven segments of light-emitting diodes (LEDs) which are assembled like numerical 8.



Working of Seven Segment Displays:

➤ The number 8 is displayed when the power is given to all the segments and if you disconnect the power for 'g', then it displays the number 0. In a seven-segment display, power (or voltage) at different pins can be applied at the same time, so we can form combinations of display numerical from 0 to 9. Since seven-segment displays can not form alphabets like X and Z, so it can not be used for the alphabet and they can be used only for displaying decimal numerical magnitudes. However, seven-segment displays can form alphabets A, B, C, D, E, and F, so they can also be used for representing hexadecimal digits.



iv) Half adder

➤ A half adder is a digital logic circuit that performs binary addition of two single-bit binary numbers. It has two inputs, A and B, and two outputs, SUM and CARRY. The SUM output is the least significant bit (LSB) of the result, while the CARRY output is the most significant bit (MSB) of the result, indicating whether there was a carry-over from the addition of the two inputs. The half adder can be implemented using basic gates such as XOR and AND gates.



Figure : Block diagram

Truth table: –

Inputs		Outputs	
A	B	Sum	Carry
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

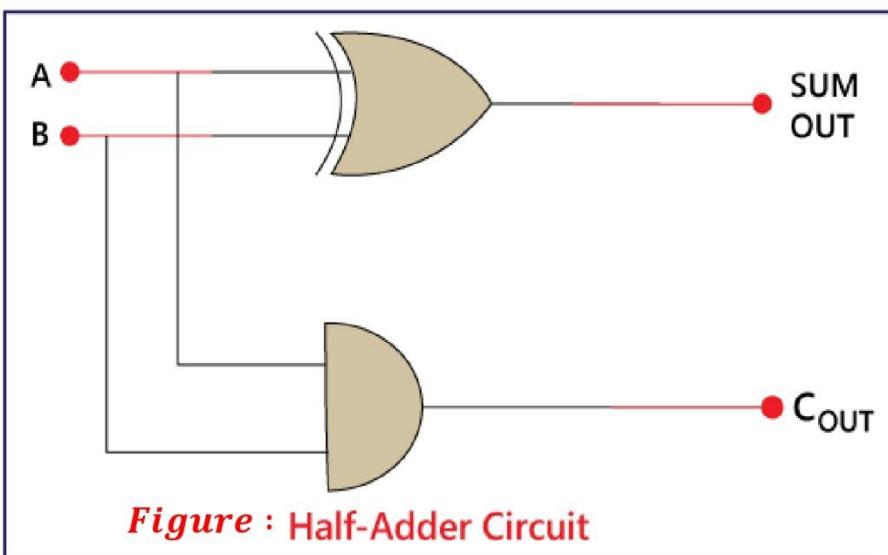


Figure : Half-Adder Circuit

v) Alphanumeric code

➤ Binary codes which are used for representing alphanumeric data like numbers & alphabetic characters are known as alphanumeric coded or character codes. These codes simply write alphanumeric data like alphabet letters, mathematical symbols, punctuation marks, and numbers in an understandable form by a computer. By using these codes, we can interface I/O devices like monitors, keyboards, and printers with a laptop. These codes also represent other characters like symbols & different required instructions for transmitting the information. An alphanumeric code should at least represent 10 digits and 26 letters of the alphabet i.e. total of 36 items.

➤ The types of Alphanumeric Code are :-

- **ASCII Code** :- ASCII (American Standard Code for Information Interchange) is a standard character encoding used in telecommunication. The ASCII pronounced 'ask-ee', is strictly a seven-bit code based on the English alphabet. ASCII codes are used to represent alphanumeric data. The code was first published as a standard in 1967.
- **EBCDIC Code** :- EBCDIC is another character encoding scheme, primarily used in IBM mainframes and midrange computers. It uses 8 bits to represent each character and includes a broader set of characters compared to ASCII. EBCDIC was historically more prevalent in IBM systems but is less common in modern computing.
- **Unicode** :- Unicode is a character encoding standard that aims to represent every character from every writing system in the world. It provides a unique code point for each character, allowing for universal representation of text across different platforms and languages. Unicode supports a wide range of characters, including those from ASCII and other character sets, making it a comprehensive and versatile encoding standard for global communication.

-The End-

Digital Logic --- (IT) 2nd Sem

(2079) Question Paper Solution.

Compile by © Arjun Chy

Website :- www.arjun00.com.np

Facebook :- www.facebook.com/Arjun00.com.np

1. a) Define signal. Differentiate between analog and digital signal with suitable example.

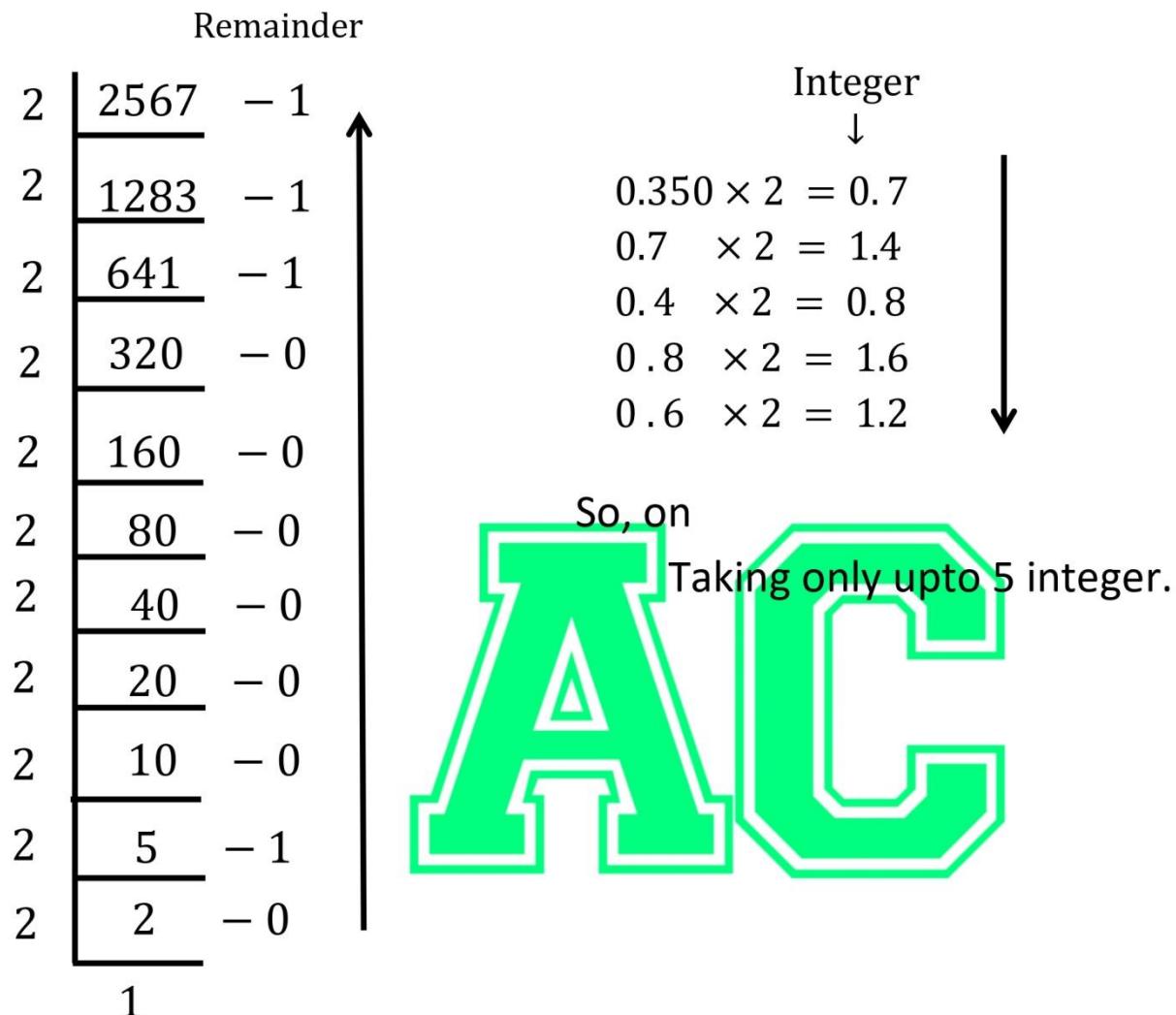
➤ Anything that carries information can be called as signal. It can also be defined as a physical quantity that varies with time, temperature, pressure or with any independent variables such as speech signal or video signal.

Analog Signals	Digital Signals
An analog signal is a continuous signal that represents physical measurements.	Digital signals are time separated signals which are generated using digital modulation.
It is denoted by sine waves	It is denoted by square waves
It uses a continuous range of values that help you to represent information.	Digital signal uses discrete 0 and 1 to represent information.
The analog signal bandwidth is low	The digital signal bandwidth is high.
Analog hardware never offers flexible implementation.	Digital hardware offers flexibility in implementation.
Examples :- Temperature sensors, FM radio signals, Photocells, Light sensor, Resistive touch screen etc.	Examples :- Computers, CDs, DVDs etc.

b. Convert the following number system:

i) $(2567.350)_{10} = (?)_2$

➤ Solⁿ



$\therefore (2567.350)_{10} = (101000000111.01011)_2$

ii) $(BCDE \cdot 4A)_{16} = (?)_8$

➤ Solⁿ

First converted to Binary, Then it is converted to octal.

HD: B(11) C(12) D(13) E(14). 4 A(10)

BN: 1011 1100 1101 1110 0100 1010

(8421)

$$\text{BN} \Rightarrow (1011110011011110.01001010)_2$$

← →

Taking pairs of 3bit.

BN: 001 011 110 011 011 110 . 010 010 100

ON: 1 3 6 3 3 6 . 2 2 4

(421)

$(136336.224)_8$

$$\therefore (BCDE \cdot 4A)_{16} = (136336.224)_8$$



iii) $(1100110011)_2 = (?)_{16}$

➤ Solⁿ

BN: 1100110011

Taking pairs of 4bit.

BN: 0011 0011 0011

HD: 3 3 3

(8421)

Hence,

$$\therefore (1100110011)_2 = (333)_{16}$$

iv) $(376.351)_8 = (?)_{10}$

➤ Solⁿ

First converted to Binary, Then it is converted to Decimal.

Octal (ON): 3 7 6 . 3 5 1
 (421)

BN: 011 111 110 . 011 101 001

$$\therefore \text{BN} \Rightarrow (01111110.011101001)_2$$

Binary to Decimal

$$\begin{aligned}
 &= 1 \times 2^7 + 1 \times 2^6 + 1 \times 2^5 + 1 \times 2^4 + 1 \times 2^3 + 1 \times 2^2 + 1 \times 2^1 \\
 &\quad + 0 \times 2^0 + 0 \times 2^{-1} + 1 \times 2^{-2} + 1 \times 2^{-3} + 1 \times 2^{-4} + 0 \times 2^{-5} \\
 &\quad + 1 \times 2^{-6} + 0 \times 2^{-7} + 0 \times 2^{-8} + 1 \times 2^{-9} \\
 &= 124 + 64 + 32 + 16 + 8 + 4 + 2 + 0 + 0 + 0.25 + 0.125 + 0.0625 \\
 &\quad + 0 + 0.015625 + 0 + 0 + 1.95312 \times 10^{-3} \\
 &= (254.45507)_{10}
 \end{aligned}$$

$$\therefore (376.351)_8 = (254.45507)_{10}$$

C. Perform the following operation:

i) Divide: $(1100110011)_2 / (1011)_2$

➤ Solⁿ

$$\begin{array}{r} 1011 \overline{)1100110011.00} \left(1001010.01 \\ - 1011 \downarrow \downarrow \downarrow \downarrow \downarrow \downarrow \\ \hline 0001110 \\ - 1011 \downarrow \downarrow \downarrow \downarrow \downarrow \downarrow \\ \hline 001101 \\ - 1011 \downarrow \downarrow \downarrow \downarrow \downarrow \downarrow \\ \hline 0010100 \\ - 1011 \hline 1001 \\ \text{Reminder} \end{array}$$

∴ $(1001010.01)_2$

ii) Multiply: $(1011001101)_2 \times (101101)_2$

➤ Solⁿ

$$\begin{array}{r}
 1011001101 \\
 \times 101101 \\
 \hline
 1011001101 \\
 + 0000000000 \qquad \qquad \qquad \Leftarrow 1^{\text{st}} \text{ intermediate Sum} \\
 \hline
 01011001101 \\
 + 1011001101 \qquad \qquad \qquad \Leftarrow 2^{\text{nd}} \text{ intermediate Sum} \\
 \hline
 111000000001 \\
 + 1011001101 \qquad \qquad \qquad \Leftarrow 3^{\text{rd}} \text{ intermediate Sum} \\
 \hline
 10010001101001 \\
 + 0000000000 \qquad \qquad \qquad \Leftarrow 4^{\text{th}} \text{ intermediate Sum} \\
 \hline
 10010001101001 \\
 + 1011001101 \qquad \qquad \qquad \Leftarrow \text{Final Sum} \\
 \hline
 111111000001001
 \end{array}$$

Hence,

$$\therefore (11111000001001)_2$$

2 a) State and prove De-Morgan's Theorem with necessary diagram and truth table.

➤ First Law:

The De Morgan's first law states that, "The complement of a sum is equal to the product of the complements".

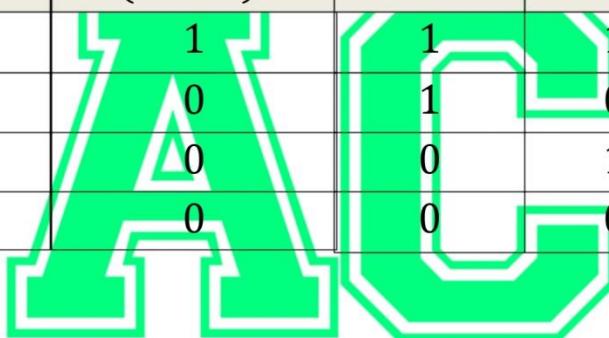
$$i.e. (A + B)' = A' \cdot B'$$

Proof:

Truth table:

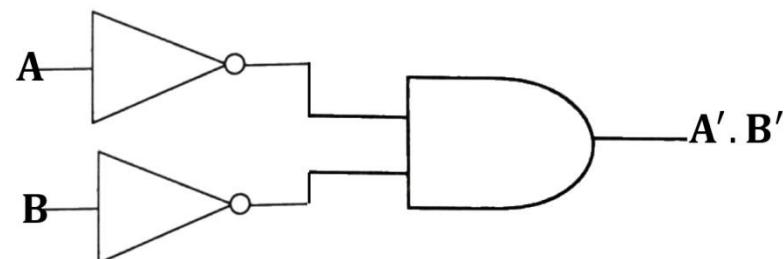
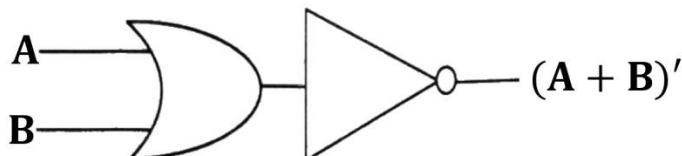
Input		$A + B$	$(A + B)'$	Output 1		$A' \cdot B'$	Output 2
A	B			A'	B'		
0	0	0	1	1	1	1	1
0	1	1	0	1	0	0	0
1	0	1	0	0	1	0	0
1	1	1	0	0	0	0	0

Conclusion:



Comparing the values of output 1 of $(A + B)'$ and output 2 $A' \cdot B'$ from the truth table, both are equal, hence proved.

Logic diagram:



➤ Second Law:

De Morgan's second law states that, "The complement of a product is equal to the sum of the complements".

$$i.e. (A \cdot B)' = A' + B'$$

Proof:

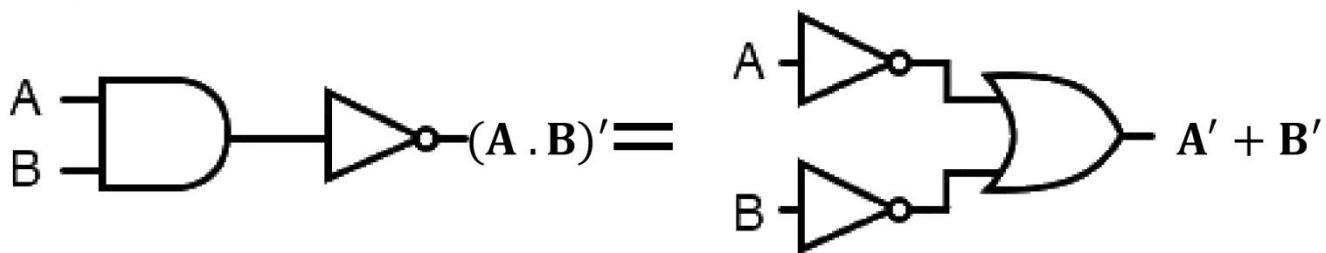
Truth table:

Input			Output 1			Output 2
A	B	$A \cdot B$	$(A \cdot B)'$	A'	B'	$A' + B'$
0	0	0	1	1	1	1
0	1	0	1	1	0	1
1	0	0	1	0	1	1
1	1	1	0	0	0	0

Conclusion:

Comparing the values of output 1 of $(A \cdot B)'$ and output 2 $A' + B'$ from the truth table, both are equal, hence proved.

Logic diagram:



b. Simplify the following expression using Boolean Algebra.

i) $Z(Y + Z)(X + Y + Z) = Z$

LHS

$$Z(Y + Z)(X + Y + Z)$$

$$Z \cdot Y (X + Y + Z) + Z \cdot Z (X + Y + Z)$$

$$ZY(X + Y + Z) + Z(X + Y + Z)$$

{ $\because ZZ = Z$, Using idempotent law}

$$Z(X + Y + Z)(1 + Y)$$

$$Z(X + Y + Z) \cdot 1$$

$$ZX + YZ + ZZ$$

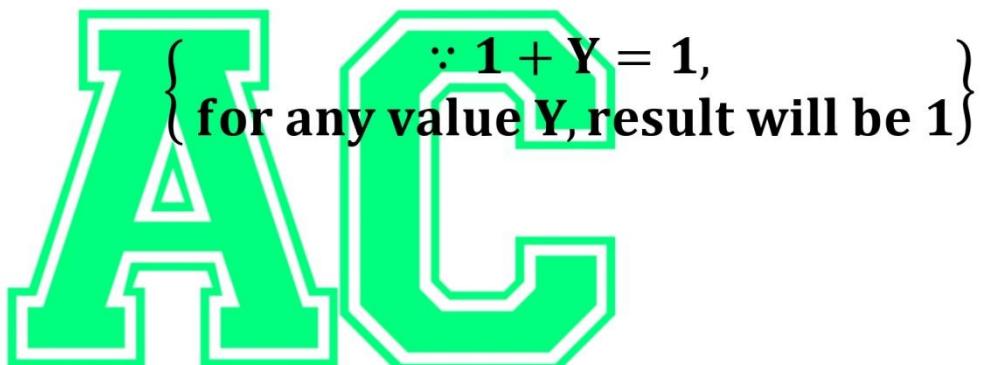
$$ZX + YZ + Z$$

$$ZX + Z(1 + Y)$$

$$ZX + Z$$

$$Z(X + 1)$$

$$Z \cdot 1$$



$\therefore Z$ Proved.

ii) $A'B'C + AB'C + ABC + BC'$

➤ Solⁿ

$$A'B'C + AB'C + ABC + BC'$$

$$= (A' + A) BC + AB'C + BC' \quad \left. \begin{array}{l} \because A' + A = 1, \\ \text{Using Complement law} \end{array} \right\}$$

$$= 1 \cdot BC + AB'C + BC'$$

$$= BC + AB'C + BC'$$

$$= B(C + C') + AB'C$$

$$= B \cdot 1 + AB'C$$

$$= B + AB'C$$

$$\therefore B + AB'C$$



3. a) Simplify the following expression using k-map.

i) $\sum F(A, B, C, D) = \pi m(2, 3, 4, 5, 7, 10, 11, 14) + \sum d(0, 1, 6, 15)$

➤ $\pi m(2, 3, 4, 5, 7, 10, 11, 14) + \sum d(0, 1, 6, 15)$

Truth table of ABCD, K – map

		CD AB	00	01	11	10	
		0	1	3	2		
00	0	X	X	0	0		
	4	0	0	7	6	X	
01	12		13	15	X	0	
	8		9	11	10	0	0

$X \rightarrow$ (don't care condition)

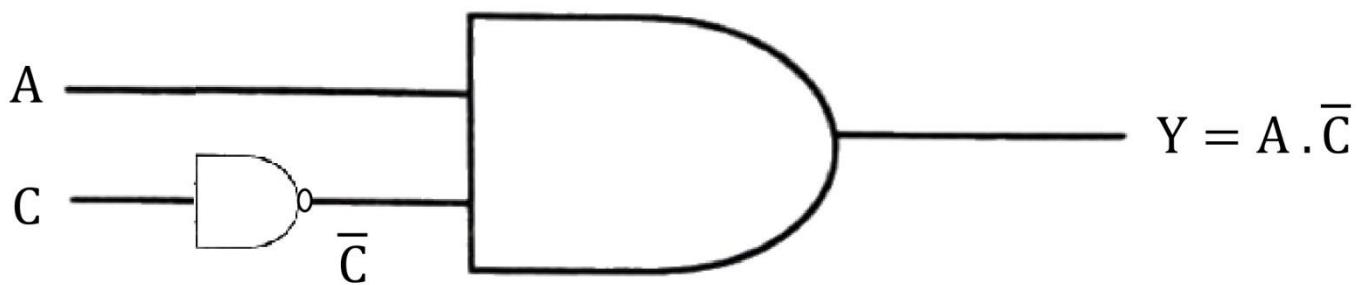
Here,

π represent Product of sum & d represent don't care condition.

From table,

Simplified POS expression is $Y = A \cdot \bar{C}$

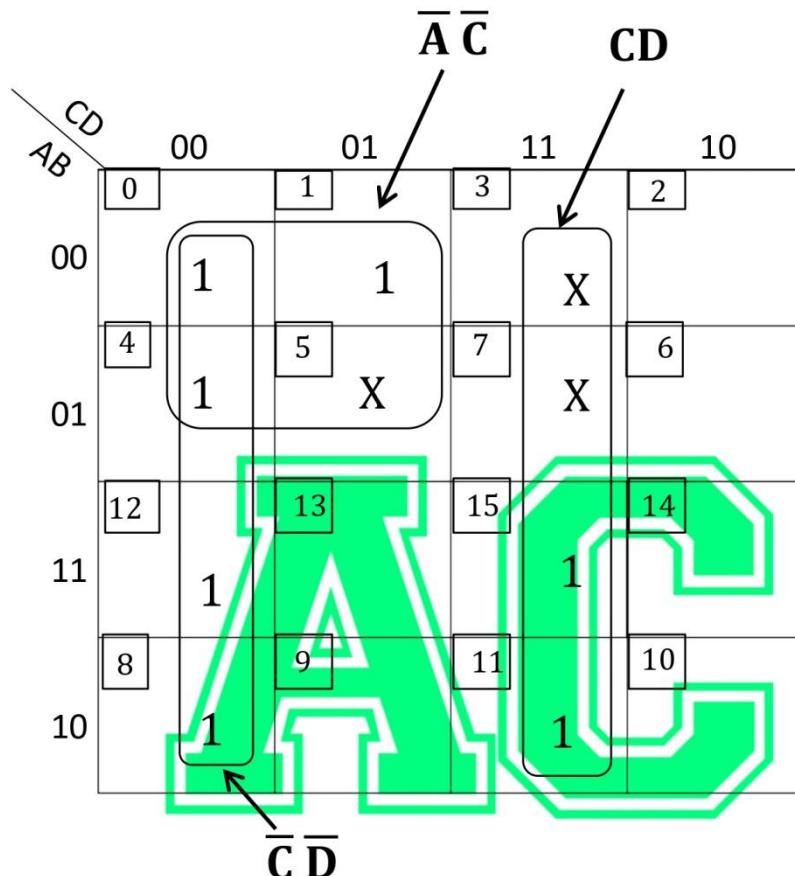
Logic diagram,



$$\text{ii.) } \sum F(A, B, C, D) = \sum M(0, 1, 4, 8, 11, 12, 15) + \sum d(2, 3, 5, 7)$$

Here,

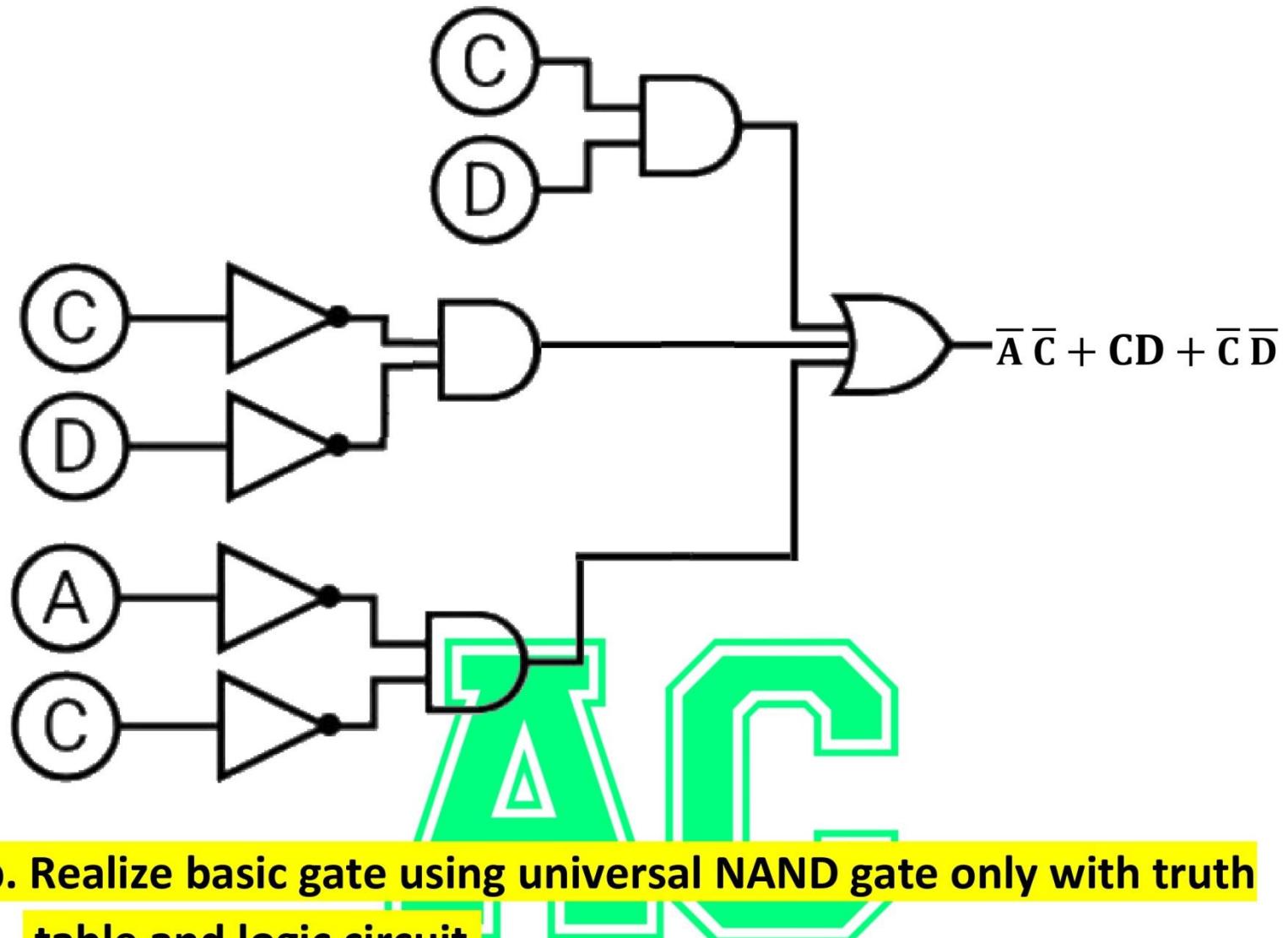
$\sum M$ represent sum of Product terms & d represent don't care condition.



Simplified SOP,

$$F(A, B, C, D) = \bar{A} \bar{C} + CD + \bar{C} \bar{D}$$

Logic diagram,



b. Realize basic gate using universal NAND gate only with truth table and logic circuit.

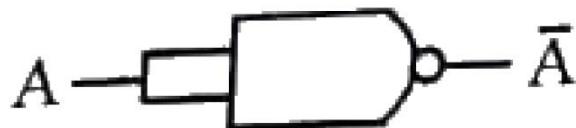
➤ NAND gate is a universal gate because the basic logic gates (NOT, OR, AND) can be realized from NAND gates.

- **NOT from NAND**

Truth table:-

Input	Output
A	X = \bar{A}
0	1
1	0

Logic circuit:-

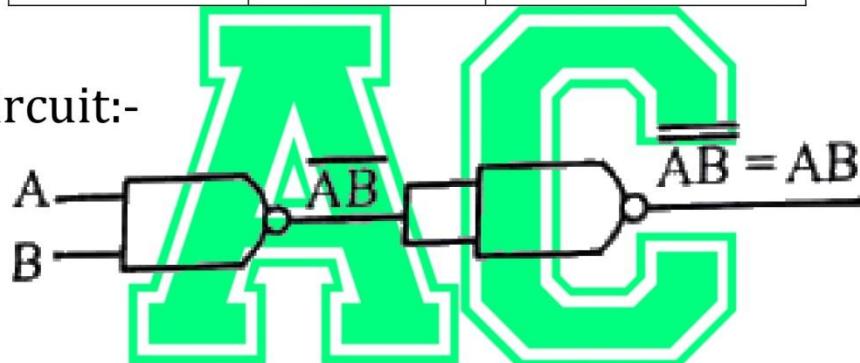


- **AND from NAND**

Truth table:-

Input		Output
A	B	X = A · B
0	0	0
0	1	0
1	0	0
1	1	1

Logic circuit:-

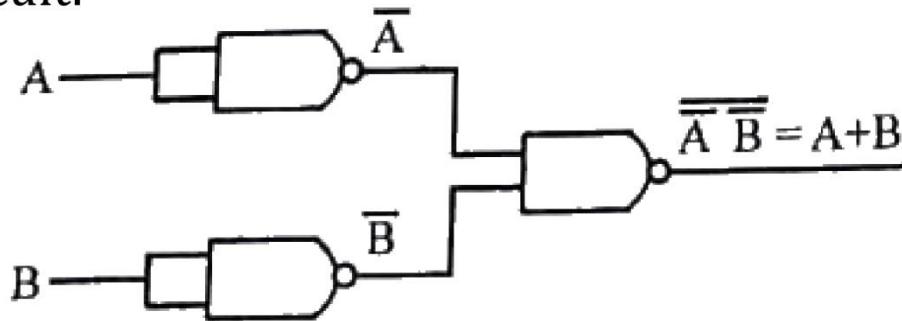


- **OR from NAND**

Truth table:-

Input		Output
A	B	X = A + B
0	0	0
0	1	1
1	0	1
1	1	1

Logic circuit:-



4. a) Define multiplexer. Explain the operation of full subtractor with clear logic diagram, truth table and expression.

- A multiplexer is a combinational circuit that selects binary information from one of many input lines and directs it to a single output line. The selection of a particular input line is controlled by a set of select ion lines. Thus, a multiplexer is also called a data selector.
- A full subtractor is a combinational circuit that performs a subtraction between two bits, taking into account that a 1 may have been borrowed by a lower significant stage. This circuit has three inputs and two outputs. The three inputs, x, y and z denote the minuend, subtrahend, and previous borrow respectively. The two outputs, D and B represent the difference and output borrow respectively.

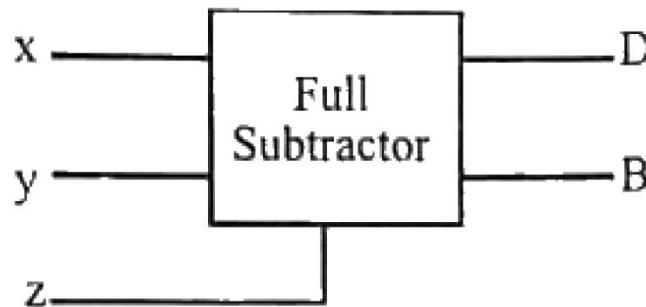


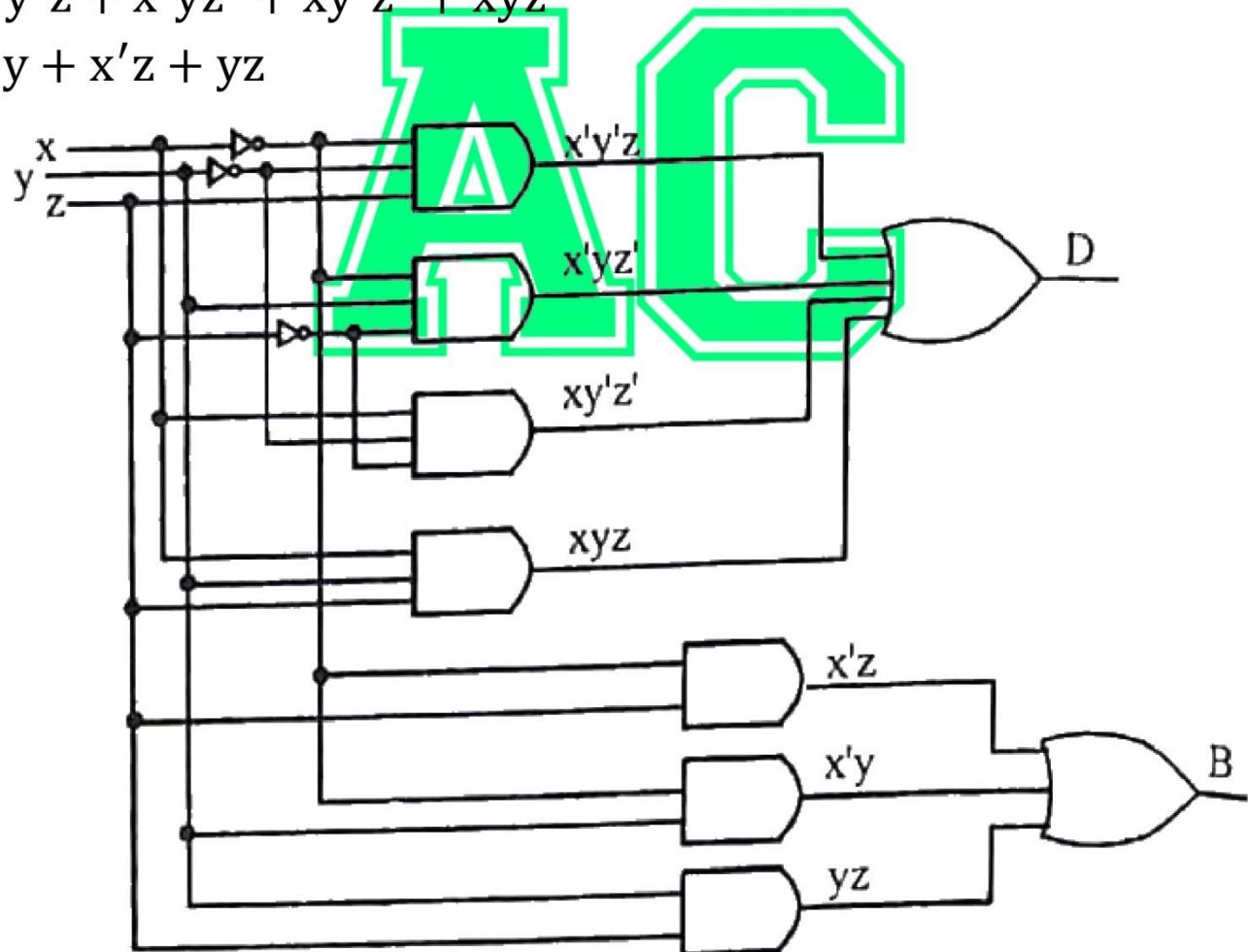
fig:- Block diagram of full subtractor

Truth table:-

x	y	z	D	B
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	0	1
1	0	0	1	0
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1

$$D = x'y'z + x'yz' + xy'z' + xyz$$

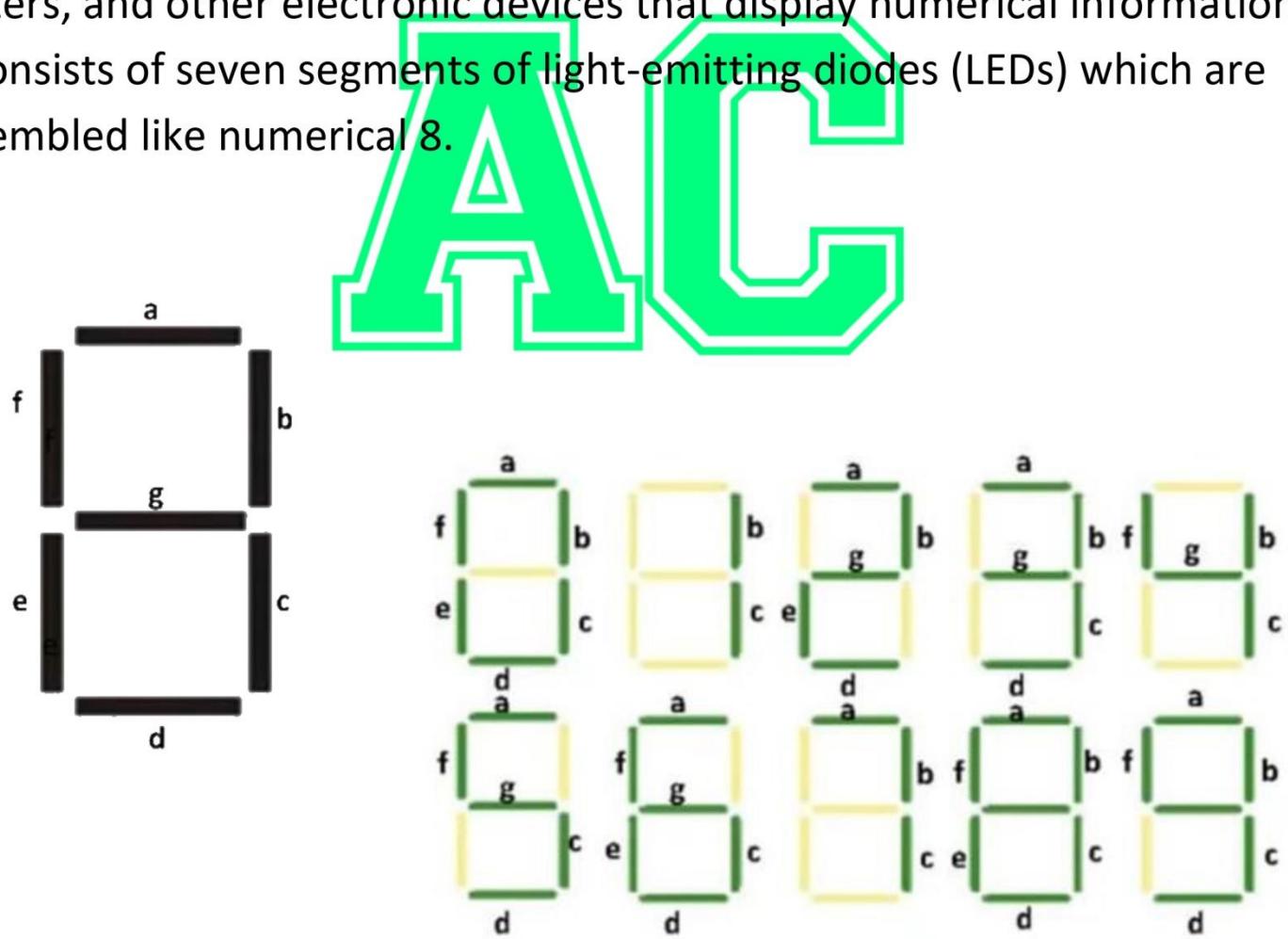
$$B = x'y + x'z + yz$$



Logic diagram

b) Define encoder. Design and explain seven segment Display decoder with necessary diagram and truth table.

- An encoder is a combinational circuit that performs the inverse operation of a decoder. An encoder has 2^n (or fewer) input lines and n output lines.
- A Seven segment displays are the output display device that provides a way to display information in the form of images or text or decimal numbers which is an alternative to the more complex dot matrix displays. It is widely used in digital clocks, basic calculators, electronic meters, and other electronic devices that display numerical information. It consists of seven segments of light-emitting diodes (LEDs) which are assembled like numerical 8.



To design seven segment display consider

Number of input: A, B, C, D

Number of output: a, b, c, d, e, f, g

Display format:

Decimal number	Segment activated
0	a, b, c, d, e, f
1	b, c
2	a, b, g ,e ,d
3	a, b, c, d, g
4	b, c, f, g
5	a, c, d, f, g
6	a, c, d, e, f, g
7	a, b, c
8	a, b, c, d, e, f, g
9	a, b, c, d, f, g

Truth table:

Inputs				Outputs						
A	B	C	D	a	b	c	d	e	f	g
0	0	0	0	1	1	1	1	1	1	0
0	0	0	1	0	1	1	0	0	0	0
0	0	1	0	1	1	0	1	1	0	1
0	0	1	1	1	1	1	1	0	0	1
0	1	0	0	0	1	1	0	0	1	1
0	1	0	1	1	0	1	1	0	1	1
0	1	1	0	1	0	1	1	1	1	1
0	1	1	1	1	1	1	0	0	0	0
1	0	0	0	1	1	1	1	1	1	1
1	0	0	1	1	1	1	1	0	1	1

5. a) Define counter. Explain the operation of 'T' flip flop with necessary diagram and truth table.

- Counter is also a register capable of add or subtract one bit for each clock pulse (i. e. at high or low state of CP) from already stored or previous binary data.
- The T-flip-flop is a modification of JK-flip-flop where JK-flip-flop is changed into single input flip-flop. In practical application intention is to change stored state i. e. state of Q and \bar{Q} (i. e toggling) and it can be done by giving input '1' to both J and K terminals while to maintain no change, we have to give input zero to both terminals. Now, why not to make single terminal now? Thus, this type of flip-flop is called T-flip-flop.

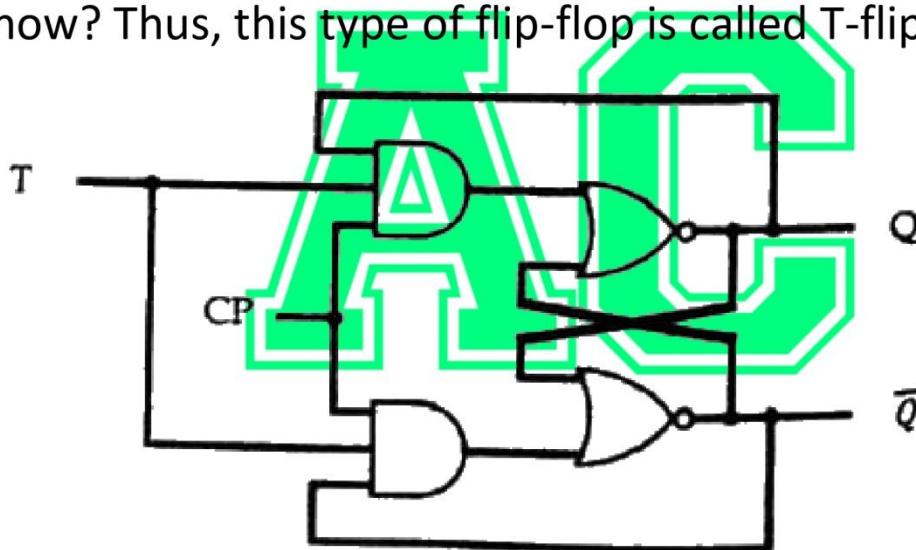


Figure : Logic Diagram

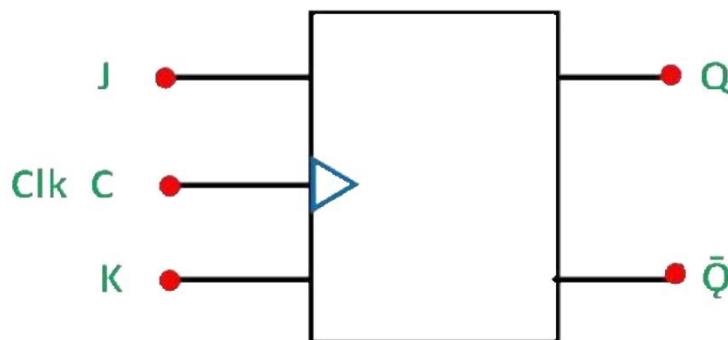


Figure : Logic Symbol

Truth table:

T	$Q(t + 1)$	Effect
0	$Q(t)$	No change
1	$Q'(t)$	Complement

b) Define adder. Explain about ripple counter with necessary diagram.

- A digital circuit that performs the addition of binary numbers or binary-coded decimal, commonly used in computer arithmetic to add two or more binary inputs.
- Ripple counter is a special type of Asynchronous counter in which the clock pulse ripples through the circuit. The n-MOD ripple counter forms by combining n number of flip-flops. The n-MOD ripple counter can count 2^n states, and then the counter resets to its initial value.

➤ Features of the Ripple Counter:-

- Different types of flip flops with different clock pulse are used.
- It is an example of an asynchronous counter.
- The flip flops are used in toggle mode.

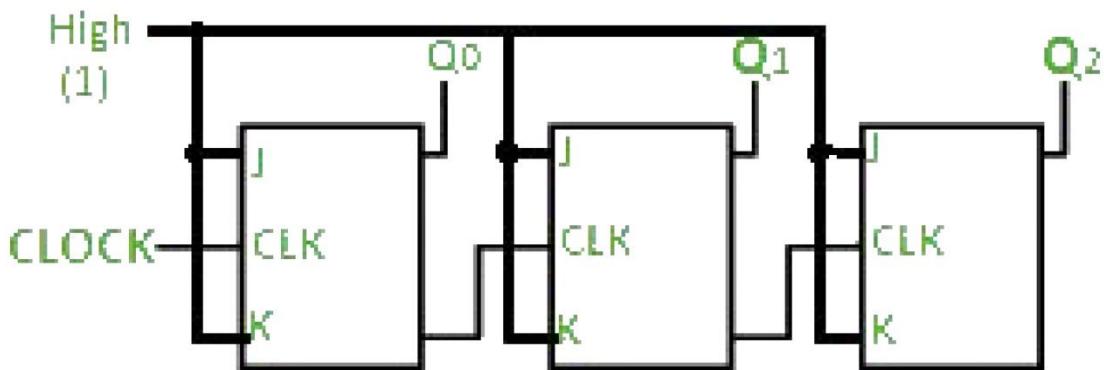


Fig:- 3-bit Ripple counter

Truth table:

Counter State	Q_2	Q_1	Q_0
0	0	0	0
1	0	0	1
2	0	1	0
3	0	1	1
4	1	0	0
5	1	0	1
6	1	1	0
7	1	1	1

6. Write short notes on: (any FOUR)

a) ASCII Code

➤ ASCII (American Standard Code for Information Interchange) is a standard character encoding used in telecommunication. The ASCII pronounced 'ask-ee', is strictly a seven-bit code based on the English alphabet. ASCII codes are used to represent alphanumeric data. The code was first published as a standard in 1967. it was subsequently updated and published as ANSI X3.4-1968, then as ANSI X3.4-1977, and finally as ANSI X3.4-1986. Since it is a seven-bit code, it can at the most represent 128 characters. it currently defines 95 printable characters including 26 upper case letters (A to Z), 26 lower case letters, 10 numerals (0 to 9), and 33 special characters including mathematical symbols, punctuation marks and space characters. They represent text in, telecommunications equipment, and devices. These include numbers, upper and lowercase English letters, functions, punctuation symbols, and some other symbols. In total, there are 256 ASCII characters, and can be broadly divided into three categories:

- ASCII control characters (0-31 and 127)
- ASCII printable characters (32-126) (most commonly referred to)
- Extended ASCII characters (128-255)

b) DTL Logic Family

➤ Diode-transistor logic (DTL) was one of the most popular manufactured circuits. The basic DTL circuit is a diode AND with a transistor inverter. The figure below shows the basic DTL configuration which is a NAND circuit.

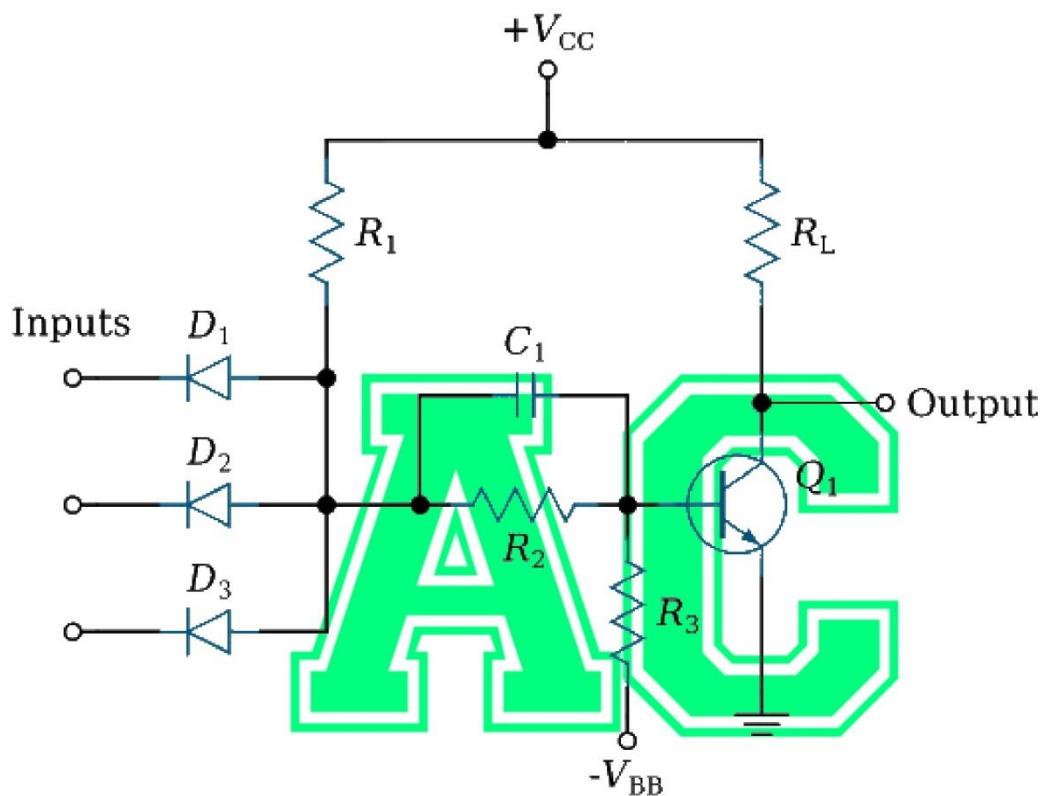


Fig:- Basic diode-transistor logic circuit.(NAND)

The divider network, consisting of R_1 , R_2 , and R_3 , is designed such that if all the input voltages are HIGH (logic 1), the base of Q_1 is relatively positive and Q_1 is on. The output voltage at the collector of Q_1 is almost zero. If any of the inputs now swing to 0 volt, that particular diode conducts. This applies a relatively negative voltage to the base of the transistor which is now cutoff. The collector tends to rise to the supply voltage $+V_{CC}$. Capacitor C_1 is used to provide an overdrive current during switching time. This reduces the switching time to some extent.

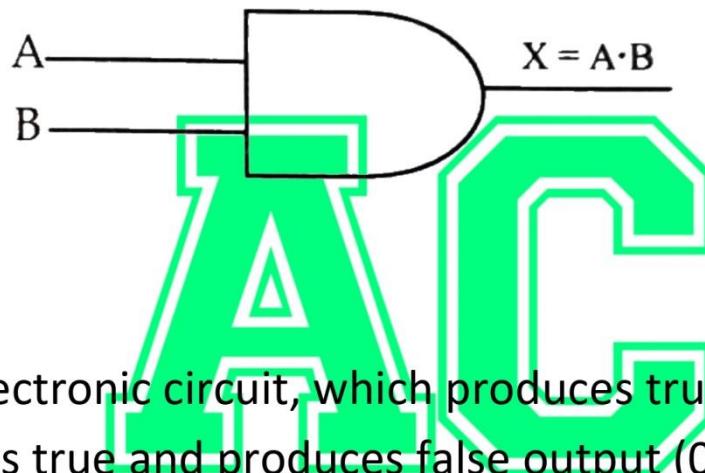
C) AND & OR Gates

❖AND Gate

➤ AND gate is an electronic circuit, which produces true output (1) only when all the inputs are true and produces false output (0) when at least one input is false.

Logical expression: $X = A \cdot B$, where A and B are inputs, X is output and represents AND operation.

Gate symbol:

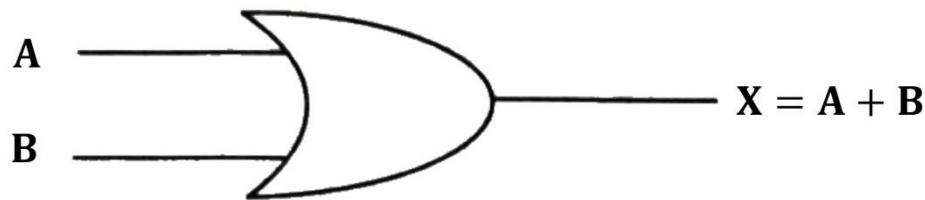


❖OR Gate

➤ OR gate is an electronic circuit, which produces true output (1) when at least one input is true and produces false output (0) only when all the inputs are false.

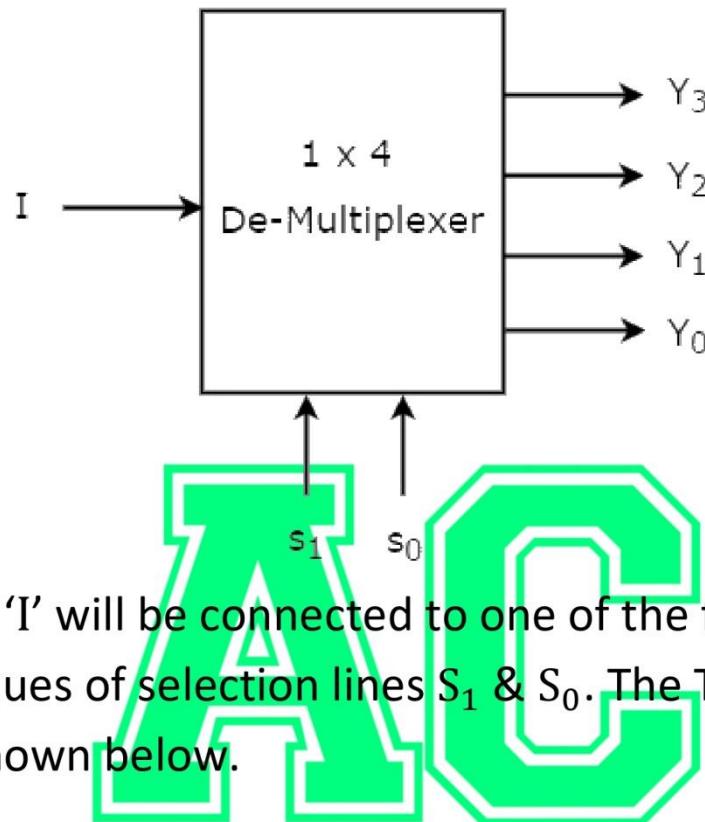
Logical expression: $X = A + B$, where A and B are inputs, X is output and + represents OR operation.

Gate symbol:



d) 1:4 De-multiplexer

- 1×4 De-Multiplexer has one input I, two selection lines, S_1 & S_0 and four outputs Y_3 , Y_2 , Y_1 & Y_0 . The **block diagram** of 1×4 De-Multiplexer is shown in the following figure.



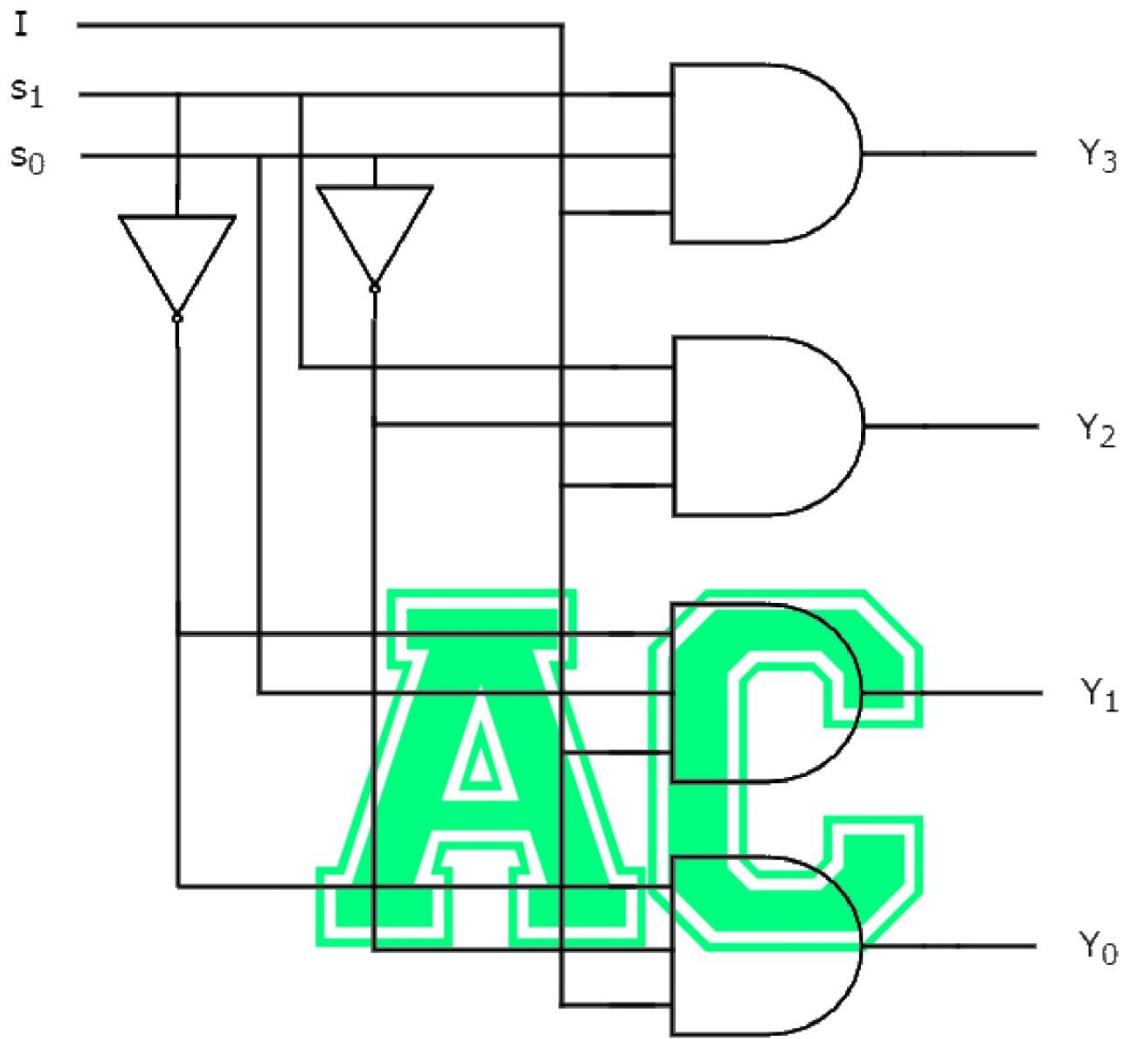
The single input 'I' will be connected to one of the four outputs, Y_3 , to Y_0 based on the values of selection lines S_1 & S_0 . The Truth table of 1x4 De-Multiplexer is shown below.

Selection Inputs		Outputs			
S_1	S_0	Y_3	Y_2	Y_1	Y_0
0	0	0	0	0	I
0	1	0	0	I	0
1	0	0	I	0	0
1	1	I	0	0	0

From the above Truth table, we can directly write the **Boolean functions** for each output as

$$Y_3 = S_1 S_0 I, \quad Y_2 = S_1 S'_0 I, \quad Y_1 = S'_1 S_0 I, \quad Y_0 = S'_1 S'_0 I$$

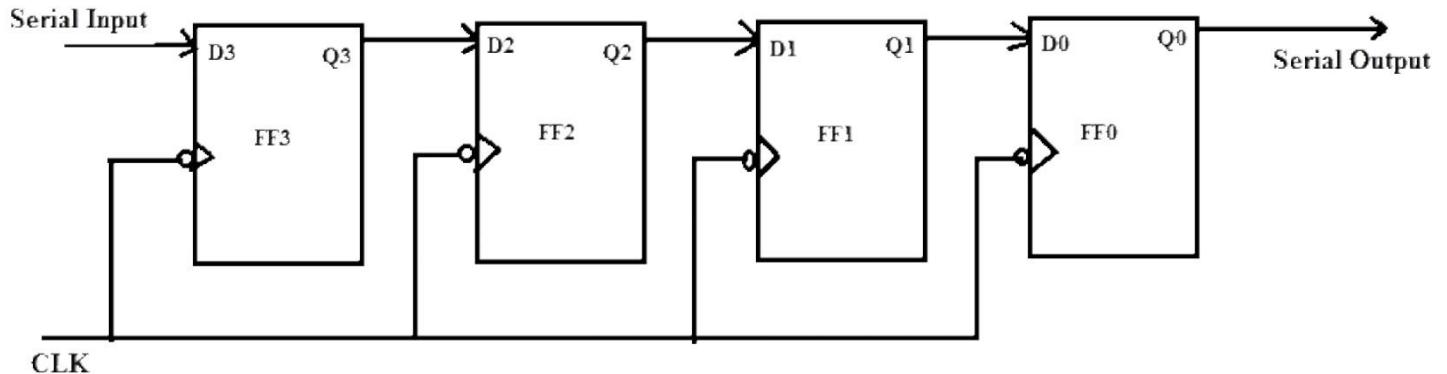
We can implement these Boolean functions using Inverters & 3-input AND gates. The **circuit diagram** of 1x4 De-Multiplexer is shown in the following figure.



e) SISO Shift Register

- The term “SISO” stands for “Serial-In Serial-Out”. The SISO shift register circuit accepts serial data on its input pin and shifts it out serially on its output pin. The number of bits that can be shifted out before the next bit arrives depends on the speed of the clock signal that controls the operation of the shift register. This type of shift register can be used as a buffer between two asynchronous devices that communicate with each other using signals with different frequencies or phases.

The SISO shift register block diagram is shown below which includes 3-D flip-flops. The connection of these FFs can be done by connecting the one Flip Flops output to the next flip flop input. So these FFs are synchronous through each other because the equal CLK signal is applied in each Flip Flop.



This shift register includes simply three connections the SI (serial input), SO (serial output), and CLK (clock signal). Here, the SI determines the input enters into the left-hand side flip-flop, the SO is the output taken from the right-hand side flip-flop & the sequencing CLK signal.

-The End-

Digital Logic --- (DCOM/IT) 3rdSem

(2080) Question Paper Solution.

Compile by © Arjun Chy

Website :- www.arjun00.com.np

Facebook :- www.facebook.com/Arjun00.com.np

1. a) Define signal. Differentiate between analog and digital signal with suitable example.

- Anything that carries information can be called as signal. It can also be defined as a physical quantity that varies with time, temperature, pressure or with any independent variables such as speech signal or video signal.

Analog Signals	Digital Signals
An analog signal is a continuous signal that represents physical measurements.	Digital signals are time separated signals which are generated using digital modulation.
It is denoted by sine waves	It is denoted by square waves
It uses a continuous range of values that help you to represent information.	Digital signal uses discrete 0 and 1 to represent information.
The analog signal bandwidth is low	The digital signal bandwidth is high.
Analog hardware never offers flexible implementation.	Digital hardware offers flexibility in implementation.
Examples :- Temperature sensors, FM radio signals, Photocells, Light sensor, Resistive touch screen etc.	Examples :- Computers, CDs, DVDs etc.

b. Convert the following number system:

i) $(10111.01)_2 = (?)_2$

➤ Solⁿ

Converting into base 10,

$$= 1 \times 2^4 + 0 \times 2^3 + 1 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 + 0 \times 2^{-1} + 1 \times 2^{-2}$$

$$= 16 + 0 + 4 + 2 + 1 + 0 + \frac{1}{4}$$

$$= 23 + 0.25$$

$$= 23.25$$

$$\therefore (10111.01)_2 \rightarrow (23.25)_{10}$$

ii) $(3471)_8 = (?)_{16}$

➤ Solⁿ



First converted to Binary, Then it is converted to Hexadecimal.

Octal: 3 4 7 1

BN: 011 100 111 001

BN $\Rightarrow (011100111001)_2$

Taking pairs of four bit from last.

BN: 0111 0011 1001

Hex: 7 3 9

$(739)_{16}$

$$\therefore (3471)_8 = (739)_{16}$$

iii) $(BCDE \cdot 4A)_{16} = (?)_8$

➤ Solⁿ

First converted to Binary, Then it is converted to octal.

HD: B(11) C(12) D(13) E(14). 4 A(10)

BN: 1011 1100 1101 1110 0100 1010

(8421)

$$BN \Rightarrow (1011110011011110.01001010)_2$$



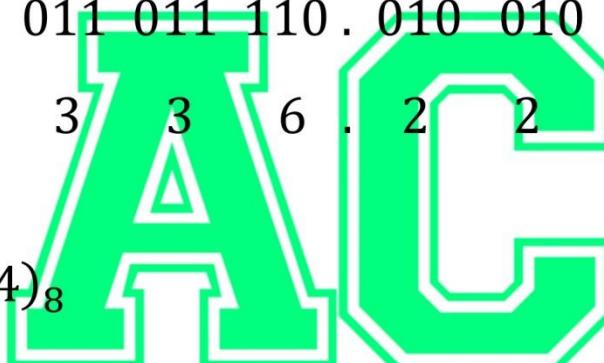
Taking pairs of 3 bit.

BN: 001 011 110 011 011 110 . 010 010 100

ON: 1 3 6 3 3 6 . 2 2 4

(421)

$$(136336.224)_8$$

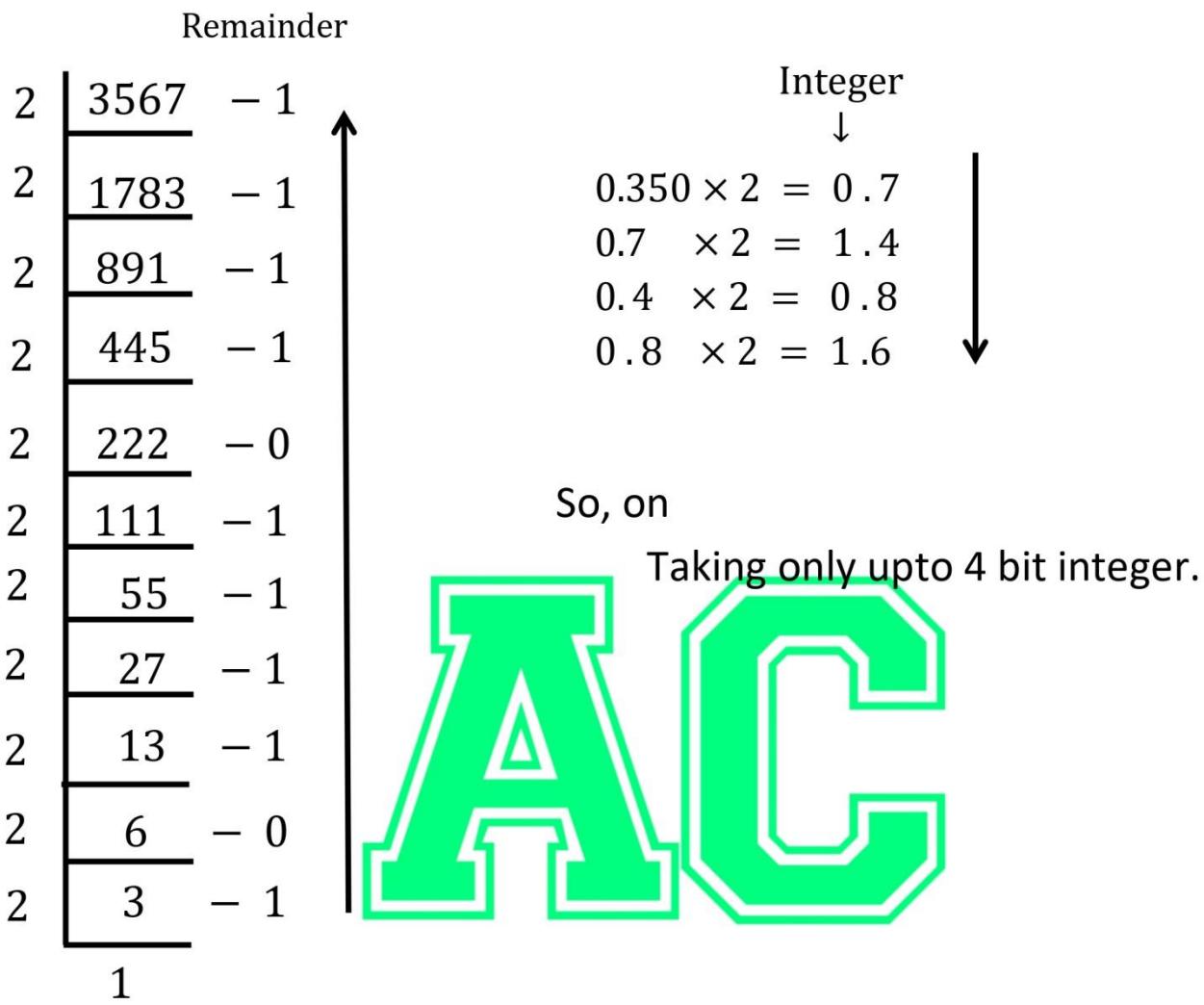


Hence,

$$\therefore (BCDE \cdot 4A)_{16} = (136336.224)_8$$

iv) $(3567.350)_{10} = (?)_2$

➤ Solⁿ



$\therefore (3567.350)_{10} = (110111101111.0101)_2$

C. Perform the following operation:

i) Multiply: $(11101.11 \times 101)_2$

➤ Solⁿ

$$\begin{array}{r} 11101.11 \\ \times 101 \\ \hline 1110111 \\ + 0000000 \\ \hline 01110111 \end{array} \quad \Leftarrow \text{1st intermediate Sum}$$
$$\begin{array}{r} + 1110111 \\ \hline 10010100.11 \end{array} \quad \begin{array}{l} \Leftarrow \text{Final Sum} \\ \text{AC} \end{array}$$

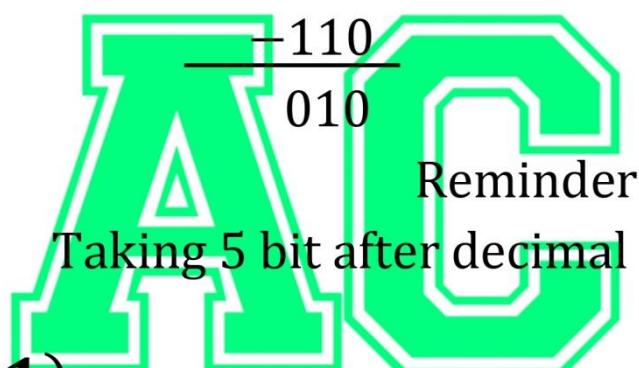
Since,

The decimal point is placed on second places from last significant bit. Because, the binary number 11101.11, decimal point is 2 places from LSB.

ii) Divide: $(110100.110 / 110)_2$

➤ Solⁿ

$$\begin{array}{r}
 110 \overline{)110100.11000} \quad | \quad 1000.11001 \\
 -110 \\
 \hline
 000100 \quad 1 \\
 -110 \\
 \hline
 0111 \\
 -110 \\
 \hline
 001000
 \end{array}$$



∴ $(1000.11001)_2$

2. a) Subtract the following $(11001100)_2$ from $(11110000)_2$ using 2's complement.

➤ Solⁿ

X: 11110000

Y: 11001100

2's complement of Y:

$$\begin{array}{r} 11001100 \\ 00110011 \\ \hline +1 \\ \hline 00110100 \end{array}$$

2's complement of Y = 00110100

Adding with X:

$$\begin{array}{r} 11110000 \\ + 00110100 \\ \hline \text{Carry} \rightarrow 100100100 \end{array}$$

Since, there is additional carry so, discard it.

$$\text{Result} = 00100100 \Rightarrow (100100)_2$$

Hence,

$$\therefore (11001100)_2 - (11110000)_2 = (100100)_2$$

b) Realize basic gate using NAND gate only with clear diagram and truth table. Also, state and prove De-Morgan's Theorem in brief.

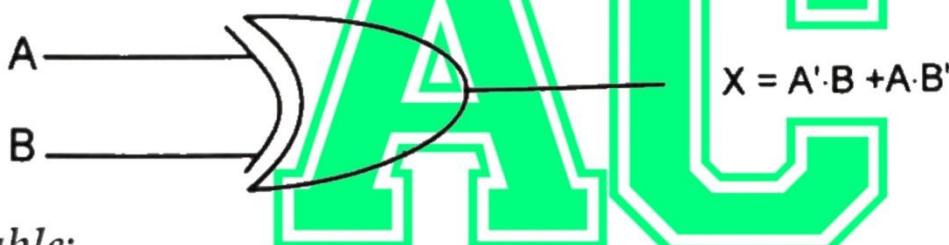
- 1st part : See the solution 2079 R/B of Q. no 3(b) on page on 38
- 2nd part : See the solution 2079 R/B of Q. no 2(a) on page on 32

c) Explain XOR and NOR gate with truth table and symbol.

- XOR gate is an electronic circuit, which produces true output (1) when the inputs are different and produces false (0) output only when both the inputs are same.

Algebraic expression: $X = A' \cdot B + A \cdot B'$, where A and B are inputs, X is output.

Gate symbol:



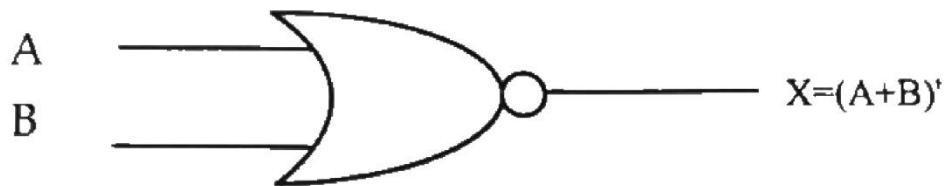
Truth table:

Inputs						Output
A	B	A'	B'	$A' \cdot B$	$A \cdot B'$	$X = A \cdot B' + A' \cdot B$
0	0	1	1	0	0	0
0	1	1	0	1	0	1
1	0	0	1	0	1	1
1	1	0	0	0	0	0

- NOR gate is an electronic circuit, which produces true output (1) only when all the inputs are false and produces false (0) output when any one input is true. It is the combination of NOT gate and OR gate.

Algebraic expression: $X = (A + B)'$, where A and B are inputs, X is output.

Gate symbol:



Truth table:

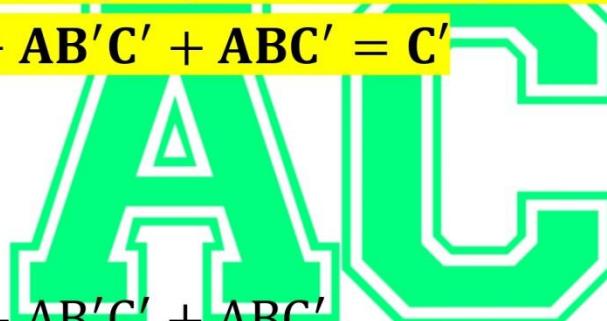
Inputs			Output
A	B	A+B	X = (A + B)'
0	0	0	1
0	1	1	0
1	0	1	0
1	1	1	0

3. a) Simplify the following expression using Boolean algebra:

i) $A'B'C' + A'BC' + AB'C' + ABC' = C'$

➤ Solⁿ

LHS



$$A'B'C' + A'BC' + AB'C' + ABC'$$

$$= A'C'(B' + B) + AC'(B' + B) \quad \{ \because B' + B = 1 \}$$

$$= A'C'.1 + AC'1$$

$$= A'C' + AC'$$

$$= C'(A + A') \quad \{ \because A + A' = 1 \}$$

$$= C'.1$$

$$\therefore C'$$

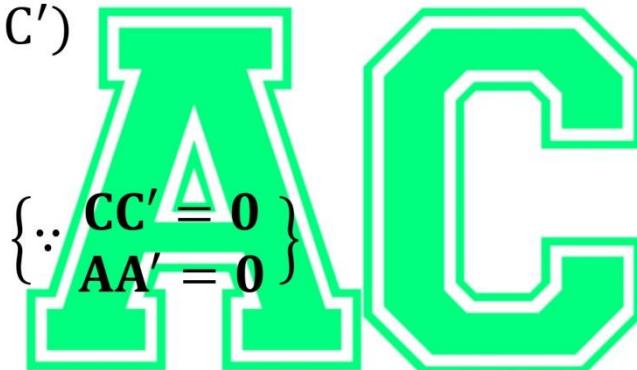
RHS, Proved.

ii) $A(A' + C)(A'B + C)(A'BC + C') = 0$

➤ Solⁿ

LHS

$$\begin{aligned}
 & A(A' + C)(A'B + C)(A'BC + C') \\
 &= (AA' + AC)(A'B + C)(A'BC + C') \\
 &\quad \{\because AA' = 0\} \\
 &= (0 + AC)(A'B + C)(A'BC + C') \\
 &= (AA'. BC + AC . C)(A'BC + C') \\
 &\quad \{\because C . C = C\} \\
 &= (0 + AC)(A'BC + C') \\
 &= AA'. BCC + ACC' \\
 &= 0 . BC + A . 0 \\
 &= 0 \\
 \therefore & 0
 \end{aligned}$$



RHS, Proved.

3. a) Simplify the following expression using k-map.

i) $\sum F(A, B, C, D) = \pi m(2, 3, 4, 5, 7, 10, 11, 14) + \sum d(0, 1, 6, 15)$

Draw logic diagram.

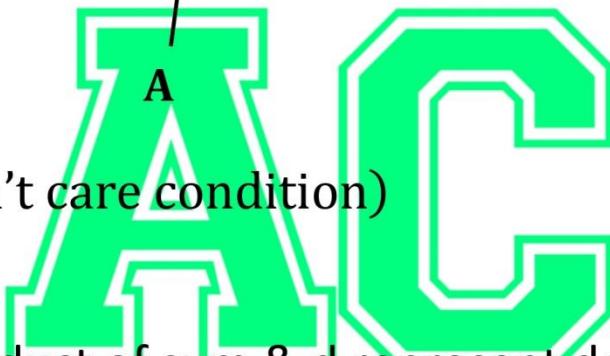
➤ $\pi m(2, 3, 4, 5, 7, 10, 11, 14) + \sum d(0, 1, 6, 15)$

Truth table of ABCD, K – map

		CD AB	00	01	11	10	
		00	0	1	3	2	
00	00	X		X	0	0	
	01	4	5	7	6	X	
11	11	12	13	15	X	0	
	10	8	9	11	10	0	0

\bar{C}

$X \rightarrow$ (don't care condition)



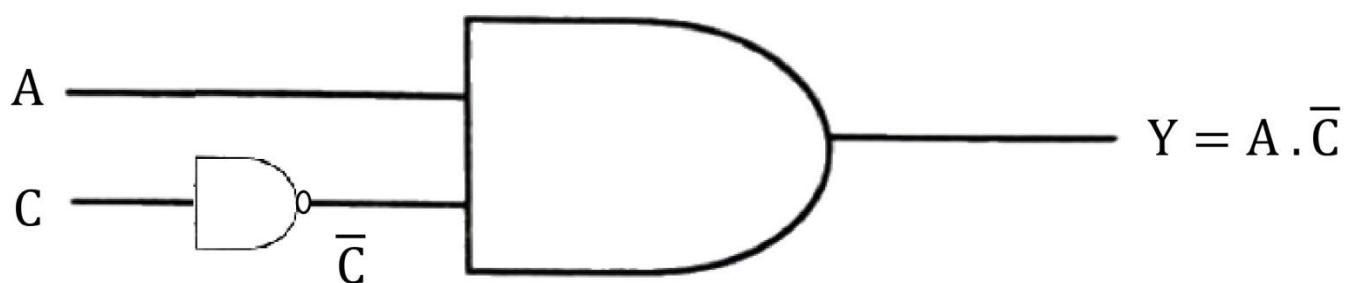
Here,

π represent Product of sum & d represent don't care condition.

From table,

Simplified POS expression is $Y = A \cdot \bar{C}$

Logic diagram,

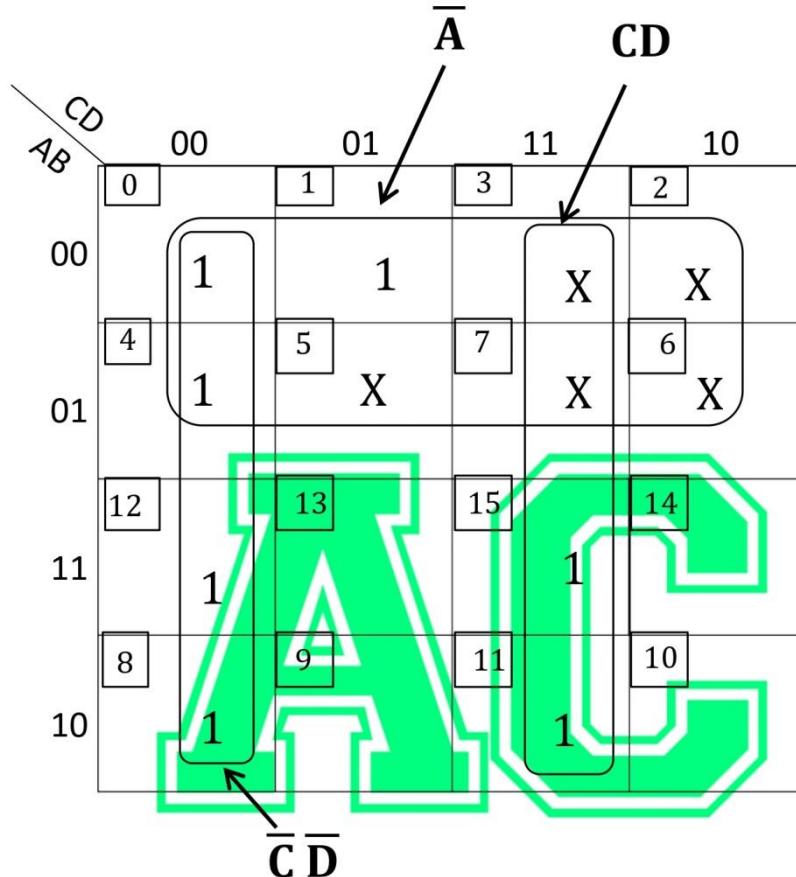


$$\text{ii.) } \sum F(A, B, C, D) = \sum M(0, 1, 4, 8, 11, 12, 15) + \sum d(2, 3, 5, 6, 7)$$

Draw logic diagram.

Here,

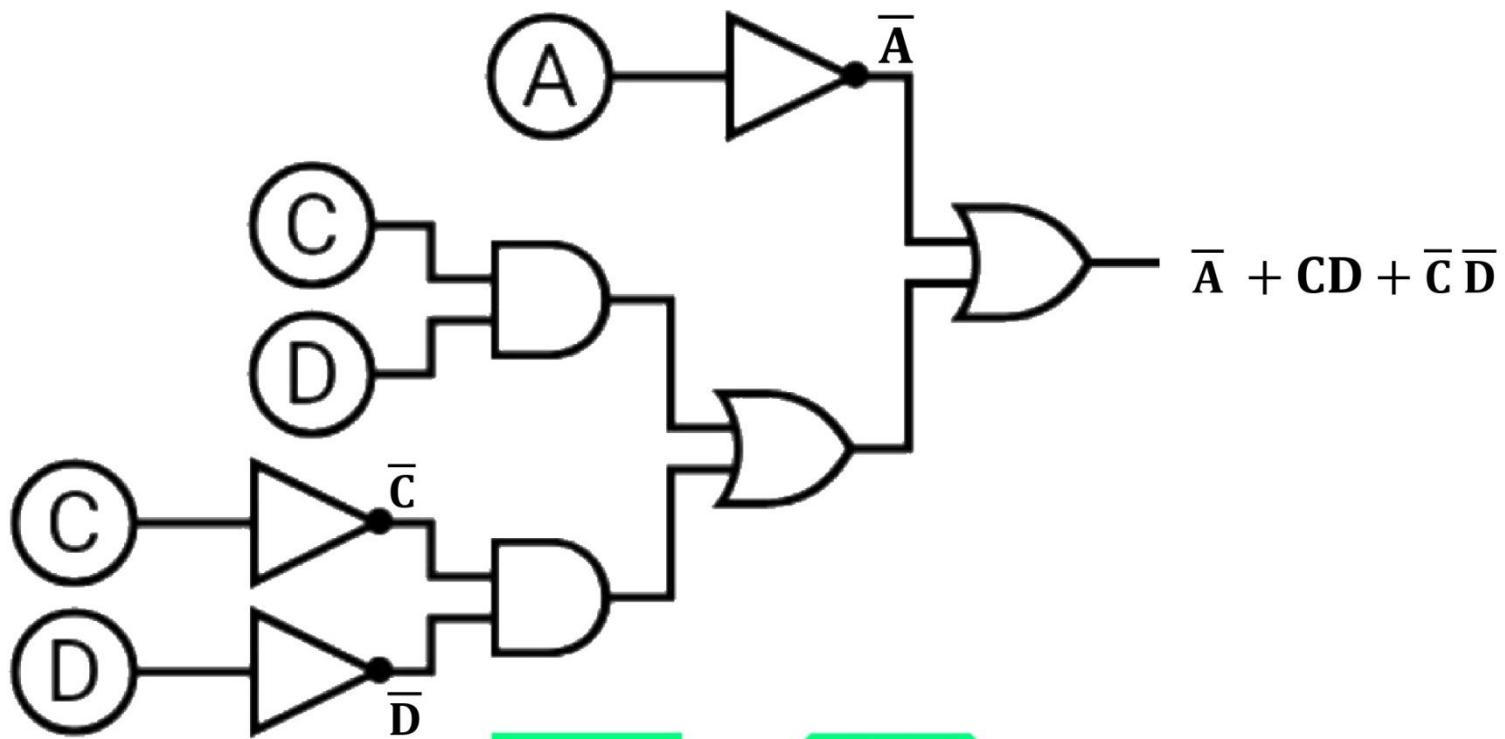
$\sum M$ represent sum of Product terms & d represent don't care condition.



Simplified SOP,

$$F(A, B, C, D) = \bar{A} + CD + \bar{C} \bar{D}$$

Logic diagram,



4. a) Define encoder. Explain the decimal to binary encoder with suitable diagram and truth table.

➤ An encoder is a combinational circuit that performs the inverse operation of a decoder. An encoder has 2^n (or fewer) input lines and n output lines.

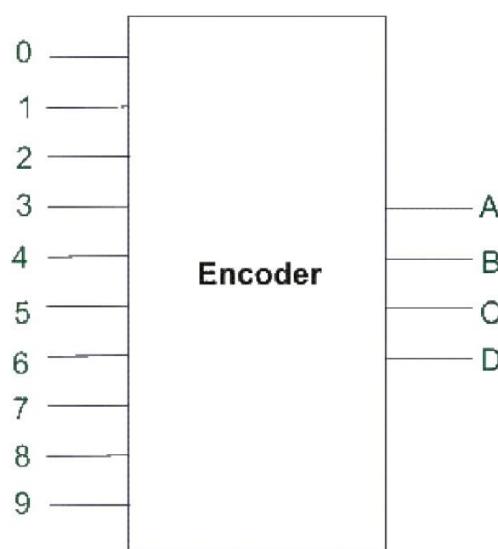


Fig:-Decimal to binary encoder block diagram

- Encoders are used as code converters in computer systems. These are available as IC's in the market. To convert a decimal number into binary a Decimal to BCD Encoder is used. In the BCD system, the decimal number is represented as the four-digit binary. It can convert the decimal numbers from 0 to 9 into the binary stream.

The encoder is a combinational logic circuit. The reverse of the encoder is a decoder that performs the reverse action. The truth table of Decimal to BCD encoder is given below.

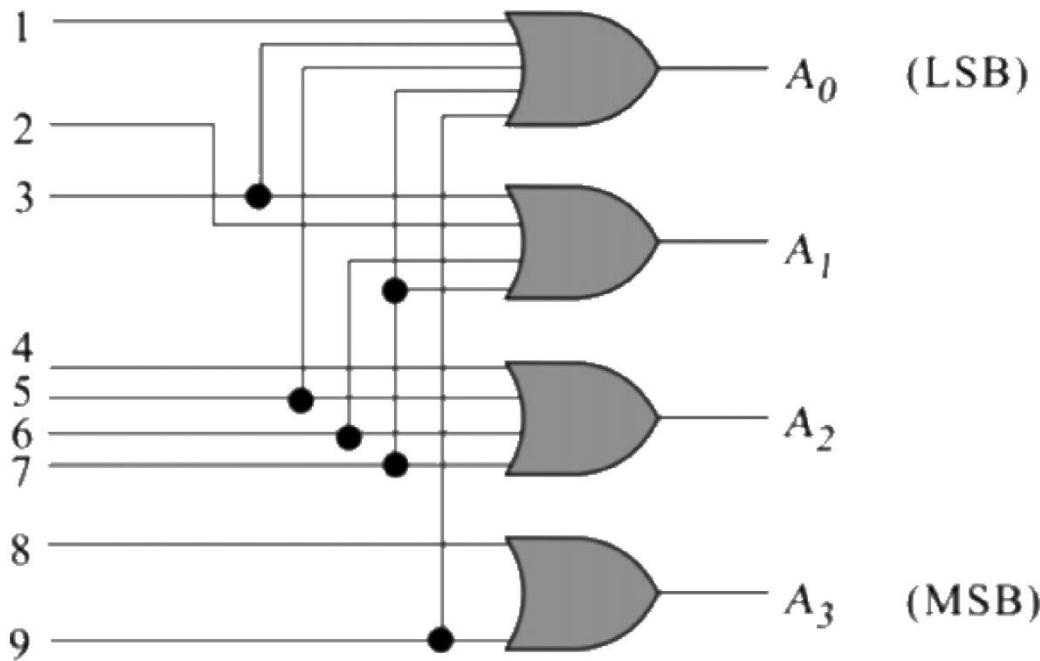
Decimal Digit	BCD Code			
	A3	A2	A1	A0
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0
9	1	0	0	1

From the truth table above form the equations for the words A3, A2, A1, A0.

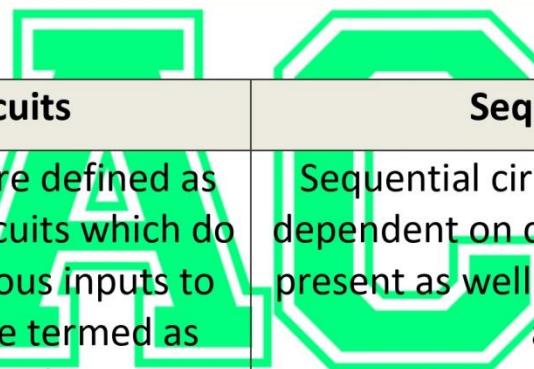
Thus the logical equations are as below:-

$$A3 = 8+9 : A2 = 4+5+6+7 : A1 = 2+3+6+7 : A0= 1+3+5+7+9$$

Now, considering the logic equations above, form the combinational circuit with OR gates.

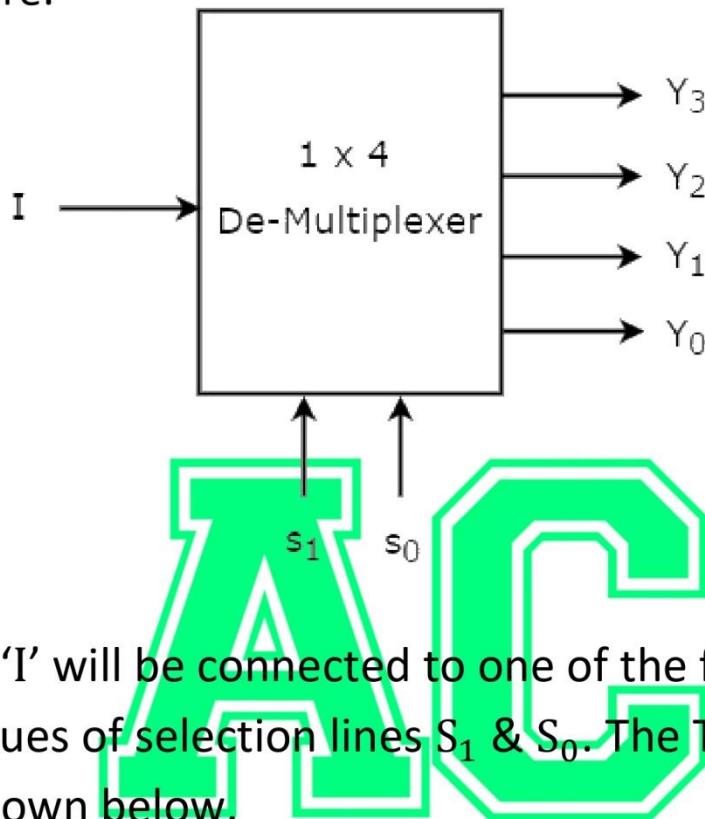


b. Differentiate between combinational and sequential circuit with example.



c. Design 1:4 De-multiplexer with clear circuit diagram and truth table in brief.

- 1x4 De-Multiplexer has one input I, two selection lines, S_1 & S_0 and four outputs Y_3 , Y_2 , Y_1 & Y_0 . The **block diagram** of 1x4 De-Multiplexer is shown in the following figure.



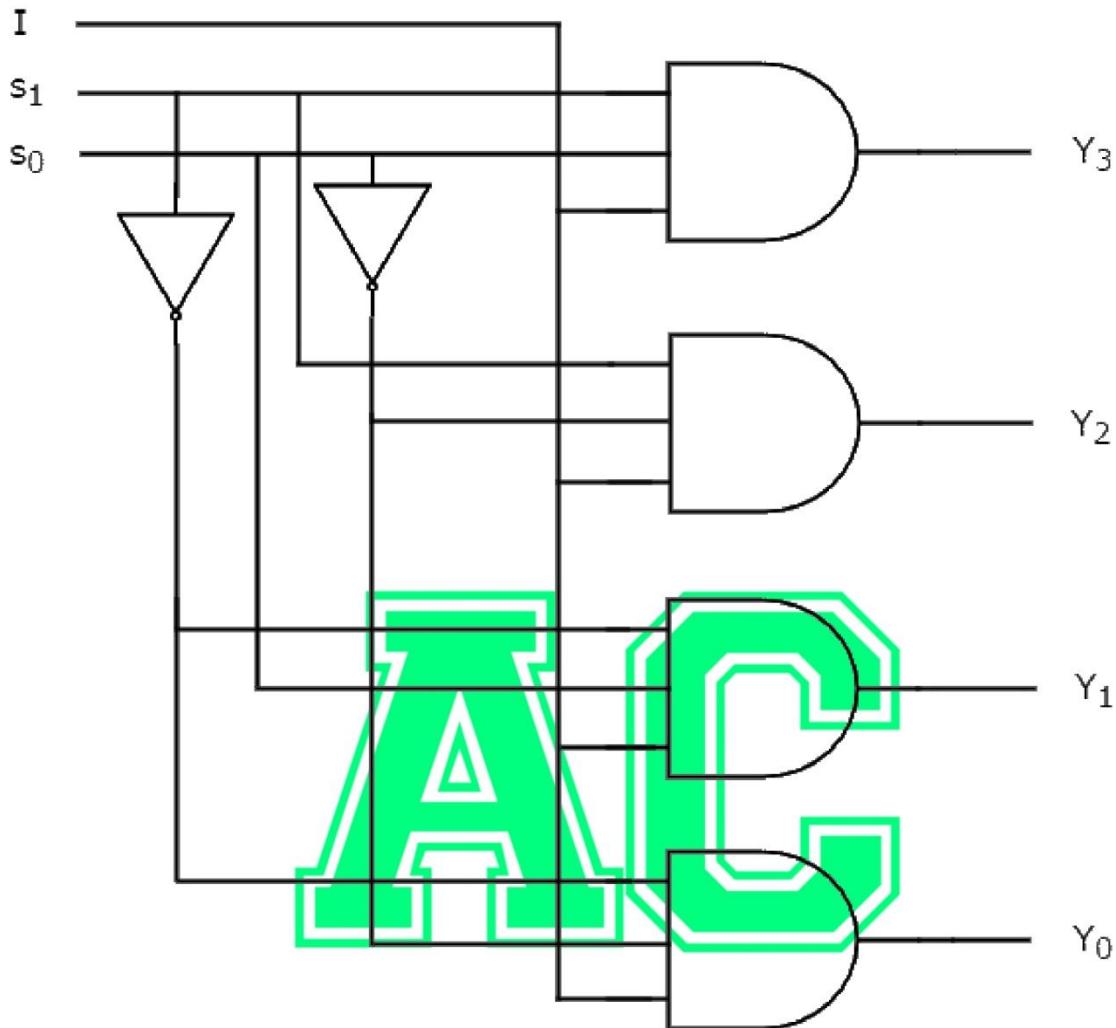
The single input 'I' will be connected to one of the four outputs, Y_3 , to Y_0 based on the values of selection lines S_1 & S_0 . The Truth table of 1x4 De-Multiplexer is shown below.

Selection Inputs		Outputs			
S_1	S_0	Y_3	Y_2	Y_1	Y_0
0	0	0	0	0	I
0	1	0	0	I	0
1	0	0	I	0	0
1	1	I	0	0	0

From the above Truth table, we can directly write the **Boolean functions** for each output as

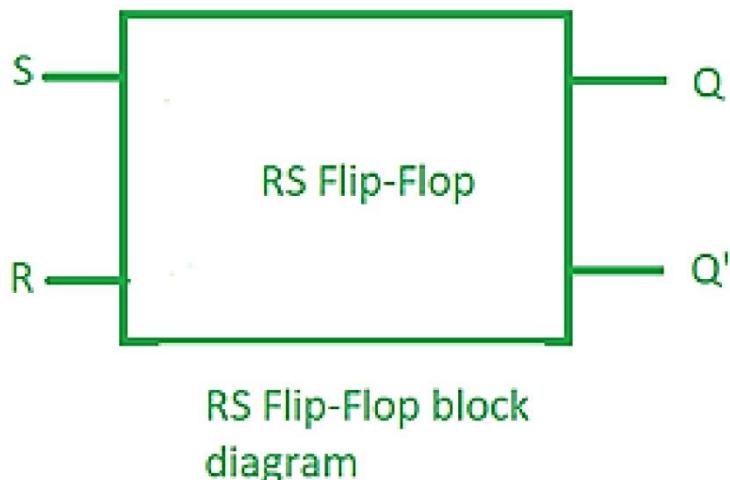
$$Y_3 = S_1 S_0 I, \quad Y_2 = S_1 S'_0 I, \quad Y_1 = S'_1 S_0 I, \quad Y_0 = S'_1 S'_0 I$$

We can implement these Boolean functions using Inverters & 3-input AND gates. The **circuit diagram** of 1x4 De-Multiplexer is shown in the following figure.



5. a) Design RS flip-flop with necessary diagram. Write the advantages of JK flip-flop.

- The RS flip-flop is used to store binary information (i.e. 0 or 1). It consists of two inputs, SET and RESET. In RS flip-flop 'R' Stands for RESET and 'S' stands for SET. The flip-flop keeps its present state even when one or both inputs are deactivated. The flip-flop enters the '0' state when the RESET input is activated, and the '1' state when the SET input is activated.



➤ The block diagram of the RS flip-flop is shown above. Since RS flip-flops are used for storing binary information it is used in simple digital applications like data registers and memory cells that require binary storage. Nevertheless, more sophisticated flip-flop designs, such as the D flip-flop or JK flip-flop, are frequently used over the RS flip-flop for their increased reliability and versatility in complex digital systems due to restrictions in handling specific input conditions.



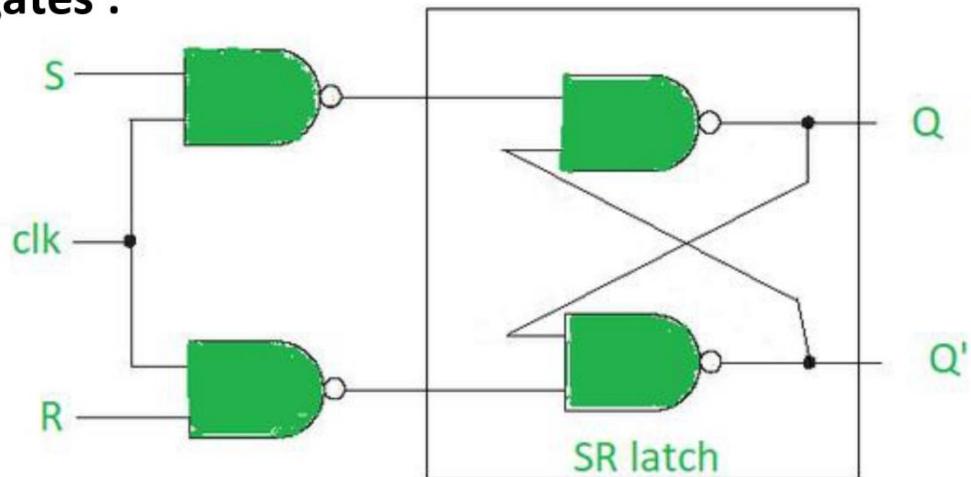
Construction and Working of RS flip flop

➤ RS flip can be constructed using basic logic gates such as NAND gates or NOR gates. Below shown a RS flip-flop constructed using a NAND gate same we can construct a RS flip flop using a NOR gate.

Working of RS flip-flop depends on its inputs.

- In initial state output can in any state SET or RESET.
- In SET , S=1 and R=0.
- Conversely in RESET , S=0 and R=1.
- When R=0 and S=0 it hold the current state
- When R=1 and S=1 it falls under Undefined / Forbidden state

RS flip-flop Using NAND gates :



Truth Table For RS Flip-flop :-

RS flip-flop

Sl no.	Clock	S	R	Q _{n+1}
1	0	X	X	Q _n
2	1	0	0	No change
3	1	0	1	Q _n
4	1	1	0	1
5	1	1	1	Undefined / Forbidden state

➤ There are several advantages to using a JK flip-flop. Some of them are listed below: -

- **Toggle capability:** It has a toggle capability, which means that it can be used to create a circuit that toggles between two states.

- **No invalid states:** Unlike the SR flip-flop, the JK flip-flop does not have any invalid states.
- **Reduced race conditions:** It is less susceptible to race conditions than the SR flip-flop, which can lead to more stable circuit operation.
- **Bi-stable operation:** Like the SR flip-flop, the JK flip-flop has a bi-stable operation, which means that it can hold a state indefinitely until it is changed by an input signal.

b. Define shift register. Explain the operation of ripple counter with clear diagram.

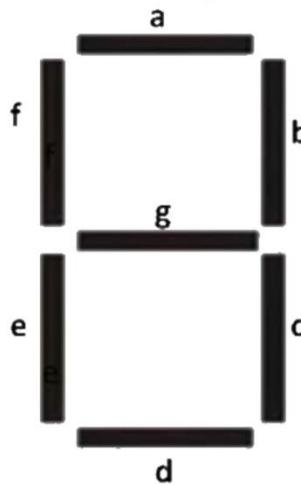
- A register capable of shifting its binary information either to the right or to the left is called a **shift register**. The logical configuration of a shift register consists of a chain of flip-flops connected in cascade, with the output of one flip-flop connected to the input of the next flip-flop.
- *See the solution of 2079 R/B of Q.no 5(b) on page no. 45*



6. Write short notes on: (any FOUR)

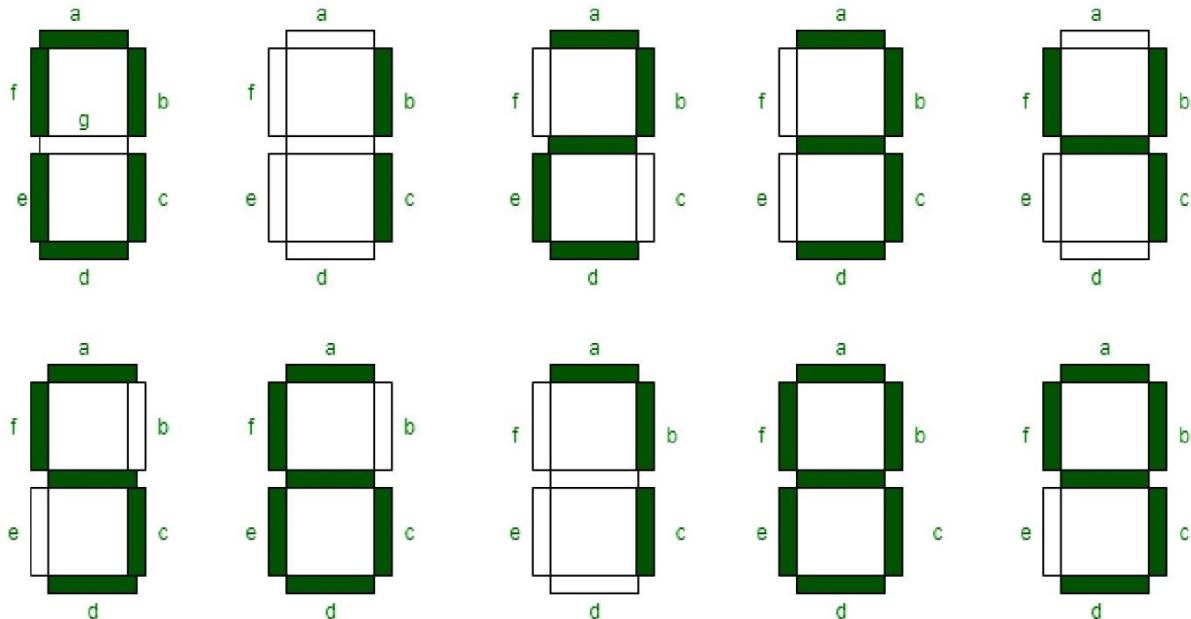
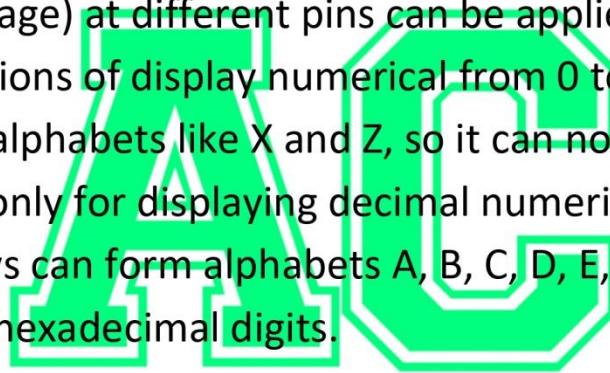
a) 7 segments display

- Seven segment displays are the output display device that provides a way to display information in the form of images or text or decimal numbers which is an alternative to the more complex dot matrix displays. It is widely used in digital clocks, basic calculators, electronic meters, and other electronic devices that display numerical information. It consists of seven segments of light-emitting diodes (LEDs) which are assembled like numerical 8.



Working of Seven Segment Displays:

- The number 8 is displayed when the power is given to all the segments and if you disconnect the power for 'g', then it displays the number 0. In a seven-segment display, power (or voltage) at different pins can be applied at the same time, so we can form combinations of display numerical from 0 to 9. Since seven-segment displays can not form alphabets like X and Z, so it can not be used for the alphabet and they can be used only for displaying decimal numerical magnitudes. However, seven-segment displays can form alphabets A, B, C, D, E, and F, so they can also be used for representing hexadecimal digits.

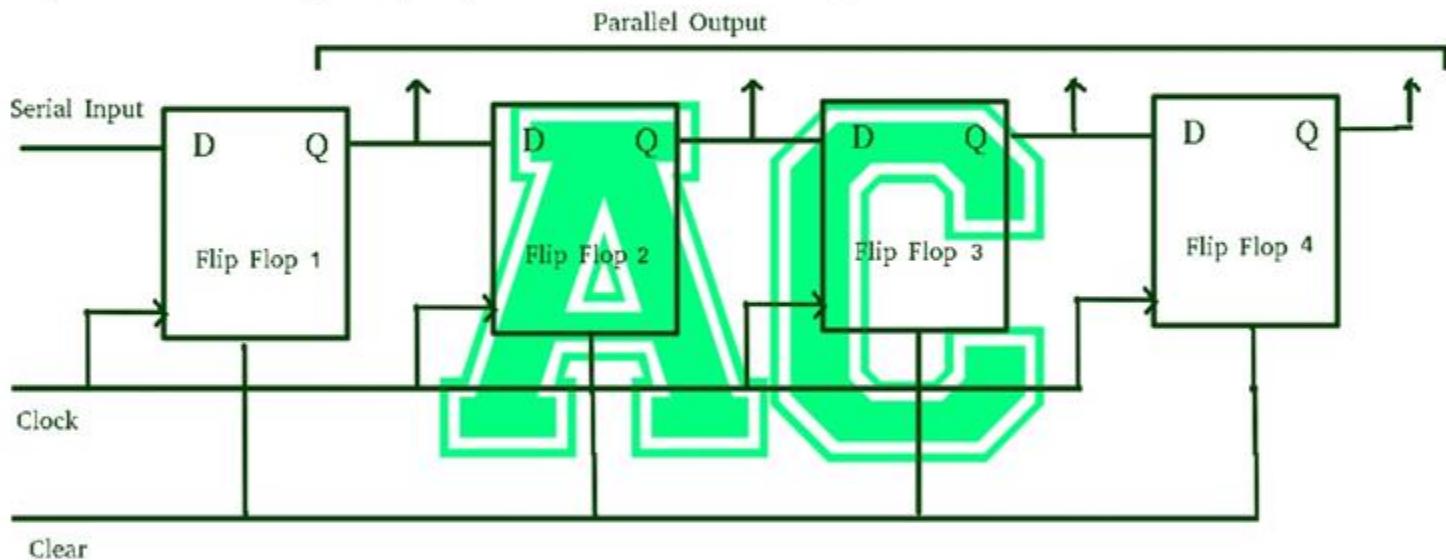


b) SIPO shift register

- A Serial-In Parallel-Out shift register is a sequential logic device that can store and shift data bits. It consists of a chain of flip-flops connected in series, with data input and output terminals. The data is shifted from one flip-flop to the next, either in a serial or parallel fashion, depending on the mode of operation.

Working Principle of SIPO

- The basic operation of a SIPO shift register involves the sequential transfer of data bits through a series of flip-flops. The register has one input line called the serial input (SI) and parallel output lines ($Q_0, Q_1, Q_2, \text{etc.}$) corresponding to each flip-flop. The clock signal (CLK) controls the shifting of data.

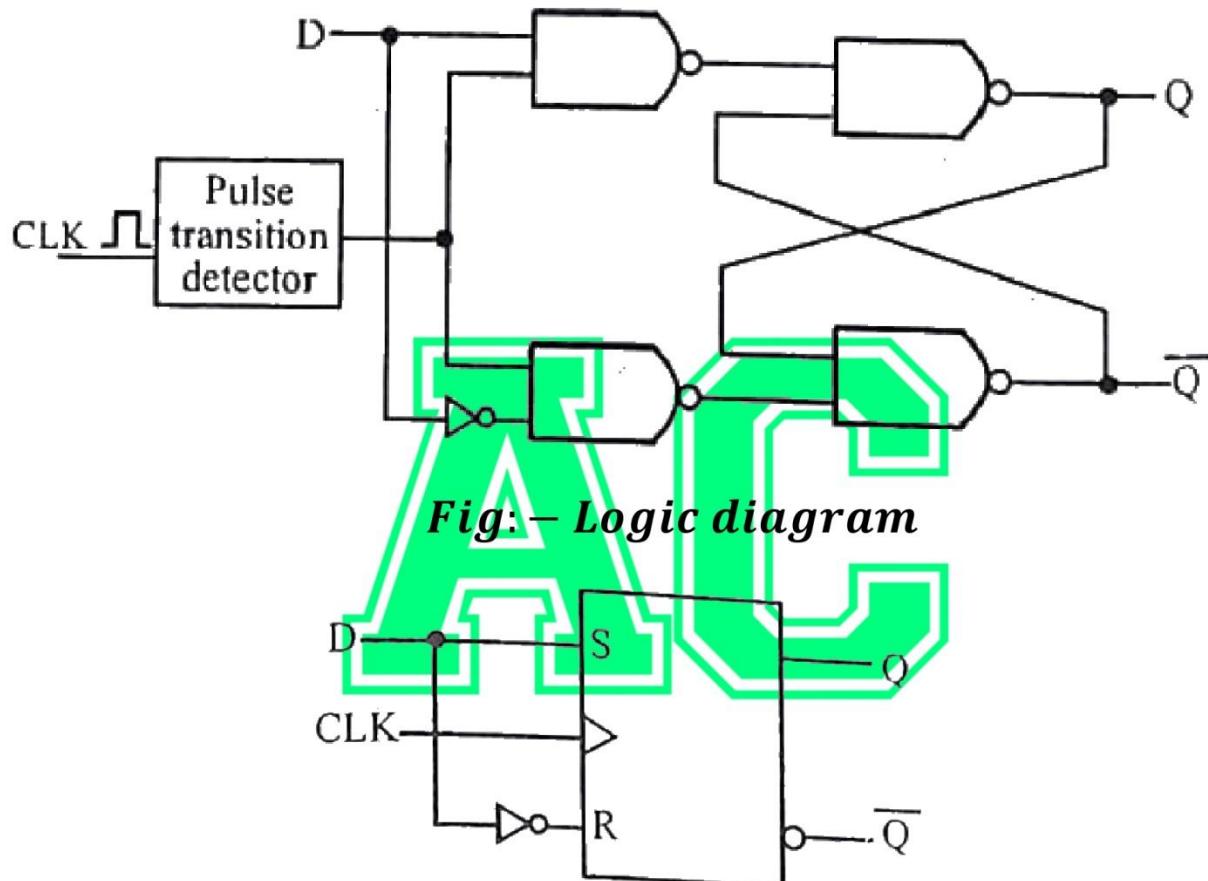


- **Serial Input:** Serial input (SI) is the entry point for the data into the shift register. The data bits are fed into the first flip-flop in the register. On each clock pulse, the data bit at the serial input is transferred to the first flip-flop and the existing data in the register shifts by one position.
- **Parallel Outputs:** The parallel outputs ($Q_0, Q_1, Q_2, \text{etc.}$) provide access to the stored data in the shift register. Each flip-flop's output is connected to a separate output line, enabling simultaneous access to the stored data bits.
- **Clock Signal:** The clock signal (CLK) synchronizes the shifting of data within the shift register. Typically, the clock edge triggers the transfer of data from one flip-flop to the next. The rising or falling edge of the clock signal can be used, depending on the specific implementation and requirements.

c. D flip-flop

➤ The D flip-flop is useful when a single data bit (1 or 0) is to be stored. It is also called delay flip-flop because it is used as delay circuits. The addition of an inverter to an SR flip-flop creates a basic D flip-flop.

The output of flip-flop sets when $D=1$ and resets when $D=0$.



INPUTS		OUTPUTS		COMMENTS
D	CLK	Q	\bar{Q}	
1	↑	1	0	SET
0	↑	0	1	RESET

Truth table: –

d. Half adder

➤ A half adder is a digital logic circuit that performs binary addition of two single-bit binary numbers. It has two inputs, A and B, and two outputs, SUM and CARRY. The SUM output is the least significant bit (LSB) of the result, while the CARRY output is the most significant bit (MSB) of the result, indicating whether there was a carry-over from the addition of the two inputs. The half adder can be implemented using basic gates such as XOR and AND gates.



Figure : Block diagram

Truth table: –

Inputs		Outputs	
A	B	Sum	Carry
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

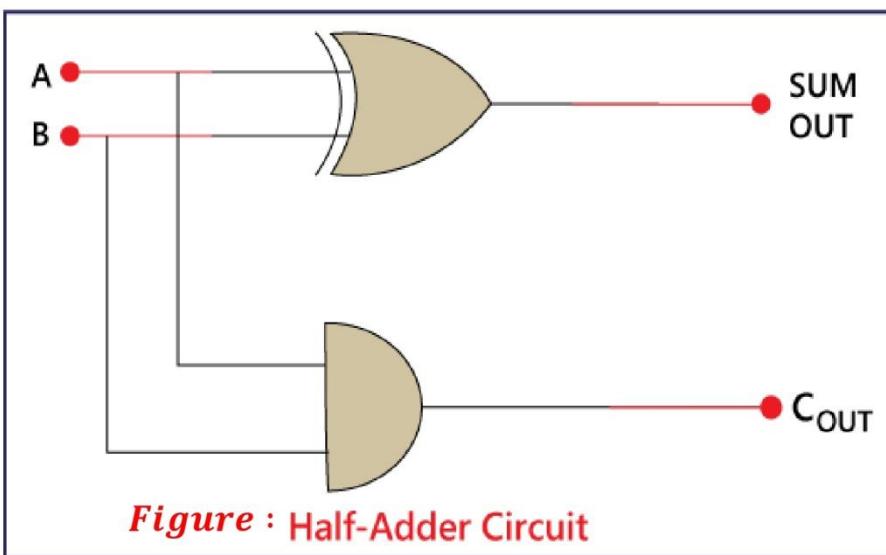
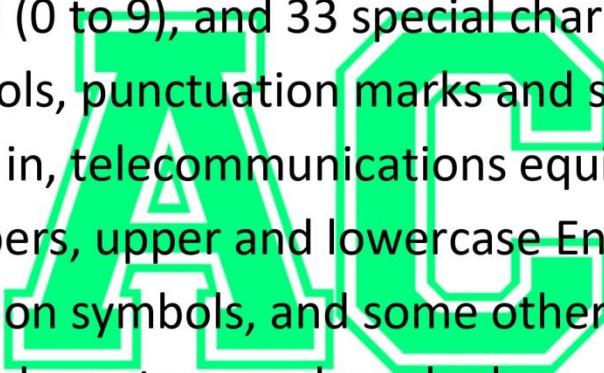


Figure : Half-Adder Circuit

e. ASCII Code

➤ ASCII (American Standard Code for Information Interchange) is a standard character encoding used in telecommunication. The ASCII pronounced ‘ask-ee’, is strictly a seven-bit code based on the English alphabet. ASCII codes are used to represent alphanumeric data. The code was first published as a standard in 1967. it was subsequently updated and published as ANSI X3.4-1968, then as ANSI X3.4-1977, and finally as ANSI X3.4-1986. Since it is a seven-bit code, it can at the most represent 128 characters. it currently defines 95 printable characters including 26 upper case letters (A to Z), 26 lower case letters, 10 numerals (0 to 9), and 33 special characters including mathematical symbols, punctuation marks and space characters. They represent text in, telecommunications equipment, and devices. These include numbers, upper and lowercase English letters, functions, punctuation symbols, and some other symbols. In total, there are 256 ASCII characters, and can be broadly divided into three categories: -



- ASCII control characters (0-31 and 127)
- ASCII printable characters (32-126) (most commonly referred to)
- Extended ASCII characters (128-255)

f. BCD code

➤ Binary Coded Decimal (BCD) (8421) Code:

- Each decimal digit is represented by a 4-bit binary number.
- In the BCD, with 4-bits we can represent 16 numbers (0000 to 1111).
- But in BCD code only first 10 numbers of these are used (0000 to 1001).
- The remaining six code combinations (1010 to 1111) are invalid in BCD.

Truth table: –

DECIMAL NUMBER	BCD
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111
8	1000
9	1001

-The End-