

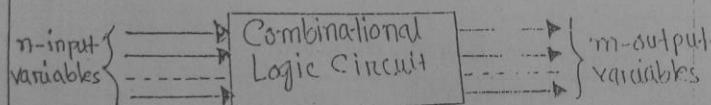
## Combinational logic

Babul chandra Roy

(Q1750-322800)

What is combinational and sequential logic circuit?

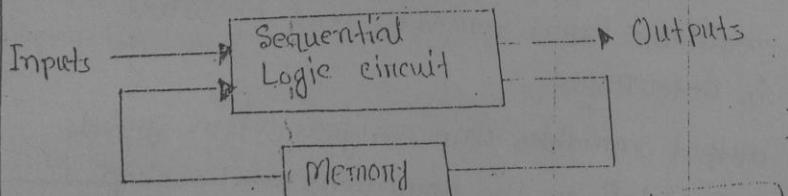
Ans: Combinational logic circuit: The present output depends on the present input values and ~~depends~~ depends on the previous one's are called combinational logic circuit. It consists of input variables, logic gates, and output variables. It has no feedback and memory element. Example of combinational logic circuit are Adder, Decoders, Multiplexers, Demultiplexers etc.



D.E

Block diagram of combinational logic circuit.

Sequential Logic circuit: The present output depends not only on the present input but also on the previous logic states of the output are called sequential logic circuit. It consists of a combinational logic circuit and one memory element. It has feedback system. Example of sequential logic circuit are S-R flip-flop, J-K flip-flop, D-flip-flop, T-flip-flop etc.



Block diagram of sequential logic circuit

\* Differentiate between combinational and sequential logic circuit.  
Ans: The difference between combinational and sequential logic circuit are given below:

Combinational logic circuit	Sequential logic circuit
1. Definition:	1. Definition:
2. Block diagram:	2. Block diagram:
3. It has no feedback system.	3. It has feedback system.
4. It has no memory element.	4. It consists of a combinational logic circuit and memory.
5. Example: Adder, subtractor, Decoder, Encoder, multiplexer, Demultiplexer etc.	5. Example: Flip Flop, Registers, counters etc.

\* What are the steps to design a combinational circuit?  
Ans: The steps to design a combinational circuit are given below:

- i) The problem is started.
- ii) The number of available input variables and required output variables is determined.
- iii) The input and output variables are assigned letter symbols.
- iv) The truth table that defines the required relationship between inputs and outputs is derived.
- v) The simplified Boolean function for each output obtained.
- vi) The logic diagram is drawn.

What is adder? Briefly explain the operation of a half adder and full adder with truth table.

Ans: Adder: An adder is a combinational logic circuit that performs the addition of bits is called adder.

Half adder: An half adder is a combinational logic circuit that performs the addition of two bits is called a half adder. The addition can be done by the rules of binary addition. It has two inputs and two outputs. The two inputs  $x$  and  $y$  and the two outputs  $S$ (sum) and  $C$ (carry). The truth table,  $S$ (sum)-map and  $C$ (carry)-map for a half adder are given below:

$x$	$y$	$S$	$C$
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

$x$	0	1	$x$	0	1
0	.	1	0	.	
1	1	0	1	.	1

$\therefore S\text{-map}$

$y$	0	1	$y$	0	1
0	.	1	0	.	
1	1	0	1	.	1

$C\text{-map}$

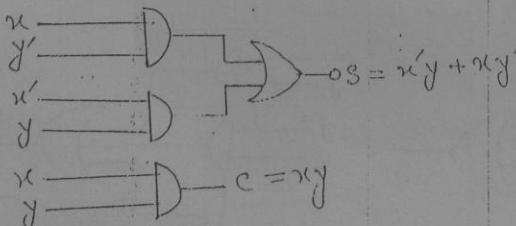
Truth-table

The carry output is '0' unless both inputs are 1. The  $S$  output represents the least significant bit of the sum.

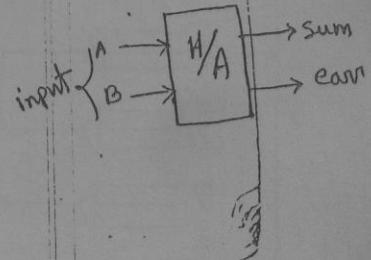
From the truth-table,  $S$ -map and  $C$ -map, we obtain the logical expression or expression for  $S$  and  $C$  outputs as

$$S = x'y + xy' \quad \text{and} \quad C = xy$$

Half adder circuit:



Block-Diagram:



The sum output is 1 if when only one input is equal to 1.  
when all three inputs are equal to 1.

Full adder: A full adder is a combinational logic circuit that performs the addition of three bits (two significant bit and one carry bit) is called a full adder. The addition can be done by the rules of binary addition. It has three inputs and two outputs. The three inputs  $x, y$  and  $z$  and the two outputs  $s$  (sum) and  $c$  (carry). The truth table,  $s$  (sum)-map and  $c$  (carry)-map for a full adder are given below:

$x$	$y$	$z$	$s$	$c$
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

Truth table

$x \setminus y \setminus z$	00	01	11	10
0	0	1		1
1	1		1	

$S$ -map

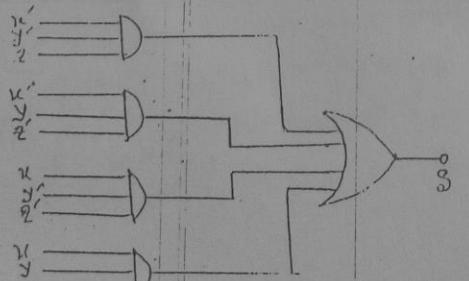
$x \setminus y \setminus z$	00	01	11	10
0			1	
1		(1)	(1)	(1)

$C$ -map

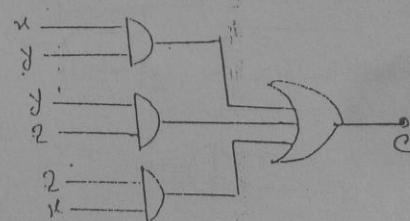
From the truth table,  $S$ -map,  $C$ -map, we obtained the logical operation or expression for  $S$  and  $C$  outputs as

$$S = x'y'z + x'y'z' + xy'z' + xyz \quad \text{and} \quad C = xy + yz + zx$$

Full adder circuit:

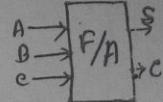


$$\text{For } S = x'y'z + x'y'z' + xy'z' + xyz$$



$$\text{For } C = xy + yz + zx$$

Block Diagram



So that, A half adder can be implemented with an Ex-OR gate and AND gate.

Half adder: A half adder is a combinational logic circuit that performs the addition of two bits is called a half adder. The addition can be done by the rules of binary addition. It has two inputs and two outputs. The two inputs  $x$  and  $y$  and the two outputs  $S$  (sum) and  $C$  (carry). The truth table,  $S$  (sum)-map,  $C$  (carry) map for a half adder are given below:

$x$	$y$	$s$	$c$
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

Truth table

$x$	$y$	0	1
0		0	1
1		1	

$S$ -map

$x$	$y$	0	1
0		0	
1			1

$C$ -map

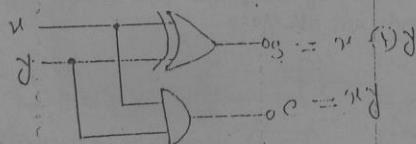
From the truth table, we  $S$ -map,  $C$ -map, we obtain the logical expression for  $s$  and  $c$  outputs as

$$s = x'y + xy' = x \oplus y \quad \text{and} \quad c = xy$$

From the expression, we can say that the half adder can be implemented with an Ex-OR gate and AND gate.

Half adder circuit:

Half



$$\begin{aligned} & xy + x'y \\ &= R \oplus x \\ & R'x + Rx \\ &= x \oplus y \end{aligned}$$

X Show that a full adder can be implemented with two half adders and an OR gate.

Ans: Full adder: A full adder is a combinational logic circuit that performs the addition of three bits (two significant bits and one carry bit) is called a full adder. The addition can be done by the rules of binary addition. It has three inputs and two outputs. The three inputs  $x, y$  and  $z$  and the two outputs  $s$ (sum) and  $c$ (carry).

The truth-table, S(sum)-map, C(carry) map for a full adder are given below:

$x$	$y$	$z$	$s$	$c$
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

Truth-table

$x \oplus y$	00	01	11	10
0		1		1
1	1		1	

$x \oplus y$	00	01	11	10
0			1	
1		(1)	(1)	(1)

S-map

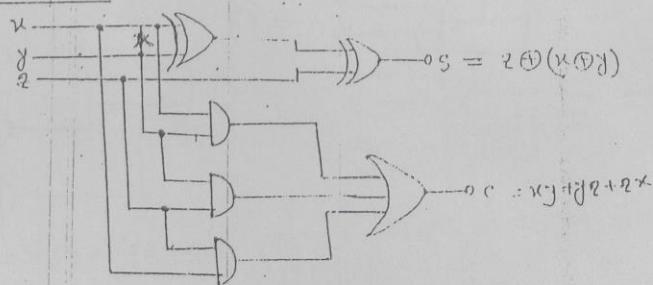
$$\text{and } c = xy + yz + zx$$

$$\begin{aligned} s &= x'y'z + x'y'z' + x'y'z + x'yz \\ &= xy'z + x'y'z + x'y'z + x'yz \\ &\Rightarrow z(xy + x'y') + z'(x'y + xy) \\ &= z(xy + x'y) + z'(x'y + xy) \\ &= z(x \oplus y) + z'(x \oplus y) \\ &= z \oplus (x \oplus y) \end{aligned}$$

$$\begin{aligned} \text{map:} \\ (xy + x'y') \\ = (x + y) \cdot (x'y) \\ = (xy + x) \end{aligned}$$

From the expression, we can say that the full adder can be implemented with two half adders and an OR gate.

Full adder circuit:



What is subtractor? Briefly explain the operation of half subtractor and full subtractor with truth table.

Ans: Subtractor: A subtractor is a combinational logic circuit that performs the subtraction of bits. It is called subtractor.

Half subtractor: A half subtractor is a combinational logic circuit that performs the subtraction of two bits and produces their difference. It also has an output to specify if a 1 has been borrowed. It has two inputs and two outputs. The two inputs are  $x$  and  $y$ , and the two outputs,  $B$  (Borrow) and  $D$  (difference). The truth table,  $B$  (Borrow)-map and  $D$  (difference)-map for a half subtractor are given below:

$x$	$y$	$B$	$D$
0	0	0	0
0	1	1	1
1	0	0	1
1	1	0	0

$x \setminus y$	0	1
0	.	1
1	.	.

B-map

$x \setminus y$	0	1
0	.	1
1	1	1

D-map

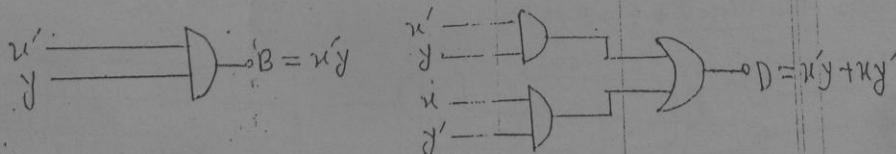
Truth-table

[The output borrow  $B$  is a 0 as long as  $x \geq y$ . It is a 1 for  $x < y$  and  $y = 1$ . [The D output is the result of the arithmetic operation,  $x - y$ .]]

From the truth table, B-map and D-map, we obtain the logical expression for B and D outputs as

$$B = x'y \quad \text{and} \quad D = x'y + xy'$$

Half subtraction circuit:



### Subtractors

Full adder: A full subtractor is a combinational logic circuit that performs a subtraction between two bits, taking into account that a 1 may have been borrowed by a lower significant stage.

It has three inputs and two outputs. The three inputs  $x, y$  and  $z$  and the two outputs output  $B$  (Borrow) and  $D$  (Difference). The truth table,  $B$  (Borrow)-map and  $D$  (Difference)-map for a full subtractor are given below:

$x'$	$y'$	$z'$	$B$	$D$
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	1	0
1	0	0	0	1
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1

$x'y'z'$	$00$	$01$	$11$	$10$
0	(1)	(0)	(1)	(1)
1			(1)	

$x'y'z'$	$00$	$01$	$11$	$10$
0		1		1
1	1		1	

Truth table

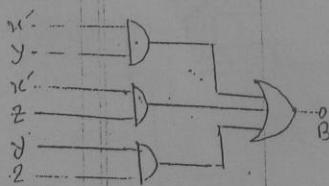
From the truth table,  $B$ -map and  $D$ -map, we obtain the logical expression for  $B$  and  $D$  outputs as

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

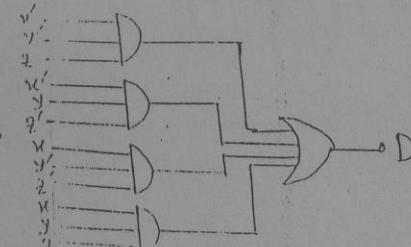
$$B = x'y'z + x'yz'$$

$$B = x'y + x'z + yz \text{ and } D = x'y'z + x'yz' + xy'z' + xyz$$

### Full subtractor circuit



$$\text{For } B = x'y + x'z + yz$$



$$\text{For } D = x'y'z + x'yz' + xy'z' + xyz$$

$$\text{first } x-y = c$$

What is a code converter? Explain the BCD to excess-3 code converter.

Ans: Code converter: A code converter is a circuit that converts one binary code to another.

BCD to excess-3 code converter: In BCD code and Excess-3

code, it must be four input variables and four output variables.  
Excess-3 code is formed by adding 3 to each BCD code.

The four input variables are A, B, C and D and the four output variables are w, u, v and z.

The truth table for BCD code and Excess-3 code are given below:

Input BCD				Output - Excess-3 Code			
A	B	C	D	w	u	v	z
0	0	0	0	0	0	1	1
0	0	0	1	0	1	0	0
0	0	1	0	0	1	0	1
0	0	1	1	0	1	1	0
0	1	0	0	0	1	1	1
0	1	0	1	1	0	0	0
0	1	1	0	1	0	0	1
0	1	1	1	1	0	1	0
1	0	0	0	1	0	1	1
1	0	0	1	1	1	0	0

AB\CD	00	01	11	10
00				
01				
11	X	X	X	X
10	1	1	X	X

$$w = A + BC + BD$$

AB\CD	00	01	11	10
00	1		(1)	
01	1		1	
11	X	X	X	X
10	1		X	X

$$y = CD + C'D'$$

AB\CD	00	01	11	10
00				
01				
11	X	X	X	X
10	1		X	X

$$u = B'C + B'D + BC'D'$$

AB\CD	00	01	11	10
00	1			1
01	1			1
11	X	X	X	X
10	1		X	X

$$z = D'$$

P.T.O

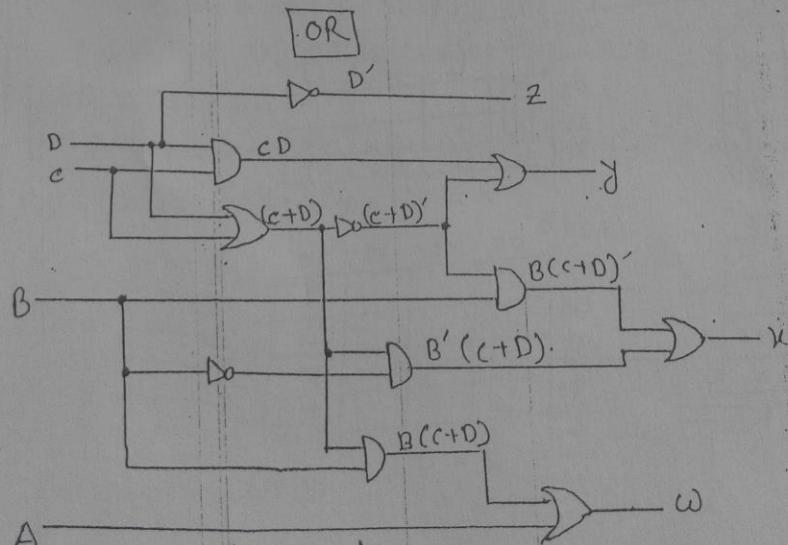
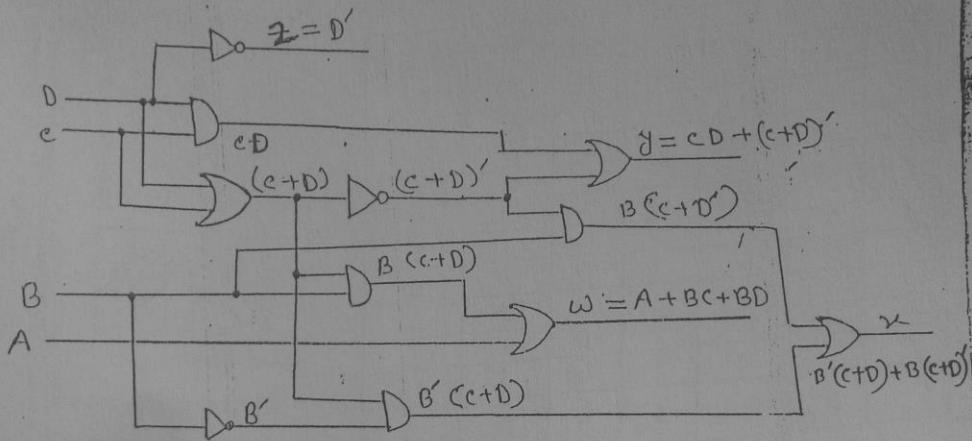
$$\therefore w = A + BC + BD = A + B(C+D)$$

$$x = B'C + B'D + BC'D' = B'(C+D) + B(C+D)'$$

$$y = CD + C'D' = CD + (C+D)'$$

$$z = D'$$

Circuit:



Logique diagram for BCD -> excess-3 code converter

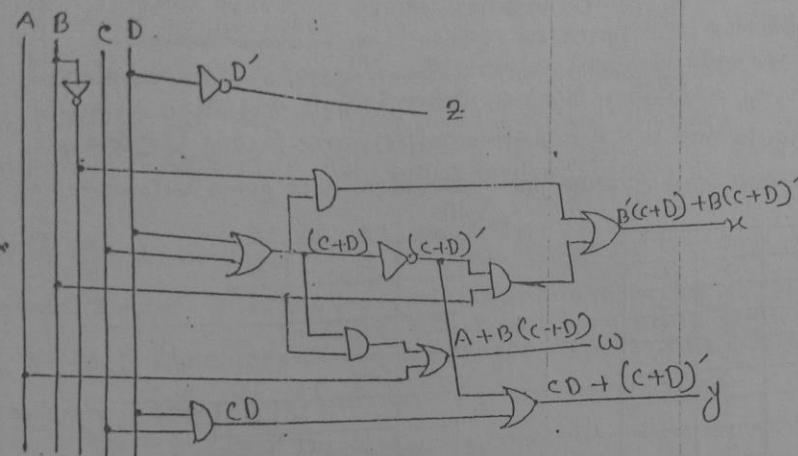
Circuit:

$$w = A + BC + BD = A + B(C+D)$$

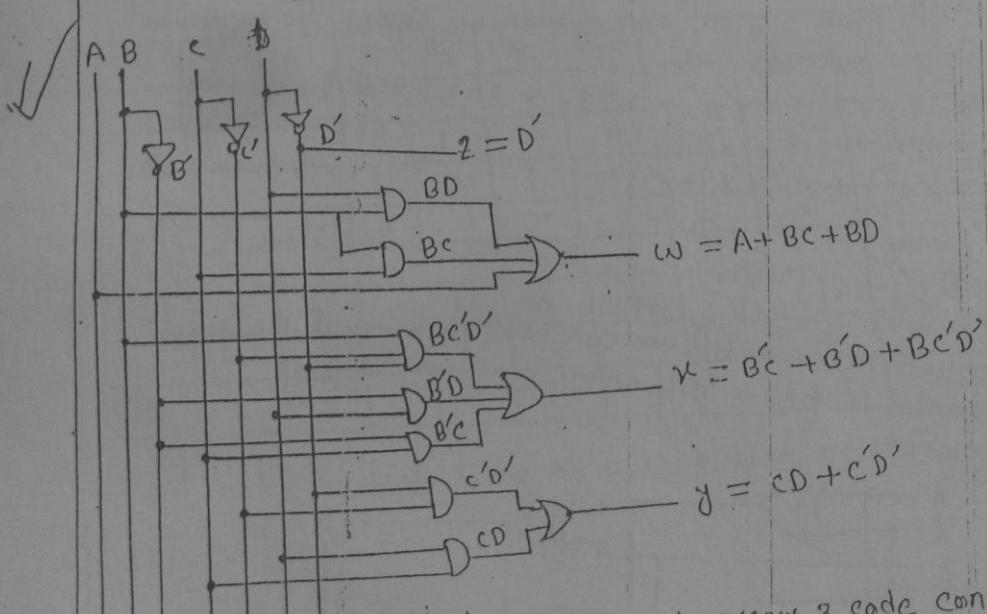
$$x = B'C + B'D + BC'D' = B'(C+D) + B(C+D)'$$

$$y = CD + C'D' = CD + (C+D)'$$

$$z = D'$$



OR



Logic diagram BCD-to-excess-3 code converter.

It shows that: A full adder can be implemented with two half adders and an OR gate.

Ans: A full adder is a combinational logic circuit that performs the addition of three bits (two significant bits and one carry bit) is called a full adder. The addition can be done by the rules of binary addition. It has three inputs and two outputs. Let, the three inputs are  $x, y, z$  and two outputs are  $S$  (sum) &  $C$  (carry).

The truth table, S-map and C-map for full adder are given below:-

$x$	$y$	$z$	$C$	$S$
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

Truth table

$x \setminus y \setminus z$	00	01	11	10
0	1		1	
1	1	1		

$x \setminus y \setminus z$	00	01	11	10
0			1	
1		1	1	1

$$\begin{aligned} \text{and } C &= \bar{x}\bar{y}z + \bar{x}y\bar{z} + xy\bar{z} + xyz \\ &= \bar{x}(y+z) + \bar{y}(x+z) \\ &= \bar{x}(x+y) + \bar{y}(x+y) \\ &= \bar{x} \oplus \bar{y} \quad \text{--- (ii)} \end{aligned}$$

From the truth table, S-map and C-map we obtain the expression for S and C outputs as

$$\begin{aligned} S &= \bar{x}\bar{y}z + \bar{x}y\bar{z}' + xy\bar{z}' + xyz \\ &= \bar{x}\bar{y}z + nyz + \bar{x}y\bar{z}' + xy\bar{z}' \\ &= z(\bar{x}y + x\bar{y}') + z'(\bar{x}y + xy') \\ &= z(\bar{x}y + xy') + z'(\bar{x} \oplus y) \\ &= z(\bar{x} \oplus y) + z'(\bar{x} \oplus y) \\ &= z \oplus (\bar{x} \oplus y) \quad \text{--- (i)} \end{aligned}$$

From the equation (i) and (ii), we can say that the full adder can be implemented with two half adder and an OR gate.

circuit design:

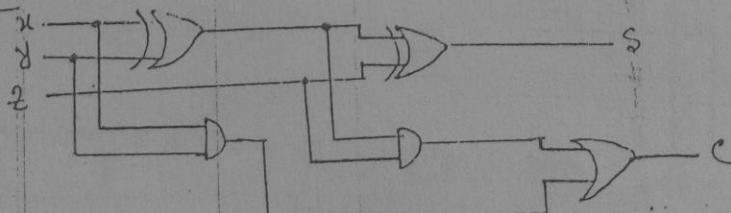


Fig: Implementation of a FA with two HAs and an OR gate.

\* What is decoders? Write the applications of decoders.

Ans: Decoder: A decoder is a combinational circuit that converts binary information from  $n$  input lines to a maximum of  $2^n$  unique output lines.

Applications of decoders:

- i) Decoder is used to identify the memory location.
- ii) " " " minimised the Boolean Algebra.
- iii) " " " in counter system.

\* Design a 3-to-8 line decoder/Design an binary-to-octal decoder.

Ans: 3-to-8 line decoder: Consider a fig @, for design the 3-to-8 line decoder. It has three inputs and eight outputs. The three inputs are decoded into eight outputs. The input variables may represent a binary number and the output variables represent the octal number system. Let the three inputs are  $x, y, z$  and outputs are  $D_0, D_1, D_2, D_3, D_4, D_5, D_6, D_7$ .

Input			Output							
$x$	$y$	$z$	$D_0$	$D_1$	$D_2$	$D_3$	$D_4$	$D_5$	$D_6$	$D_7$
0	0	0	1	0	0	0	0	0	0	0
0	0	1	0	1	0	0	0	0	0	0
0	1	0	0	0	1	0	0	0	0	0
0	1	1	0	0	0	1	0	0	0	0
1	0	0	0	0	0	0	1	0	0	0
1	0	1	0	0	0	0	0	1	0	0
1	1	0	0	0	0	0	0	0	1	0
1	1	1	0	0	0	0	0	0	0	1

Fig @: Truth table of 3-to-8 line decoder

P.T.O

From the truth table, we obtain the logical expression one

$$D_0 = x'y'z', \quad D_4 = x'y'z'$$

$$D_1 = x'y'z$$

$$D_5 = x'y'z$$

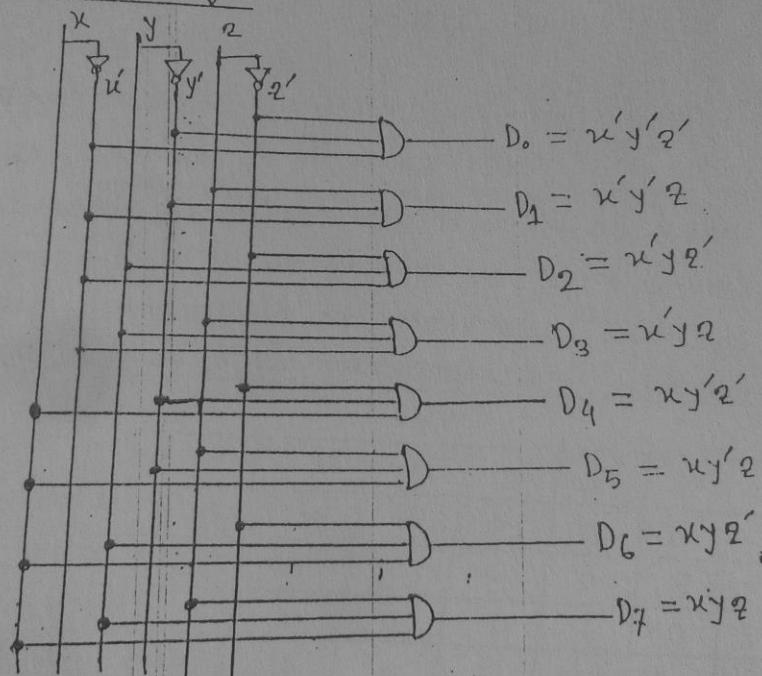
$$D_2 = x'y z'$$

$$D_6 = x y z'$$

$$D_3 = x'y z$$

$$D_7 = x y z$$

### Circuit design:



What is encoder? Design a 3-to-8 or octal to binary line encoder.

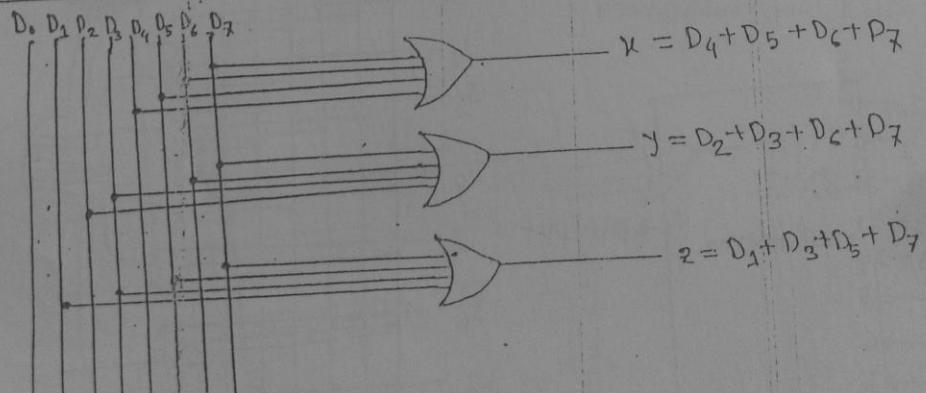
Aim: Encoder: An encoder is a combinational circuit that produces a reverse operation from that of a decoder. It has  $2^n$  input lines and  $n$  output lines.

8-to-3 line encoder: Consider a fig ①, for design the 8-to-3 line encoder. [It has  $2^3$  input lines and  $3$  output lines]. The input variables may represent a octal number and the output variables represent the binary number system. If it has eight inputs and three outputs. Let inputs are  $D_0, D_1, D_2, D_3, D_4, D_5, D_6, D_7$  and outputs are  $x, y, z$ .

Input								Output		
$D_0$	$D_1$	$D_2$	$D_3$	$D_4$	$D_5$	$D_6$	$D_7$	$x$	$y$	$z$
1	0	0	0	0	0	0	0	0	0	0
0	1	0	0	0	0	0	0	0	0	1
0	0	1	0	0	0	0	0	0	1	0
0	0	0	1	0	0	0	0	0	1	1
0	0	0	0	1	0	0	0	1	0	0
0	0	0	0	0	1	0	0	1	1	0
0	0	0	0	0	0	1	0	0	1	1
0	0	0	0	0	0	0	1	1	1	1

Fig ①: Truth table of 8-to-3 line encoder  
From the fig ①, we get  
 $\therefore x = D_4 + D_5 + D_6 + D_7$ ;  $y = D_2 + D_3 + D_6 + D_7$  and  $z = D_1 + D_3 + D_5 + D_7$

Circuit design:



\* Explain the operation of a  $8 \times 1$  line multiplexer.

Ans: It has 8 line inputs  $I_0$  to  $I_7$  is applied to one input of an AND gate. Selection lines  $S_1, S_2, S_3$  are decoded to select a particular AND gate.

Fig @ shows the  $8 \times 1$  line multiplexer:

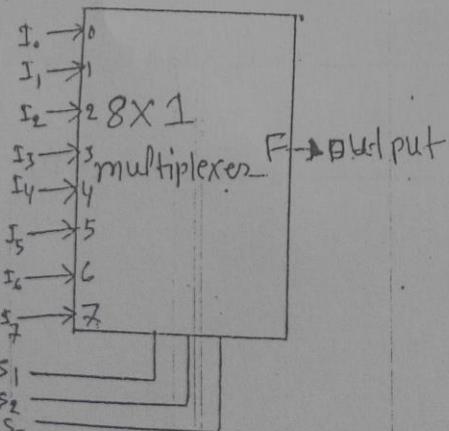
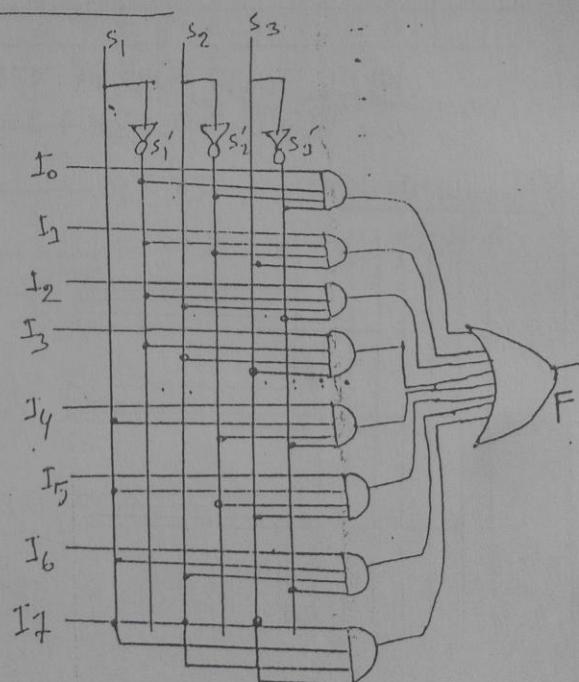
$S_1$	$S_2$	$S_3$	$F$
0	0	0	$I_0$
0	0	1	$I_1$
0	1	0	$I_2$
0	1	1	$I_3$
1	0	0	$I_4$
1	0	1	$I_5$
1	1	0	$I_6$
1	1	1	$I_7$

Fig @: Function table of  $8 \times 1$  line multiplexer.

From the function table, we get

$$F = S_1' S_2' S_3 I_0 + S_1' S_2 S_3' I_1 + S_1 S_2' S_3 I_2 + S_1 S_2 S_3' I_3 + \\ S_1 S_2 S_3' I_4 + S_1 S_2' S_3 I_5 + S_1 S_2 S_3 I_6 + S_1 S_2 S_3 I_7$$

Circuit design:



Block diagram

\* What is multiplexer? Write the applications of multiplexers.

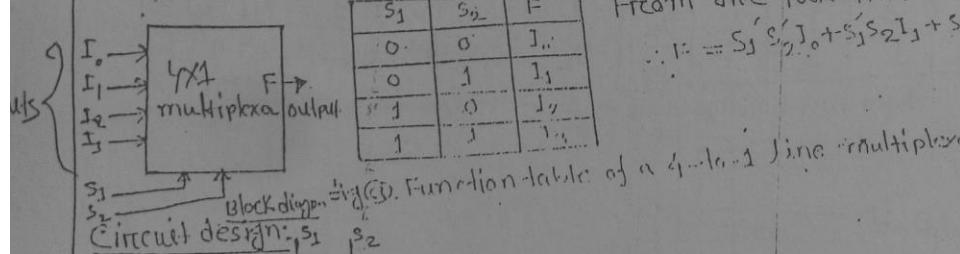
Ans: Multiplexer: A multiplexer is a combinational circuit that selects binary information from one of many input lines and directs it to a single output line. A multiplexer is also called a data selector.

The application of multiplexers:

- i) Data selection;
- ii) Data routing;
- iii) Logic function generation
- iv) Wave form generation;
- v) Operation sequencing;
- vi) Parallel to serial conversion.

\* Describe the operation of a 4-to-1 line multiplexer.

Ans: 4-to-1 line multiplexer: Consider a fig @, to explain the operation of 4-to-1 line multiplexer. It has four input lines  $I_0, I_1, I_2$  and  $I_3$  and  $S_1, S_2$  are applied to one input lines of an AND gate. Selection lines  $S_1$  and  $S_2$  are decoded to select a particular AND gate. To demonstrate the circuit operation, consider the case when  $S_1S_2=10$ .

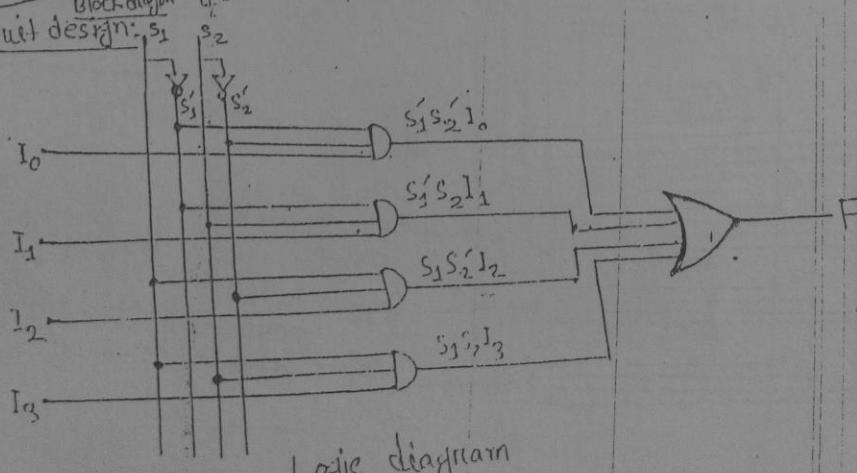


$S_1$	$S_2$	F
0	0	$I_0$
0	1	$I_1$
1	0	$I_2$
1	1	$I_3$

From the function-table, we get

$$F = S_1' S_2' I_0 + S_1' S_2 I_1 + S_1 S_2' I_2 + S_1 S_2 I_3$$

Fig (b). Function-table of a 4-to-1 line multiplexer



\* What is demultiplexer? Describe the operation of 1-to-4 line demultiplexer.

Ans: Demultiplexer: A demultiplexer is a circuit that performs the reverse operation on a single line input and distributes it over several outputs. It is also called data distribution.

1-to-4 line demultiplexer: Consider a fig①, for explain the operation of 1-to-4 line demultiplexer. It has a single input  $I$  and four outputs  $y_0, y_1, y_2$  and  $y_3$ . Input  $I$  is connected to all outputs. But the input information is directed to only one of the output lines as specified by the two selection variables with enable input  $E$ .

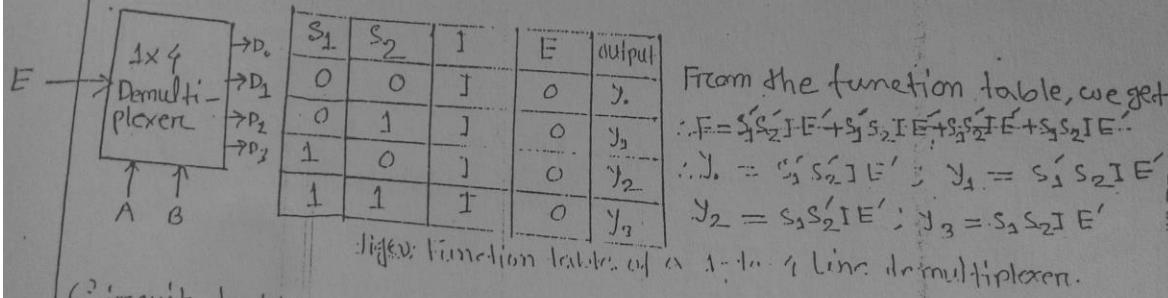
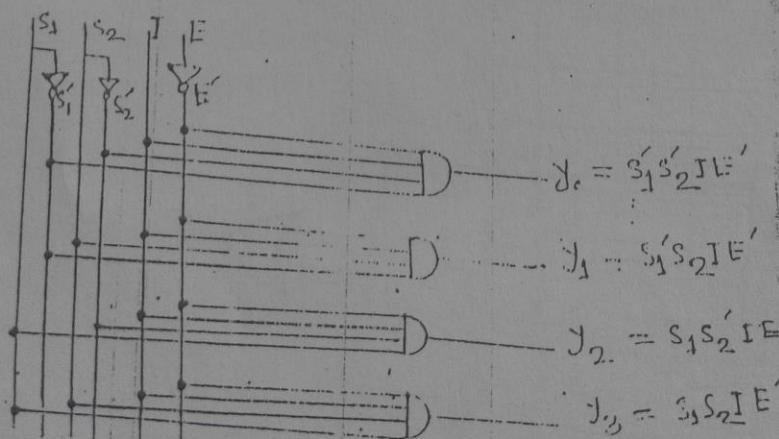


Fig: Function table of a 1-to-4 line demultiplexer.

### Circuit design



\* Describe the operation of 2x4 or 2-to-4 line demultiplexer.

Ans: 2-to-4 line demultiplexer: Consider a fig @, to explain the operation of 2-to-4 line demultiplexer. A 2-to-4 line decoder with an enable input constructed with NAND gates. All outputs are equal to 1 if enable input E is 1. When the input enable input is 0, the circuit operates as a decoder with complemented outputs.

The X's under A and B are don't care conditions.

E	A	B	D <sub>0</sub>	D <sub>1</sub>	D <sub>2</sub>	D <sub>3</sub>
1	x	x	1	1	1	1
0	0	0	0	1	1	1
0	0	1	1	0	1	1
0	1	0	1	1	0	1
0	1	1	1	1	1	0

$$\therefore D_0 = AB'E'$$

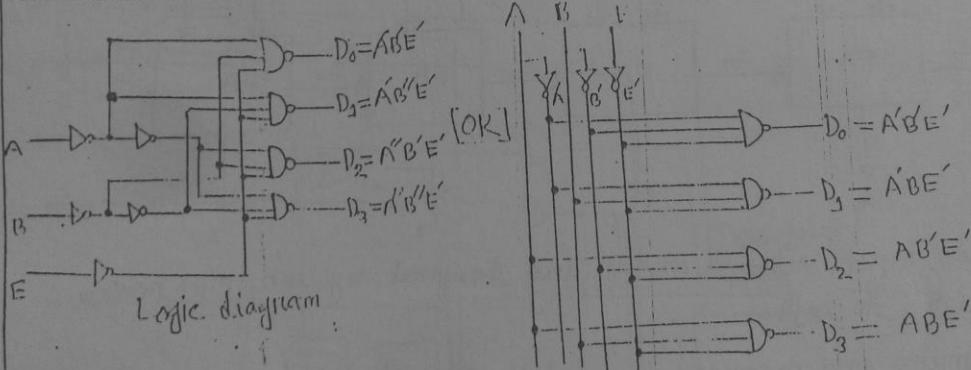
$$D_1 = A'B'E' = ABE'$$

$$D_2 = A''B'E' = AB'E'$$

$$D_3 = A''B'E' = ABE'$$

Fig @ Function table of a 2-to-4 line Demultiplexer.

### Circuit Design:



Logic diagram

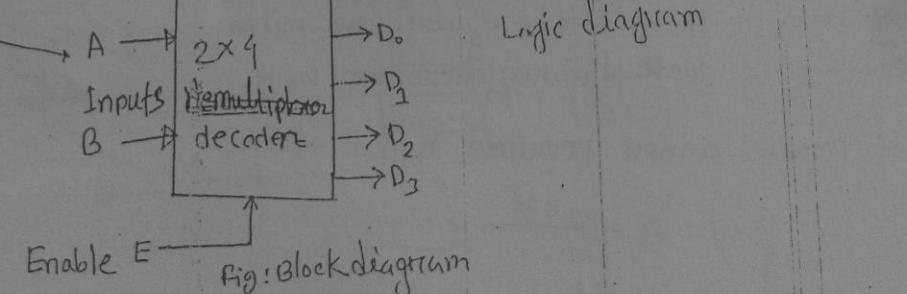


Fig: Block diagram

61926707126

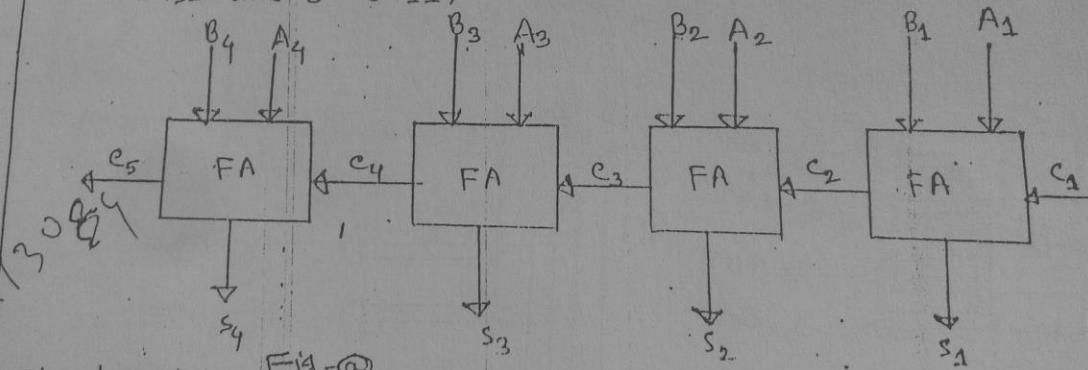
Q. What is binary parallel adder? Explain it?

Ans: Binary parallel adder: A binary parallel adder is a digital function that produces the arithmetic sum of two binary numbers in parallel.

It consists of full adders connected in cascade, with the output carry from one full adder connected to the input carry of the next full adder.

Fig(a) shows the interconnection of four full-adder circuits to provide a 4-bit binary parallel adder. Consider two binary numbers,

$A = 1011$  and  $B = 0011$ , whose sum is  $S = 1110$ .



The two input values, A and B, are designed by subscript numbers from right to left.

The carries are connected in a chain through the full-adders.

The input carry to the adder is  $c_1$  and the output carry is  $c_5$ .

The outputs generate the required sum bits.

An n-bit parallel adder requires n full-adders.

\* what is magnitude comparator? Explain 4-bit magnitude comparator.

Ans: Magnitude comparator: A magnitude comparator is a combinational circuit that compares two numbers, A and B, and determines their relative magnitudes. Since the relation between A and B are  $A = B$ ,  $A > B$  and  $A < B$ .

4-bit magnitude comparator: Let the two numbers are A and B where each number have 4-digits.

$$\therefore A = A_0 A_1 A_2 A_3 \text{ and } B = B_0 B_1 B_2 B_3$$

Since the relation between A and B are  $A = B$ ,  $A > B$  and  $A < B$  will be held when the boolean function are

$$\text{When } A = B, \text{ then } (A = B) = x_0 x_1 x_2 x_3$$

$$\therefore A > B, \quad A > B = x_1 x_2 x_3 A_0 B_0 + x_2 x_3 A_1 B_1 + x_3 A_2 B_2 + A_3 B_3$$

$$\therefore A < B, \quad A < B = x_0 x_1 x_3 A_0 B_0 + x_0 x_3 A_1 B_1 + x_3 A_2 B_2 + A_3 B_3$$

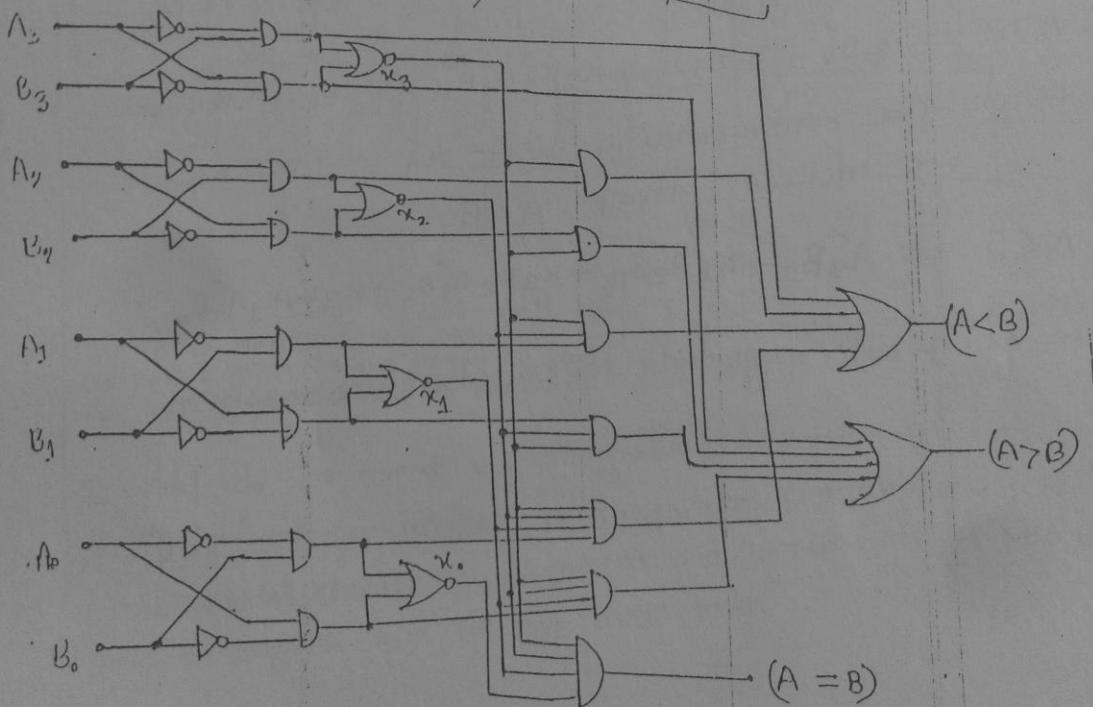


Fig: 4 bit magnitude comparator

### Alternative operation of 4-bit magnitude comparators:

Let, the two numbers are A and B where each number have 4-digits.

$$\therefore A = A_3 A_2 A_1 A_0 \text{ and}$$

$$B = B_3 B_2 B_1 B_0$$

The binary variables ( $A=B$ ) is equal to 1 only if all pairs of digits of the two numbers are equal.

$$\therefore (A=B) = x_3 x_2 x_1 x_0$$

When the corresponding digit of A is 1 and B is 0, we include that:  $A > B$ .

$$\therefore A > B = A_3 B'_3 + x_3 A_2 B'_2 + x_3 x_2 A_1 B'_1 + x_3 x_2 x_1 A_0 B'_0$$

Again when the corresponding digit of A is 0 and B is 1, we include that:  $A < B$

$$\therefore A < B = A'_3 B_3 + x_3 A'_2 B_2 + x_3 x_2 A'_1 B_1 + x_3 x_2 x_1 A'_0 B_0$$

The figure of 4-bit magnitude comparators are shown in Fig (a).

Q) Alternative: What is binary parallel adder? Explain it.

Ans: Binary parallel adder: A binary parallel adder is a digital function that produces the arithmetic sum of two binary numbers in parallel.

It consists of full adders connected in cascade, with the output carry from one full adder connected to the input carry of the next full adder.

This is shown in the following table:

Subscript	4	3	2	1	
Input carry	0	1	1	0	$c_4$
Augend	1	0	1	1	$A_4$
Addend	0	0	1	1	$B_4$
					$S_4$
Sum					$s_4$
Output carry					$c_1$
	$B_3$	$A_3$	$B_2$	$A_2$	$B_1$
	$c_3$	$c_2$	$c_1$		$A_3$
					$S_3$
					$S_2$
					$S_1$

Fig: 4-bit full adder.  
The input carry to the adder is  $c_4$  and the output carry is  $c_4$ . The outputs generate the required sum bits.  
An n-bit parallel adder requires n full adders.

Q) What do you mean by magnitude comparators?

Ans: Magnitude comparators: A magnitude comparator is a combinational circuit that compares two numbers, A and B, and determine their relative magnitudes.

Q1. What is Decimal Adder? Explain the BCD adder with circuit block diagram.

Decimal Adder: A decimal adder is one kind of adder that performs the addition of decimal bits which requires of nine inputs and five outputs since four bits are required to code each decimal digit and the circuit must have an input carry and output carry.

BCD Adder: A BCD adder is a circuit that adds two BCD digits in parallel and produces a sum digit also in BCD.

binary numbers are then labeled by symbols  $z_1, z_2, z_4, z_8$  and  $K$  where  $K$  is the carry.

The output carry can be expressed by the Boolean function:

$$J_C = K + z_8 z_4 + z_8 z_2$$

When  $C = 1$ , it is necessary to add 0110 to the binary sum and provide an output carry for the next stage.

Derivation of a BCD Adder.

Binary sum				BCD sum				Decimal
$K$	$z_1$	$z_2$	$z_4$	$c$	$s_8$	$s_4$	$s_2$	$s_1$
0	0	0	0	0	0	0	0	0
0	0	0	0	1	0	0	0	1
0	0	0	1	0	0	0	1	0
0	0	0	1	1	0	0	1	1
0	0	1	0	0	0	1	0	0
0	0	1	0	1	0	1	0	0
0	0	1	1	0	0	1	0	1
0	0	1	1	1	0	1	1	1
0	1	0	0	0	1	0	0	0
0	1	0	0	1	1	0	0	1
0	1	0	1	0	1	0	0	2
0	1	1	0	0	1	0	1	3
0	1	1	0	1	1	0	1	4
0	1	1	1	0	1	1	0	5
0	1	1	1	1	1	1	0	6
0	1	1	1	1	1	1	1	7
0	1	0	0	0	1	0	0	8
0	1	0	0	1	1	0	0	9
1	0	1	0	0	0	0	0	10
1	0	1	1	0	0	0	1	11
1	1	0	0	0	0	0	1	12
1	1	0	1	0	0	1	0	13
1	1	1	0	0	0	1	0	14
1	1	1	1	0	0	1	0	15
1	1	1	1	1	0	1	0	16
1	0	0	0	0	1	1	1	17
1	0	0	0	1	0	1	1	18
1	0	0	1	0	1	0	0	19
1	0	0	1	1	1	0	0	19

PTU

### Circuit Design:

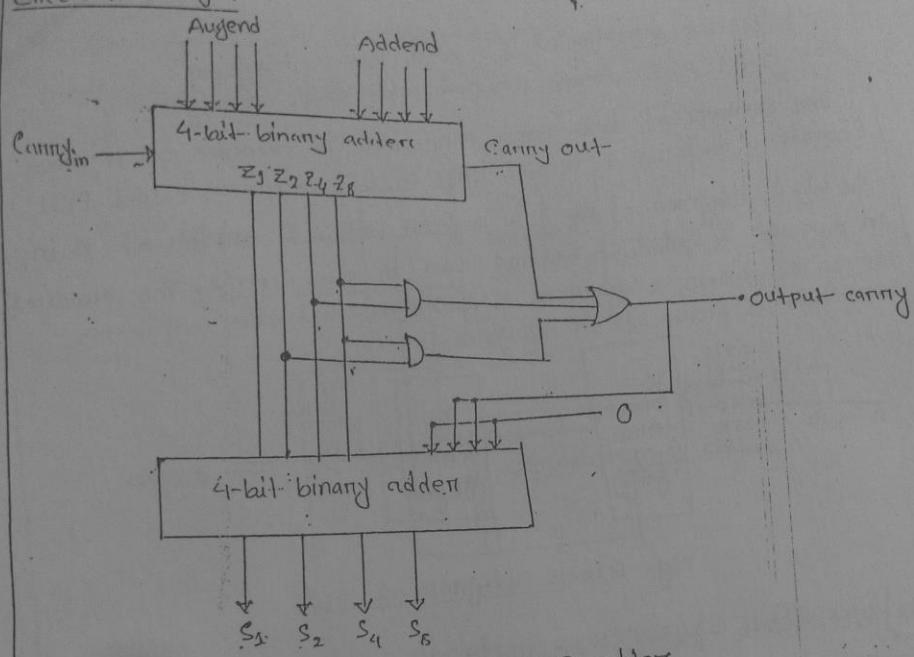


Fig: Block diagram of a BCD adder.

A decimal parallel adder that adds  $n$  decimal digits needs  $n$  BCD adder stages.

~~Q1 Explain the operation of PLA with Block diagram.~~

Ans: PLA: PLA means programmable logic array.

The number of don't care conditions is excessive, so it is more economical to use a second type of LSI component called PLA.

The block diagram of the PLA is given below: It consists of  $n$  inputs,  $m$  outputs,  $K$  product terms and  $m$  sum terms. The product terms constitute a group of  $K$  AND gates and the sum terms constitute a group of  $m$  OR gates.

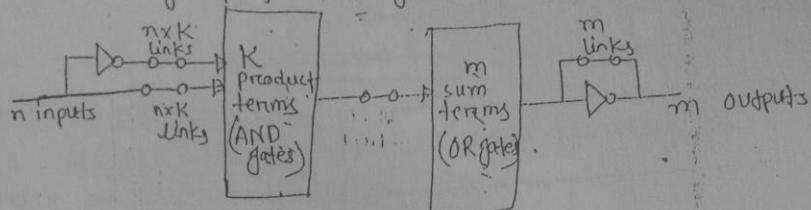


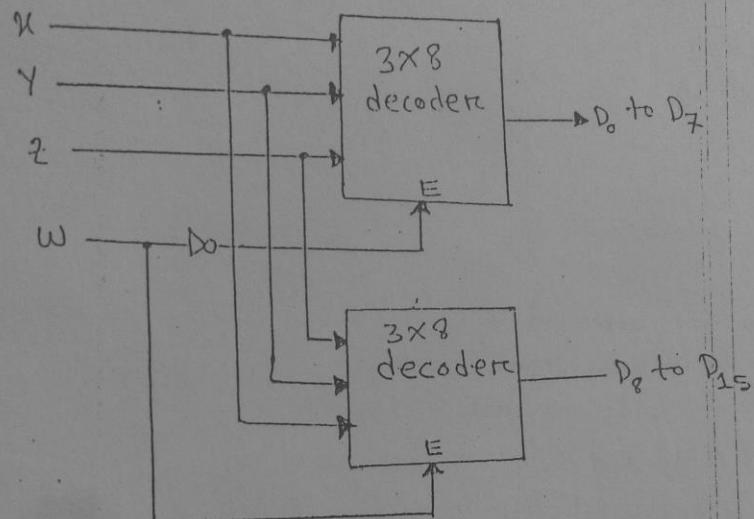
Fig: Block diagram of PLA.

\* Differentiate between multiplexer and demultiplexer.

Multiplexer	Demultiplexer
1. Definition: —	1. Definition: —
2. Also known as data selector.	2. Also known as data distributor.
3. It has many inputs and one output.	3. It has one input and many outputs.
4. It is used to parallel-to serial conversion.	4. It is used to serial-to parallel conversion.
5. When we design multiplexer, we don't need additional gates.	5. When we design demultiplexer we need additional gates
6. Example: $4 \times 1$ line multiplexer	6. Example: $2 \times 4$ line demultiplexer

\* Construct a  $4 \times 16$  decoder with two  $3 \times 8$  decoders.

Ans: A  $4 \times 16$  decoder with two  $3 \times 8$  decoders are shown in Fig@.



Fig@: A  $4 \times 16$  decoder constructed with two  $3 \times 8$  decoders.

Explain the operation:

- i) when  $w = 0$ , the top decoder is enabled and bottom decoder is disabled. In this case, the top decoder outputs generate minterms 0000 to 0111, while the outputs of the bottom decoder are all 0's.
- ii) when  $w = 1$ , the top decoder is disabled and bottom decoder is enabled. In this case, the bottom outputs generate minterms 1000 to 1111, while the output of the top decoder are all 0's.