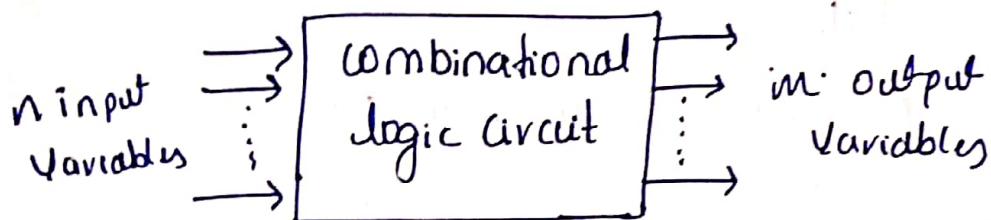


Digital Electronics

Module - 3. Combinational Circuits Design

combinational logic circuits: In this type of logic circuits, outputs depend only on the current inputs.

combinational logic circuits



- A combinational circuit consists of input Variables (n), logic gates, & output Variables (m)
 - For n input Variables there are 2^n possible combination of binary I/O Values
 - For each possible input combination there is one and only one possible o/p combinational circuit can be describe by (m) Boolean functions, one for each o/p variables
 - Each output function expressed in terms of the (n) input Variables.
- While designing following to be consider
1. Min no of gates
 2. Min. no of I/O's to gates
 3. Min. no of interconnections
 4. Min. propagation time of signal throw the circuit.

Adder & Subtractor (Half and Full).

Adder : Digital computers perform a variety of information processing task. Among the basic functions encountered are the various arithmetic operations (addition).

Binary Arithmetic

1 addition : The rules of addition are

$$0 + 0 = 0$$

$$0 + 1 = 1$$

$$1 + 0 = 1$$

$$1 + 1 = 10$$

$$1 + 1 + 1 = 11$$

Binary adder - Half adder

Half adder : is a combinational circuit that performs the addition of two bits. This circuit needs 2 binary inputs and two binary outputs.

$$n=2 \text{ bit} \quad n=2^2=4$$

Inputs		Outputs	
x	y	c	s
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

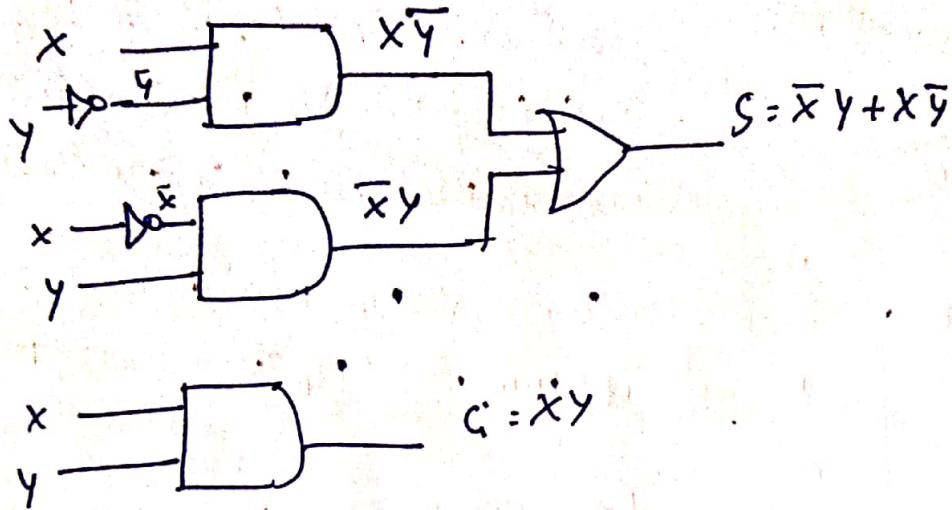
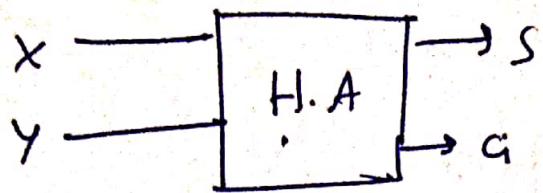
where S = Sum

C = Carry

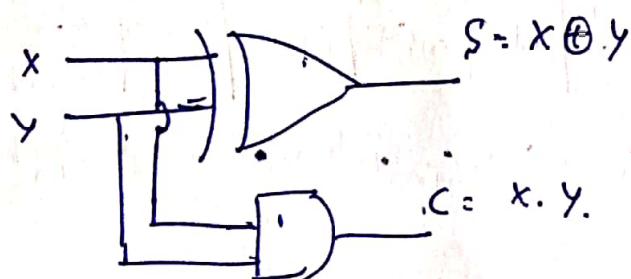
$$\left. \begin{array}{l} S = \bar{x}y + x\bar{y} \\ C = xy \end{array} \right\} \text{using SOP form}$$

$$\left. \begin{array}{l} S = x \oplus y \\ C = xy \end{array} \right\} \text{using XORG AND gate}$$

The Block diagram for the half adder



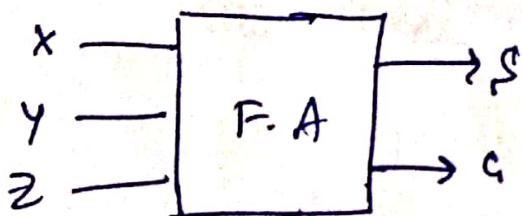
using XOR & AND gate



Full adder: is a combinational circuit that performs the addition of three bits. It consists of 3 stages 2 - outputs.

$$n = 3 \text{ bits} \quad n = 2^3 = 8$$

The Block diagram for the Full adder



Truth table for F.A

Inputs			Outputs	
X	Y	Cin	S	Cout
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

→ The S output is equal to 1 when only one I/p is equal to 1. Otherwise, all three I/p's are equal to 1.

→ The Cout O/p has a carry 1 if two or 3 I/p's are equal to 1.

		For Sum			
		YC	YC̄	YC	YC̄
		00	01	11	10
x	0	0	1	0	1
x	1	1	0	1	0

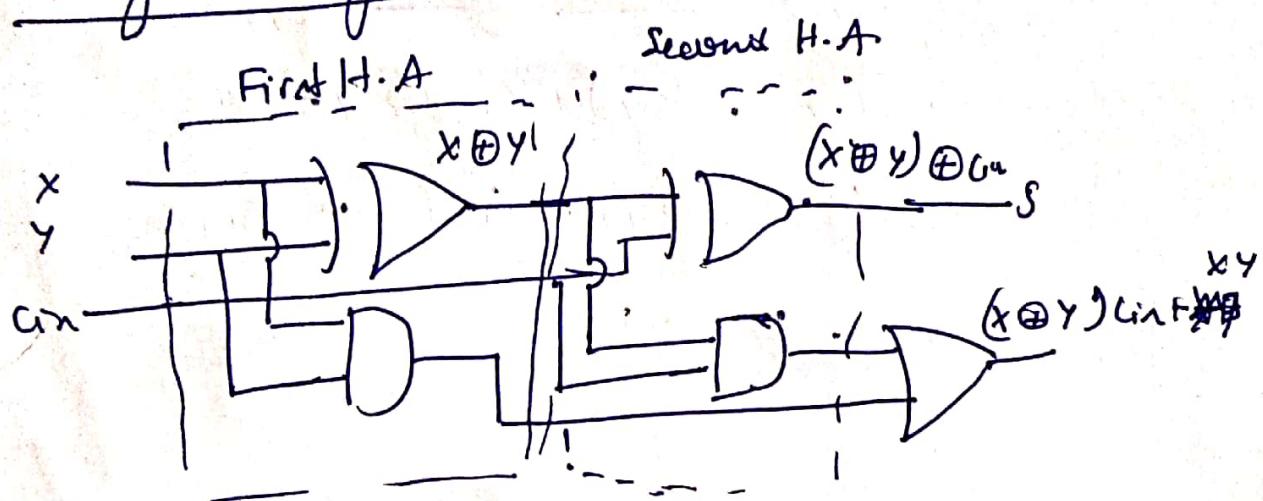
$$S = \bar{x}\bar{y}C_{in} + \bar{x}y\bar{C}_{in} + x\bar{y}C_{in} + xyC_{in}$$

For carry

		C _y			
		00	01	11	10
		0	0	1	0
x	0	0	1	0	
x	1	1	0	1	0

$$C_{out} = xy + xC_{in} + yC_{in}$$

Using XOR gate



Subtractor: Digital computers perform a variety of information processing tasks. Among the basic functions encountered are the various arithmetic operations (Subtraction).

Subtraction rules

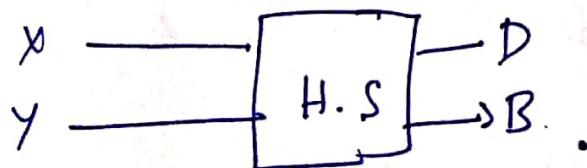
$$0 - 0 = 0$$

$$\emptyset - 0 = 1$$

$$10 - 1 = 1 \quad (\text{The } 1 \text{ borrowed from the next higher stage})$$

$$1 - 1 = 0$$

Half Subtractor,



where $D = \text{difference}$

$B = \text{Borrow}$.

$$\text{The Sum} = \overline{x}\overline{y}c_{in} + \underline{\overline{x}y\overline{c}_{in}} + \underline{x\overline{y}c_{in}} + xy c_{in}$$

$$= \overline{c_{in}} (\overline{x}y + x\overline{y}) + c_{in} (\overline{x}\overline{y} + xy)$$

$$= \overline{c_{in}} (xy + x\overline{y}) + c_{in} (\overline{x}\overline{y} + x\overline{y})$$

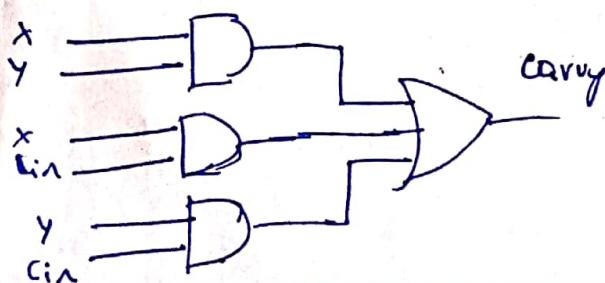
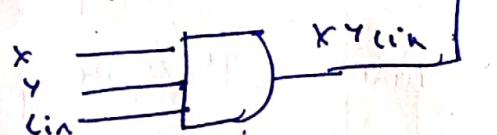
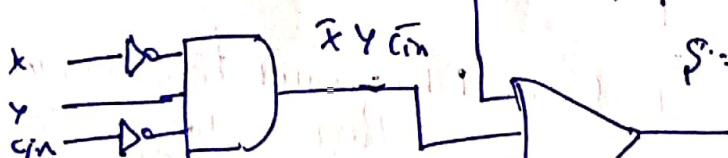
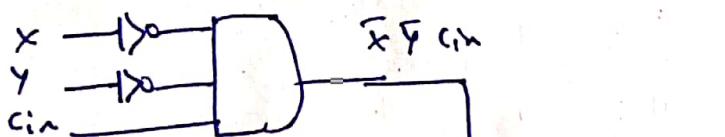
$$S = c_{in}(\overline{x} \oplus y)$$

The carry:

$$c_{out} = \overline{x}\overline{y}c_{in} + \underline{x\overline{y}c_{in}} + \underline{x\overline{y}c_{in}} + \overline{x}\overline{y}c_{in}$$

$$c_{in} (\overline{x}y + x\overline{y}) + xy (c_{in} + \overline{c_{in}}) \because A + \overline{A} = 1$$

$$c_{out} = c_{in}(x \oplus y) + xy$$

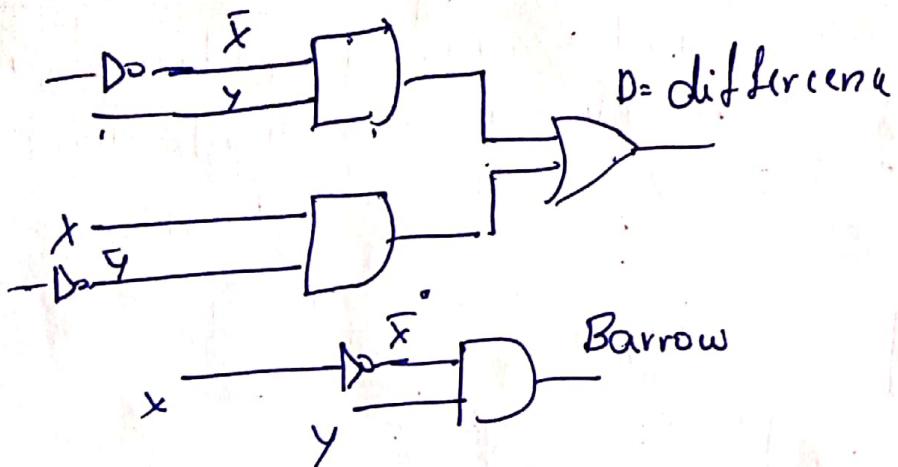


Inputs		outputs	
x	y	B	D
0	0	0	0
0	1	1	1
1	0	0	1
1	1	0	0

$$\begin{aligned} D &= \bar{x}y + x\bar{y} = x \oplus y \\ B &= \bar{x}\bar{y} \end{aligned}$$

From
SOPs
XOR gate

Difference:



Full subtractor:

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

Difference

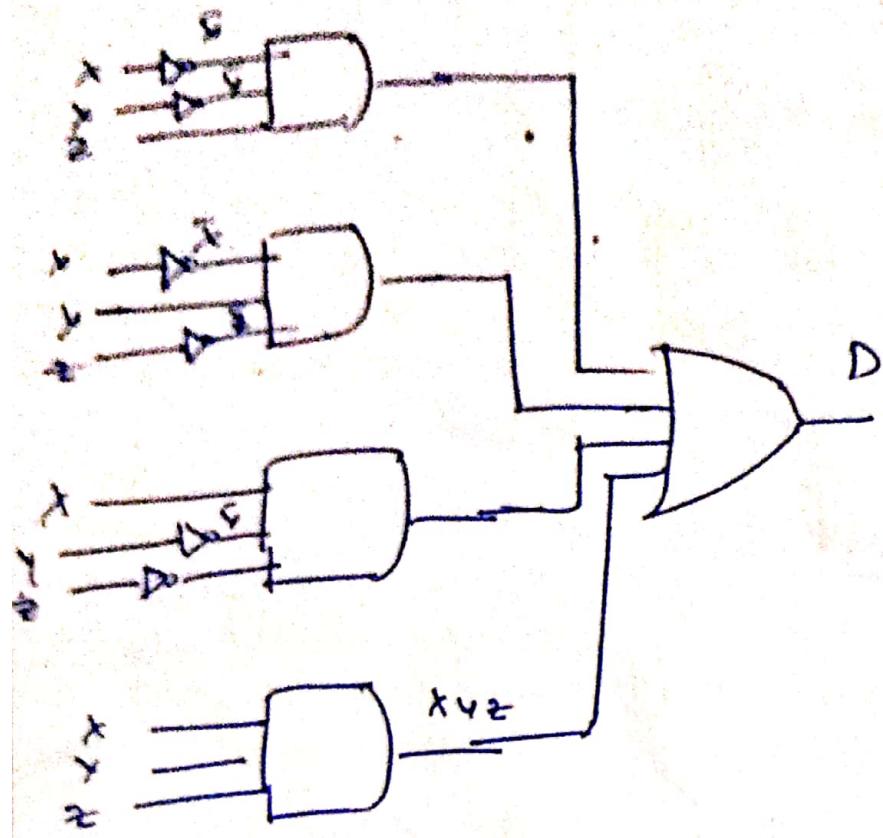
x	00	01	11	10
0	0	1	1	0
1	1	0	0	1

$$D = \bar{x}\bar{y}z + \bar{x}y\bar{z} + x\bar{y}z + xy\bar{z}$$

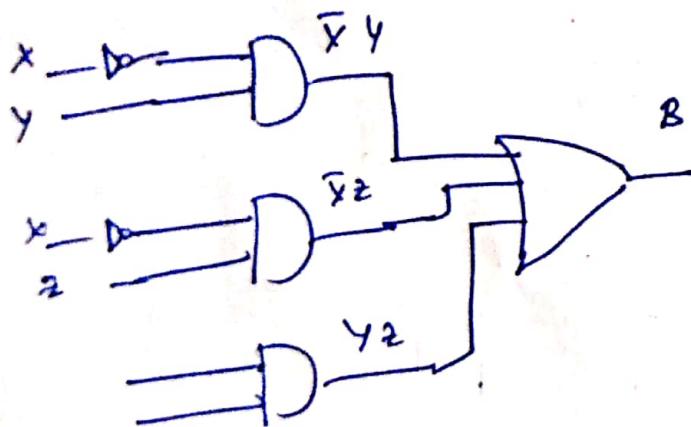
Borrow

x	00	01	11	10
0	0	1	1	0
1	1	0	0	1

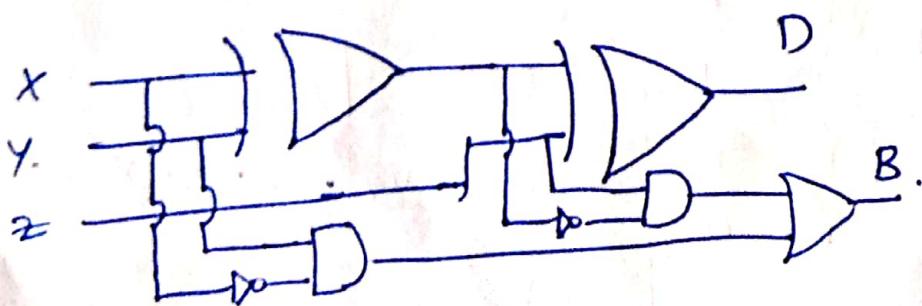
$$B = \bar{x}y + \bar{x}z + yz$$



$$B = \bar{x}y + \bar{x}\bar{z} + y\bar{z}$$



$$\begin{aligned}
 B_{\text{alt}} &= \bar{x}\bar{y}z + \bar{x}y\bar{z} + \bar{x}yz + xy\bar{z} \\
 &= y(\bar{x}y + \bar{x}\bar{z}) + \bar{x}y(z + \bar{z}) \\
 &= y(x \oplus y) + \bar{x}y
 \end{aligned}$$



Parallel Binary adder

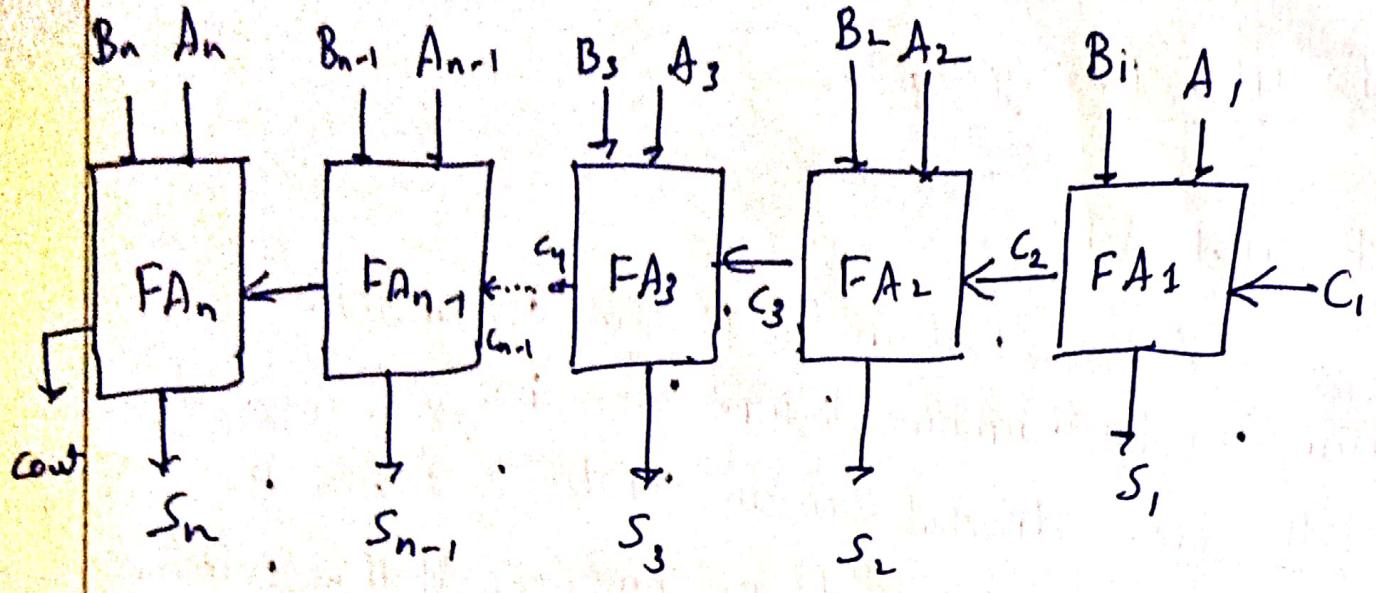
Parallel Adder:

A single Full adder performs the addition of two one bit numbers & input carry but a parallel-adder is a digital circuit capable of finding the arithmetic sum of two binary numbers that is greater than one bit in length by operating on correspondingly pair of bits in parallel.

If consists of Full adder connected in a chain where the output carry from each full adder is connected to the carry I/p of the next higher Order F.A. in the chain.

A "n" bit Parallel adder requires "n" Full adder to perform the operation. So for 2-bit numbers two adder are needed while for 4-bit "no" 4-FAdder required.

Parallel adder normally incorporate carry lookahead logic to ensure that carry propagation between subsequent stages of addition does not limit addition speed.

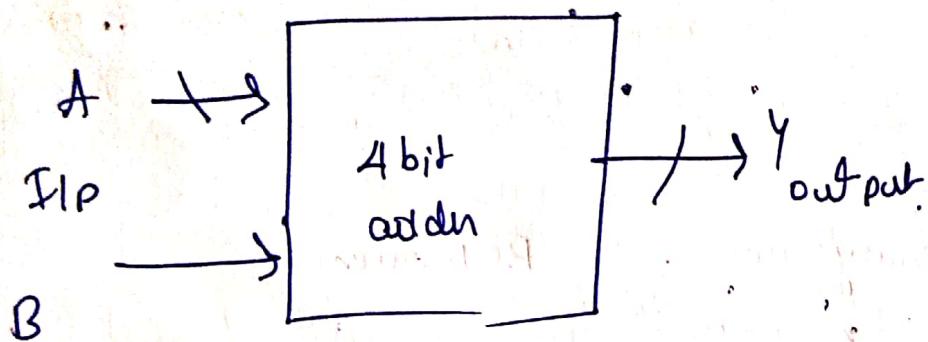


Working of Parallel Adder

1. As shown in the fig. above firstly full adder FA₁ adds A₁ & B₁ along with the carry C₁ to generate the sum (S₁) (The first bit of the O/p sum) and the carry C₂ which is connected to the next adder in chain.
2. Next the full adder FA₂ uses this carry bit C₂ do add with I/p bits A₂ & B₂ to generate the sum (S₂) and carry C₃ which is again further connected to the next adder in chain & so on.
3. The process continues till the last full adder F_n. This full adder uses the carry bit C_n to add with I/p bits A_n & B_n to generate the last bit of the O/p along with Cout.

BCD adder:

BCD stands for Binary coded decimal. Suppose we have two 4-bit numbers, A and B. the value of A and B can varies from 0 (0000) to 9 (1001). because we are considering decimal numbers.



The o/p. will varies from 0 to 18, if we are not considering the carry from previous sum but if we are considering the carry, then the maximum value of output will be 19 (ie $9 + 9 + 1 = 19$)

when we are simply adding A and B, then we get the binary sum. Here get the o/p in the BCD adder

Addition of two- RCD No.

Sum 59

$$\sum m \leq q_c = 1$$

$$\text{Sum} \geq 0$$

Ans is correct
No correction
given.

Add -6 to the
sum term to
get correct Ans

Add 'G' to the
sum to get
correct Ans.

Decimal	Binary Sum	BCD Sum
0	G S ₃ S ₂ S ₁ S ₀	
1	0 0 0 0 0	
2	1 0 0 0 0 1	
3	1 0 0 0 1 0	
4	1 0 0 0 1 1	
5	1 0 0 1 0 0	
6	1 0 0 1 0 1	
7	1 0 0 1 1 0	
8	1 0 0 1 1 1	
9	1 0 0 0 0 0	
10	1 0 1 0 0 1	
11	1 0 1 0 1 0	
12	1 0 1 0 1 1	
13	1 0 1 1 0 0	
14	1 0 1 1 0 1	
15	1 0 1 1 1 0	
16	1 0 1 1 1 1	
17	1 0 0 0 0 0	
18	1 0 0 0 0 1	
19	1 0 0 1 0	
20	1 0 0 1 1	

Code Converters :-

Gray code System is a binary number system in which every successive pair of numbers differs in only one bit. It is used in application in which the normal sequence of binary numbers generated by the hardware may produce an error or ambiguity the transition from one no. to the next.

Converting Binary to Gray code

Let B_0, B_1, B_2, B_3 be the bits representing the binary number. Where B_0 is the LSB & B_3 be the MSB.

Let $g_{40}, g_1, g_2 \& g_3$ be the bits representing the gray code of the binary number where g_0 is LSB & g_3 be the MSB.

Binary				Gray			
B_3	B_2	B_1	B_0	g_3	g_2	g_1	g_0
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	1
0	0	1	0	0	0	1	1
0	0	1	1	0	0	1	0
0	1	0	0	0	1	1	0
0	1	0	1	0	1	1	1
0	1	1	0	0	1	0	1
0	1	1	1	0	1	0	0
1	0	0	0	1	1	0	0
1	0	0	1	1	1	0	1
1	0	1	0	1	1	1	1
1	0	1	1	1	1	1	0
1	1	0	0	1	0	1	0
1	1	0	1	1	0	1	1
1	1	1	0	1	0	0	1
1	1	1	1	1	0	0	0

To find the corresponding digital circuit we will use the K-map Techniques for each of the Gray code bits as g_0 with all of the binary bits as B_i .

K-map for g_0

B_3, B_2	00	01	11	10
00	0	1	0	1
01	0	1	0	1
11	0	1	0	1
10	0	1	0	1

K-map for g_1

B_3, B_2	00	01	11	10
00	0	0	1	1
01	1	1	0	0
11	1	1	0	0
10	0	0	1	1

$$G_0 = B_0 \bar{B}_1 + B_1 \bar{B}_0 = B_0 \oplus B_1$$

$$G_1 = B_2 \bar{B}_3 + B_3 \bar{B}_2 = B_2 \oplus B_3$$

K-map for g_2

B_3, B_2	00	01	11	10
00	0	0	0	0
01	1	1	1	1
11	0	0	0	0
10	1	1	1	1

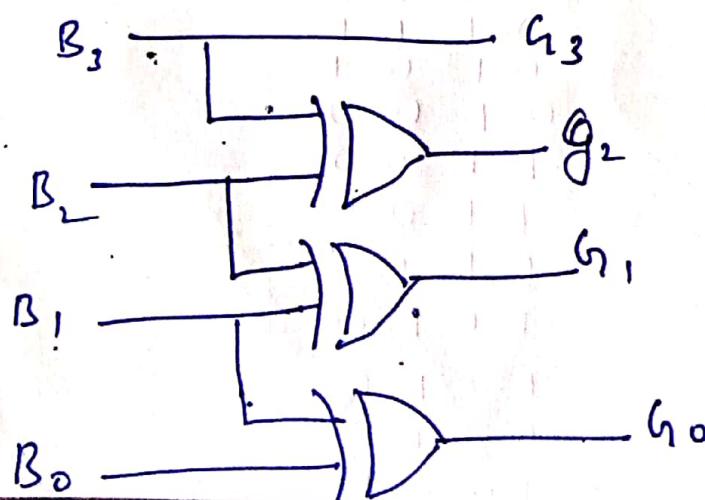
$$G_2 = B_2 \bar{B}_3 + B_3 \bar{B}_2 = B_2 \oplus B_3$$

K-map - g_3

B_3, B_2	00	01	11	10
00	0	0	0	0
01	0	0	0	0
11	1	1	1	1
10	1	1	1	1

$$G_3 = B_3$$

corresponding circuit



Converting Gray code to Binary

Let $B_0, B_1, B_2 \& B_3$ be the Binary no. $B_0 = LR B, B_3 = MSB$.

Let $g_0, g_1, g_2 \& g_3$ be the Gray code. $G_0 = LRB \quad G_3 = MSB$

Gray code.

Binary

$g_3 \quad g_2 \quad g_1 \quad g_0$	$B_3 \quad B_2 \quad B_1 \quad B_0$
0 0 0 0	0 0 0 0
0 0 0 1	0 0 0 1
0 0 1 0	0 0 1 1
0 0 1 1	0 0 1 0
0 1 0 0	0 1 1 1
0 1 0 1	0 1 1 0
0 1 1 0	0 1 0 0
0 1 1 1	0 1 0 1
1 0 0 0	1 1 1 1
1 0 0 1	1 1 1 0
1 0 1 0	1 1 0 0
1 0 1 1	1 1 0 1
1 1 0 0	1 0 0 0
1 1 0 1	1 0 0 1
1 1 1 0	1 0 1 1
1 1 1 1	1 0 1 0

K-map for B_0

B_0	g_1, g_0	00	01	11	10
g_3, g_2	00	0	1	0	1
00	0	1	0	1	0
01	1	0	0	0	0
11	0	0	0	1	0
10	1	0	1	0	0

$$\begin{aligned}
 B_0 &= \cancel{g_3} \cancel{g_2} \cancel{g_1} g_0 + \cancel{g_3} \cancel{g_2} g_1 \bar{g}_0 + \cancel{g_3} g_2 \cancel{g_1} \bar{g}_0 + \cancel{g_3} g_2 g_1 g_0 + g_3 \bar{g}_2 \bar{g}_1 \bar{g}_0 \\
 &\quad + g_3 g_2 \bar{g}_1 g_0 + g_3 g_2 \bar{g}_1 \bar{g}_0 + g_3 g_2 g_1 \bar{g}_0 \\
 &= \cancel{g_3} \cancel{g_2} (\cancel{g_1} g_0 + g_1 \bar{g}_0) + \cancel{g_3} g_2 (\cancel{g_1} \bar{g}_0 + g_1 g_0) + g_3 \bar{g}_2 (\cancel{g_1} \bar{g}_0 + g_1 g_0) \\
 &\quad + g_3 g_2 (\cancel{g_1} g_0 + g_1 \bar{g}_0) \\
 &= g_3 \bar{g}_1 (g_0 \oplus g_1) + \cancel{g_3} g_2 (g_1 \oplus g_0) + g_3 \bar{g}_2 (g_0 \oplus g_1) + \\
 &\quad g_3 g_2 (g_0 \oplus g_1) \\
 &= (g_0 \oplus g_1) (g_2 \oplus g_3) + (g_0 \oplus g_1) (g_2 \oplus g_3)
 \end{aligned}$$

$$B_0 = g_3 \oplus g_2 \oplus g_1 \oplus g_0$$

K-map for B_1

		G ₁ , G ₂			
		00	01	11	10
G ₃ , G ₄		00	0 0	1 1	
00		0 0	1 1		
01		1 1	0 0		
11		0 0	1 1		
10		1 1	0 0		

K-map for B_2

		G ₁ , G ₂			
		00	01	11	10
G ₃ , G ₄		00	0 0	0 0	
00		0 0	0 0	0 0	
01		1 1	1 1	1 1	
11		0 0	0 0	0 0	
10		1 1	1 1	1 1	

$$B_1 = \overline{g}_3 \overline{g}_2 g_1 + \overline{g}_3 g_2 \overline{g}_1 + g_3 \overline{g}_2 \overline{g}_1 + g_3 \overline{g}_2 \overline{g}_1$$

$$\overline{g}_3 (\overline{g}_2 g_1 + g_2 \overline{g}_1) + g_3 (g_2 \overline{g}_1 + \overline{g}_2 \overline{g}_1)$$

$$g_3 (g_2 \oplus g_1) + \overline{g}_3 (g_2 \ominus g_1)$$

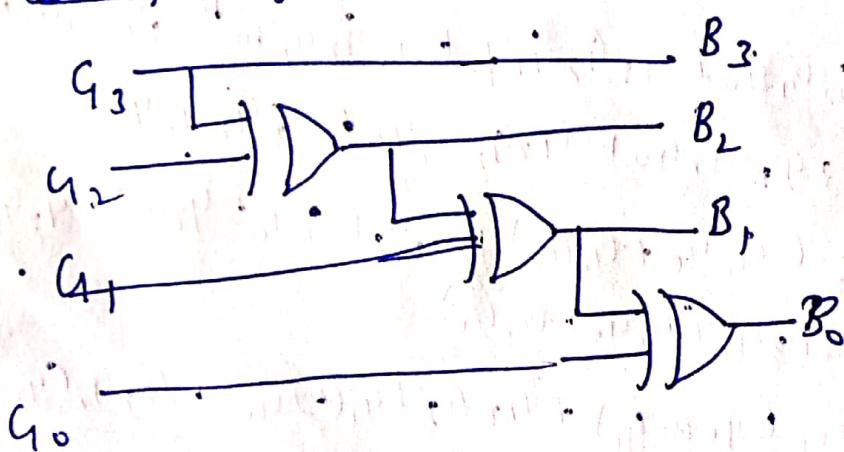
$$B_1 = g_3 \oplus g_2 \oplus g_1$$

K-map for B_3

		G ₁ , G ₂			
		00	01	11	10
G ₃ , G ₄		00	0 0	0 0	0 0
00		0 0	0 0	0 0	0 0
01		0 0	0 0	0 0	0 0
11		1 1	1 1	1 1	1 1
10		1 1	1 1	1 1	1 1

$$B_3 = g_3$$

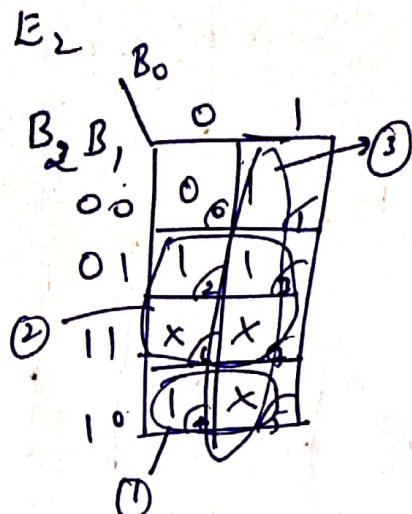
corresponding circuit



3 Bit Binary to Excess-3 code conversion

An Excess 3 - code as can be predicted from its name Excess of '3' of the Binary

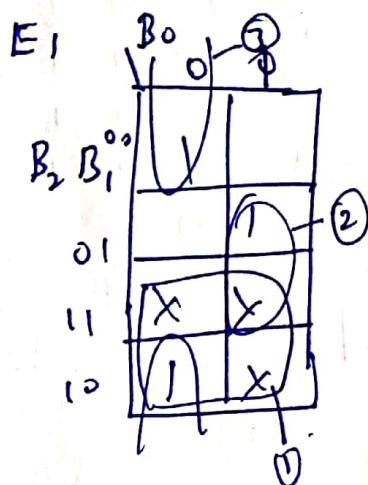
B_2	B_1 , B_0	E_2	E_1	E_0
0	0 0	0	1	1
0	0 1	0	0	0
0	1 0	1	0	1
0	1 1	1	1	0
1	0 0	1	1	1
1	0 1	x	x	x
1	1 0	x	x	x
1	1 1	x	x	x



$$E_2 = B_2 \bar{B}_1 + B_1 + B_0$$

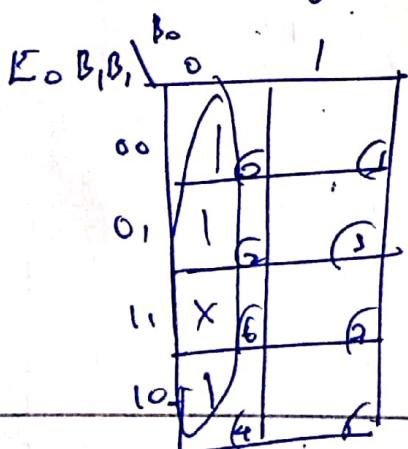
$$E_2 = B_0 + B_1 + B_2 \text{ (by you)}$$

take cells 4, 5, 6, 7 (bottom row)



$$E_1 = B_2 + B_1 B_0 + \bar{B}_0 \bar{B}_1$$

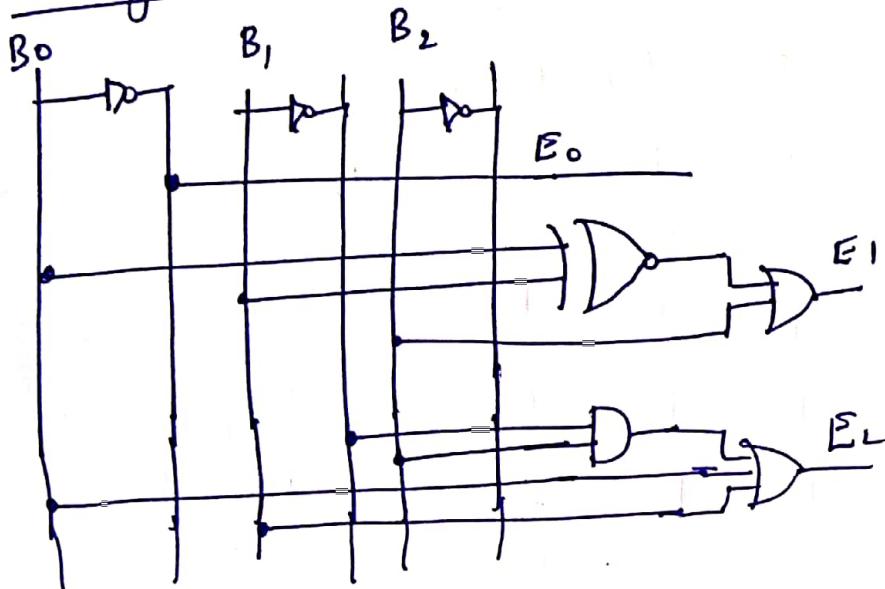
$$E_1 = B_2 + B_0 \oplus B_1$$



$$E_0 = \bar{B}_0$$

corresponding circuit

Binary to Excess



$$E_0 = \overline{B_0}$$

$$E_1 = B_2 + \overline{B_0} \oplus B_1$$

$$E_2 = B_0 + B_1 + \overline{B_1} B_2 \text{ or } B_0 + B_1 + B_2$$

Assignment:
Design 4-Bit Binary to Excess 3-code converter

$$E_3 = B_2 (B_0 + B_1) + B_3$$

$$E_2 = \overline{B_2} (B_0 + B_1) + \overline{B_0} \overline{B_1} B_2$$

$$E_1 = \overline{B_0} \oplus B_1$$

$$E_0 = \overline{B_0}$$

Parity Bit Generator

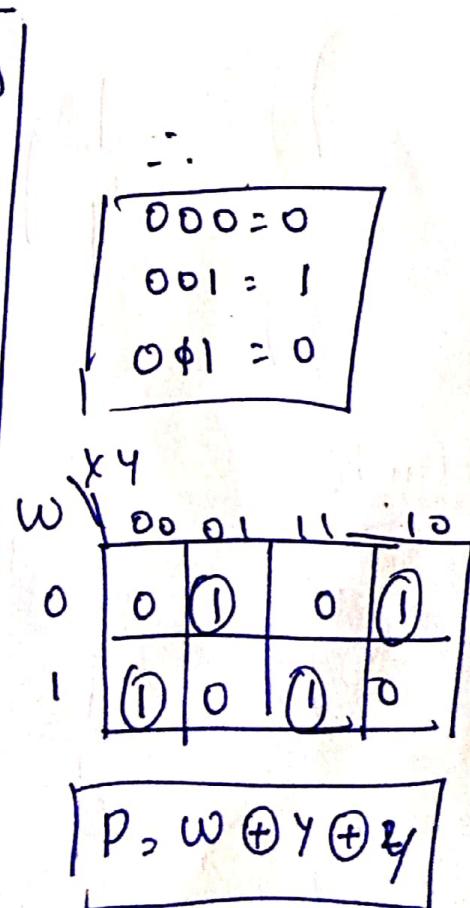
There are two types of parity bit generator based on the type of parity bit being generated.

Even parity generator generates an even parity bit
Similarly, odd parity generator generates an odd parity bit

Even Parity Generator

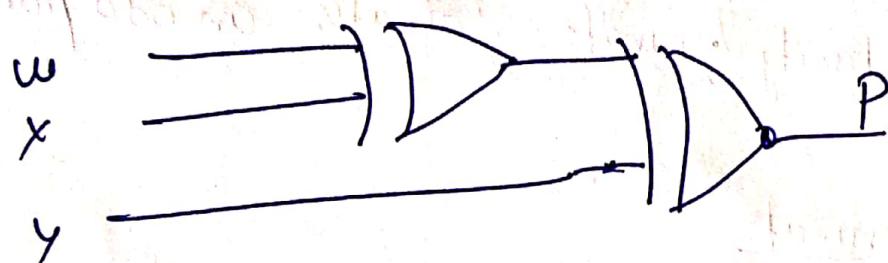
Now let us implement an even parity generator for a 3-bit binary I/p WXY it generates an even parity bit 'P'. if odd number of ones present in the I/p. then the even parity bit should be '1' so that the resultant word contains even no. of ones. for the other combination of I/p's even parity 'P' should be '0'.

Binary WXY	Even parity 'P'
0 0 0	0
0 0 1	1
0 1 0	1
0 1 1	0
1 0 0	1
1 0 1	0
1 1 0	0
1 1 1	1



Odd Parity :

if even no of ones present in the I/p then the odd parity bit, $P = 1$, others $P = 0$.

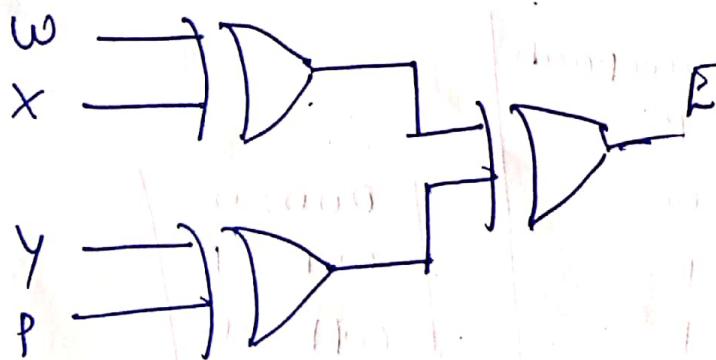


Assignment:

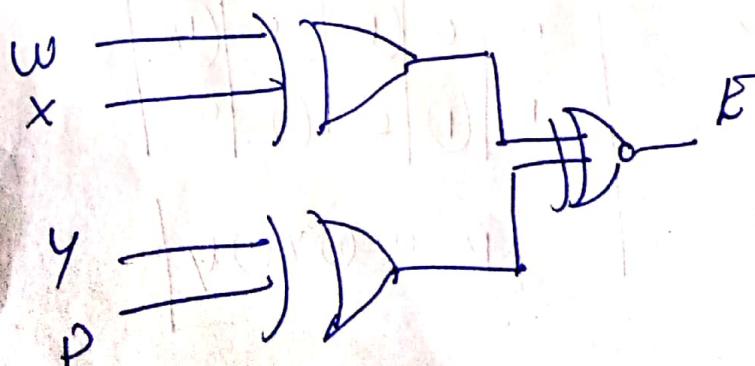
Parity checker: for 4-bit

$$\text{Even Parity} \quad P = \bar{w}\bar{x} + w\bar{x} + \bar{w}x + w\bar{x}$$

$$w \oplus x \oplus y \oplus P$$



odd Parity:



Comparators :-

The comparison of two numbers in an operations that determines if one number is greater than, less than, or equal to the other number. A comparator compares two numbers, A, B & it determines their relative magnitudes. The outcome of the comparison is indicated by three variables that indicate whether $A > B$, $A = B$, $A < B$.

A_0	B_0	$Y_{A=B}$	$Y_{A>B}$	$Y_{A<B}$
0	0	1	0	0
0	1	0	0	1
1	0	0	1	0
1	1	1	0	0

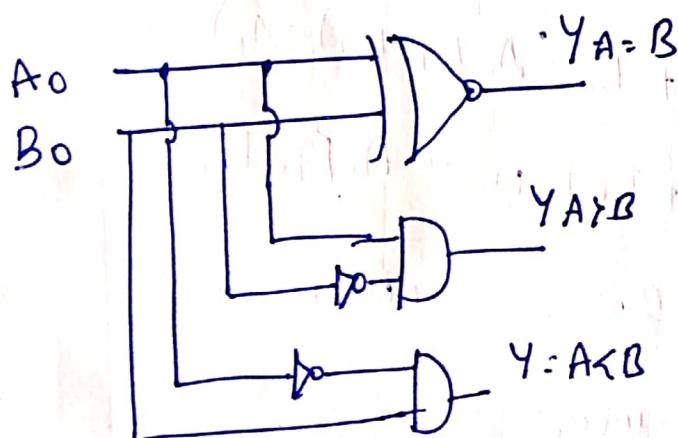
$$Y_{A=B} = \overline{A_0} \overline{B_0} + A_0 B_0$$

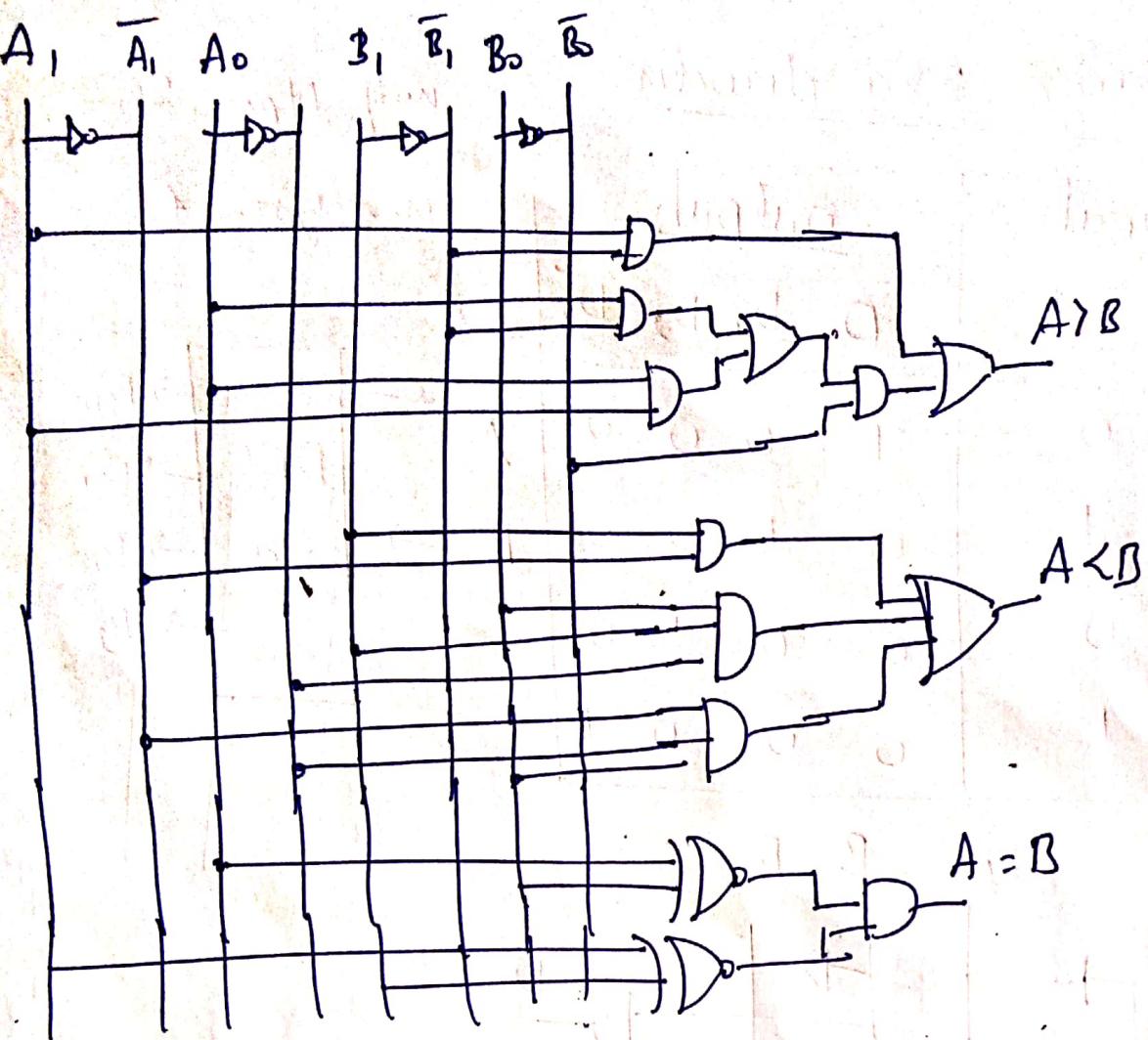
$$Y_{A>B} = A_0 \overline{B_0}$$

$$Y_{A<B} = \overline{A_0} B_0$$

Combinational logic

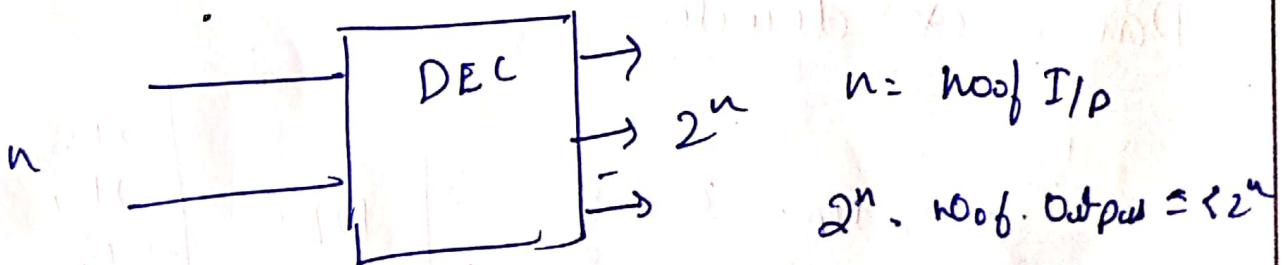
Corresponding circuit





Decoder :-

A decoder is a combinational logic circuit that converts binary information from n I/P lines to a maximum of 2^n unique O/P lines.

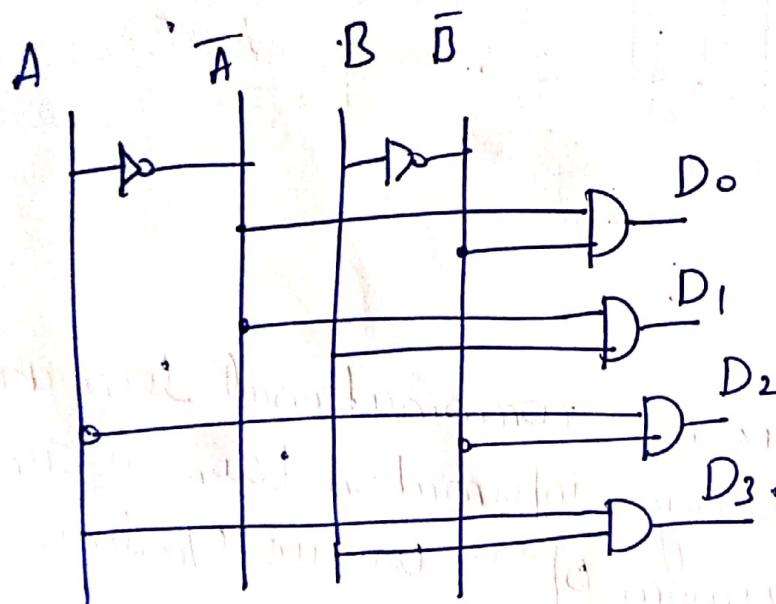
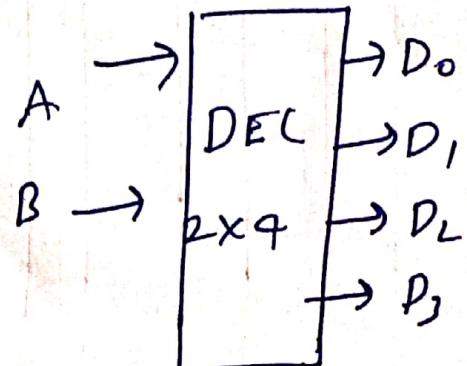


Design 2x4 decoder

Inputs		Outputs			
A	B	D ₀	D ₁	D ₂	D ₃
0	0	1	0	0	0
0	1	0	1	0	0
1	0	0	0	1	0
1	1	0	0	0	1

no of I/O's = 2

no of O/P's = 4

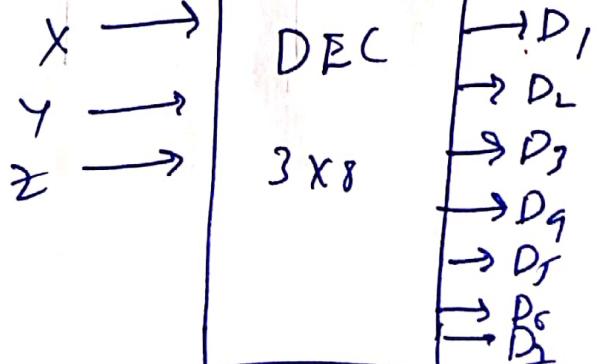


Design 3x8 decoder

Sol.

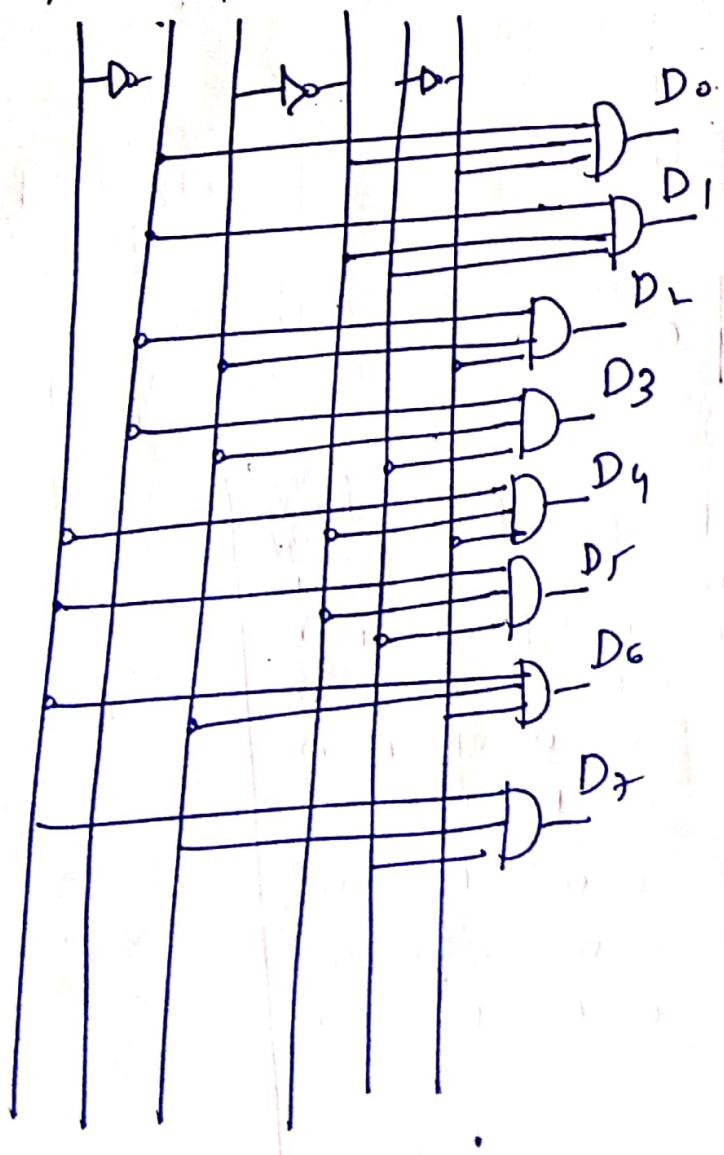
no of I/O's = 3

no of O/P's = 8



I/P		O/P								
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

x \bar{x} y \bar{y} z \bar{z}

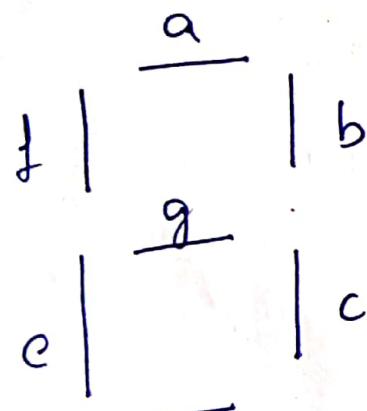


BCD to 7-Segment decoder

Most often Seven-Segment displays are used to display the digits in digital watches, calculators, clocks.

A display decoder is used to convert a BCD or binary code into a 7-Segment code. It generally has 4-IOP lines 7 O/P lines.

7-Segment Display Decoder circuit design



<u>Digit</u>	<u>A B C D</u>	<u>d a b c d e f g</u>
0	0 0 0 0	0 0 0 0 0 0 1
1	0 0 0 1	1 0 0 1 1 1 1
2	0 0 1 0	0 0 1 0 0 1 0
3	0 0 1 1	0 0 0 0 0 1 1 0
4	0 1 0 0	1 0 0 1 1 0 0
5	0 1 0 1	0 1 0 0 1 0 0
6	0 1 1 0	0 1 0 0 0 0 0 0
7	0 1 1 1	0 0 0 1 1 1 1
8	1 0 0 0	0 0 0 0 0 0 0 0
9	1 0 0 1	0 0 0 0 0 1 0 0

$$a = F_1(A, B, C, D) = \sum m(0, 2, 3, 5, 7, 9)$$

$$b = F_2(A, B, C, D) = \sum m(0, 1, 2, 3, 4, 7, 8, 9)$$

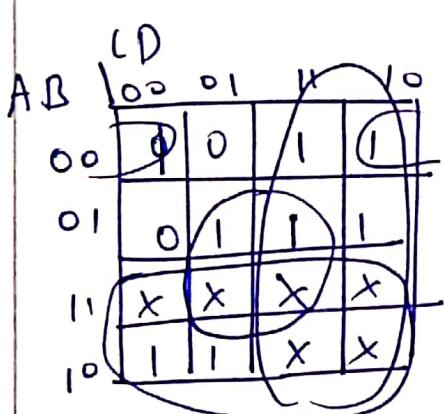
$$c = F_3(A, B, C, D) = \sum m(0, 1, 3, 4, 5, 6, 7, 8, 9)$$

$$d = F_4(A, B, C, D) = \sum m(0, 2, 3, 5, 6, 8)$$

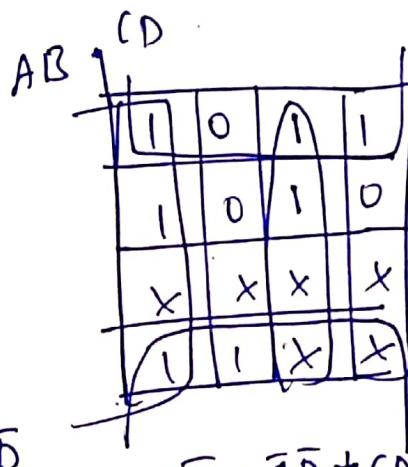
$$e = F_5(A, B, C, D) = \sum m(0, 2, 6, 8)$$

$$f = F_6(A, B, C, D) = \sum m(0, 4, 5, 6, 8, 9)$$

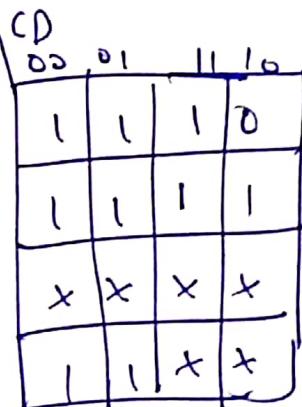
$$g = F_7(A, B, C, D) = \sum m(2, 3, 4, 5, 6, 8, 9)$$



$$a = A + C + BD + \underline{B}\bar{D}$$



$$b = \bar{B} + \bar{C}\bar{D} + CD$$



$$c = B + \bar{C} + D$$

Similarly,

$$d = \bar{B}\bar{D} + \bar{C}\bar{D} + B\bar{C}D + \bar{B}C + A$$

$$e = \bar{B}\bar{D} + C\bar{D}$$

$$f = A + \bar{C}\bar{D} + B\bar{C} + B\bar{D}$$

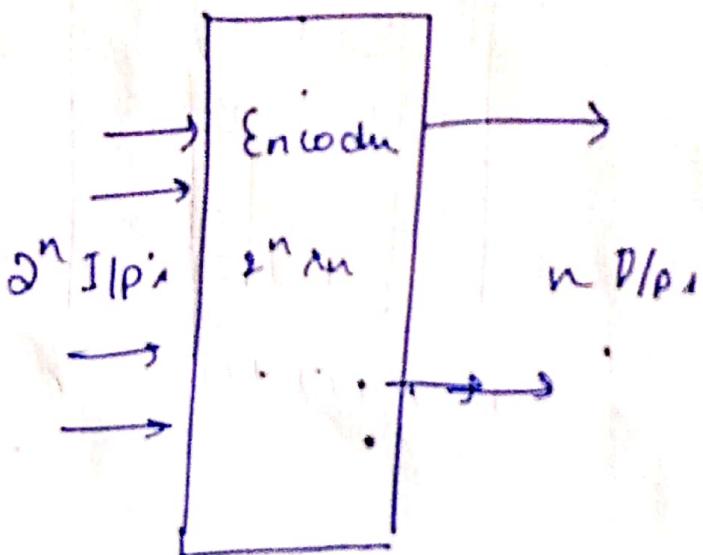
$$g = \bar{B}C + \bar{C}\bar{D} + B\bar{C} + \underline{A}$$

construct above Equations using Basic gates

Encoder:

An Encoder is a device circuit, software program, algorithm or person that converts information from one format or code to another. The purpose of Encoder is Standardization, Speed / Secrecy, Security, or saving space by shrinking size.

If a device D/p code has fewer bits than the I/p code has the device is usually called an Encoder.



Encoder Block diagram.

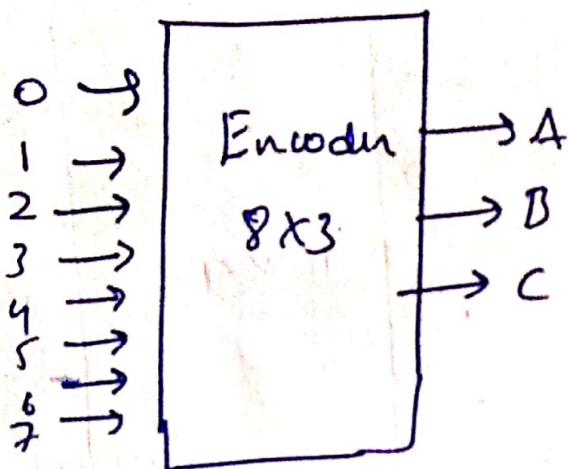
$$\text{no of Inputs} = < 2^n$$

$$\text{no of O/p's} = n$$

Design 8x3 Encoder

No of I/p's = 8×2^3

No of O/p's = 3



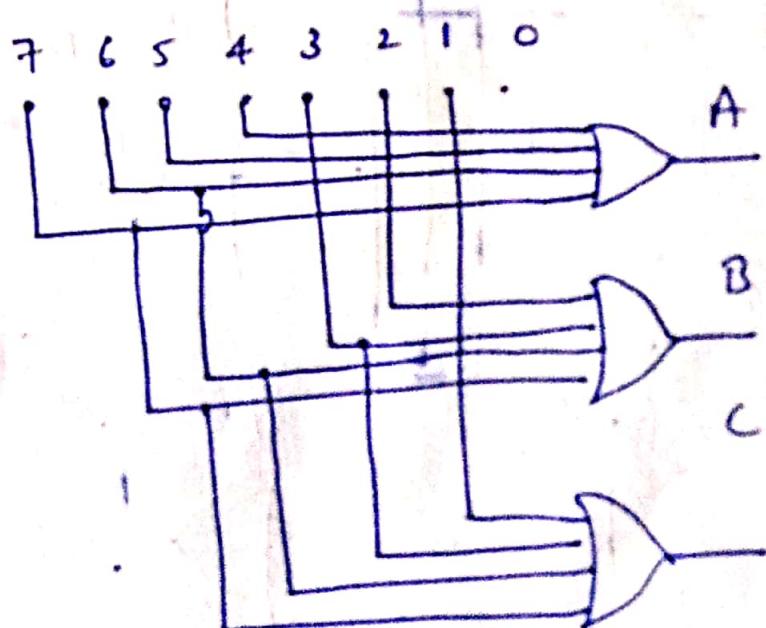
Octal to Binary Encoder

Inputs	O/P's
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

$$A = \sum m(4, 5, 6, 7)$$

$$B = \sum m(2, 3, 6, 7)$$

$$C = \sum m(1, 3, 5, 7)$$



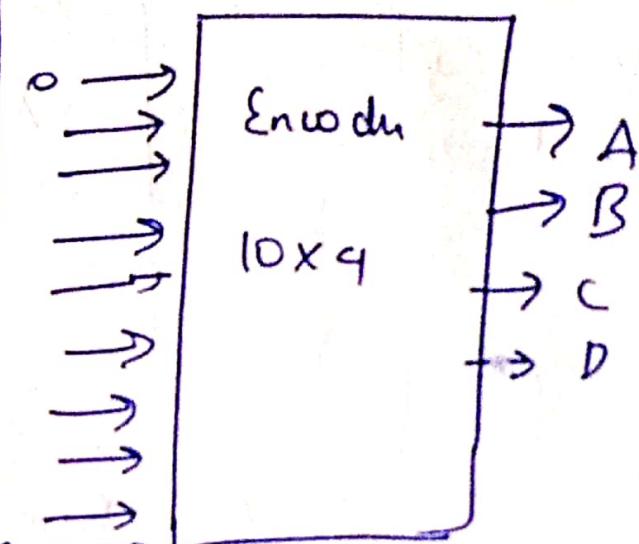
Design a Decimal to BCD Encoder

No of I/P = 10

No of O/P = 4

Input	Output			
	A	B	C	D
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



From T.T

$$A = \sum m(8, 9)$$

$$B = \sum m(4, 5, 6, 7)$$

$$C = \sum m(2, 3, 6, 7)$$

$$D = \sum m(1, 3, 5, 7, 9)$$

9 8 7 6 5 4 3 2 1 0

