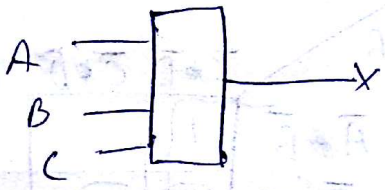
Module III

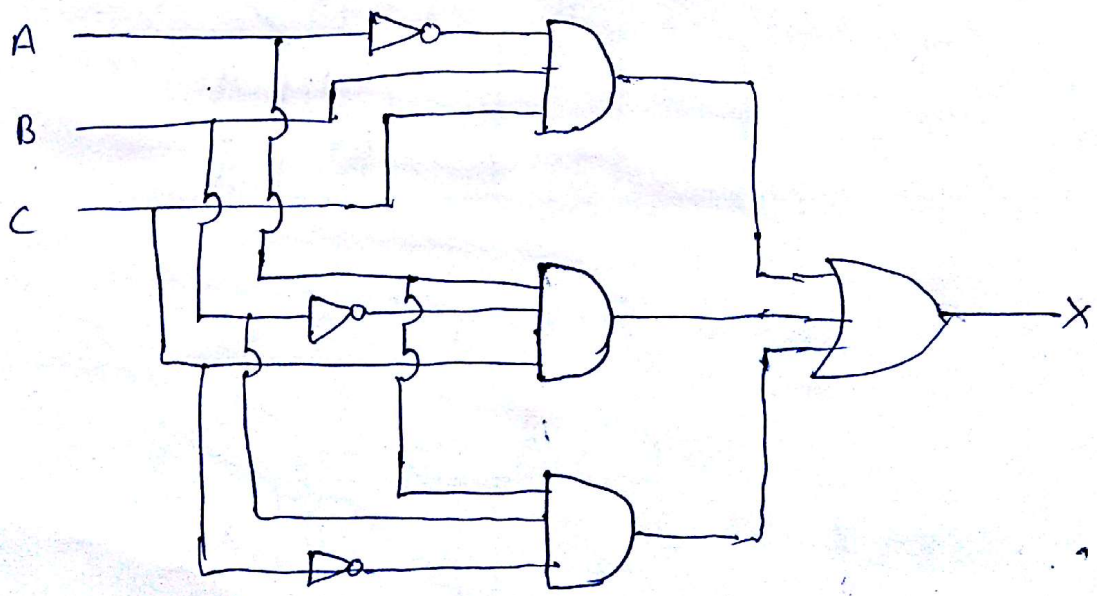
Combination Circuits

Design a logic ckt to implement the operation specified in the following table.

Input A	B	C	Output X
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	0



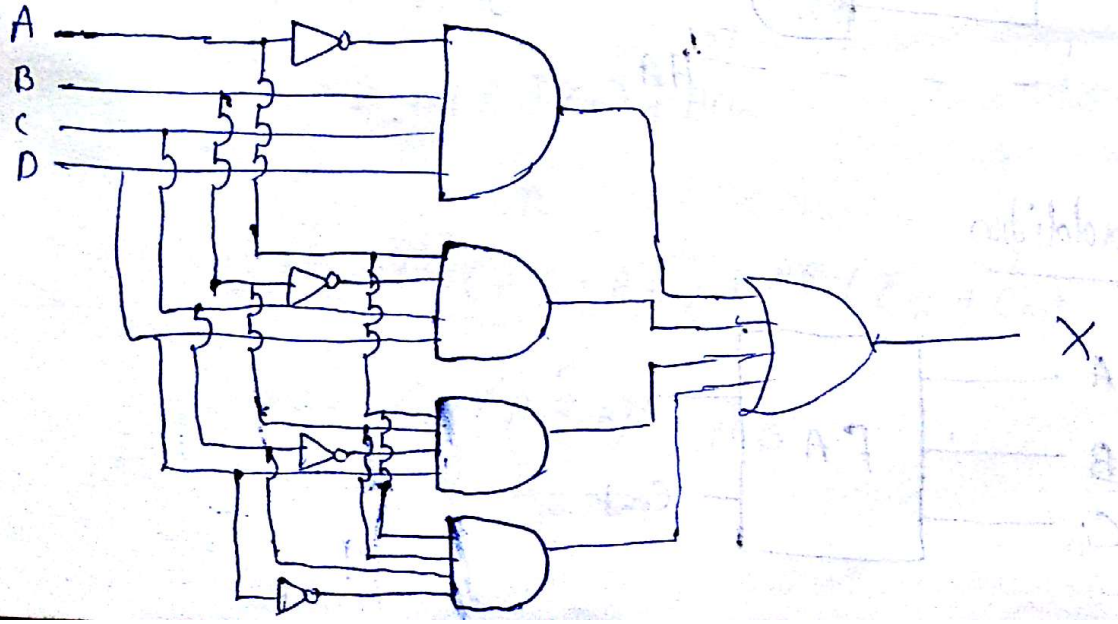
$$X = \bar{A}BC + \bar{A}\bar{B}C + AB\bar{C}$$



Q. Develop a logic ckt with 4 input variables that will produce a '1' output when any three and only three input variables are ones.

input				Output
A	B	C	D	X
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	0
0	1	0	0	0
0	1	0	1	0
0	1	1	0	0
0	1	1	1	1
1	0	0	0	0
1	0	0	1	0
1	0	1	0	0
1	0	1	1	1
1	1	0	0	0
1	1	0	1	1
1	1	1	0	1
1	1	1	1	0

$$X = \bar{A}BCD + A\bar{B}CD + AB\bar{C}D + ABC\bar{D}$$

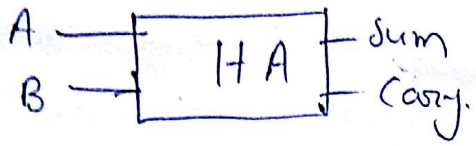


ADDERS

```
   1 ← Carry
   1 0 1
 + 1 0 1
 ---------
 1 0 1 0
```

Carry out.

2-bit addition

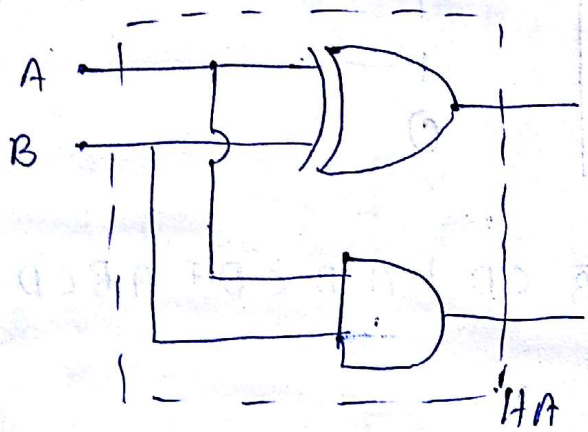


HA → half adder.

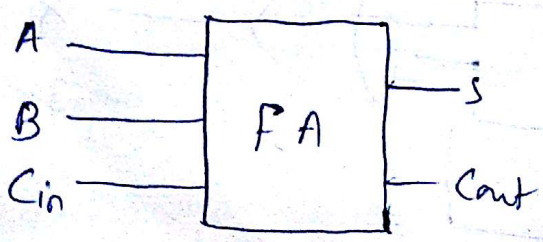
Input		Output	
A	B	S	C
0	0	0	0
0	1	1	0
1	0	1	0.
1	1	0	1

Here Sum = $\bar{A}B + A\bar{B} = A \oplus B$

$C = AB$.



3-bit addition



A	B	C_{in}	S	C_{out}
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

$$Sum = \bar{A}\bar{B}C_{in} + \bar{A}B\bar{C}_{in} + A\bar{B}\bar{C}_{in} + ABC_{in}$$

$$C_{out} = \bar{A}BC_{in} + A\bar{B}C_{in} + AB\bar{C}_{in} + ABC_{in}$$

$$Sum = C_{in}(\bar{A}\bar{B} + AB) + \bar{C}_{in}(\bar{A}B + A\bar{B})$$

$$= C_{in}(A \odot B) + \bar{C}_{in}(A \oplus B)$$

$$= C_{in}\bar{Z} + \bar{C}_{in}Z$$

$$= C_{in} \oplus Z = A \oplus B \oplus C_{in}$$

$$C_{out} = \bar{A}BC_{in} + AB\bar{C}_{in} + AC_{in}(B + \bar{B})$$

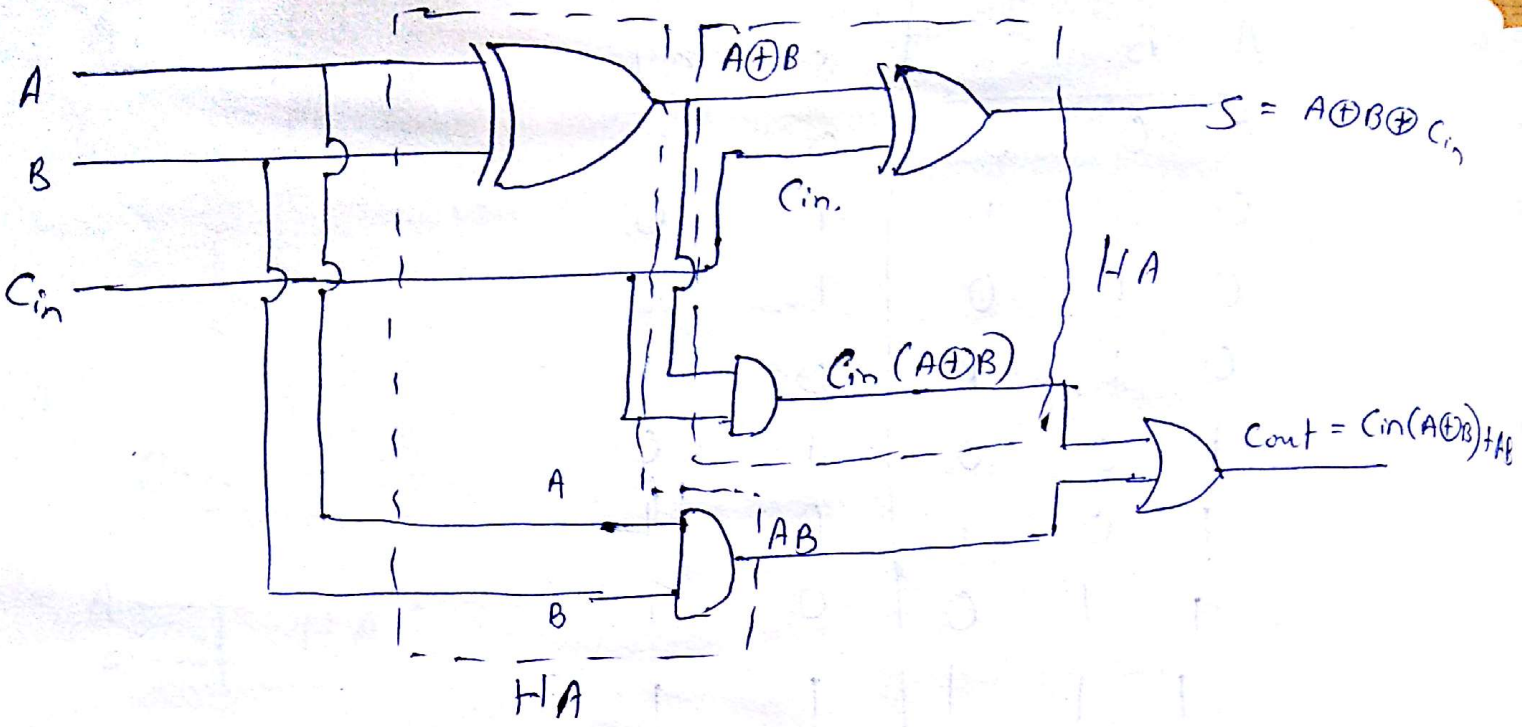
$$= B(\bar{A}C_{in} + A\bar{C}_{in}) + AC_{in}$$

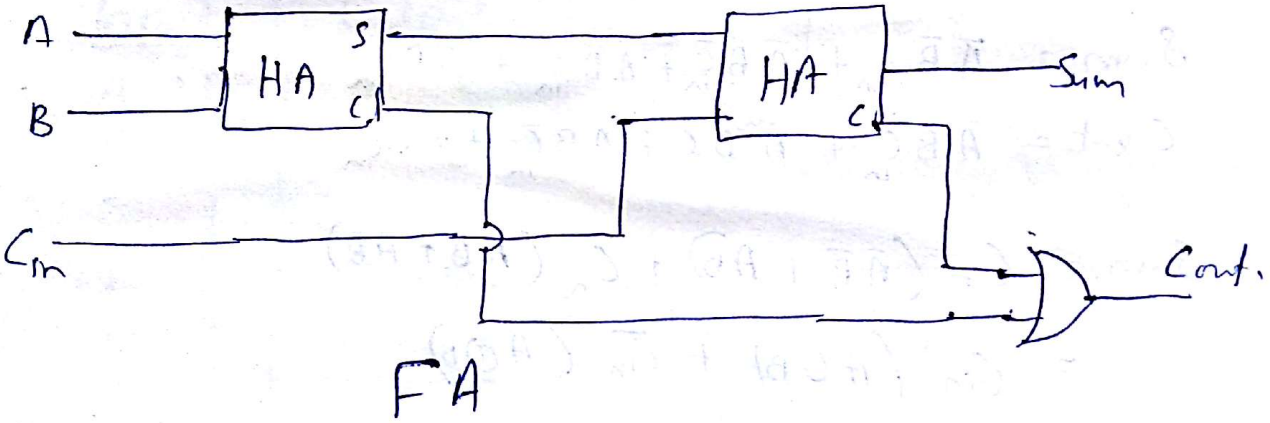
$$= B(A \oplus C_{in}) + AC_{in}$$

OR

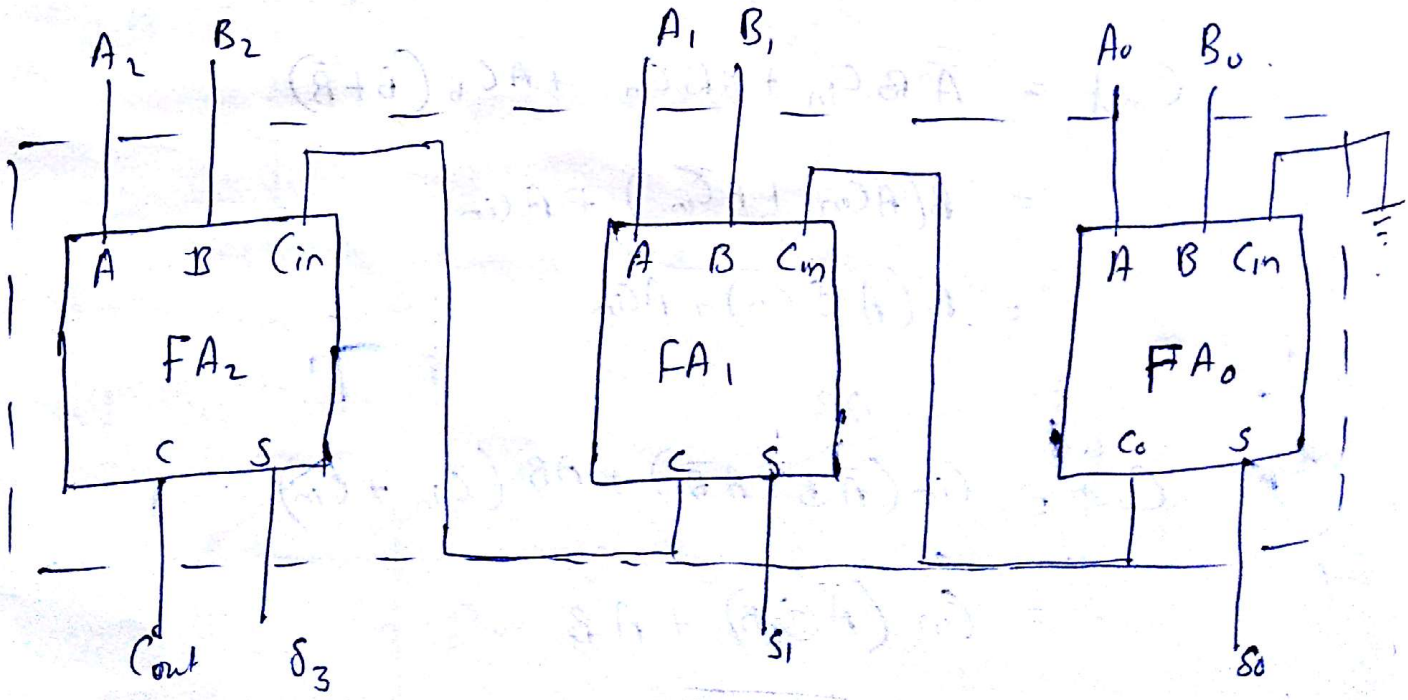
$$C_{out} = C_{in}(\bar{A}B + A\bar{B}) + AB(\bar{C}_{in} + C_{in})$$

$$= C_{in}(A \oplus B) + AB$$

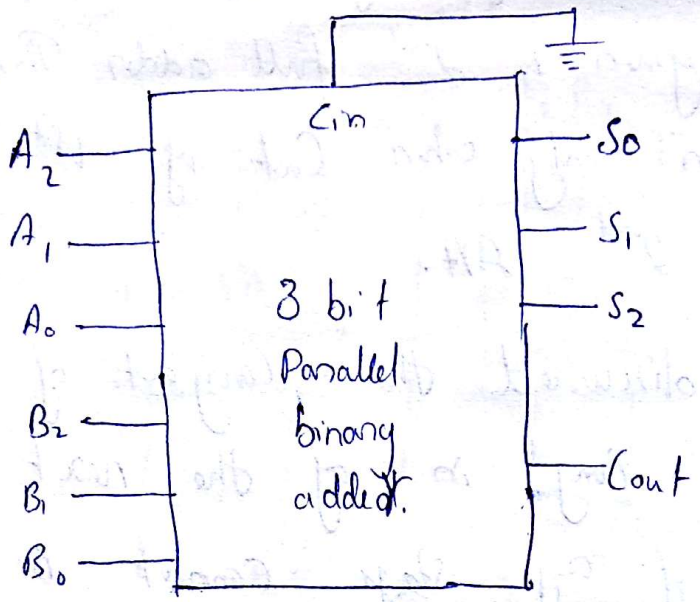




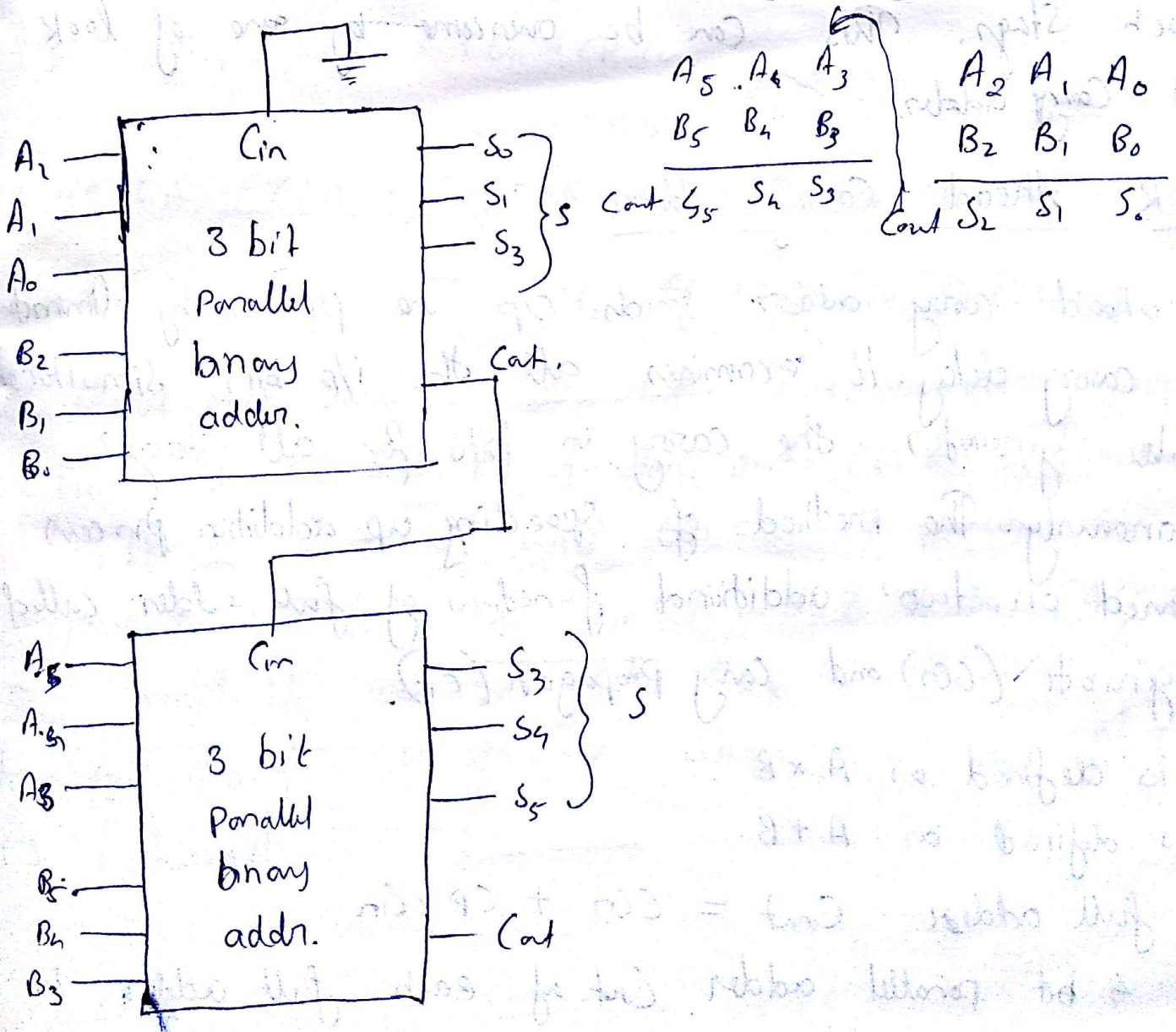
Parallel binary adder.



Block diagram of 3 bit Parallel binary added.



Q. Show how two 3 bit parallel adder can be connected to form a 6 bit parallel adder.



Propagation delay - The first full adder performs addition.
→ The carry out is given to 2nd full adder. The 2nd full adder performs addition only when Cout of 1st FA is given to Cin of 2nd FA.

In parallel adders discussed the carry out of each stage is connected to the carry in of the next stage. The Sum and carry outs of any stage cannot be produced until the i/p carry occurs, which leads to time delay. The total delay vary depending on carries produced by each stage. This can be overcome by use of look ahead carry adder.

Look Ahead Carry Adder.

Look ahead carry adder speeds up the process by eliminating ripple carry delay. It examines all the i/p bits simultaneously and also generates the carry in bits for all stages simultaneously. The method of speeding up addition process is based on two additional functions of full adder called carry generate (CG) and carry propagate (CP).

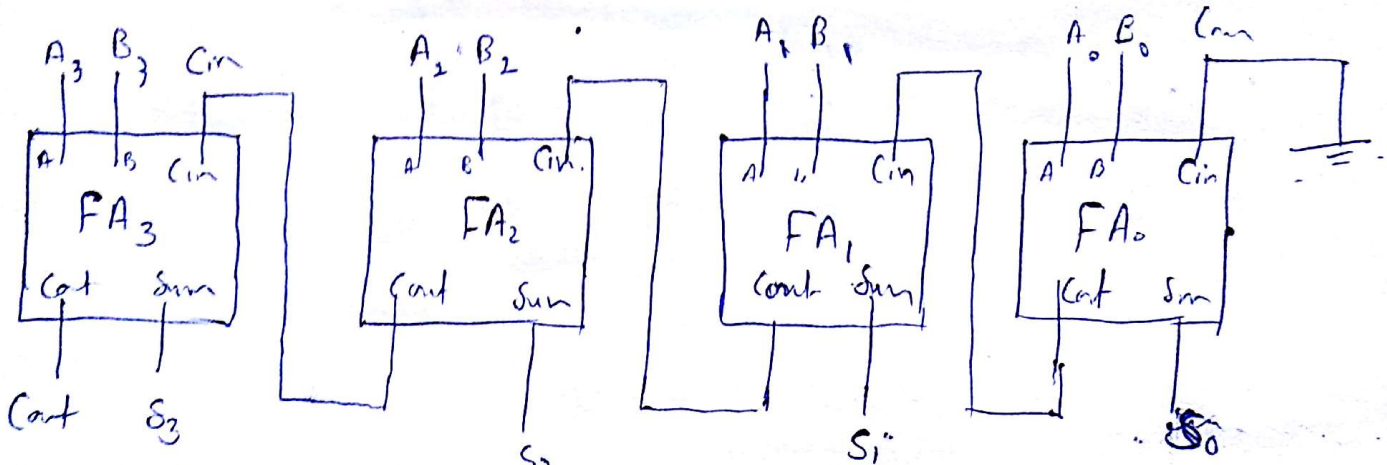
CG is defined as $A \times B$

CP is defined as $A + B$.

For a full adder $Cout = CG + CP \times Cin$

For a 4 bit parallel adder Cout of each full adder is depended on its CG, CP and its Cin. CG and CP for

each stage are immediately available as soon as input bits are applied to each adder.



$CG_3 = A_3 B_3$ $\quad CG_2 = A_2 B_2$ $\quad CG_1 = A_1 B_1$ $\quad CG_0 = A_0 B_0$

$CP_3 = A_3 + B_3$ $\quad CP_2 = A_2 + B_2$ $\quad CP_1 = A_1 + B_1$ $\quad CP_0 = A_0 + B_0$

1st FA, $Cout_0 = CG_0 + CP_0 Cin_0$

2nd FA, $Cout_1 = CG_1 + CP_1 \cdot Cin_1$

$= CG_1 + CP_1 (CG_0 + CP_0 Cin_0)$

3rd FA, $Cout_2 = CG_2 + CP_2 \times [CG_1 + CP_1 (CG_0 + CP_0 Cin_0)]$

4th FA, $Cout_3 = CG_3 + CP_3 \times [CG_2 + CP_2 \times (CG_1 + CP_1 (CG_0 + CP_0 Cin_0))]$

Cout of each stage is dependent only in initial carry in Cin_0. Its CG and CP functions and CG and CP functions of preceeding stages. Since CG and CP functions can be expressed in forms of i/p's A and B to the FA, all o/ps are immediately available and there is no need for carry to ripple through all of these stages before the final stage is achieved.

SUBTRACTORS

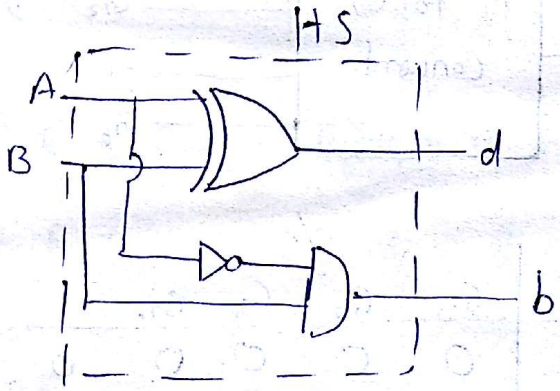
Half Subtractors

Inputs		Outputs	
A	B	d	b
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0

$\sim b$
10
-1
1

$d = \bar{A}B + A\bar{B} = A \oplus B$

$b = \bar{A}B$



Full Subtractor

Inputs			Outputs	
A	B	bin	d	bout
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

$d = \bar{A}\bar{B}b_{in} + \bar{A}B\bar{b}_{in} + A\bar{B}\bar{b}_{in} + AB b_{in}$

$= \bar{A}(\bar{B}b_{in} + B\bar{b}_{in}) + A(\bar{B}\bar{b}_{in} + Bb_{in})$

$= \bar{A}(B \oplus b_{in}) + A(B \odot b_{in})$

$B \oplus b_{in} = Z$

$\therefore d = \bar{A}Z + A\bar{Z} = A \oplus Z = A \oplus B \oplus b_{in}$

$b_{out} = \bar{A}\bar{B}b_{in} + \bar{A}B\bar{b}_{in} + \bar{A}Bb_{in} + AB b_{in}$

$= \bar{A}(\bar{B}b_{in} + B\bar{b}_{in}) + B(\bar{A}b_{in} + Ab_{in})$

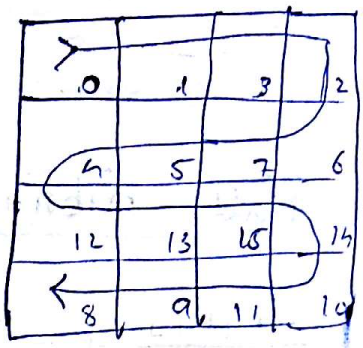
$= \bar{A}B + (A \odot B)b_{in}$

Code Converters

Code converters are logic ~~Converters~~ circuits whose inputs are bit patterns representing no.s in one code and whose outputs are corresponding representation in a different code.

Q. Design and implement a 4 bit binary to Gray Code converter.

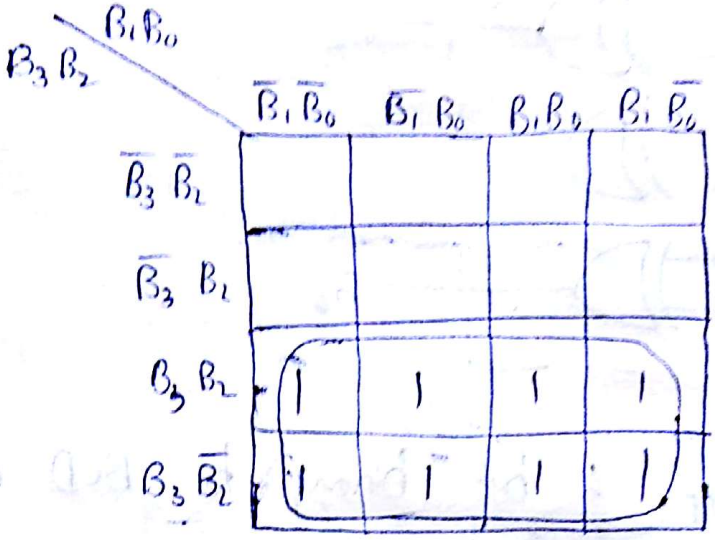
	i/p				o/p				
	B_3	B_2	B_1	B_0	G_3	G_2	G_1	G_0	
0	0	0	0	0	0	0	0	0	0
1	0	0	0	1	0	0	0	1	1
2	0	0	1	0	0	0	1	1	3
3	0	0	1	1	0	0	1	0	2
4	0	1	0	0	0	1	1	0	6
5	0	1	0	1	0	1	1	1	7
6	0	1	1	0	0	1	0	1	5
7	0	1	1	1	0	1	0	0	4
8	1	0	0	0	1	1	0	0	12
9	1	0	0	1	1	1	0	1	13
10	1	0	1	0	1	1	1	1	15
11	1	0	1	1	1	1	1	0	14
12	1	1	0	0	1	0	1	0	10
13	1	1	0	1	1	0	1	1	11
14	1	1	1	0	1	0	0	1	9
15	1	1	1	1	1	0	0	0	8



$C_3 = B_3 \bar{B}_2 \bar{B}_1 \bar{B}_0 + B_3 \bar{B}_2 \bar{B}_1 B_0 + B_3 \bar{B}_2 B_1 \bar{B}_0 + B_3 \bar{B}_2 B_1 B_0 +$

$B_3 B_2 \bar{B}_1 \bar{B}_0 + B_3 B_2 \bar{B}_1 B_0 + B_3 B_2 B_1 \bar{B}_0 + B_3 B_2 B_1 B_0$

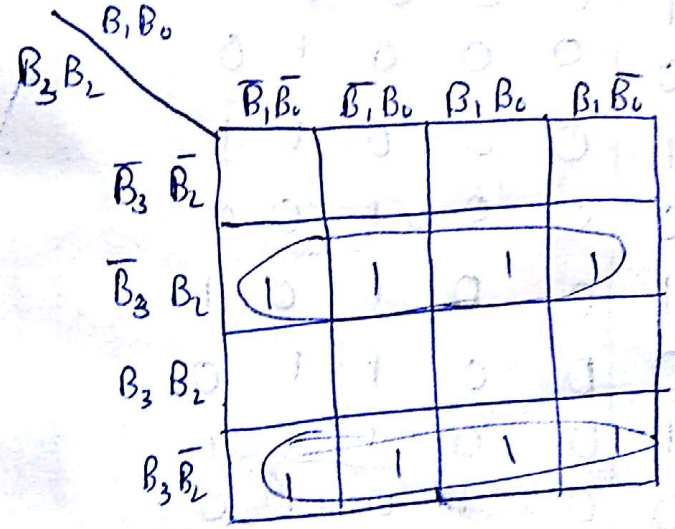
Using K-MAP



$\therefore C_3 = B_3$

$C_2 = \bar{B}_3 B_2 \bar{B}_1 B_0 + \bar{B}_3 B_2 \bar{B}_1 B_0 + \bar{B}_3 B_2 B_1 \bar{B}_0 + \bar{B}_3 B_2 B_1 B_0 + B_3 \bar{B}_2 \bar{B}_1 B_0$

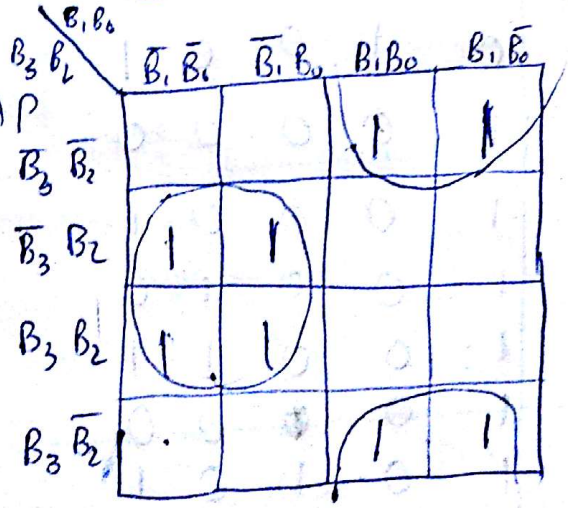
$+ B_3 \bar{B}_2 B_1 \bar{B}_0 + B_3 \bar{B}_2 B_1 B_0 + B_3 B_2 \bar{B}_1 \bar{B}_0$



$C_2 = \bar{B}_3 B_2 + B_3 \bar{B}_2$

$= B_3 \oplus B_2$

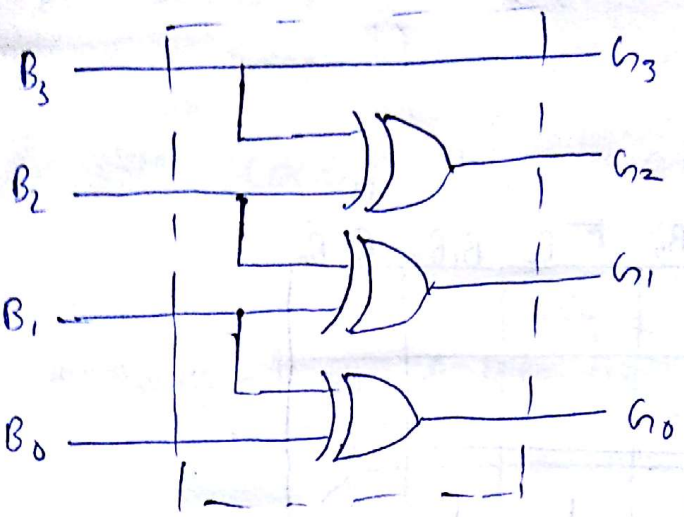
C_1 to K-MAP



$C_1 = B_2 \bar{B}_1 + \bar{B}_2 B_1$

$= B_2 \oplus B_1$

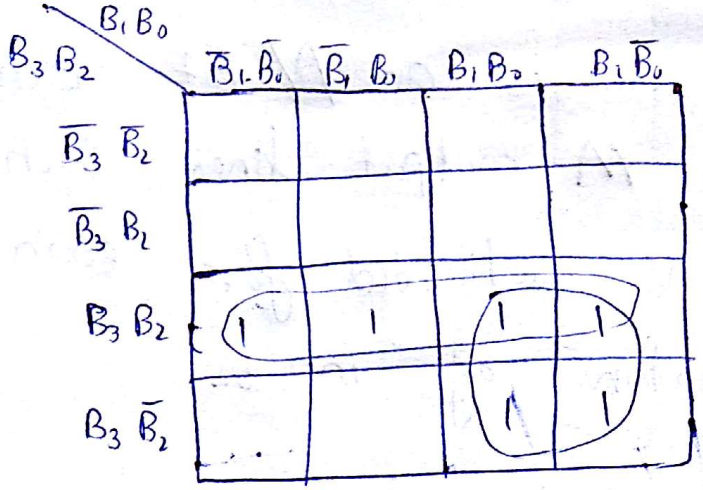
Similarly $G_0 = B_1 \oplus B_0$.



Q. Design and Implement 4 bit binary to BCD Converter.

i/p				o/p.				
B_3	B_2	B_1	B_0	A	B	C	D	E
0	0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	0	1
0	0	1	0	0	0	0	1	0
0	0	1	1	0	0	0	1	1
0	1	0	0	0	0	1	0	0
0	1	0	1	0	0	1	0	1
0	1	1	0	0	0	1	1	0
0	1	1	1	0	0	1	1	1
1	0	0	0	0	1	0	0	0
1	0	0	1	0	1	0	0	1
1	0	1	0	1	0	0	0	0
1	0	1	1	1	0	0	0	1
1	1	0	0	1	0	0	1	0
1	1	0	1	1	0	0	1	1
1	1	1	0	1	0	0	0	0
1	1	1	1	1	0	1	0	1

A, using K-MAP.



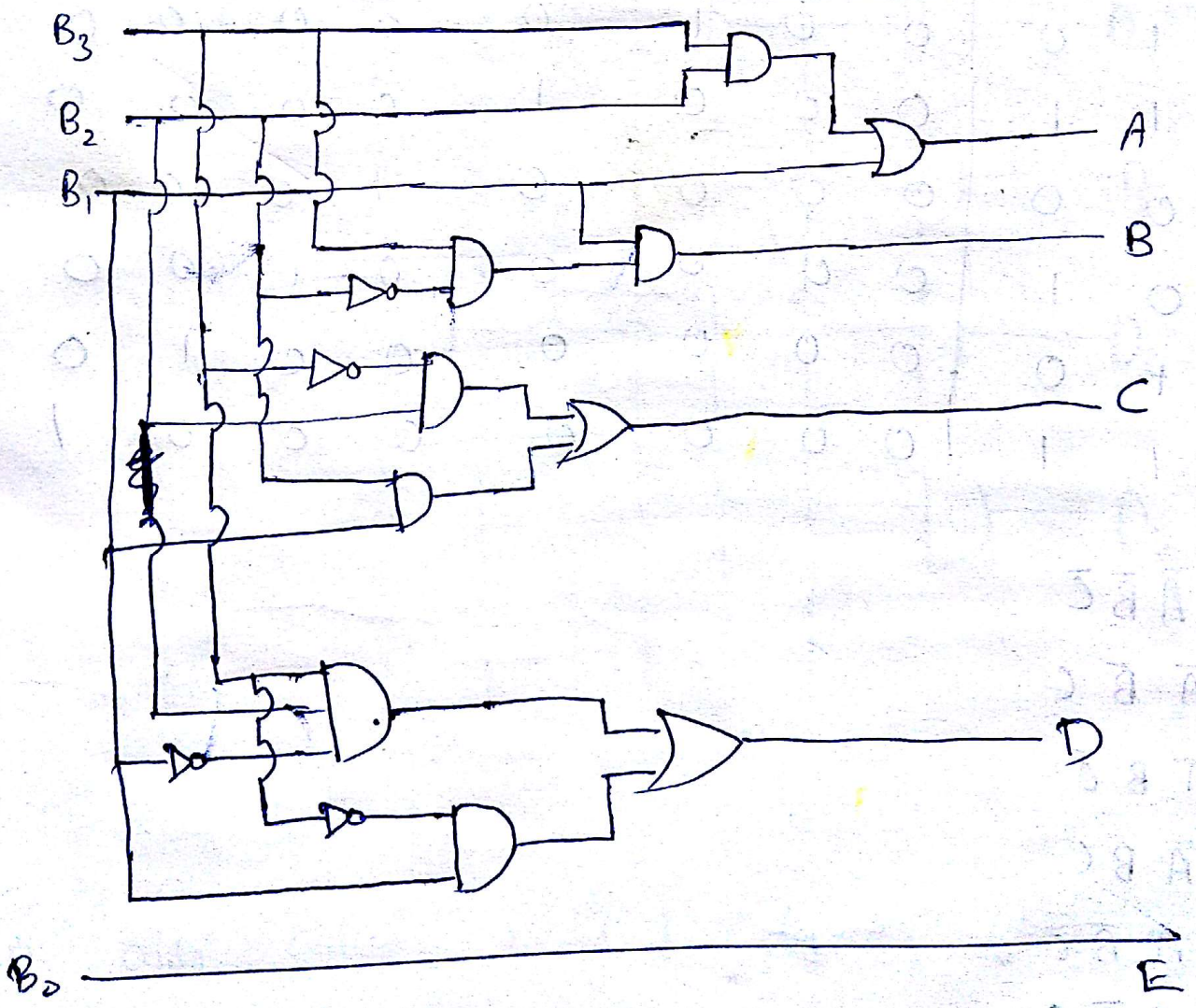
$A = B_3B_2 + B_3B_1$

Similarly $B = B_3\bar{B}_2B_1$

$C = \bar{B}_3B_2 + B_2B_1$

$D = B_3B_2\bar{B}_1 + \bar{B}_3B_1$

$E = B_0$



Decoders

Logic circuits that converts an N bit binary input code logic into M output lines such that only one output line is activated for each one of the possible combinations of inputs.

3-to-8 Decoder

3 i/ps and 8 o/ps with 7 low and 1 high.

inputs			ACTIVE HIGH							
A	B	C	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

$D_0 = \bar{A}\bar{B}\bar{C}$

$D_1 = \bar{A}\bar{B}C$

$D_2 = \bar{A}B\bar{C}$

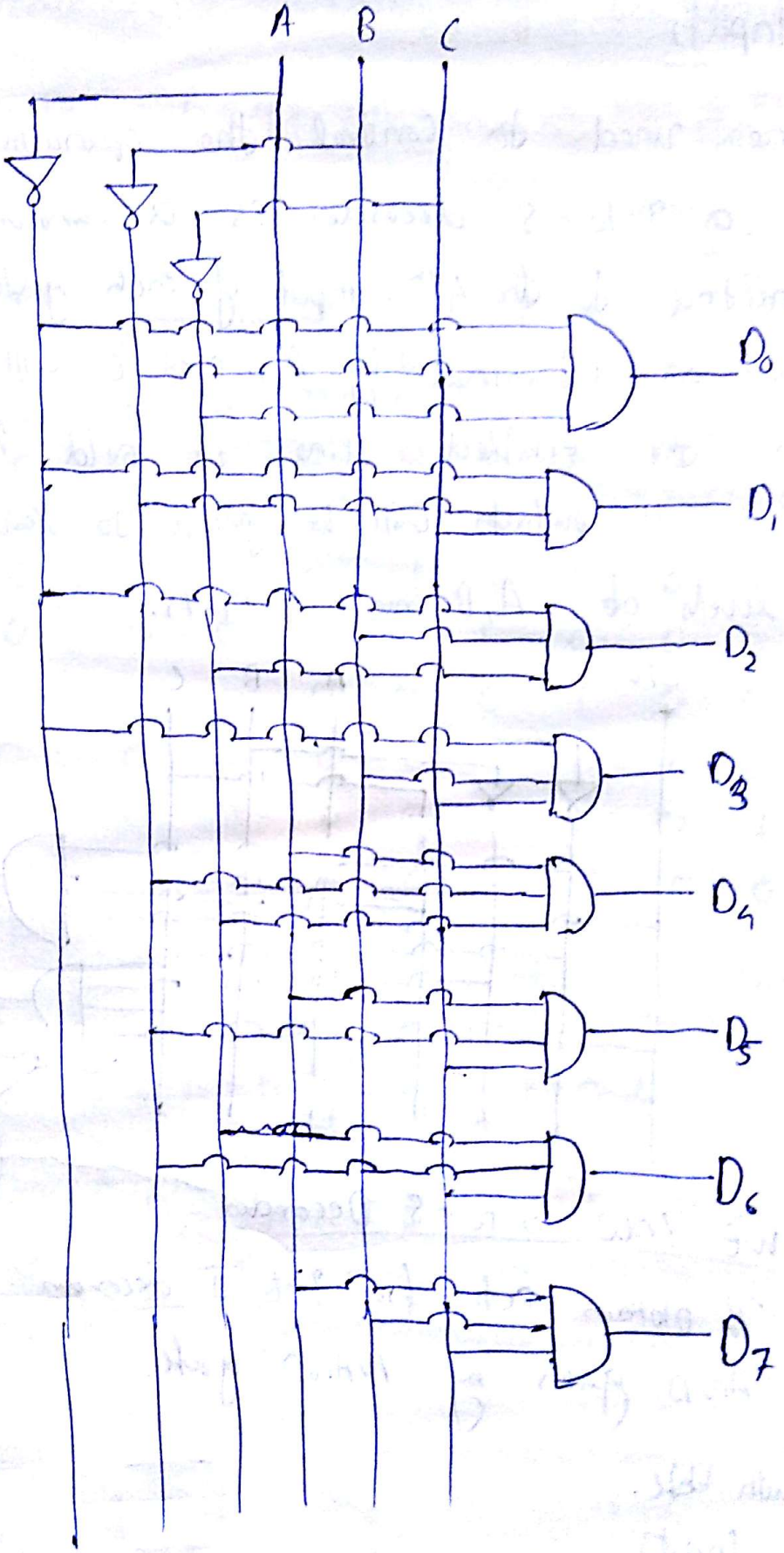
$D_3 = \bar{A}BC$

$D_4 = A\bar{B}\bar{C}$

$D_5 = A\bar{B}C$

$D_6 = AB\bar{C}$

$D_7 = ABC$



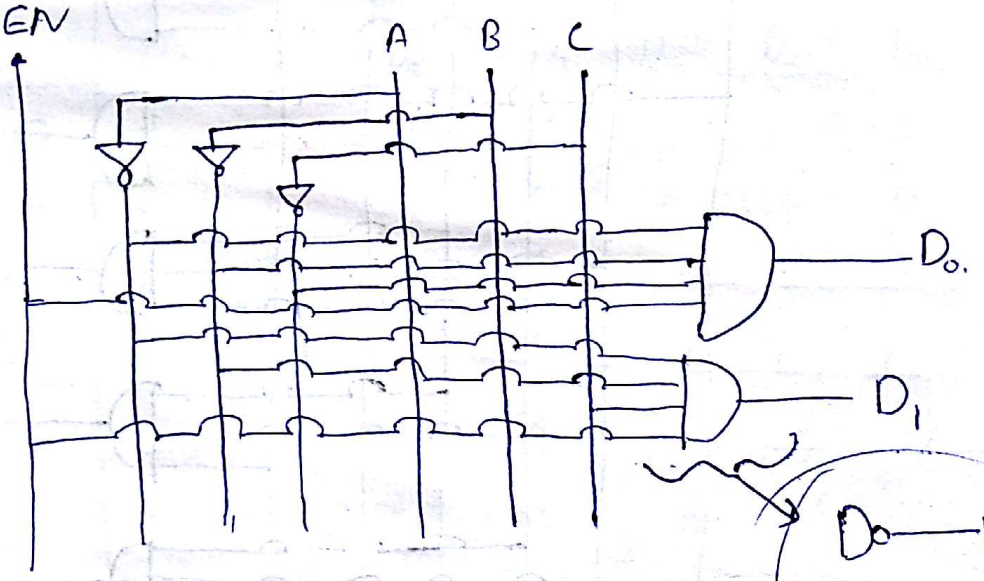
It is also called 1 of 8 decoder since one of 8 is activate.

Enable Inputs

They are used to control the operation of a decoder.

Eg:- In a 3-to-8 decoder is a common enable input is connected to the 4th input of each gate, a particular output as determined by A, B and C will go high only when the enabled line is held high. If the Enable is low all outputs will be forced to low regardless of the levels at A, B and C inputs.

Eg:-



ACTIVE LOW 3-to-8 Decoder

To obtain active low 3-to-8 decoder we replace the AND gates by NAND gate.

Eg: Truth table.

Inputs									
A	B	C	D_0	D_1	D_2	D_3	D_4	D_5	D_6	D_7
0	0	0	0	1	1	1	1	1	1	1
0	0	1	1	0	1	1	1	1	1	1
0	1	0	1	1	0	1	1	1	1	1
0	1	1	1	1	1	0	1	1	1	1

Assignment Q. Active LOW 4-to-16 / 1of16 Decorder.

4 bit BCD - to - Decimal decorder

	Inputs													
	A	B	C	D	D_0	D_1	D_2	D_3	D_4	D_5	D_6	D_7	D_8	D_9
0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
1	0	0	0	1	0	1	0	0	0	0	0	0	0	0
2	0	0	1	0	0	0	1	0	0	0	0	0	0	0
3	0	0	1	1	0	0	0	1	0	0	0	0	0	0
4	0	1	0	0	0	0	0	0	1	0	0	0	0	0
5	0	1	0	1	0	0	0	0	0	1	0	0	0	0
6	0	1	1	0	0	0	0	0	0	0	1	0	0	0
7	0	1	1	1	0	0	0	0	0	0	0	1	0	0
8	1	0	0	0	0	0	0	0	0	0	0	0	1	0
9	1	0	0	1	0	0	0	0	0	0	0	0	0	1
10	1	0	1	0										

Only 10 Combination is possible else others are invalid.
It is also called 4-to-10 / 1-of-10 decoder.

$D_0 = \bar{A}\bar{B}\bar{C}\bar{D}$

$D_1 = \bar{A}\bar{B}\bar{C}D$

$D_2 = \bar{A}\bar{B}C\bar{D}$

$D_3 = \bar{A}\bar{B}CD$

$D_4 = \bar{A}B\bar{C}\bar{D}$

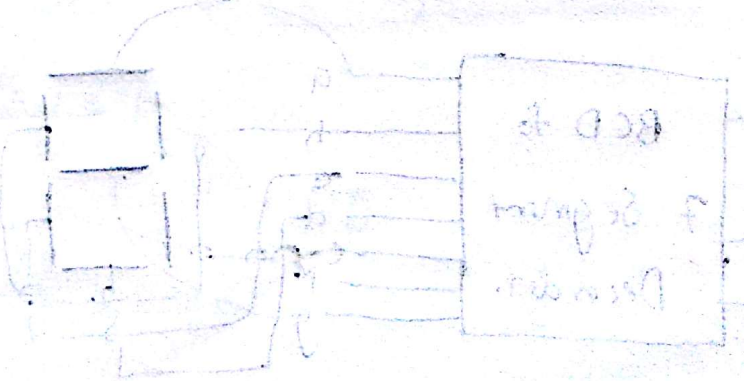
$D_5 = \bar{A}B\bar{C}D$

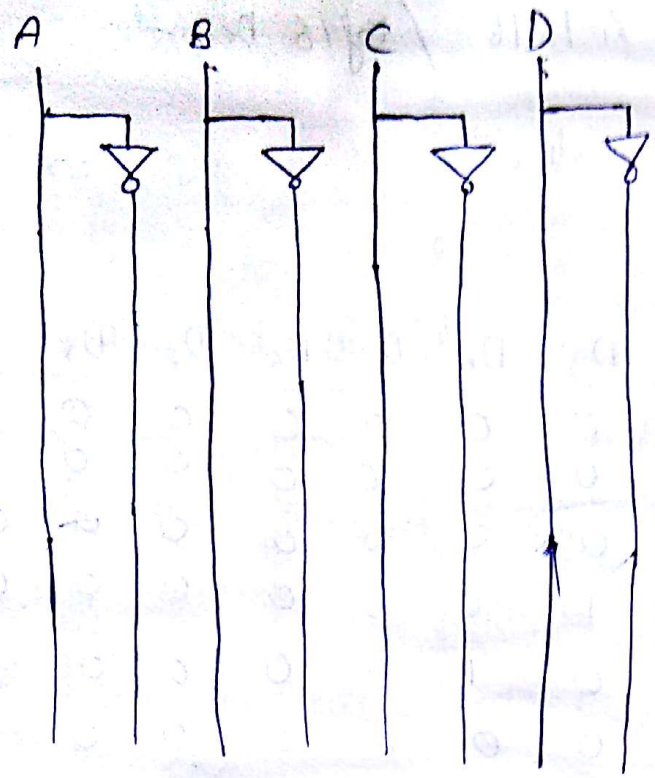
$D_6 = \bar{A}BC\bar{D}$

$D_7 = \bar{A}BCD$

$D_8 = A\bar{B}\bar{C}\bar{D}$

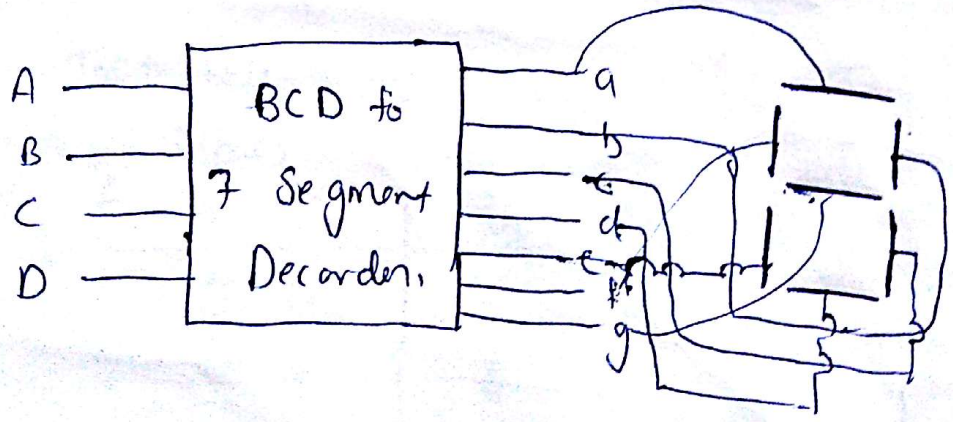
$D_9 = A\bar{B}\bar{C}D$

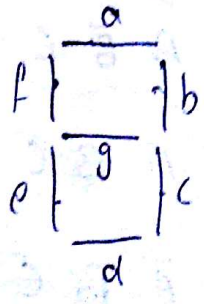




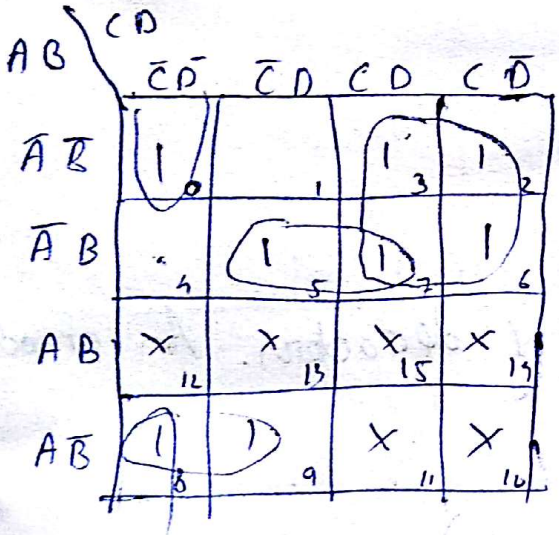
BCD to 7 segment

This accepts BCD code and provides o/p to energise 7 segment display devices inorder to produce a decimal readout. A 7 segment display is normally used for displaying any one of the decimal digits 0 to 9. It is composed of 7 light emitting sections designated as letters from a to g. By eliminating various numbers from 0 to 9 can be obtained





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



$a = A + C + B \oplus D$

$a = \bar{A}B\bar{C}\bar{D} + \bar{A}B C\bar{D} + \bar{A}B CD + \bar{A}B\bar{C}D + \bar{A}BC\bar{D} + \bar{A}BCD + A\bar{B}\bar{C}\bar{D} + A\bar{B}\bar{C}D.$

Similarly $b = \bar{B} + C\overline{\oplus}D$

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

$d = A + \overline{CD} + B \oplus (C + \bar{D})$

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

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

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

ENCODERS

They perform the reverse operation of decoders. An encoder have M inputs and N o/ps.

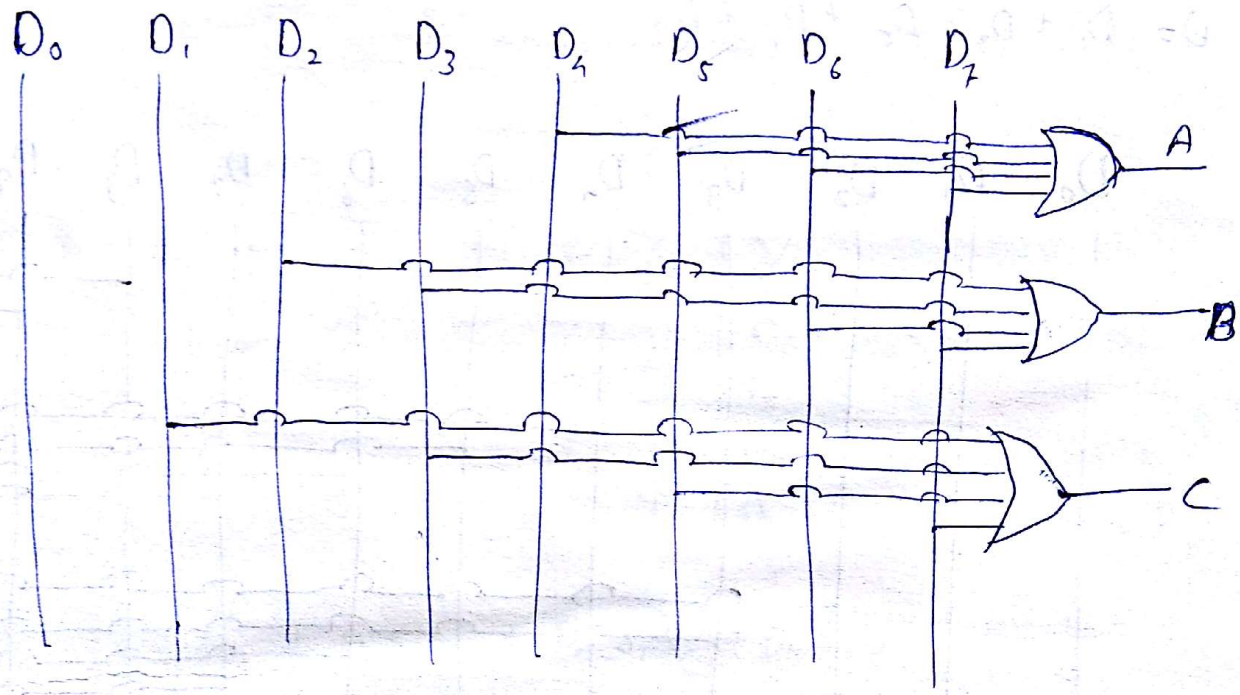
Octal - to - Binary Encoder (8 - to - 3 Encoder)

I/P	O/P A	B	C
D_0	0	0	0
D_1	0	0	1
D_2	0	1	0
D_3	0	1	1
D_4	1	0	0
D_5	1	0	1
D_6	1	1	0
D_7	1	1	1

$A = D_4 + D_5 + D_6 + D_7$

$B = D_2 + D_3 + D_6 + D_7$

$C = D_1 + D_3 + D_5 + D_7$



Decimal – to – BCD Encoder

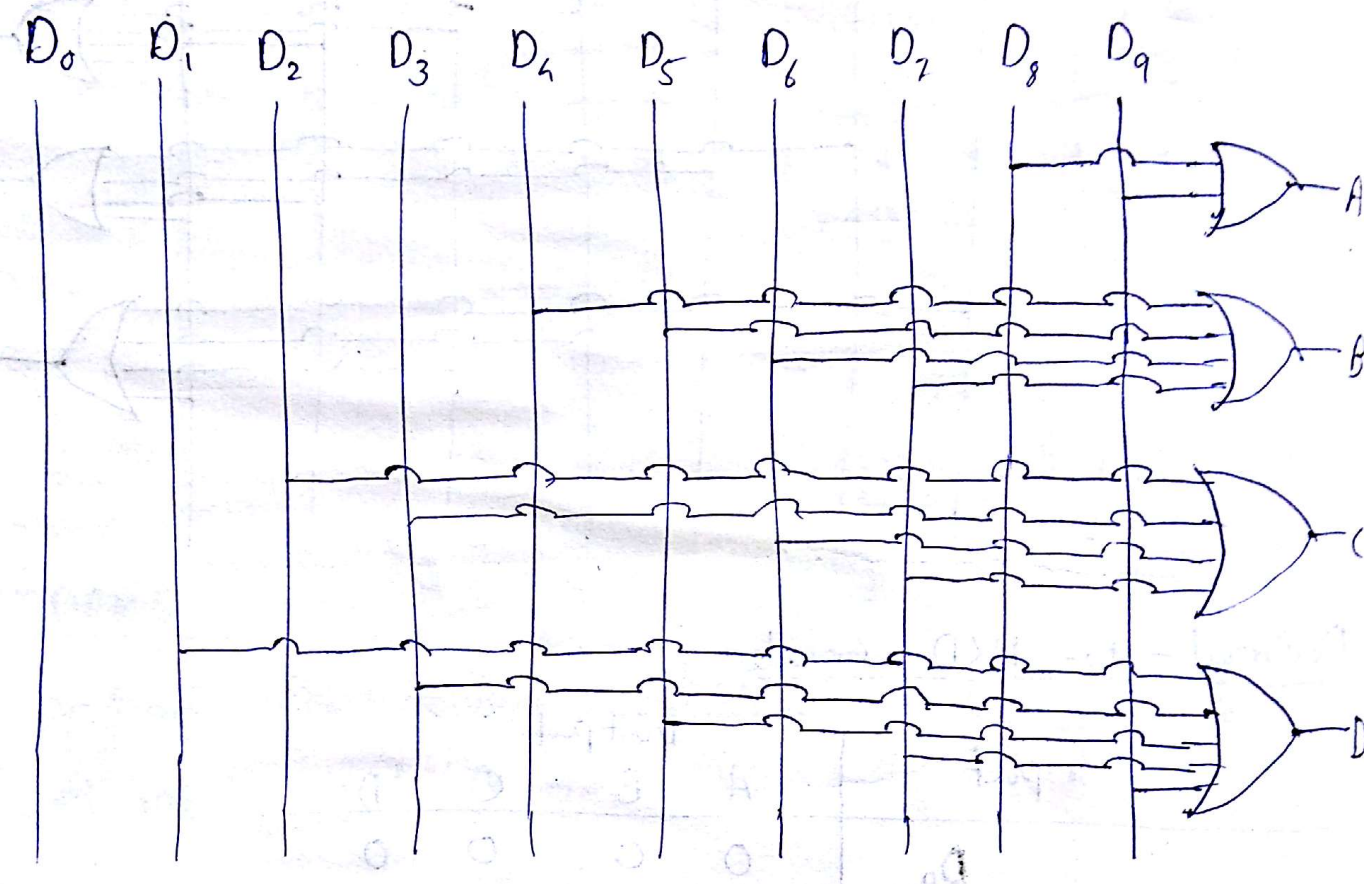
input	Output A	B	C	D
D_0	0	0	0	0
D_1	0	0	0	1
D_2	0	0	1	0
D_3	0	0	1	1
D_4	0	1	0	0
D_5	0	1	0	1
D_6	0	1	1	0
D_7	0	1	1	1
D_8	1	0	0	0
D_9	1	0	0	1

$A = D_8 + D_9$

$B = D_4 + D_5 + D_6 + D_7$

$C = D_2 + D_3 + D_6 + D_7$

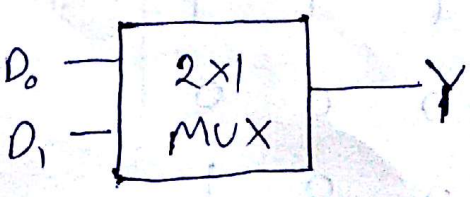
$D = D_1 + D_3 + D_5 + D_7 + D_9$



MULTIPLEXERS (MUX) / Data Selectors (Many to One)

Multiplexer is a logic ckt that excepts several data i/ps and allows only one of them at a time to get through the o/p. For N input lines, the no. of select lines m should be such that $2^m = N$.

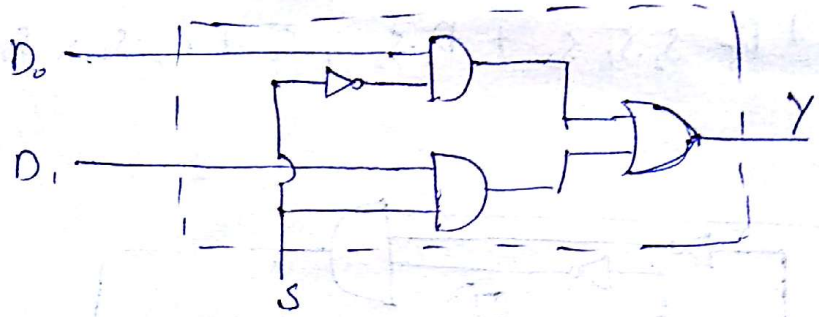
2×1 MUX,



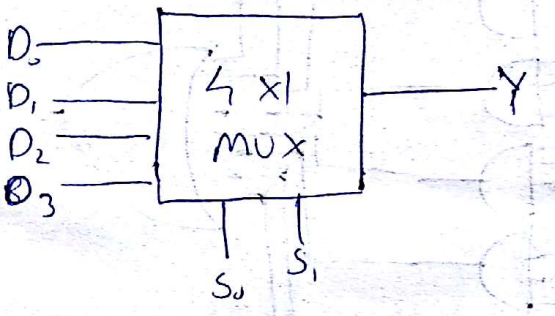
S	Y
0	D_0
1	D_1

$Y = D_0 \bar{S} + D_1 S$

2 x 1 MUX

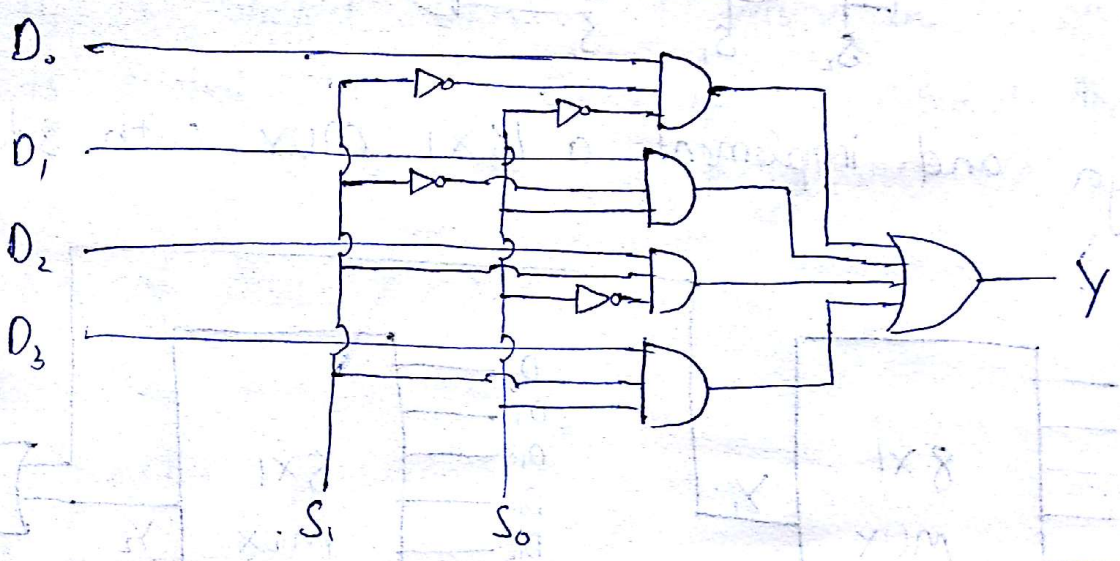


4 x 1 MUX

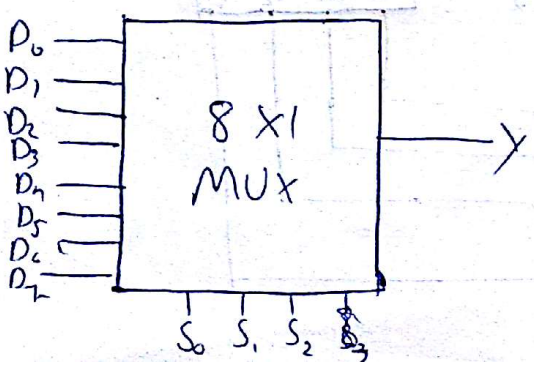


S_1	S_0	Y
0	0	D_0
0	1	D_1
1	0	D_2
1	1	D_3

$Y = D_0 \bar{S}_1 \bar{S}_0 + D_1 \bar{S}_1 S_0 + D_2 S_1 \bar{S}_0 + D_3 S_1 S_0$

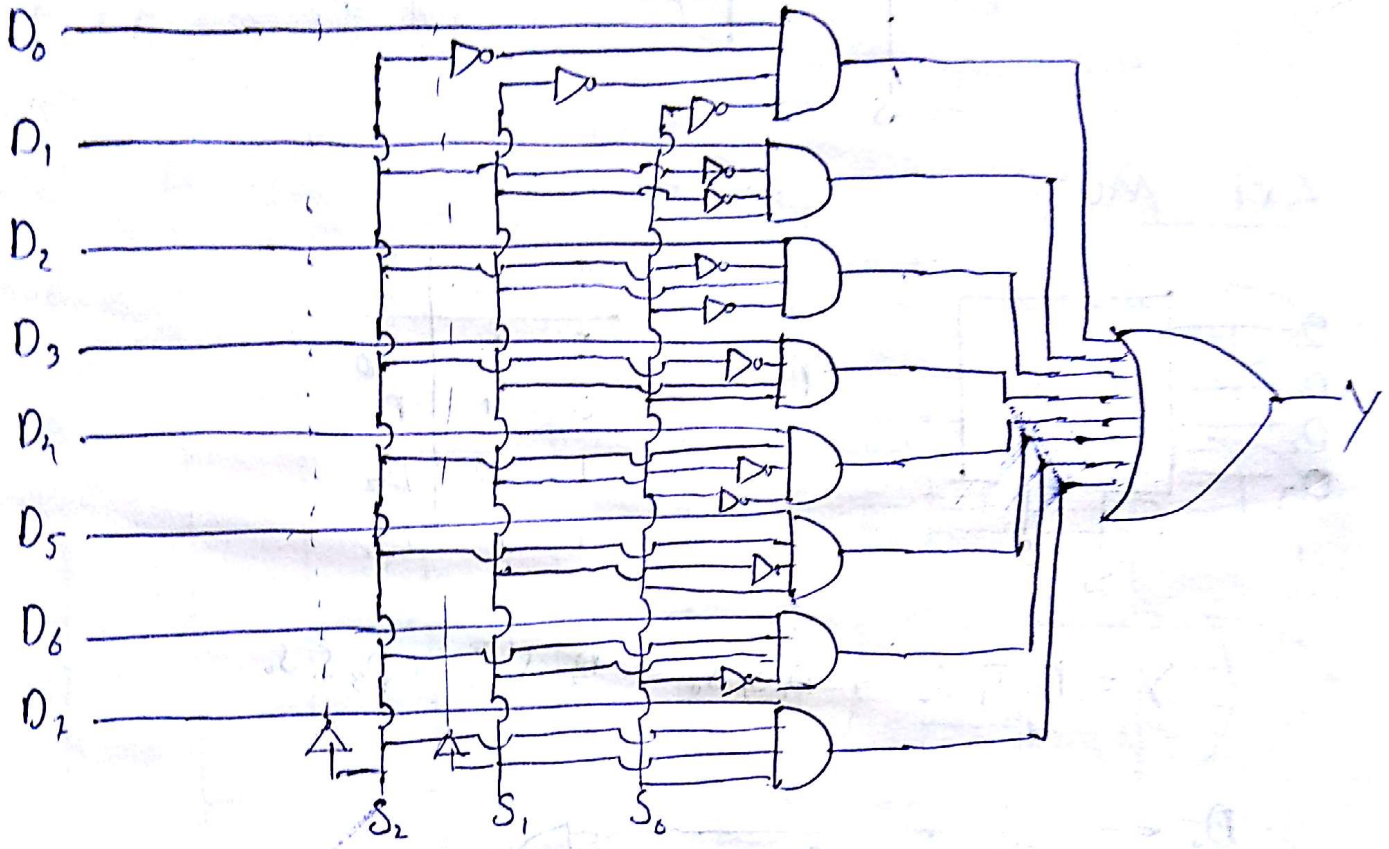


8 x 1 MUX

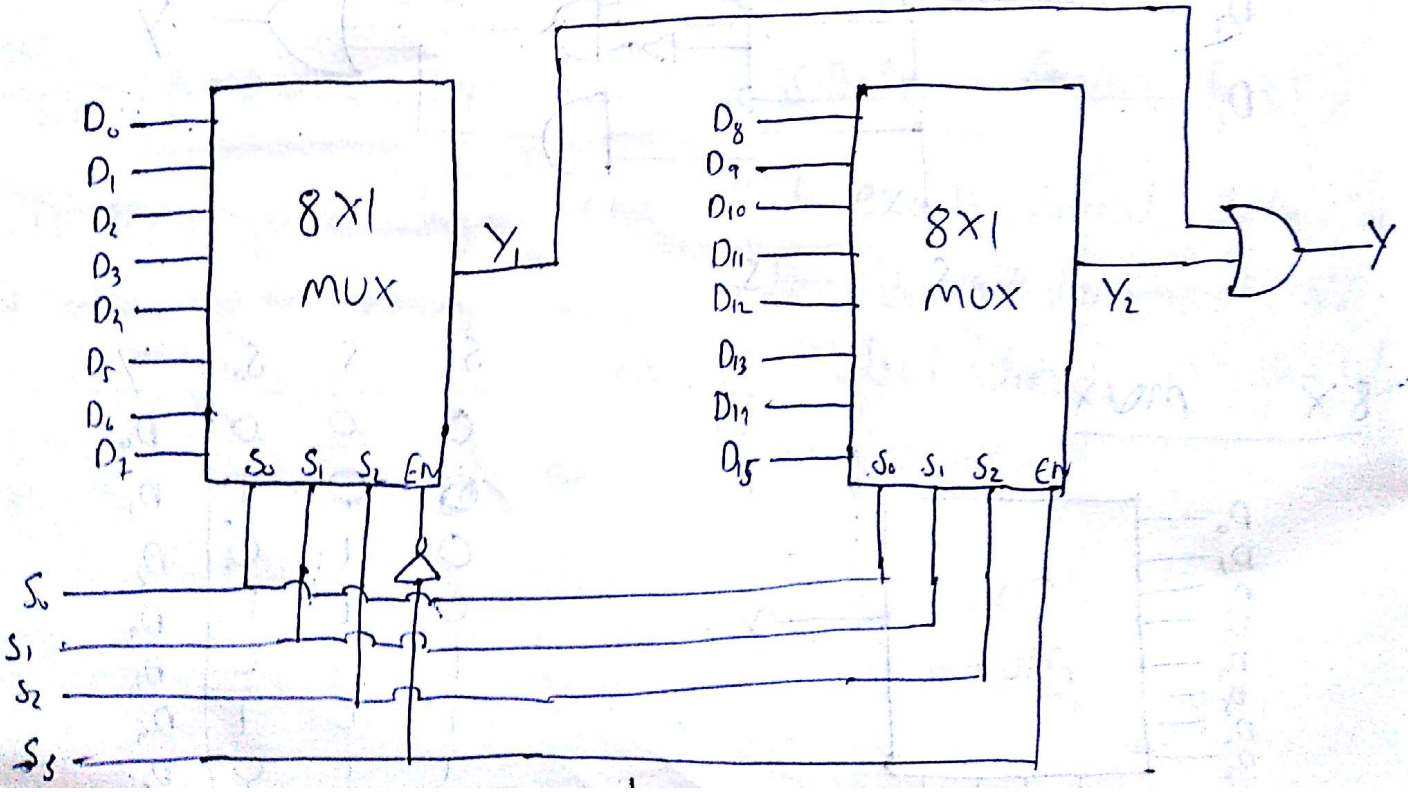


S_2	S_1	S_0	Y
0	0	0	D_0
0	0	1	D_1
0	1	0	D_2
0	1	1	D_3
1	0	0	D_4
1	0	1	D_5
1	1	0	D_6
1	1	1	D_7

$$Y = D_0 \bar{S_0} \bar{S_1} \bar{S_2} + D_1 \bar{S_2} \bar{S_1} S_0 + D_2 \bar{S_2} S_1 \bar{S_0} + D_3 \cdot \bar{S_2} S_1 S_0$$
$$+ D_4 S_2 \bar{S_1} \bar{S_0} + D_5 S_2 \bar{S_1} S_0 + D_6 \cdot S_2 S_1 \bar{S_0} + D_7 \cdot S_2 S_1 S_0$$



Q. Design and implement a 16×1 MUX with two 8×1 m



S_3	S_2	S_1	S_0	
0	0	0	0	D_0
0	0	0	1	D_1
⋮	⋮	⋮	⋮	
0	1	1	1	D_7
1	0	0	0	D_8
1	0	0	1	D_9
⋮	⋮	⋮	⋮	
1	1	1	1	D_{15}

S_3 is zero in the part which is given to 1st MUX through NOT.

To select one of the 16 inputs, 4 select lines (S_3, S_2, S_1, S_0) are required. Among the 4 select lines, the least significant 3 select lines are connected with the select i/ps of both the MUX. Most significant bit S_3 is directly connected to ENABLE of MUX_1 through an inverter and directly connected to ENABLE of MUX_2. when $S_3 = 0$, MUX one is enabled, D_0 to D_7 is selected. when $S_3 = 1$, MUX one is disabled, MUX 2 is enabled and i/ps D_8 to D_{15} are selected. The outputs of MUX1 and MUX 2 are OR gated to generate o/p Y.

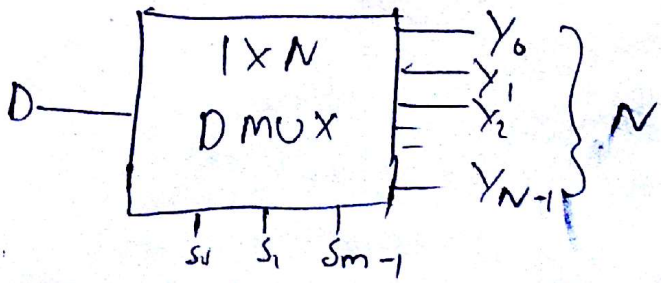
Application of MULTIPLEXERS

1. Data routing or data selecting.
2. Logic function generator.
3. Control sequencer.
4. Parallel to serial converter.

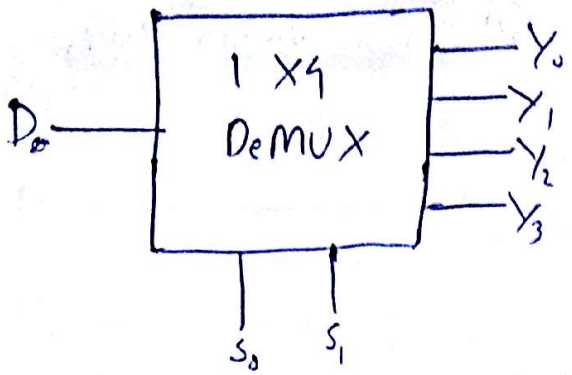
Demultiplexers

one to many.

data selectors.



1 X 4 DeMUX



S_1	S_0	Y_0	Y_1	Y_2	Y_3
0	0	D	0	0	0
0	1	0	D	0	0
1	0	0	0	D	0
1	1	0	0	0	D

$Y_0 = D \bar{S_1} \bar{S_0}$

$Y_1 = D \bar{S_1} S_0$

$Y_2 = D S_1 \bar{S_0}$

$Y_3 = D S_1 S_0$