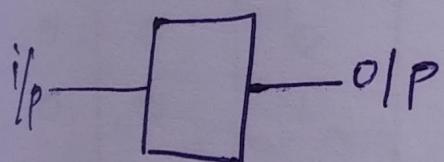


Digital circuits (combination of logic gates)

Combinational circuit

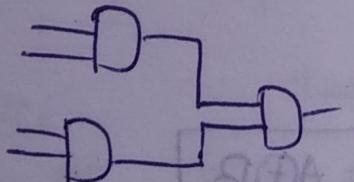


O/p depends on present I/P

No-feedback

No-memory element

ex:-



doesn't depend on past O/P

① Address

$H \cdot A / F \cdot A$

② Subtractor

$+S / F \cdot S$

③ Multiplexer

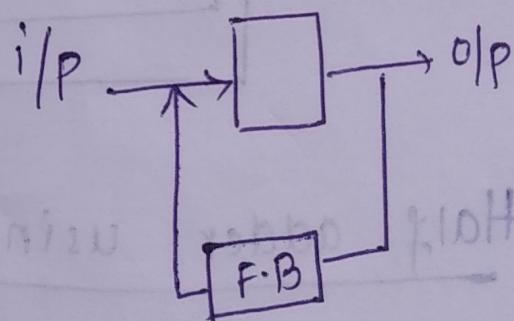
④ demultiplexer

⑤ encoder

⑥ decoder

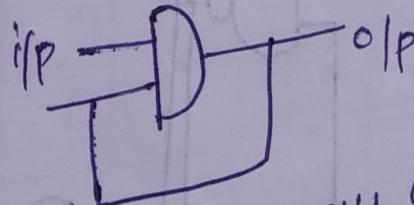
⑦ code convertor

sequential circuit



O/p depends on present I/P + past O/P

contains feed back & memory element

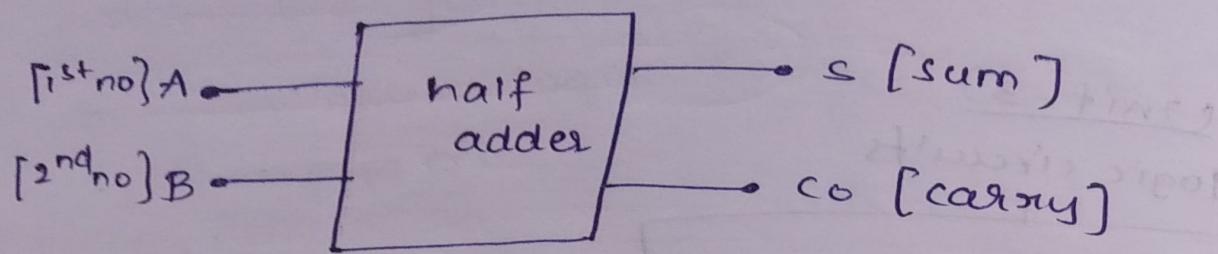


① Asynchronous counter
(Ripple counter)

② Synchronous counter
(with ripple carry & without ripple carry)

③ Sequential Counter
ring counter & Johnson counter

Half adder :- [combinational C.R.T.S]



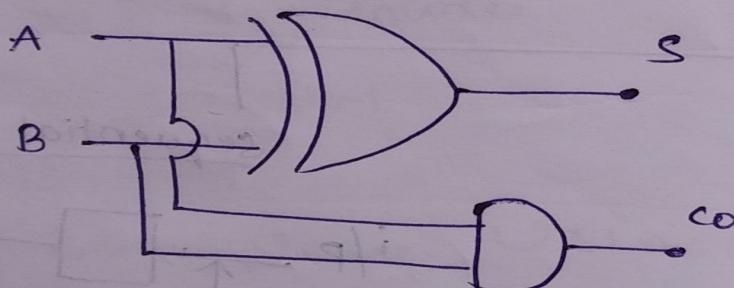
→ it is used to add the single bit nos
→ it doesn't take carry from previous sum

possible combinations are (Truth table)

A	B	S	CO
0	0	0	0
0	1	1	0
1	0	1	0

$$S = A \oplus B \text{ (XOR)}$$

$$CO = A \cdot B$$



Half adder using NAND gate only:-

$$S = A \oplus B, CO = A \cdot B$$

$$[A \cdot (\bar{A} \cdot B)]' = A' + A \cdot B$$

$$S = A \oplus B$$

$$[(A' + AB)(B' + AB)]'$$

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

$$[B(\bar{A} \cdot B)]' = B' + AB = \overline{A \oplus B} = A \oplus B$$

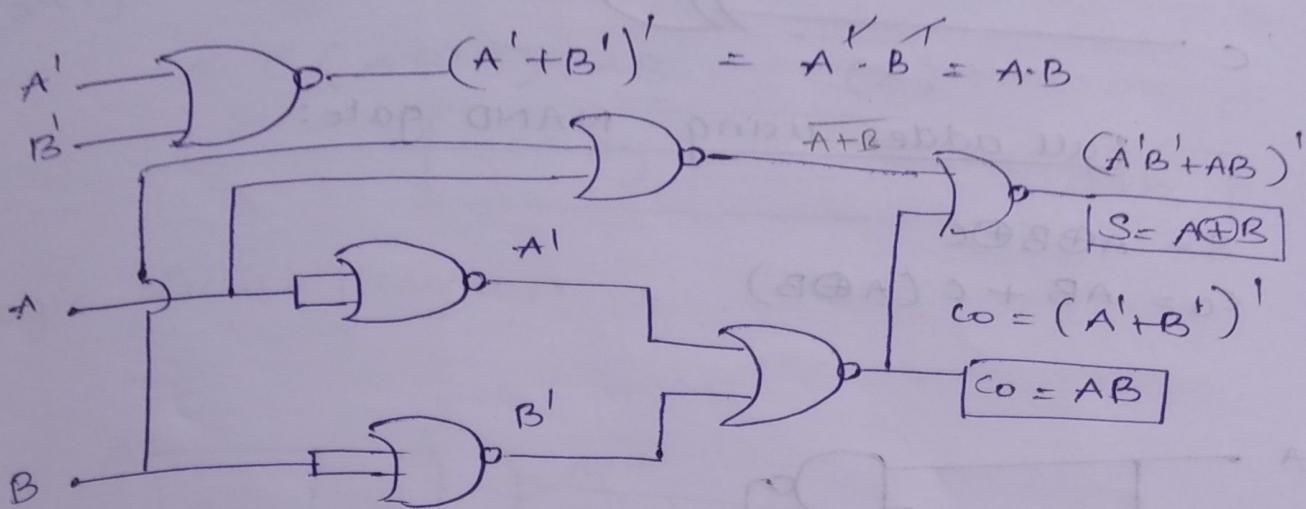
$$\times \text{NOR} = \text{XOR}$$

$$A \cdot B = CO$$

Half Adder using NOR gate only:

$$S = A \oplus B$$

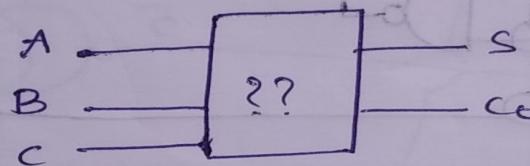
$$CO = A \cdot B$$



$$S = A \oplus B = A'B + B'A$$

$$= (\overline{A'B} + AB)$$

Full adder :-



3 ips & 20 ips

A	B	C	S	CO
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

for S -

BCi		00	01	11	10
A	B	0	1	0	1
0	0	0	1	0	1
1	0	1	0	1	0

for CO -

BCi		00	01	11	10
A	B	0	0	1	0
0	0	0	0	1	0
1	0	1	0	1	0

check board configuration

$$S = A \oplus B \oplus C$$

$$CO = BC + AB + AC$$

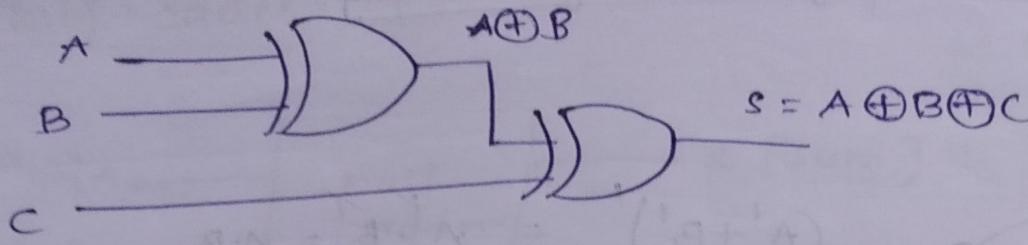
Simplify we get

$$CO = AB + C(A \oplus B)$$

$$CO = AB + \bar{A}\bar{B}C + \bar{A}B\bar{C}$$

$$CO = AB + C(\bar{A}\bar{B} + \bar{A}B) = AB + C(A \oplus B)$$

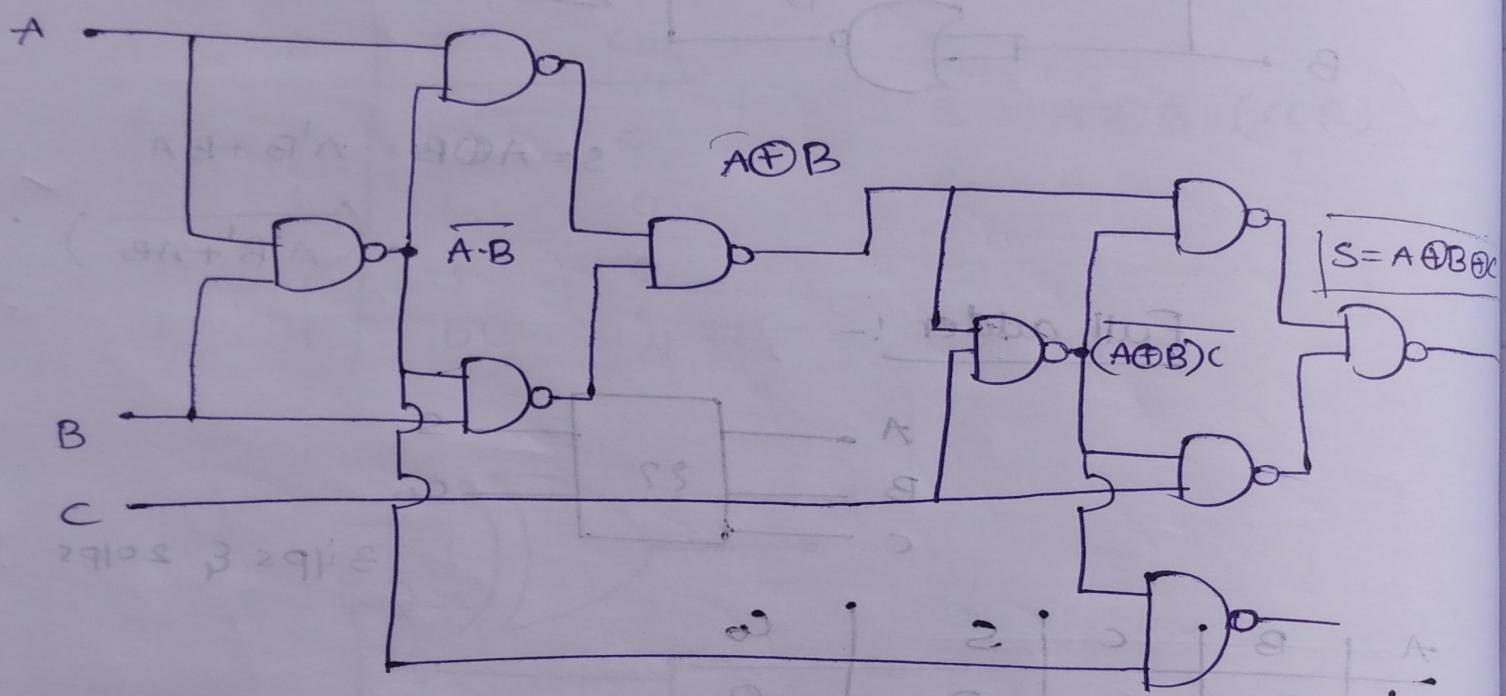
BCi		00	01	11	10
A	B	0	0	1	0
0	0	0	0	1	0
1	0	1	0	1	0



Full adder using NAND gate:

$$S = A \oplus B \oplus C$$

$$C_0 = AB + C(A \oplus B)$$



$$[\overline{AB} \cdot (\overline{(A \oplus B)} \cdot C)]$$

$$\boxed{C_0 = AB + C \cdot (A \oplus B)}$$

Full adder using NOR gate:-

$$\text{Sum} = A \oplus B \oplus C$$

$$\begin{aligned} \text{let } X &= A \oplus B = \overline{AB} + \overline{A}\overline{B} = \overline{AB} + A\overline{A} + B\overline{B} + \overline{AB} \\ &= \overline{A}(A+B) + \overline{B}(A+B) \\ &= \underline{\underline{(A+B)(\overline{A}+\overline{B})}} \end{aligned}$$

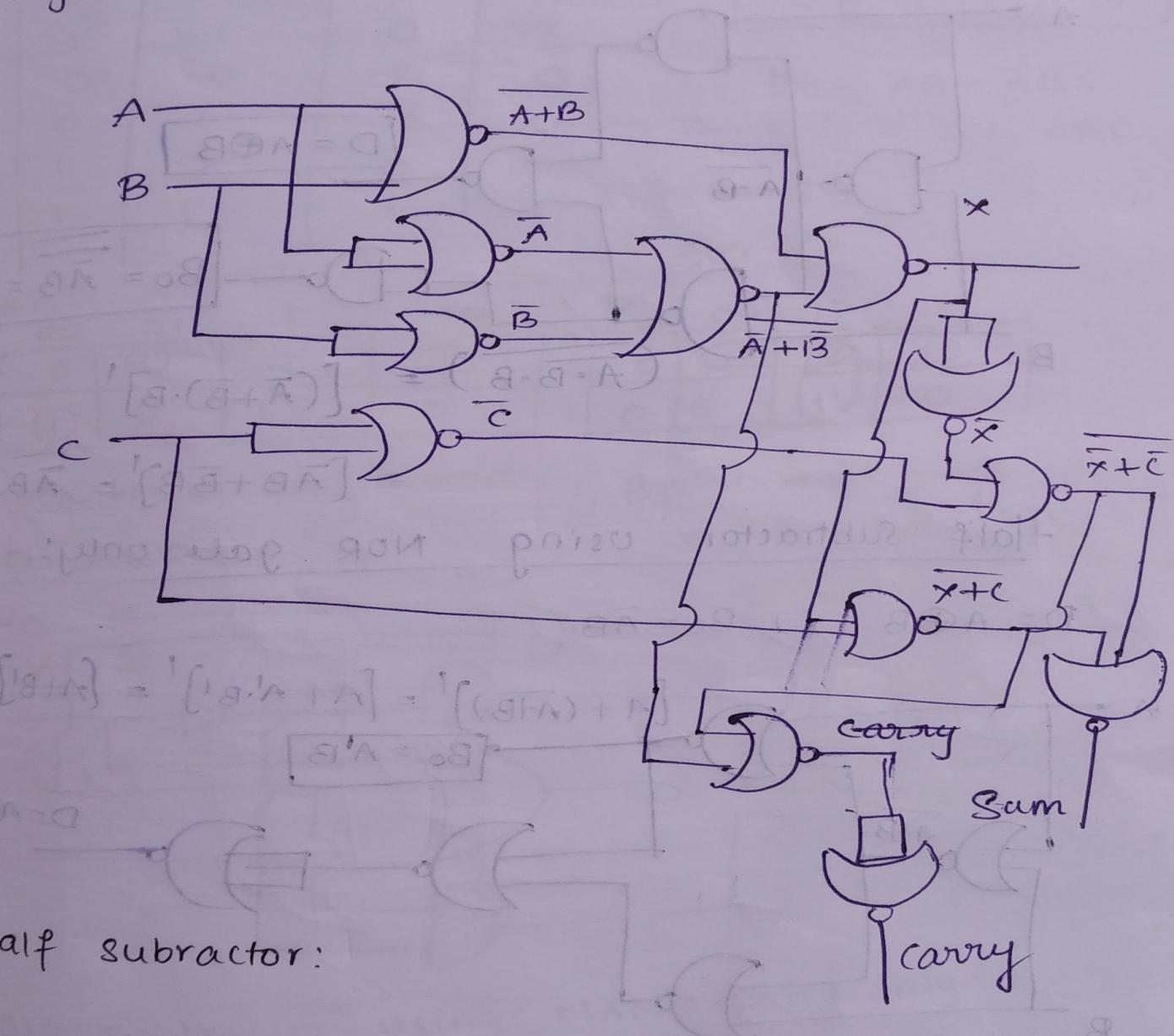
$$\begin{aligned} \text{Sum} &= X \oplus C \\ X &= \overline{\overline{A}+\overline{B}} + \overline{A+B} \end{aligned}$$

$$\text{Sum} = \overline{\overline{x} + \overline{c}} + \overline{x + c}$$

$$\begin{aligned}\text{Carry} &= C(\overline{A}B + A\overline{B}) + AB \\ &= C(A \oplus B) + AB = C(x) + AB\end{aligned}$$

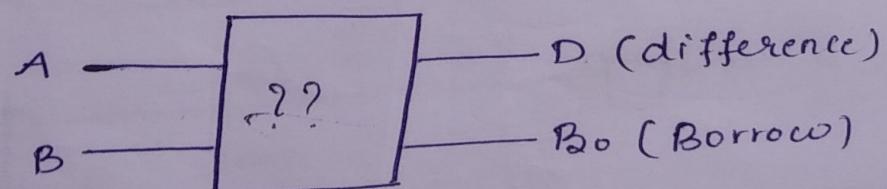
$$\text{Carry} = \overline{\overline{c} + \overline{x}} + \overline{\overline{A} + \overline{B}}$$

logic circuit diagram



Half subtractor:

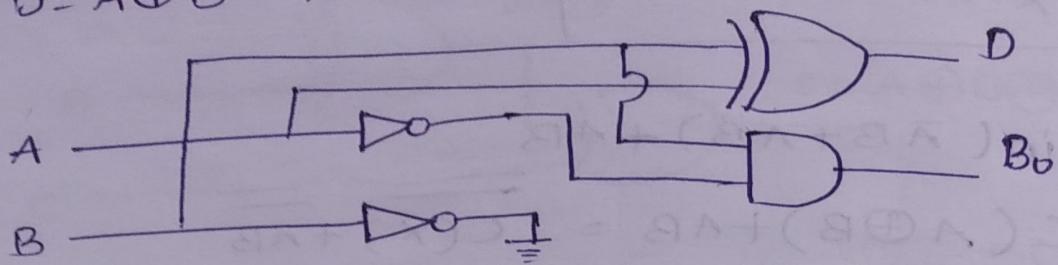
A	B	D	B_0
0	0	0	0
0	1	1	1
1	1	0	0
1	0	1	0



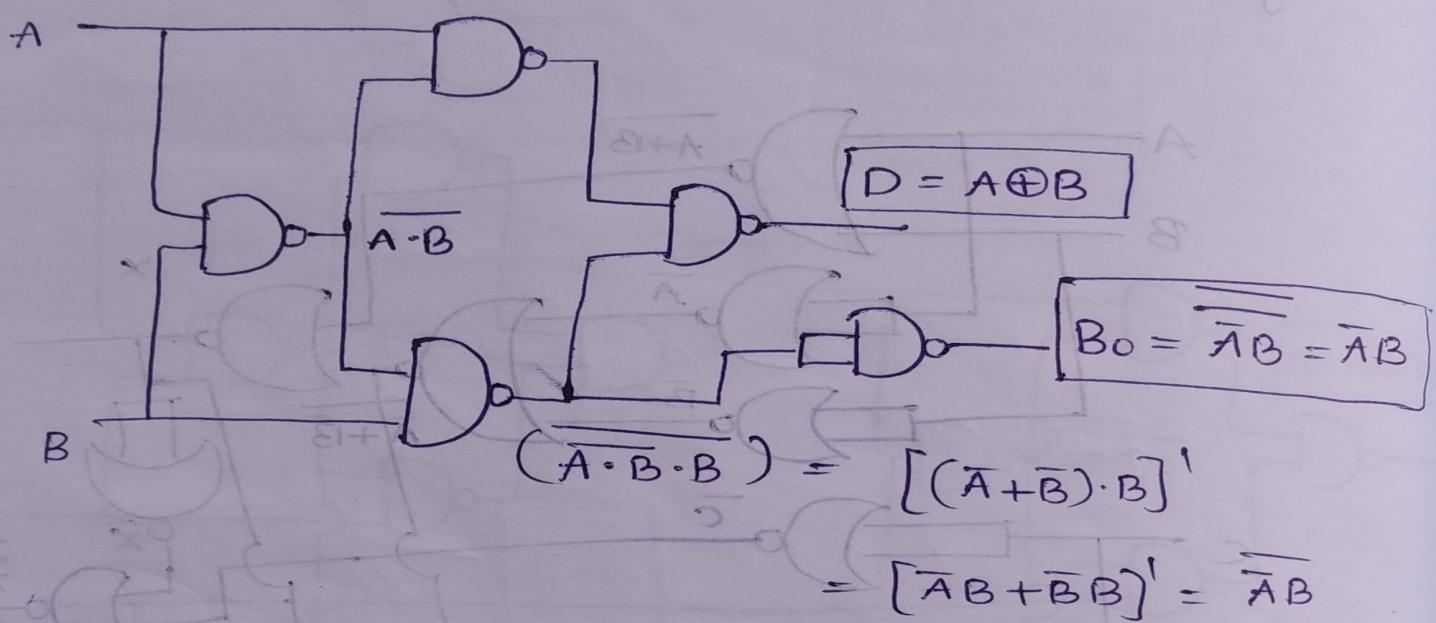
$$D = A \oplus B$$

$$B_0 = \overline{A}B$$

$$D = A \oplus B, B_0 = \overline{AB}$$

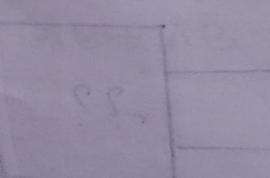
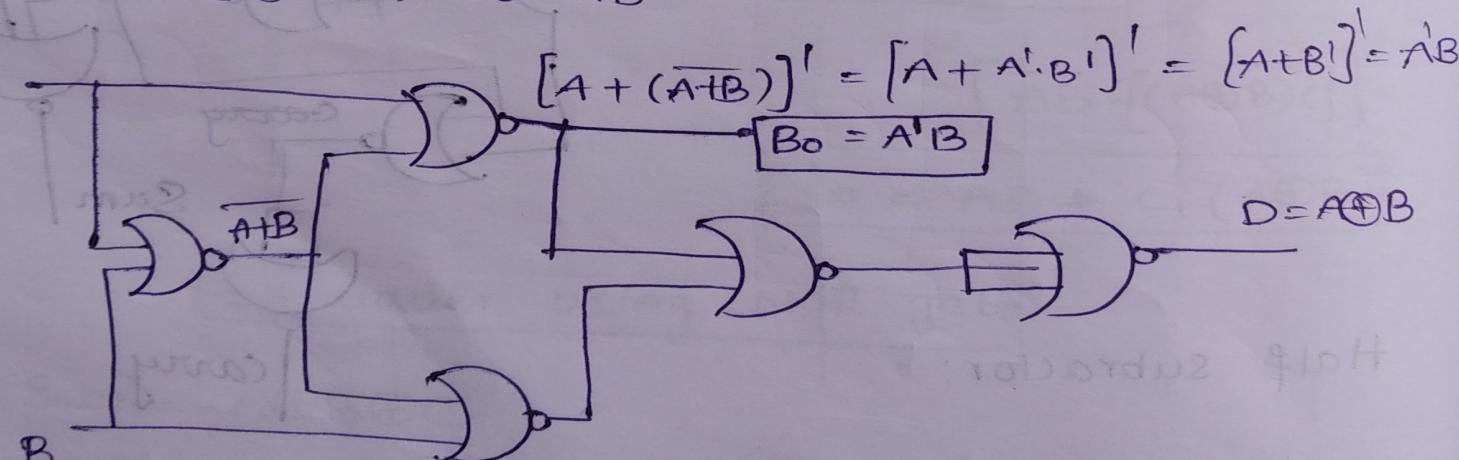


Half subtractor using NAND gate only:



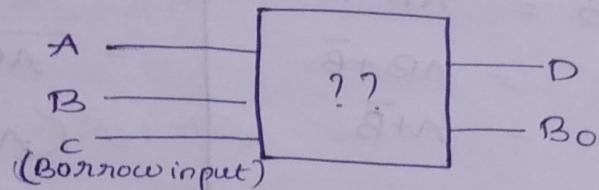
Half Subtractor using NOR gate only:

$$D = A \oplus B \rightarrow, B_0 = \overline{AB}$$



A	B	D	B ₀
0	0	0	0
0	1	1	0
1	0	1	1
1	1	0	0

full subtractor:-



A	B	C	D	B _o
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

for D :-

BC	00	01	11	10
A	0	1	0	1
B	1	0	1	0

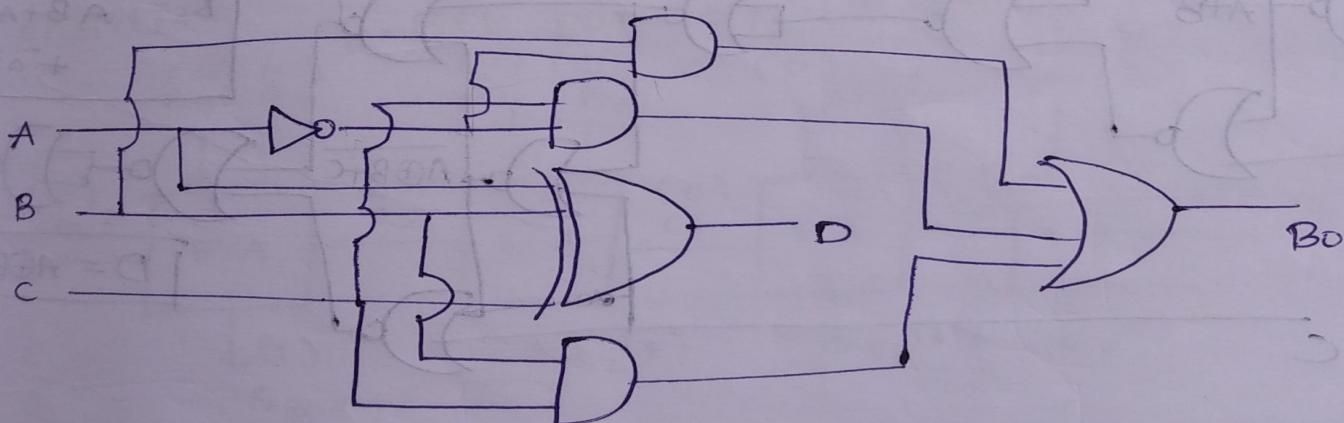
for B_o :-

BC	00	01	11	10
A	0	1	0	1
B	0	0	1	0

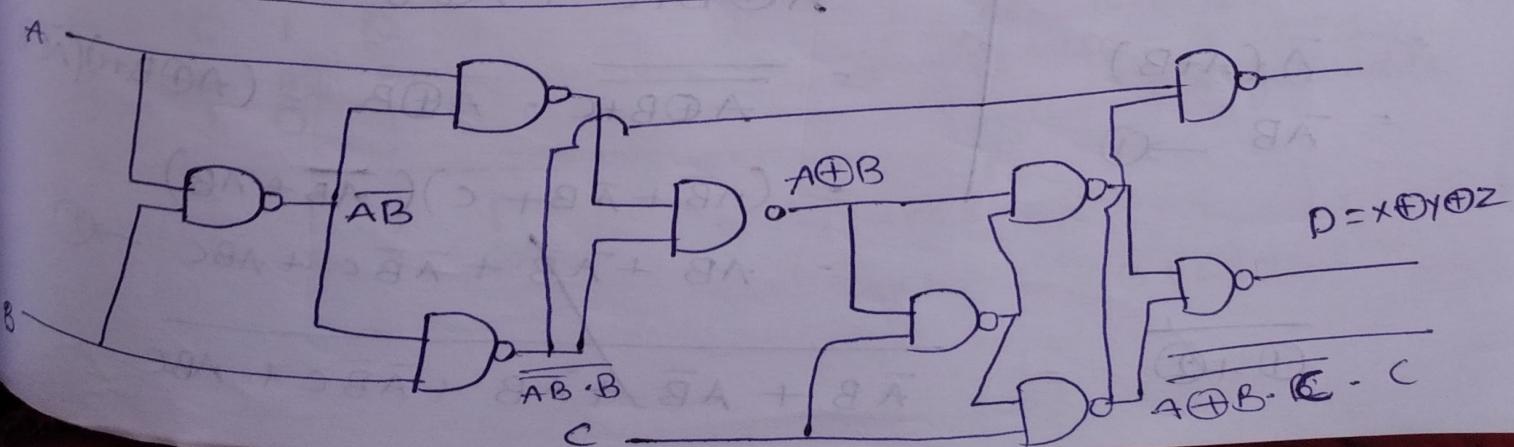
check board configuration

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

$$\begin{aligned} B_o &= BC + \bar{A}C + \bar{A}\bar{B} \\ &= \bar{A}B + \bar{A}\bar{B}C + ABC \end{aligned}$$



full subtractor using NAND gates only:-



$$\overline{\overline{AB} \cdot B} = \overline{\overline{AB} + \overline{B}}$$

$$= AB + \overline{B}$$

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

$$\overline{\overline{A \oplus B \cdot C} \cdot C}$$

$$= \overline{\overline{A \oplus B \cdot C}} + \overline{C}$$

$$= (A \oplus B) \cdot C + \overline{C}$$

$$= A \oplus B + \overline{C}$$

$$B_0 = (A + \overline{B}) \cdot (\overline{A \oplus B + \overline{C}})$$

$$= \overline{A + \overline{B}} + \overline{A \oplus B + \overline{C}}$$

$$= \overline{A} \cdot B + \overline{A \oplus B} \cdot C$$

$$= \overline{A} \cdot B + A \odot B \cdot C$$

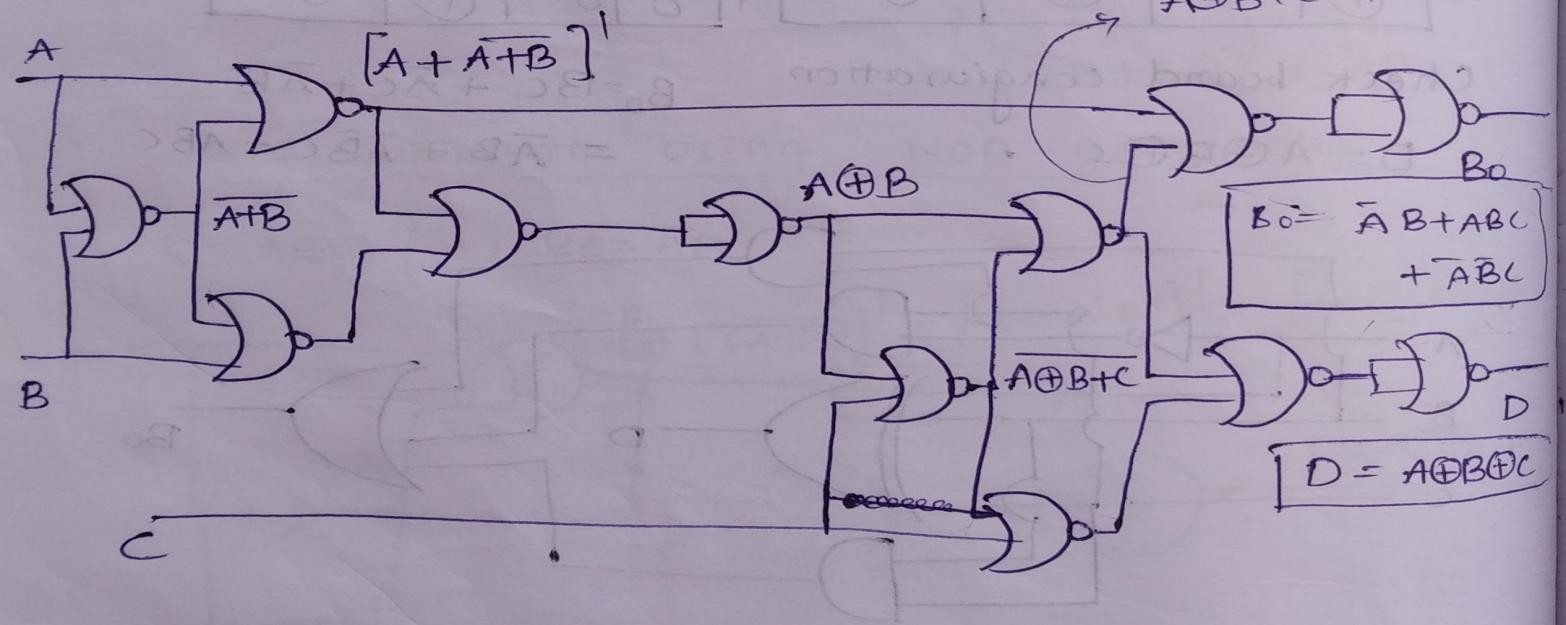
$$= \overline{A}B + (AB + \overline{A}\overline{B}) \cdot C$$

$$= \overline{A}B + ABC + \overline{ABC}$$

$\overline{\text{XOR}} = \text{XNOR}$

full subtractor using NOR gates only:

$$\overline{A \oplus B + C} + A \oplus B$$



$$A + \overline{A+B}$$

$$= \overline{A} \cdot (A + B)$$

$$= \overline{A}B \rightarrow ①$$

$$\overline{A \oplus B + C} + A \oplus B$$

$$= \overline{\overline{A \oplus B + C} \cdot \overline{A \oplus B}} = (A \oplus B + C)(\overline{A \oplus B})$$

$$= (\overline{A}B + \overline{A}\overline{B} + C)(\overline{A}B + AB)$$

$$= \overline{A}\overline{B} + \overline{A}B + \overline{A}\overline{B}C + ABC \rightarrow ②$$

$$\overline{① + ②}$$

$$= \overline{A}B + A\overline{B} + \overline{A}\overline{B} + \overline{A}\overline{B}C + ABC$$

$$= \overline{AB} + A\overline{B} + \overline{A}\overline{B}C + ABC$$

$$= (\overline{AB} + \overline{A}\overline{B}C) (\overline{A}\overline{B} + AB)$$

$$= 0 + \overline{A}\overline{B}C + ABC$$

$$\overline{\textcircled{1} + \textcircled{2}} = B_0 = \overline{AB} + ABC + \overline{A}\overline{B}C$$

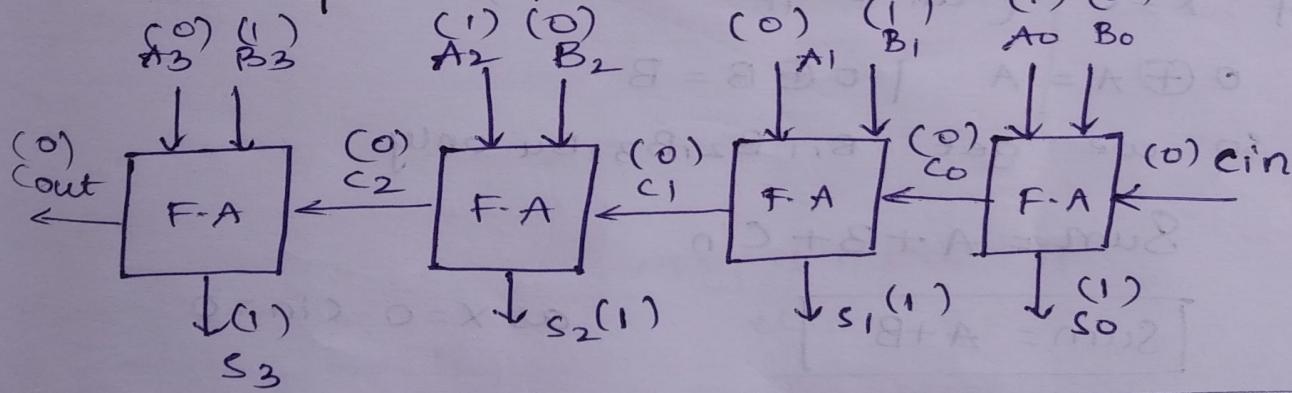
N-Bit parallel adder

every time we are not going to integrate these full adders for single single bit so if there is any device is available for adding multiple bits i.e N-Bit parallel adder
 \rightarrow cascade of full adders (so we can use it readily)

Note:-

how many bits are there we need to cascade that many full adders

Consider 4-Bit parallel full adder



each contains (3 i/p, 2 o/p)
 cin = carry input
 A0, B0 - inputs
 S = sum, co = carry o/p
 FA = full adder

$$A = \begin{matrix} A_3 & A_2 & A_1 & A_0 \\ 0 & 1 & 0 & 1 \end{matrix}$$

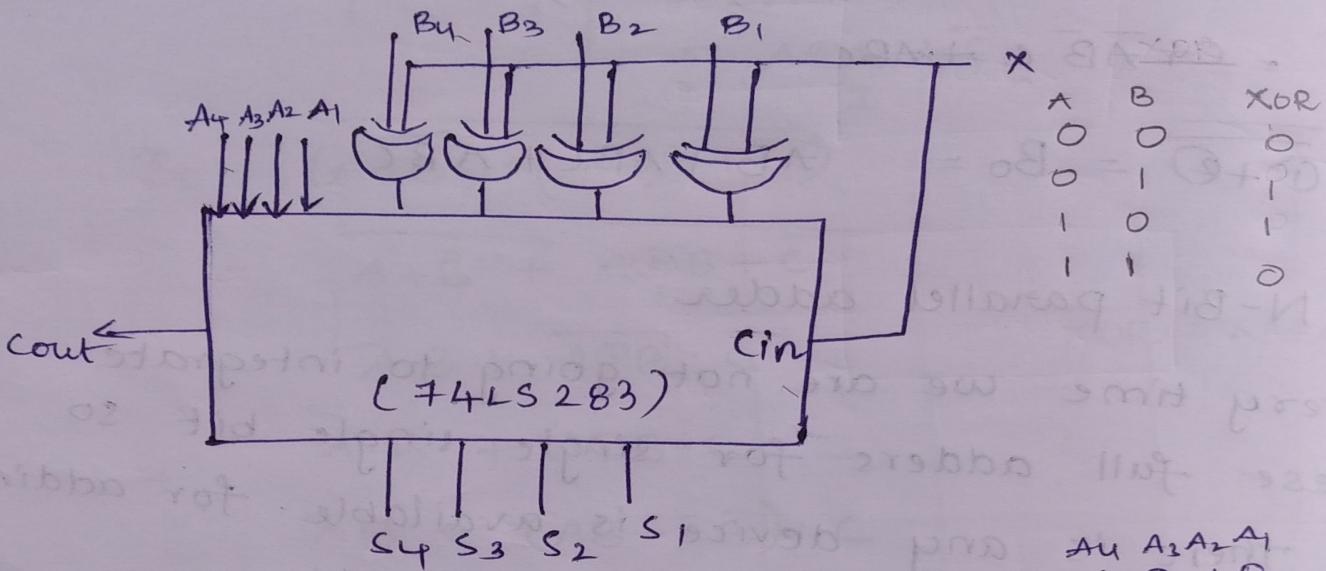
$$B = \begin{matrix} B_3 & B_2 & B_1 & B_0 \\ 1 & 0 & 1 & 0 \end{matrix}$$

$$S_3 \quad S_2 \quad S_1 \quad S_0 \quad (S_0) \text{ Sum} = A \oplus B \oplus C$$

$$(C_0) \text{ Carry} = AB + BC + CA$$

4-Bit parallel adder / subtractor

→ This is adder cum subtractor circuit so that we can add (or) subtract 4 bit nos.



Observations:-

→ circuit with IC 74LS283

→ 2 i/p XOR gates are taken

1 i/p — coming from bits of the second no.

2 i/p — Given to common i/p Cin

if $x=0$ (all 2 i/p's are 0)

$$0 \oplus A = A \quad | \quad 0 \oplus B = B$$

so we get B_1, B_2, B_3, B_4 only

$$\text{Sum} = A + B + \text{Cin}$$

$$\boxed{\text{Sum} = A + B}$$

(1) as $x=0$ $\text{Cin}=0$

if $x=1$

$$1 \oplus A = \bar{A} \quad | \quad 1 \oplus B = \bar{B}$$

so we get $\bar{B}_1, \bar{B}_2, \bar{B}_3, \bar{B}_4$ only

$$\text{Sum} = A + (\text{i}'s \text{compliment of } B) + \text{Cin}$$

$$= A + (\text{i}'s \text{compliment of } B) + 1$$

$$= A + 2's \text{compliment of } B$$

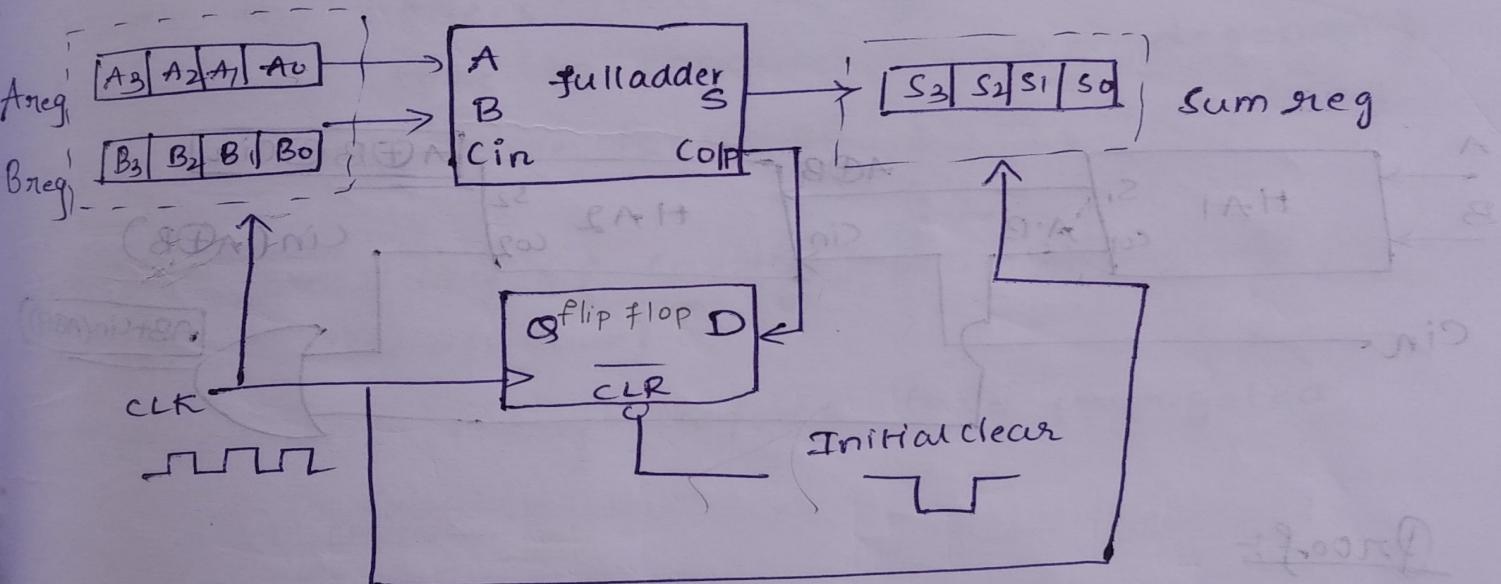
$$\boxed{\text{Sum} = A - B}$$

so if $X=0$ circuit will act as adder &
if $X=1$ circuit will act as subtractor

Serial adder:-

- parallel adders add two binary numbers in relatively faster rate than serial adders
 - But the disadvantage of parallel adder is it requires large amount of circuitry
- As serial adder performs addition bit by bit it requires simpler circuitry but low speed.

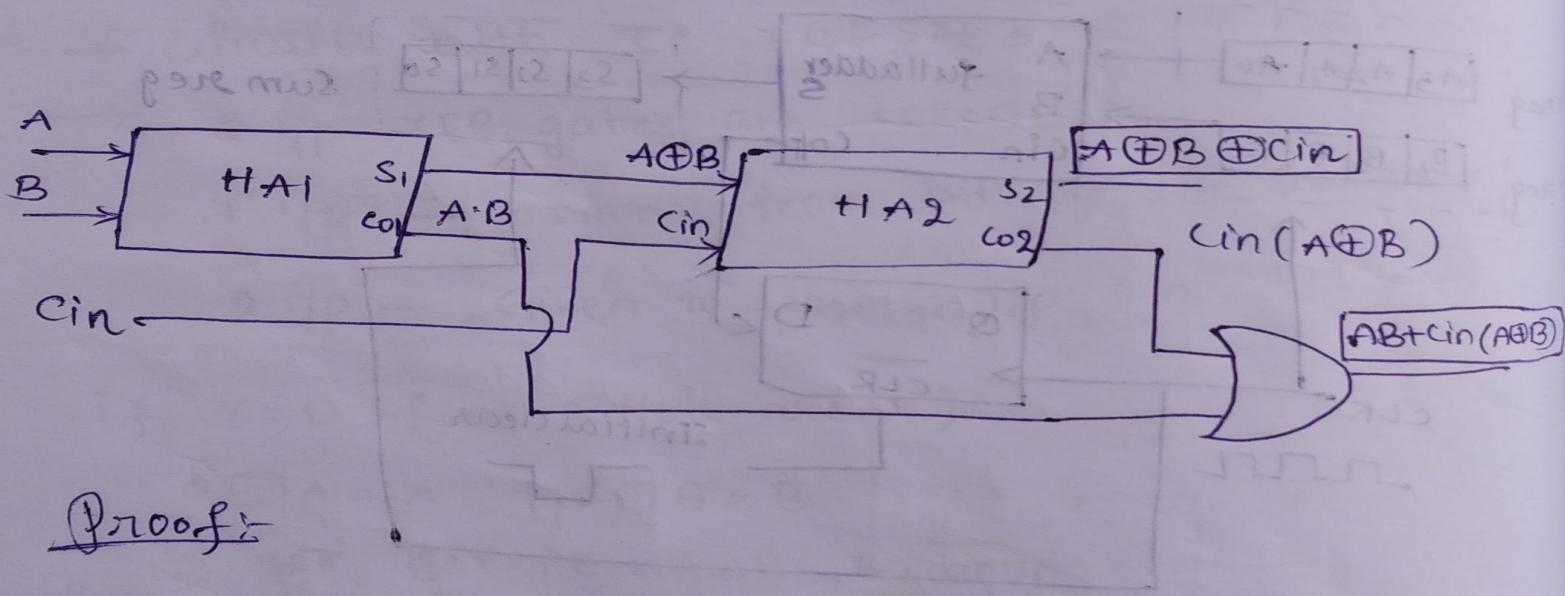
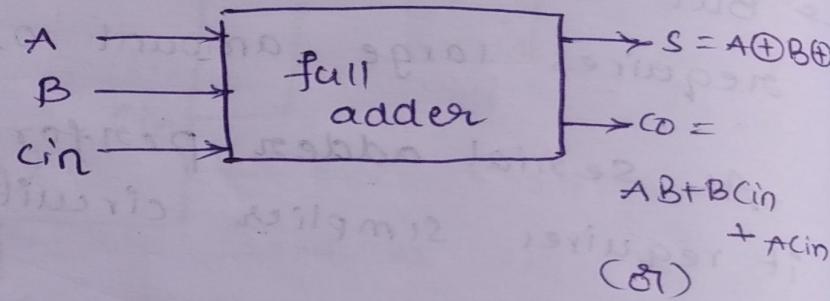
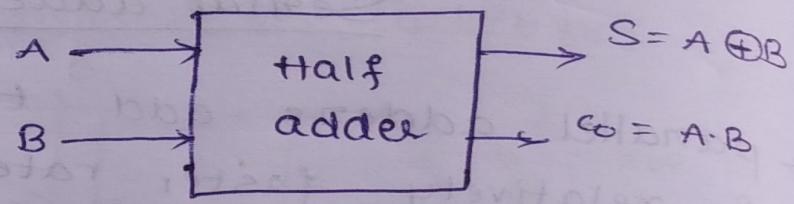
A, B, sum are shift registers



Working:-

- 1) $A_0 + B_0 = S_0$ | if carry is there $\text{ColP} = 1$
else $\text{ColP} = 0$
- 2) enters D flip flop so that clock moves the bits in right direction ($A_3 \ A_2 \ A_1 \ A_0$) in both A & B registers, as well as sum register ($S_3 \ S_2 \ S_1 \ S_0$)
- 3) $A_1 + B_1 = S_1$ and process repeats until all bits are added

Full adder using Half adders



Proof:

$$AB + \text{cin}(A \oplus B) = AB + \text{cin}(\bar{A}B + \bar{B}A)$$

$$= AB + A\bar{B}\text{cin} + \bar{A}B\text{cin}$$

$$= A(B + \bar{B}\text{cin}) + \bar{A}B\text{cin}$$

$$= A(B + \text{cin}) + \bar{A}B\text{cin}$$

$$= A \cdot B + A\text{cin} + \bar{A}B\text{cin}$$

$$= AB + \text{cin}(A + \bar{A}B)$$

$$\Rightarrow AB + \text{cin}A + \text{cin}B$$

full adder truth table

A	B	cin	sum	carry	Condition
0	0	0	0	0	No carry due to cin
0	0	1	1	0	
0	1	0	1	0	
0	1	1	0	1	Carry propagator
1	0	0	0	0	NO carry
1	0	1	0	1	Carry propagator
1	1	0	0	1	Carry Generator
1	1	1	1	1	Carry generated using A+B

Carry = 1 cases

- either A or B is 1 and $\text{cin} = 1$ (carry propagator case)
- Both A and B = 1 (carry generator case)

Carry propagator (P_i) & carry generator (G_i)

$$\text{So } P_i = A_i \oplus B_i \rightarrow ①$$

$$G_i = A_i \cdot B_i \rightarrow ②$$

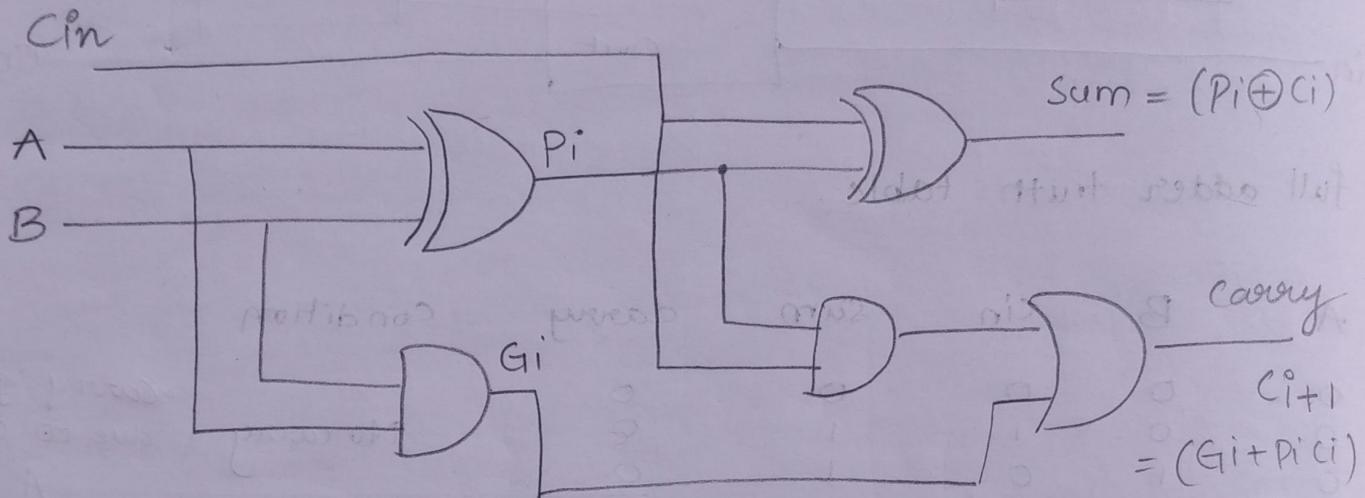
for full adder:-

$$\text{sum} = A \oplus B \oplus C_i$$

$$\text{Carry} = AB + BC_i + AC_i \xrightarrow{\text{Solving by}} C_i(A \oplus B) + AB$$

by using ① & ② re-write as

$\text{Sum} = P_i \oplus C_i$
 $\text{Carry} = G_i + P_i C_i$
 (C_{i+1})
 logic circuit



deriving C_1, C_2, C_3 from above eqn's

$$i=0$$

$$C_{0+1} = C_1 = (C_{in} P_0) + G_0$$

$$i=1$$

$$C_{1+1} = C_2 = (C_1 \cdot P_1) + G_1$$

$$= (((C_{in} P_0) + G_0) \cdot P_1) + G_1$$

$$= (C_{in} P_0 P_1) + (G_0 P_1) + G_1$$

$$i=2$$

$$C_{2+1} = C_3 = (C_2 \cdot P_2) + G_2$$

$$= (((C_1 \cdot P_1) + G_1) \cdot P_2) + G_2$$

$$= (((((C_1 \cdot P_1) + G_1) \cdot P_1) + G_1) \cdot P_2) + G_2$$

$$= G_2 + (P_2 \cdot G_1) + (P_2 P_1 G_0) + (P_2 P_1 P_0 \cdot C_{in})$$

Similarly

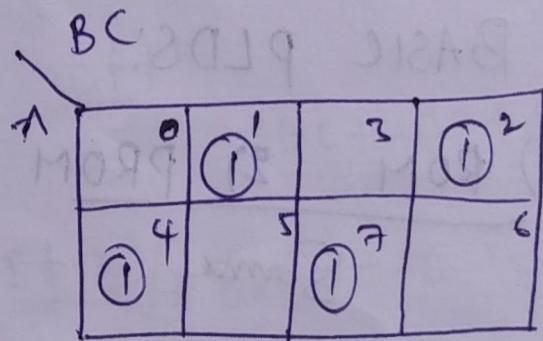
$$C_4 = G_3 + (P_3 \cdot G_2) + (P_3 P_2 G_1) + (P_3 P_2 P_1 G_0)$$

$$+ (P_3 P_2 P_1 P_0 \cdot C_{in})$$

PARITY BIT Generator:

even parity generator:

A	B	C	even parity bit
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

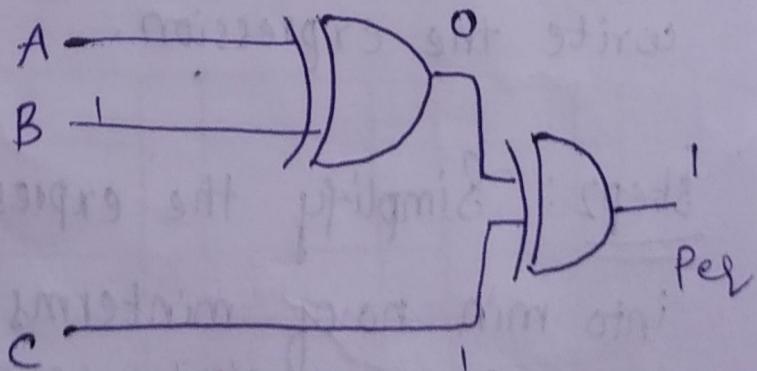


check board

$$P_{eq} = A \oplus B \oplus C$$

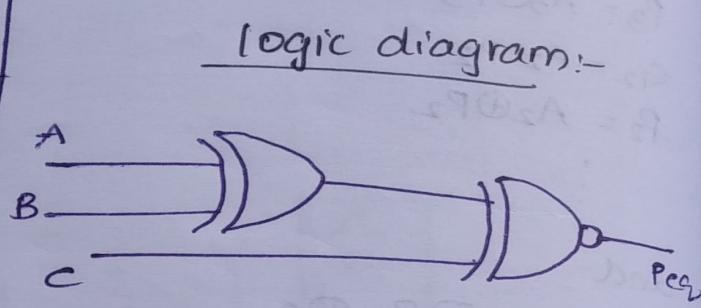
logic diagram:-

$$\begin{matrix} & 0 & 1 & 0 \\ A & & & \\ B & & & \\ C & & & \end{matrix} \quad P_{eq} = 1$$



odd parity generator:-

A	B	C	odd parity bit
0	0	0	1
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



$$P_{eq} = \overline{A \oplus B \oplus C}$$

PROGRAMMABLE LOGIC DEVICES:

- contains an array of AND gates & OR gates

Advantages:-

→ Low cost → Design a large circuit

→ Reprogramming (Modify the design)

BASIC PLDS:

- 1) ROM 2) PROM 3) PAL 4) PLA
Same

PROM / ROM :- Fixed AND array
programmable OR array.

Step 1:- If question is in the form of table or $\sum m$ write the expression.

Step 2:- Simplify the expression into min no. of minterms or cancel similar terms

Step 3:- draw Input buffers

no. of Input buffers = no. of input variables

Note:- Question can either be in $\sum m$ or in table format or in terms of expression

Step 4:- And gates = diff in 3 cases

Step 5:- Or gates = no. of functions

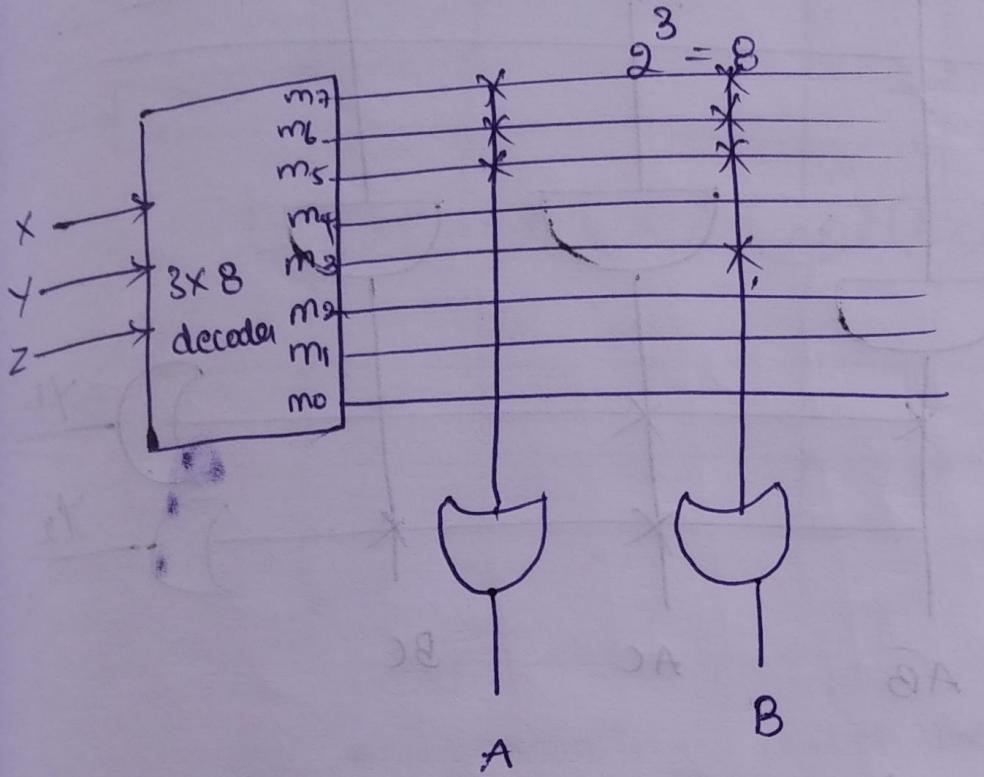
PROM / ROM :-

$$A(x, y, z) = \sum m(5, 6, 7)$$

$$B(x, y, z) = \sum m(3, 5, 6, 7)$$

min no. of minterms = 4

no. of functions = 2 ($A \oplus B$) = no. of OR gates



OR - variable

AND - fixed

PLA:-

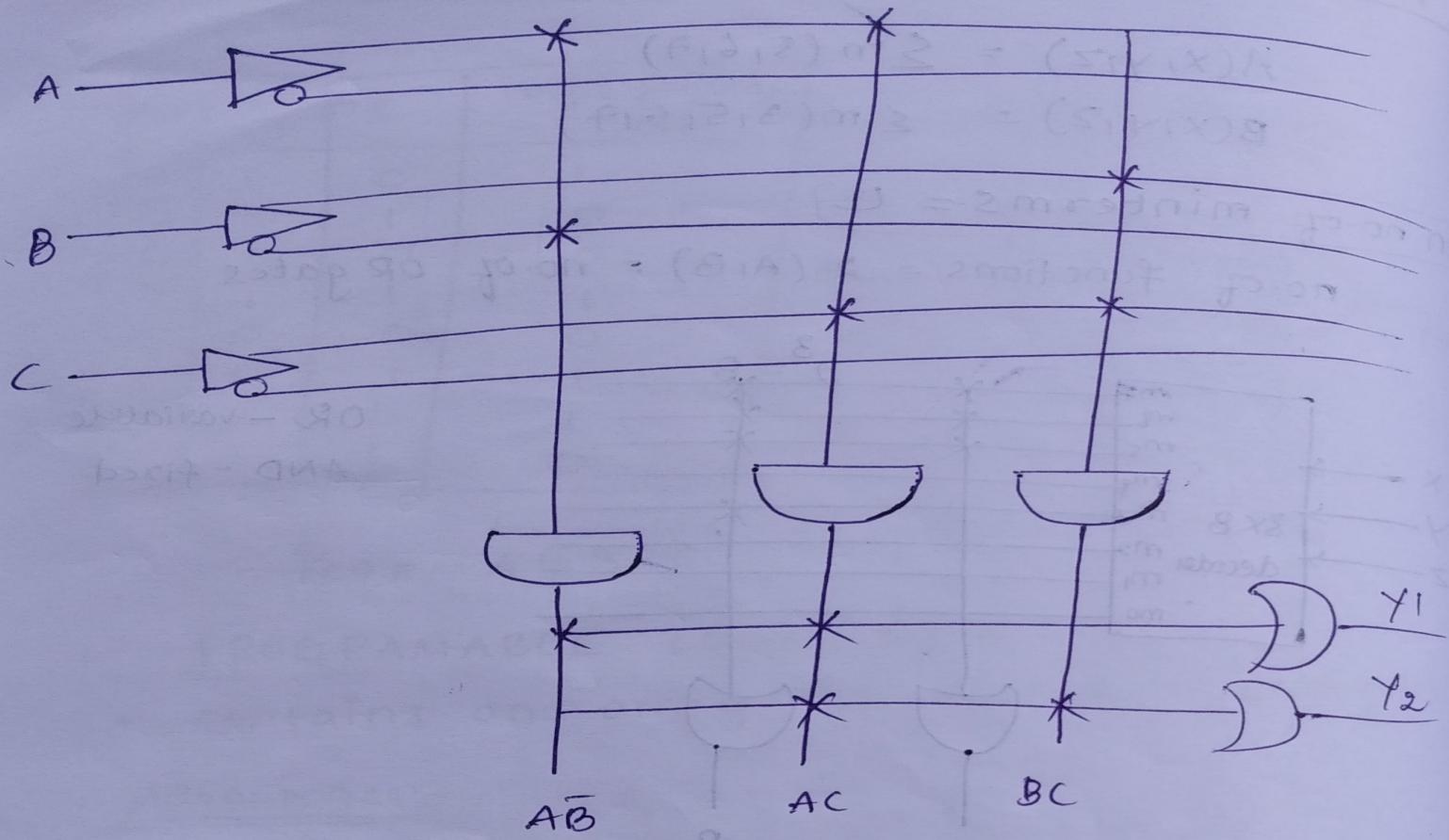
OR-variable and AND-variable

A	B	C	Y_1	Y_2	$\sum_{i=1}^{n+1} Y_i$
0	0	0	0	0	$\sum_{i=1}^{n+1} Y_i = \bar{A}\bar{B}\bar{C} + \bar{A}\bar{B}C + ABC$
0	0	1	0	0	$\sum_{i=1}^{n+2} Y_i = \bar{A}\bar{B} + AC$
0	1	0	0	0	
0	1	1	0	1	
1	0	0	1	0	
1	0	1	1	1	
1	1	0	0	0	
1	1	1	1	1	

here NAND gates = min no. of minterms

(Removing common terms)

$\bar{A}B_1\bar{A}C_1BC$



PAL:-

~~AND fixed~~

AND variable OR fixed

$$X(A_1B_1C) = \sum m(2, 3, 5, 7) = \bar{A}B + AC \quad \{ 1$$

$$Y(A_1B_1C) = \sum m(0, 1, 1, 5) = \bar{A}\bar{B} + \bar{B}C \quad \{ 2$$

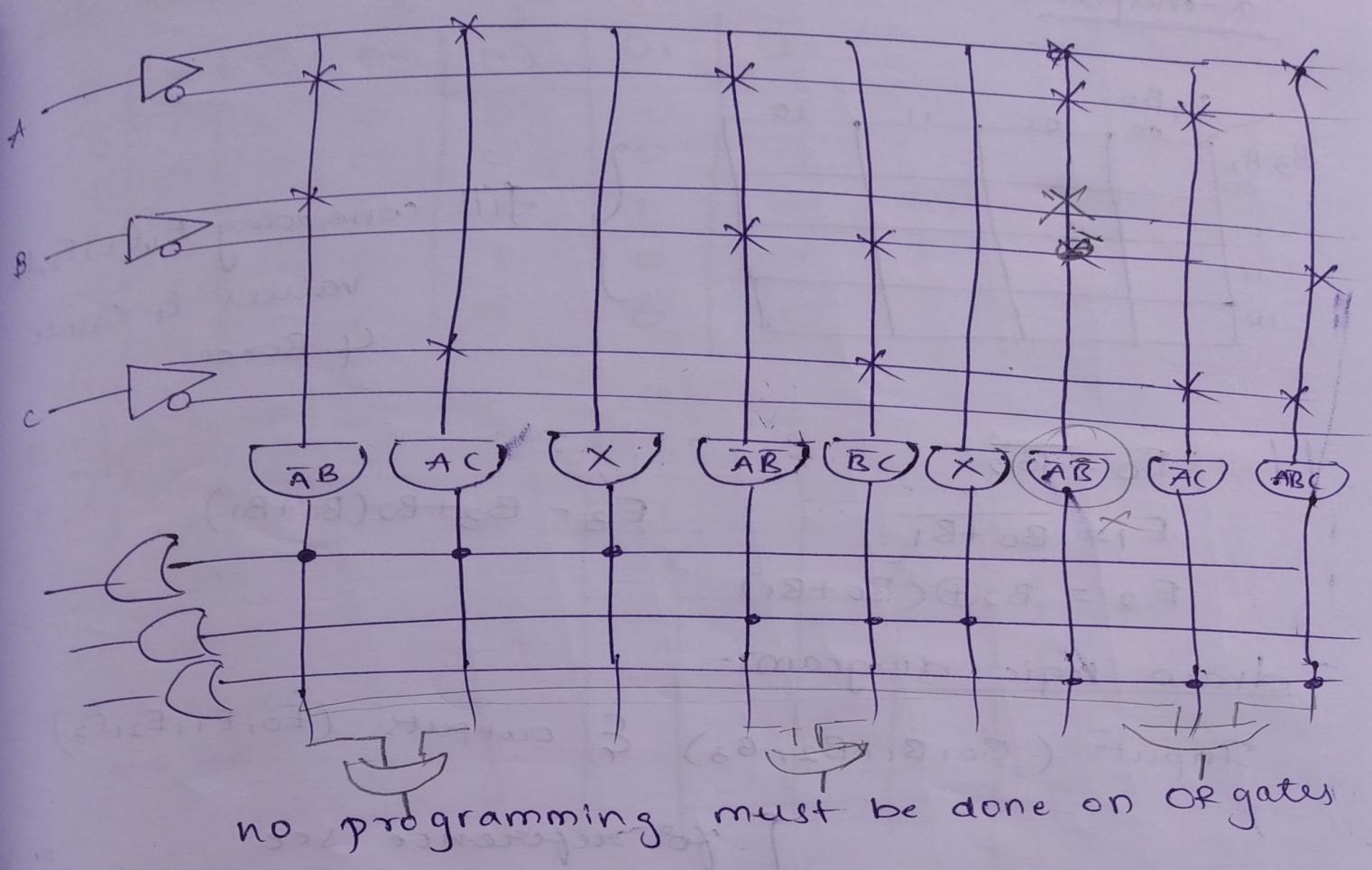
$$Z(A_1B_1C) = \sum m(0, 2, 3, 5) = \bar{A}B + \bar{A}C + \bar{A}\bar{B}C \quad \{ 3$$

here

no. of AND gates = no. of ip variables ×
max no. of min term

$$= 3 \times 3 = 9$$

for each Variable 3 nand gates



BCD to XS-3 convertor:

[see pdf]

Functional table:-

BCD Input				XS-3 output			
B ₃	B ₂	B ₁	B ₀	F ₃	F ₂	F ₁	F ₀
0	0	0	0	0	0	0	0
0	0	0	1	0	1	0	0
0	0	1	0	1	0	0	0
0	0	1	1	1	1	0	0
0	1	0	0	x	x	x	x
0	1	0	1	x	x	x	x
0	1	1	0	x	x	x	x
1	0	0	0	x	x	x	x
1	0	0	1	x	x	x	x
1	0	1	0	x	x	x	x
1	0	1	1	x	x	x	x
1	1	0	0	x	x	x	x
1	1	0	1	x	x	x	x
1	1	1	0	x	x	x	x
1	1	1	1	x	x	x	x

K-maps:-

	B ₁ B ₀ 00	01	11	10
B ₃ B ₂ 00				
01				
11				
10				

fill corresponding E₀, E₁, E₂, E₃ values & create 4 Boxes

$$E_0 = \overline{B_0}$$

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

$$E_2 = B_2 \oplus (B_0 + B_1)$$

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

draw logic diagram! -

Inputs (B₀, B₁, B₂, B₃) & outputs (E₀, E₁, E₂, E₃)

[for reference see

UNIT-3 PDF(PPT)-2

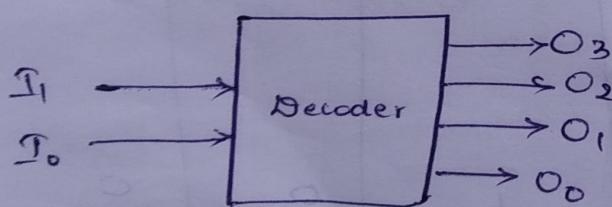
by sagar

Decoders:-

n - inputs

2ⁿ - outputs

2x4 decoder:-



E=0 | decoder enables
E=1 | decoder disables

$$A = I_1 \quad B = I_0 \quad I_0$$

$$O_3 = AB$$

$$O_2 = A\bar{B}$$

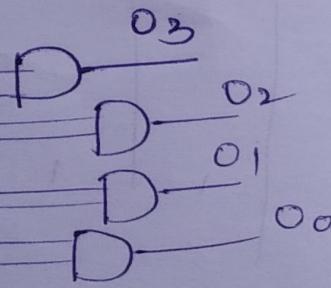
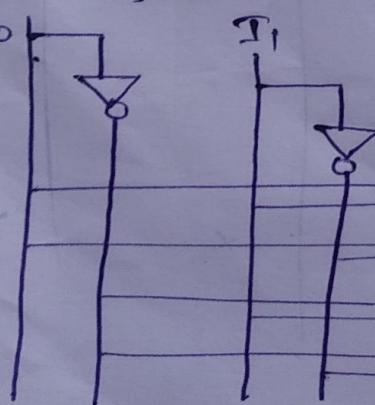
$$O_1 = \bar{A}B$$

$$O_0 = \bar{A}\bar{B}$$

$$I_1 = A \oplus B$$

$$I_0 = B$$

	I ₁	I ₀	O ₃	O ₂	O ₁	O ₀
0 =	0	0	0	0	0	1
1 =	0	1	0	0	1	0
2 =	1	0	0	1	0	0
3 =	1	1	1	0	0	0



2x4 Decoder with enable:

E	A	B	O_3	O_2	O_1	O_0
0	x	x	0	0	0	0
1	0	0	0	0	0	1
1	0	1	0	0	1	0
1	1	0	0	1	0	0
1	1	1	1	0	0	0

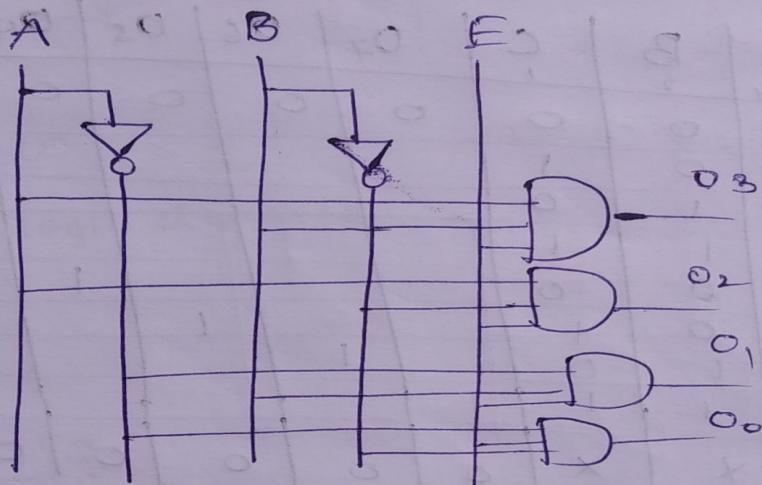
$E=0$ enables
 $E=1$ disables

$$O_3 = AB\bar{E}$$

$$O_2 = A\bar{B}E$$

$$O_1 = \bar{A}BE$$

$$O_0 = \bar{A}\bar{B}E$$



2x4 decoder with active low input:

A	B	O_3	O_2	O_1	O_0
0	0	1	1	1	0
0	1	1	1	0	1
1	0	1	0	1	1
1	1	0	1	1	1

$$\overline{O_3} = \bar{A}\bar{B}$$

$$\overline{O_2} = A\bar{B}$$

$$\overline{O_1} = \bar{A}B$$

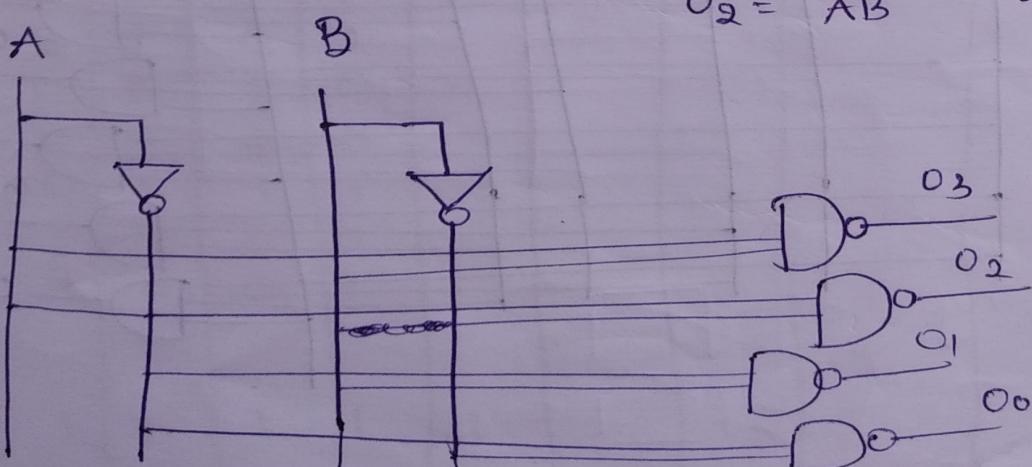
$$\overline{O_0} = \bar{A}\bar{B}$$

$$O_3 = \overline{AB}$$

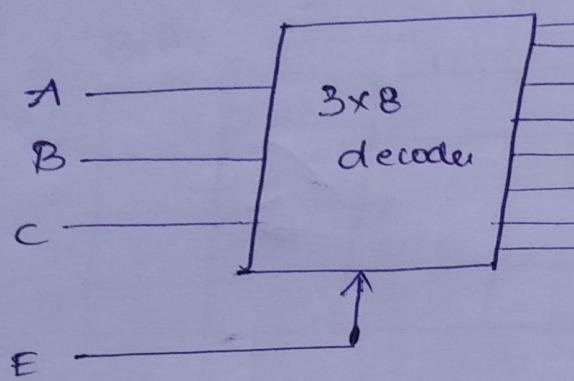
$$O_2 = \overline{A\bar{B}}$$

$$O_1 = \overline{\bar{A}B}$$

$$O_0 = \overline{\bar{A}\bar{B}}$$



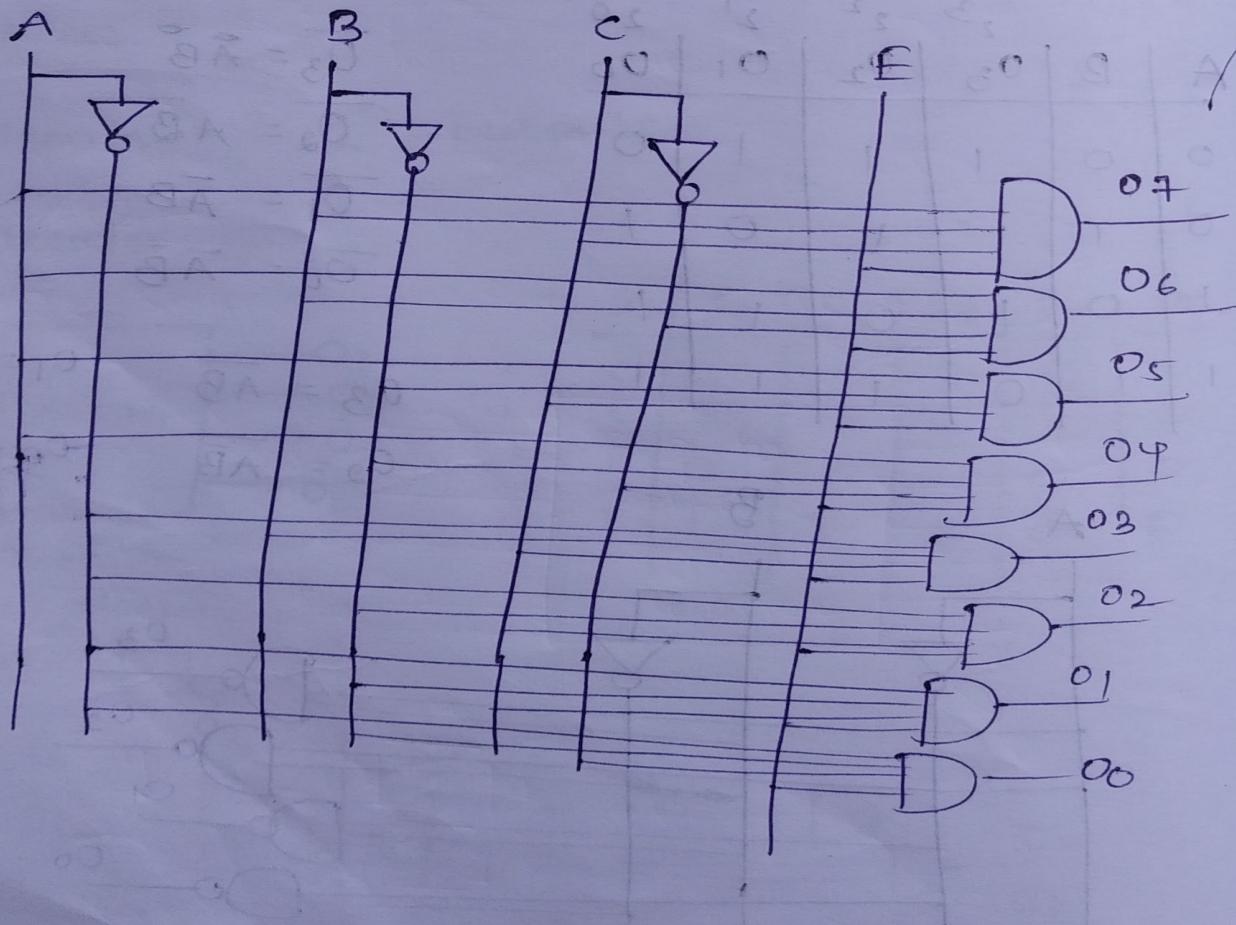
3x8 decoder:-



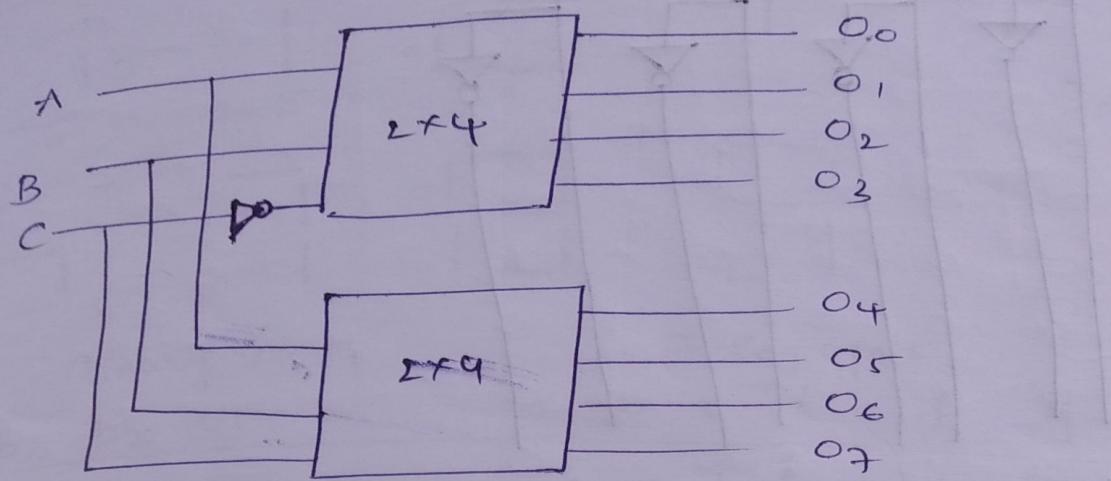
	00	01	10	11	20	21	22	23	24	25	26	27	
	07	06	05	04	03	02	01	00					
A	0	0	0	0	0	0	0	0	1	1	1	1	1
B	0	0	1	1	0	1	0	1	0	1	0	1	0
C	0	1	0	1	1	0	0	1	1	0	1	0	1
E	X	X	X	0	0	0	0	0	0	0	0	0	0

E	A	B	C	07	06	05	04	03	02	01	00	27	26	25	24	23	22	21	20
1	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1
1	0	0	1	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1
1	0	1	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1
1	0	1	1	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1
1	1	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1
1	1	0	1	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1
1	1	1	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1
0	X	X	X	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

$07 = ABCE$
 $06 = AB\bar{C}E$
 $05 = A\bar{B}CE$
 $04 = A\bar{B}\bar{C}E$
 $03 = \bar{A}BCE$
 $02 = \bar{A}\bar{B}CE$
 $01 = \bar{A}\bar{B}\bar{C}E$
 $00 = \bar{A}\bar{B}\bar{C}E$



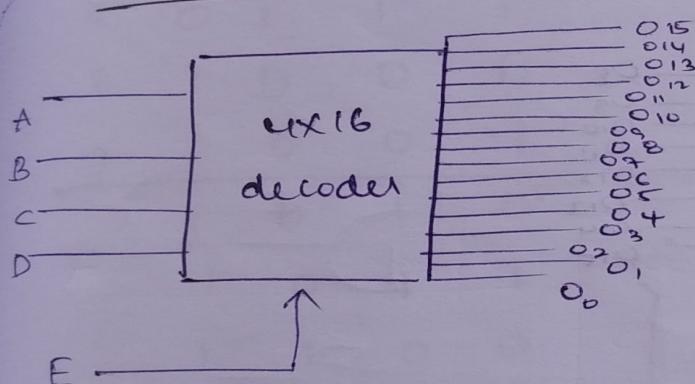
3×8 decoder using 2×4 decoders:-



Same table

Same logic diagram

4x16 decoder:-



E	A	B	C	D	O ₁₅	O ₁₄	O ₁₃	...	O ₃	O ₂	O ₁	O ₀	
1	0	0	0	0	0	0	0	...	0	0	0	1	
1	0	0	0	1									
1	0	0	1	0									
1	0	1	0	1									
1	0	1	1	1									
1	1	0	0	0									
1	1	0	0	1									
1	1	1	1	1	1	1	1	...	1	1	1	1	
0	x	x	x	x	0	0	0	...	0	0	0	0	

Octal to binary encoder

$$2^3 - 2$$

Inpts

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0

0	0	0	0	0	0	0	1
---	---	---	---	---	---	---	---

0	0	0	0	0	0	1	0
---	---	---	---	---	---	---	---

0	0	0	0	0	1	0	0
---	---	---	---	---	---	---	---

0	0	0	0	0	0	0	0
---	---	---	---	---	---	---	---

0	0	0	0	0	0	0	0
---	---	---	---	---	---	---	---

0	0	0	0	0	0	0	0
---	---	---	---	---	---	---	---

0	0	0	0	0	0	0	0
---	---	---	---	---	---	---	---

0	0	0	0	0	0	0	0
---	---	---	---	---	---	---	---

0	0	0	0	0	0	0	0
---	---	---	---	---	---	---	---

0	0	0	0	0	0	0	0
---	---	---	---	---	---	---	---

0	0	0	0	0	0	0	0
---	---	---	---	---	---	---	---

0	0	0	0	0	0	0	0
---	---	---	---	---	---	---	---

0	0	0	0	0	0	0	0
---	---	---	---	---	---	---	---

0	0	0	0	0	0	0	0
---	---	---	---	---	---	---	---

0	0	0	0	0	0	0	0
---	---	---	---	---	---	---	---

0	0	0	0	0	0	0	0
---	---	---	---	---	---	---	---

0	0	0	0	0	0	0	0
---	---	---	---	---	---	---	---

0	0	0	0	0	0	0	0
---	---	---	---	---	---	---	---

0	0	0	0	0	0	0	0
---	---	---	---	---	---	---	---

0	0	0	0	0	0	0	0
---	---	---	---	---	---	---	---

0	0	0	0	0	0	0	0
---	---	---	---	---	---	---	---

outputs

x	y	z
4	2	1

0	0	0
---	---	---

0	0	1
---	---	---

0	1	0
---	---	---

0	1	1
---	---	---

1	0	0
---	---	---

1	0	1
---	---	---

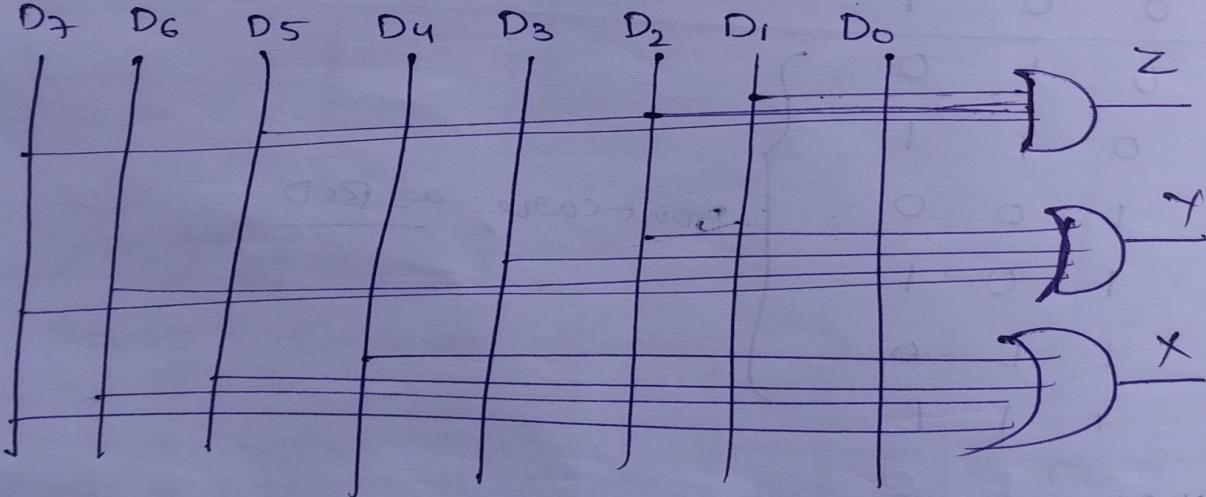
1	1	0
---	---	---

1	1	1
---	---	---

$$Z = D_1 + D_3 + D_5 + D_7$$

$$Y = D_2 + D_3 + D_6 + D_7$$

$$X = D_4 + D_5 + D_6 + D_7$$



Priority encoder:-

$I_3^{2^3}$	$I_2^{2^2}$	$I_1^{2^1}$	$I_0^{2^0}$	Y_1	Y_0
-------------	-------------	-------------	-------------	-------	-------

0	0	0	0	x	x
---	---	---	---	---	---

0	0	0	1	0	0
---	---	---	---	---	---

0	0	1	x	1	1
---	---	---	---	---	---

0	1	x	x	0	2
---	---	---	---	---	---

1	x	x	x	1	3
---	---	---	---	---	---

Priority order

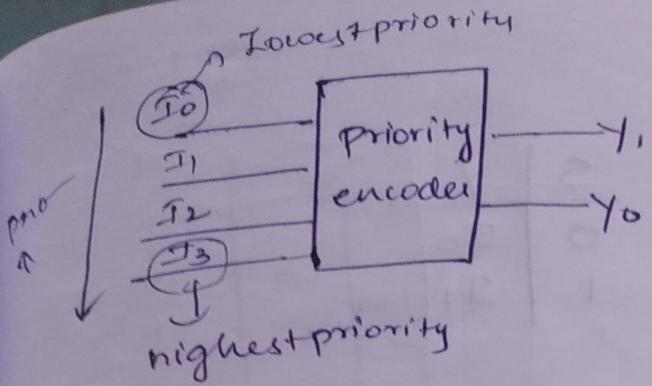
$I_3 > I_2 > I_1 > I_0$

$2^3 > 2^2 > 2^1 > 2^0$

Priority - S

Priority - Y

Priority - X



for Y_1

$I_3 \backslash I_2$	00	01	11	10
00	X	0	0	0
01	1	1	1	1
11	1	1	1	1
10	1	1	1	1

$$Y_1 = I_2 + I_3$$

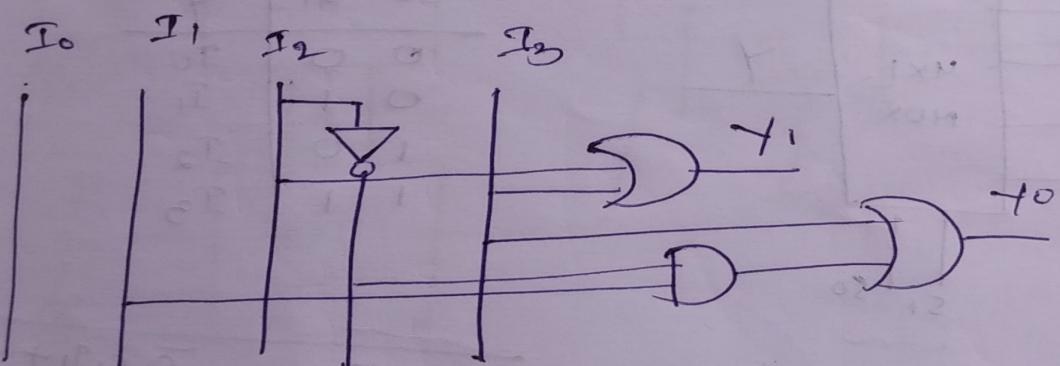
2 - qud octs

for Y_0

$I_3 \backslash I_2$	00	01	11	10
00	X	0	1	1
01	0	0	0	0
11	*	*	*	*
10	1	1	1	1

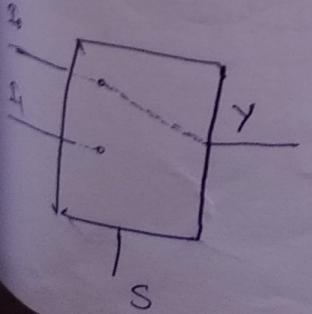
1 - qud 1 - oct

$$Y_0 = I_3 + I_1 I_2$$



Multiplexers:-

- 1 - output
- 2 - select lines
- 2^n - data inputs



types 2×1 MUX

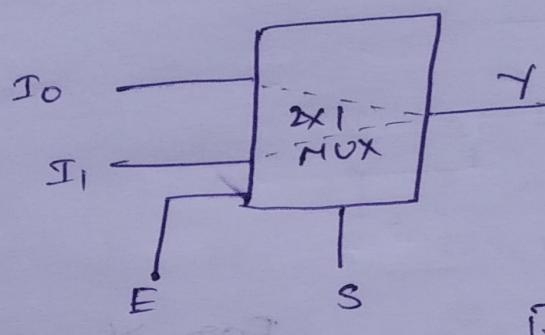
4×1 MUX

8×1 MUX

16×1 MUX

no. of data inputs \times no. of o/p

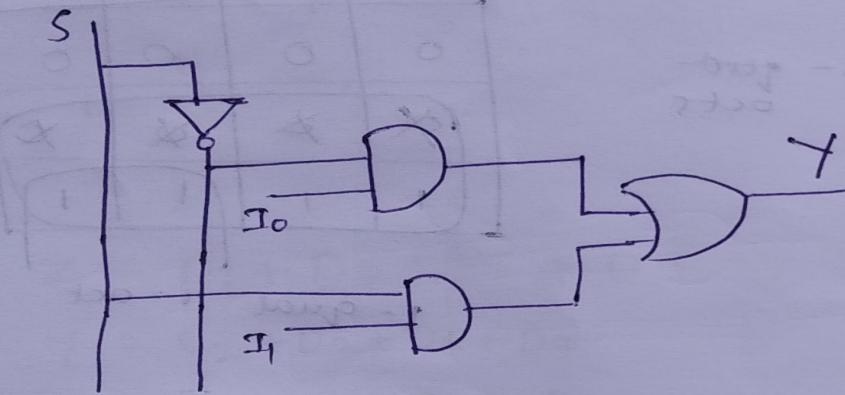
2x1 MUX:



S	Y
0	I0
1	I1

$$Y = \bar{S} I_0 + S I_1$$

logic diagram:-

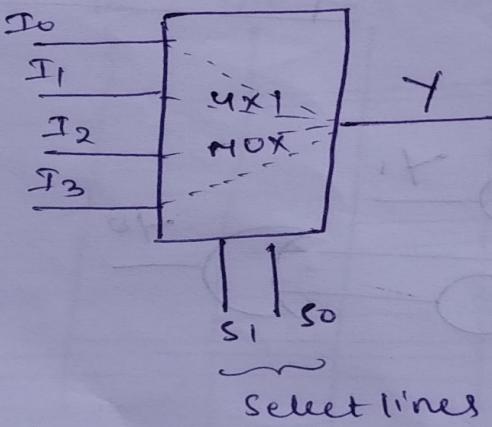


0	0	0	0
0	0	1	1
1	0	1	1
1	1	1	1

$$\text{ct} + \text{ct} = 15$$

4x1 MUX:-

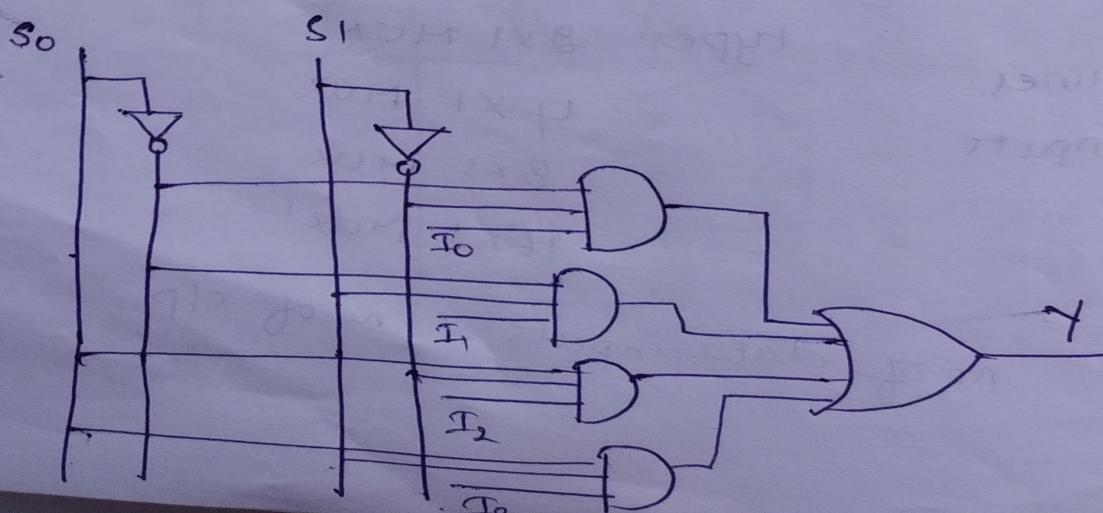
data i/p lines



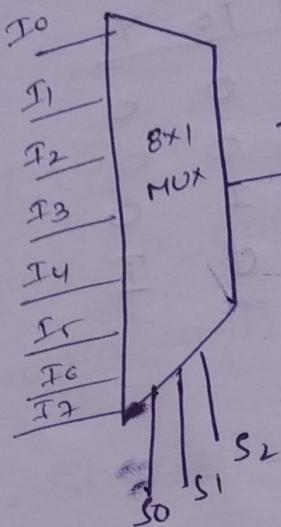
S0	S1	Y
0	0	I0
0	1	I1
1	0	I2
1	1	I3

$$Y = \bar{S}_0 \bar{S}_1 I_0 + \bar{S}_0 S_1 I_1 + S_0 \bar{S}_1 I_2 + S_0 S_1 I_3$$

logic diagram:-



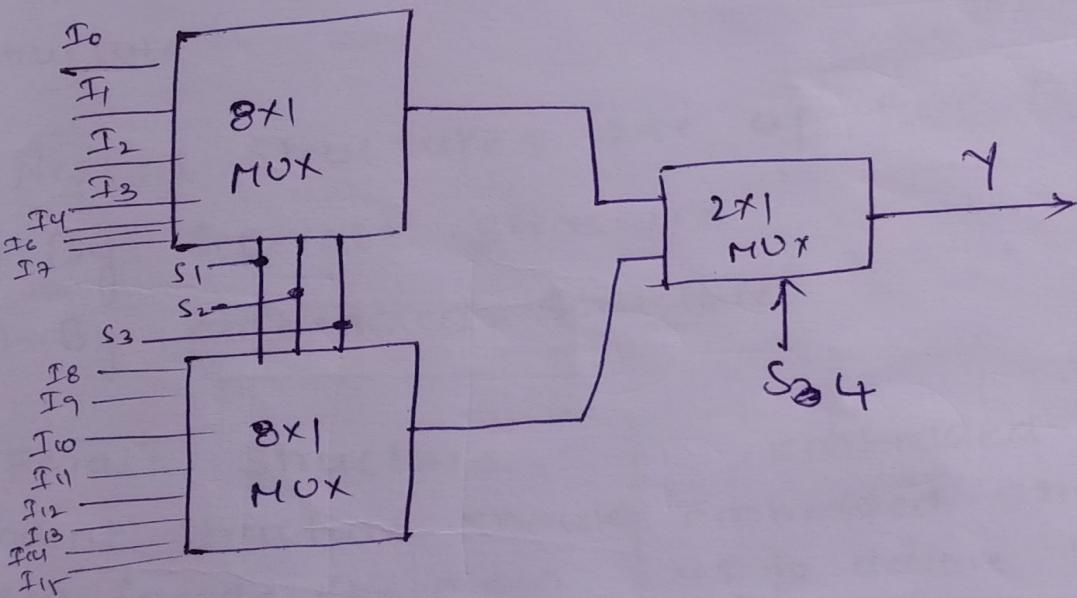
8x1 MUX:-



S_0	S_1	S_2	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

$$Y = \overline{S_0} \overline{S_1} \overline{S_2} I_0 + \overline{S_0} \overline{S_1} S_2 I_1 + \overline{S_0} S_1 \overline{S_2} I_2 + \overline{S_0} S_1 S_2 I_3 \\ + S_0 \overline{S_1} \overline{S_2} I_4 + S_0 \overline{S_1} S_2 I_5 + S_0 S_1 \overline{S_2} I_6 \\ + S_0 S_1 S_2 I_7$$

16x1 MUX Using 8x1 MUX:-

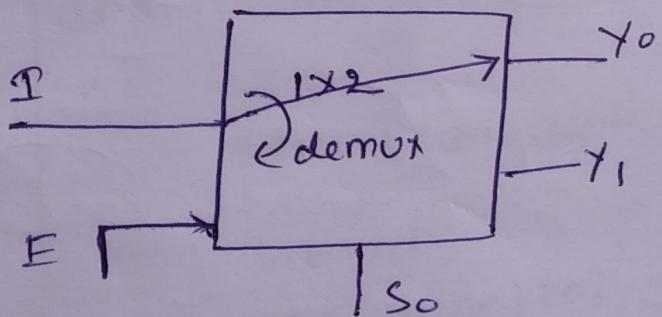


Demultiplexer:-

Output: 2^n

Select lines: n

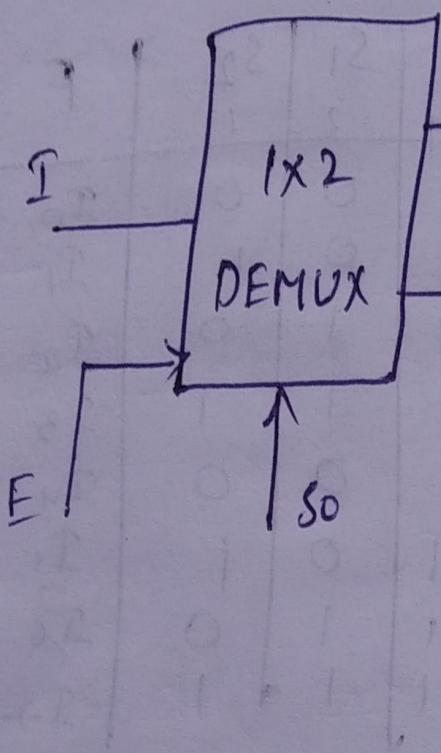
Input : 1



Simple reverse operation of multiplexer

if $S_2 \ S_1 \ S_0 \ | 01P$

1x2 DEMUX :-



y_0

y_1

E	S_0	y_0	y_1
0	0	0	0
0	1	0	0
1	0	1	0
1	1	0	1

$$y_0 = E \bar{S}_0 I$$

$$y_1 = E S_0 I$$

