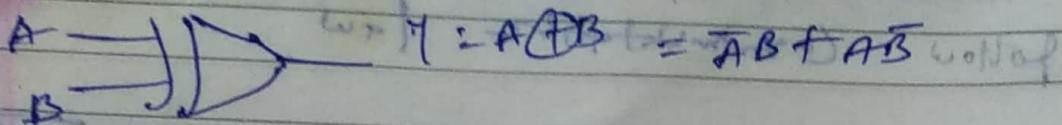


- Ex-or Gate

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



A	B	Y
0	0	0
0	1	1
1	0	1
1	1	0

$$Y = \sum m(1, 3) \\ = \overline{A} \cdot B + A \cdot \overline{B}$$

If no of 1's are odd ans is logic 1

so cktr is called odd 1's detector.

$$\begin{array}{l} A \oplus A = 0 \\ A \oplus A = 1 \end{array}$$

$$A \oplus 0 = A$$

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

Lecture 7

Q If $A \oplus B = C$ then $A \oplus C = ?$

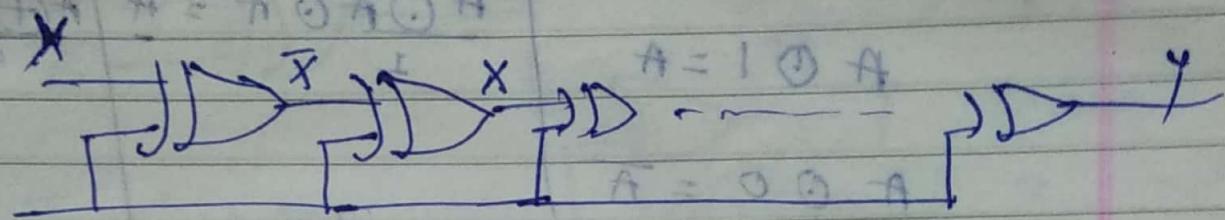
Given $A \oplus B = C$

$$\therefore A \oplus C = B$$

Lecture-7

gate

Q The ckt shown in fig contains $\frac{20}{\text{NOR}} \text{ gates}$ then the op of y is?



1.

$$a) 0 \oplus A = A$$

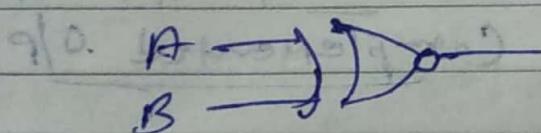
$$b) 1 \oplus A = \overline{A} \oplus A$$

c) X

d) \overline{X} (No sup old in PV makes
to compare with this much)

EX-NOR (XNOR)

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



A	B	Y
0	0	1
0	1	0
1	0	0
1	1	1

$$\begin{aligned} Y(A, B) &= \sum m(0, 3) \\ &= \overline{A}\overline{B} + AB \end{aligned}$$

\Leftrightarrow Coincidence logic gate

\Leftrightarrow Equivalence gate

$$\begin{aligned}
 A \odot A &= 1 \\
 A \odot \bar{A} &= 0 \\
 A \odot 1 &= A \\
 A \odot 0 &= \bar{A}
 \end{aligned}$$

$$A \oplus A = 1$$

part stop

$$A \odot A \oplus A = A$$

$$A \oplus A \oplus A = A$$

$$A \oplus B = \overline{A \odot B}$$

$$A \oplus B \oplus C = A \odot B \odot C$$

when variables are odd NOR &
XNOR will give same o/p

when variables are even NOR &
XNOR will give complemented o/p

Q Calculate $\overline{A} \oplus B = C \oplus B$ suppose $\overline{A} = C$

suppose we take $A = C$

$$\begin{aligned}
 (\overline{A} \oplus B) &\Rightarrow (\overline{C} \oplus B) \\
 \overline{A} + \overline{B} &\Rightarrow \\
 &= C\bar{B} + \bar{C}B \\
 &= \overline{AB} + AB \\
 &= \overline{A} \odot B
 \end{aligned}$$

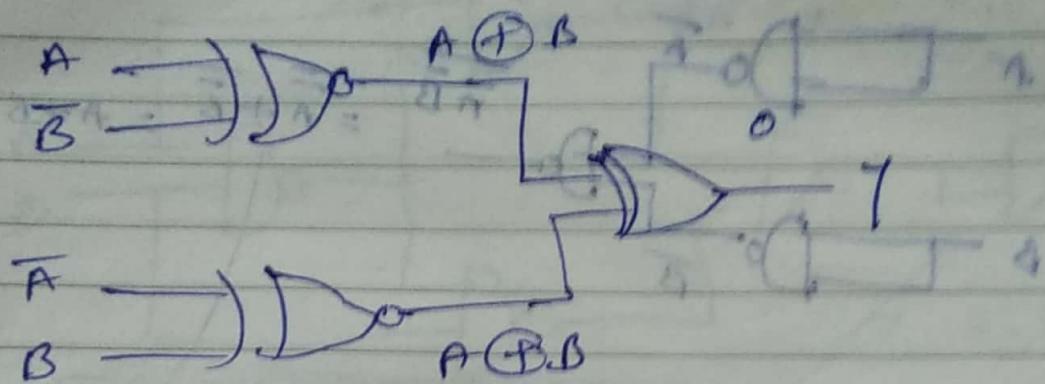
put $C = \bar{A}$

$$A \oplus \bar{B} = A \odot B$$

$$\overline{A} \odot B = A \oplus B$$

$$A \odot \bar{B} = A \oplus B$$

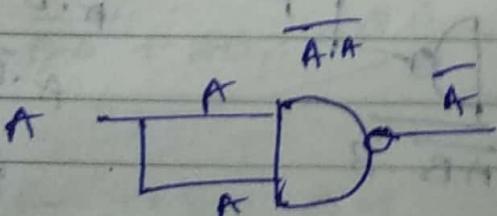
Q - For the shown in figure calculate O/P Y



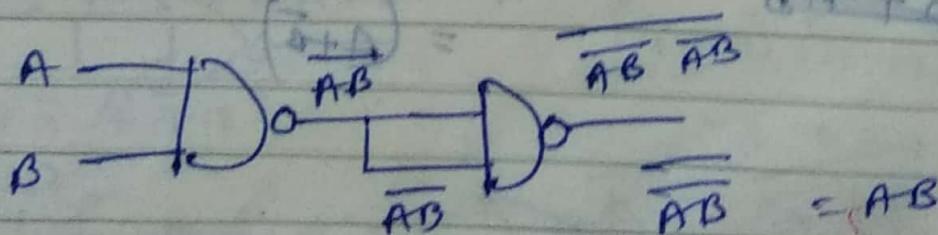
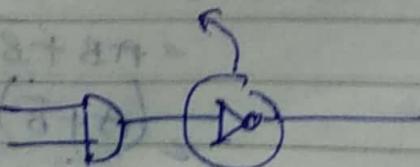
* Universal gate

NAND as Universal gate

i) NOT

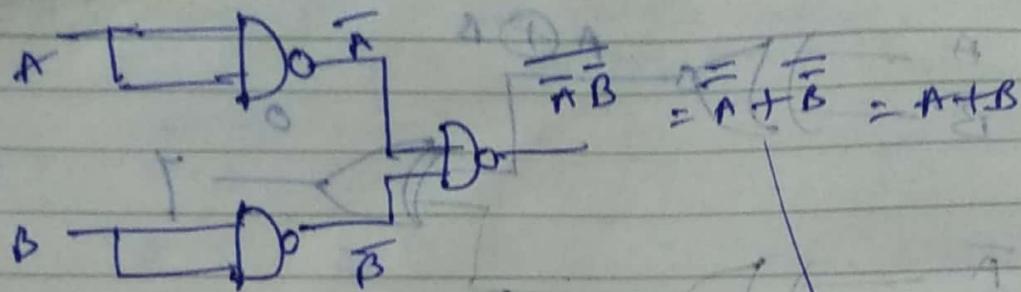


ii) AND

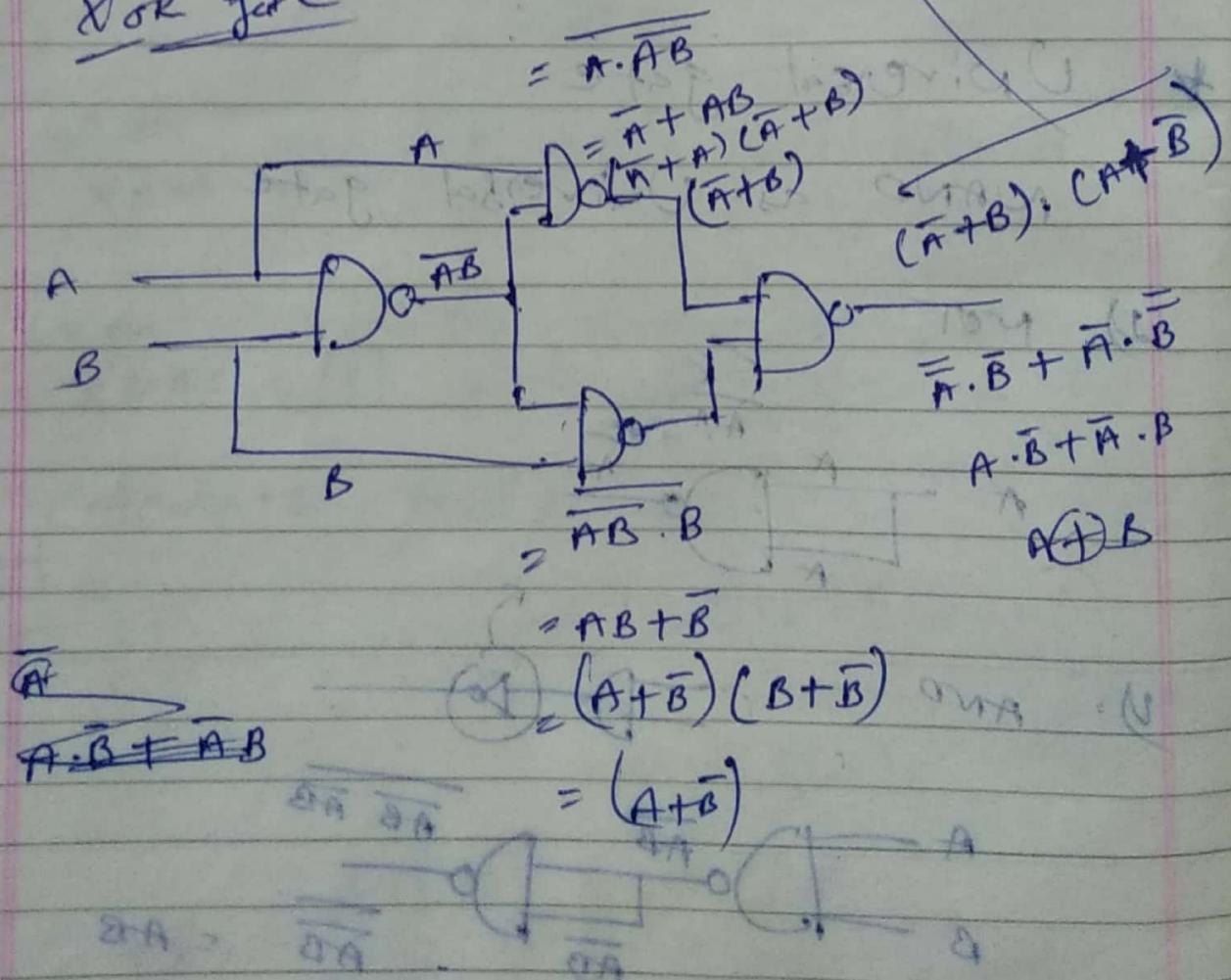


$$n \rightarrow D \rightarrow \overline{A \cdot B} \quad \overline{\overline{A} \cdot \overline{B}} = A + B$$

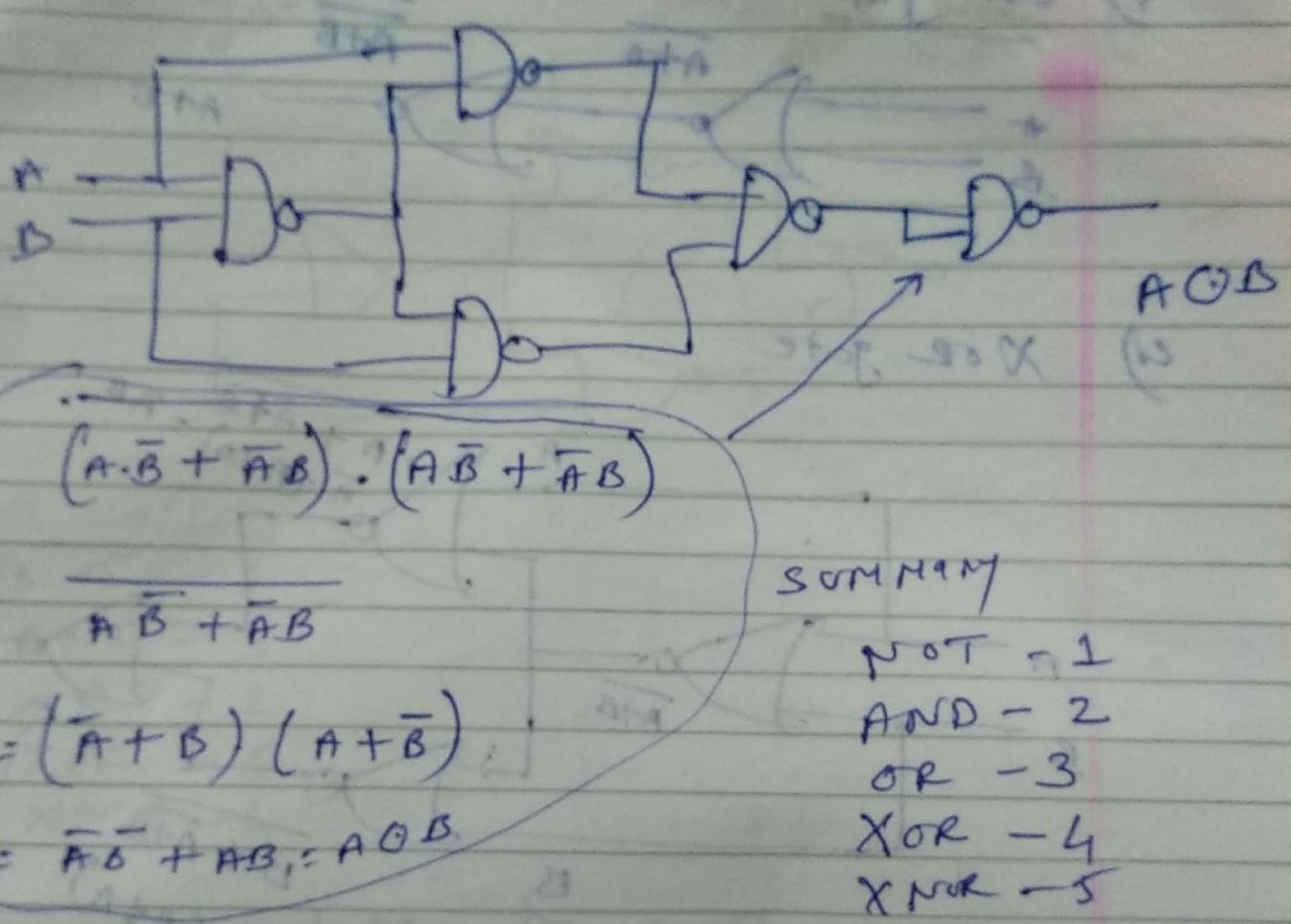
b) OR



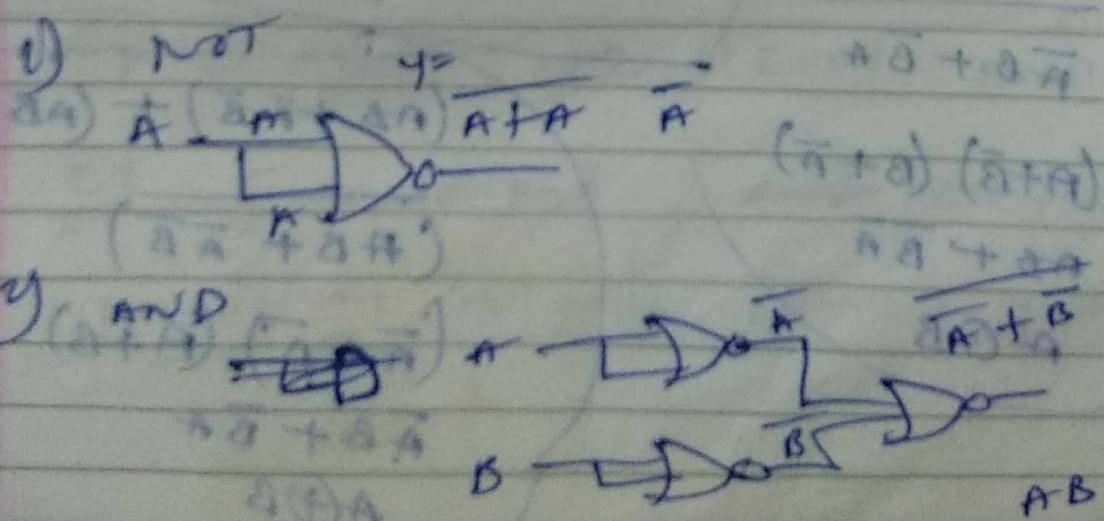
c) NOR gate



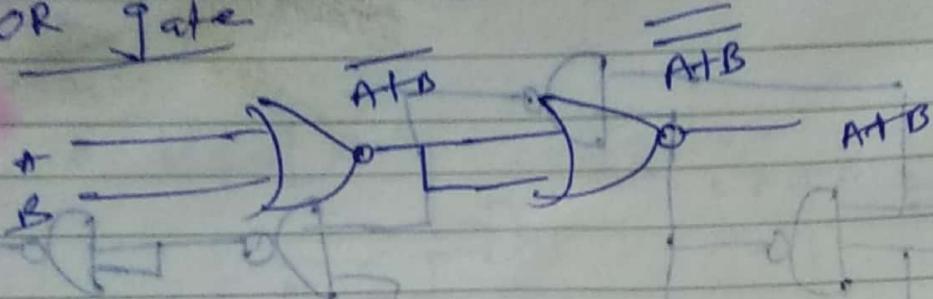
5) XNOR gate



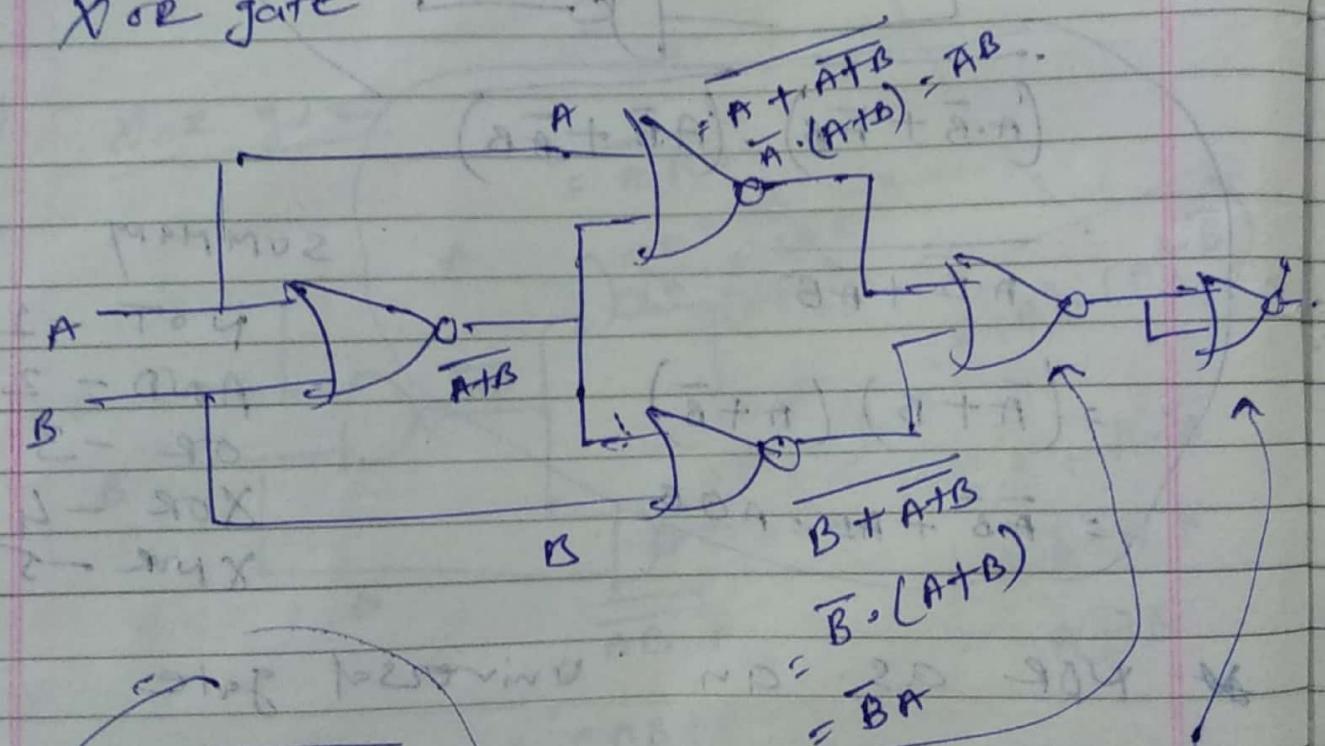
6. NOR as an universal gate



3) OR gate



4) NOR gate



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

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

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

$$A \oplus B$$

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

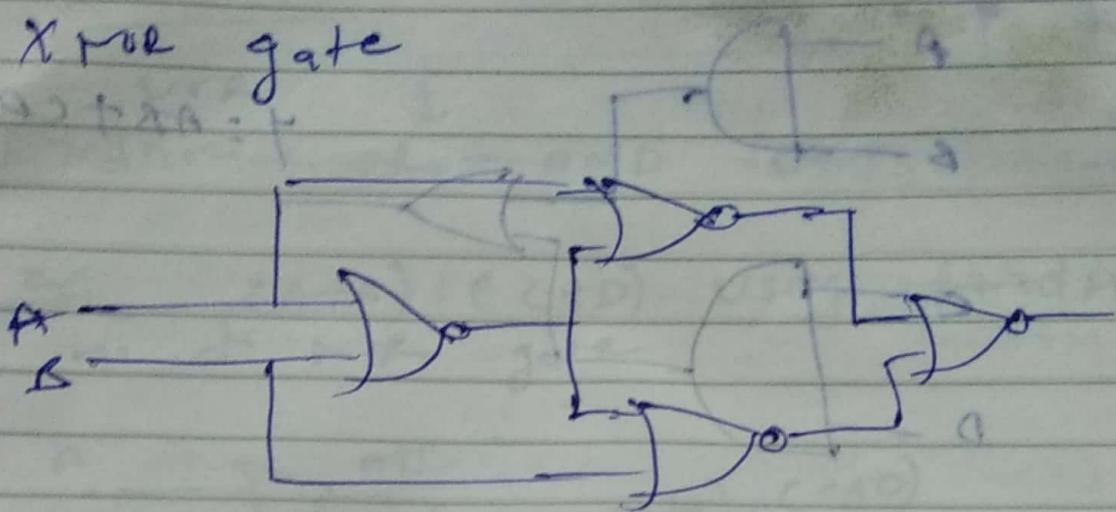
$$(AB + \overline{A}\overline{B})$$

$$(\overline{A} + B), (A + B)$$

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

$$A \oplus B$$

④ Xnor gate



NOT - 1

AND - 3

OR - 2

XOR - 5

Xnor - 4

Lecture :- 8

Digital circuit

Combinational circuit

Sequential circuit

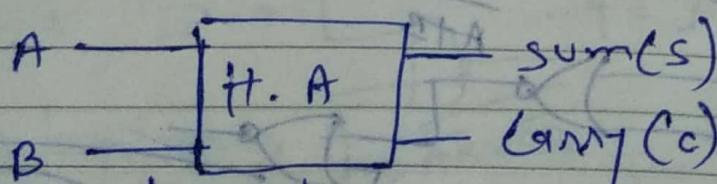
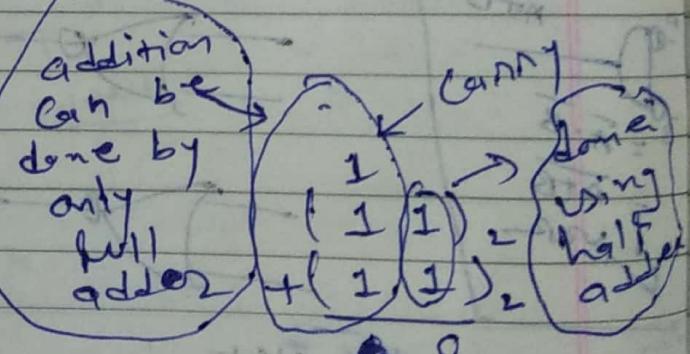
- No feedback - Memory
- No memory - Memory

* Combinational circuit

- Adders
- Subtractors
- Multiplexers
- Demultiplexers
- Decoder
- Encoder

* ADDERS

Half Adder

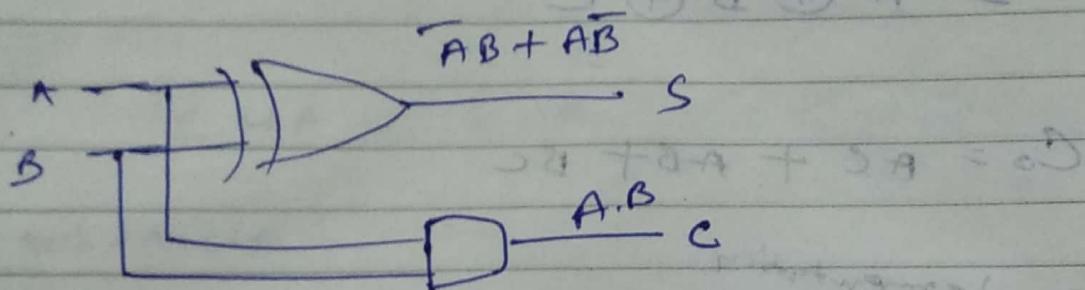


A	B	\oplus	C
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

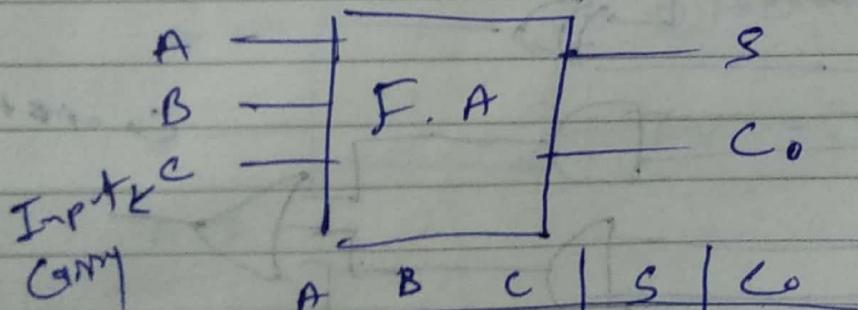
Sum : $\bar{A}B + A\bar{B}$

$$A \oplus B = (\bar{A}, B, 1) \rightarrow S = (\bar{A}, B)^2$$

Carry : $\bar{A}A \cdot B + \bar{B}A + B\bar{A} = 2$



* Full Adder



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

$$S = \sum_{m=0}^{7} m$$

	$A \setminus BC$	$\bar{B}C$	$B\bar{C}$	BC	$B\bar{C}$	$\bar{B}\bar{C}$
\bar{A}	0	1	0	1	0	1
A	1	0	1	0	1	0
	0	5	7	4	6	3

	$A \setminus BC$	$\bar{B}C$	$B\bar{C}$	BC	$B\bar{C}$	$\bar{B}\bar{C}$
\bar{A}	0	0	0	1	1	0
A	0	1	1	0	0	1
	0	5	7	4	6	3

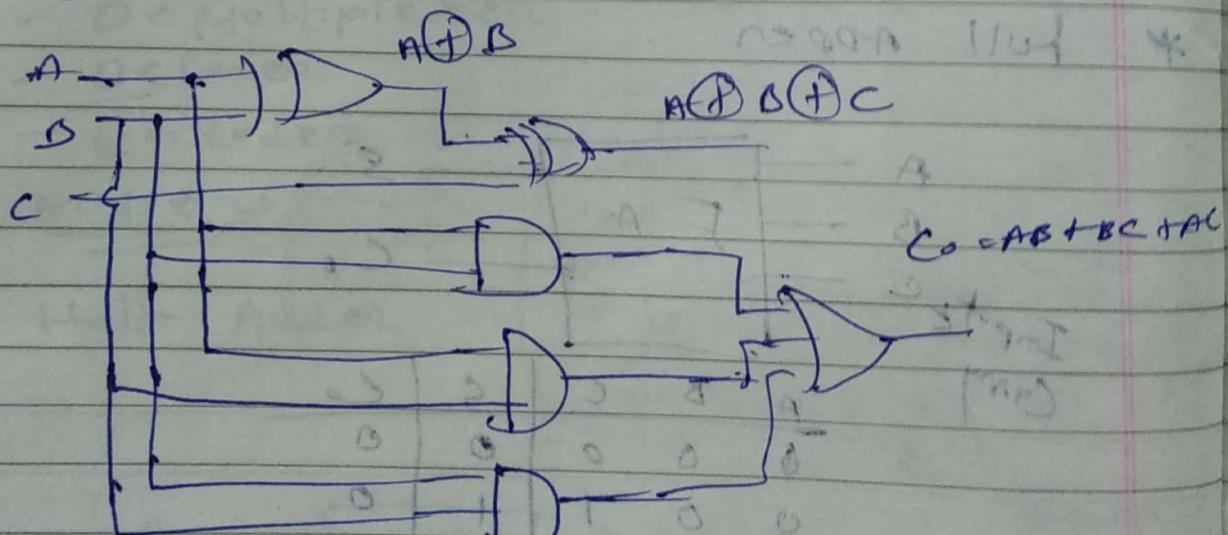
$$S(C_A, B, C) = \Sigma m(1, 2, 4, 7)$$

$$S = \bar{A}\bar{B}C + \bar{A}B\bar{C} + A\bar{B}\bar{C} + ABC$$

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

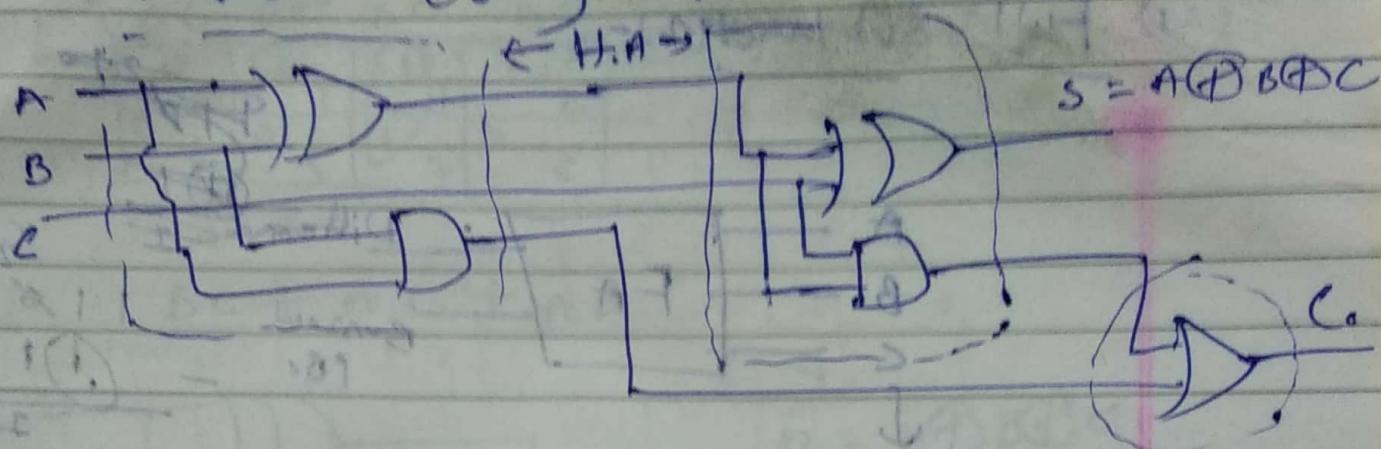
$$C_0 = AC + AB + BC$$

Implementation



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

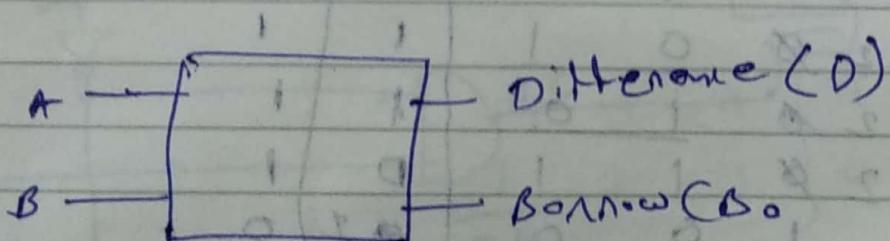
Full adder using half adder



$2 \text{ HA} + 1 \text{ OR}$

* Subtractor

i) Half subtractor:-

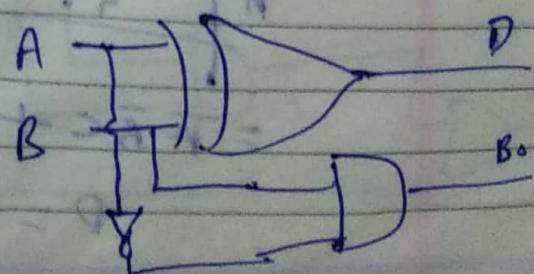


$A - B$

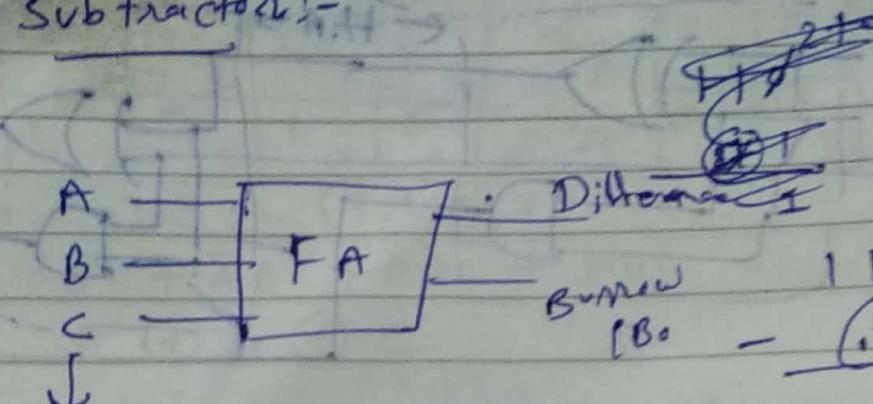
A	B	D	B_{10}
0	0	0	0
0	1	1	1
1	0	1	0
1	0	0	0

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

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



v) Full Subtractor



Borrow
input

$$\begin{array}{r} 110 \\ - 101 \\ \hline 001 \end{array}$$

A	B	C	D	B_0
0	0	0	0	0
1	0	1	1	1
0	1	0	1	1
1	1	1	0	1
1	0	0	1	0
1	0	1	0	0
1	1	0	0	0
3X	1	1	1	1

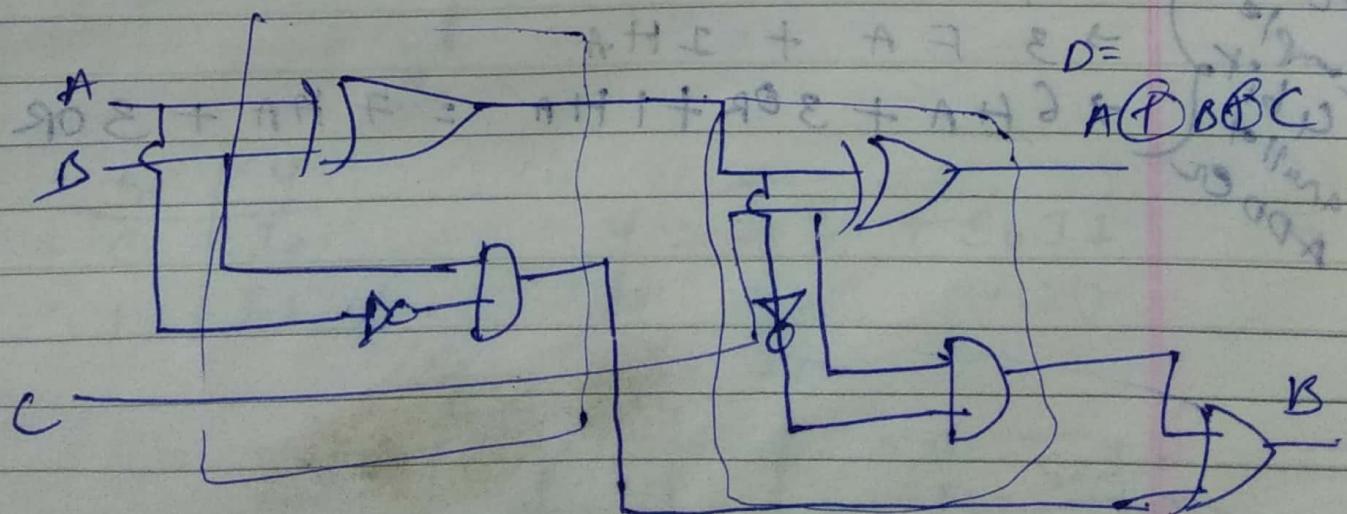
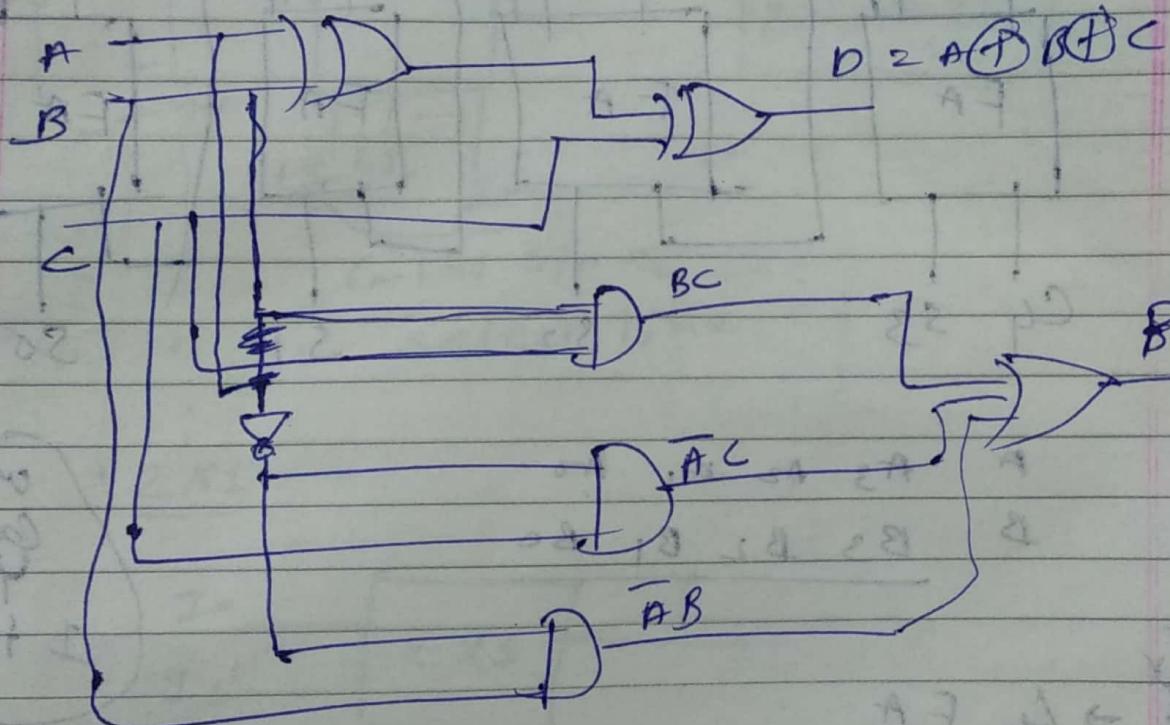
D	$\bar{A} \bar{B} \bar{C}$	$\bar{B} \bar{C}$	$\bar{B} C$	$B \bar{C}$	$B C$
D_A	0 0	1 1	0 1	1 0	0 1
D_B	1 1	0 0	1 0	0 1	1 0

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

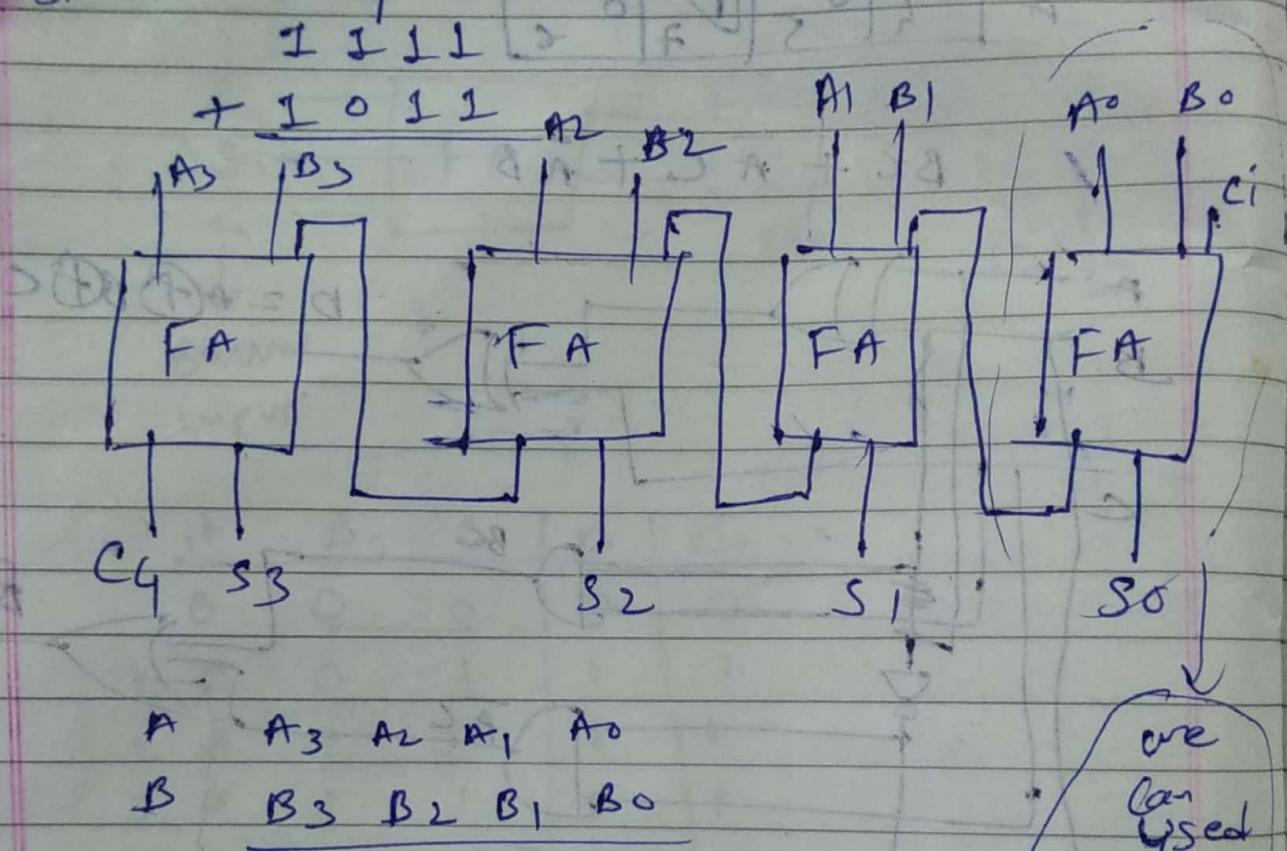
$$D = A \oplus B \oplus C$$

	$\bar{B}C$	$\bar{B}c$	BC	$B\bar{C}$
\bar{A}	0	1	1	1
A	0	0	1	0
	0	5	1	7

$$BC + \bar{A}C + \bar{A}B$$



* parallel adder - It add all bit simultaneously.



$$\begin{array}{r}
 A \quad A_3 \ A_2 \ A_1 \ A_0 \\
 B \quad \underline{B_3 \ B_2 \ B_1 \ B_0}
 \end{array}$$

we can use
1 HA

require
to implement
I adder
parallel
adder

$$\rightarrow 4 \text{ FA}$$

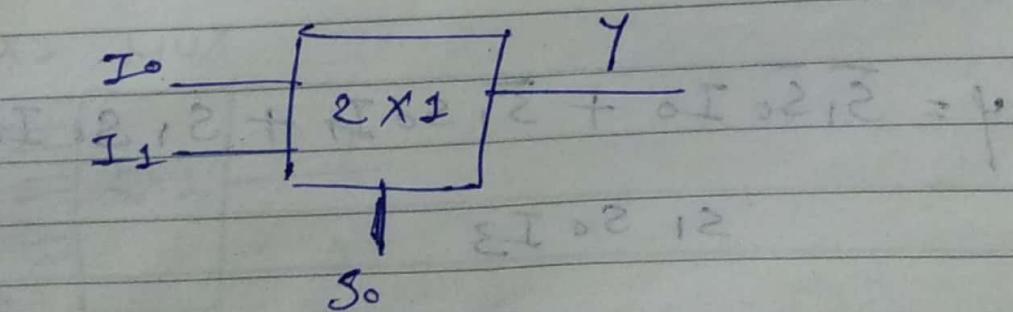
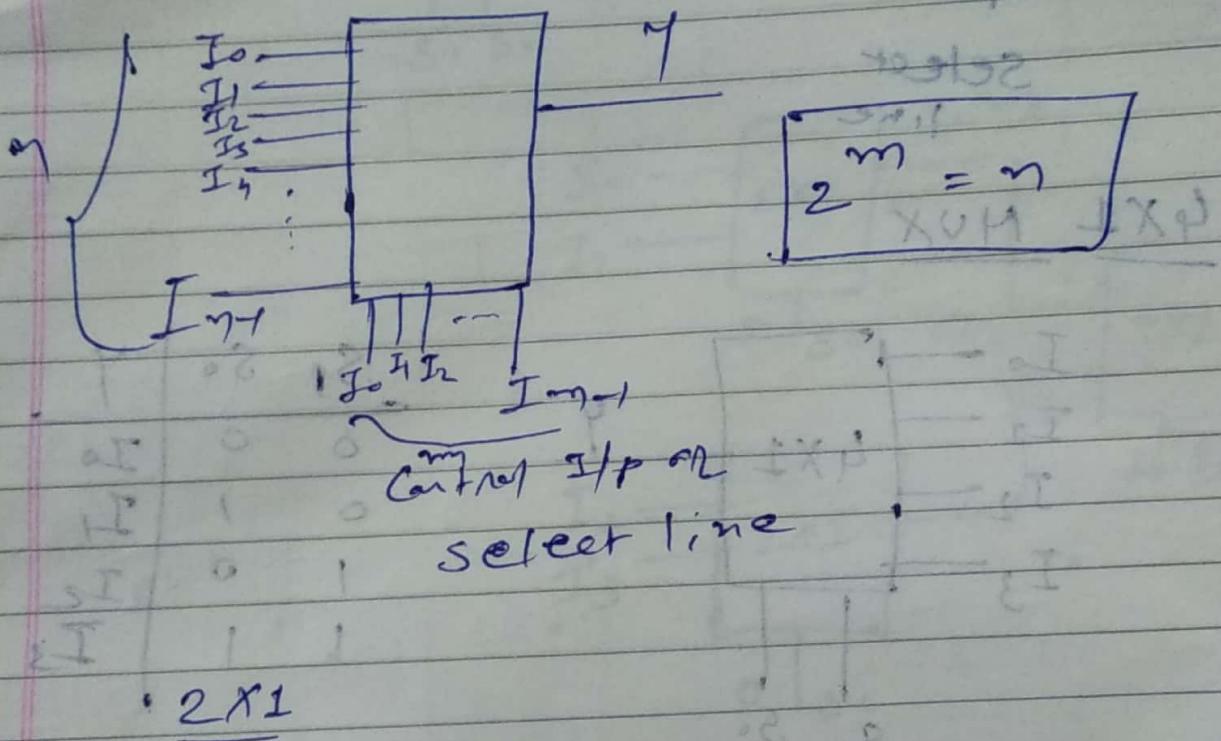
$$\rightarrow 3 \text{ FA} + 1 \text{ HA}$$

$$\rightarrow 6 \text{ HA} + 3 \text{ OR} + 1 \text{ HA} = 7 \text{ HA} + 3 \text{ OR}$$

Lecture - 9

Multiplexer :-

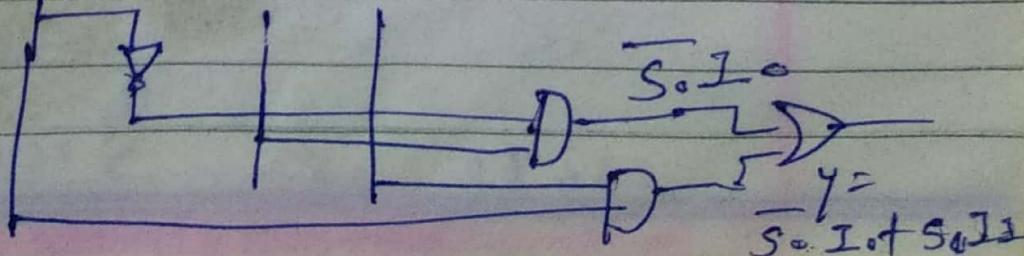
Data Selector

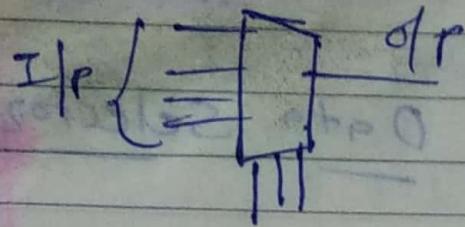


$$\begin{array}{c|cc}
 S_0 & X & Y \\
 \hline
 0 & I_0 & I_0 \\
 1 & I_1 & I_1
 \end{array}$$

$y = S_0 I_0 + S_0 I_1$

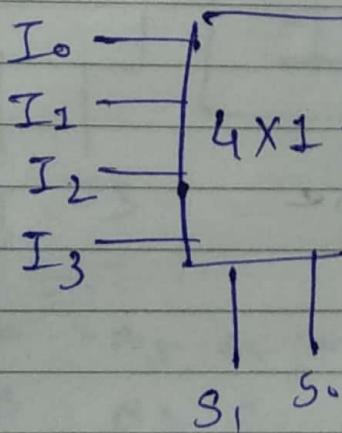
X \rightarrow I_0 \rightarrow I_1





select
line

4x1 MUX



S_1	S_0	Y
0	0	I_0
0	1	I_1
1	0	I_2
1	1	I_3

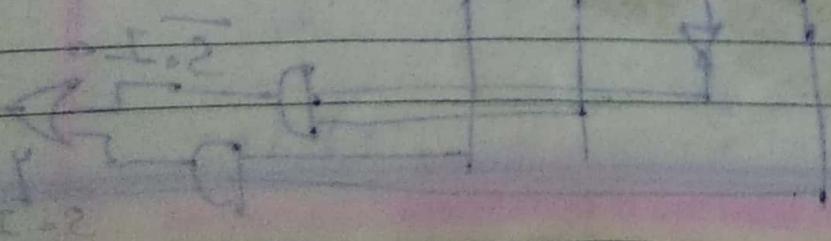
$$Y = \bar{S}_1 \bar{S}_0 I_0 + \bar{S}_1 S_0 I_1 + S_1 \bar{S}_0 I_2 + S_1 S_0 I_3$$

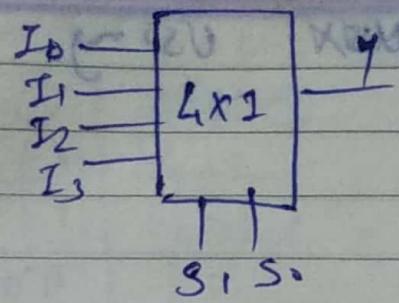
$S_1 \bar{S}_0 I_3$

Q

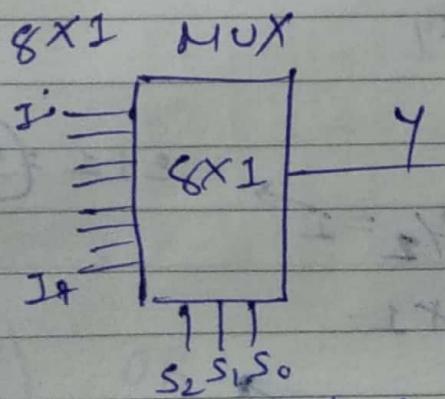
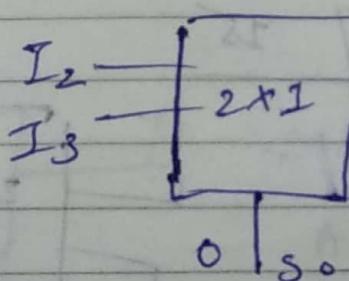
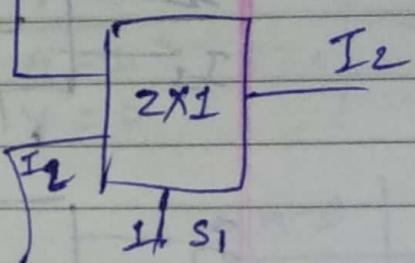
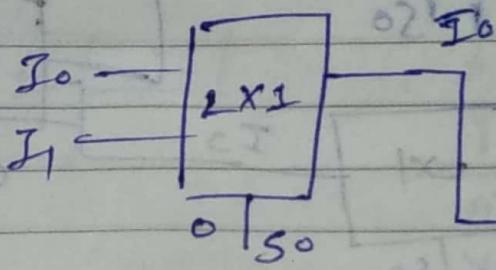
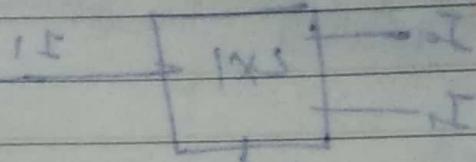
Implement a 4x1 MUX

using 2x1 MUX





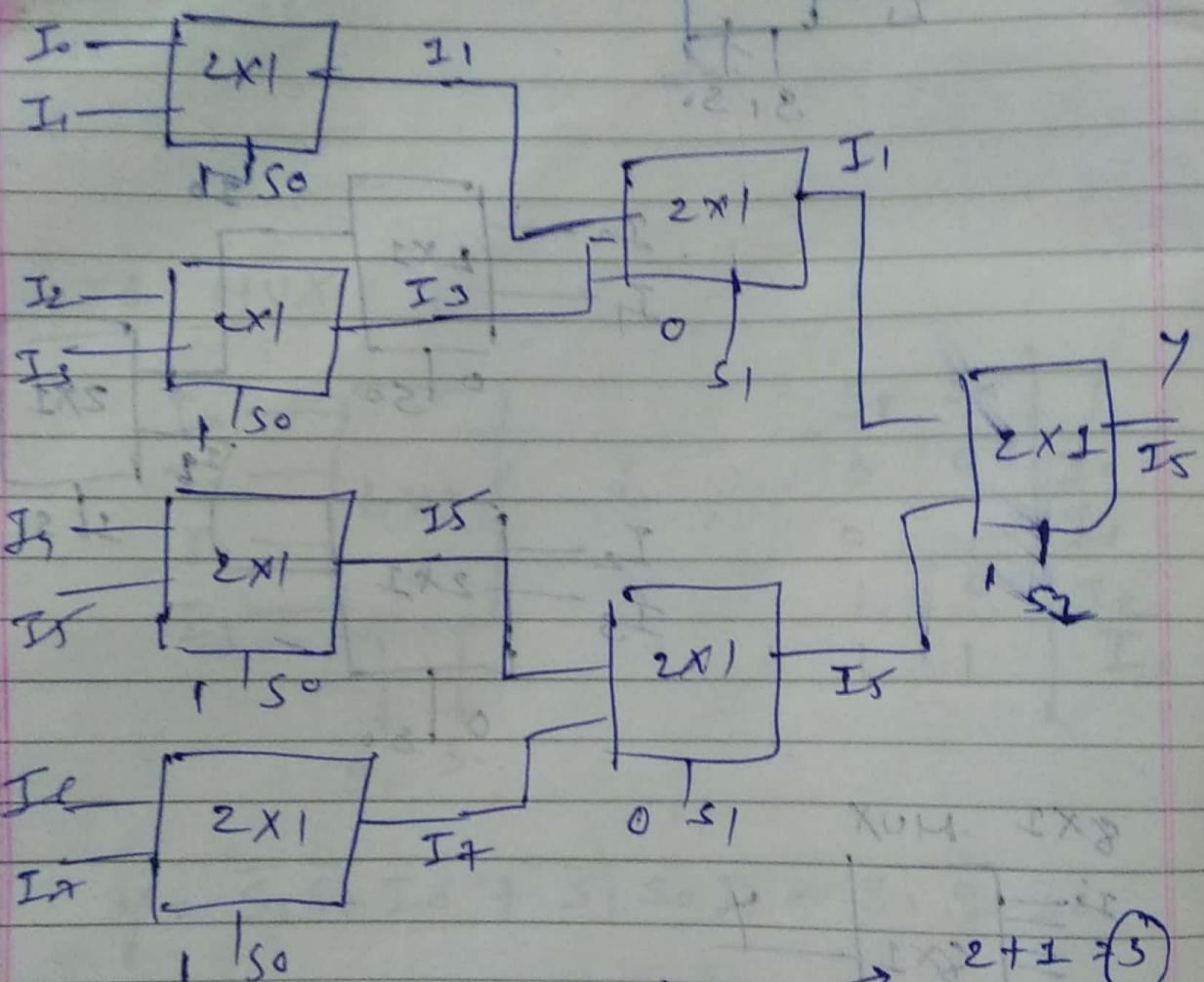
S_1 S_0 Y



$S_2 + S_1, S_0$	7	I_0	I_1	I_2	I_3	I_4	I_5	I_6	I_7
0 0 0	7	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						

1 of
8x1

Implement 8x1 MAX using 2x1



$$4 \times 1 \xrightarrow{4/2 = 2} 2 \times 1 \quad 2+1=5$$

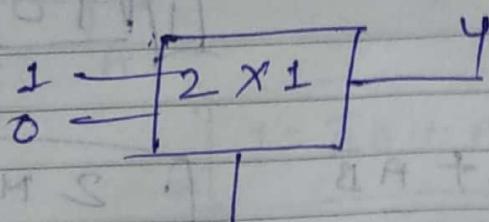
$$8 \times 1 \xrightarrow{8/2 = 4} 4 \times 1 \xrightarrow{4/2 = 2} 2 \times 1 \quad 4+2+1=7$$

$$16 \times 1 \xrightarrow{16/4 = 4} 4 \times 1 \xrightarrow{4/2 = 2} 2 \times 1 \quad 4+1=5$$

$$256 \times 1 \xrightarrow{256/16 = 16} 16 \times 1 \xrightarrow{16/2 = 8} 8 \times 1 \quad 16+1=17$$

Now we will implement All
Logic gate Using Multiplexer
(2x1)

1) NOT



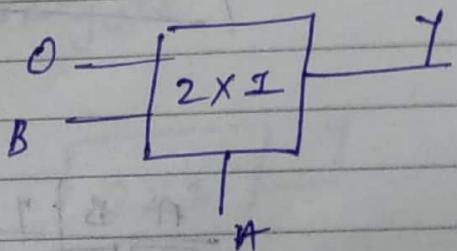
$$A = 0$$

$$Y = 1$$

$$A = 1$$

$$Y = 0$$

y AND

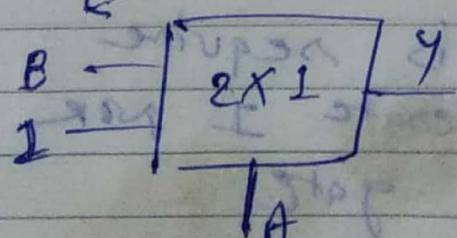


A	Y
0	0
1	B

A	B	Y
0	0	0
0	1	0
1	0	0
1	1	1

$$Y = \bar{A} \cdot 0 + AB \\ = AB$$

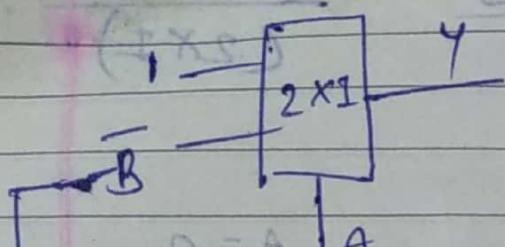
3) OR



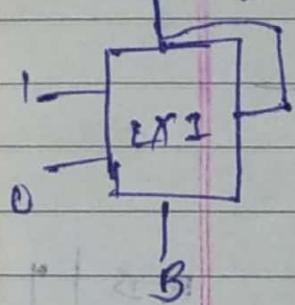
$$Y = \bar{A}B + A = (\bar{A} + \bar{A})(A + B) = A + B$$

A	B	Y
0	0	0
0	1	1
1	0	1
1	1	1

i) NAND



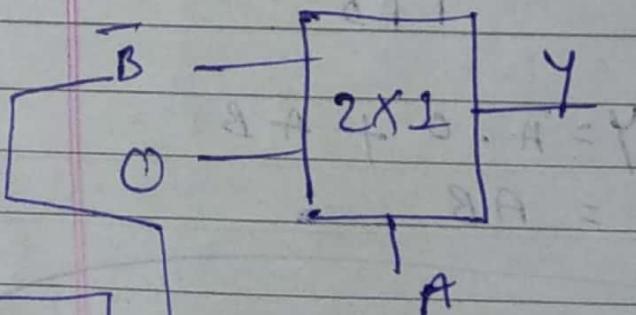
A	B	Y
0	0	1
0	1	0
1	0	0
1	1	0



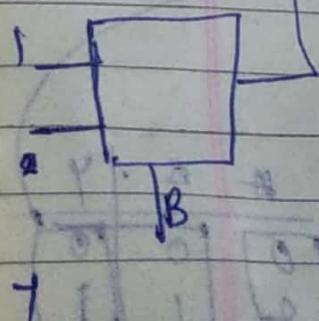
$$\begin{aligned}
 Y &= \bar{A} \cdot 1 + A \bar{B} \\
 &= \bar{A} + A \bar{B} \\
 &= (\bar{A} + A)(\bar{A} + \bar{B}) \\
 &= (\bar{A} + \bar{B}) \\
 &= \bar{A} \bar{B}
 \end{aligned}$$

2 MUX
require
to generate
NAND gate

ii) NOR



A	B	Y
0	0	1
0	1	0
1	0	0
1	1	0



2 MUX is require
to generate 1 NOR
gate.

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

~~$$= \bar{A} \bar{B} + A \cdot 0$$~~

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

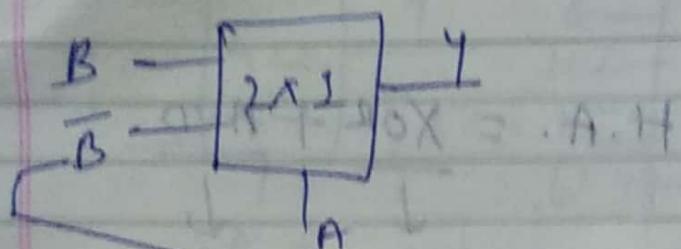
IGATE | MUX
 NOT - 1
 NAND - 1
 NOR - 1
 OR - 1

NAND - 2
 NOR - 2
 EX-OR - 2

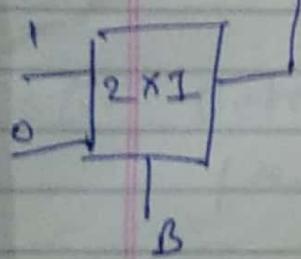
XNOR - 2

+ v2)

- EX-OR + gate logic to make
 half full adder

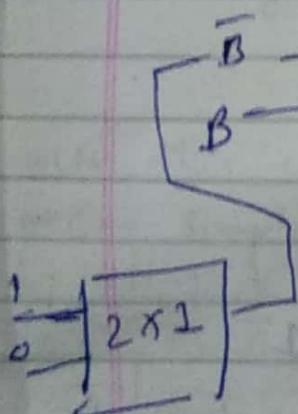


A	B	Y
0	0	0
0	1	1
1	0	1
1	1	0



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

- EX-NOR gate



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

A	B	Y
0	0	1
0	1	0
1	0	0
1	1	1

$$\Delta A = Y$$

Q How Many no. of 2×1 MUX required
for Implementation of Half adder

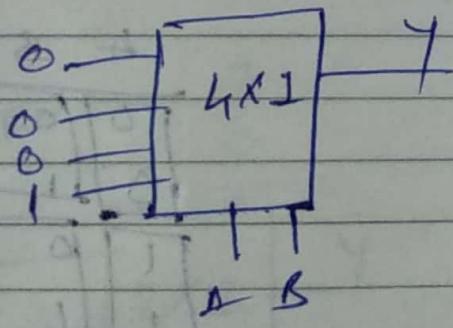


$$\text{H.A.} = \frac{\text{XOR} + \text{AND}}{2 + 1}$$

$$= 3$$

Q AND gate

Implement AND gate using 4×2



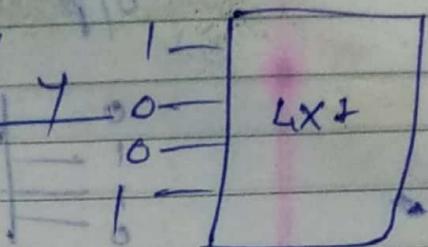
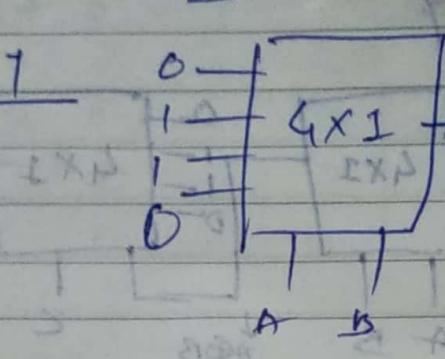
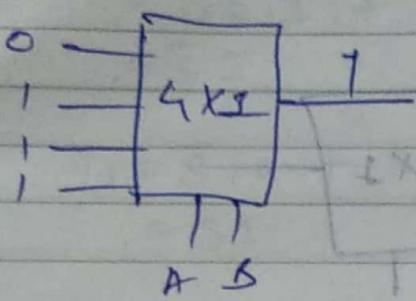
$$Y = \bar{A}\bar{B} \cdot 0 + \bar{A}B \cdot 0 + A\bar{B} \cdot 0 + AB \cdot 1$$

$$Y = AB$$

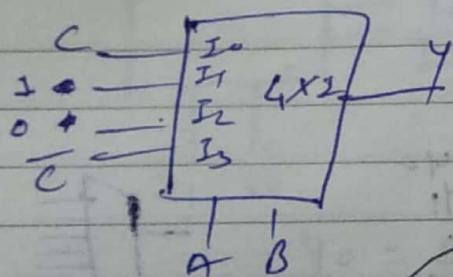
Using 4×1 MUX

Only $MUX(4 \times 2)$
require
For each gate
EX-NOR

OR in majority EXOR



Q- Determine the minimized expression for o/p Y



$$Y = \overline{A}\overline{B} \cdot C + \overline{A}B \cdot \underline{\overline{C}} + \overline{A}\overline{B} \cdot \underline{\overline{C}}$$

$A\overline{B}$

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

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

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

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

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

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

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

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

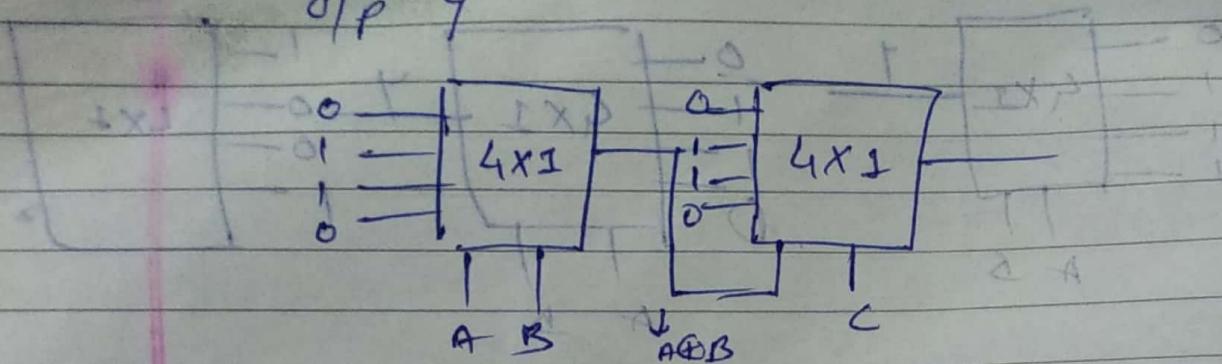
$$= B\overline{C} + \overline{A}C$$

(consensus
Theorem)

	$\overline{B}\overline{C}$	$\overline{B}C$	BC	$B\overline{C}$
\overline{A}	1	1	1	1
A	1	1	1	1
	0	1	1	0

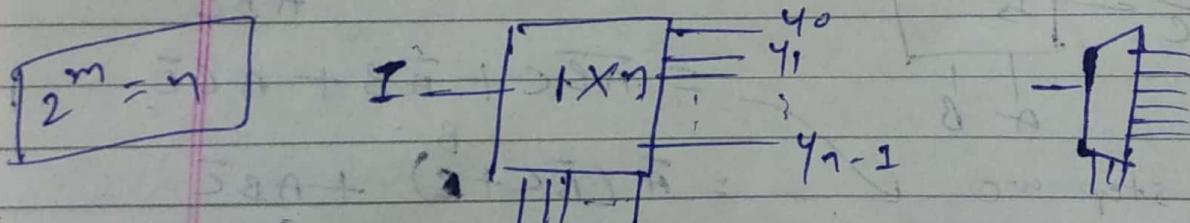
$$= \overline{A}C + B\overline{C}$$

Q Determine the expression for
O/P Y



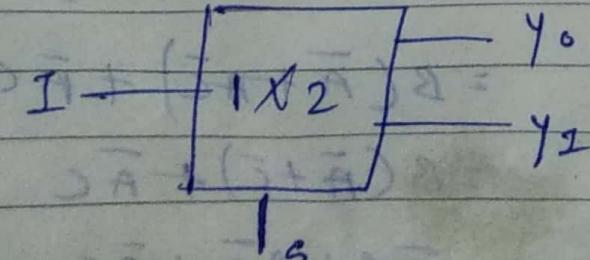
$$Y = A \oplus B \oplus C$$

* Demultiplexer



$$\bar{I} X_2 (S+2) A =$$

$$S A + \bar{S} A + 2 A =$$



S	Y0	Y1
0	I	0
1	0	I

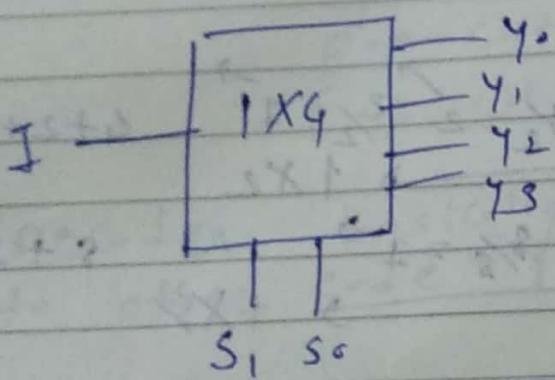
$$Y_0 = \bar{S}_1 \cdot I + S_0 \cdot 0$$

$$= \bar{S}I$$

$$Y_1 = \bar{S} \cdot 0 + S I$$

$$= SI$$

1X4



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

$$Y_0 = \bar{S}_1 \bar{S}_0 I + \cancel{\bar{S}_1 S_0 \cdot 0} + \cancel{S_1 \bar{S}_0 \cdot 0} + \cancel{S_1 S_0 \cdot 0}$$

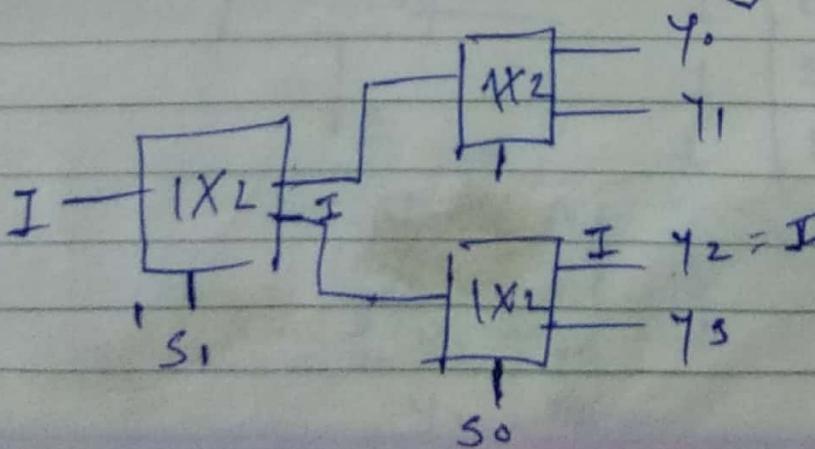
$$Y_1 = \bar{S}_1 S_0 I + \cancel{\bar{S}_1 \bar{S}_0 \cdot 0} + \cancel{S_1 \bar{S}_0 \cdot 0} + \cancel{S_1 S_0 \cdot 0}$$

$$Y_2 = S_1 \bar{S}_0 I + \cancel{\bar{S}_1 \bar{S}_0 \cdot 0} + \cancel{\bar{S}_1 S_0 \cdot 0} + \cancel{S_1 S_0 \cdot 0}$$

$$Y_3 = S_1 S_0 I + \cancel{\bar{S}_1 \bar{S}_0 \cdot 0} + \cancel{\bar{S}_1 S_0 \cdot 0} + \cancel{S_1 S_0 \cdot 0}$$

Q

Implement 1x4 DEMUX using 1x2 DEMUX



S_1	S_0	Y_0	Y_1	Y_2	Y_3
0	0	Y0	Y1	Y2	Y3
0	1	Y0	Y1	Y2	Y3
1	0	Y0	Y1	Y2	Y3
1	1	Y0	Y1	Y2	Y3

$$\left| \begin{array}{cc|cccccc} S_1 & S_0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ \hline 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 \end{array} \right| \quad 1 \cdot 1 \cdot 1 = 1$$

~~ex~~

$$1 \times 4 \quad \frac{y_2 = 2 \rightarrow 2y_2 = 1 \rightarrow}{\longrightarrow} 1 \times 2 \quad [2+1=3]$$

$$1 \times 8 \quad \frac{y_2 = 4 \rightarrow 4y_2 = 2 \rightarrow 2y_2 = 1 \rightarrow}{\longrightarrow} 1 \times 2 \quad 4+2+1=7$$

$$1 \times 64 \quad \frac{8y_8 = 8 \rightarrow 8y_8 = 1 \rightarrow}{\longrightarrow} 1 \times 1 \quad 8+1=9$$

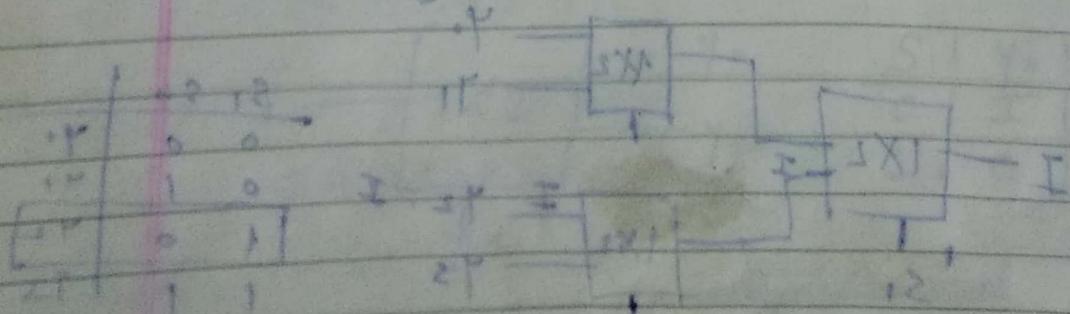
$$0.2 - 0.2, 0.2 + 0.2, 1.2 + 1.2, 1.2 = 0^P$$

$$0.2 + 0.2, 2 + 0.2, 2 + 1.2, 1.2 = 1^P$$

$$0.2, 2 + 0.2, 2 + 0.2, 2 + 1.2, 1.2 = 3^P$$

$$0.2, 2 + 0.2, 2 + 0.2, 2 + 1.2, 1.2 = 2^P$$

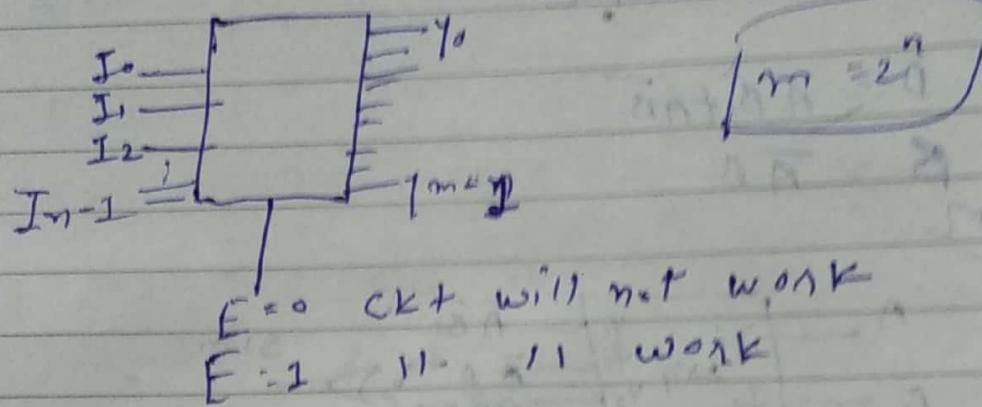
あると 571 で 100% が 1 である



Lecture 10

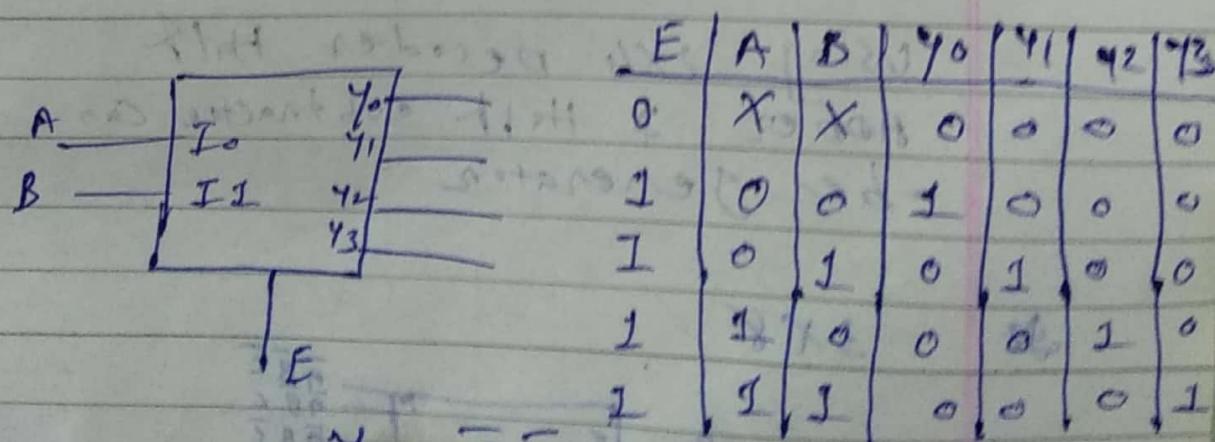
* Decoder :-

Combinational CKT which has many
inp & Many o/p



Decoder basically convert binary data to other code like Octal, Hexa etc.

2x4 Decoder



$$Y_0 = \overline{A}\overline{B}$$

$$Y_1 = \overline{A}B$$

$$Y_2 = A\overline{B}$$

$$Y_3 = AB$$

* Half adder

$$S = \bar{A}B + A\bar{B}$$

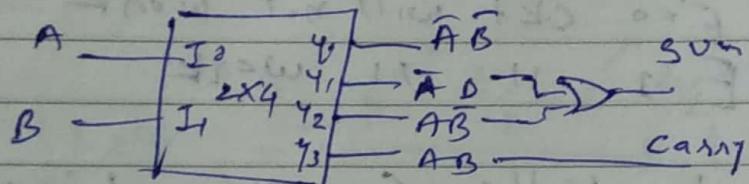
$$C = AB$$

- Half subtractor

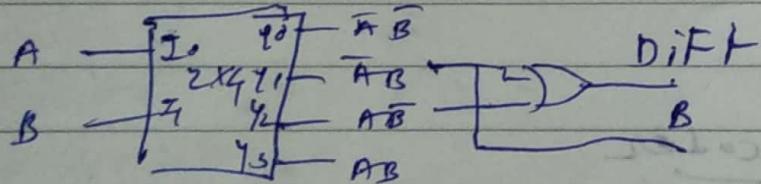
$$D = \bar{A}B + A\bar{B}$$

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

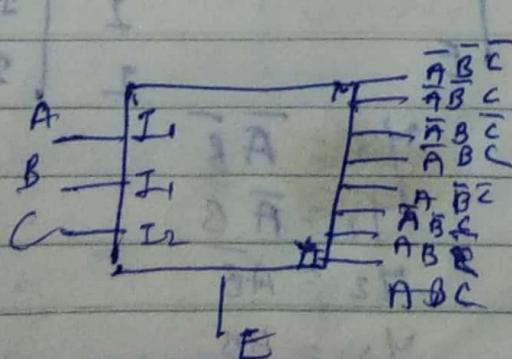
Half adder



Half subtractor



Using 2x4 decoder Half adder & Half subtractor can be generator



$$\Sigma m(1, 2, 4, 7)$$

Full Adder

$$S = A \oplus B \oplus C \\ = \bar{A} \bar{B} C + \bar{A} B \bar{C} + A \bar{B} \bar{C} + A B C$$

$$C = \Sigma m(3, 5, 6, 7)$$

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

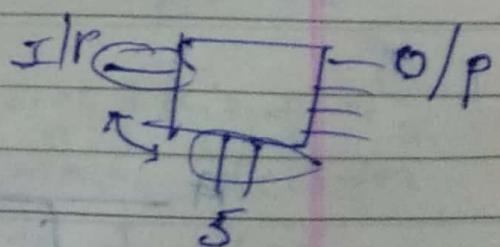
- Using 3x8 Decoder we can generate full adder & full subtractor

- basically 3x8 Decoder is binary to octal converter
- 4x16 Decoder is binary to hex converter

- Decoder and demultiplexer have same internal circuit

DEMUX

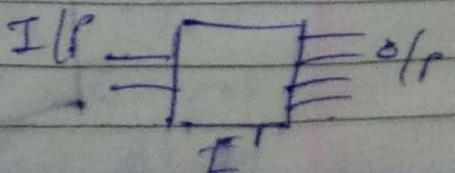
$$2 \times 4 = 1 \times 4$$



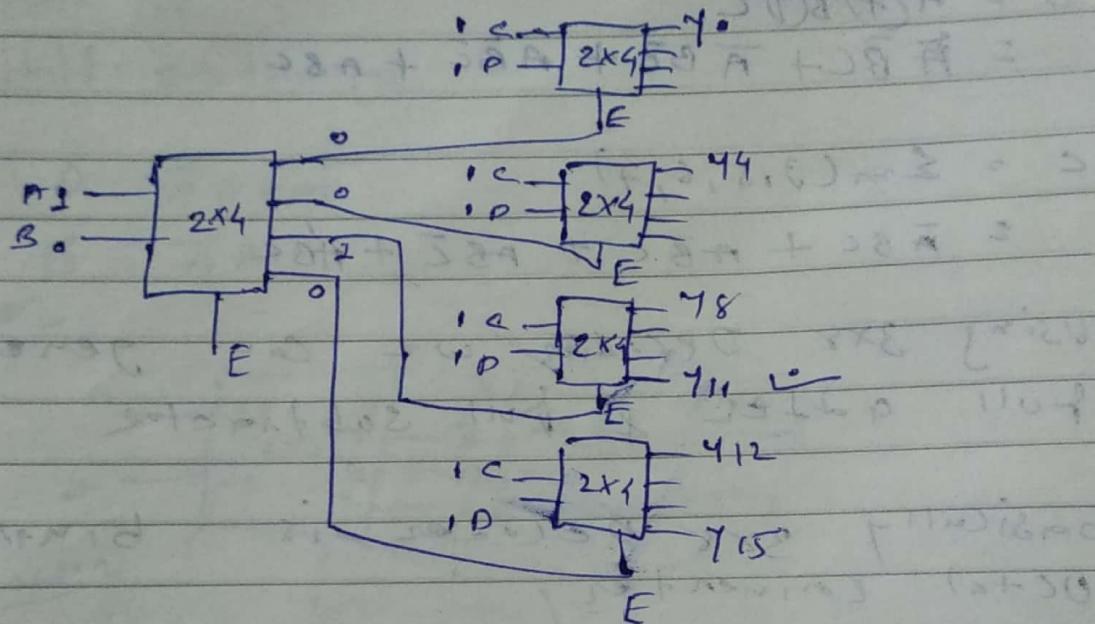
$$3 \times 8 = 1 \times 8$$

$$4 \times 16 = 1 \times 16$$

Decoder

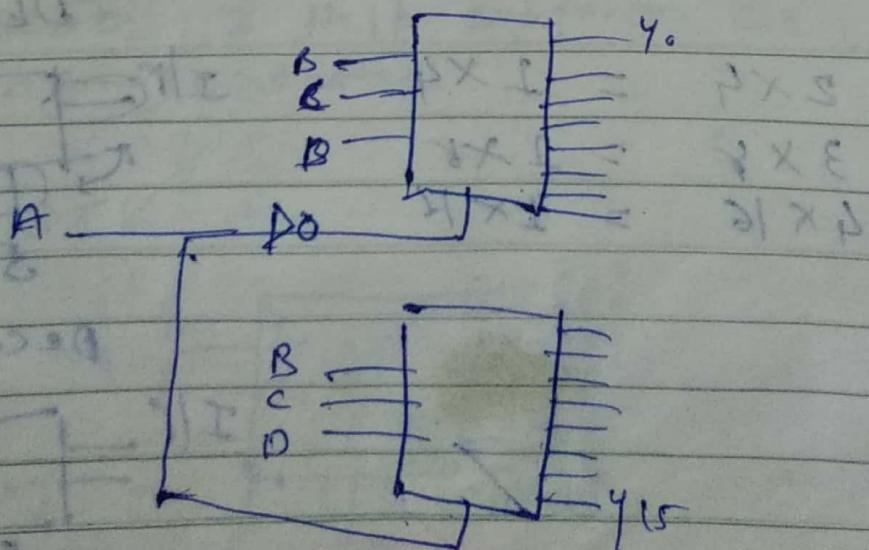


* Implement a 4×16 decoder HW
Using 2x4 decoder



$$\text{ex } \begin{array}{cccc} A & B & C & D \\ 1 & 0 & 1 & 1 \end{array} = Y_{11}$$

Q Implement 4×16 decoder using
 3×8 Decoder



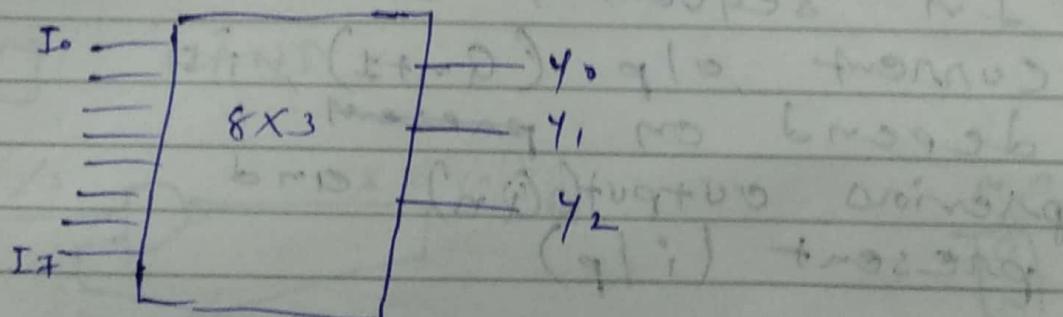
I₀ → 110 In priority Encoder give priority to
 I₇ → high higher number input

* Encoder:-

(95) (combinational) CKT which have Many I/P & Many O/P

- used to convert other code to binary

Base-4 }
 octal } → binary
 hexa }



If Input I₄ is 1 then output will be 100.

priority Encoder: It's special kind of CKT, now suppose normal decoder I₃ & I₆ both are set to 1 then Priority Encoder give priority to I₆ Mean output will be 110