

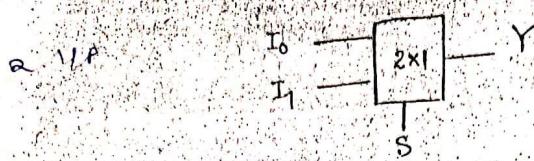
Multiplexer is a ⁿ combinational ckt which have many data input and one o/p. Depending on control or select one of the input is transfer to the o/p. It is also known as selector
 (or) Many to one ckt (or) Universal logic cks. (or)

(or) Parallel to serial convertor

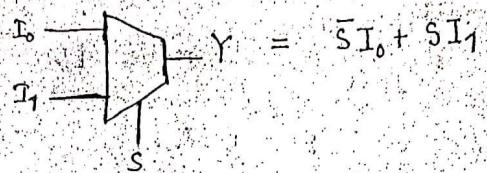
$$\rightarrow m = 2^n \quad \begin{matrix} m \dots \text{data input} \\ n \dots \text{select input} \end{matrix}$$

$$n = \log_2 m$$

$\rightarrow 2 \times 1 \text{ MUX}$



\rightarrow Symbol for MUX

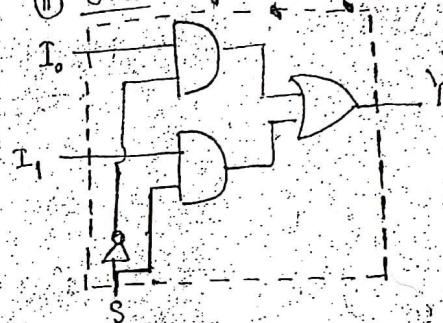


① Truth table

S	Y
0	I_0
1	I_1

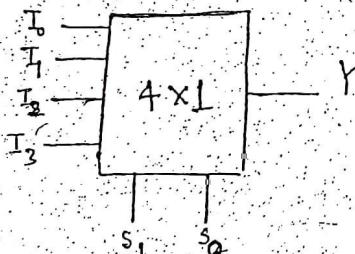
$$Y = \bar{S}I_0 + SI_1$$

② CKT



$\leftrightarrow 4 \times 1 \text{ MUX}$

①

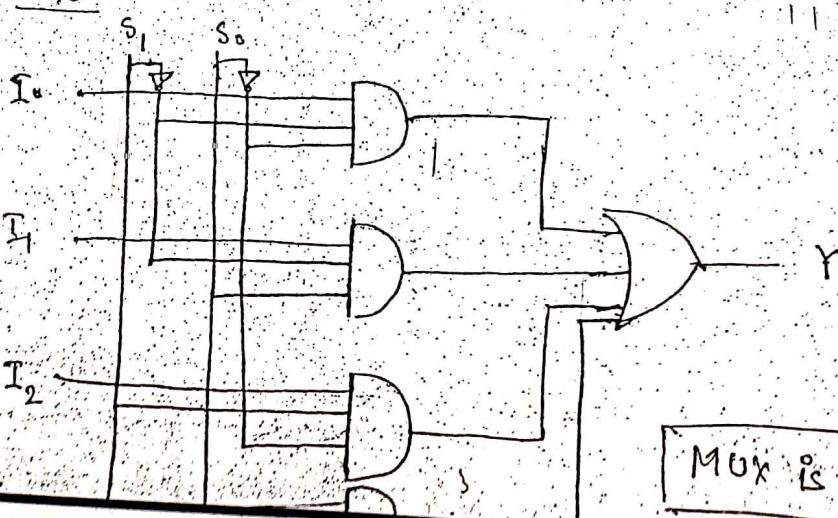


② Truth table

s_1	s_0	Y
0	0	I_0
0	1	I_1
1	0	I_2
1	1	I_3

$$Y = \bar{s}_1\bar{s}_0 I_0 + \bar{s}_1s_0 I_1 + s_1\bar{s}_0 I_2 + s_1s_0 I_3$$

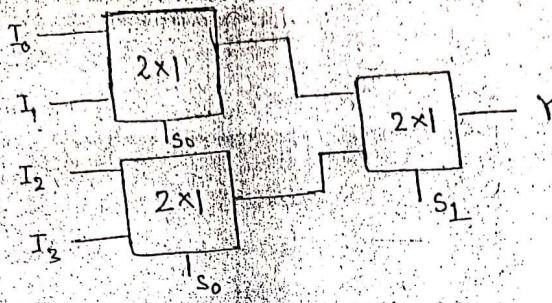
③ CKT



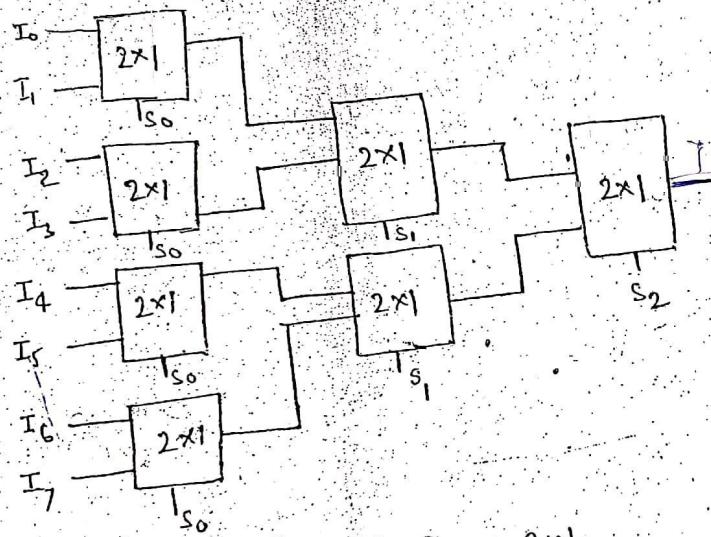
MUX is AND Gate followed by OR

43. Implementation of Higher order MUX using Lower

① 4x1 MUX $\xrightarrow{\text{using}}$ 2x1



② Bx1 MUX using 2x1



$$\begin{array}{l} \text{I}^{\text{st}} \text{ stage} \\ \begin{array}{ccccccc} & & & & & & \\ \begin{matrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{matrix} & \xrightarrow{\frac{8}{2}, \frac{4}{2}, \frac{2}{2}} & \begin{matrix} 1 & 2 & 1 & 2 & 1 & 2 & 1 \end{matrix} & \xrightarrow{\frac{4}{2}, \frac{2}{2}} & \begin{matrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{matrix} \\ \text{by 2} & & \text{by 2} & & \text{by 2} & & \end{array} \end{array}$$

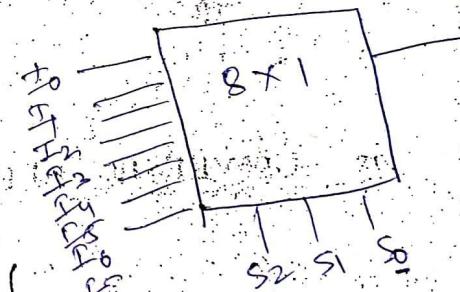
③ 16x1 MUX using 2x1

$$16 \times 1 \xrightarrow{15} 2 \times 1$$

$$64 \times 1 \xrightarrow{63} 2 \times 1$$

$$256 \times 1 \xrightarrow{255} 2 \times 1$$

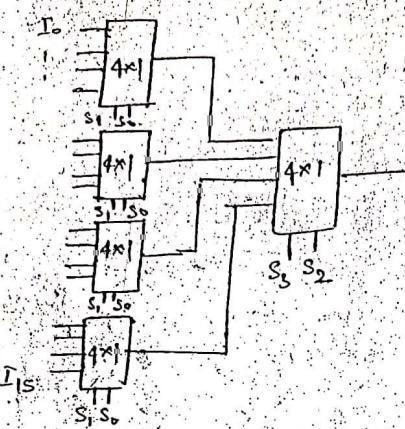
$$2^n \times 1 \xrightarrow{2^n - 1} 2 \times 1$$



ii) 16×4 using 4×1

$$4 + 1 = 5$$

$$\frac{16}{4} = 4 \quad \frac{4}{4} = 1$$



S₃ S₂ S₁ S₀ ... Select input

D) 64×1 using 4×1

$$16 + 4 + 1 = 21$$

D) 64×1 using 8×1

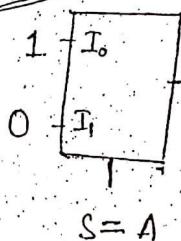
$$8 + 1 = 9$$

) 256×1 using 16×1

$$16 + 1 = 17$$

MUX as UNIVERSAL ckt.

NOT



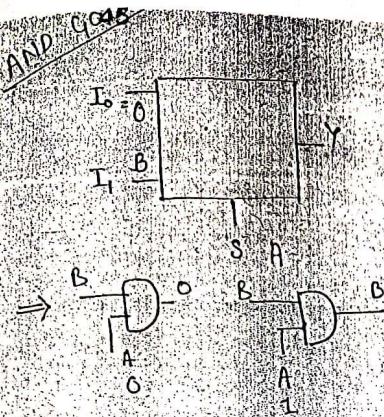
$$Y = \bar{S}_1 I_0 + S_1 I_1$$

$$= \bar{A} \cdot 1 + A \cdot 0$$

$$Y = \bar{A}$$

also.. $\begin{array}{c} A \quad Y \\ \hline 0 & 1 \\ 1 & 0 \end{array}$

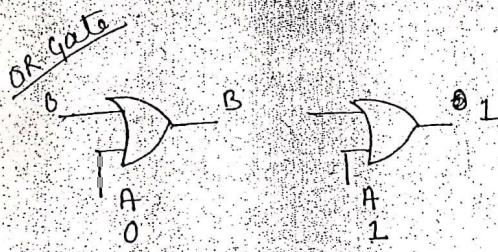
→ In Not Gate, one 2×1 MUX is required



$$Y = \overline{S} I_o + S I_i$$

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

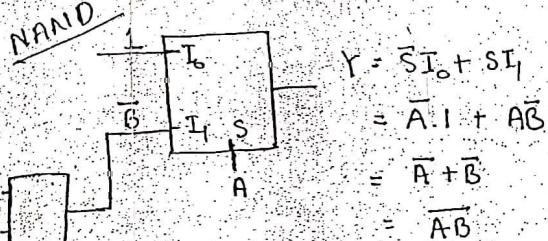
$$Y = AB$$



$$Y = \overline{S} I_o + S I_i$$

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

$$= A + B$$

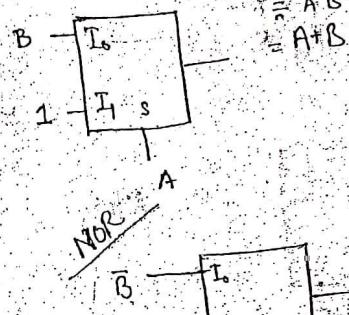


$$Y = \overline{S} I_o + S I_i$$

$$= A \cdot 1 + \bar{A} \cdot \bar{B}$$

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

$$= \overline{AB}$$

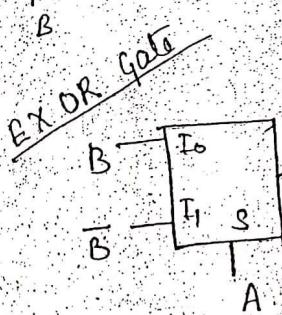


$$Y = \overline{S} I_o + S I_i$$

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

$$= \cancel{AB}$$

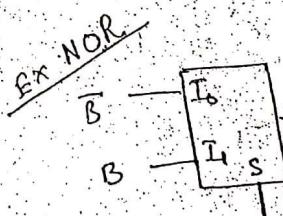
$$= \overline{A+B}$$



$$Y = \overline{S} I_o + S I_i$$

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

$$= A \oplus B$$



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

$$= A \odot B$$

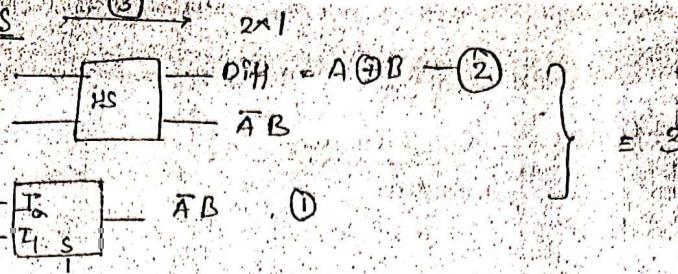
To implement ex-OR and AND gate minimum number of 2x1 MUX require respectively.

- ④ 2,1 ⑥ 1,2,1 ⑦ 2,1 ⑧ 2,2

Doubt
HA $\xrightarrow{3}$ 2x1

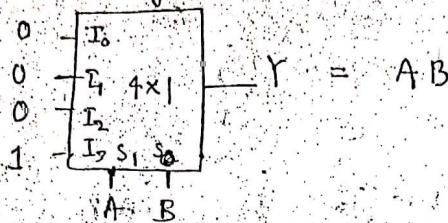
Dues

49S

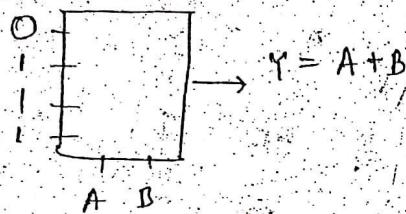


= 3

→ AND Gate using 4x1



→ OR Gate



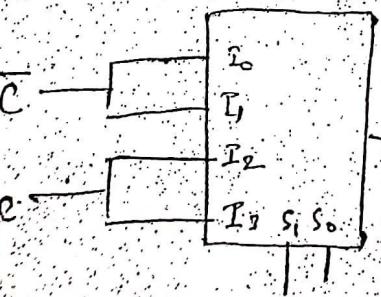
Dues

Logical Exp. for f?



- (a) xy
- (b) $x+y$
- (c) $\frac{xy}{x+y}$
- (d) $\frac{x+y}{xy}$

I - Determine minimized logical expression



- (a) $A \oplus B$
- (b) $A \ominus B$
- (c) $A \ominus C$
- (d) $A \oplus C$

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

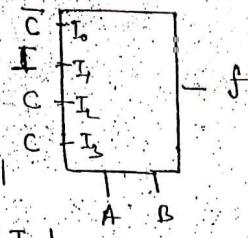
$$= \bar{A} \bar{B} \bar{C} + \bar{A} B \bar{C} + A \bar{B} C + A B C$$

47

$$\begin{aligned} &= \bar{A}C + AC \\ &= A \oplus C \end{aligned}$$

IV Implementation of given logical exp.

Dues Implement $f(A,B,C) = \sum m(0,2,3,5,7)$ using 4×1 MUX shown in figure.



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

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

Sol²

	I ₀	I ₁	I ₂	I ₃
C	0	2	4	6
C	1	3	5	7
C	1	0	1	0

A	B	C	
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

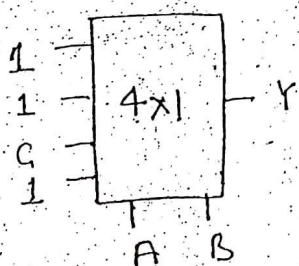
→ 1 4×1 MUX and 1 NOT gate require.

Dues 2 $F(A,B,C) = \sum m(0,1,2,3,5,6,7)$, using \bar{A}, B as select

- (i) AC as select
- (ii) BC as select

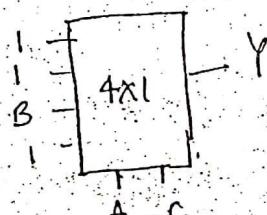
Sol²

(i)



	I ₀	I ₁	I ₂	I ₃
C	0	2	4	6
C	1	3	5	7
C	1	1	0	1

(ii)



	I ₀	I ₁	I ₂	I ₃
B	0	1	4	5
B	2	3	6	7

(iii)



	I ₀	I ₁	I ₂	I ₃
A	0	1	2	3
A	4	5	6	7

→ Using one 4x1 MUX using two multiplexers exp.

some of 3 variables

→ using one 4x1 Mux and one NOT gate All three variables

initially 3 variable

→ using one 8x1 all 8 variable

some of 4 variables

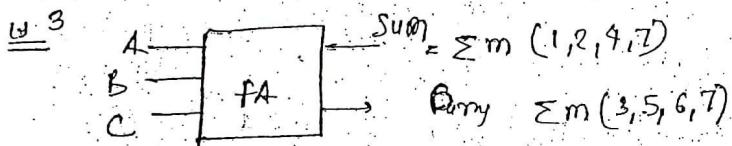
Q. 1) $\Sigma m(0,1,3,7)$

1) first convert $\Sigma m(2,4,5,6)$

Ans 2) $A + \overline{B}C$

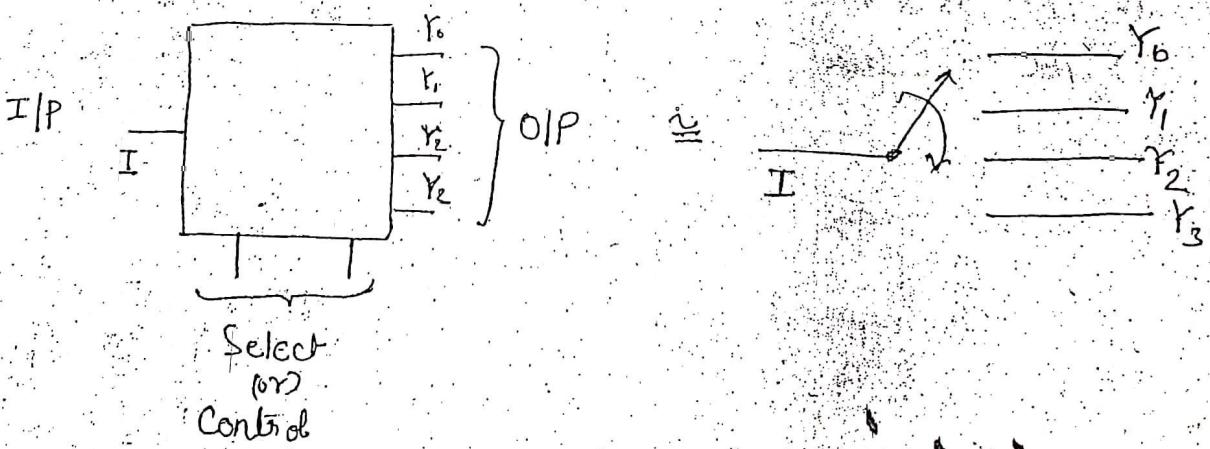
→ Change into canonical form

Ans: A (



Multiplexers

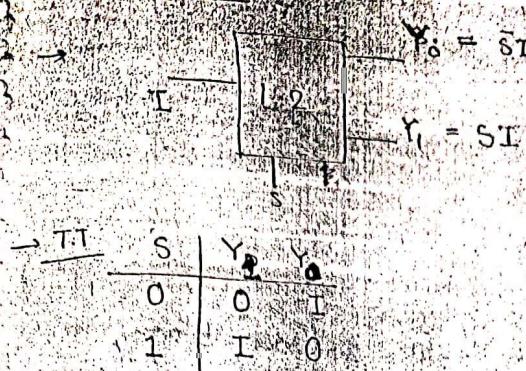
1. It have one input and many output



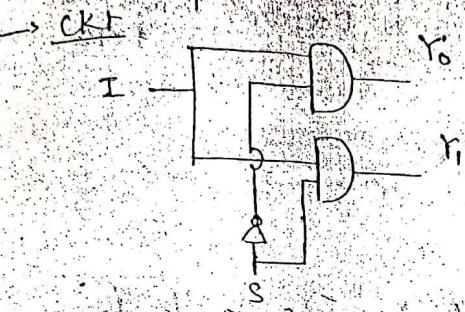
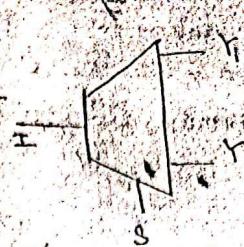
2) Mux is a combinational ckt which have 1 data input and many output. Depend on select input, input is transferred to one of the output lines. Hence it is known as One to many CKT.

(iii) Data Distributors

1:2 MUX



Symbol



⇒ De MUX have only AND gate
not OR gate

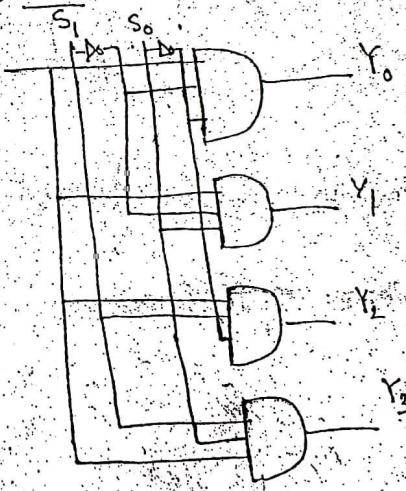
Higher order MUX

1:4

(i) T.T.

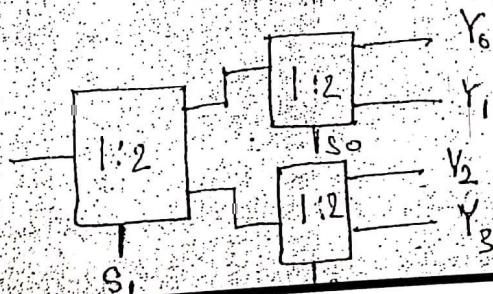
I	Y ₀	Y ₁	Y ₂	Y ₃
0	$\bar{S}_1 \bar{S}_0 I$	$\bar{S}_1 S_0 I$	$S_1 \bar{S}_0 I$	$S_1 S_0 I$

(ii) CKF



Implement 1:4 Demux

→ 1:2 require



Answer remain
same for
MUX and Demux

FS using $HSS = 2^{H>}$, OR

- 1:8 Demux $\xrightarrow{5}$ 1x2 require
- 1:16 Demux $\xrightarrow{5}$ 1x4 require
- 1:64 Demux $\xrightarrow{9}$ 1x8 require
- 1:256 Demux $\xrightarrow{17}$ 1x16 require

Decoder

Decoder is a combinational CKT which will convert binary to other codes such as Binary to Octal, BCD to Decimal, Binary to Hexa decimal

It has many inputs and many outputs.

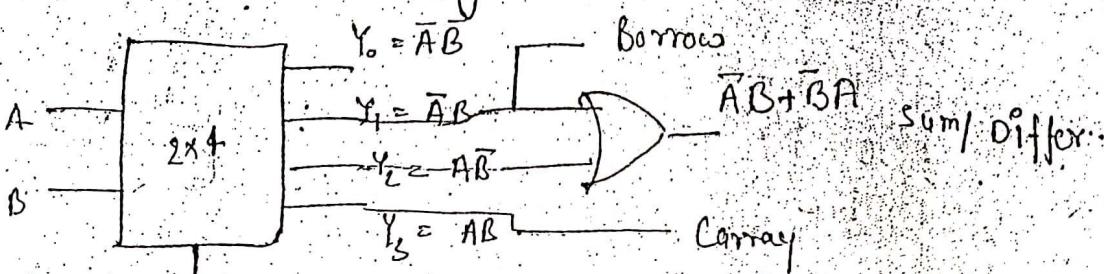
Min^m possible decoder is 2x4.

2x4 decoder

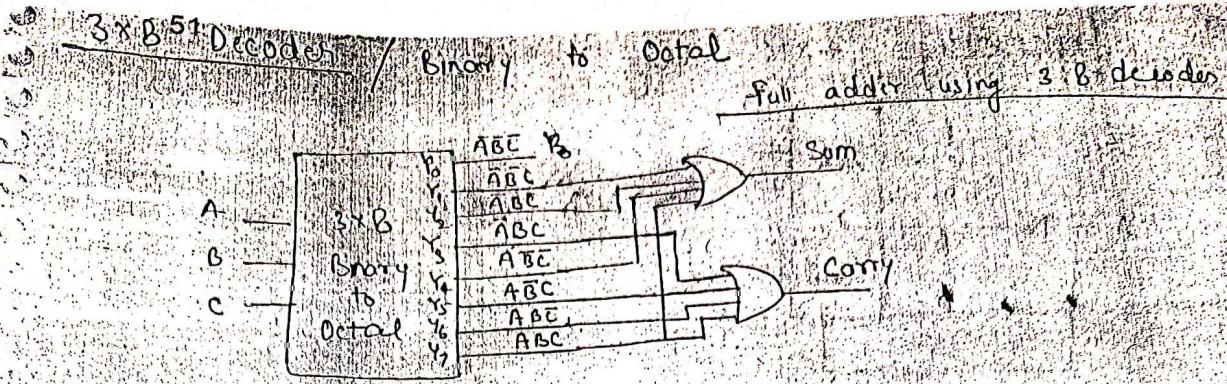
A	B	E	$Y_0 = \bar{A}\bar{B}E$	$Y_1 = \bar{A}BE$	$Y_2 = A\bar{B}E$	$Y_3 = ABE$	A	B	Y_3	Y_2	Y_1	Y_0
0	0	X	X	X	0	0	0	0	0	0	0	0
0	1	0	0	0	0	0	1	0	0	0	0	1
1	0	0	0	1	0	0	0	1	0	1	0	0
1	1	1	0	0	1	0	1	1	0	1	0	0

→ Decode and De Mux have same internal circuit

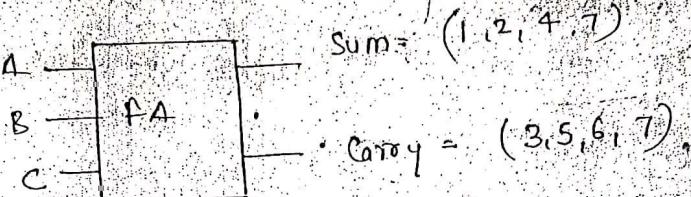
Ques Implement half adder using 2:4 Decoder



1 2:4 decoder and 1 OR gate require for HA & HS



full Adder



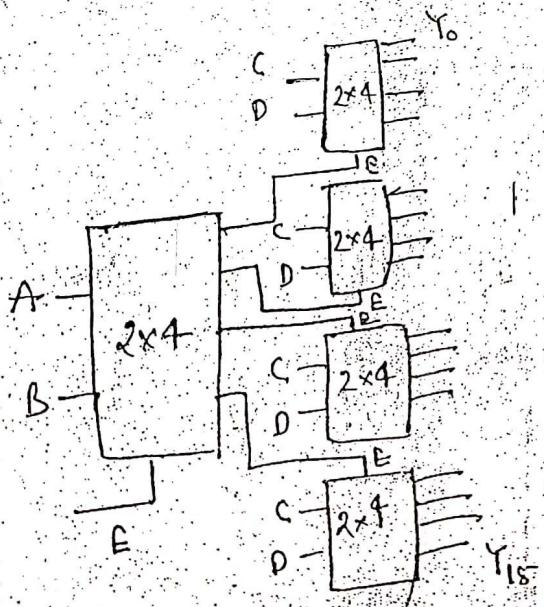
Canonical form require for Decoder, Demux

II - How to implement high order Decoder using smaller

Ques Implement 4×16 $\xleftarrow[1 \times 16, \text{ demux}]{\cong}$ 2×4 $\xrightarrow[5]{} 1 \times 4, \text{ Demux}$

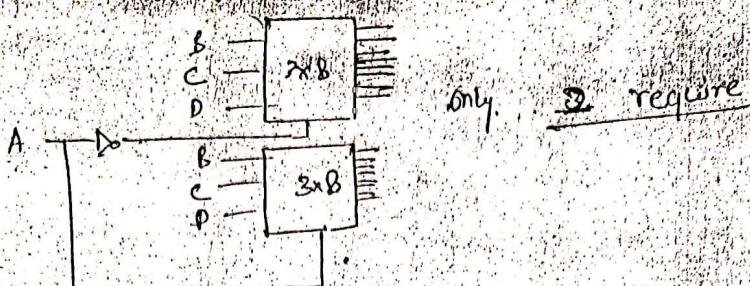
$$\frac{GF}{A} = 16 \\ 16 + 4 + 1$$

$6 \times 64 \xrightarrow[16 + 4 + 1]{21} 2 \times 4 \xleftarrow[21]{} 1 \times 4 \text{ demux}$



III

52. Implementing 4×16 using 3×8



only 2 require

Encoder

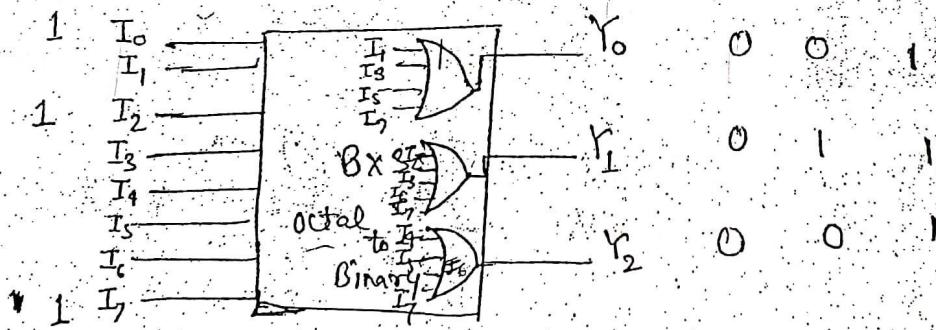
Encoder is used to convert other codes to binary

Octal - Binary $\rightarrow 8 \times 3$

Decimal - BCD $\rightarrow 10 \times 4$

Hex - Binary $\rightarrow 16 \times 4$

Octal to Binary Encoder



In encoder one of the input line is high and the corresponding Binary is available at the output.

	I ₇	I ₆	I ₅	I ₄	I ₃	I ₂	I ₁	I ₀
0	0	0	0	0	0	0	0	1
1	0	0	0	0	0	0	1	0
2	0	0	0	0	0	1	0	0
3	0	0	0	0	1	0	0	0
4	0	0	0	1	0	0	0	0
5	0	0	1	0	0	0	0	0
6	0	1	0	0	0	0	0	0

	Y ₂	Y ₁	Y ₀
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

58

$$Y_0 = I_1 + I_3 + I_5 + I_7$$

~~$$Y_1 = I_2 + I_3 + I_6 + I_7$$~~

$$Y_2 = I_4 + I_5 + I_6 + I_7$$

Encoder using OR Gates only