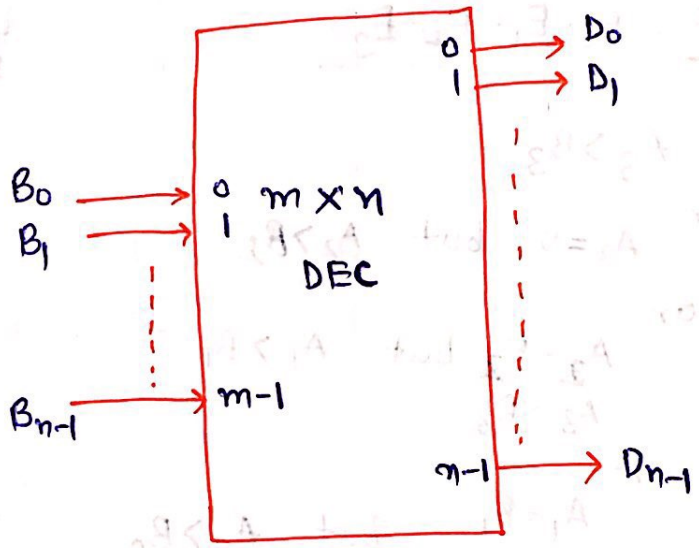
Decoder :-

It is a combinational ckt which convert the code one form to another form, having m input lines and n output lines.

$$(2^m \geq n)$$

- 2 × 4
- 3 × 8
- 4 × 10
- 4 × 16



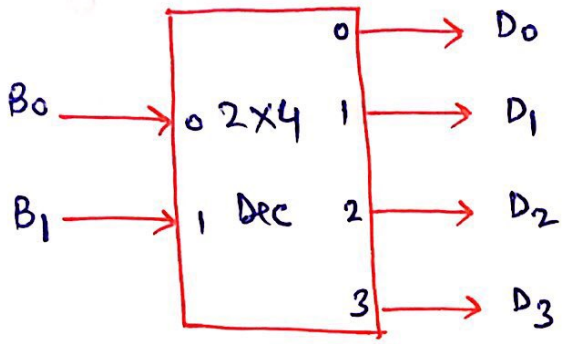
2 : 4 or 2 × 4 ⇒ Binary to Nibble Dec

3 : 8 or 3 × 8 ⇒ Binary to Octal Dec

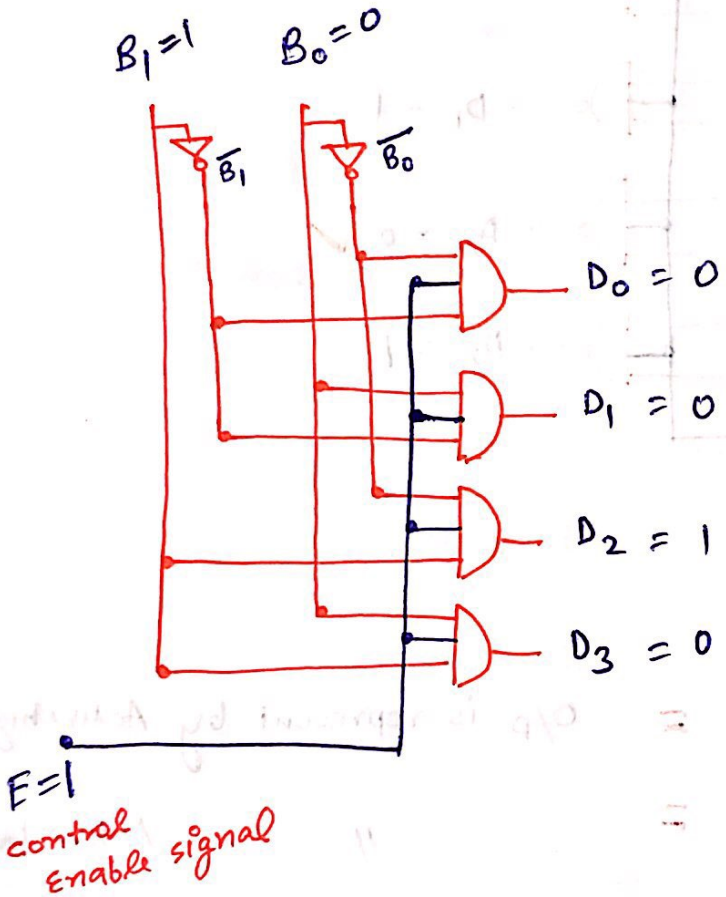
4 : 10 or 4 × 10 ⇒ BCD to decimal Dec

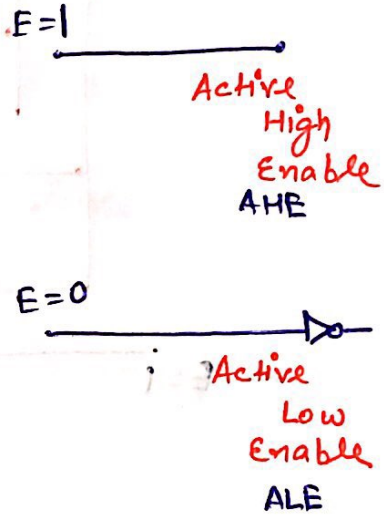
4 : 16 or 4 × 16 ⇒ Binary to hexadecimal Dec

2X4 Decoder :-



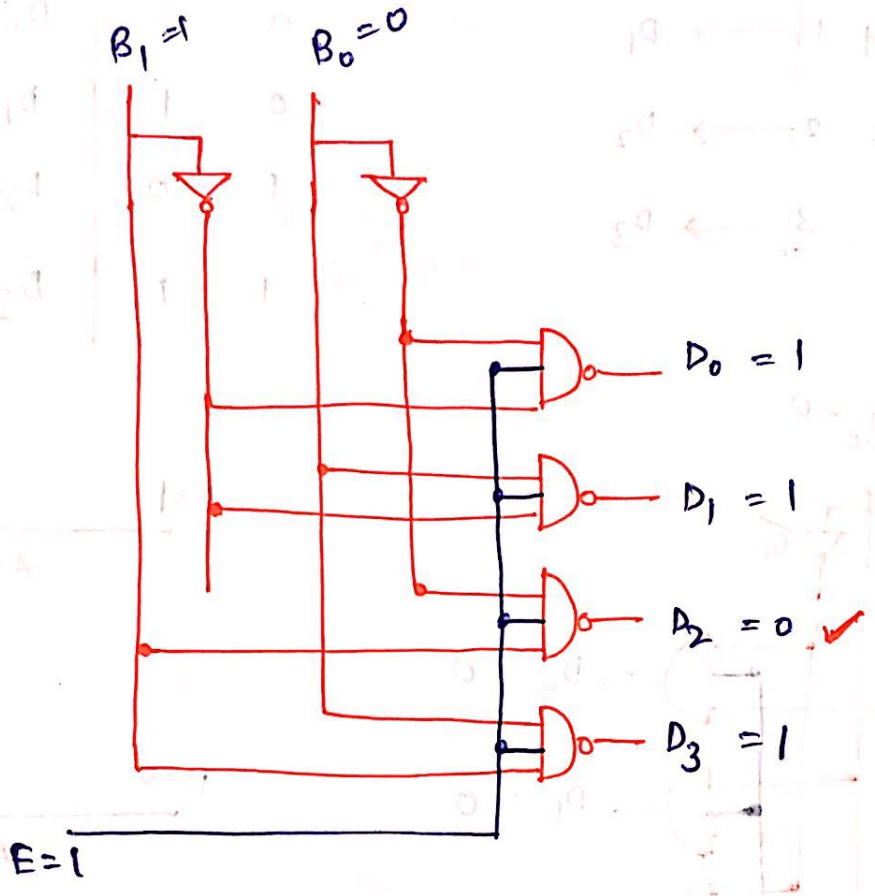
B_1	B_0	O/P
0	0	$D_0 = \overline{B_1}\,\overline{B_0}$
0	1	$D_1 = \overline{B_1} B_0$
1	0	$D_2 = B_1 \overline{B_0}$
1	1	$D_3 = B_1 B_0$





	AHE	ALE
D_0	$\overline{B_1}\,\overline{B_0}E$	$\overline{B_1}\,\overline{B_0}\,\overline{E}$
D_1	$\overline{B_1}B_0E$	$\overline{B_1}B_0\overline{E}$
D_2	$B_1\overline{B_0}E$	$B_1\overline{B_0}\,\overline{E}$
D_3	B_1B_0E	$B_1B_0\overline{E}$

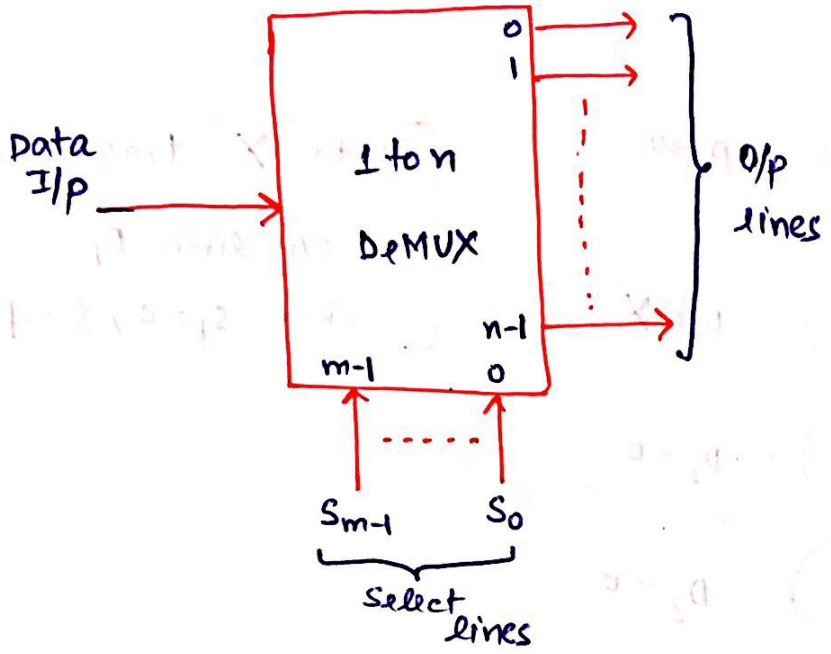
NOTE:- In the decoder circuit if we replace AND with NAND gates, circuit will work at decoder only.



{ Decoder with AND = O/p is represent by Active high
// NAND = // Active Low }

De-Multiplexer :-

It is combinational circuit having reverse opⁿ of MUX.

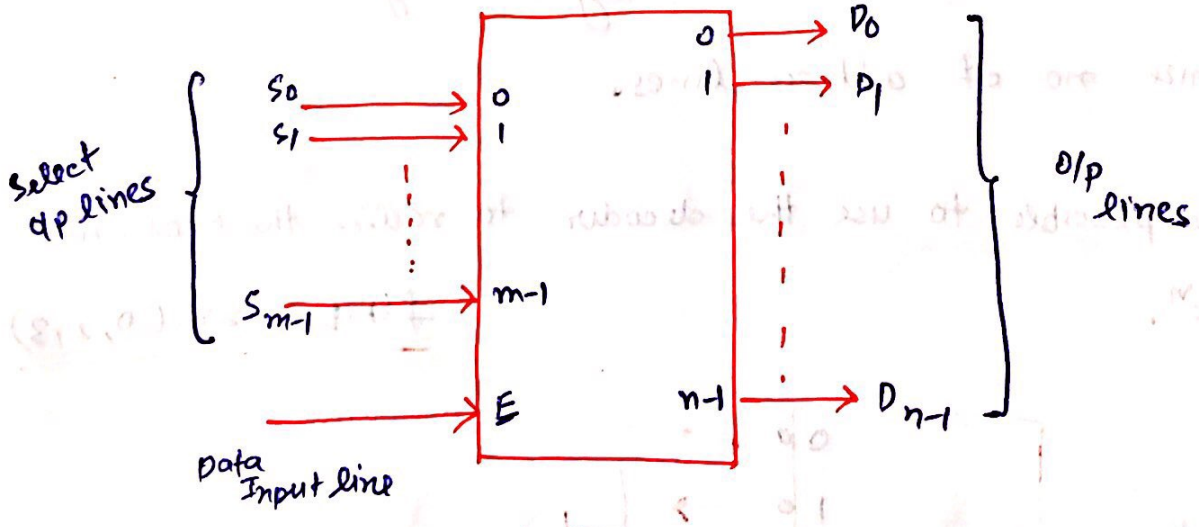


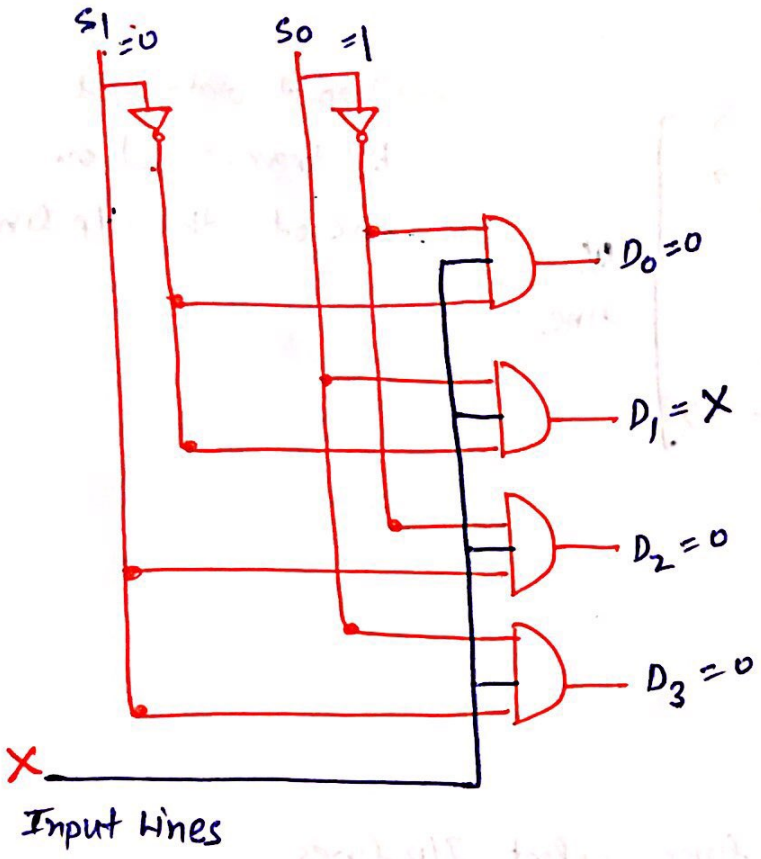
Input data will be transfered on one of the O/p line

NOTE:- IMUX → Select lines select I/p lines

DeMUX → " " O/p lines

* Decoder can be used as DeMUX by renaming the terminals of Decoder.



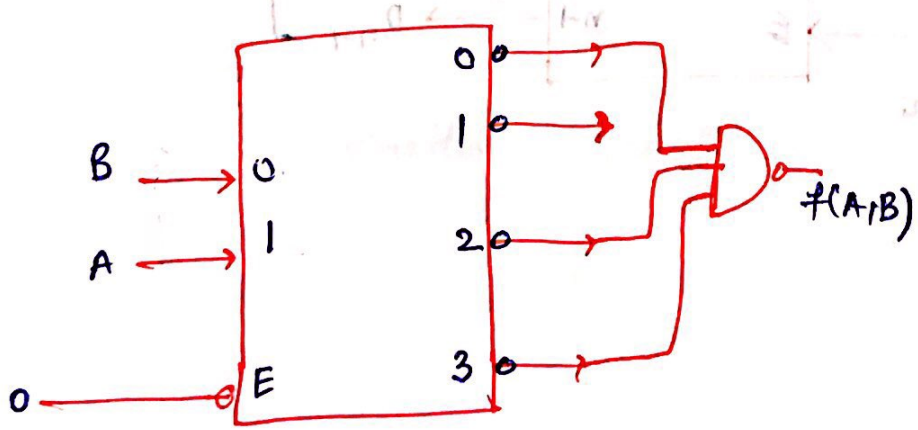


[Data 'X' transferred to o/p line D_1 when $S_1=0$, $S_0=1$]

Application of Decoder / De-MUX :-

- It is possible to use to convert serial data to parallel form.
- Decoders are used in memory mapping circuits to minimize no. of address lines.
- It is possible to use the decoder to realize the boolean func^n.

$f(A,B) = \Sigma m(0,2,3)$



- It is possible to realize multi-o/p functions or ckts.

Full Adder
- 3×8 Dec (1) + OR (2) / NAND (2)
  - Active High
  - Active Low
- 8×1 MUX (2)
- 4×1 MUX (2) + NOT (1)

Q: Realize 16:1 MUX using :

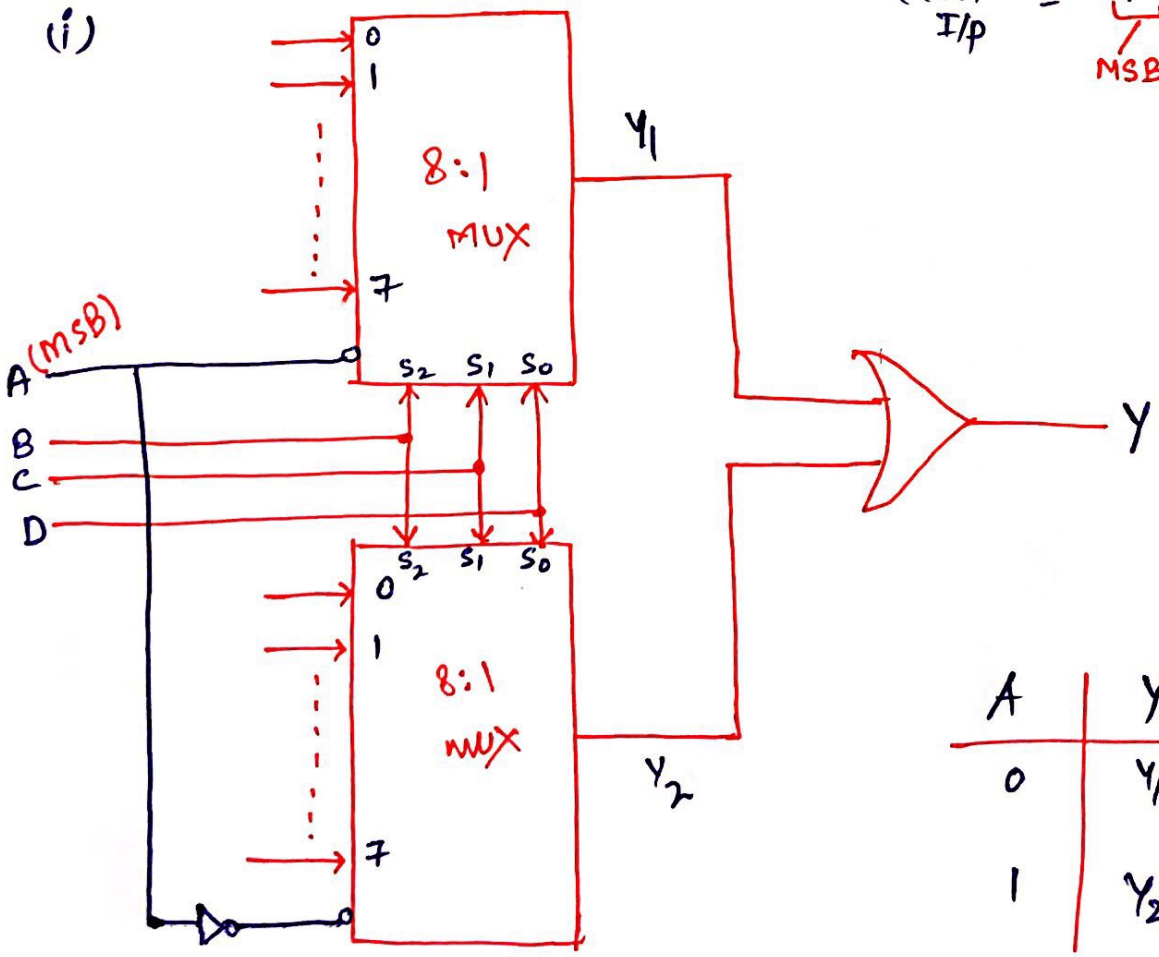
(i) 8×1 MUX $n_1/n_2 = 2$

(ii) 4×1 MUX $n_1/n_2 = 4$

(n_1, n_2)

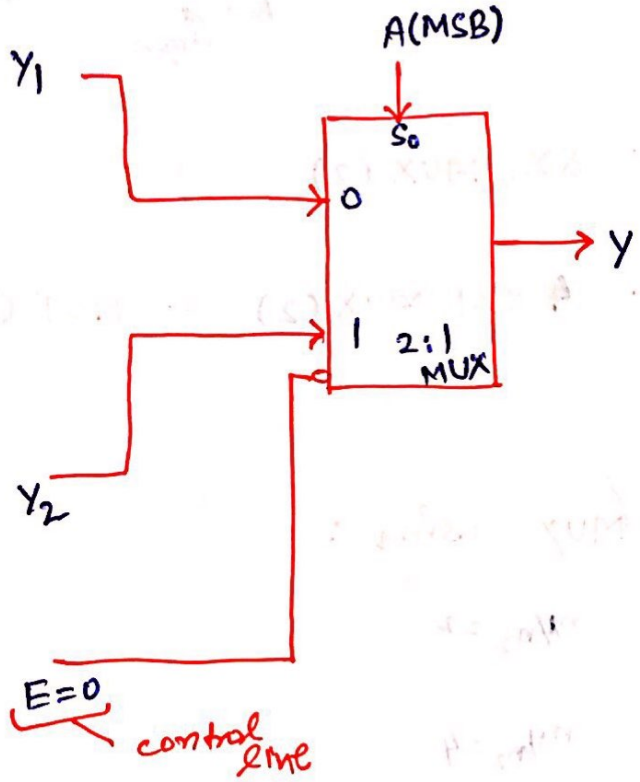
(i)

Select I/p = A B C D (A: MSB)

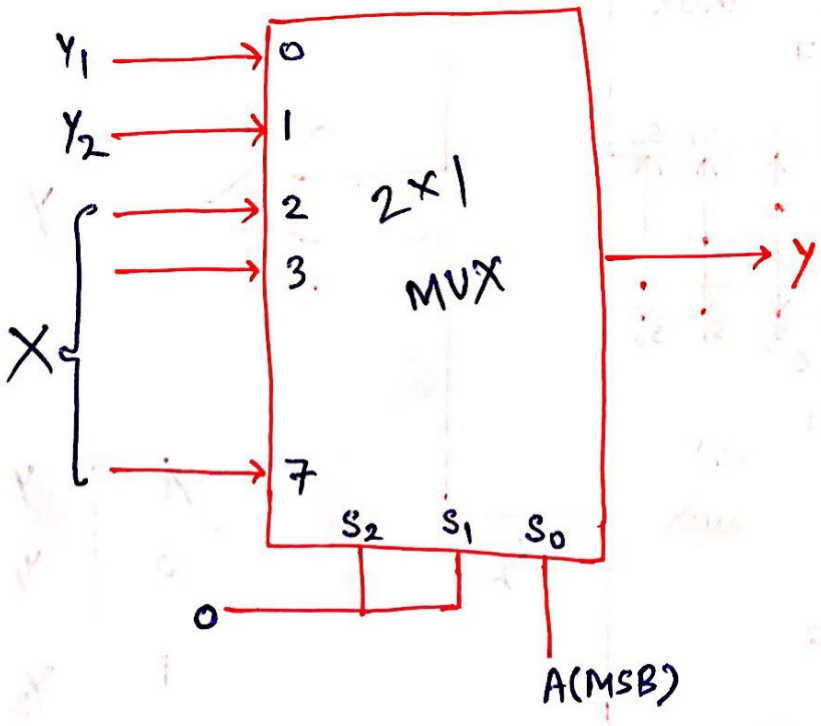


A	Y	
0	Y_1	MUX_1 ✓, MUX_2 ✗
1	Y_2	MUX_1 ✗, MUX_2 ✓

Since the control lines is used by A(MSB) bit so there is no control lines, so replace OR gate with 2:1 MUX.



Convert 8:1 MUX to 2:1 MUX :-



Q: Implementation of 16:1 MUX is possible using:

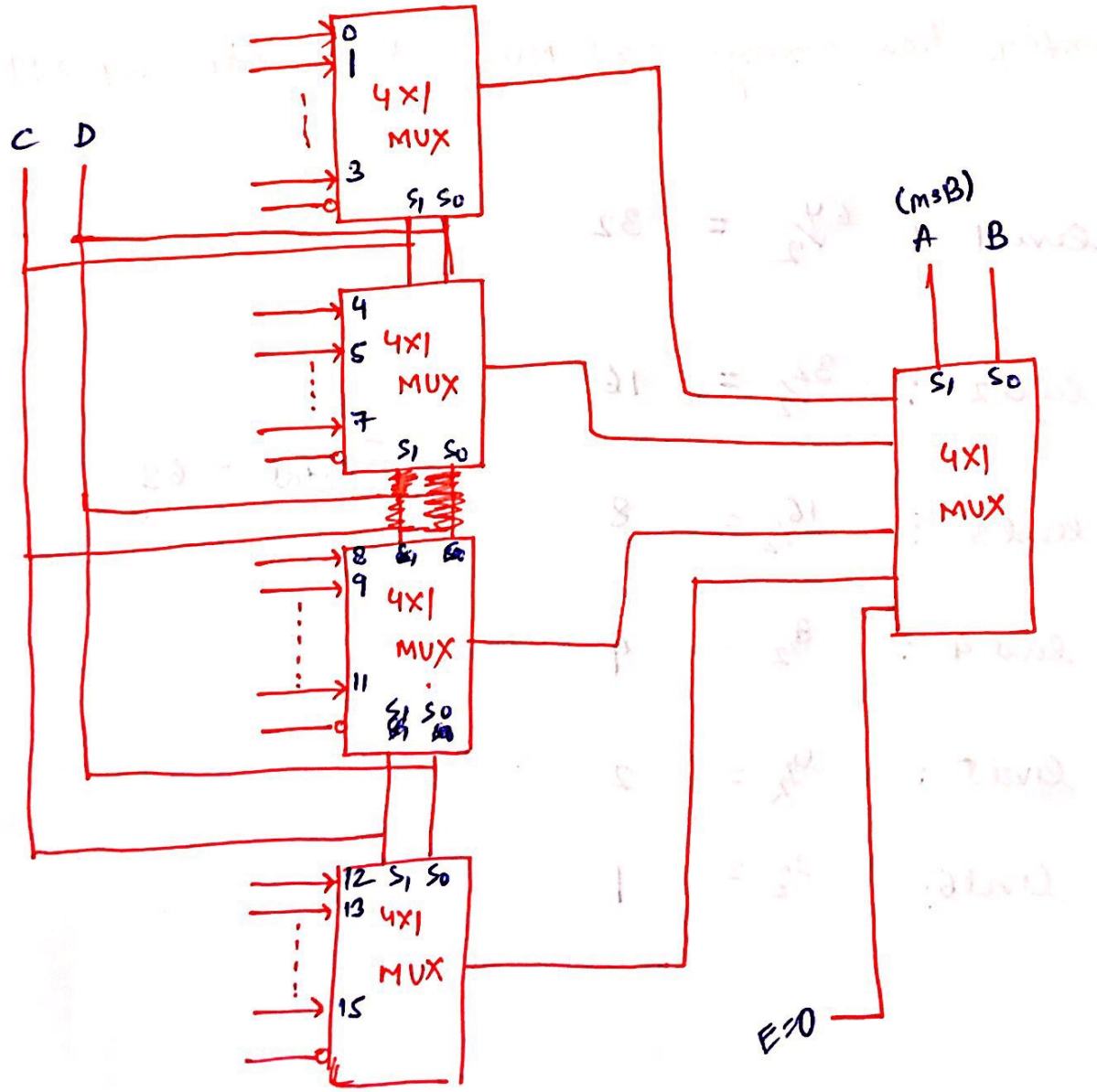
(a) 2(8:1 MUX) ₹, 1 OR and 1 NOT

(b) 2(8:1 MUX) and 1 (2:1 MUX)

(c) 3(8:1 MUX)

(d) ✓ All of these

(ii) 16:1 MUX using 4:1 MUX :-



Q. How many 4×1 MUX req. to realize $\underbrace{256}_{2^8} \times 1$ MUX?

level 1 : $\frac{256}{4} = 64$

level 2 : $\frac{64}{4} = 16$

level 3 : $\frac{16}{4} = 4$

level 4 : $\frac{4}{4} = 1$

Total = 85

Q. Identify how many 2×1 MUX to realize 64 X1 MUX?

level 1 : $64/2 = 32$

level 2 : $32/2 = 16$

level 3 : $16/2 = 8$

level 4 : $8/2 = 4$

level 5 : $4/2 = 2$

level 6 : $2/2 = 1$

Total = 63

Q. Realize 4 to 16 Decoder using :

(i) 3×8 Dec (3 Dec req)

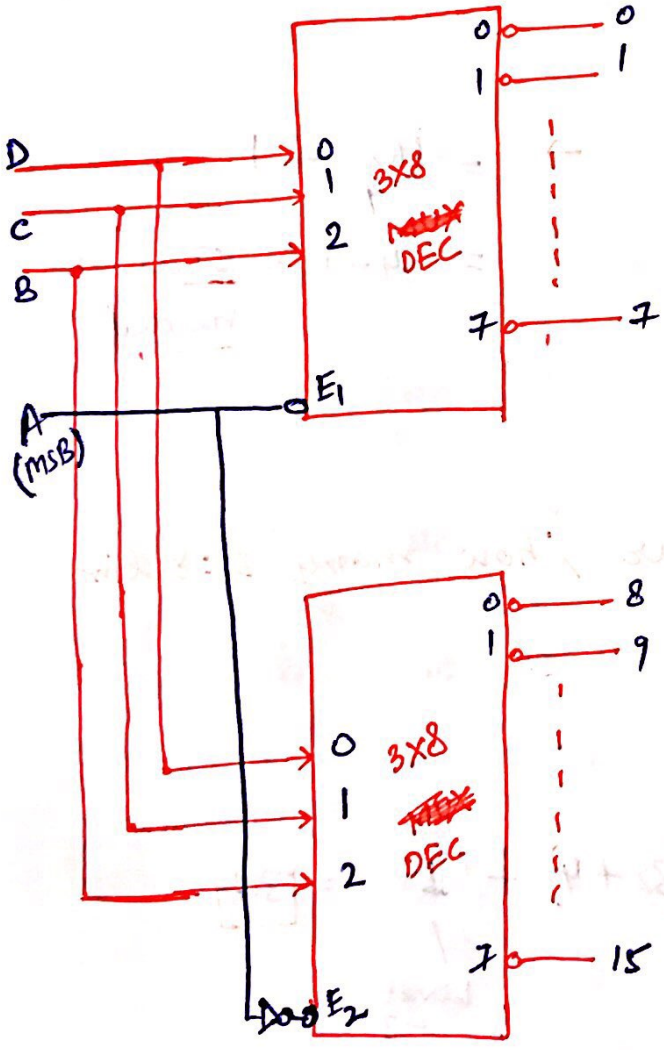
(ii) 2×4 Dec (5 Dec req)

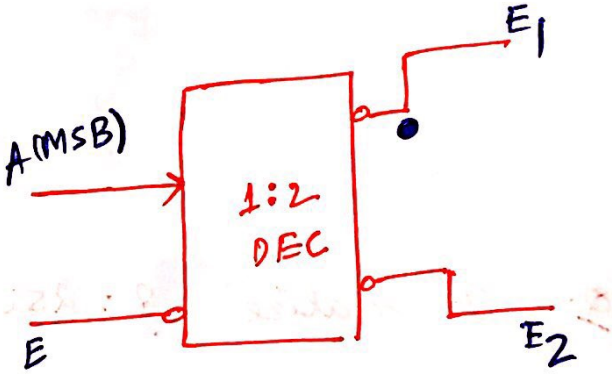
if required use minimum no. of logic gates.

(i)

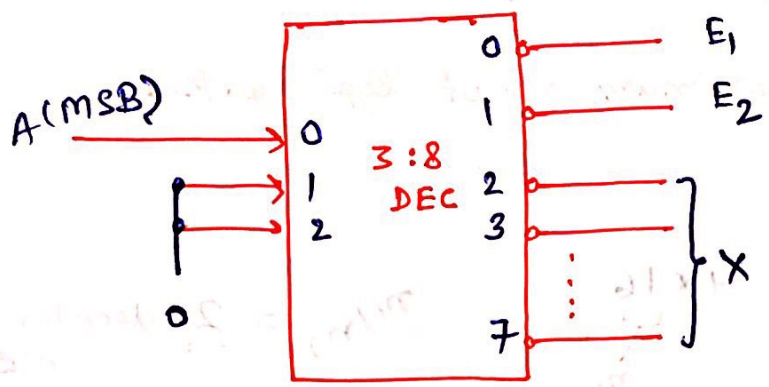
$3\times8 \longrightarrow 4\times16$

$n_1/n_2 = 2$ decoder needed + 1





→ Convert 3:8 Dec to 1:2 Dec :-



1:2 DECODER

(ii) 2:4 ⟶ 4:16 → $= 16/4 + 1$

$= 4 + 1 = 5$ decoder needed

Q: To realize 8:256 Decoder, how many 3:8 line decoders required.

Level-1 $\frac{256}{8} = 32$

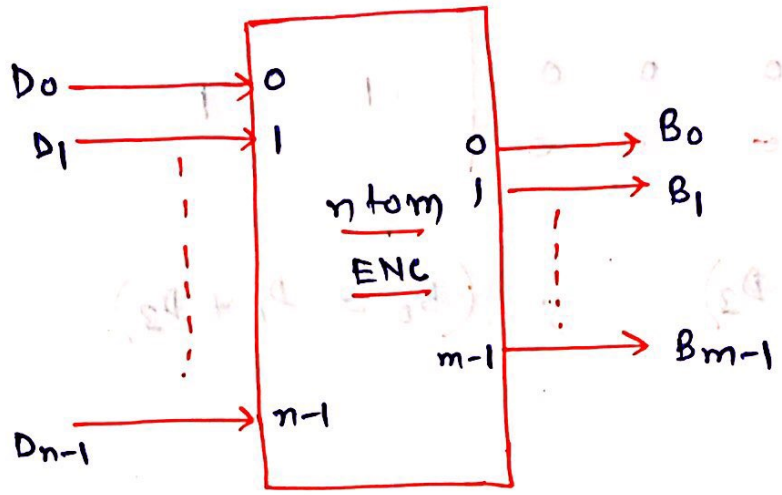
Level-2 $\frac{32}{8} = 4$

$(32+4) + 1 = 37$

Level-3 ...

Encoder :-

It is a combinational circuit, having reverse opn of the decoder.

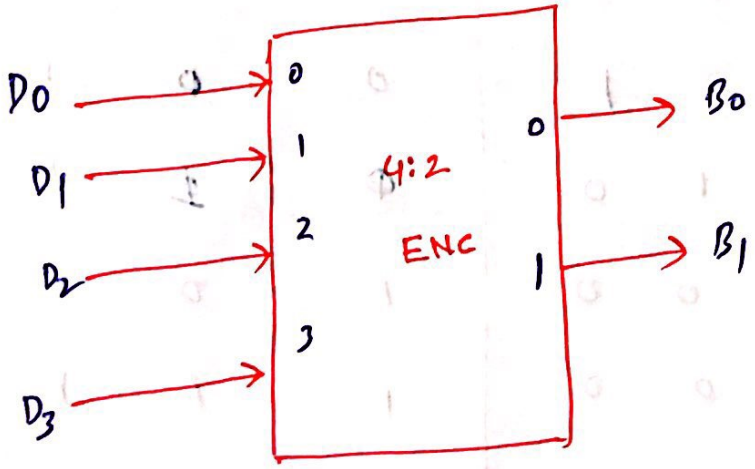


4 : 2 ENC ⇒ Nibble to binary

8 : 3 ENC ⇒ Octal to "

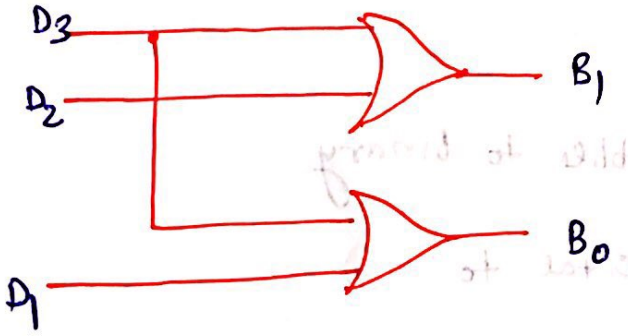
10 : 4 ENC ⇒ Decimal to BCD

16 : 4 ENC ⇒ Hexa Deci to Binary



D_3	D_2	D_1	D_0	B_1	B_0
0	0	0	1	0	0
0	0	1	0	0	1
0	1	0	0	1	0
1	0	0	0	1	1

$(B_1 = D_2 + D_3)$; $(B_0 = D_1 + D_3)$



Since ckt is independent of D_0, which will give same o/p to 0000 & 0001 Input.

So,

D_3	D_2	D_1	D_0	B_1	B_0	V (valid)
0	0	0	0	x	x	0
0	0	0	1	0	0	1
0	0	1	0	0	1	1
0	1	0	0	1	0	1
1	0	0	0	1	1	1

So, $V = D_3 + D_2 + D_1 + D_0$

Encoder with False data rejection facility

if $V = 0$ (Invalid)

$V = 1$ (Valid)

Application :-

- It is used as an interfacing circuit b/w the keyboard and system.

Priority Encoder :-

If more than one I/p line are enabled, then o/p has to be decided on the basis of priority of Input lines.

D_3	D_2	D_1	D_0	B_1	B_0	V
0	0	0	0	x	x	0
0	0	0	1	0	0	1
0	0	1	x	0	1	1
0	1	x	x	1	0	1
1	x	x	x	1	1	1

$B_1 = D_3 + \overline{D_3} D_2$; $B_0 = D_3 + \overline{D_3}\,\overline{D_2} D_1$

$= D_3 + D_2$ $= D_3 + \overline{D_2} D_1$

B_2 $V = D_3 + D_2 + D_1 + D_0$