Unit : II

Binary Codes

- Numeric Codes
  - number as a series of 0s & 1s
    - BCD (Binary Coded Decimal)
- alphanumeric Codes
  - letters of the alphabet & decimal numbers

Decimal digit	8 4 2 1	2 4 2 1	Ex-3
0	0 0 0 0	0 0 0 0	0 0 1 1
1	0 0 0 1	0 0 0 1	0 1 0 0
2	0 0 1 0	0 0 1 0	0 1 0 1
3	0 0 1 1	0 0 1 1	0 1 1 0
4	0 1 0 0	0 1 0 0	0 1 1 1
5	0 1 0 1	0 1 0 1	1 0 0 0
6	0 1 1 0	0 1 1 0	1 0 0 1
7	0 1 1 1	0 1 1 1	1 0 1 0
8	1 0 0 0	1 1 1 0	1 0 1 1
9	1 0 0 1	1 1 1 1	1 1 0 0

BCD Codes are Weighted & non-weighted

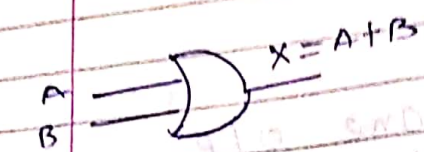
Weighted Codes

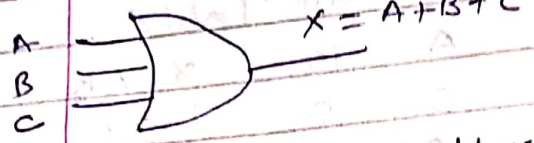
which obeys the position-weighting principle.

Excess-3 & Gray Codes are non-weighted Co

(both the i/ps can be present)

(*) OR GATE:-

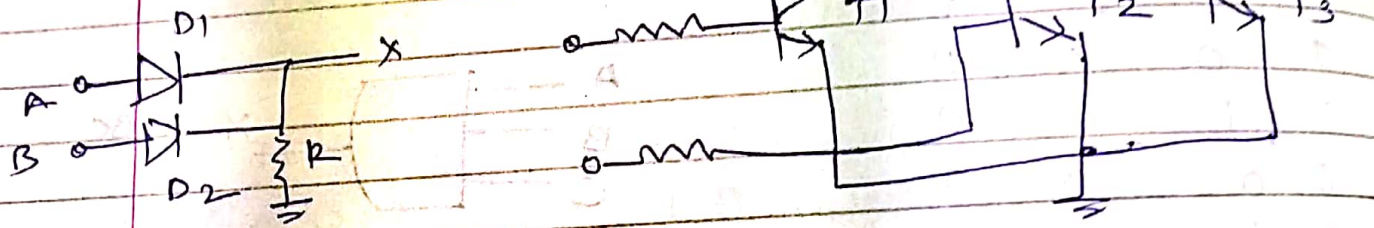




A	B	X
0	0	0
0	1	1
0	1	1
1	0	1
1	1	1

A	B	C
0	0	0
0	0	1
0	1	0
0	1	1
1	0	0
1	0	1
1	1	0
1	1	1

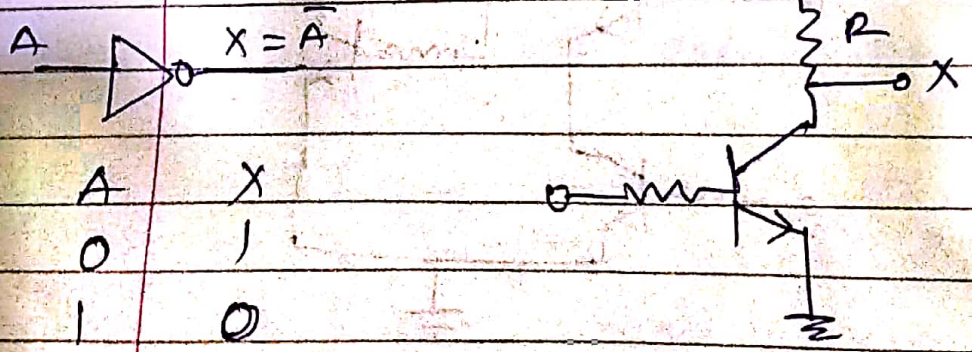
Two or three ~~OR~~ i/p gate



Two-i/p diode ~~OR~~ Gate

Two-i/p Transistor OR

NOT GATES (inverter)



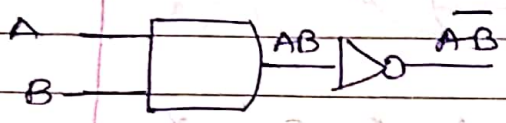
A	X
0	1
1	0

(*) The Universal Gates:-

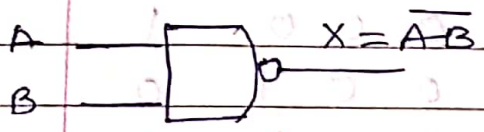
NAND and NOR

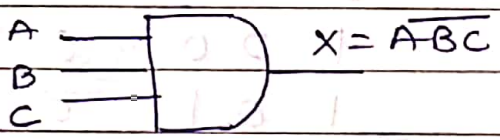
The NAND Gate

NAND - NOT, AND combination



A	B	X
0	0	1
0	1	1
1	0	1
1	1	0

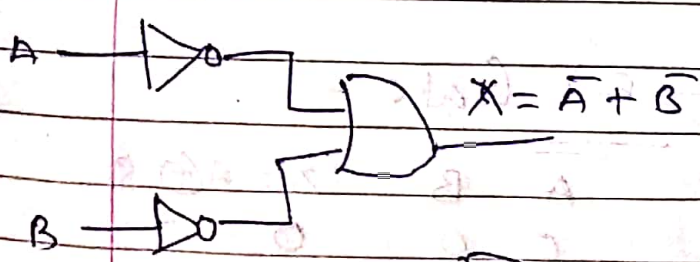




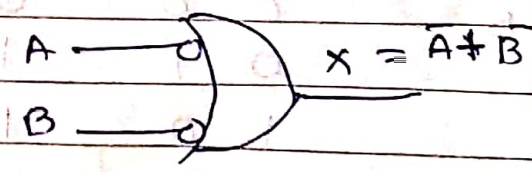
A	B	C	X
0	0	0	1
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	0

Two & three i/p NAND gates

↳ (Negative OR gate)



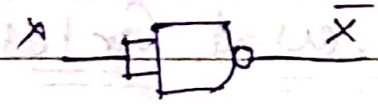
A	B	$\bar{A}$	$\bar{B}$	$X = \bar{A}+\bar{B}$
0	0	1	1	0
0	1	1	0	1
1	0	0	1	1
1	1	0	0	0



$X = \overline{AB} = \bar{A} + \bar{B}$

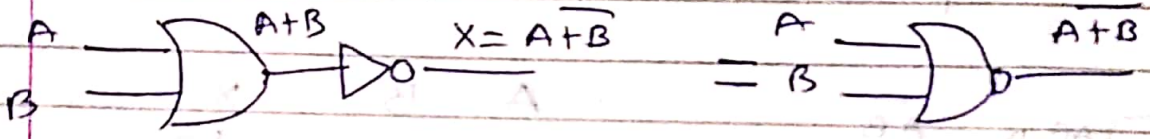
Bubbled OR Gate

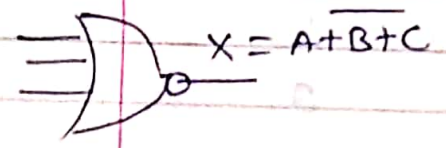
NAND gate as inverter



The NOR Gate

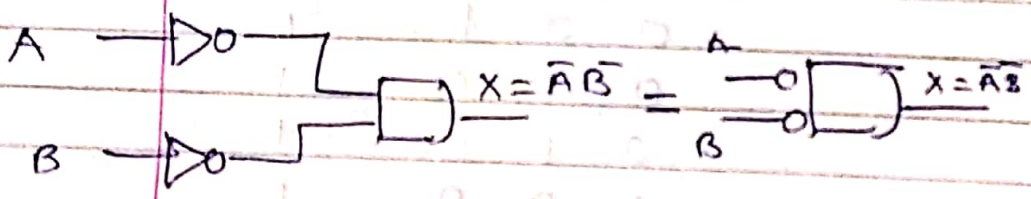
NOR - NOT, OR Combination

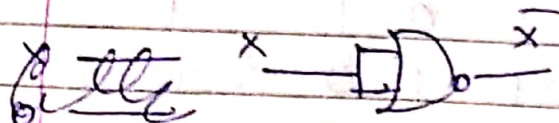




A	B	X
0	0	1
0	1	0
1	0	0
1	1	0

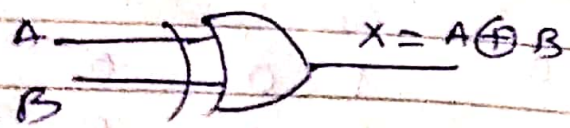
A	B	C	X
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	0





NOR Gate as an inverter

(*) Exclusive-OR (Ex-OR) Gate :-



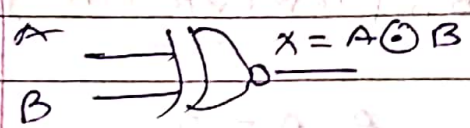
A	B	X = A ⊕ B
0	0	0
0	1	1
1	0	1
1	1	

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

Page No.:
Date: / /

Anti-Coincidence gate or inequality detector

(*) X-NOR Gate:- (Equality detector)



A	B	X
0	0	1
0	1	0
1	0	0
1	1	1

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

$A \odot B \odot C \neq \overline{A \oplus B \oplus C}$

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

Axiom:

$0.0 = 0$
$0.1 = 0$
$1.0 = 0$
$1.1 = 1$
$0+0 = 0$
$0+1 = 1$
$1+0 = 1$
$1+1 = 1$
$\bar{1} = 0$
$\bar{0} = 1$

$A.0 = 0$
$A.1 = A$
$A.A = A$
$A.\bar{A} = 0$
$A+0 = A$
$A+1 = 1$
$A+A = A$
$A+\bar{A} = 1$
$A+B = B+A$ } Commutative laws
$A.B = B.A$ }
$(A+B)+C = A+(B+C)$ } Associative laws
$(A.B).C = A.(B.C)$ }

$A(B+C) = AB + AC$
(Distributive laws)

(*) The KARNAUGH MAPS (K-MAP)

An n Variable function can have 2^n possible combinations of SOP form (↳ Product terms of Sum terms) in POS form.

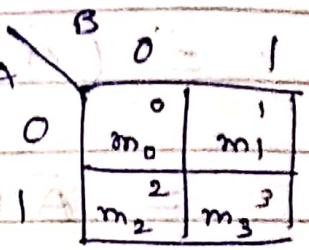
2-variable $2^2 = 4$ cells

3-variable $2^3 = 8$ cells

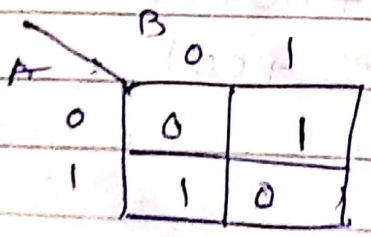
Product terms in SOP form – minterm

Sum terms in POS form – maxterm

Maxterm		minterm	I/P A	I/P B	O/P
$A+B$	M_0	0	0	0	$m_0 = \bar{A}\bar{B}$
$A+\bar{B}$	M_1	1	0	1	$m_1 = \bar{A}B$
$\bar{A}+B$	M_2	2	1	0	$m_2 = A\bar{B}$
$\bar{A}+\bar{B}$	M_3	3	1	1	$m_3 = AB$

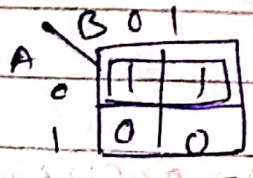


Ex:- Map the Expression $\bar{A}B + A\bar{B}$

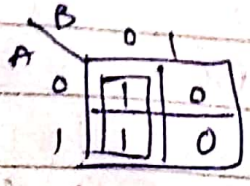


minterms

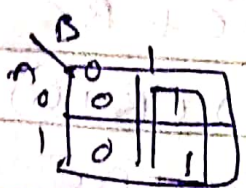
Combination



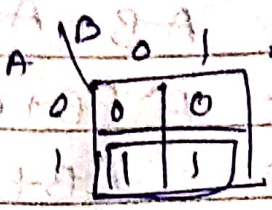
$f = \bar{A}$



$f_2 = \bar{B}$



$f_3 = B$

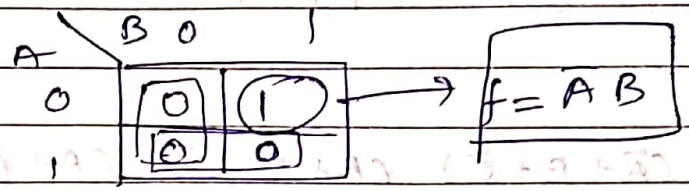


$f_4 = A$

$f_5 = 1$

POS → 0s are placed in the Squares

Ex:- Plot the Expression $(A+B)(\bar{A}+B)(\bar{A}+\bar{B})$



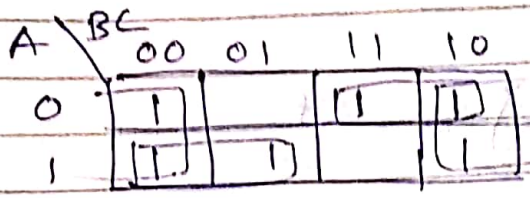
Three Variable K-Map:-

A \ BC	00	01	11	10
0	m_0 0	m_1 1	m_3 3	m_2 2
1	m_4 4	m_5 5	m_7 7	m_6 6

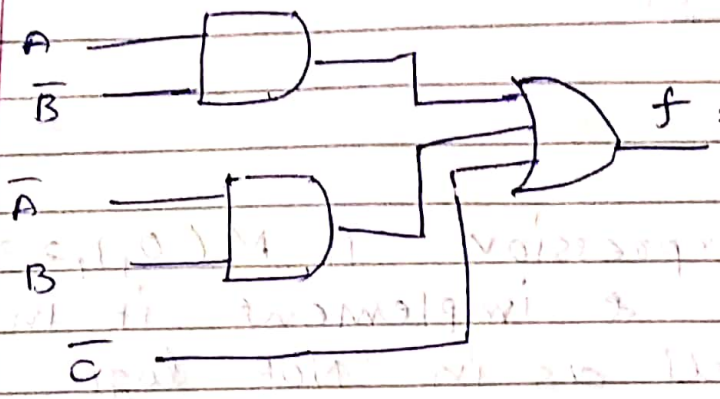
A \ BC	00	01	11	10
0	M_0	M_1	M_3	M_2
1	M_4	M_5	M_7	M_6

	A	B	C	minterm		Maxterm	
0	0	0	0	$\bar{A}\bar{B}\bar{C}$	m_0	$A+B+C$	M_0
1	0	0	1	$\bar{A}\bar{B}C$	m_1	$A+B+\bar{C}$	M_1
2	0	1	0	$\bar{A}B\bar{C}$	m_2	$A+\bar{B}+C$	M_2
3	0	1	1	$\bar{A}BC$	m_3	$A+\bar{B}+\bar{C}$	M_3
4	1	0	0	$A\bar{B}\bar{C}$	m_4	$\bar{A}+B+C$	M_4
5	1	0	1	$A\bar{B}C$	m_5	$\bar{A}+B+\bar{C}$	M_5
6	1	1	0	$AB\bar{C}$	m_6	$\bar{A}+\bar{B}+C$	M_6
7	1	1	1	ABC	m_7	$\bar{A}+\bar{B}+\bar{C}$	M_7

Ex:- Reduce the Expression $\Sigma m(0,2,3,4,5,6)$

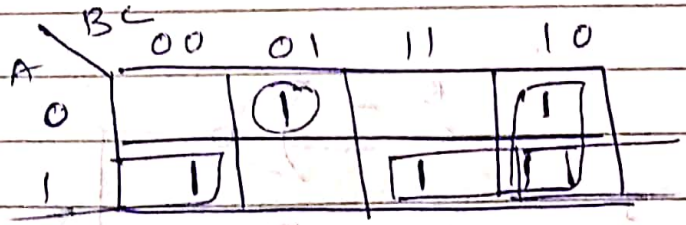


$f = \bar{C} + A\bar{B} + \bar{A}B$



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

Ex:- $\Sigma m(1,2,4,6,7)$ & implement using Universal gate

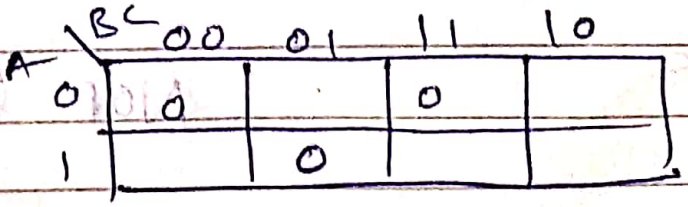


SOP

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

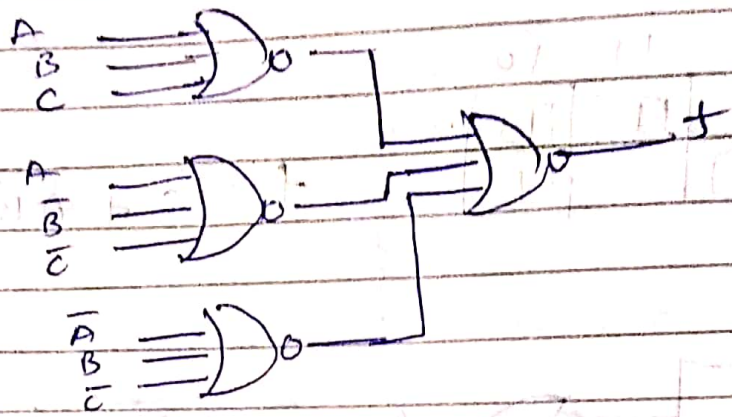
For Universal gate

POS



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

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

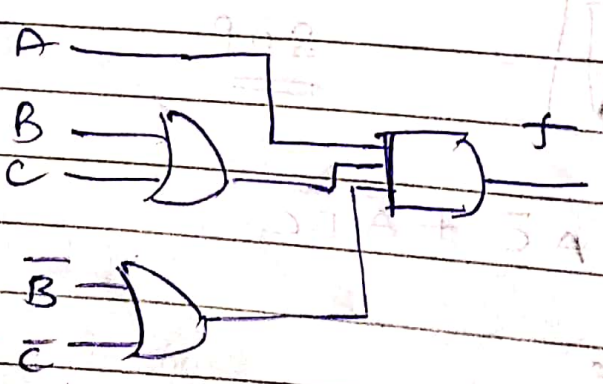


(8) ~~Reduce~~

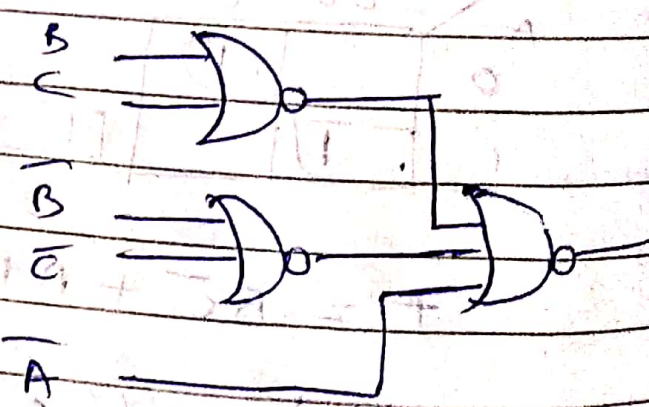
Ex:- Reduce the Expression $\Pi M(0,1,2,3,$ using Mapping & implement it in logic as well as in NOR logic.

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

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



AOI logic



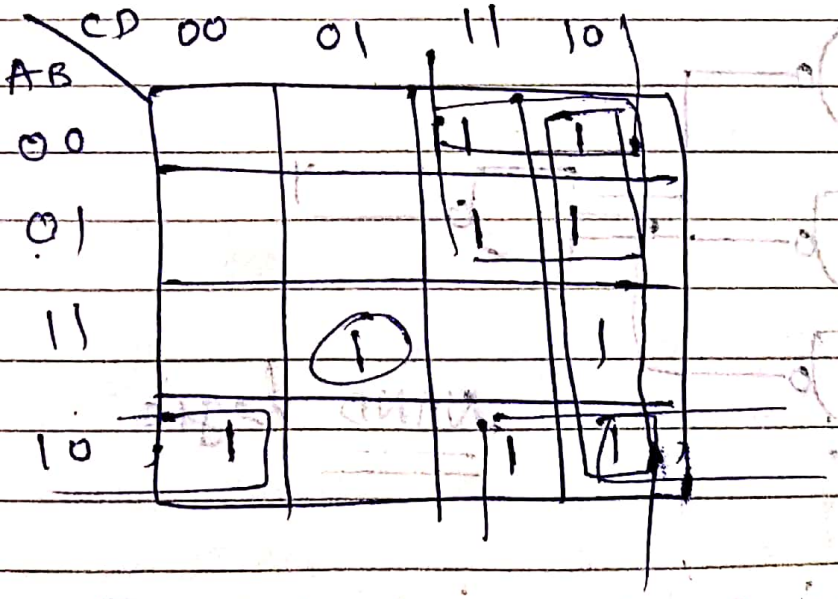
NOR logic

4-Variable $2^4 = 16$ Cells

AB \ CD	00	01	11	10
00	m_0	m_1	m_3	m_2
01	m_4	m_5	m_7	m_6
11	m_{12}	m_{13}	m_{15}	m_{14}
10	m_8	m_9	m_{11}	m_{10}

Ex:-

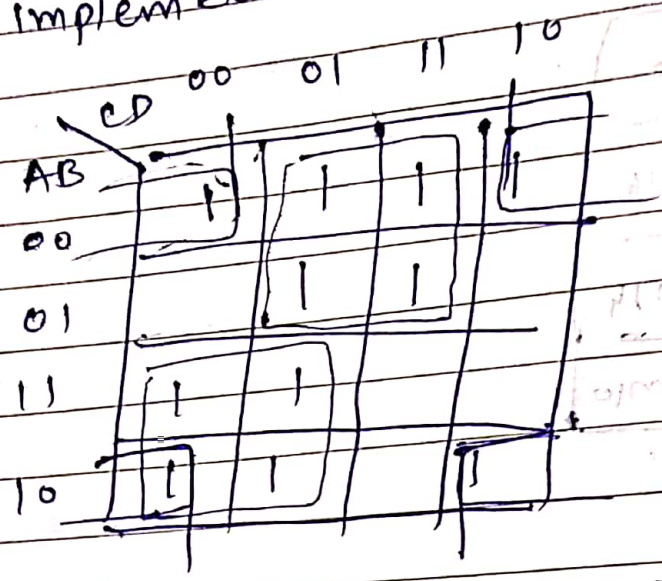
⇒ Reduce using mapping the Expression $\Sigma m(2,3,6,7,8,10,11,13,14)$.



$$Y = \bar{A}C + B\bar{C} + C\bar{D} + A\bar{B}\bar{D} + AB\bar{C}D$$

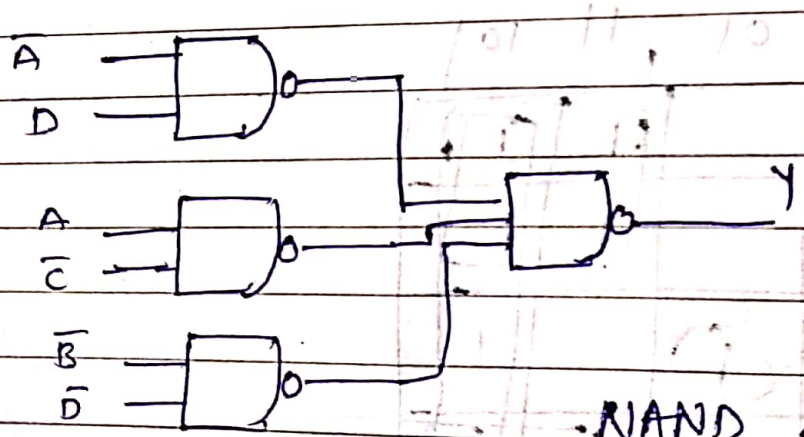
Ex:- Reduce using mapping the Expression $\Sigma m(0,1,2,3,5,7,8,9,10,12,13)$ and implement it is universal logic.

→



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

$= \overline{\bar{A}D} \cdot \overline{A\bar{C}} \cdot \overline{\bar{B}\bar{D}}$



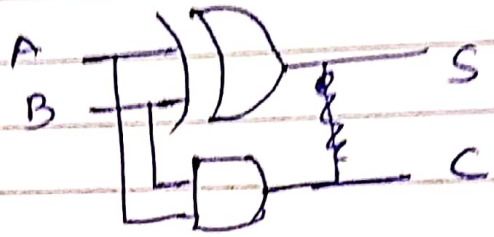
NAND logic

(*) Half - Adder :-

A	B	Sum	Carry
0	0	0	0
0	1	1	0
0	1	1	0
1	0	1	0
1	1	0	1

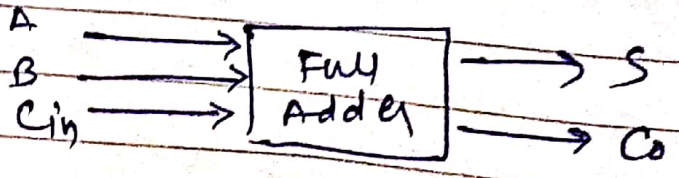
$Sum = \bar{A}B + A\bar{B} = A \oplus B$ (XOR)

Carry = AB (AND)



(*) Full - Adder :-

A	B	Cin	S	Cout
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

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

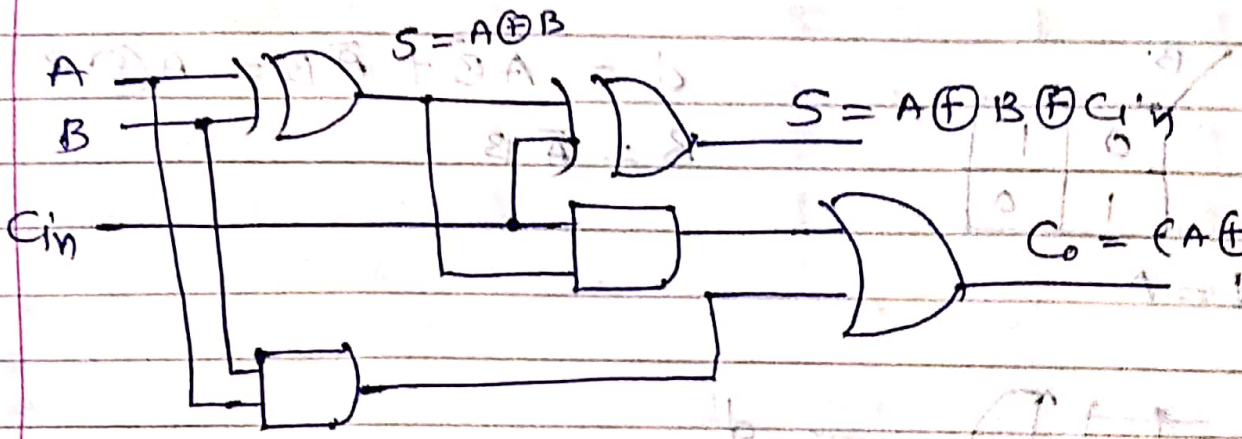
$S = \bar{A}\bar{B}\bar{C} + \bar{A}B\bar{C} + ABC + \bar{A}B\bar{C}$

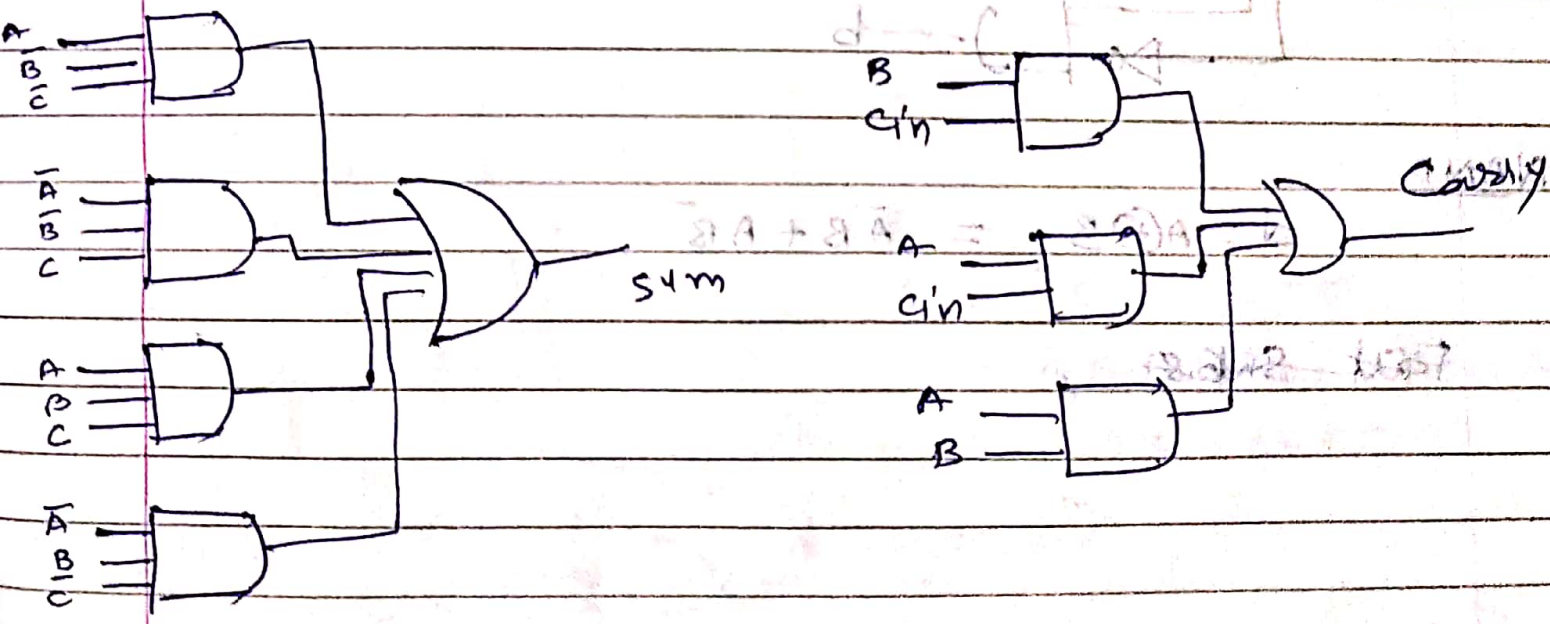
$= A \oplus B \oplus C_{in}$

Carry:

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

$Carry = BC_{in} + AC_{in} + AB$





(*) Half - Subtractor :-

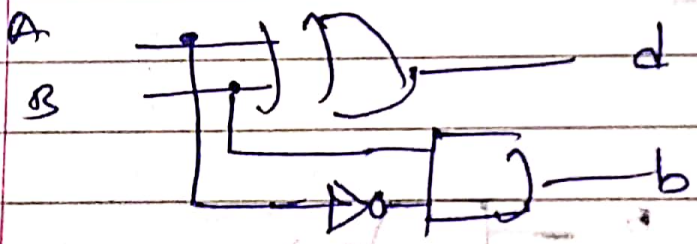
I/P		O/P	
A	B	d	b
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0

A \ B	0	1
0	0	1
1	1	0

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

$B = \bar{A}B$

$d = A$



NAND

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

~~Fault Subst~~

(*) Full-Substructor :-

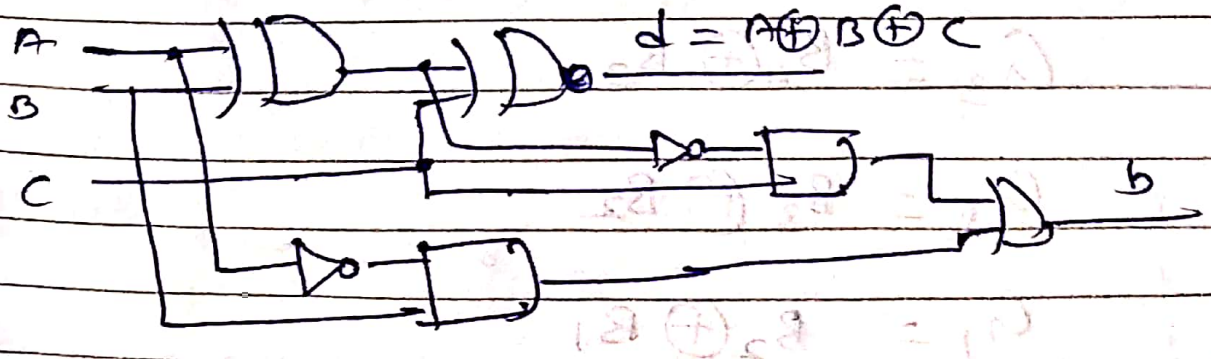
A	B	C	D	B
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	0	1
1	0	0	1	0
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1

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

$$d = \bar{A}\bar{B}C + \bar{A}B\bar{C} + A\bar{B}\bar{C} + ABC$$
$$= C(\bar{A}\bar{B} + AB) + \bar{C}(\bar{A}B + A\bar{B})$$
$$= C(A \odot B) + \bar{C}(A \oplus B)$$
$$= A \oplus B \oplus C$$

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

$$b = \bar{A}C + \bar{A}B + BC$$
$$= \bar{A}\bar{B}C + \bar{A}B\bar{C} + \bar{A}BC + ABC$$
$$= \bar{A}B + C(A \odot B)$$



$d = A \oplus B \oplus C$

(*) Binary to Gray Converter:-

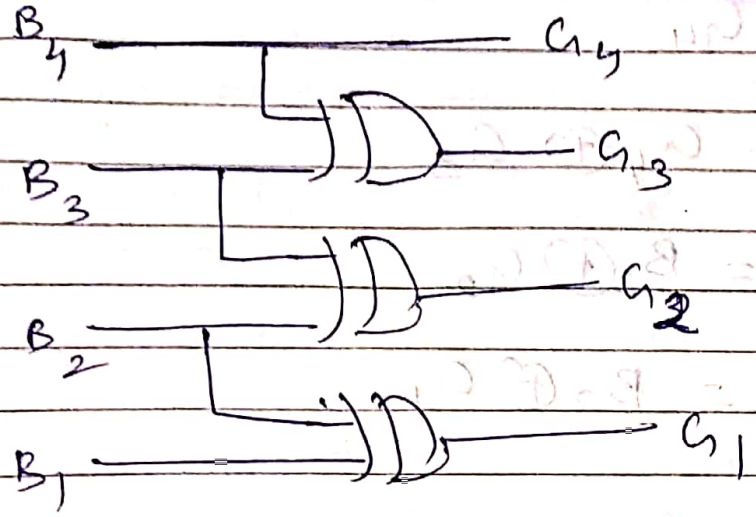
4-bit Binary				4-bit Gray			
B_4	B_3	B_2	B_1	G_4	G_3	G_2	G_1
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	1
0	0	1	0	0	0	1	1
0	0	1	1	0	0	1	0
0	1	0	0	0	1	1	0
0	1	0	1	0	1	1	1
0	1	1	0	0	1	0	1
0	1	1	1	0	1	0	0
1	0	0	0	1	1	0	0
1	0	0	1	1	1	0	1
1	0	1	0	1	1	1	1
1	0	1	1	1	1	1	0
1	1	0	0	1	0	1	0
1	1	0	1	1	0	1	1
1	1	1	0	1	0	0	1
1	1	1	1	1	0	0	0

$$G_4 = B_4$$

$$G_3 = B_4 \oplus B_3$$

$$G_2 = B_3 \oplus B_2$$

$$G_1 = B_2 \oplus B_1$$



(*) Gray to Binary Converter! -

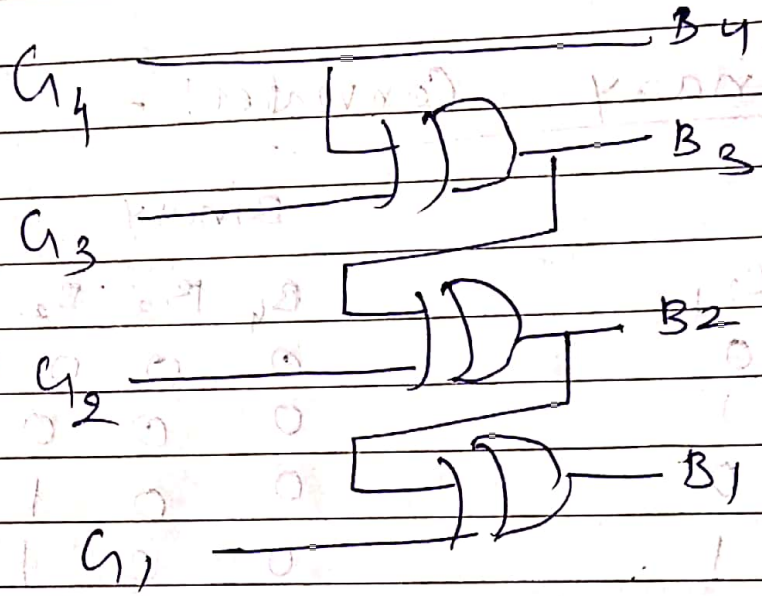
Gray				Binary			
G_4	G_3	G_2	G_1	B_4	B_3	B_2	B_1
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	1
0	0	1	0	0	0	1	0
0	0	1	1	0	0	1	1
0	1	0	0	0	1	0	0
0	1	0	1	0	1	0	1
0	1	1	0	0	1	1	0
0	1	1	1	0	1	1	1
1	0	0	0	1	0	0	0
1	0	0	1	1	0	0	1
1	0	1	0	1	0	1	0
1	0	1	1	1	0	1	1
1	1	0	0	1	1	0	0
1	1	0	1	1	1	0	1
1	1	1	0	1	1	1	0
1	1	1	1	1	1	1	1

$B_4 = G_4$

$B_3 = G_4 \oplus G_3$

$B_2 = B_3 \oplus G_2$

$B_1 = B_2 \oplus G_1$



(*) Decoder:-

→ Binary code of n bit is capable of representing up to 2^n distinct elements of the coded information

→ A decoder is a combinational ckts that converts n-i/p lines to a maximum of 2^n unique output lines

n-to-m-line decoder
$(m \leq 2^n)$

→ It is used for code converters such as BCD to seven-segment decode.

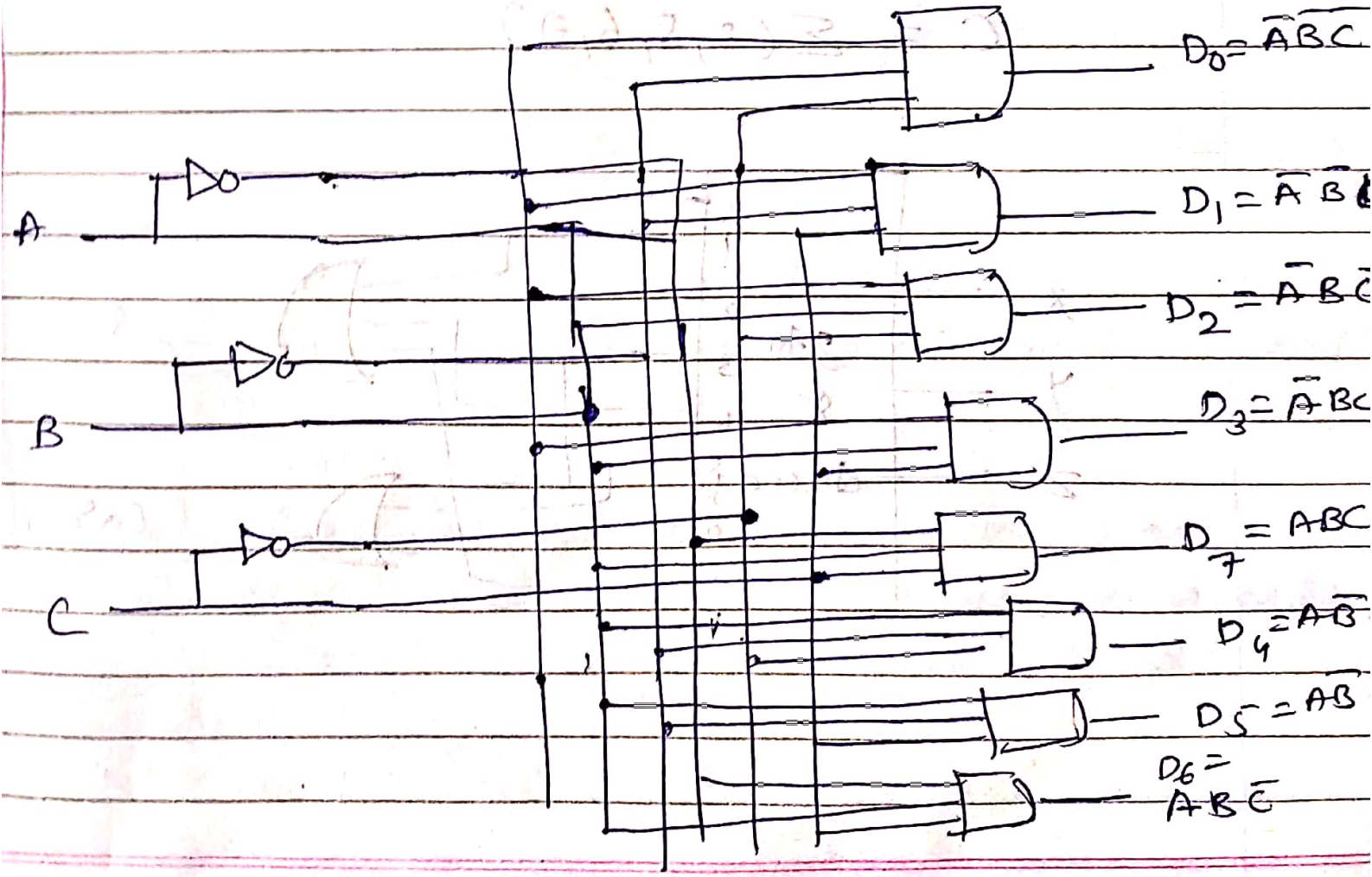
3 to 8-line decodes

I/p: $n=3$ → o/p $2^n=8$

- Output represent one of the minterms of the 3-input variables.

- 3 inverters provide complements of the input & each one of the eight AND gates generates one of the minterms.

— Application : Binary-to-Octal Conversi

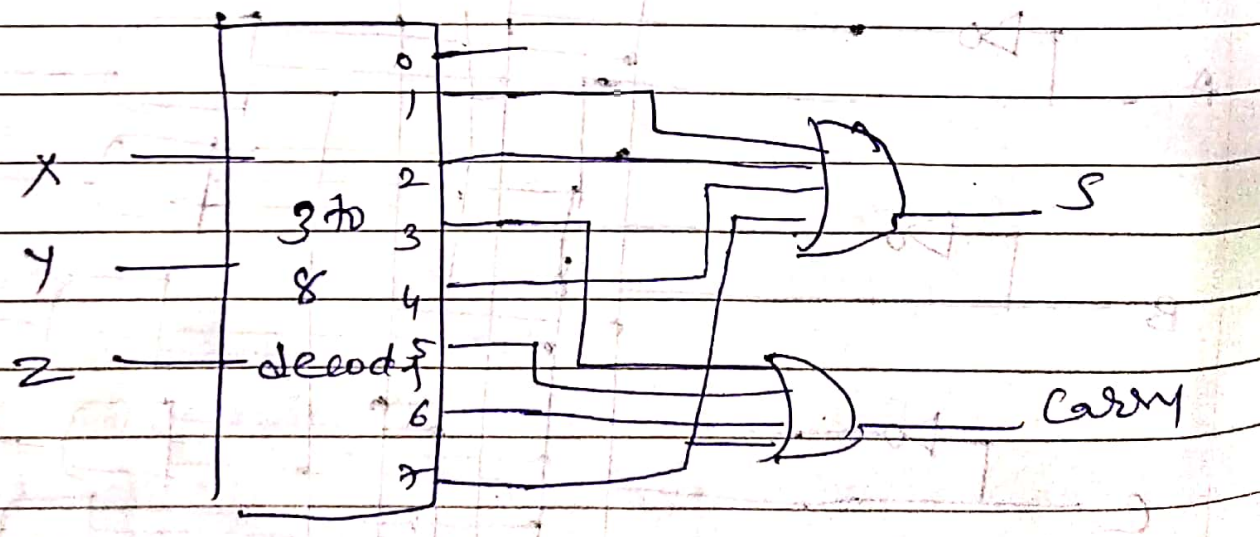


I/P			O/P							
A	B	C	D_0	D_1	D_2	D_3	D_4	D_5	D_6	D_7
0	0	0	1	0	0	0	0	0	0	0
0	0	1	0	1	0	0	0	0	0	0
0	1	0	0	0	1	0	0	0	0	0
0	1	1	0	0	0	1	0	0	0	0
1	0	0	0	0	0	0	1	0	0	0
1	0	1	0	0	0	0	0	1	0	0
1	1	0	0	0	0	0	0	0	1	0
1	1	1	0	0	0	0	0	0	0	1

Example Implement a full-adder circuit with a decoder & two OR gates

→ $S = \Sigma(1,2,4,7)$

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

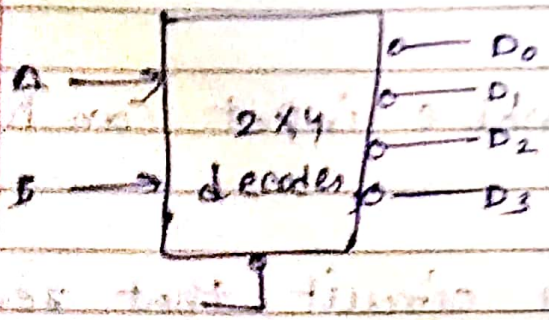


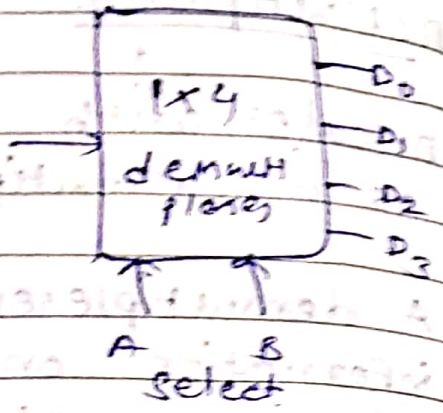
(*) Demultiplexer:-

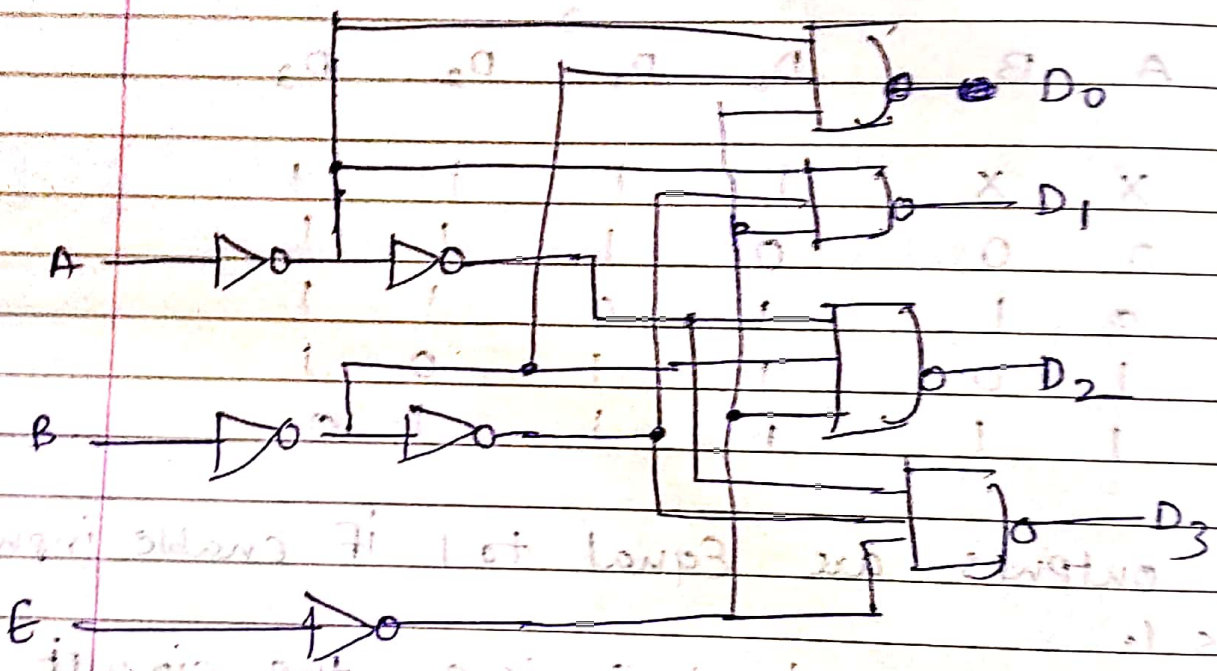
- Decoder with an enable input can function as a demultiplexer.
- A demultiplexer is a circuit that receives information on a single line & transmits this information on one of 2^n possible output lines.

E	A	B	D_0	D_1	D_2	D_3
1	X	X	1	1	1	1
0	0	0	0	1	1	1
0	0	1	1	0	1	1
0	1	0	1	1	0	1
0	1	1	1	1	1	0

- All outputs are Equal to 1 if enable input E is 1.
- When Enable input E is 0, the circuit operates as a decoder with Complemented outputs.
- Decoder, and demultiplexer operations are obtained from the same circuit, a decoder with enable input is referred as a decoder / demultiplexer.





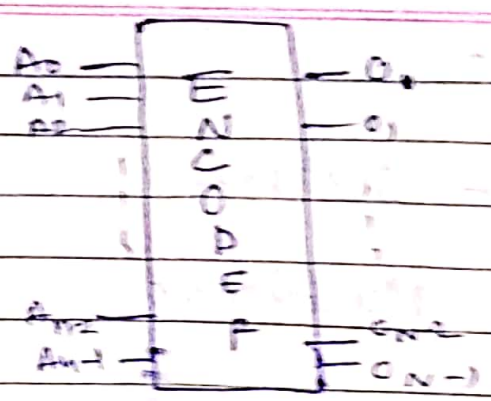


(*) Encoders

→ An Encoder is a digital circuit that performs inverse operation of a decoder.

→ An Encoder has 2^n i/p lines & n output lines.

ORing in Encoder



Octal-to-Binary Encoder

- decoder (3 line-to-8 decoder)

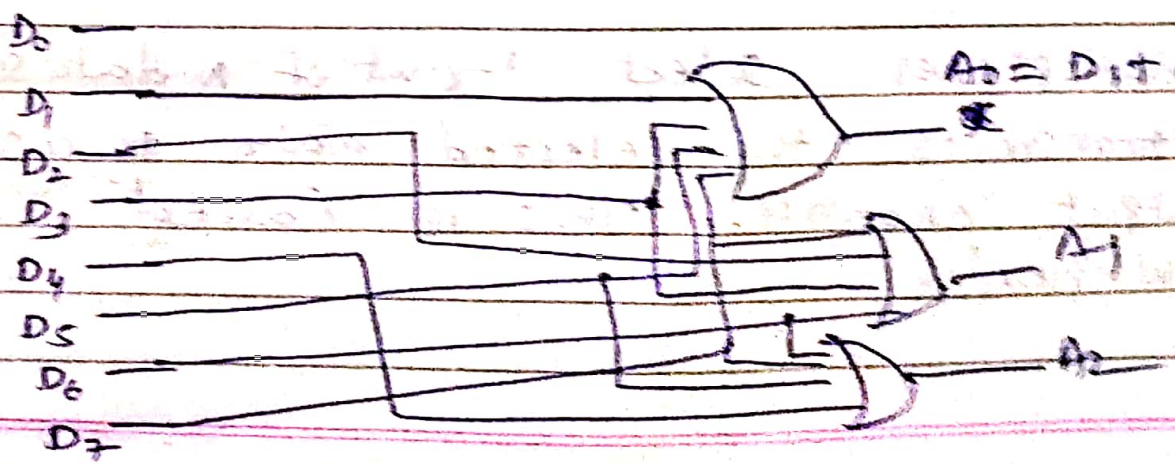
Encoder (8 line to 3-line Encoder)

Octal Digits		Binary		
		A_2	A_1	A_0
0	D_0	0	0	0
1	D_1	0	0	1
2	D_2	0	1	0
3	D_3	0	1	1
4	D_4	1	0	0
5	D_5	0	0	1
6	D_6	1	1	0
7	D_7	1	1	1

$A_0 = D_1 + D_3 + D_5 + D_7$

$A_1 = D_2 + D_3 + D_6 + D_7$

$A_2 = D_4 + D_5 + D_6 + D_7$



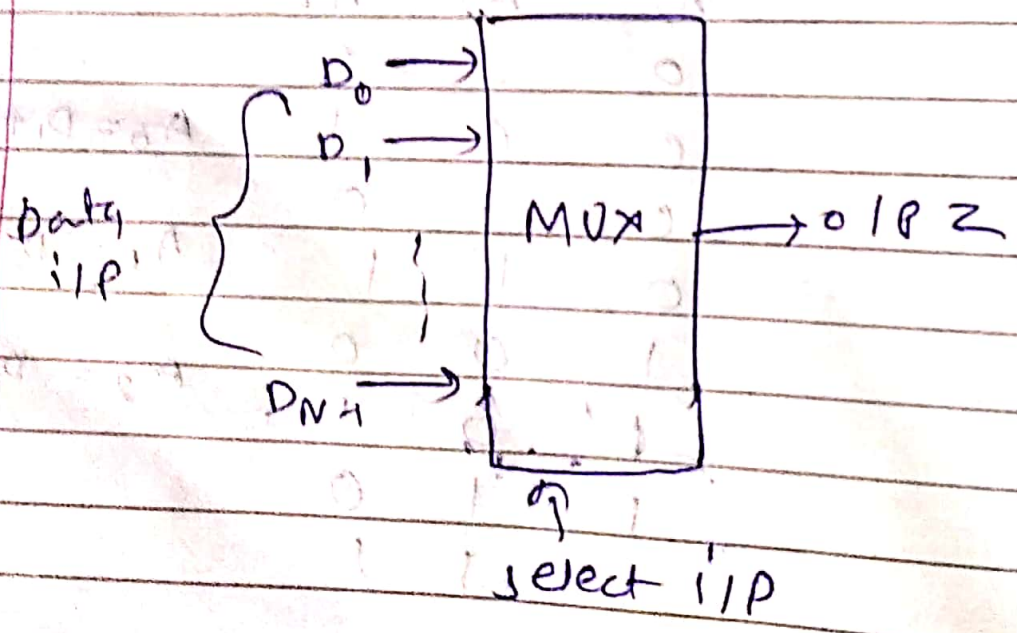
(*) Multiplexers:-

Multiplexing - Sharing

2^N to 1 Multiplexer

- A Multiplexer or data selector is a logic circuit that selects several data inputs and allows only one of them at a time to get through the output.

- Routing of the desired data input to the output is controlled by SELECT inputs.



→ Multiplexer select 1-out of N data sou & transmits the selected data to a sin output channel. This is called Multiplexing.