

DIGITAL LOGIC → logic gates

- analog → continuous while digital → 0 (off) / 1
- XOR → like inputs → 0; unlike inputs → 1

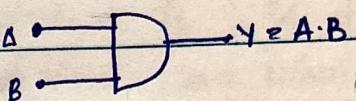
Basic gates → AND, OR and NOT

Universal gates → NAND and NOR

Exclusive gates → XOR and XNOR

(1) Logic expression: $A \cdot B$ (AND) $A \cdot B = Y$

Logic symbol:

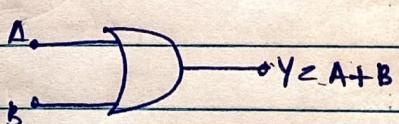


Truth table :

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

(2) logic expression: $A + B = Y$

Logic symbol:



Truth Table :

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

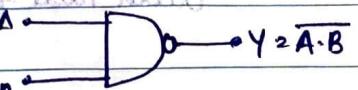
(3) logic expression: $\bar{A} = Y$

Logic symbol:



Truth Table:

	A	B	Y
0	-	1	1
1	-	0	0

(4) logic expression: $Y = \overline{A \cdot B}$ logic symbol: 

NOT of NOT

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

(5) logic expression: $Y = \overline{A + B}$ logic symbol: 

NOT of OR

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

Truth table:

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

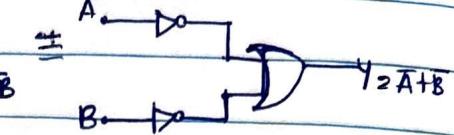
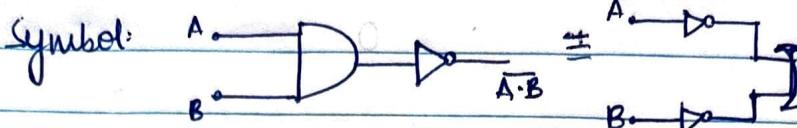
1st LAW: Complement of the product is equal to sum of the individual components1st LAW: $A \cdot B = \overline{\overline{A} + \overline{B}}$ 2nd LAW: Complement of the sum is equal to product of the individual components2nd LAW: $\overline{A + B} = \overline{A} \cdot \overline{B}$

step may give step for writing

(Q) Prove the 1st LAW:

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

Symbol:

 $\overline{A} = Y$ indicates true signal

Truth table for LHS:

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

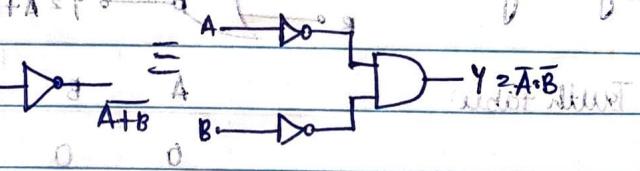
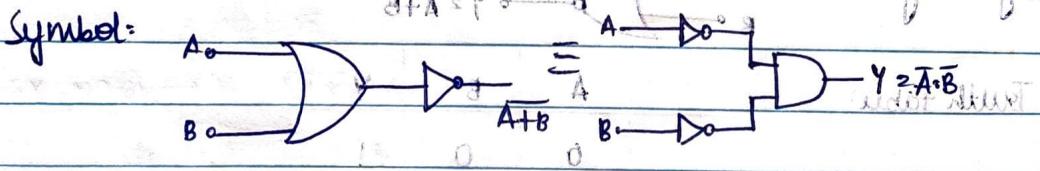
Truth table for RHS

A	B	\overline{A}	\overline{B}	$Y = \overline{A} + \overline{B}$
0	0	1	1	1
0	1	1	0	1
1	0	0	1	1
1	1	0	0	0
		0	1	1

(Q) Prove the 2nd LAW:

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

Symbol:



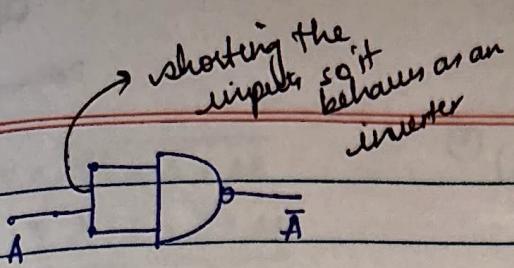
Truth table for LHS:

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

Truth table for RHS

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

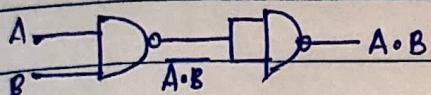
- (*) REALISATION OF BASIC GATES USING UNIVERSAL GATES : TFA
- Realisation of NOT gate using NAND gate



$$A \bar{A} = 1 \quad (1)$$

$$A \cdot (\bar{A} \cdot \bar{A}) = A \cdot 0 = 0 \quad (2)$$

(2) Realisation of AND gate using NAND gate.

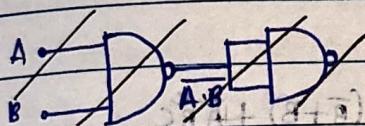


$$\bar{A} \bar{B} = 1 \quad (3)$$

$$(\bar{A} \bar{B}) \cdot \bar{A} = \bar{A} \bar{B} + \bar{A} \bar{B} = \bar{A} \bar{B} \quad (4)$$

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

(3) Realisation of OR gate using NAND gate



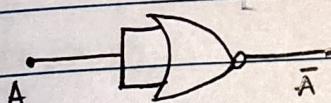
$$A \bar{A} + \bar{A} \bar{B} + \bar{A} B = 1 \quad (5)$$

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

$$[1 + \bar{A} \bar{B}] \bar{A} =$$

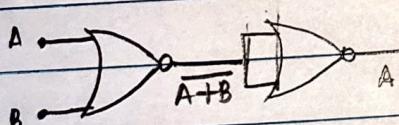
$$[1 + \bar{A}] \bar{A} =$$

(4) Realisation of NOT gate using NOR gate



$$A \oplus A = 0 \quad (6)$$

(5) Realisation of AND gate using NOR gate



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

$$A \bar{A} + \bar{B} \bar{A} = V \leftarrow (2)(4)X$$

(6) $\bar{A} = A$

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

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

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

$$\bar{A} \bar{A} + \bar{Y} \bar{A} + \bar{Y} \bar{A} = V \quad (7)$$

$$[1 + \bar{Y} + \bar{A}] \bar{A} =$$

$$[1 + \bar{A}] \bar{A} =$$

$$[1 + 1] \bar{A} =$$

$$[2 + 1] \bar{A} =$$

$$[3 + 1] \bar{A} =$$

BOOLEAN LAWS:

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

AND LAWS:

$$\bar{0} = 1$$

$$A \cdot 0 = 0$$

$$\bar{1} = 0$$

$$A \cdot 1 = A$$

OR LAWS: $(\bar{A} + 1)(1 + A) = 1 \quad (8)$

$$A + 0 = A$$

$$(1 + A) = 1$$

$$A + 1 = 1$$

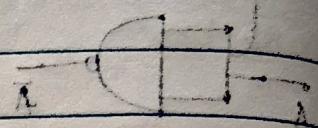
$$A + A = A$$

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

POSTULATES

(1) $A + AB$

$$A(I+A) \stackrel{!}{=} A(I) = A$$



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

$$\downarrow A + AB + \bar{A}B$$

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

$$A + B$$



(*) SIMPLIFICATION OF BOOLEAN EXPRESSION:

(1) $ABC + \bar{A}BC + \bar{A}BC$

$= \bar{A}[B\bar{C} + B\bar{C} + BC] \quad \text{MAM'S WAY: } \bar{A}\bar{C}(\bar{B} + B) + \bar{A}BC$

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

$= \bar{A}[\bar{C} + BC]$

$\Rightarrow \bar{A}[\bar{C} + BC] \quad \text{to minimize}$

$\Rightarrow \boxed{A + AB = A + B}$

EX(OR) $\rightarrow Y = A \oplus B$

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

EX(NOR) $\rightarrow Y = \overline{A \oplus B}$

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

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

$\Rightarrow A[B\bar{C} + \bar{B}C + \bar{B}\bar{C}]$

$\Rightarrow A[C(B + \bar{B}) + \bar{B}\bar{C}]$

$\Rightarrow A[C + \bar{B}\bar{C}]$

$\Rightarrow A[C + B]$

(3) $f = XY + XYZ + XY\bar{Z} + X\bar{Y}Z$

$\Rightarrow XY[1 + \bar{Z}] + YZ[X + \bar{X}]$

$\Rightarrow XY[1 + \bar{Z}] + YZ$

$\Rightarrow Y[X + X\bar{Z} + Z]$

$\bar{Z} + \bar{Z} + \bar{Z}$

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

$\Rightarrow (A+B)($

$A = 0 + A$

$0 = 0 \cdot A$

$I = I + A$

$A = I \cdot A$

$A = A + A$

$A = A \cdot A$

$I = \bar{A} + A$

$0 = A \cdot A$

$A = \bar{A}$

$I = \bar{0}$

$0 \neq I$

$$\begin{aligned}
 @1) & Y = \overline{AB} + \overline{A} + AB \\
 & \Rightarrow \overline{A} + \overline{B} + \overline{A} + AB \\
 & \Rightarrow \overline{AB} + \overline{A} + B \\
 & \Rightarrow \overline{A} + \overline{B} + \overline{A} + B \\
 & = \overline{1 + 2\overline{A}}
 \end{aligned}$$

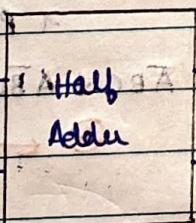
(2)

$$\begin{aligned}
 & Y = \overline{L + M + MN + LN} \\
 & Y = \overline{L \cdot \overline{M}} + \overline{M \cdot \overline{N}} \\
 & \Rightarrow \overline{M} \cdot (\overline{L} + \overline{N}) \\
 & \Rightarrow M + \overline{L} + \overline{N} \\
 & \Rightarrow M + \overline{L + N}
 \end{aligned}$$

$$\begin{aligned}
 @3) & Y = \overline{LM} + \overline{LN} + MN + \overline{LN} \\
 & \Rightarrow \overline{LM} + N(\overline{L} + M) + \overline{LN} \\
 & \Rightarrow \overline{L}(\overline{M} + N) + N(\overline{L} + M) \\
 & \Rightarrow (\overline{L} + M)(\overline{L} + N)
 \end{aligned}$$

$$\begin{array}{ccccc}
 0 & 0 & 0 & 0 & 0 \\
 M(\overline{L} + N) + LN & & & & 0 \quad 0 \\
 M(\overline{L} + \overline{N}) + LN & & & 1 & 0 \\
 M\overline{L} + MN + LN & & & 1 & 0 \\
 0M\overline{L} + LN(M+1) & & & 0 & 1 \\
 M\overline{L} + LN & & & 1 & 0 \quad 1 \\
 \hline
 1 & 0 & 0 & 1 & 1 \\
 1 & 1 & 1 & 1 & 1
 \end{array}$$

HALF ADDER



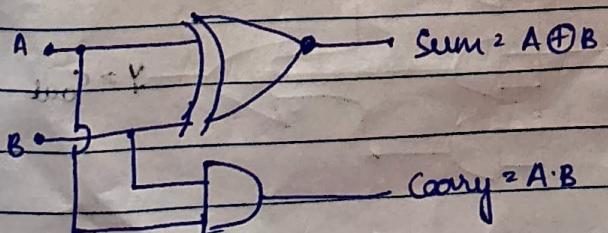
- if output depends on the previous state \rightarrow sequential logic circuit
- if output is independent on the previous state \rightarrow combinational logic circuit.

$$1+0 = (1)_2$$

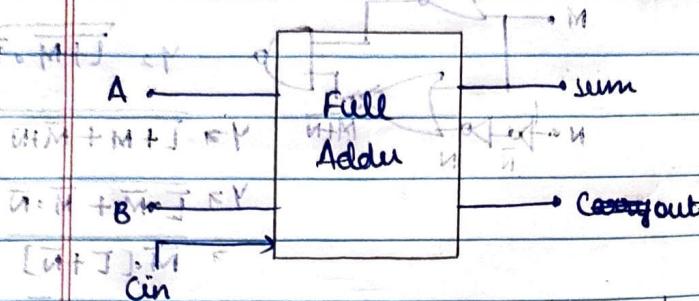
	A	B	Sum	Carry
0+1 = (1) ₂	0	0	0	0
0+0 = (0) ₂	0	1	1	0
1+1 = (10) ₂	1	0	1	0
	1	1	0	1

$$\text{Sum} \Rightarrow A \oplus B$$

$$\text{Carry} \Rightarrow A \cdot B$$



FULL ADDER:



Sum + M	Cin	Sum	Cout
A	B	Cin	Sum
0	0	0	0
0	0	1 0 1 + (111) 1 0	0
0	1	1 0 1 + (111) 1 0	0
0	1	1 1 + 0 1 0 + 1 1 1	✓
01	0	(0+1) 1 1 0 1 0	0
1	0	1 + 1 0 + 1 1 1	✓
1	1	0	0
1	1	1	1 ✓

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

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

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

$$17 \quad \overline{AB} + AB = \overline{a}$$

$$AB + \bar{AB} = x$$

$$\textcircled{1} \leftarrow C(\bar{x}) + \bar{C}(x) \Rightarrow A \oplus B \oplus C$$

$$C_{out} = \overline{ABC} + ABC + ABC +$$

ABC 300A JAH

$$\Rightarrow P\bar{A}BC + A\bar{B}C + AB(\bar{C} + C)$$

$$2A \leftarrow C(A+B)+AB \xrightarrow{AB} 2B$$

$$2 \leftarrow C(A \oplus B) + AB \quad 3$$

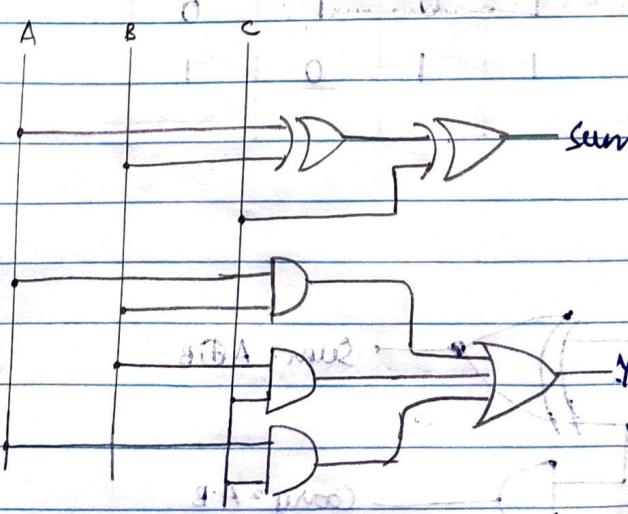
$$L(A \oplus B) + AB = L(A) + L(B)$$

$$\Rightarrow ABC + A(\bar{B}C + B)$$

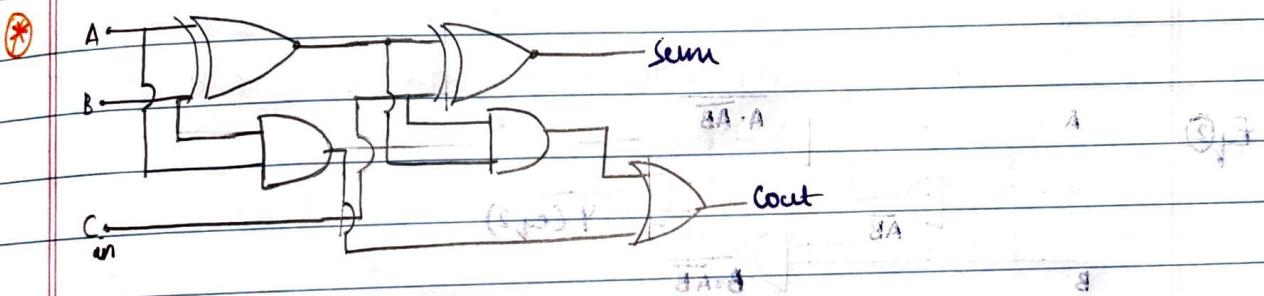
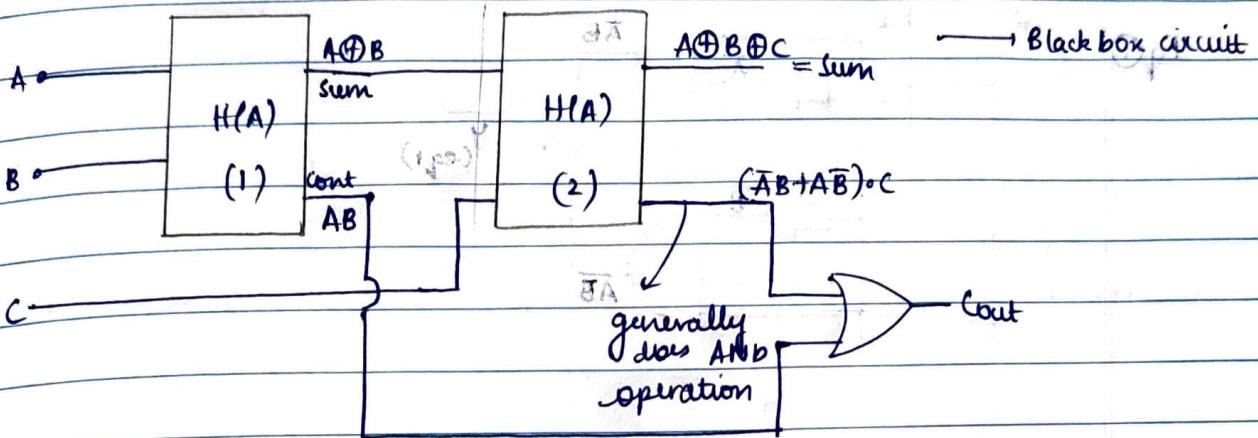
$$ABC + AB + AC \quad \text{and} \quad A \quad \text{(i)st}$$

$$\exists B(AC+A)+AC$$

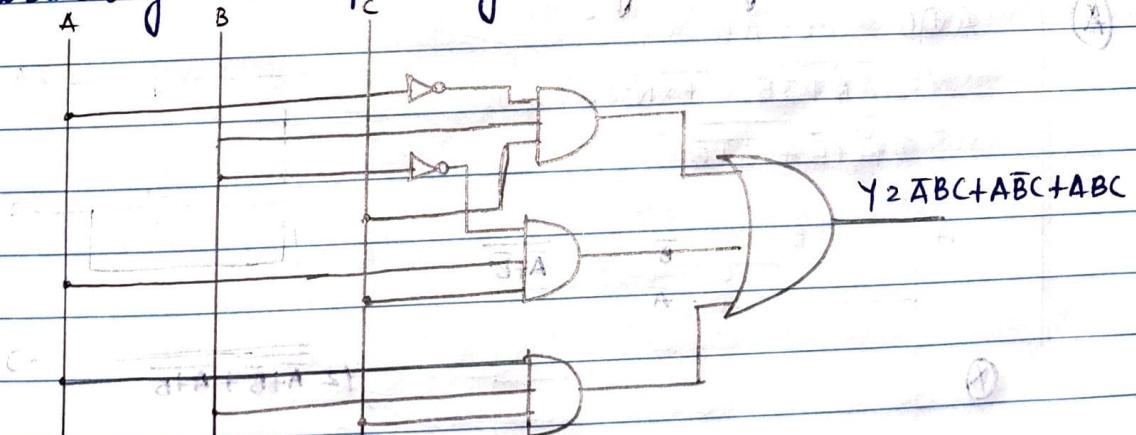
$$\text{GATE}^2 \quad AB + BC + AC \quad \xrightarrow{2} \quad \xrightarrow{3}$$



REALISATION OF FULL ADDER USING TWO HALF ADDERS BY



(Q) Draw the logic circuit for the given expression: $Y = \bar{A}BC + A\bar{B}C + ABC$



REALISATION OF XOR USING NAND GATES (minimum NAND gates)

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

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

$$\Rightarrow \overline{A\bar{B} \cdot \bar{A}B} \quad \text{① need 5 NAND gates}$$

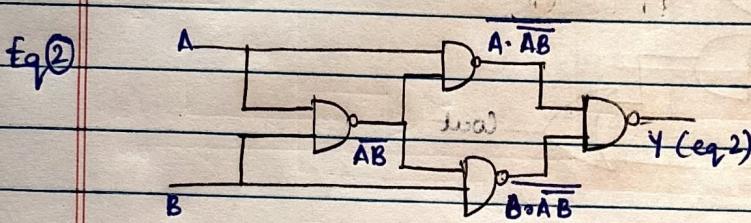
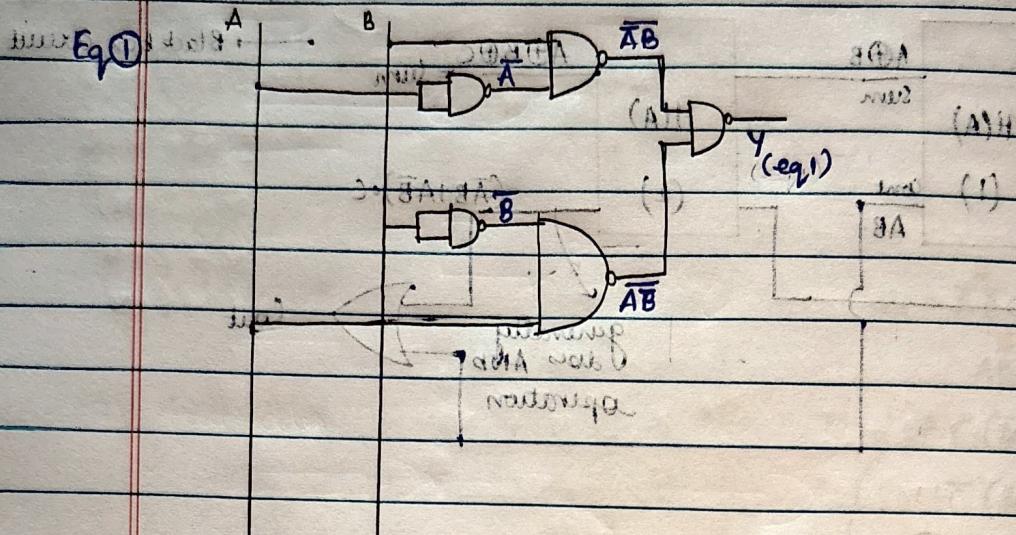
$$\Rightarrow (\bar{A} + B) \cdot (\bar{A} + \bar{B})$$

$$\Rightarrow \overline{(\bar{A} + B) \cdot (\bar{B} + AB)}$$

$$\Rightarrow \overline{\bar{A} + AB} \cdot \overline{\bar{B} + AB}$$

Taking double compliment

Y₁ A · AB + B · AB \Rightarrow ② 1 NAND gate required



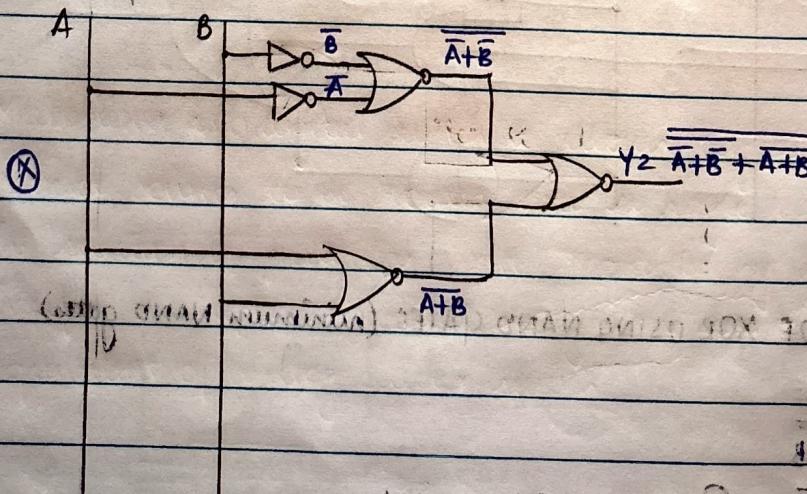
Q (a) Realise XOR using NOR gates.

$$(A) A \oplus B \Rightarrow A\bar{B} + \bar{A}B$$

$$Y_2 \overline{AB + \bar{AB}} = \text{EX-NOR} = \text{EX-OR}$$

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

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



Q (a) Realise XNOR using minimum NAND gates.

(A)

$$\text{minimum width grid} \quad (\bar{A} + \bar{B}) \cdot (\bar{A} + B) =$$

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

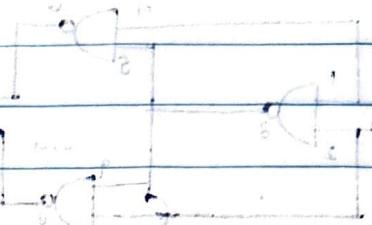
$$Y = \overline{AB + \overline{A}\overline{B}}$$

$$\Rightarrow (\overline{A}\overline{B}) \cdot (\overline{A}B) \rightarrow 5 \text{ gates required}$$

A

B

(X)



Author: Nitin (S)

(Q) Realise XNOR using NOR gates. (minimum) \rightarrow 4 gates

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

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

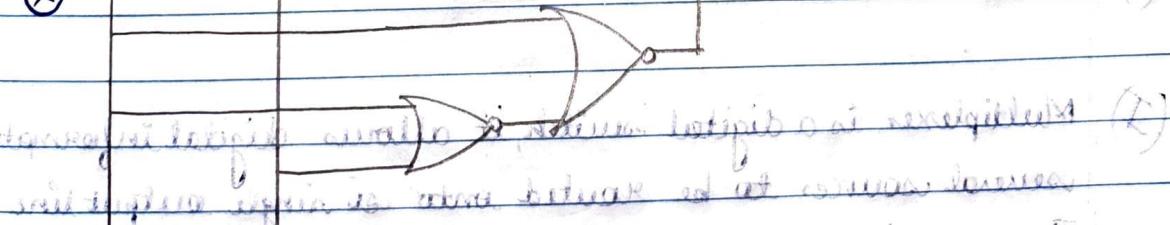
$$\Rightarrow \overline{B + \overline{A}B} + \overline{A + \overline{A}B}$$

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

A

B

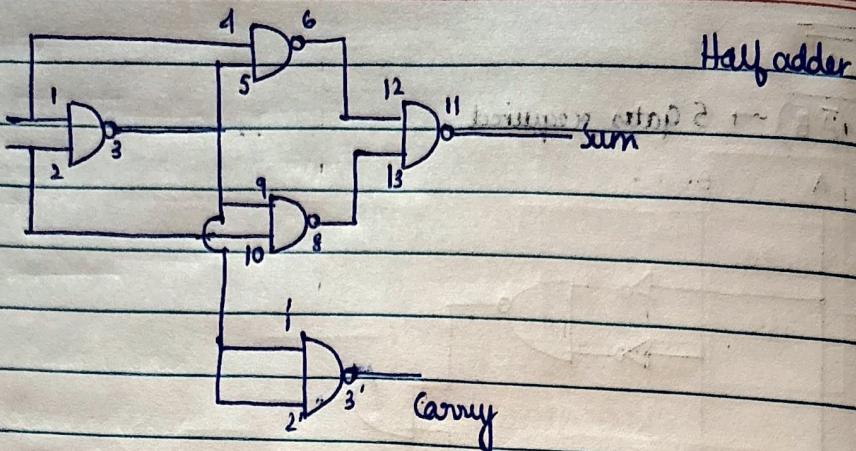
(X)



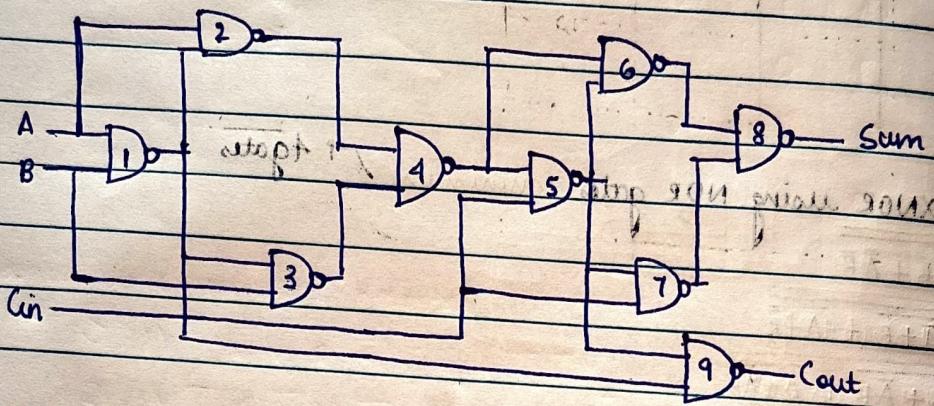
(Q) Realise half adder using only NAND gates. (in column form tabularly)

(Q) Realise full adder using only NAND gates

(1)



(2) Full adder.



COMBINATIONAL LOGIC CIRCUITS:

transmitter part
(1) Multiplexer → acts as a digital switch.

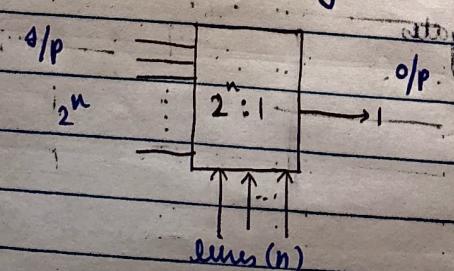
(2) Demux → receiver part of communication

(3) Encoder

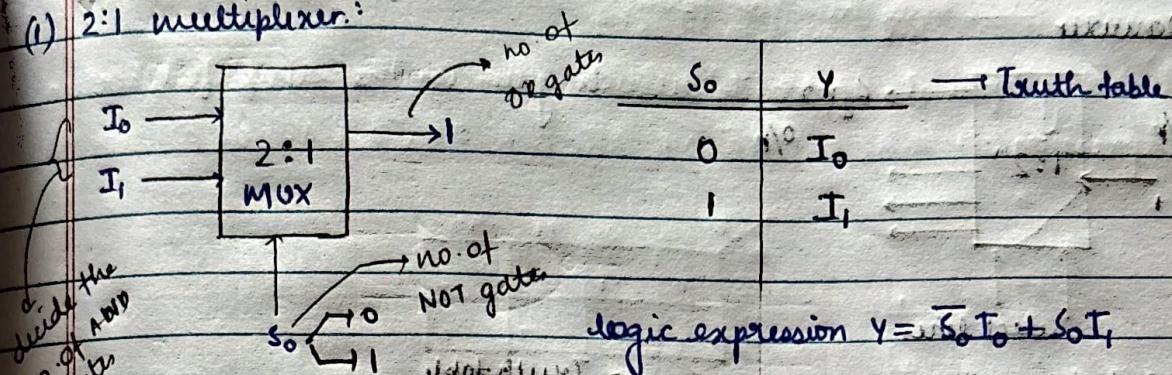
(4) Decoder

(5)

Multiplexer is a digital switch, it allows digital information from several sources to be routed onto a single output line. The selection of particular input is controlled by a set of select lines. Therefore, multiplexer is called many to one and it provides digital equivalent of analog selector switch.

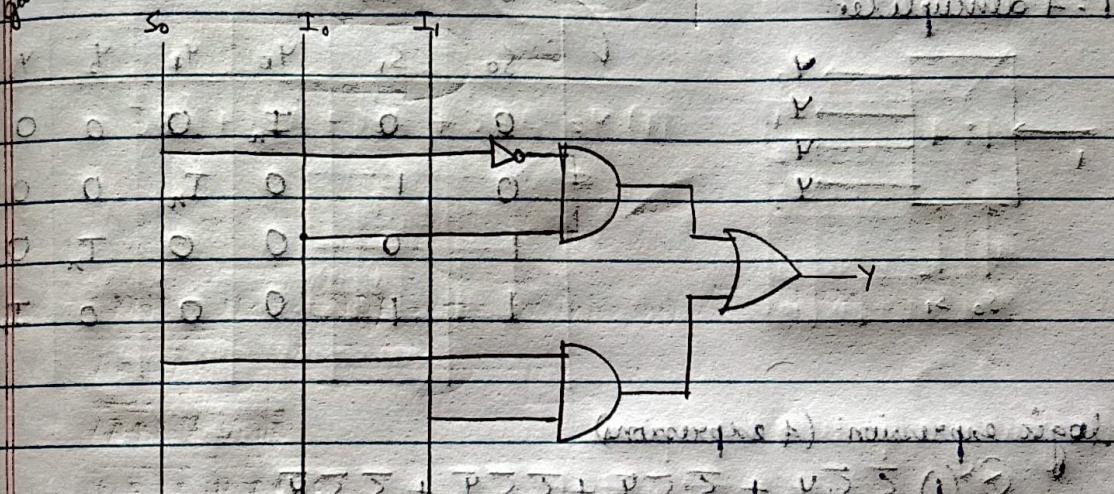


(1) 2:1 multiplexer:

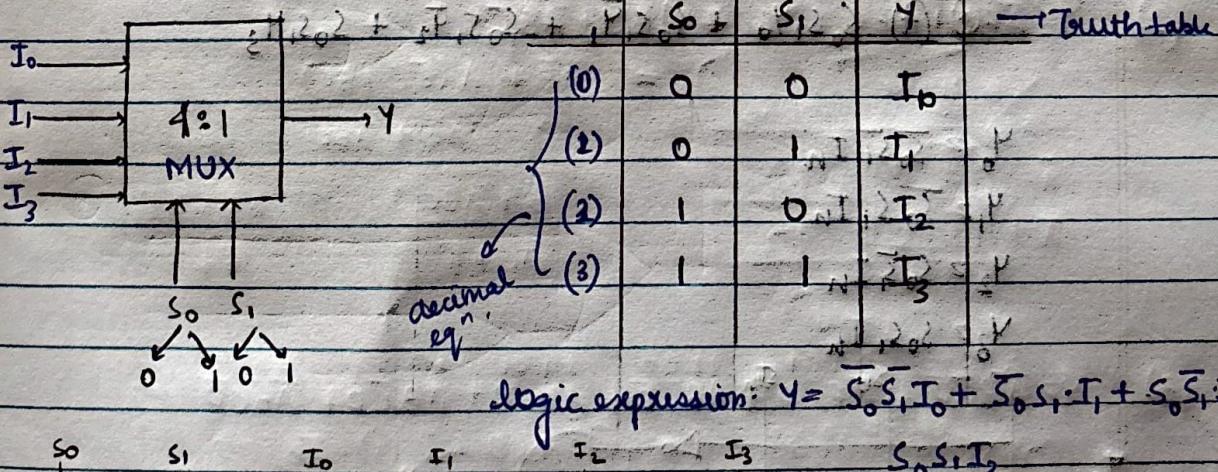


$$\text{logic expression } Y = \bar{S}_0 I_0 + S_0 I_1$$

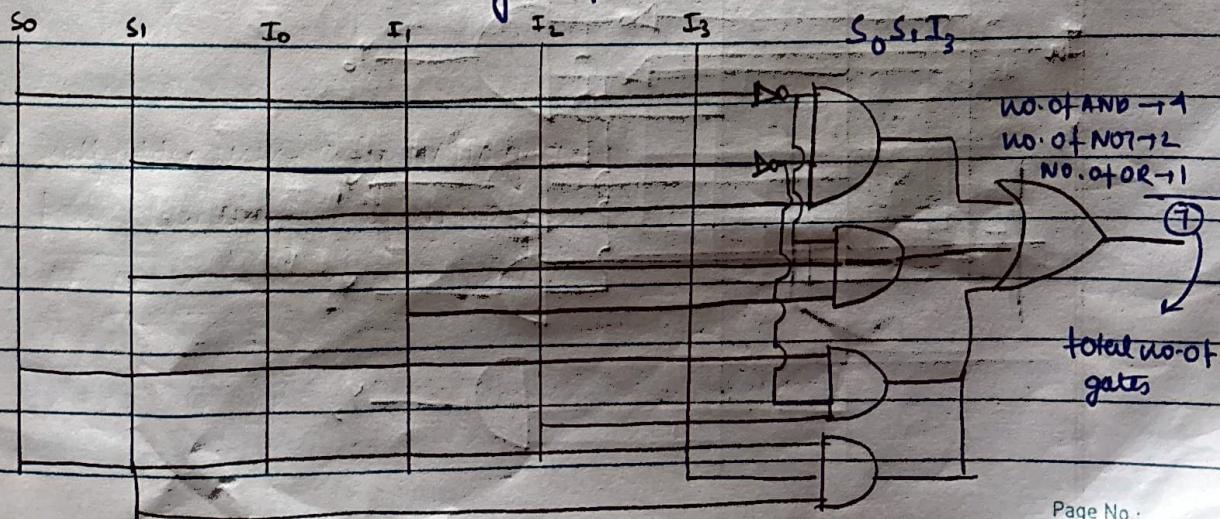
experiments 1:1 (i)



(2) 4:1 multiplexer:

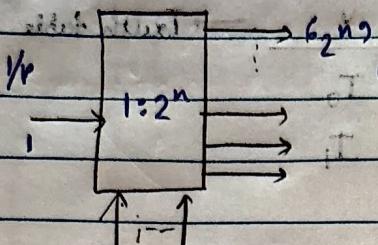


$$\text{logic expression: } Y = \bar{S}_0 \bar{S}_1 I_0 + \bar{S}_0 S_1 I_1 + S_0 \bar{S}_1 I_2 + S_0 S_1 I_3$$



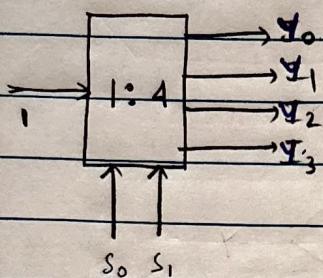
(II)

Demuxer



To select lines Y corresponding to signal

(i) 1:4 demultiplexer.



Truth-table

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

Logic expression: (4 expressions)

$$\textcircled{1} \quad (1) \bar{S}_0 \bar{S}_1 Y_0 + \bar{S}_0 S_1 \bar{Y}_1 + S_0 \bar{S}_1 \bar{Y}_2 + S_0 S_1 \bar{Y}_3$$

$$\textcircled{2} \quad \bar{S}_0 S_1 \bar{Y}_0 + \bar{S}_0 S_1 Y_1 + S_0 \bar{S}_1 \bar{Y}_2 + \bar{S}_0 S_1 \bar{Y}_3$$

$$\textcircled{3} \quad S_0 \bar{S}_1 \bar{Y}_0 + S_0 \bar{S}_1 Y_1 + S_0 S_1 \bar{Y}_2 + S_0 S_1 Y_3$$

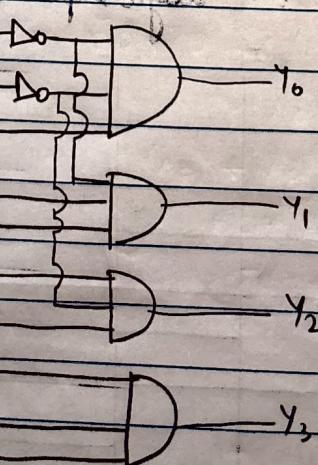
$$\textcircled{4} \quad S_0 S_1 \bar{Y}_0 + S_0 S_1 Y_1 + S_0 \bar{S}_1 \bar{Y}_2 + S_0 \bar{S}_1 Y_3$$

$$Y_0 = \bar{S}_0 \bar{S}_1 I_n$$

$$Y_1 = \bar{S}_0 S_1 I_n$$

$$Y_2 = S_0 \bar{S}_1 I_n$$

$$Y_3 = S_0 S_1 I_n$$



Home

188 → demux.

Tutorial 1

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

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

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

$$\Rightarrow \overline{AB+AC} + A$$

$$\Rightarrow \overline{AB} \cdot \overline{AC} + A$$

$$\Rightarrow AB + AC \cdot \bar{A}$$

$$\Rightarrow A(B+C) \cdot \bar{A} = 0$$

$$(b) Y = AB + \bar{A}\bar{C} + \bar{A}BC (AB + C) \quad | A = Y (1)$$

$$\Rightarrow AB + \bar{A} + \bar{C} + ABC + \bar{ABC}$$

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

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

$$(c) Y_2 = \bar{ABC} + \bar{ABC} + \bar{ABC} + \bar{ABC}$$

$$\Rightarrow \bar{AB} + \bar{ABC} + \bar{ABC}$$

$$\Rightarrow \bar{B}(\bar{A} + \bar{AC}) + \bar{ABC}$$

$$\Rightarrow \bar{B}(\bar{A} + \bar{C}) + \bar{ABC}$$

$$\Rightarrow \bar{AB} + \bar{BC} + \bar{ABC}$$

$$\Rightarrow \bar{AB} + \bar{C}(B + AB)$$

$$\Rightarrow \bar{AB} + \bar{C}(\bar{A} + \bar{B})$$

$$\Rightarrow \bar{AB} + \bar{AC} + \bar{CB}$$

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

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

$$\Rightarrow (\bar{A} \cdot B \cdot \bar{C} + A \cdot B \cdot C + A \cdot \bar{B})$$

$$\Rightarrow (B(\bar{A}C + A\bar{C}) + (A\bar{B})) + 03A + 05\bar{A} + 03\bar{B} + 05\bar{A} \quad (6)$$

$$\Rightarrow (B(1) + A\bar{B})$$

$$\Rightarrow B$$

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

→ Karnaugh

KMAPS: → only have SOP form

It is a graphical approach which gives us the simplified Boolean expression

	A	'2'
B	0	0 1
1	2 3	

	A	'2'
B' C	00 01 11 10	
0 1 3 2	1 5 7 6	

	CD
AB	00 01 11 10
00	0 1 3 2
01	4 5 7 6
11	12 13 15 14
10	8 9 11 10

$$(1) Y = \frac{0}{\bar{A}\bar{B}C} + \frac{0}{\bar{A}BC} + \frac{0}{ABC} + \frac{1}{\bar{A}\bar{B}C} + \frac{1}{\bar{A}BC} + \frac{1}{ABC}$$

sum of product form

$$\Rightarrow A \quad \begin{matrix} BC \\ 00 01 11 10 \end{matrix}$$

Pair $Y = \bar{A}C$

$$\begin{matrix} A \\ 0 \\ 1 \end{matrix} \quad \begin{matrix} 0 \\ 1 \\ 3 \\ 2 \end{matrix}$$

Pair $Y = AB$

If you group two 1 → pair
 " " " four 1 → quartet
 " " " 8 1 → octet

you cannot pair 3 and 7
redundant gap

$$(2) Y = \frac{1}{\bar{A}\bar{B}C} + \frac{1}{\bar{A}\bar{B}C} + \frac{1}{\bar{A}BC} + \frac{1}{ABC} + \frac{1}{\bar{A}\bar{B}C} + \frac{1}{\bar{A}BC}$$

	BC	$\bar{B}C$	$B\bar{C}$	BC
A	0	00 01	11 10	
A	1	00 01	11 10	

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

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

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

F4

$$(3) Y = \frac{0}{\bar{A}\bar{B}\bar{C}\bar{D}} + \frac{0}{\bar{A}\bar{B}\bar{C}D} + \frac{0}{\bar{A}\bar{B}CD} + \frac{1}{\bar{A}\bar{B}C\bar{D}} + \frac{1}{\bar{A}\bar{B}CD} + \frac{1}{\bar{A}\bar{B}C\bar{D}}$$

	AB	$\bar{C}\bar{D}$	$\bar{C}D$	$C\bar{D}$	CD
$\bar{A}\bar{B}$	00	00 01	11 10		
$\bar{A}B$	01	00 01			
AB	11	00 01			
$A\bar{B}$	10		00 01		

$$\Rightarrow \bar{C}\bar{D} + \bar{C}D = \bar{C}(1) \cdot \bar{A}\bar{B}$$

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

$$\Rightarrow \bar{C}(AB) + \bar{A}\bar{B}\bar{C}$$

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

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

$$(1) Y_2 = \overline{A}\overline{B}CD + A\overline{B}CD + \overline{ABC}\overline{D} + ABC\overline{D} + \overline{ABC}\overline{D} + A\overline{BC}\overline{D} + \overline{ABC}\overline{D} + \overline{ABC}\overline{D}$$

	CD	00	01	11	10
AB	00	01	11	10	
00	1	1	1	1	
01	1	1	1	1	
11	1	1	1	1	
10	1	1	1	1	

→ four corners will make a quadrangle

→ make another quadrant

$$(2) Y_2 = \overline{ABC}\overline{D} + A\overline{BC}\overline{D} + \overline{ABC}D + \overline{ABC}\overline{D} + A\overline{BC}\overline{D} + A\overline{BC}D + ABC\overline{D}$$

	CD	00	01	11	10
AB	00				
00					
01		1	1	1	1
11	1	1	1	1	1
10	1	1	1	1	1

SOF → m; POS → M

$$\Rightarrow f(A, B, C, D) = \sum m(0, 1, 4, 8, 9, 10)$$

	CD	00	01	11	10
AB	00	1	1		
00					
01		1			
11					
10		1	1	1	1

$$\overline{ACD} + ABD + \overline{BC}$$

$$\overline{BC} + \overline{ACD} + \overline{DAB}$$

$$(3) ny(w, x, y, z) = \sum m(0, 2, 6, 7, 8, 10, 14, 15)$$

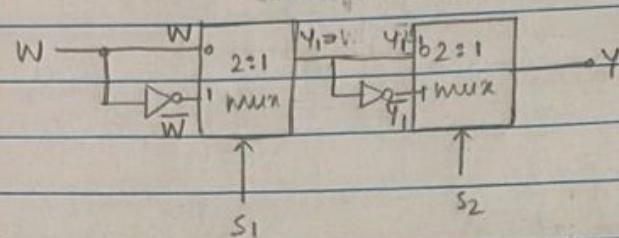
	CD	00	01	11	10
AB	00	1	2	1	2
00		1	2	1	2
01		1	5	1	1
11		1	1	1	1
10		1	9	10	10

(Q) $y = \sum m(0, 1, 4, 5, 7, 10, 11, 13, 14, 15)$

$$y = \sum m(4, 6, 12, 13, 14, 5, 7, 10)$$

$$Y = \overline{A}\overline{B}\overline{C}D + \overline{A}\overline{B}CD + \overline{ABC}D + ABC\overline{D} + A\overline{B}\overline{C}D + A\overline{B}CD + A\overline{BC}D + ABCD$$

(Q)



$$Y = (\overline{W}\overline{S}_1 + \overline{W}S_1)\overline{S}_2 + (\overline{W}\overline{S}_1 + \overline{W}S_1) \cdot S_2$$

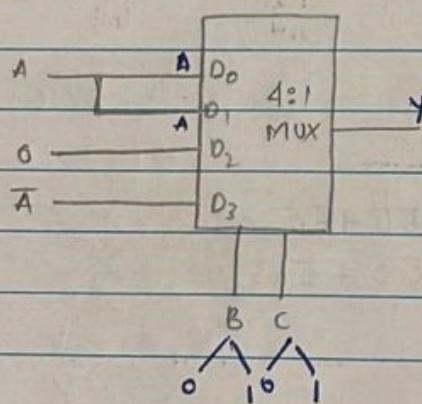
$$\Rightarrow (\overline{W}\overline{S}_1 + \overline{W}S_1)\overline{S}_2 + (\overline{W} + \overline{S}_1) \cdot S_2$$

$$\Rightarrow \overline{W}\overline{S}_1\overline{S}_2 + \overline{W}S_1\overline{S}_2 + (\overline{W} + S_1)S_2$$

$$= \overline{W}\overline{S}_1\overline{S}_2 + \overline{W}S_1\overline{S}_2 + \overline{W}S_2 + S_1S_2 + WS_2 + \overline{S}_1S_2$$

$$= W \oplus S_1 \oplus S_2$$

(Q)



$$Y = \overline{ABC} + A\overline{BC} + \dots$$

$$\Rightarrow (\overline{AB}C + \overline{ABC}) + (\overline{A}BC + ABC)$$

$$\Rightarrow \overline{AB} + \overline{ABC}$$

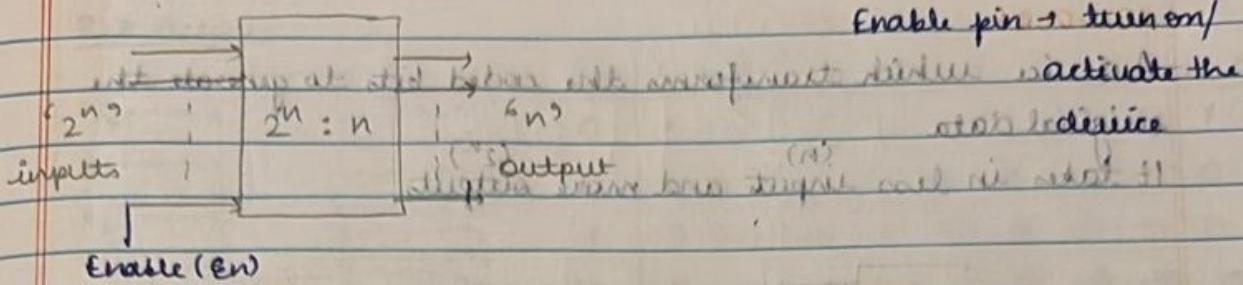
$$(A + C)$$

ENCODER:

It is a device which transforms information/message signal to binary format or coded format.

It takes in multiple input and gives multiple output, converts any complex signal to simplest form.

It converts one binary input to another.



Encoder will get activated when enable input is high (1)

4:2 Encoder:

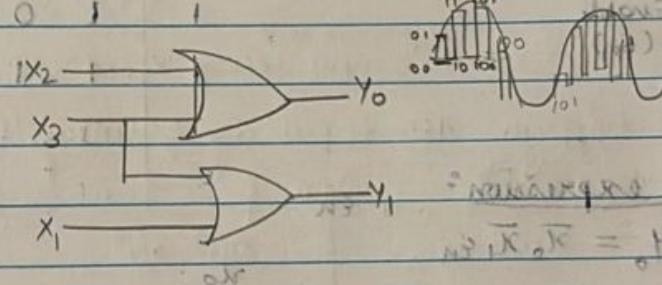
En	x_0	x_1	x_2	x_3	y_0	y_1
0	x	x	x	x	0	0
1	1	0	0	0	0	0
0	0	1	0	0	1	0
1	0	0	1	0	1	1
0	0	0	0	1	1	0
1	0	0	0	0	1	1
0	0	0	1	0	1	0
1	0	0	0	1	0	1

$x \rightarrow$ don't care (either 0 or 1)

Logic expression:

$$y_0 = (x_2 + x_3)$$

$$y_1 = (x_1 + x_3)$$



8:3 Encoder (Octal into binary converter)

En	x_0	x_1	x_2	x_3	x_4	x_5	x_6	x_7	y_0	y_1
0	x	x	x	x	x	x	x	x	00	00
1	1	0	0	0	0	0	0	0	00	00
1	0	1	0	0	0	0	0	0	00	1
1	0	0	1	0	0	0	0	0	01	0
1	0	0	0	1	0	0	0	0	01	1
1	0	0	0	0	1	0	0	0	100	0
1	0	0	0	0	0	1	0	0	101	1
1	0	0	0	0	0	0	1	0	110	0
1	0	0	0	0	0	0	0	1	111	1

$x \rightarrow$ don't care

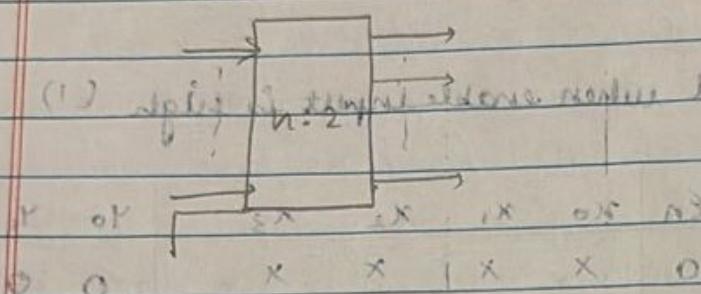
$$y_0 = (x_4 + x_5 + x_6 + x_7); y_1 = (x_2 + x_3 + x_6 + x_7)$$

$$y_2 = (x_1 + x_3 + x_5 + x_7)$$

DECODER

A device which transforms the coded bits to generate the original data.

It takes in less inputs and more outputs.



2:4 Decoder

	0	0	1	0	Y ₀	Y ₁	Y ₂	Y ₃
X ₀	0	1	0	1	Y ₀	Y ₁	Y ₂	Y ₃
X ₁	1	0	1	0	0	0	0	0
					0	0	0	0
					1	0	0	0
					0	1	0	0
					1	0	0	0
					1	1	0	0
Enable (En)					0	0	0	0

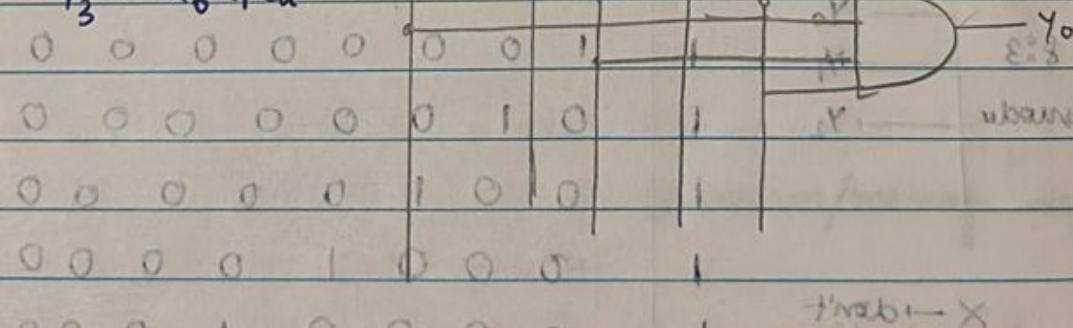
Logic expression:

$$Y_0 = \bar{X}_0 \bar{X}_1 En$$

$$Y_1 = \bar{X}_0 X_1 En$$

$$Y_2 = X_0 \bar{X}_1 En$$

$$Y_3 = X_0 X_1 En$$



$$(X_0 + X_1 + X_0 X_1) = Y_0 \cdot (X_0 + X_1 + X_0 X_1 + X_1 X_0) = Y_0$$

3:8 decoder

	x_0	x_1	x_2	En	x_0	x_1	x_2	y_0	y_1	y_2	y_3	y_4	y_5	y_6	y_7
$x_0 \rightarrow$				0	x	x	x	x	x	x	x	x	x	x	x
$x_1 \rightarrow$				1	0	0	0	1	0	0	0	0	0	0	0
$x_2 \rightarrow$					0	0	1	0	1	0	0	0	0	0	0
outputs enable is binary as follows					0	1	0	0	0	0	1	0	0	0	0

Enable

(En)

X → don't care

Input 0 1 1 0 0 0 0 0 1 0 0 0 0 0 0 0

atmosphere without enable (either off) 1 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0

no intermediate variables, 1 0 1 0 0 0 0 0 0 0 0 0 0 1 0 0

no other effects like thresholding etc 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0

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

tuple pattern

unique entry w/ v

v → pattern learned

v, v has no effect, t

INT
phase

tuple pattern non

v → pattern learned

(C, R, V) pattern tuple exist

behaviour - v = v or pattern (tuple v is present) : 1 + 1A

$$\text{no } 1A = \text{no } v$$

$$01 - 01$$

$$10 - 00 = 10$$

$$(C, R, V) A = \text{no } v$$

only what finds match $\leftarrow SCA + (CC \rightarrow) A^c$

behaviour - v

$$10A = \text{no } v$$

$$11 - 10 = 10$$

$$(C, R) A = \text{no } v$$