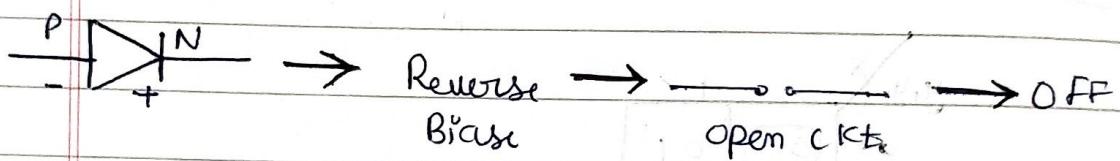
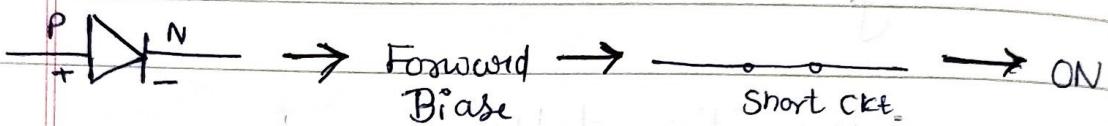


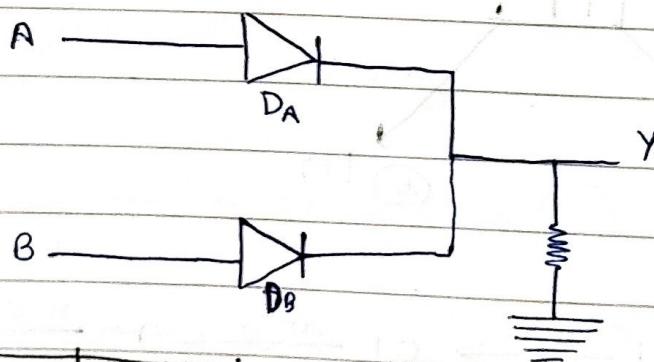
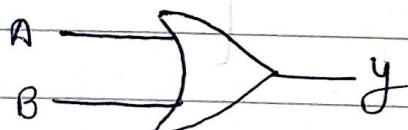
## → UNIT - 5

### → Concept of logic Families

» In Case of forward bias Diode is ON and in case of reverse bias Diode is OFF.

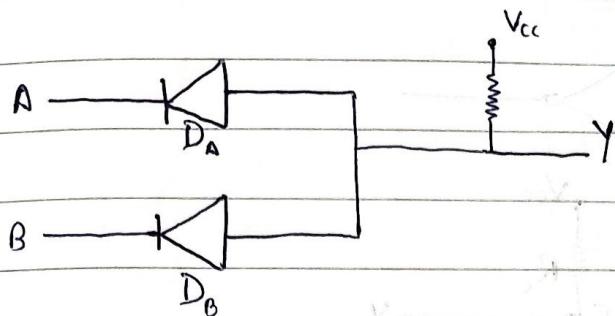
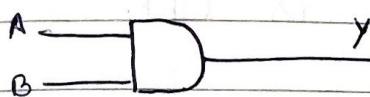


#### (1) OR-Gate



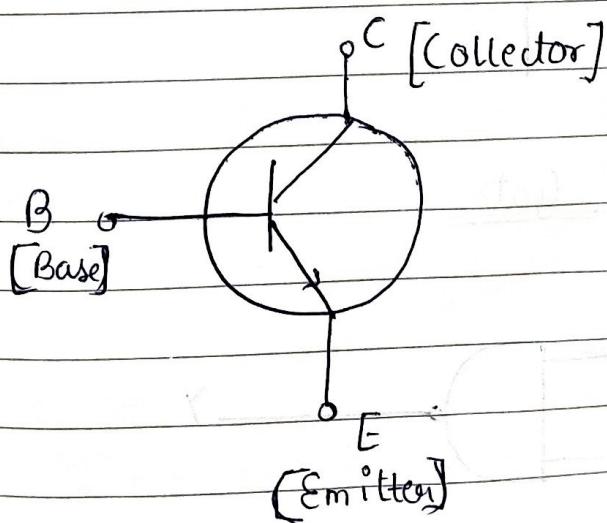
A	B	D <sub>A</sub>	D <sub>B</sub>	Y
0	0	OFF	OFF	0
0	1	OFF	ON	1
1	0	ON	OFF	1
1	1	ON	ON	1

} OR Gate

(2) AND Gate

A	B	$D_A$	$D_B$	Y
0	0	ON	ON	0
0	1	OFF	ON	0
1	0	OFF	ON	0
1	1	OFF	OFF	1

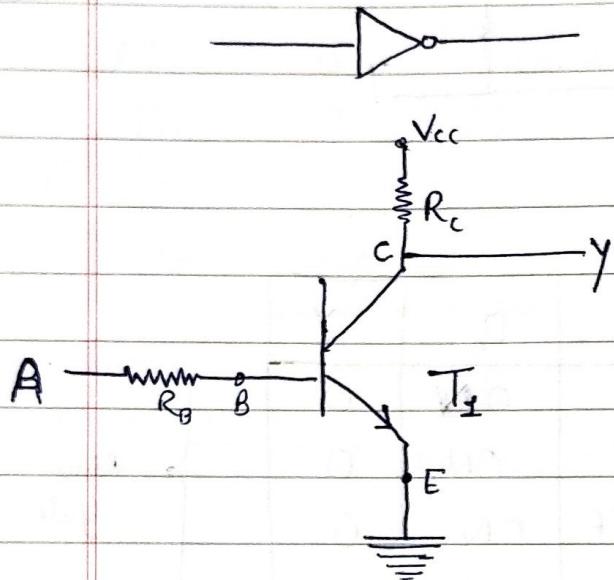
} AND  
Gate

# BJT (Bi-polar junction Transistor)

» When terminal "B" is at high logic (1) than transistor is ON.

» When terminal "B" is at low logic (0) than transistor is OFF.

### <1> NOT Gate

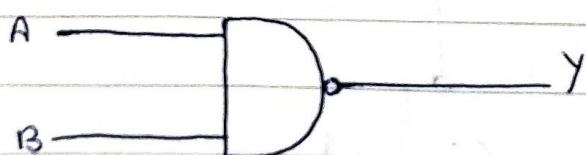


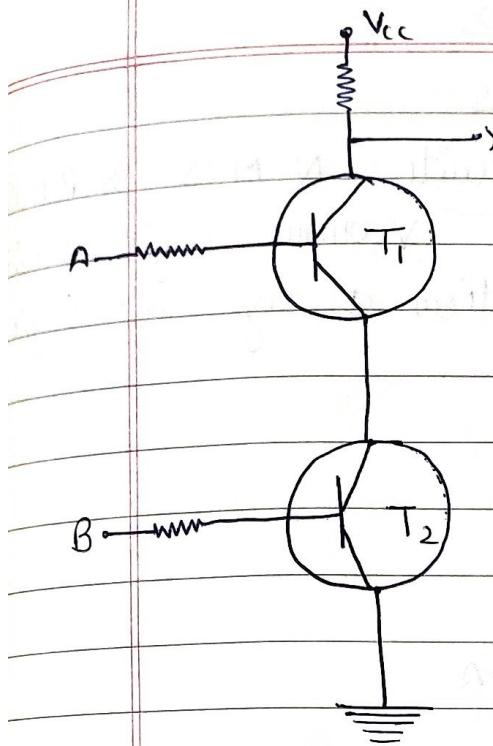
A	T.	Y
0	OFF	1
1	ON	0

} NOT Gate

### <2>

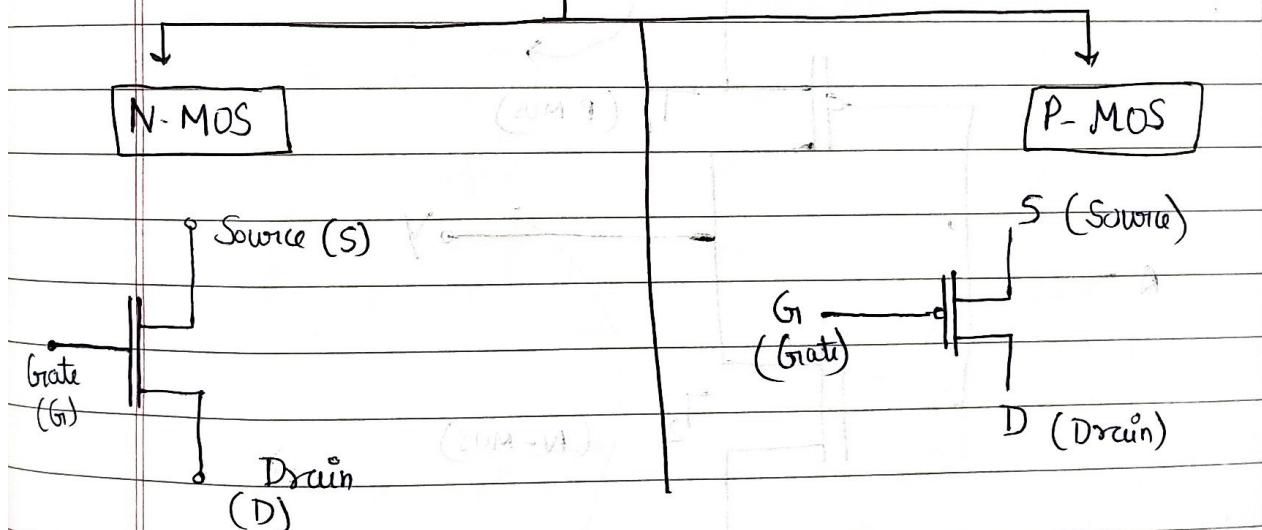
### NAND Gate



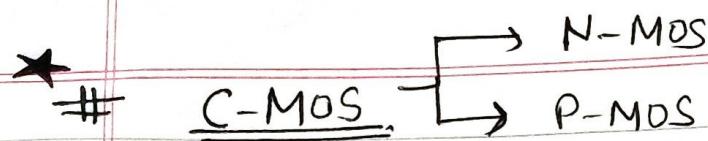


A	B	T <sub>1</sub>	T <sub>2</sub>	Y
0	0	OFF	OFF	0
0	1	OFF	ON	1
1	0	ON	OFF	1
1	1	ON	ON	0

#

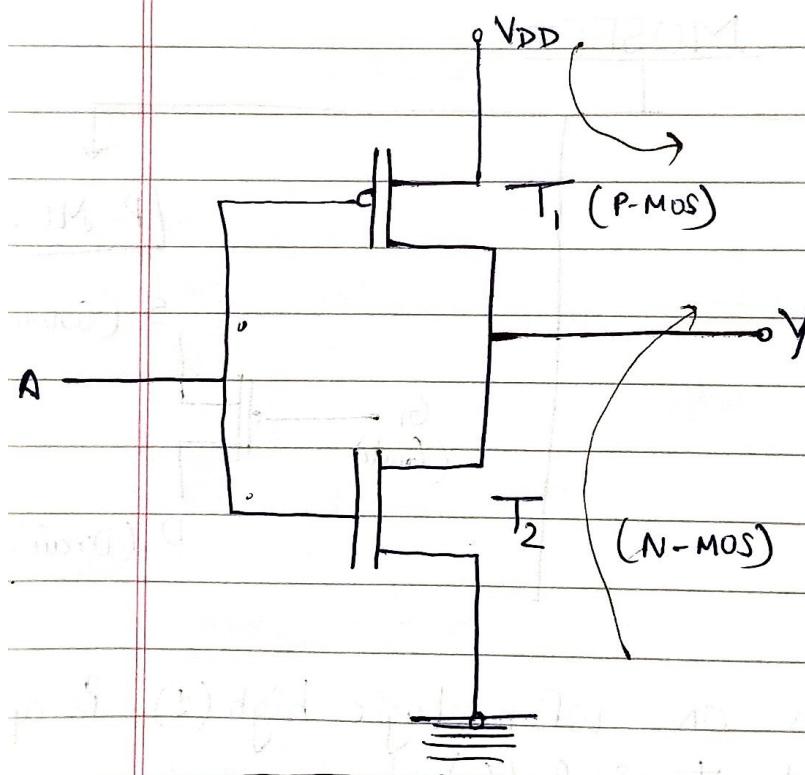
MOSFET

- N  
MOS
 } ➤ N-MOS is ON when logic high (1) is applied at Gate terminal (G<sub>1</sub>).  
 ➤ N-MOS is OFF when logic low (0) is applied at Gate terminal (G<sub>1</sub>).
- P  
MOS
 } ➤ P-MOS is OFF when logic high (1) is applied at Gate terminal (G<sub>1</sub>).  
 ➤ P-MOS is ~~ON~~ OFF when logic low (0) is applied at Gate terminal (G<sub>1</sub>).



- » In C-MOS logic circuits N-MOS & P-MOS are used to make various logic circuit.
- » Lowest power dissipation among all logic family is C-MOS.

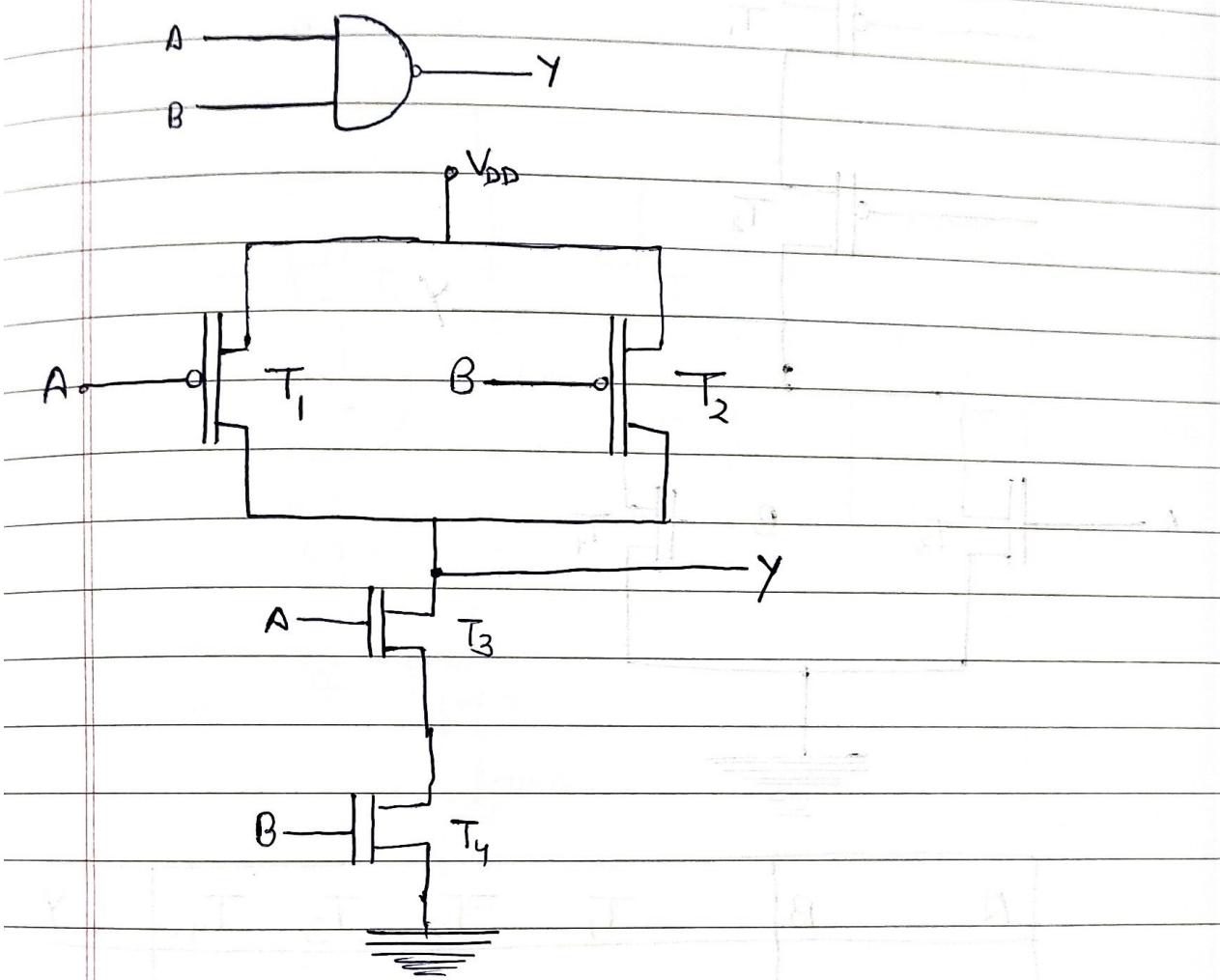
### ① NOT Gate



A	T <sub>1</sub>	T <sub>2</sub>	Y
0	ON	OFF	1
1	OFF	ON	0

} NOT  
Gate.

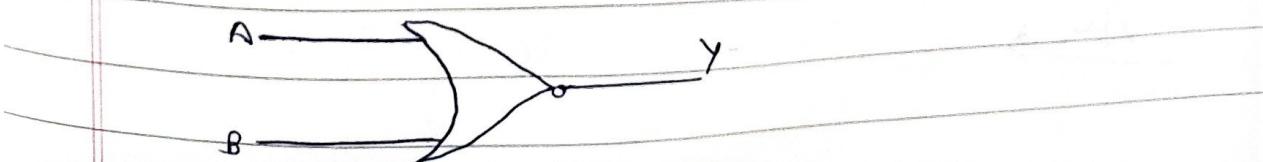
② NAND Gate

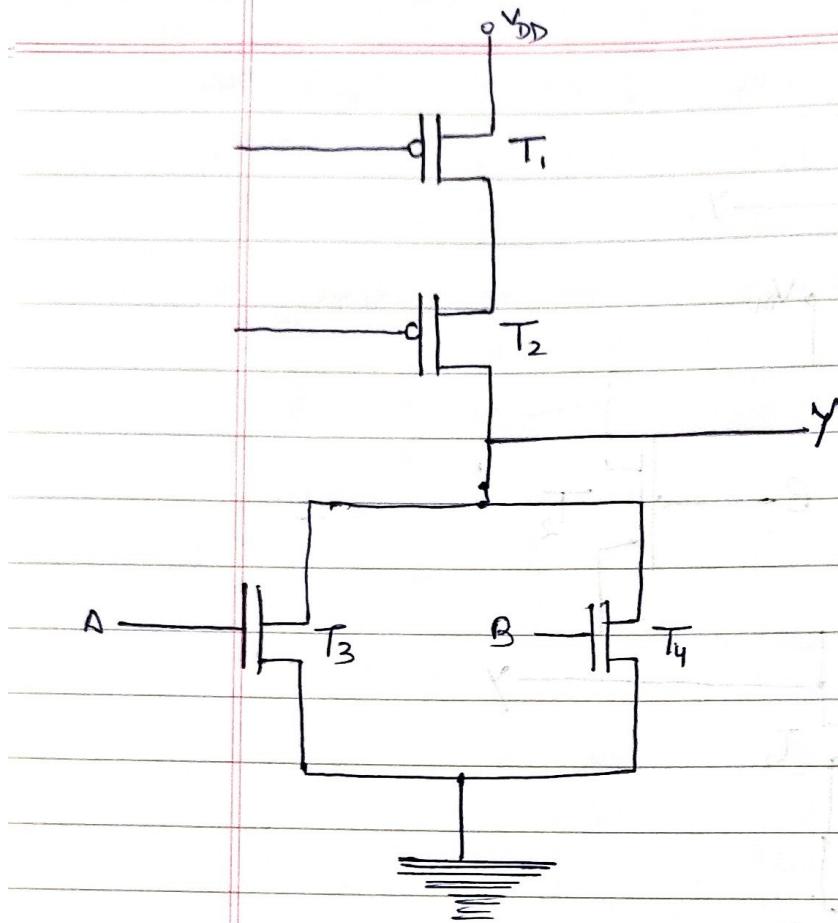


A	B	T <sub>1</sub>	T <sub>2</sub>	T <sub>3</sub>	T <sub>4</sub>	Y
0	0	ON	ON	OFF	OFF	1
0	1	ON	OFF	OFF	ON	1
1	0	OFF	ON	ON	OFF	1
1	1	OFF	OFF	ON	ON	0

} NAND gate

③ NOR - Gate





A	B	T <sub>1</sub>	T <sub>2</sub>	T <sub>3</sub>	T <sub>4</sub>	y
0	0	ON	ON	OFF	OFF	1
0	1	ON	OFF	OFF	ON	0
1	0	OFF	ON	ON	OFF	0
1	1	OFF	OFF	ON	ON	0

Ques. :- Implement two input AND and OR Gate by using C-MOS?

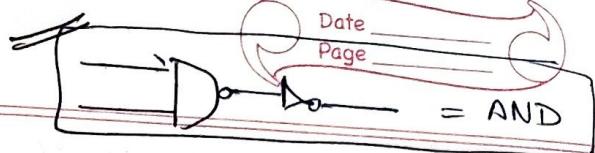
Ans. →

$\text{AND} = \text{NAND} + \text{NOT}$

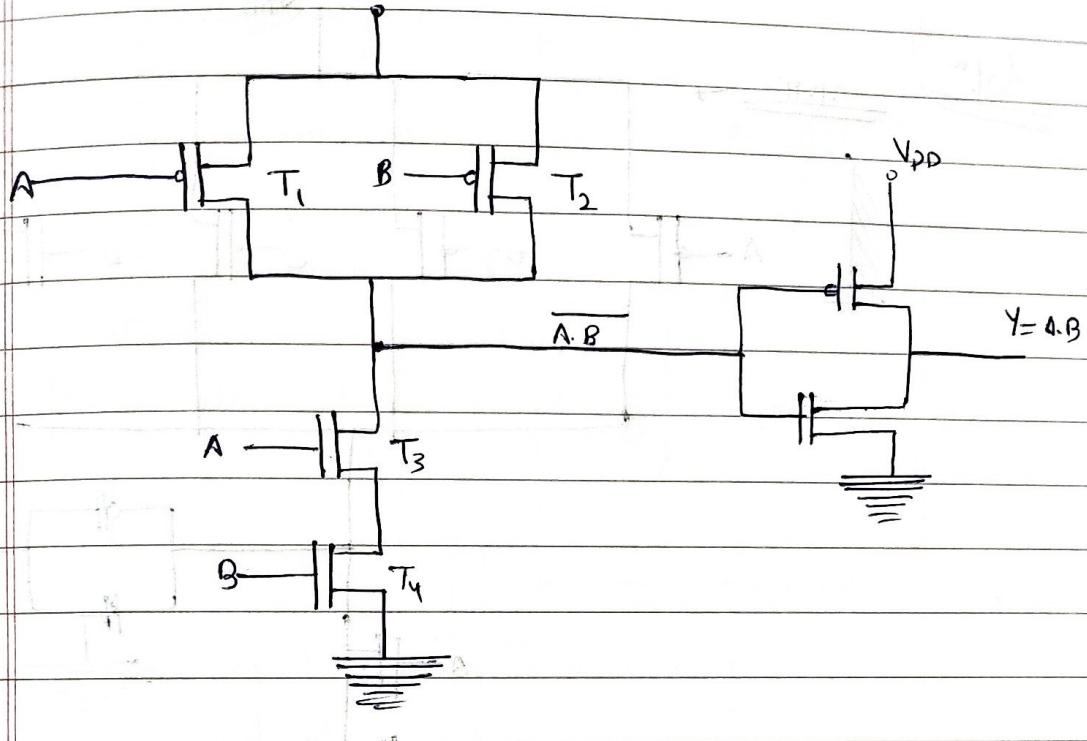
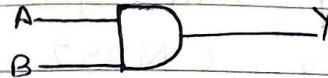
classmate

Date \_\_\_\_\_

Page \_\_\_\_\_

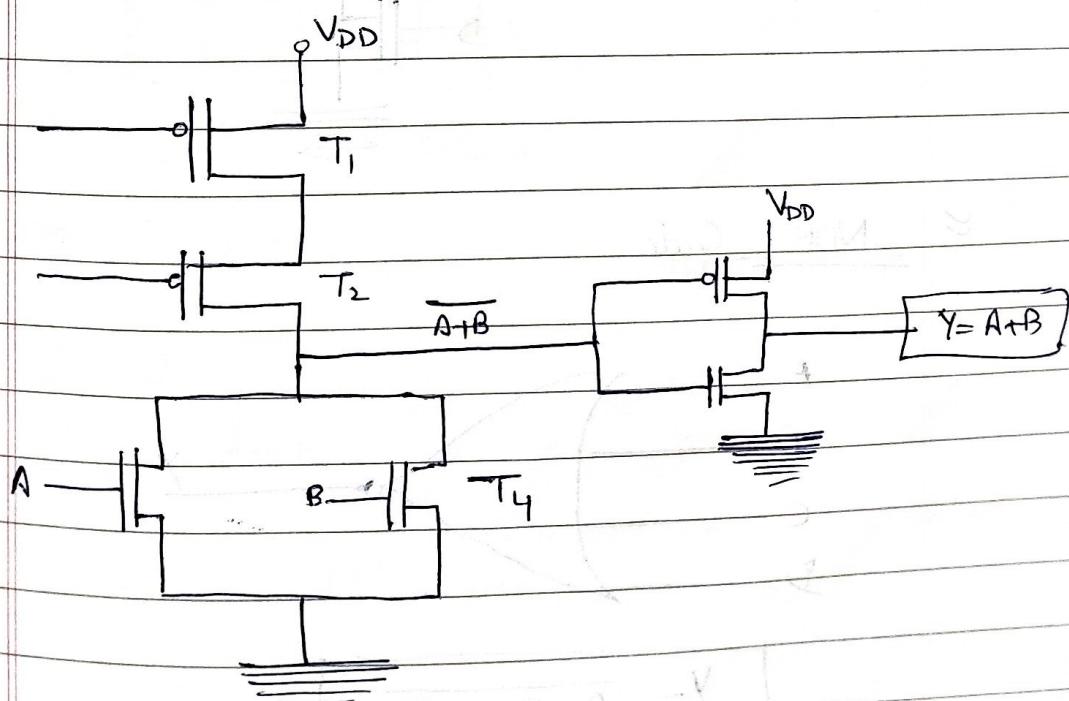


AND



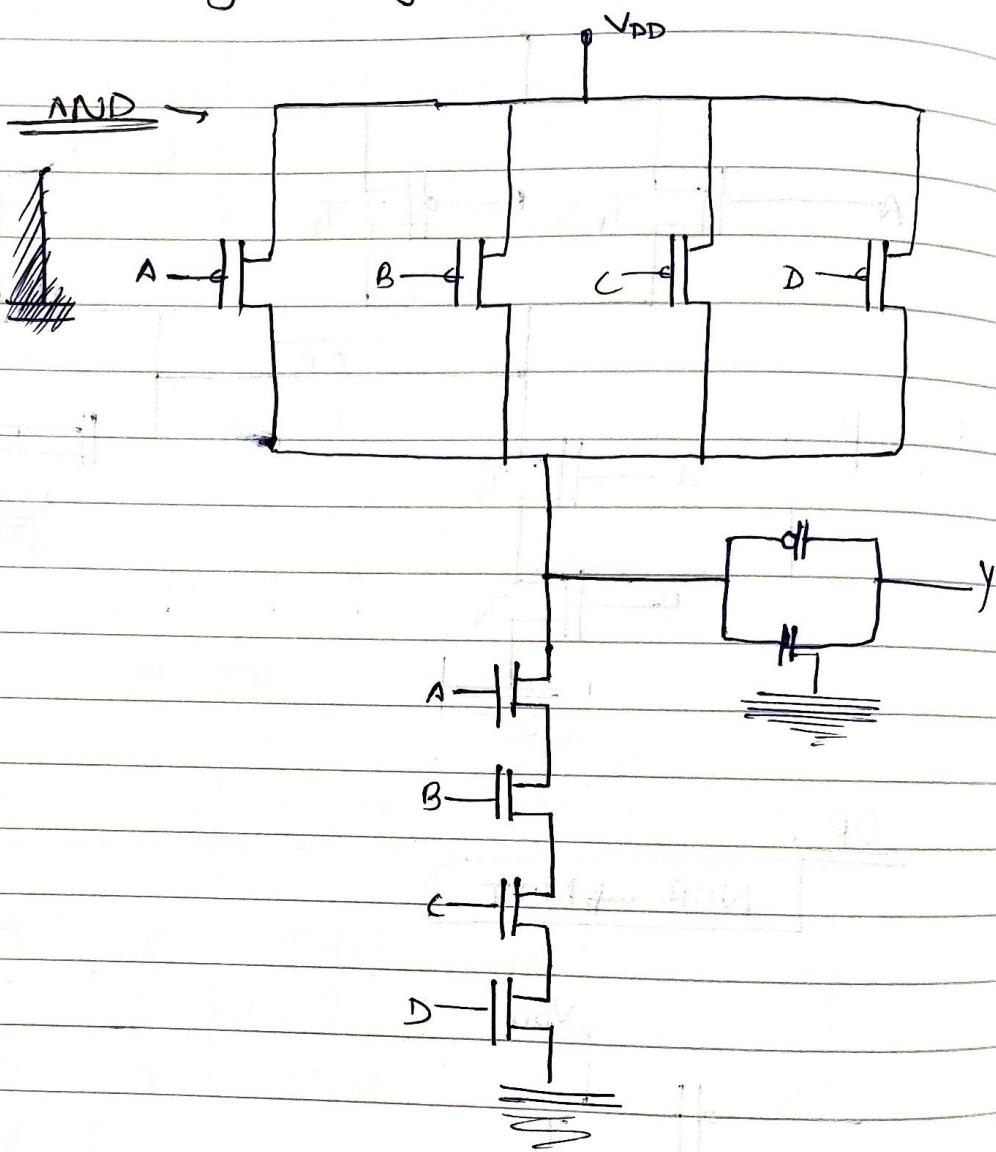
OR.

NOR  $\rightarrow$  NOT

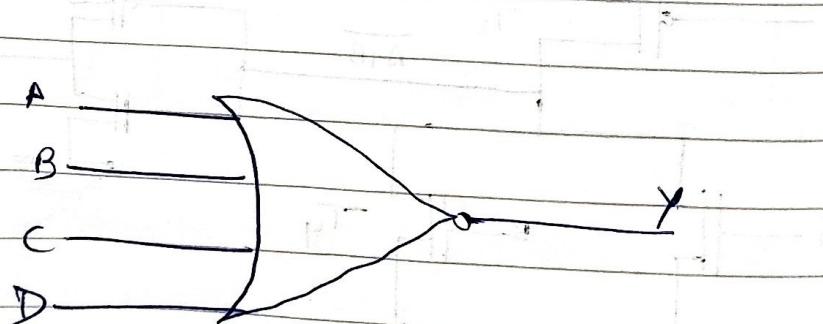


Ques: Implement 4 i/p NAND gate, NOR, AND, OR by using C-MOS?

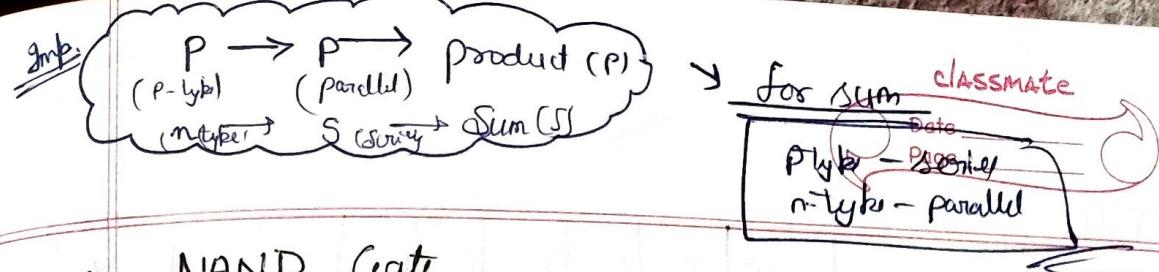
Soln



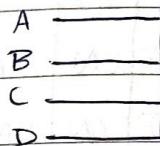
∴ NOR Gate



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

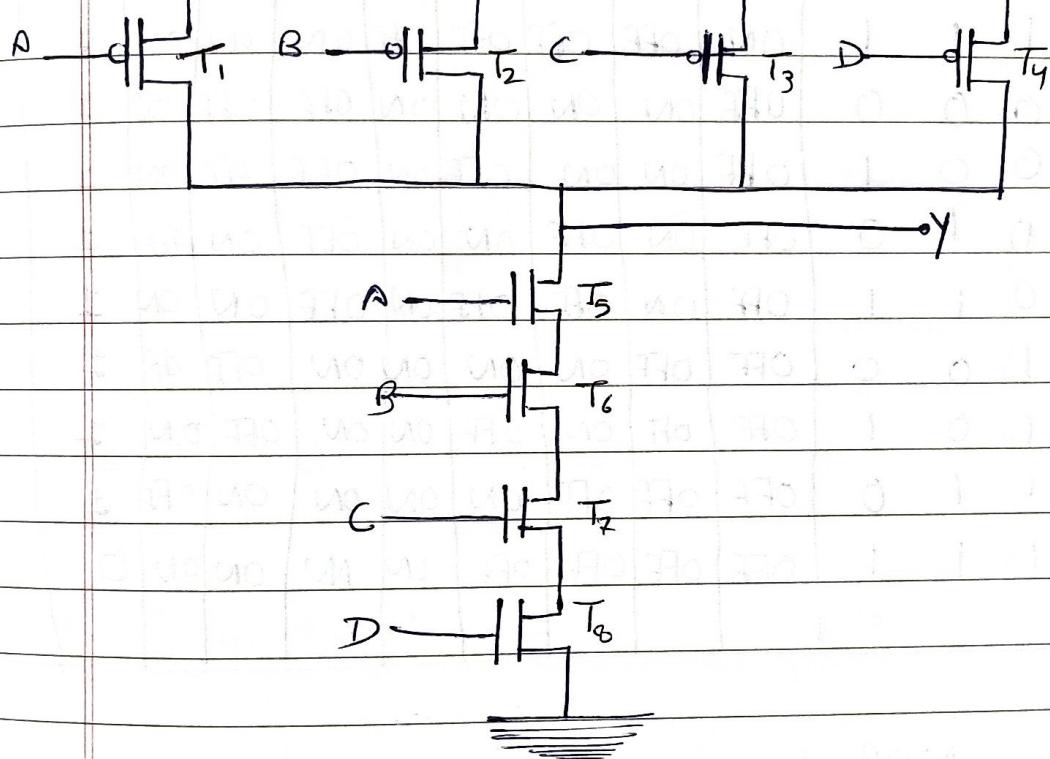


## NAND Gate



Y

$$Y = \overline{ABCD}$$



A B C D

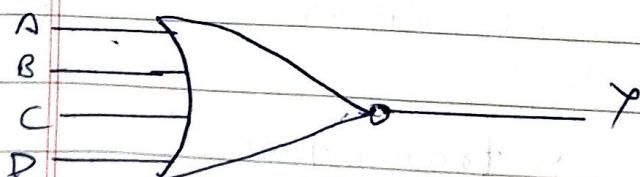
Table → 8 transistors

1 o/p

4 i/p

Table

A	B	C D	T <sub>1</sub>	T <sub>2</sub>	T <sub>3</sub>	T <sub>4</sub>	T <sub>5</sub>	T <sub>6</sub>	T <sub>7</sub>	T <sub>8</sub>	Y
0	0	0 0	ON	ON	ON	ON	OFF	OFF	OFF	OFF	1
0	0	0 1	ON	ON	ON	OFF	OFF	OFF	OFF	ON	1
0	0	1 0	ON	ON	OFF	ON	OFF	OFF	ON	OFF	1
0	0	1 1	ON	ON	OFF	OFF	OFF	OFF	ON	ON	1
0	1	0 0	ON	OFF	ON	ON	OFF	ON	OFF	OFF	1
0	1	0 1	ON	OFF	ON	OFF	OFF	ON	OFF	ON	1
0	1	1 0	ON	OFF	OFF	ON	OFF	ON	ON	OFF	1
0	1	1 1	ON	OFF	OFF	OFF	OFF	ON	ON	ON	1
1	0	0 0	OFF	ON	ON	ON	ON	OFF	OFF	OFF	1
1	0	0 1	OFF	ON	ON	OFF	ON	OFF	OFF	ON	1
1	0	1 0	OFF	ON	OFF	ON	ON	OFF	ON	OFF	1
1	0	1 1	OFF	ON	OFF	OFF	ON	OFF	ON	ON	1
1	1	0 0	OFF	OFF	ON	ON	ON	ON	OFF	OFF	1
1	1	0 1	OFF	OFF	ON	OFF	ON	ON	OFF	ON	1
1	1	1 0	OFF	OFF	OFF	ON	ON	ON	ON	OFF	1
1	1	1 1	OFF	OFF	OFF	OFF	ON	ON	ON	ON	0

NOR

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

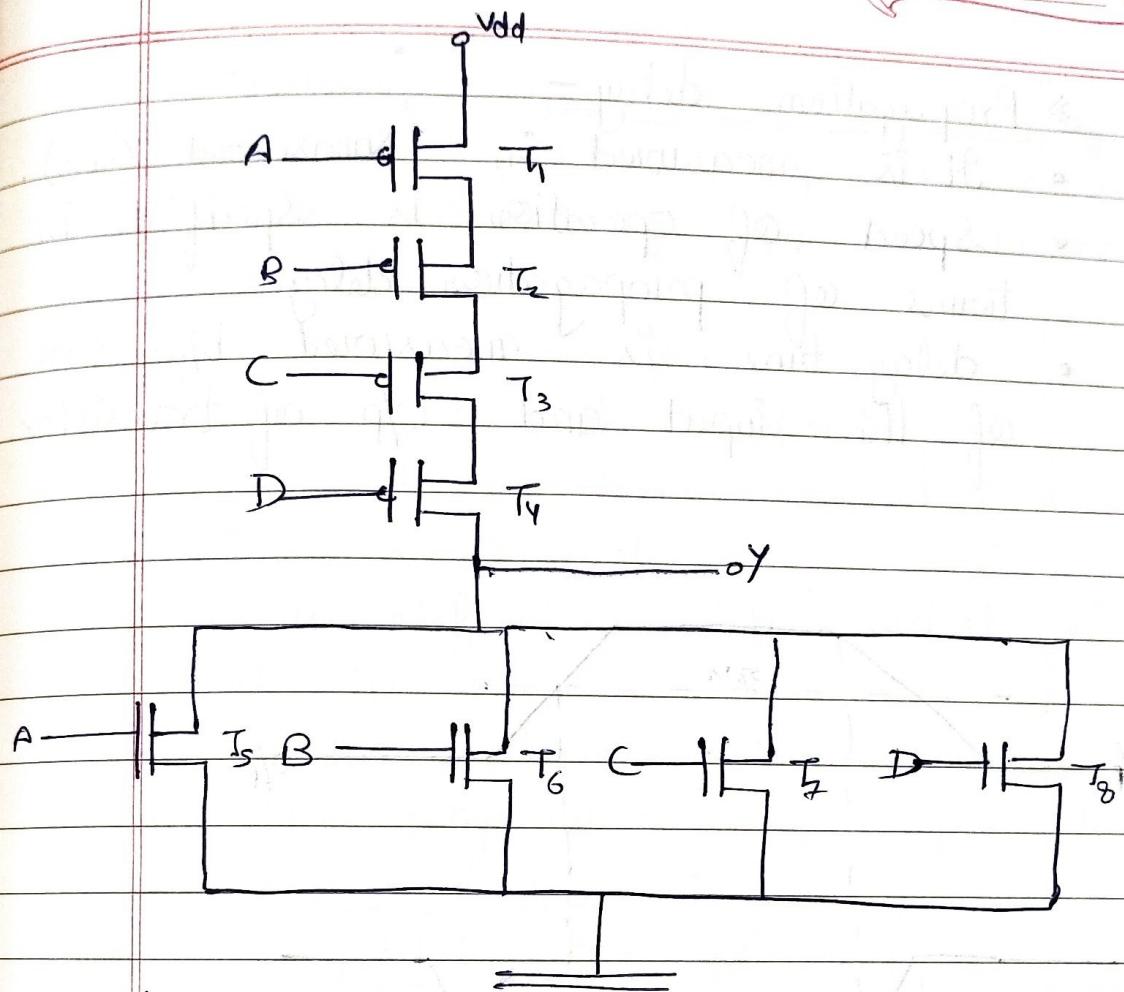


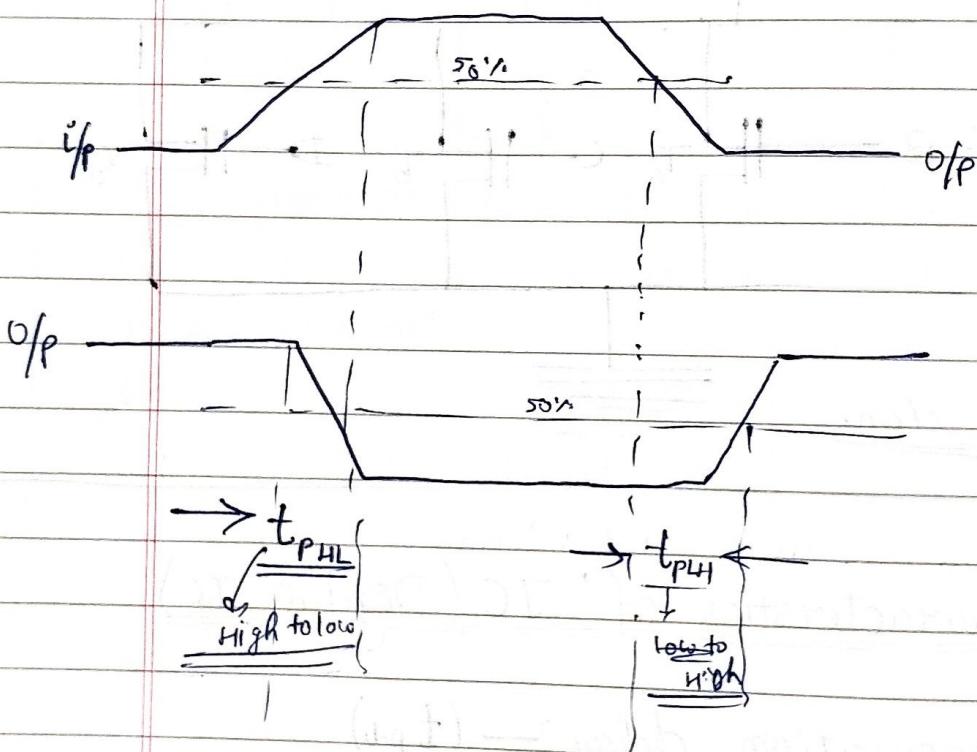
Table  $\rightarrow$  done.

## # Characteristics of JC (Digital IC)

- 1 > Propagation delay - (t<sub>pd</sub>)
- 2 > Power dissipation (P<sub>diss</sub>)
- 3 > figure of merit (FOM)
- 4 > Fan out
- 5 > Fan in
- 6 > Noise Margin (NM)
- 7 > operating temperature

### » Propagation delay -

- It is measured in Nanosecond (ns).
- Speed of operation is specified in terms of propagation delay.
- delay time is measured b/w 50% of the input and o/p of transistor.



$$\Rightarrow \text{Avg. Propagation delay} = \frac{t_{PHU} + t_{PHL}}{2} \text{ (ns)}$$

### » Power dissipation (P<sub>diss</sub>)

- It is measured in milliwatt (mw)
- This is the amount of power dissipated in an "IC" in the form of heat.

- excessive temperature can damage the IC so power dissipation is low.
- It is determined by the Current ( $I_{cc}$ ) and Voltage ( $V_{cc}$ ).

$$= P_{cc} = V_{cc} \times I_{cc} \text{ [mW]}$$

» Figure of Merit (FOM)

- It is defined as the product of speed and power.

$$FOM = \text{propagation delay} \times \text{power dissipation}$$

Unit - p-s

or  $10^{-12} \frac{\text{W}}{\text{s}} = \text{Pico}$

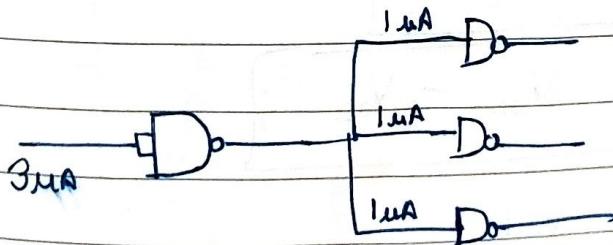
jmp

Both should be lesser. (propagation delay and Power dissipation).

» Fan Out - It gives idea about how many similar gate can be driven by a gate.

- Higher is the fan out more is the current supply capacity of gate.

Eg

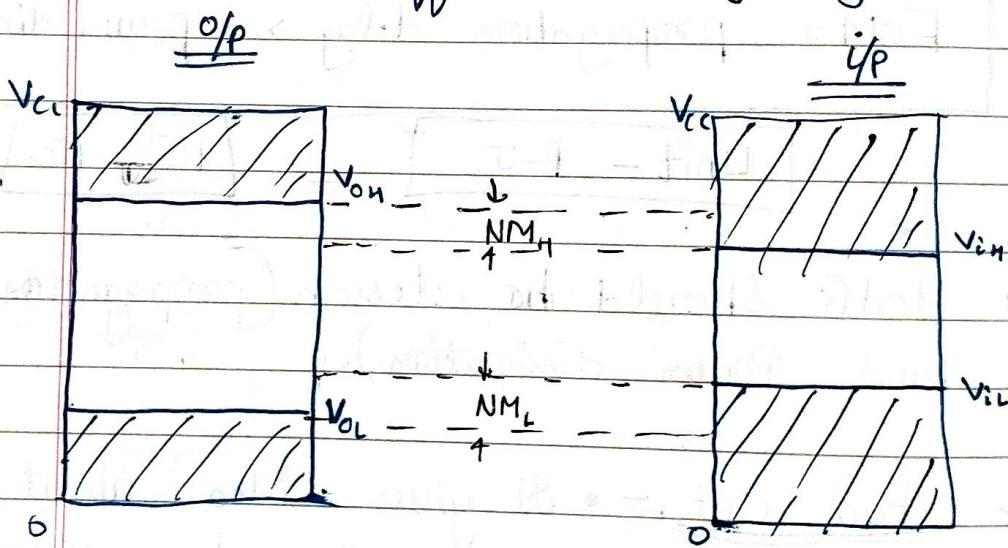


» Fan IN

- Number of i/p that can be connected to a circuit. in another words -
- Fan in is a term that define the maximum no. of digital input that a single logic gate can accept.

» Noise Margin (NM)

- It is the maximum margin that can be added in logic gate Pinput which will not effect the logic gate.



$$\approx [V_{OH} > V_{IH} > V_{IL} > V_{OL}]$$

$$\approx NM_H = V_{OH} - V_{IH}$$

$$\approx NM_L = V_{IL} - V_{OL}$$

Unit = Volt.

### Operating temperature

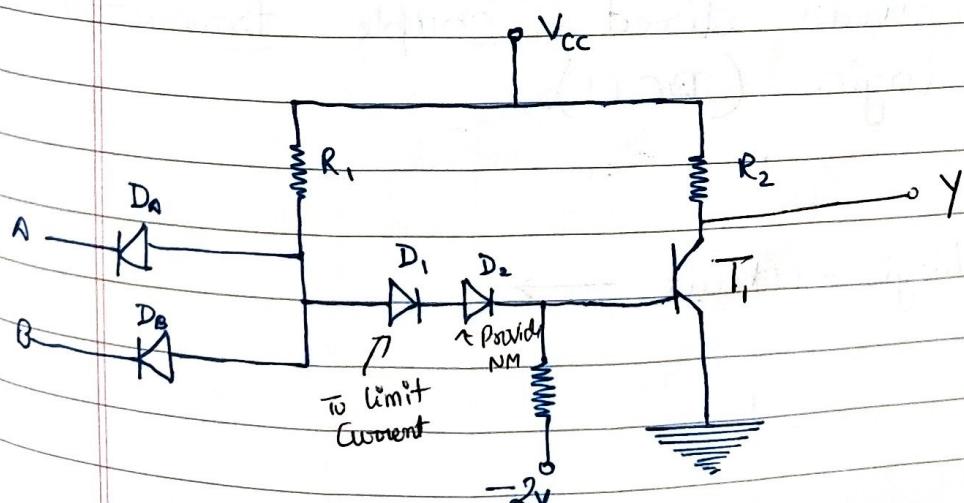
- 0 to  $70^{\circ}\text{C}$  for commercial use.
- 0 to  $85^{\circ}\text{C}$  for industrial application
- $-55^{\circ}\text{C}$  to  $125^{\circ}\text{C}$  for military application.

### # Interfacing of digital logic family —

- ① Diode Transistor Logic (DTL)
- ② Diode Couple Transistor Logic (DCTL)
- ③ <sup>M.I.T.</sup> Transistor Transistor Logic (TTL)
- ④ Emitter Couple Logic (ECL)

(1)

- = DTL :- • Basic gate which is designed using DTL Technology is "NAND" Gate.  
 • This circuit represent two i/p NAND Gate designed using DTL technology if we want n- i/p NAND Gate then, we can take n- number of diode at the input side.



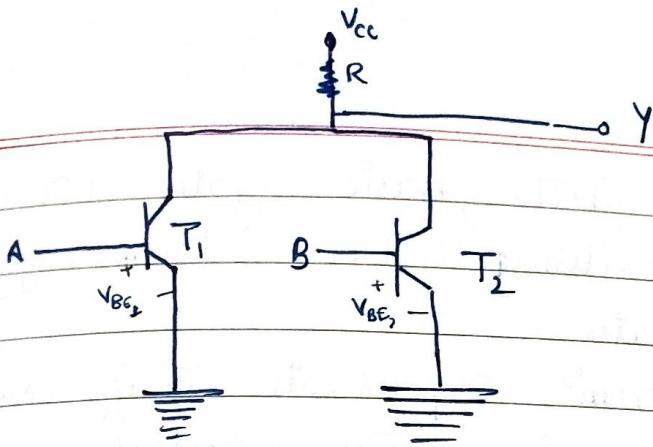
A	B	$D_A$	$D_B$	T	Y	
0	0	ON	ON	OFF	1	
0	1	ON	OFF	OFF	1	
1	0	OFF	ON	OFF	1	
*	1	OFF	OFF	ON	0	

②

DCTL :- Basic gate which is realised using DCTL Technology is NOR Gate.

- This circuit represents two i/p NOR Gate design using DCTL technology.
- if we want n-i/p NOR Gate then we can take n-number of transistor.
- in DCTL i/p has been directly connected with Base of transistor. i.e. we have not used resistance b/w i/p and Base of transistor so because of this direct coupling of i/p with Base of transistor is called direct couple transistor logic (DCTL).

\* logic - circuit →



classmate

Date \_\_\_\_\_

Page \_\_\_\_\_

A	B	T <sub>1</sub>	T <sub>2</sub>	Y
0	0	OFF	OFF	1
0	1	OFF	ON	0
1	0	ON	OFF	0
1	1	ON	ON	0

NOR  
gate.

OFF

Current Hogging

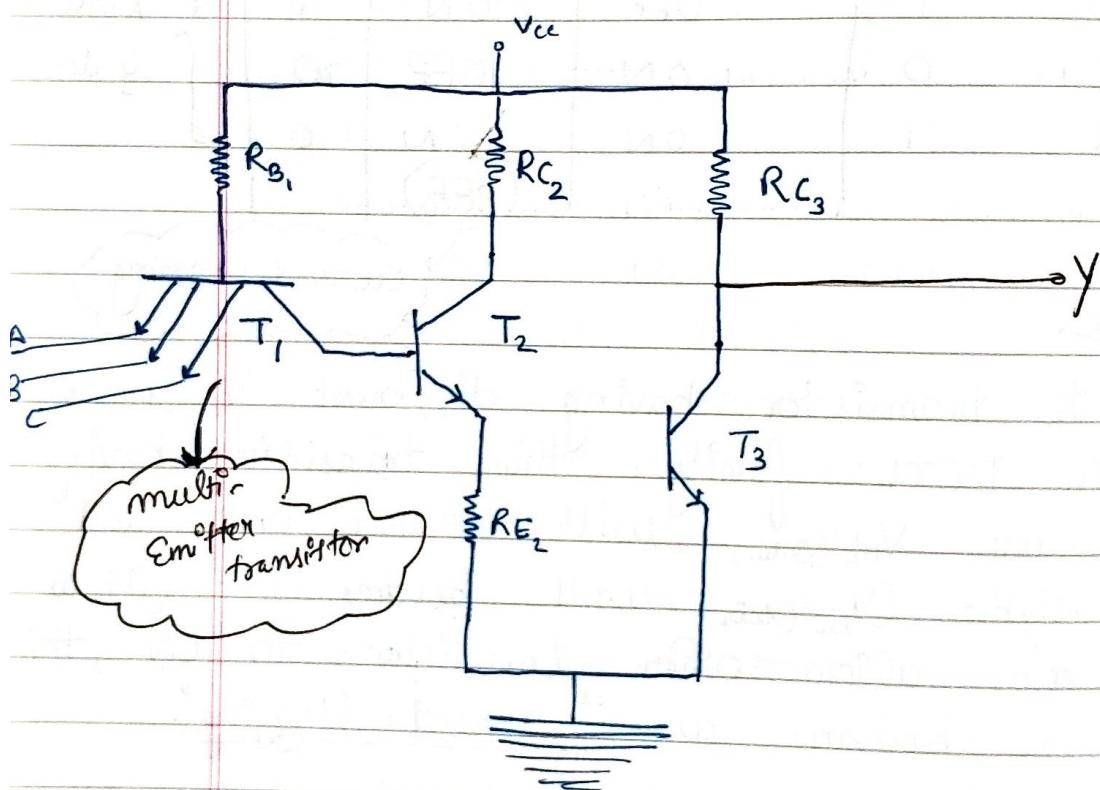
2 Marks (Qmb)

⇒ If transistor having different  $V_{BE}$  (saturation) in DCTL family than transistor having lower  $V_{BE}$  (sat.) will become "ON" and Higher ( $V_{BE}$  (sat.)) will become 'OFF'. (It will not allow other transistor to ON, this is known as "Current Hogging".

③ TTL :- TTL is modified version of DTL.

- In DTL, when diode ( $D_A, D_B$ ) will be replaced by multi-emitter transistor ( $T_1$ ) and diode ( $D_1, D_2$ ) will be replaced by Transistor  $T_2$  and  $T_3$  then we will get TTL.

- As in DTL basic gate which is designed using TTL technology is NAND Gate.
- This circuit represents 3-i/p NAND gate designed using TTL technology, if we want n-i/p NAND gate then we will have to use multi-emitter transistor having n-emitter terminal.



A	B	C	T <sub>1</sub>	T <sub>2</sub>	T <sub>3</sub>	Y
0	0	0	OFF	OFF	OFF	1
0	0	1	OFF	OFF	OFF	1
0	1	0	OFF	OFF	OFF	1
0	1	1	OFF	OFF	OFF	1
1	0	0	OFF	OFF	OFF	1
1	0	1	OFF	OFF	OFF	1
1	1	0	OFF	OFF	OFF	1
1	1	1	ON	ON	ON	0

} 'NAND' gate

$t_{pd} < 1 \text{ n-sec}$

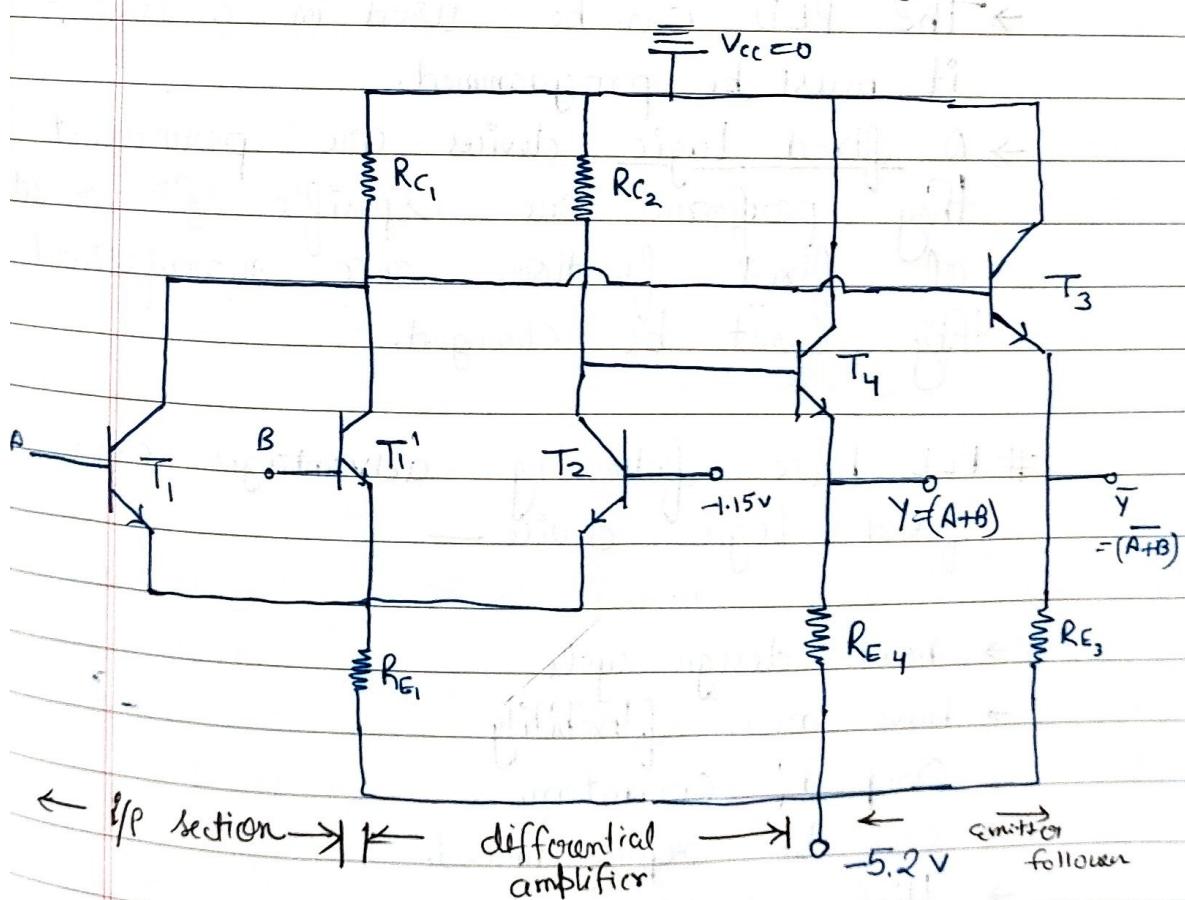
↳ ECL :- ECL logic family is the fastest logic family.

→ In this, transistor are operating in cut off and active mode, Hence it is non-saturated logic family.

→ due to the use of differential amplifier configuration Complementary O/p are available in ECL logic family.

→ Basic Gate is OR, and NOR Gate.

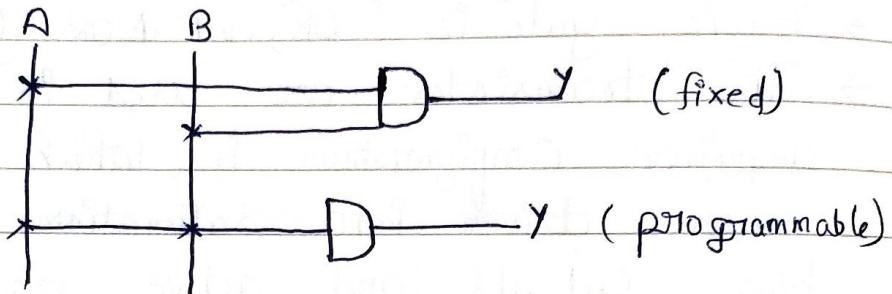
→ The transistors are used in differential amplifier configuration in which they are never driven into saturation. (switched b/w cut-off and active region).



*Most Imp.*

## # Programmable logic devices (PLD)

- A programmable logic devices used to build digital circuits.
- logic device can be classified into two broad categories —
  - fixed
  - Programmable.



- The PLD can be used in a circuit it must be programmed.
- fixed logic devices are permanent they perform one specific fn or set of fixed function. once manufactured they can't be changed.

## # PLD have following advantages over fixed logic device —

- Short design cycle
- have more flexibility
- Compact circuitry
- Can be reprogrammed
- High switching speed
- low development cost

# There are three types of programmable logic device -

- (1) ROM (Read Only memory)
- (2) PAL (Programmable Array logic)
- (3) PLA (Programmable logic array).

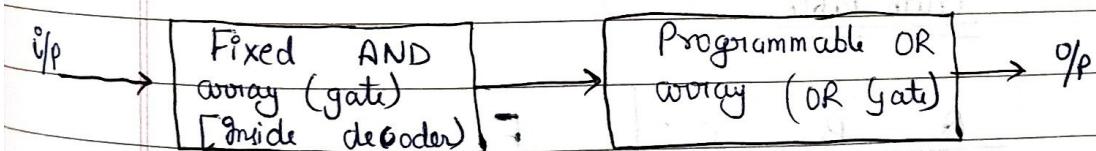
\* Basic Implementation

	<u>AND</u>	<u>OR</u>
→ ROM	fixed (decoder)	Programmable
→ PAL	programmable	fixed
→ PLA	Programmable	Programmable

Volatile

# ROM (Read Only Memory)

» ROM is made with the help of decoder and programmable OR gate.



» ROM come with special internal links that can be fused or open.

» The desired interconnection for a particular application requires that links be fused to form the required circuit path.

» Once a pattern is established for ROM

It is fixed, even when power is not available.

$$\text{Configuration of ROM} = 2^k \times n$$

$\therefore K = \text{no. of i/p of decoder}$  i.e. no. of  
 $2^k = \text{no. of o/p of decoder}$        $\boxed{\text{i/p of ROM}}$   
 $n = \text{no. of OR gate needed.}$

### Implementation of ROM

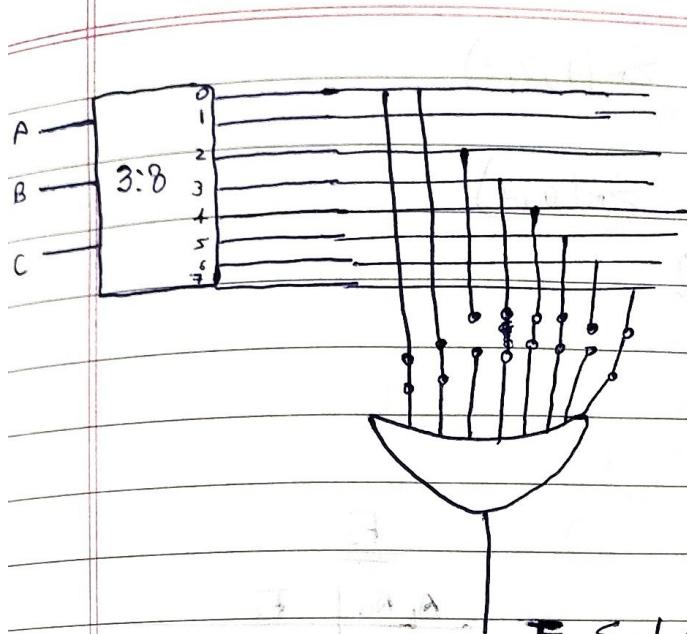
- no. of i/p in decoder = no. of variable in f.
- no. of OR Gate equal to no. of function to be implemented.
- no. of i/p in OR Gate equal to number of o/p of decoder.

Ques:- Implement  $f(A, B, C) = \sum m(0, 1, 4, 5, 7)$  by using ROM?

Sol:-

### Truth table

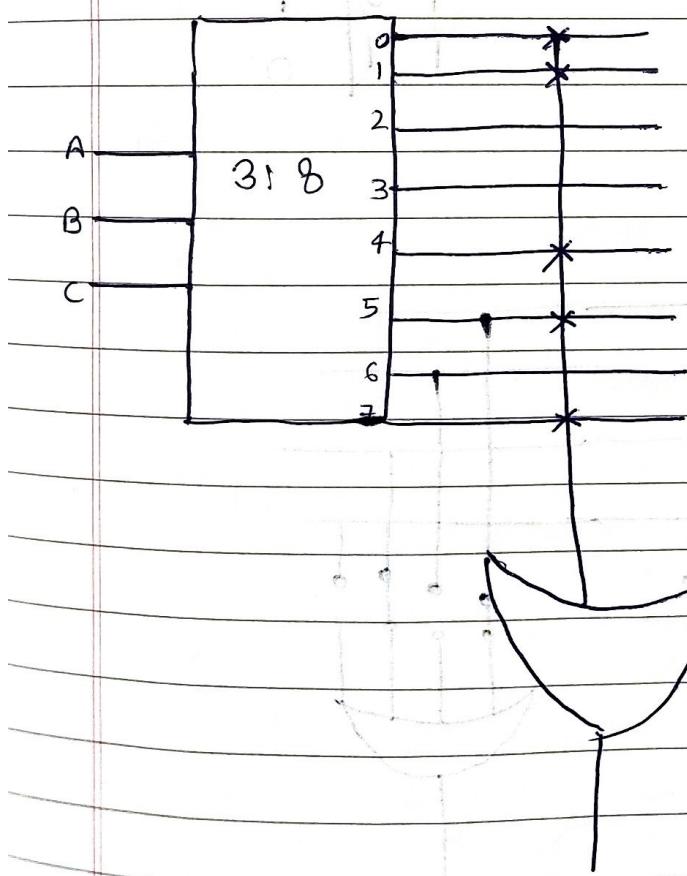
A	B	C	F
0	0	0	1
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	0
1	1	1	1



fuse connection

When the term is High  
otherwise use

$$F = \Sigma m(0, 1, 4, 5, 7)$$



$$F = \Sigma m(0, 1, 4, 5, 7)$$

$$\text{Ques:- } F_1(A_1, A_0) = \sum_m(1, 2, 3)$$

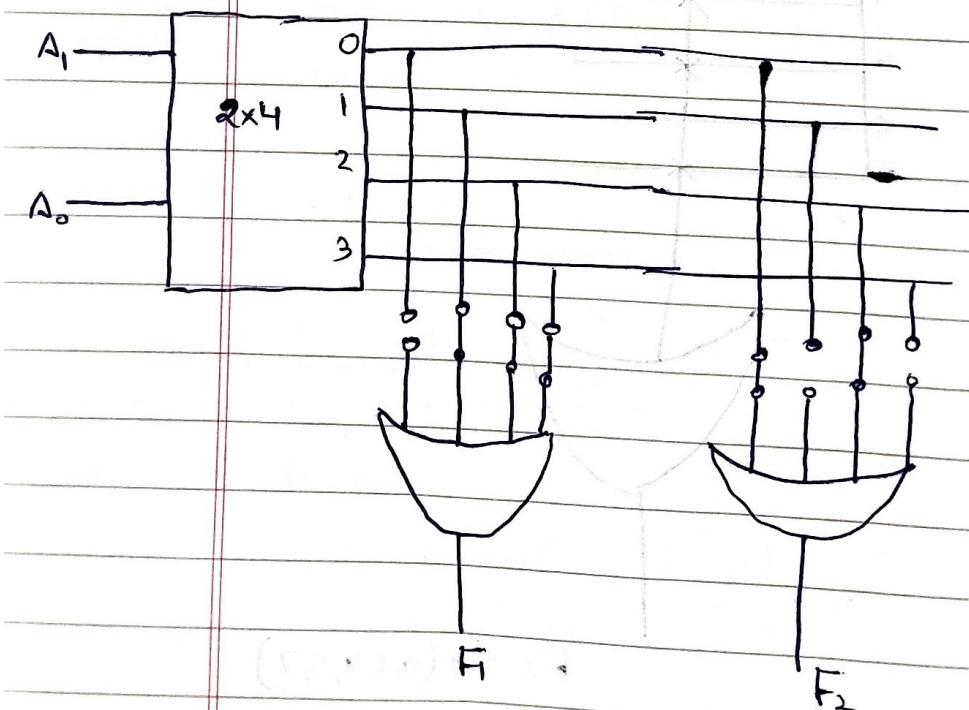
$$F_2(A_1, A_0) = \sum_m(0, 2)$$

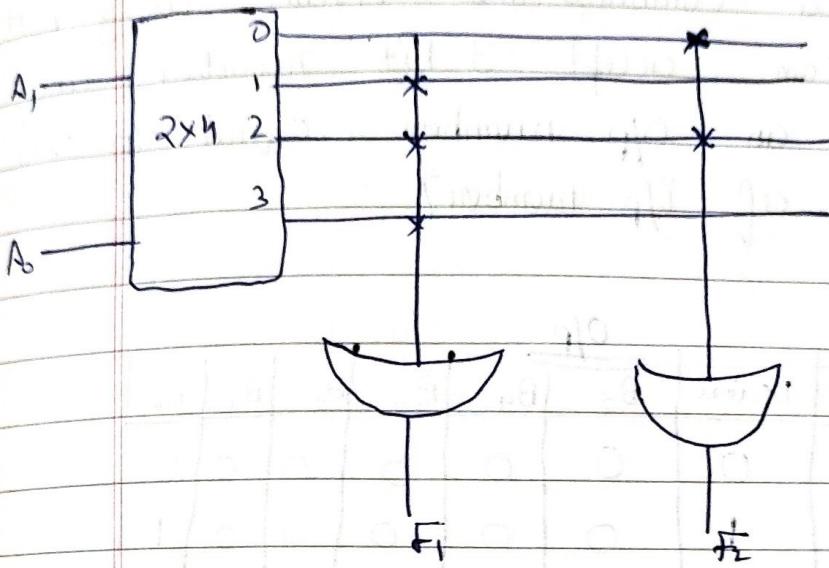
by using ROM?

~~10V.~~

Truth table

<u><math>F_1</math></u>		<u><math>F_2</math></u>	
$A_1$	$A_0$	$F_1$	$F_2$
0	0	0	0
0	1	1	1
1	0	1	0
1	1	1	0





Ques: Implement Using ROM of Boolean function

$$F = A + BC$$

Soln

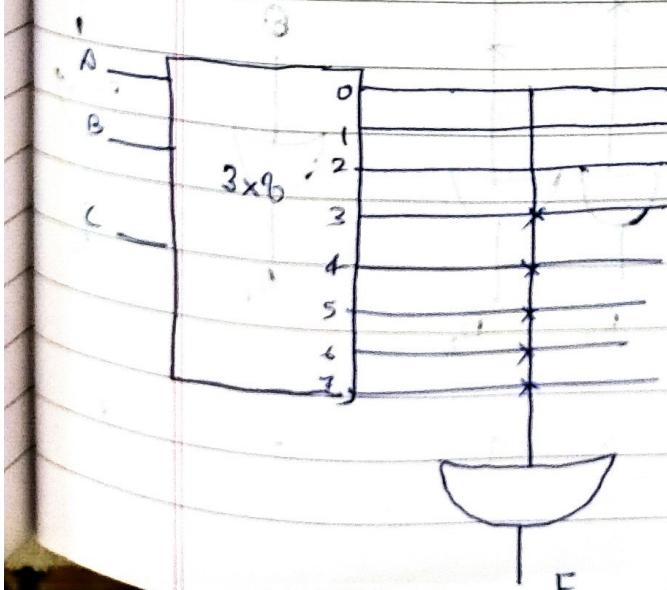
Reverse - K-map

$$F = A + BC$$

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

$$F = \sum m(3, 4, 5, 6, 7)$$

A	B	C	F
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	1
*	1	1	1



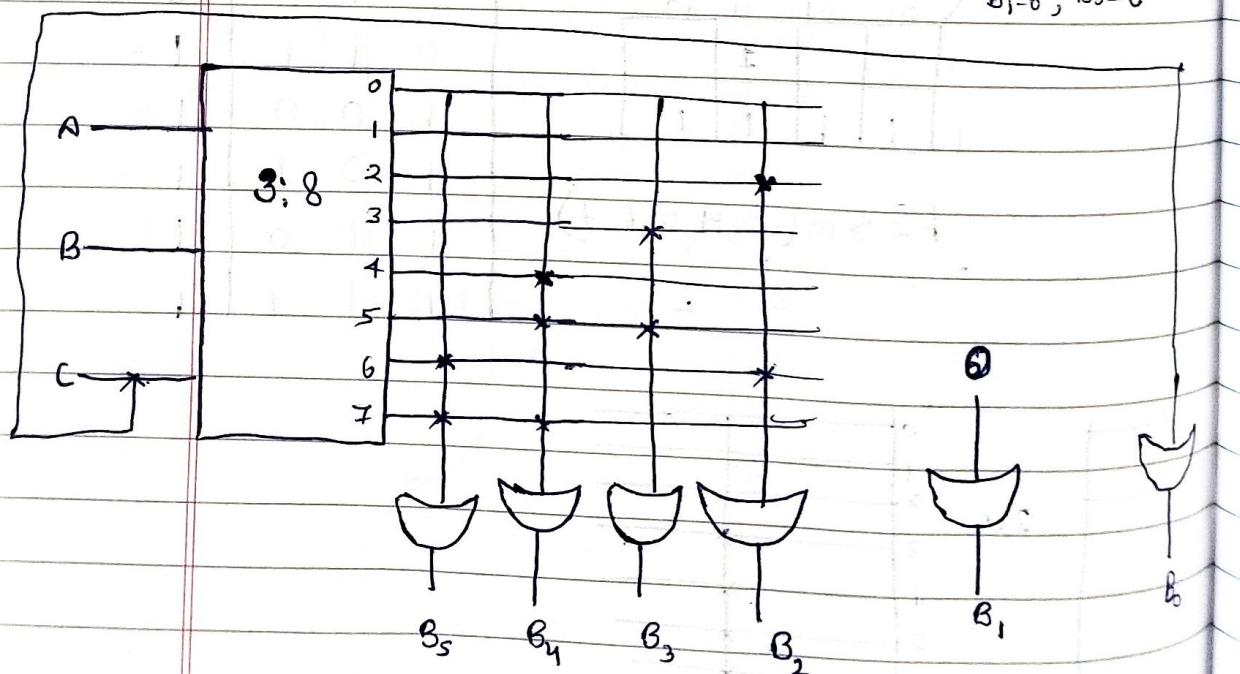
Ques. Design a Combinational Circuit using ROM which can accept 3-bit number and generate an o/p number equal to the square of i/p number?

Soln.

i/p.

A	B	C	Decimal	O/P					
				$B_5$	$B_4$	$B_3$	$B_2$	$B_1$	$B_0$
0	0	0	0	0	0	0	0	0	0
0	0	1	1	0	0	0	0	0	1
0	1	0	4	0	0	0	1	0	0
0	1	1	9	0	0	1	0	0	1
1	0	0	16	0	1	0	0	0	0
1	0	1	25	0	1	1	0	0	1
1	1	0	36	1	0	0	1	0	0
1	1	1	49	1	1	0	0	0	1

$B_1 = 0, B_0 = C$



110      1      0      6      1      0      0

Ques ① Design 3-bit Gray to binary code converter by using ROM?

Soln.

Ques ② Implement 3-bit binary to Gray code by using ROM?

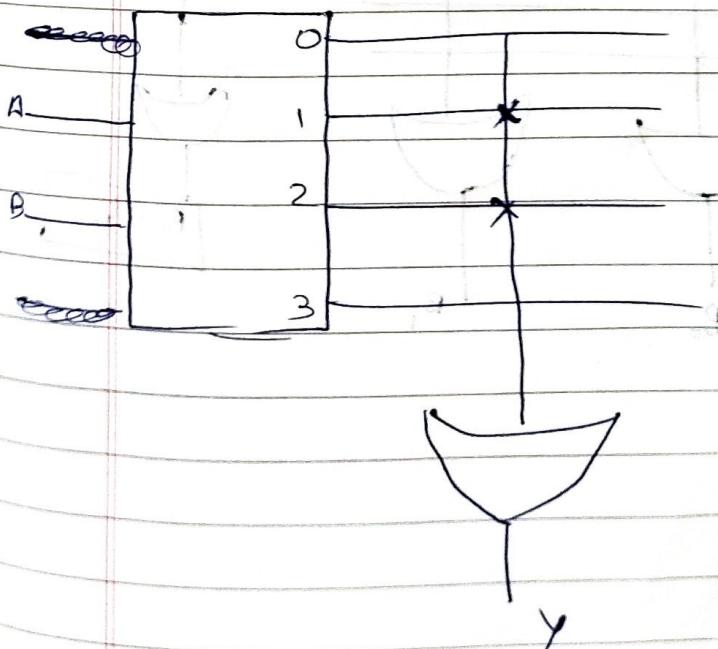
Ques ③: Implement 2 i/p X-OR Gate by using RAM?

Soln.

Truth table

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

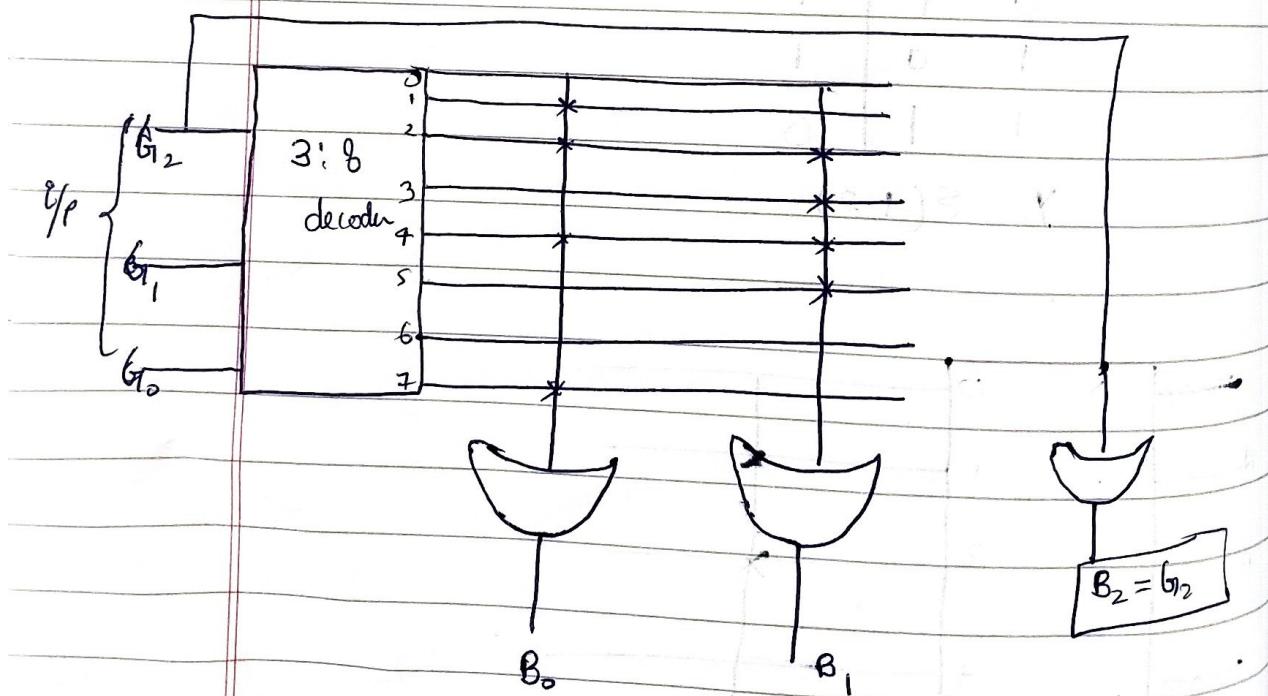
$$Y = \Sigma(1, 2)$$



#(1) Gray to Binary

<u>Gray</u>			<u>Binary</u>		
$G_2$	$G_1$	$G_0$	$B_2$	$B_1$	$B_0$
0	0	0	0	0	0
0	0	1	0	0	1
0	1	0	0	1	1
0	1	1	0	1	0
1	0	0	1	1	1
1	0	1	1	1	0
1	1	0	1	0	0
1	1	1	1	0	1

$$B_2 = G_2$$



Q.

Binary to Gray codeBinary $B_2 \ B_1 \ B_0$ 

0 0 0

0 0 1

0 1 0

0 1 1

1 0 0

1 0 1

1 1 0

1 1 1

Gray code $G_2 \ G_1 \ G_0$ 

0 0 0

0 0 1

0 1 1

0 1 0

1 1 0

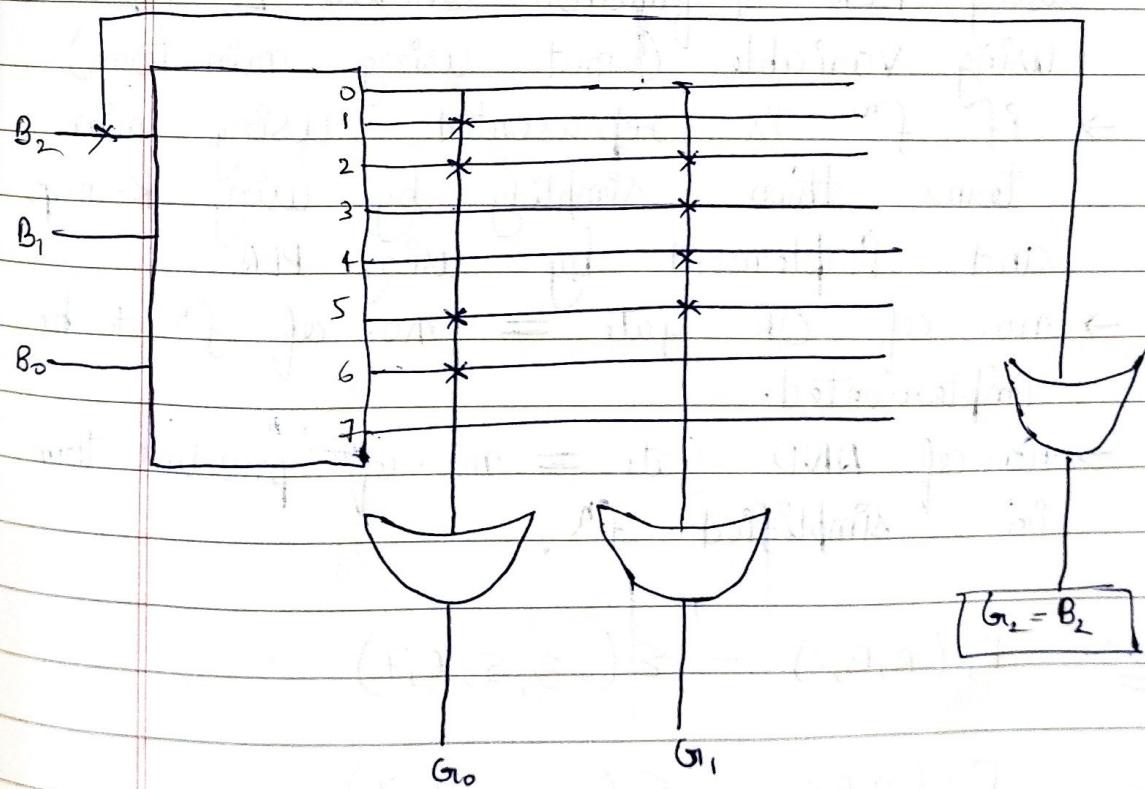
1 1 1

1 0 1

1 0 0

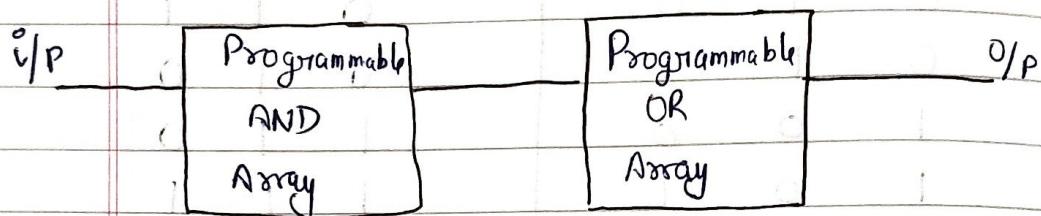
$$\boxed{B_2 = G_2}$$

Now,



# PLA (Programmable Logic Array)

» It is a logic circuit made using programmable "and" gate and programmable OR Gate.



→ In PLA both AND and OR are programmable.

→ for implementation of any function using PLA, function should be represented using variable (not using min term).

→ if  $f^n$  is represented using min terms than simplify by using K-map and implement by using PLA.

→ no. of OR Gate = no. of  $f^n$  to be implemented.

→ no. of AND Gate = no. of product term in simplified  $f^n$ .

Ques :-  $F_1(A, B, C) = \Sigma(3, 5, 6, 7)$

$F_2(A, B, C) = \Sigma(0, 2, 4, 7)$   
by using PLA?

Ans. $F_1$ 

		BC	00	01	11	10
		A	0	.	1	0
		B	0	1	1	1
1			1	1	1	1

$$F_1 = BC + AC + AB$$

 $F_2$ 

Sorry

		BC	00	01	11	10
		A	0	.	1	0
		B	0	1	0	0
1			0	0	0	0

OK

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

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

 $F_2$ 

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

$$F_2 = \bar{A}\bar{C} + \bar{B}\bar{C} + A\bar{B}C$$

		$\bar{F}_2$	BC	00	01	11	10
		A	0	0	0	0	0
		B	1	0	0	0	0
1			0	0	0	0	0

$$\bar{F}_2 = AC + \bar{B}\bar{C} + A\bar{B}\bar{C}$$

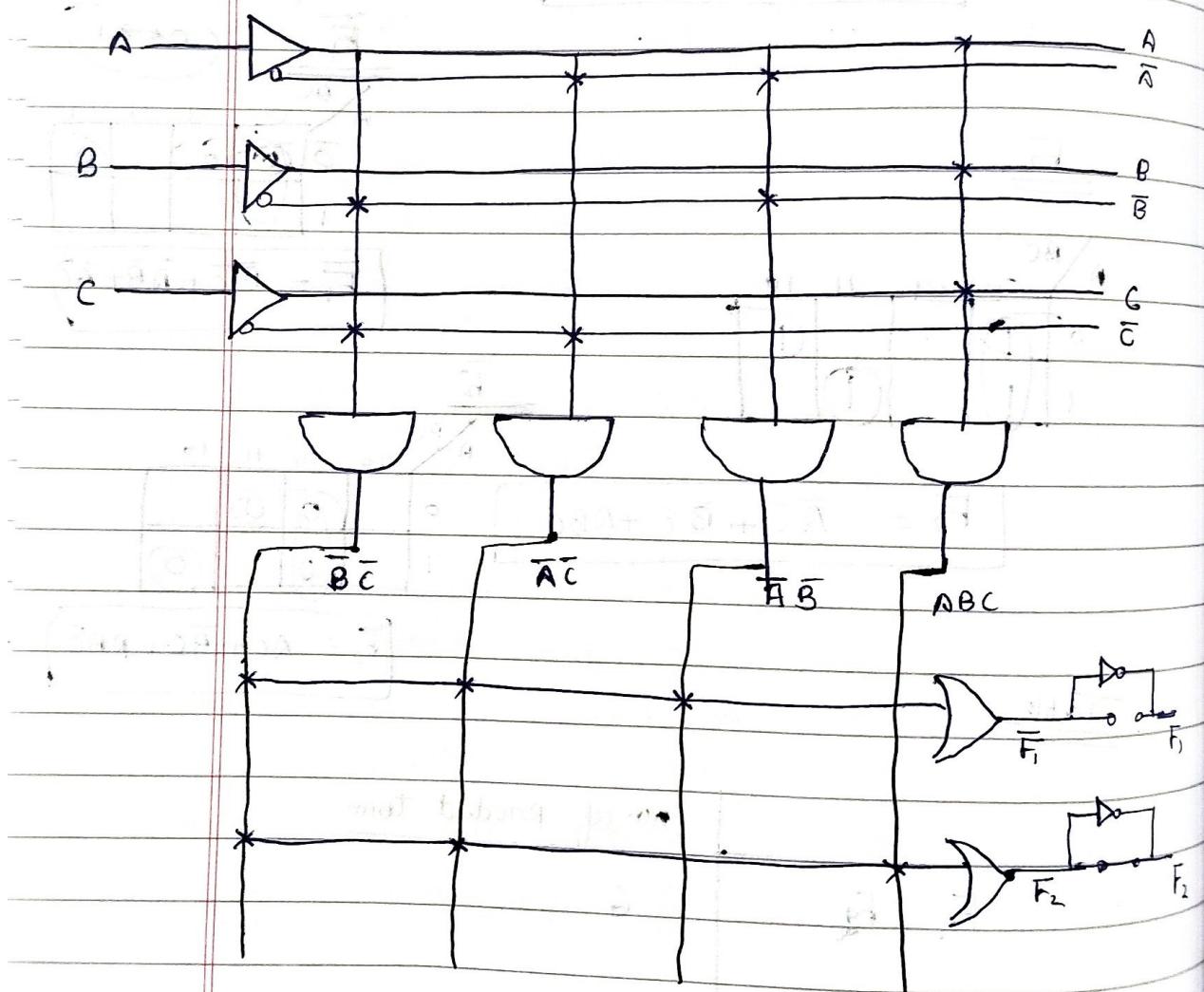
Table

				no. of product term
$F_1$	$\bar{F}_1$	$F_2$	$\bar{F}_2$	6
$F_1$	$\bar{F}_1$	$F_2$	$\bar{F}_2$	6
$\bar{F}_1$	$F_1$	$F_2$	$\bar{F}_2$	6
$\bar{F}_1$	$F_1$	$\bar{F}_2$	$F_2$	4 ←
$F_1$	$\bar{F}_1$	$\bar{F}_2$	$F_2$	6

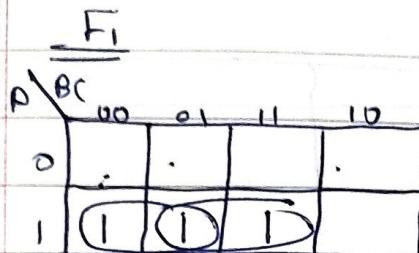
$F_1, F_2 = 4$ 

PLA table

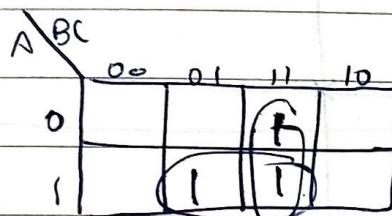
Product term:	Input A B C	Output F
$\bar{B}\bar{C}$	— 0 0	1 1
$\bar{A}\bar{C}$	0 — 0	1 1
$\bar{A}\bar{B}$	0 0 —	1 1
$A\bar{B}C$	1 1 1	— 1



Ques:-  $F_1 = \sum(4, 5, 7)$  &  $F_2 = \sum(3, 5, 7)$ .  
 by using PLA = ?

Sol<sup>n</sup>

$$F_1 = A\bar{B} + AC$$



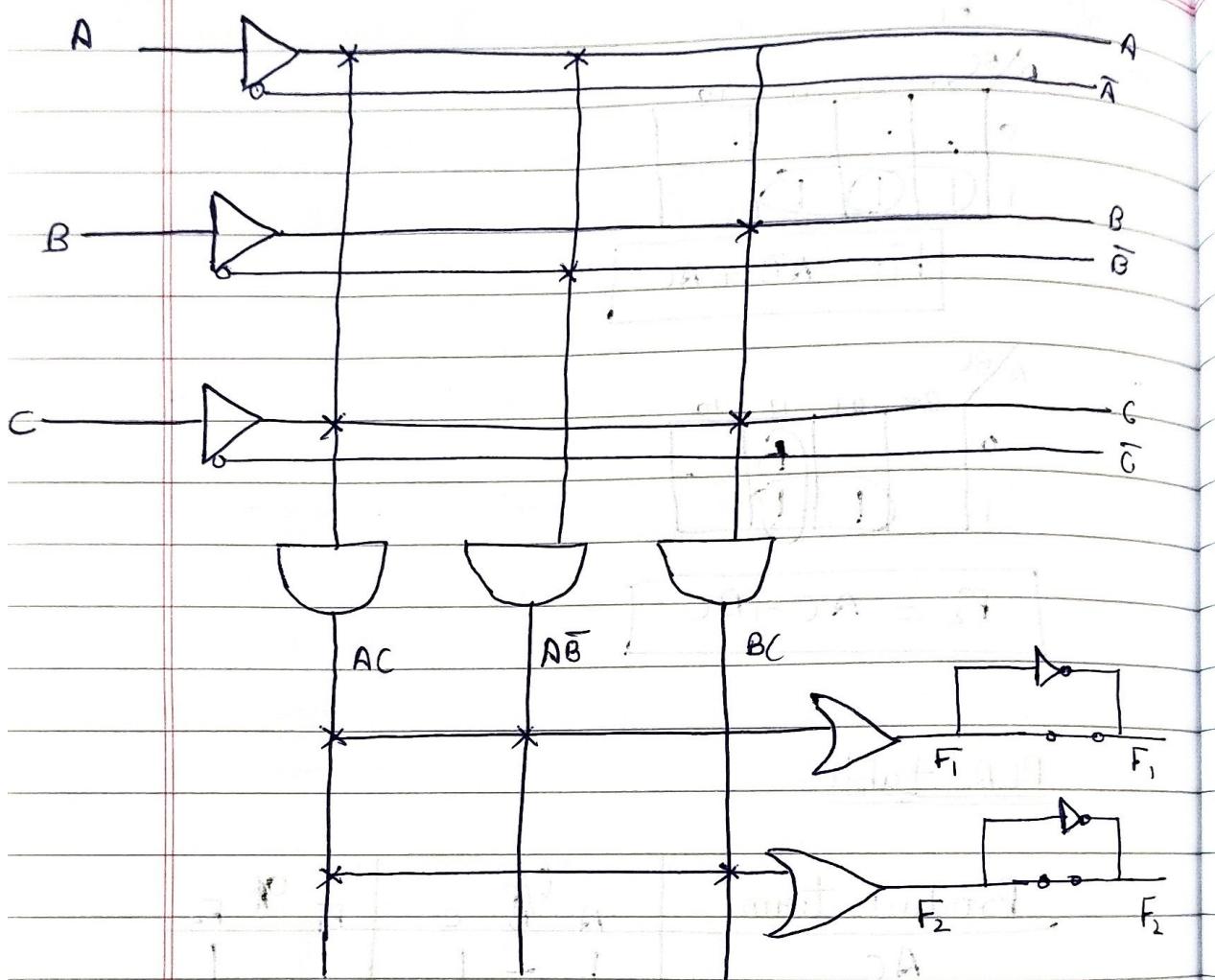
$$F_2 = AC + BC$$

### PLA Table

Product term	$A \frac{i/p}{\bar{B}} C$	$F_1 \oplus F_2$
AC	1 - 1	1 1
$A\bar{B}$	1 0 -	1 -
BC	- 1 1	- 1

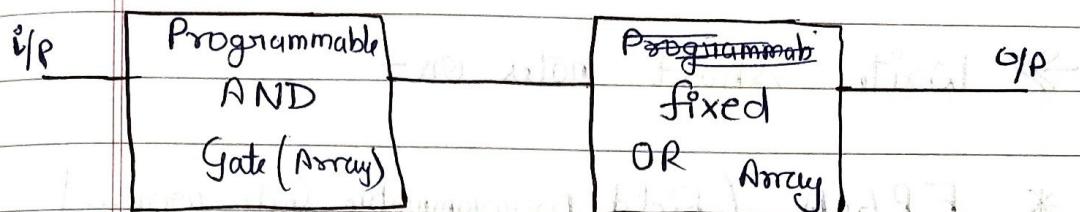
» Implement PLA

next page



## # P A L (Programmable Array logic)

» It is a logic device in which AND gate is programmable and OR Gate is fixed.

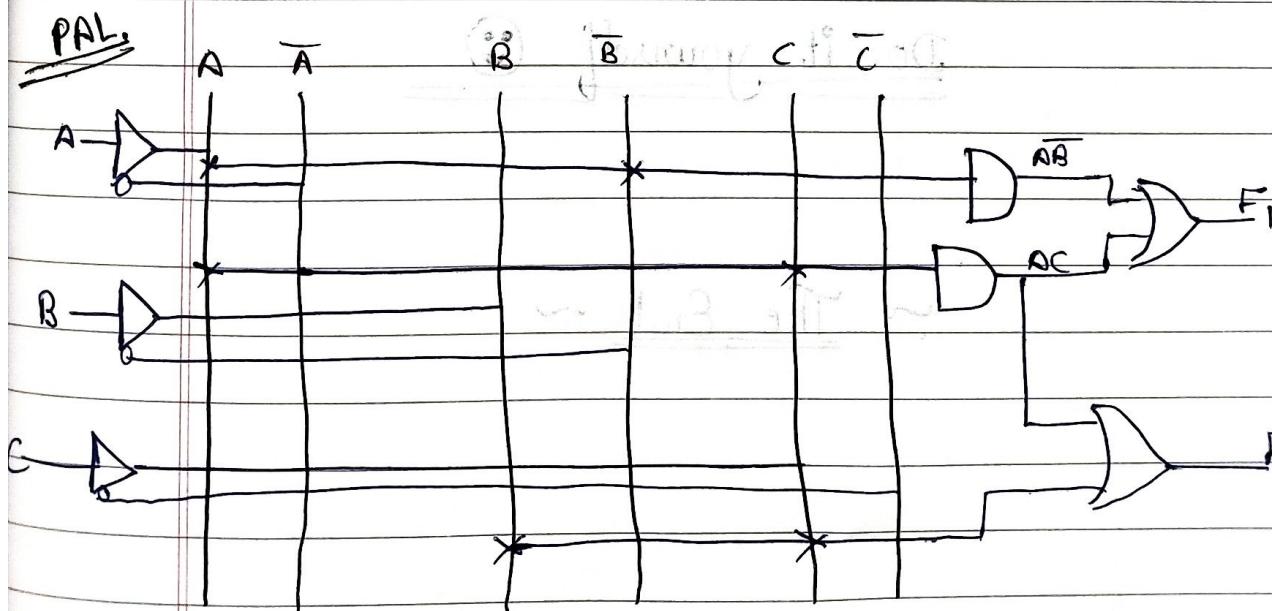


Eg.:  $F_1 = \Sigma(4, 5, 7)$        $F_2 = \Sigma(3, 5, 7)$

By K-map

$$F_1 = A\bar{B} + AC$$

$$F_2 = AC + BC$$



Ques: \* Implement —

$$\textcircled{1} W(A, B, C, D) = \Sigma(2, 12, 13)$$

$$\textcircled{2} x(A, B, C, D) = \Sigma(7, 8, 9, 10, 11, 12, 13, 14, 15)$$

$$\textcircled{3} \quad y(A,B,C,D) = \Sigma(0, 2, 3, 4, 5, 6, 7, 8, 10, 11, 15)$$

and

$$\textcircled{4} \quad z(A,B,C,D) = \Sigma(1, 2, 8, 12, 13)$$

→ Write short notes on -

- \* FPGA (Field programmable gate array)
- \* ROM
- \* PROM
- \* EEPROM
- \* EEPROM

or

E<sup>2</sup> PROM

Do it yourself ☺

~ The End ~