

## Logic Gates

Subtraction, 1's complement  
, 2's complement.

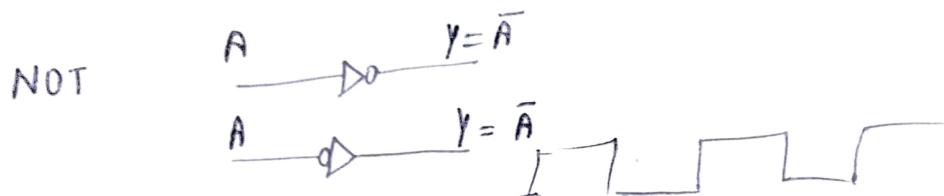
It is a physical device which performs logic operation using given two or more inputs and produce a single logical output.

Basic Gates - AND, NOT & OR.

Universal Gates - NAND & NOR

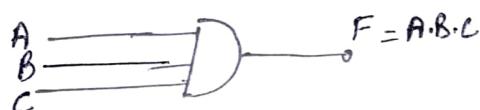
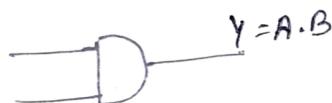
Arithmetic Gates - X-OR & X-NOR.

T.T - o/p possible from all comb. of I/o.



AND

Symbol -



Truth Table -

	A	B	$Y = A \cdot B$
enable	1	1	1
	1	0	0
disable	0	1	0
	0	0	0

A fixed B change O/P  
vice-versa

A fixed B change O/P  
No change

Associative Law -

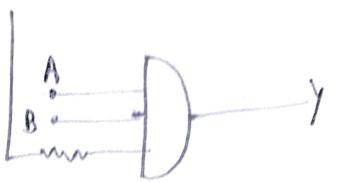
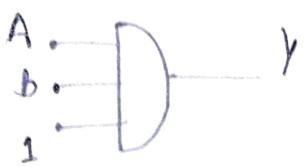
$$A \cdot (B \cdot C) = (\underbrace{A \cdot B}_{A \cdot X}) \cdot C$$

Commutative Law -  $A \cdot B = B \cdot A$

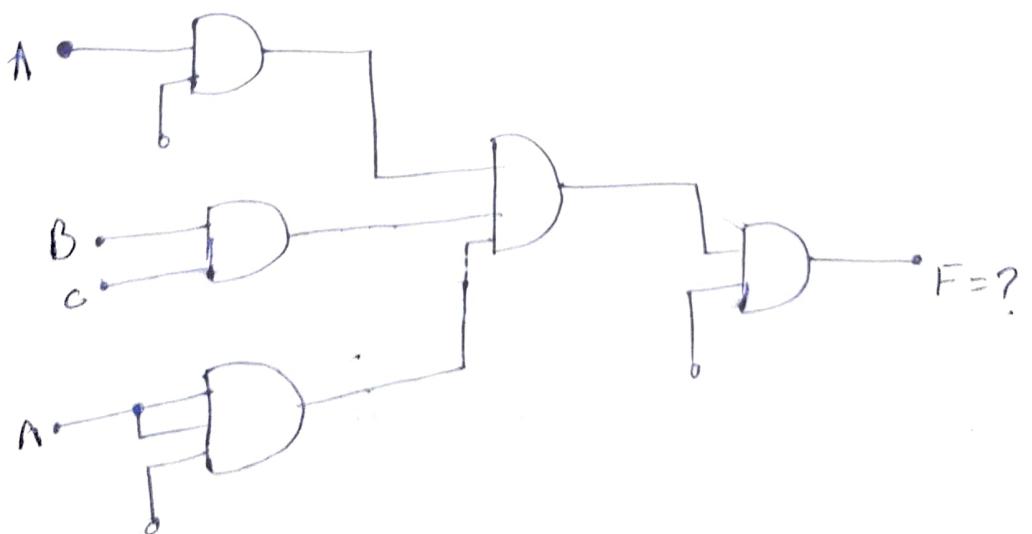
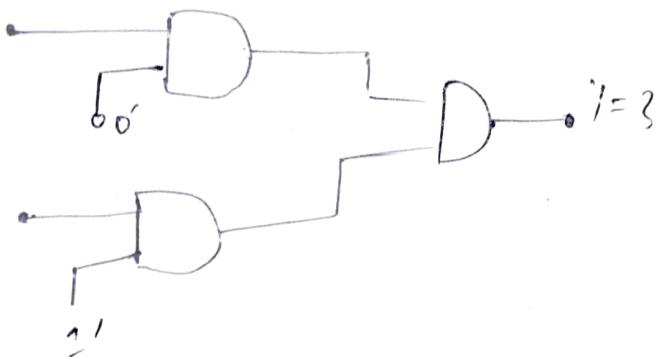
Enable & Disable - "0"  $\rightarrow$  disable with O/P = 0;

Buffer  $\rightarrow$  "1"  $\rightarrow$  enable with O/P = 0 and

Unused Input - In (transistor-transistor logic), if any input is open or floating it will act as one.  
 $+ V_{CC} \rightarrow "0"$



✓ TTL logic      A — D — Y = A · 1  
 ✗ ECL logic  $\rightarrow 0$



# OR Operator -

Symbol -  $A+B$



$\oplus$  or  $+$

$$Y = A + B$$

A	B	$Y = A + B$
0	0	0
0	1	1
1	0	1
1	1	1

Associative Law  $\rightarrow$  Commutative Law

## ⑤ Enable & Disable

A	B	$A+B$
enable [	0	0
	1	1

Buffer  
o/p change if A fixed  
Enable

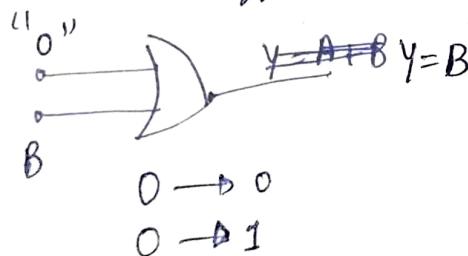
A	B	$A+B$
disable [	1	0
	1	1

disable o/p fixed if A fixed

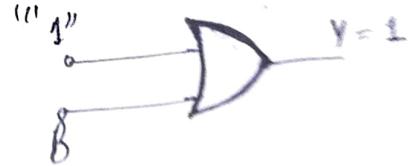
"0"  $\rightarrow$  enable  $\rightarrow$  Buffer

"1"  $\rightarrow$  disable

Enable / Buffer



disable



unused input

Unused Input

TTL logic  
ECL

floating input  $\rightarrow$  "1"  
 $\rightarrow$  "0"

① Connect to zero.



② TTL logic  $\rightarrow$  "0" enable and "1"



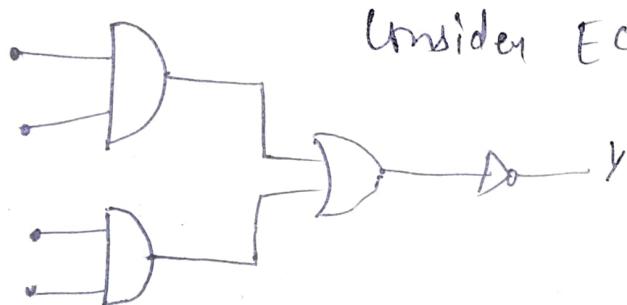
Consider TTL logic



open {

open imp part how to decide

(A)

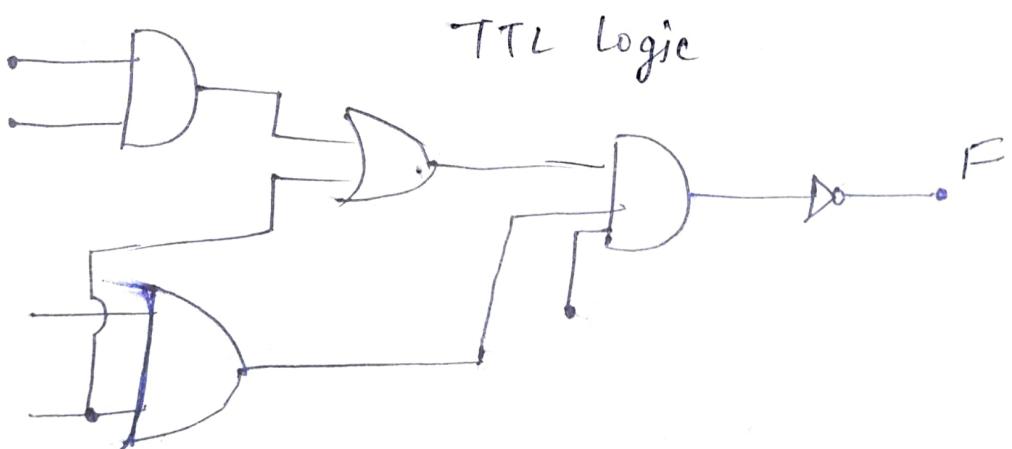


Consider ECL logic

TTL = 1

ECL = 0

(B)

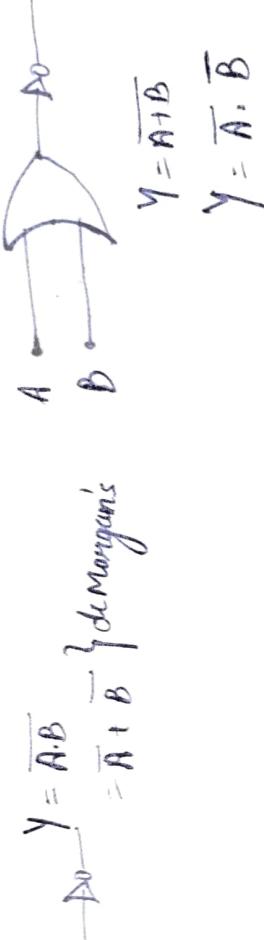
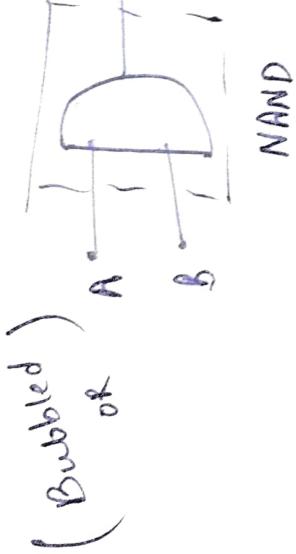


TTL logic

## Universal Gate.

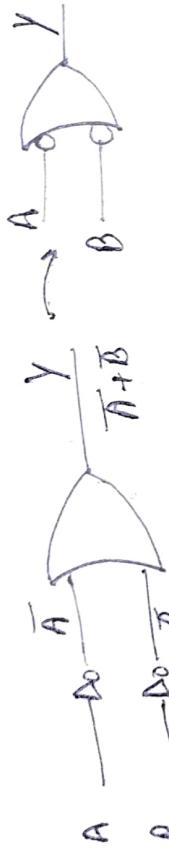
### NAND Gates.

NOR gate.



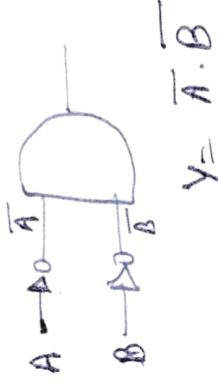
NAND

OR and NOT

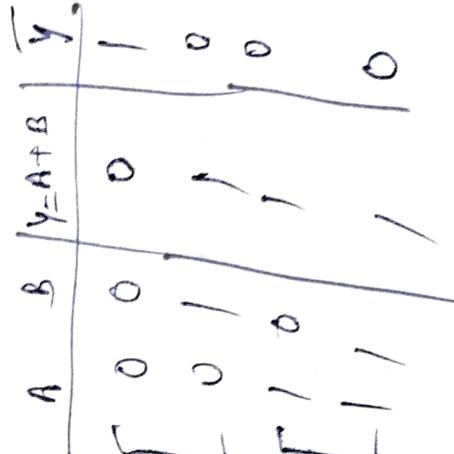
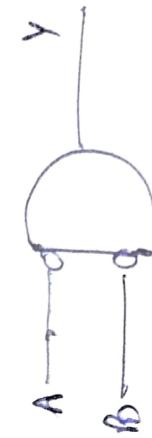


symbol -   $\rightarrow$  Y

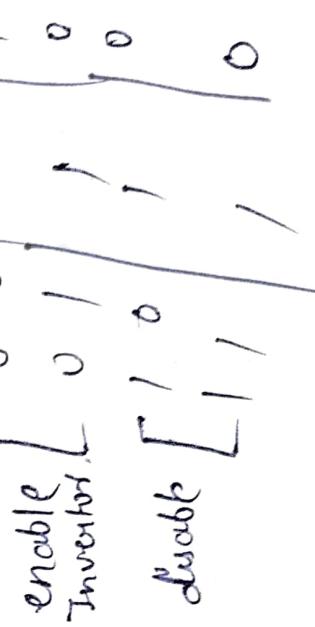
AND and NOT



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



$Y = A + B$



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

"0"  $\rightarrow$  disable

"1"  $\rightarrow$  enable

Same as  
AND / gate

commutative law - ✓

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

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

Invertor  
because  $Y = \overline{A}$

Associative law how X

commutative law - ✓

"0"  $\rightarrow$  enable

"1"  $\rightarrow$  disable

Same as  
OR gate.

Inclusive OR

Inclusive NOR

Symbols -



$$Y = A \oplus B$$

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

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

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

Truth table.

		Y	
A	B	0	1
Enable	0	0	1
Buffer	1	1	0
Enable	1	0	1
Inverter			
Enable			
Buffer			

Enable  
Inverter  
Enable  
Buffer



$$Y = A \oplus B$$

Symbols -



$$Y = A \oplus B$$

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

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

Truth table.

		Y	
A	B	0	1
Enable	0	0	1
Buffer	1	1	0
Enable	1	0	1
Inverter			
Enable			
Buffer			

Enable  
Inverter  
Enable  
Buffer

		Y	
A	B	0	1
Enable	0	0	1
Buffer	1	1	0
Enable	1	0	1
Inverter			
Enable			
Buffer			

Enable  
Inverter  
Enable  
Buffer

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

→ Absorptive and Commutative laws

Properties

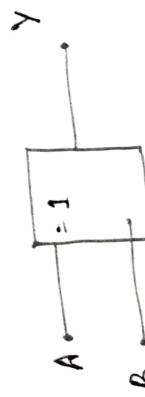
$$\rightarrow n = \text{even } 1$$

$$A \oplus A = 1$$

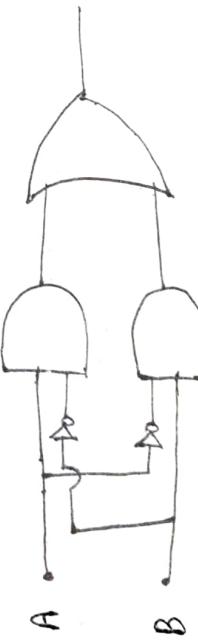
$$A \oplus 0 = \bar{A}$$

$$A \oplus 1 = A$$

$$B \oplus \bar{A} = 0$$



Truth table



Controlled Inverter - Because

of Buffer and  
Inverter  
property

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

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

$$= 1$$

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

$$= A$$

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

## Properties.

$$\textcircled{1} \quad A \oplus A = 0$$

$$A \oplus A \oplus A \dots n$$

$$\begin{aligned} n \text{ even} &= 0 \\ n \text{ odd} &= A \end{aligned}$$

$$\textcircled{2} \quad A \oplus \bar{A} = A \cdot \bar{A} + \bar{A} \cdot A$$

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

$$= 1$$

$$\textcircled{3} \quad A \oplus 1 = A \cdot 0 + \bar{A} \cdot 1$$

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

$$= \bar{A}$$

$$\textcircled{4} \quad A \oplus 0 = A \cdot 1 + \bar{A} \cdot 0$$

$$= A + 0$$

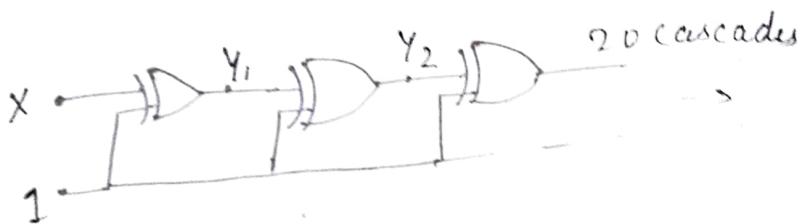
$$= A$$

$$\textcircled{5} \quad A \oplus B = C$$

$$B \oplus C = A$$

$$A \oplus C = B$$

$$A \oplus B \oplus C = 0$$



$$\begin{aligned} Y_1 &= X \cdot \bar{1} + \bar{X} \cdot 1 \\ &= 0 + \bar{X} \\ &= \bar{X} \leftarrow \text{Odd} \end{aligned} \quad \begin{aligned} Y_2 &= \bar{X} \cdot \bar{1} + \bar{\bar{X}} \cdot 1 \\ &= 0 + X \\ &= X \leftarrow \text{Even} \end{aligned}$$

at 20 cascades =  $Y = X$

Examples -  $A \oplus \bar{A} \oplus \bar{A} \oplus A \oplus A \oplus \bar{A} \oplus \bar{A} \oplus A \oplus A$

- ①  $1 + E$
- ②  $A \oplus 0$
- ③ 1 NOR B
- ④ A

$$\textcircled{2} \quad f(A, B) = A \oplus B$$

Simplified form of function  $f(f(x \oplus y, z) w)$

$$\textcircled{a} \quad (x \oplus y \oplus z) \cdot w \quad \textcircled{b} \quad x \oplus y \oplus z \oplus w$$

$$\textcircled{c} \quad (x \oplus y \oplus z) + w \quad \textcircled{d} \quad \text{None.}$$

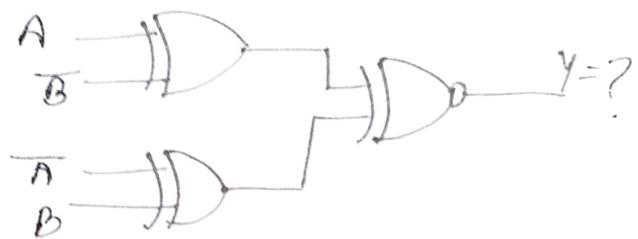
\*\*\* XOR and XNOR  $\xrightarrow{\text{complement}}$   
to each other when even input

\*\*\* XOR and XNOR are same when  
odd no. of input.

Examples -

proof -  $\underbrace{A \oplus B \oplus C}_{\times} = \underbrace{A \oplus B \oplus C}_{Y} \oplus C$

$$A \oplus B \oplus C \oplus D \neq A \oplus B \oplus C \oplus D$$

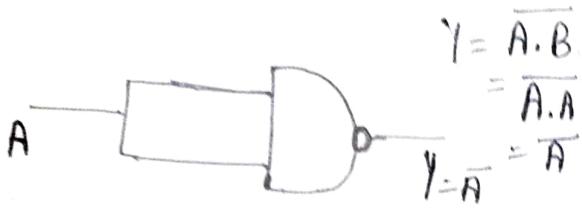


$$Y = A \oplus A \oplus A \oplus A \oplus A \oplus A \oplus \bar{A} \oplus$$

NAND as Universal Gate

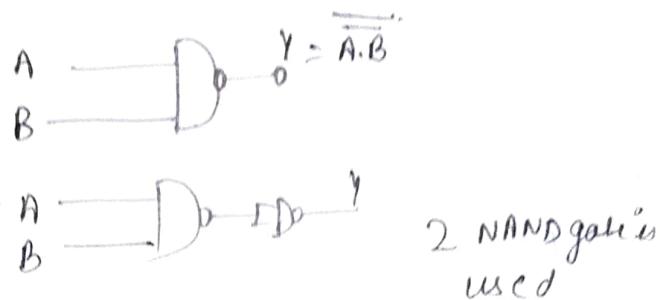
If we show NAND  $\rightarrow$  as OR, AND and NOT, then it proves that NAND or NOR are universal gate.

NAND as NOT



If J/P is same

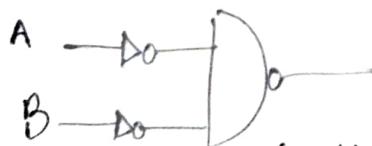
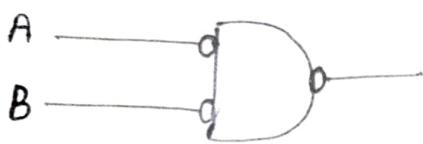
NAND as AND



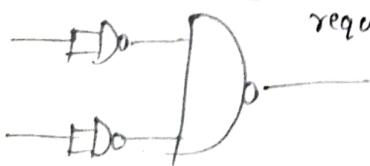
NAND as OR.

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

$$OR = A + B$$

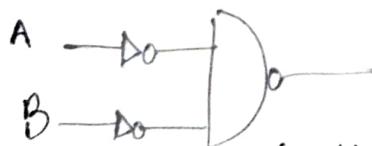
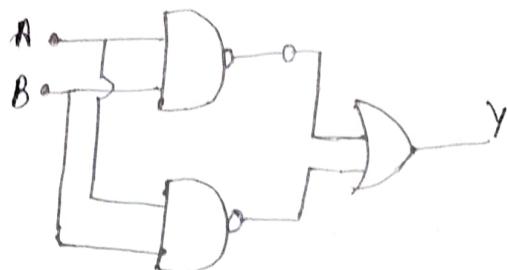


Complete NAND gate is used.  
3 NAND gate required



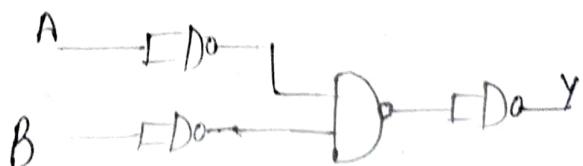
NAND as NOR, EXNOR

$$Y_1 = \bar{A} \cdot \bar{B} \quad Y = A \cdot B + \bar{A} \cdot \bar{B} \\ = \bar{Y} + Y$$



NAND as NOR

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



four NAND gate required,

## NAND as EX-OR

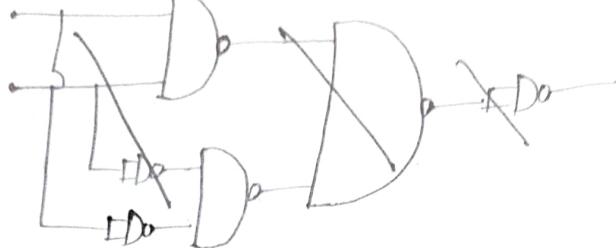
$$Y = A \cdot \bar{B} + \bar{A} \cdot B$$

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

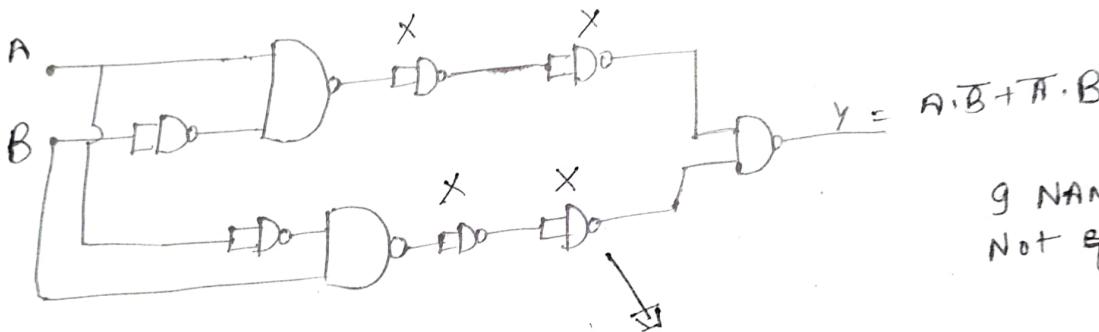
$$= \overline{A + B}$$

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

**NAND**  
 ④ 2 AND  
 2 NOT  
 1 OR GATE 3

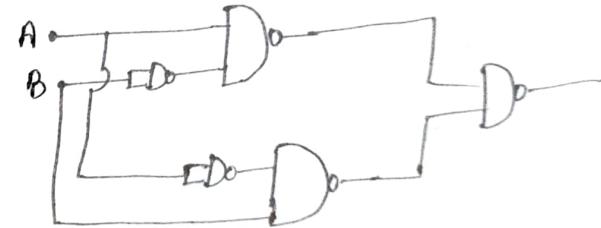


NAND  $\rightarrow$  OR  
 $Y = A + B$

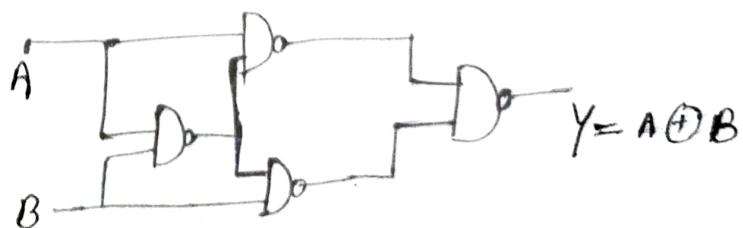


NAND as EX NOR

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



most efficient



$$Y = A \oplus B$$

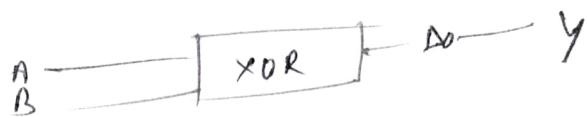
## NAND as EXNOR

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

~~2 NOT~~ → 2  
 + OR → 3  
~~2 AND~~ →

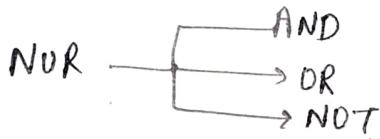
Complement Ex-NOR

Total required  
4 NAND + 1 NAND



- ① min. no. of two I/p NAND gate required to implement  $\overline{x}y.z$
- ② ~~min. no. of~~ 11  $\boxed{\overline{x}y + wz}$

NOR Gate as Universal Gate.



NOR



NOT

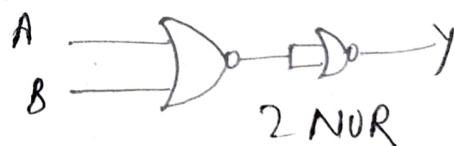
$$\begin{aligned}
 Y &= \overline{A+B} = \overline{A} \cdot \overline{B} \\
 &= \overline{A+A} = \overline{A} \cdot \overline{A} \\
 &= \overline{A}
 \end{aligned}$$

A → NOR →  $Y = \overline{A}$

~~AND~~ OR

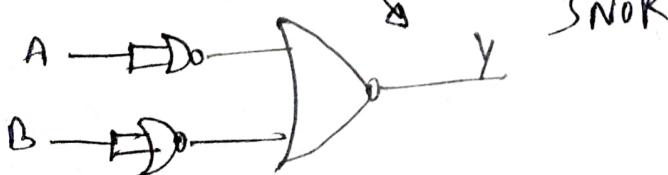
$$Y = \overline{A+B}$$

$$Y = A+B$$



AND  $Y = A \cdot B$

$$\begin{aligned}
 Y &= \overline{A+B} \\
 &= \overline{\overline{A} \cdot \overline{B}} \\
 &\Downarrow \\
 &= \overline{A} \cdot \overline{B}
 \end{aligned}$$



NOR — NAND

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

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

$$= (\overline{A}) \cdot (\overline{B})$$

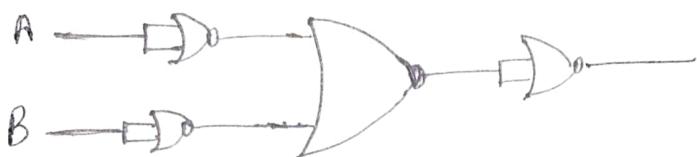
$$Y = \overline{A \cdot B}$$

$$= \overline{A}$$

complement o/p AND gate

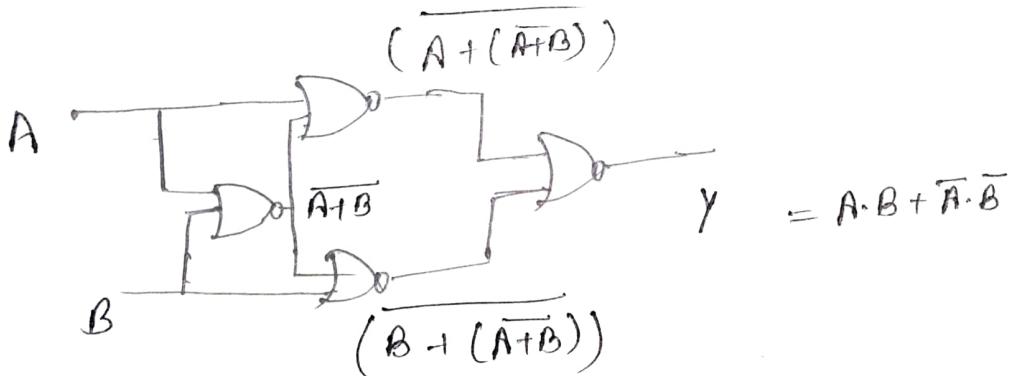
$$\overline{A \cdot B} = A \cdot \overline{B}$$

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



XNOR -

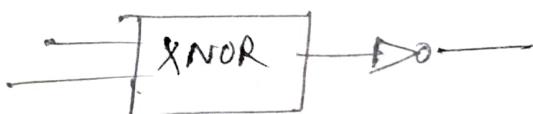
$$\hookrightarrow Y = A \cdot B + \overline{A} \cdot \overline{B}$$



↳ NOR Gate

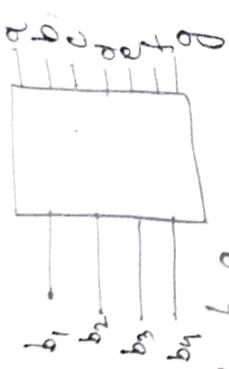
for

XOR S NOR Gate



Seven segment display decodes.

0 1 2 3 4 5 6 7 8 9



$b_1, b_2, b_3, b_4$	$a, b, c, d, e, f, g$
0 0 0 0 (0)	1 1 1 1 1 1 0
0 0 0 1 (1)	0 1 1 0 0 0 0
0 0 1 0 (2)	1 1 0 1 1 0 1



g full

$$Imp = a +$$

	$A$	$B$	$C$	$F$
$m_0$	0	0	0	0
$m_1$	0	0	1	0
$m_2$	0	1	0	1
$m_3$	0	1	1	0
$m_4$	1	0	0	1
$m_5$	1	0	1	1
$m_6$	1	1	0	1
$m_7$	1	1	1	1

Sop

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

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

Minterm

$$F(A, B, C) = m_2 + m_4 + m_5 + m_6 + m_7$$

$$F(A, B, C) = \sum m(2, 4, 5, 6, 7)$$

minimization.

$$\begin{aligned}
 F &= \underline{\bar{A} \cdot B \cdot C} + \underline{\bar{A} \cdot \bar{B} \cdot \bar{C}} + A \cdot \bar{B} \cdot C + A \cdot \bar{B} \cdot \bar{C} + A \cdot B \cdot C \quad (\text{Canonical}) \\
 &= \underline{\bar{B} \cdot \bar{C}} (\underline{\bar{A} \cdot B + AB} \quad \underline{\bar{B} \cdot \bar{C} + A \cdot \bar{B}} + A \cdot C + A \cdot B + \\
 &= \underline{\bar{A} \cdot B \cdot \bar{C}} + A \cdot \bar{B} (C + \bar{C}) + AB(C + \bar{C}) \\
 &= \underline{\bar{A} \cdot B \cdot \bar{C}} + A \cdot \bar{B} + A \cdot B \\
 &= \underline{\bar{A} \cdot B \cdot \bar{C}} + A \cdot (B + \bar{B}) \\
 &= \underline{\bar{A} \cdot B \cdot \bar{C}} + A \\
 &= A + B \cdot \bar{C} \quad \text{minimal form}
 \end{aligned}$$

Canonical / Standard SOP - Each minterm is having all the variables in normal or complemented form.

Minimal SOP form - Each minterm doesn't have all the variables in normal or complemented form

	A	B		Y
m <sub>0</sub>	0	0		1
m <sub>1</sub>	0	1		1
m <sub>2</sub>	1	0		0
m <sub>3</sub>	1	1		1

$$Y(A, B) = \sum m(0, 1, 3)$$

$$\begin{aligned}
 &= \bar{A} \cdot \bar{B} + \bar{A} \cdot B + A \cdot \bar{B} \\
 &= \bar{A} \cdot \bar{B} + B(A + \bar{A}) \\
 &= \bar{A} \cdot \bar{B} + B \\
 &= \bar{A} + B
 \end{aligned}$$

POS form product of sum

$$\begin{aligned}
 1 &\rightarrow \bar{A} \\
 0 &\rightarrow A
 \end{aligned}$$

I/P 3

Total 2<sup>3</sup>  
cnt.

POS form is used when the output is low or "zero"

	A	B	C		Y
m <sub>0</sub>	0	0	0		0
m <sub>1</sub>	0	0	1		0
m <sub>2</sub>	0	1	0		1
m <sub>3</sub>	0	1	1		0
m <sub>4</sub>	1	0	0		1
m <sub>5</sub>	1	0	1		1
m <sub>6</sub>	1	1	0		1
m <sub>7</sub>	1	1	1		1

$$\begin{aligned}
 F(A, B, C) &= \pi(0, 1, 3) \\
 Y &= (A + B + C) \cdot (A + B + \bar{C}) \cdot (A + \bar{B} + C) \\
 &\downarrow \quad \downarrow \quad \downarrow \\
 &\text{Maxterm M}
 \end{aligned}$$

Better to convert POS  $\rightarrow$  SOP

$$\begin{aligned}Y &= (A+B+C) \cdot (A+B+\bar{C}) \cdot (A+\bar{B}+\bar{C}) \\&= (x+C) \cdot (x+\bar{C}) \cdot (A+\bar{B}+\bar{C}) \\&= (x+x\bar{C}+xC) \cdot (A+\bar{B}+\bar{C}) \\&= (x+x) \cdot (A+\bar{B}+\bar{C}) \\&= (\underline{A+B}) \cdot (\underline{A+\bar{B}+\bar{C}}) = \begin{array}{l}\text{distributive law} \\ A+B, (\bar{B}+\bar{C})\end{array} \\&= \cancel{A} + \cancel{A} \cdot \cancel{B} + \cancel{A} \cdot \cancel{C} + A \cdot B + B \cdot \cancel{C} = \cancel{A} + B \cdot \bar{C} \\&= \cancel{A} + \cancel{A} + (A+B) \cdot \cancel{C} = \cancel{A} + \cancel{(A+B)} \cdot \cancel{C} \\&= \cancel{A} + (A+B) \cdot \bar{C}\end{aligned}$$

SOP Minimal to Canonical form Ex  $A+B'C$

Step 1 - Total no. of Variable 3

Step 2 - Variable absent in Each minterm

$$N_1 = Bx \quad Cx \quad A\checkmark$$

$$N_2 = Ax \quad B\checkmark \quad C\checkmark$$

$$\begin{aligned}g &= ABC + AB'C \\&= AC(B+B') \\&= AC \cdot 1\end{aligned}$$

Step 3 -  $A + B'C$

$$A \cdot 1 \cdot 1 + 1 \cdot B'C$$

$$A \cdot (B+B')(C+C') + (A+A') \cdot B' \cdot C$$

$$(A \cdot B + A \cdot B')(C+C') + A \cdot B' \cdot C + A' \cdot B' \cdot C$$

$$A \cdot B \cdot C + A \cdot B \cdot C' + A \cdot B' \cdot C + A \cdot B' \cdot C' + A' \cdot B' \cdot C$$

$$A \cdot B \cdot C + A \cdot B \cdot C' + A \cdot B' \cdot C + A \cdot B' \cdot C' + A' \cdot B' \cdot C$$

$$y = AC$$

Minimal to Canonical form conversion. (SOP)

$$F = (A+B+C')(A'+C)$$

① Total no. of Variable  
② Variable absent in each minterm

$$= (A+B+C')(A'+B \cdot B' + C)$$

③

$$N_1 = A \vee B \vee C \vee$$

$$N_2 = A \vee B \times C \vee$$

$$= (A+B+C')(A'+C+B \cdot B')$$

$B \cdot B'$

$$= \boxed{(A+B+C')(A'+C+B)(A'+B+C)}$$

$$Ex = (A+B) \cdot (A'+C)$$

Distributive Law.

$$a+b \cdot c = (a+b)(a+c)$$

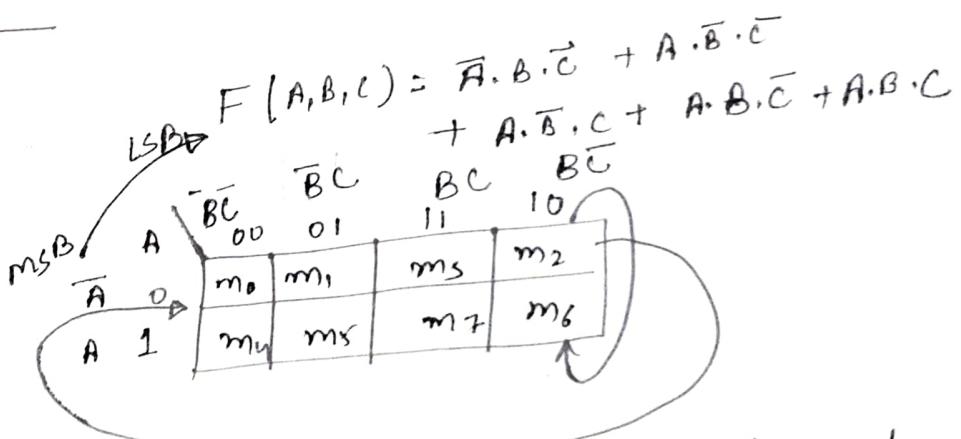
k-maps

Karnaugh map

Rule:  
~~pairwise of selection~~  
adjacency

$$F = A + BC$$

	A	B	C	Y
$m_0$	0	0	0	0
$m_1$	0	0	1	0
$m_2$	0	1	0	1
$m_3$	0	1	1	0
$m_4$	1	0	0	1
$m_5$	1	0	1	1
$m_6$	1	1	0	1
$m_7$	1	1	1	1



	$\overline{BC}$	$\overline{BC}$	$\overline{BC}$	$BC$	$BC$
$\overline{A}$	0	0	0	1	1
A	1	1	0	1	0
$m_0$	0	0	0	1	1
$m_1$	0	0	1	1	1
$m_2$	0	1	0	1	1
$m_3$	0	1	1	1	1
$m_4$	1	0	0	1	1
$m_5$	1	0	1	1	1
$m_6$	1	1	0	1	1
$m_7$	1	1	1	1	1

adjacency follow in such order

paving concept

$$Y = I + II$$

$$= A + BC$$

diagonal pairs X

minimally 2 1's  
higher 4 1's  
8 1's

16 1's Z

changing concept apply in whole considered block  $B: 0 \rightarrow 1$   
 $C: 0 \rightarrow 1$

### Example 4

$$(i) f(A, B, C) = \sum m(1, 3, 5, 7)$$

	A	B	C	
$m_0$			0	
$m_1$			1	
$m_2$			0	
$m_3$			1	
$m_4$			0	
$m_5$			1	
$m_6$			0	
$m_7$			1	

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

$A \ 0 \rightarrow 1$   
 $B \ 0 \rightarrow 1$

$$f(A, B, C) = C \text{ Any}$$

Total no. of variables  
 $n$

No. of cell =  $2^n$ .

$$(ii) F(A, B, C) = \sum m(0, 1, 2, 4, 7)$$

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

$$\begin{aligned} F(A, B, C) &= I + II + III + IV \\ &= \bar{A} \cdot \bar{B} \cdot \bar{C} + \bar{B} \cdot \bar{C} + A \cdot \bar{C} \\ &\quad + A \cdot B \cdot C \end{aligned}$$

$$(3) F(A, B, C) = \sum m(1, 3, 6, 7)$$

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

Redundancy

Redundancy theory

III X

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

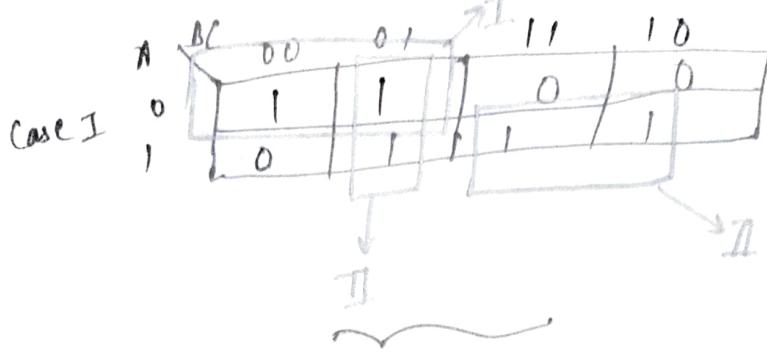
→ 3 variables

→ each var. is repeated twice

$$F = I + II + III = AB + \bar{A}C$$

→ 1 variable is complemented

$$F(A, B, C) = \Sigma m(0, 1, 5, 6, 7)$$



$$F(A, B, C) = \overline{A} + \overline{B}$$

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

$$\text{III}' \quad \text{II}^{\text{case}} \quad [1 \text{ ms} \mid 1 \text{ ms}]$$

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

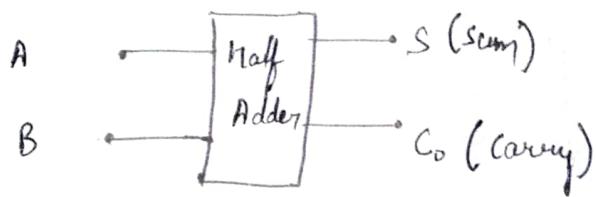
\* Note - the result is min. but may not be same or unique.

16 cell K-map =

$$F(A, B, C, D) = \Sigma m(0, 2, 3, 5, 7, 8, 10, 12, 14, 15)$$

		CD	
		A	B
		0	1
		0	1
		1	0
		0	1
		1	1

## Half Adder



→ Single bit no's

→ It doesn't take carry from previous sum

T.T -

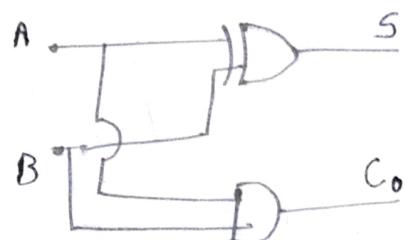
A	B	S	C <sub>0</sub>
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

$$S = A \oplus B$$

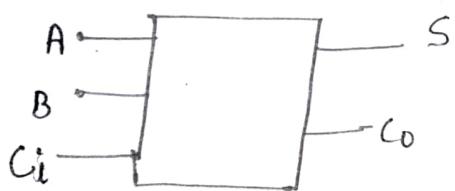
$$C_0 = A \cdot B$$

Internal.

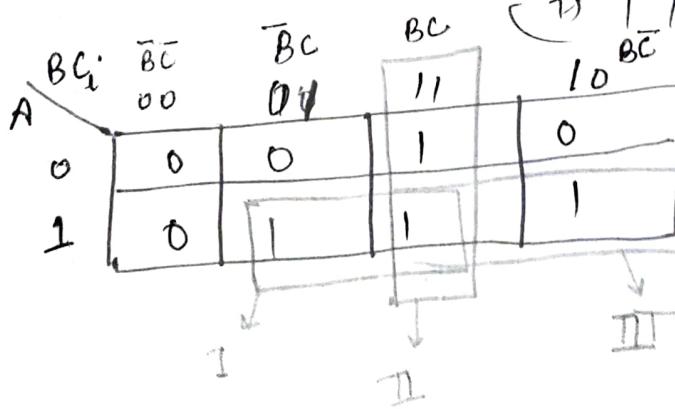
CKT



## Full Adder -



	A	B	C <sub>i</sub>	S	C <sub>0</sub>
(0)	0	0	0	0	0
(1)	0	0	1	1	0
(2)	0	1	0	1	0
(3)	0	1	1	0	1
(4)	1	0	0	1	0
(5)	1	0	1	0	1
(6)	1	1	0	0	1
(7)	1	1	1	1	1



Ans

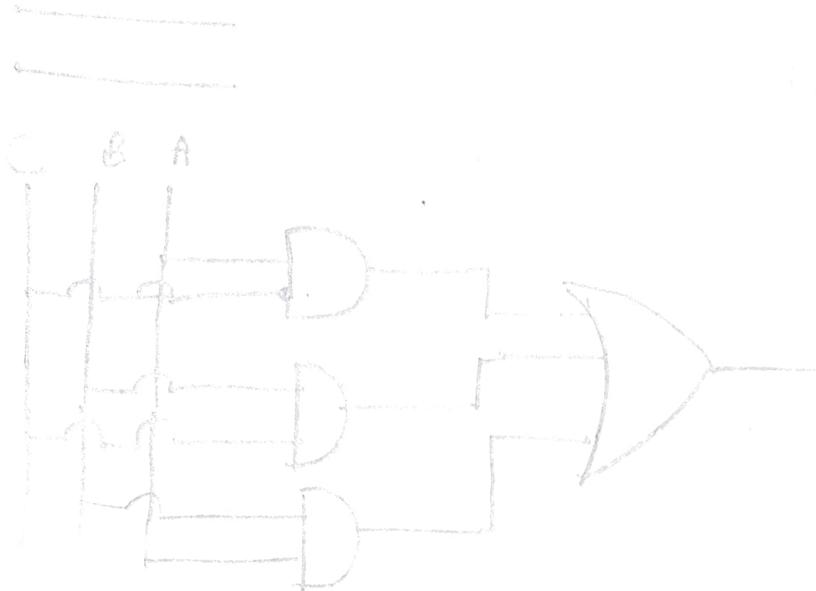
0	0	1	6
0	1	1	1

$$S = A \cdot C + B \cdot C + A \cdot B$$

$$C_0 = AB + C_i(A \oplus B)$$

$$AB + A\bar{B}C_i + \bar{A}BC_i$$

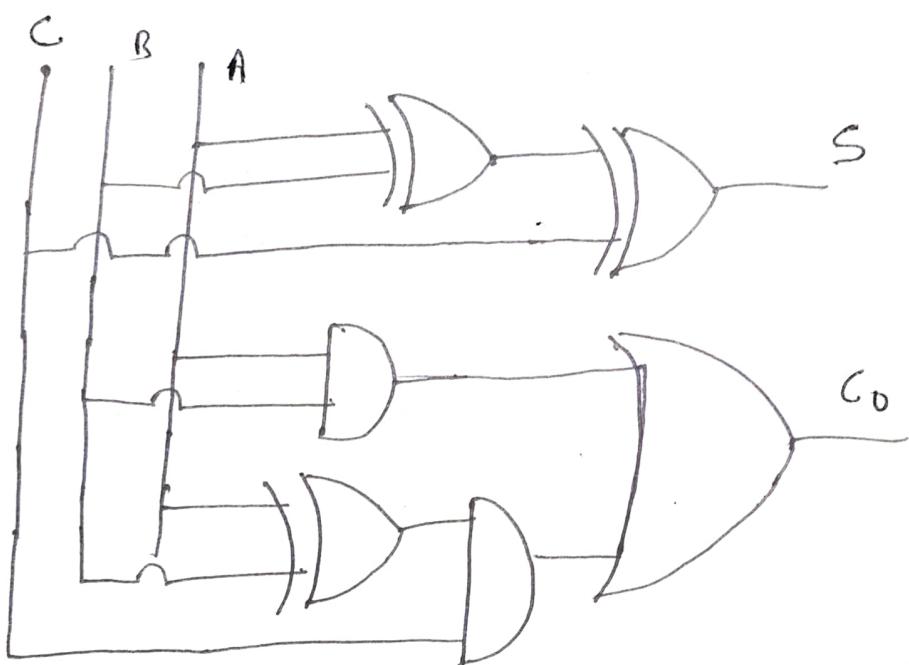
$$AB + C_i(A \oplus B)$$



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

ChK-board  
configuration

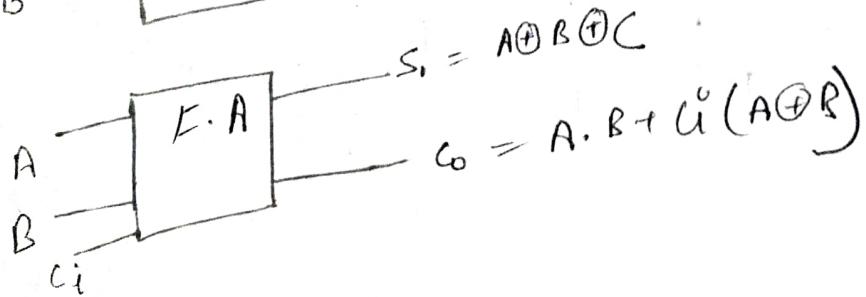
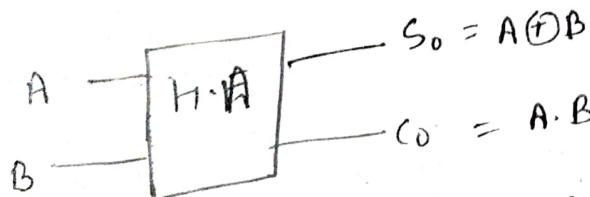
$$S = A \oplus B \oplus C_i$$

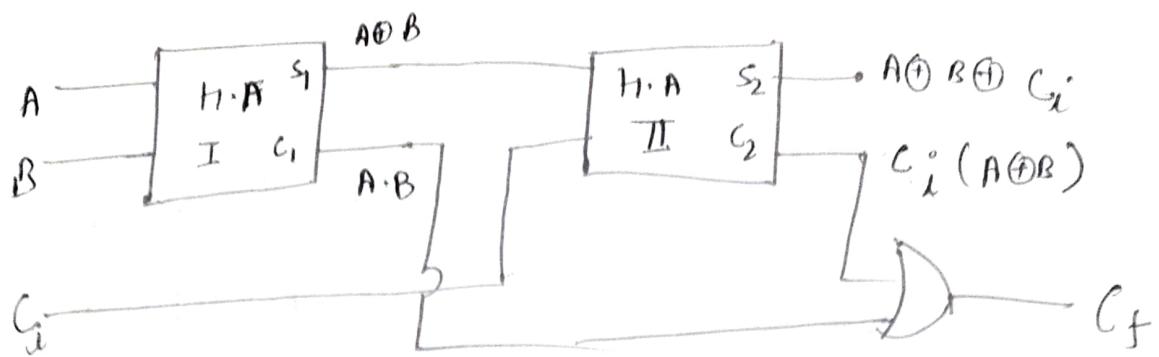


Full Adder using Half Adder

D/F Full Adder and Half Adder

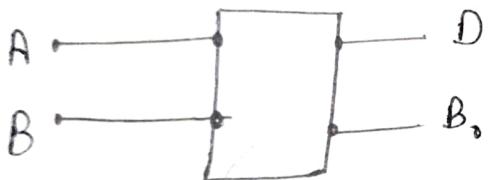
Carry doesn't shift  
from previous sum  
in Half Adder.





$$P.f \quad [A \cdot B + C_i(A \oplus B) = A \cdot B + BG + G_i A]$$

### HALF SUBTRACTOR



Half Subtractor

$$\begin{array}{r}
 2^1 \quad 2^0 \\
 10 \\
 - 01 \\
 \hline
 01
 \end{array}$$

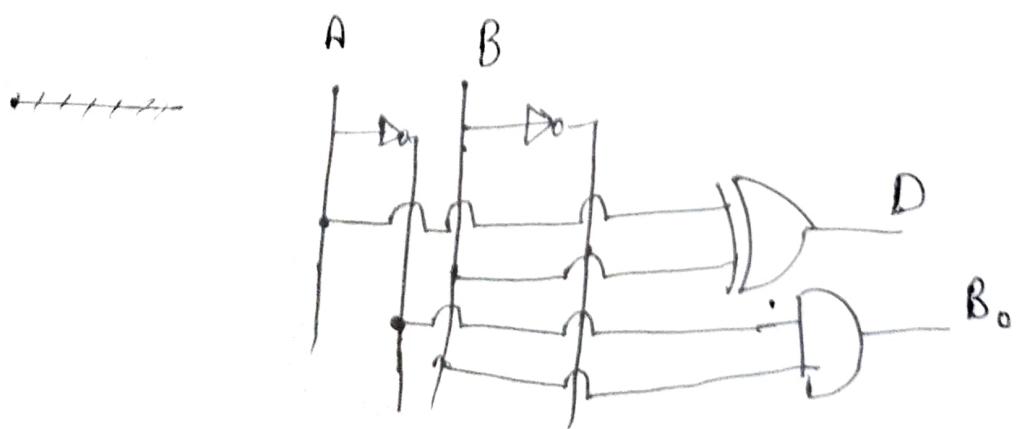
A	B	D	B <sub>o</sub>
0	0	0	0
1	0	1	0
1	1	0	1

odd  
gen's  
detector

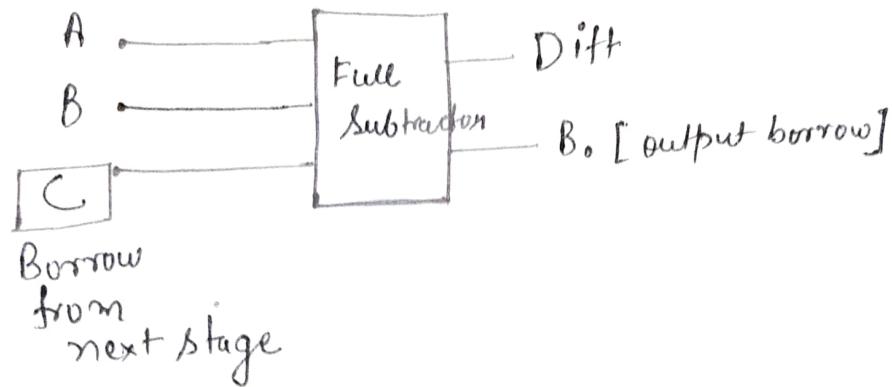
$$A \oplus B = D$$

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

~~$$\begin{array}{r}
 100 \\
 10 \\
 \hline
 10
 \end{array}$$~~



# FULL Subtractor.



A	B	C <sub>i</sub>	D	B <sub>o</sub>
0 - 0 = 0	0	0	0	0
0 - 0 = 0	0	1	1	1
0 - 1 = 1	0	1	1	1
0 - 1 = 1	1	0	1	1
1 - 0 = 1	0	1	0	0
1 - 0 = 1	0	0	0	0
1 - 1 = 0	0	0	0	0
1 - 1 = 0	1	1	1	1

D	0	1	0	1
0	0	1	0	1
1	1	0	1	0

D	00	01	11	10
0	0	1	1	1
1	0	0	1	0

$$\underline{B_o} = \overline{A} \cdot C_i + A \cdot B \cdot C$$

$$\begin{aligned} B_o &= \overline{A} \cdot \overline{B} \cdot C_i + \overline{B} \cdot C + \overline{A} \cdot B \\ &= \overline{A} (B \oplus C_i) + BC \end{aligned}$$

$$B_o = AB$$

$$\text{In case } \underline{B_o} = \overline{A} \cdot C + B \cdot C + \overline{A} \cdot B$$

Imp question -  
 Your Home work task to  
 make All adder using  
 NAND and NOR gate only

## Sequential Circuit

flip-flops, registers

In sequential Ckt, the present output depends on the present input as well as past output / outputs.

IN



$X = 0 \log$

$Y = 1 \log$

$Z = 0$

$W = 1$

[SR Latch]

Basic storage element

LATCH

Name suggests it latches "0" or "1"

$$R \cdot \overline{D} \cdot S + D \cdot \overline{S} = Q$$

disadvantages

"0" R  $\rightarrow$  Reset  
"1" S  $\rightarrow$  Set

$Q/P$  stat.  
 $Q_0$

if Rand's value changes,  
stored data gets change  
as automatically due to  
low clock is used

$S = 0, R = 1, Q = 0, \bar{Q} = 1$

$S = 0, R = 0, Q = 0, \bar{Q} = 1$

A	B	Y
0	0	0
0	1	0

1	1
---	---

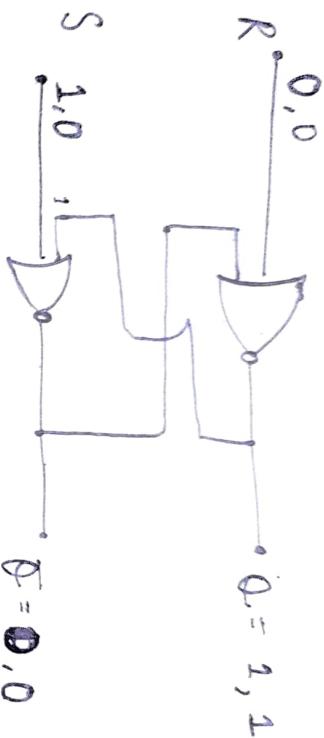


$R = 1, 0$

$\bar{Q} = 0, 0$



Case II

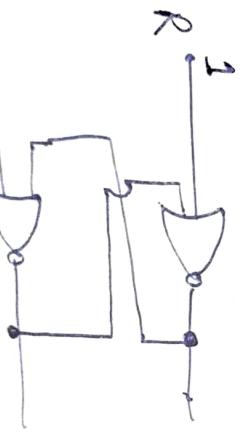


$S = 1, R = 0, \Theta = 1 \text{ & } \bar{\Theta} = 0$

Memory.

case III - Not used in SR latch

$S = 1, R = 1, \Theta = 1, \bar{\Theta} = 1$   
 $S = 0, R = 0, \Theta = 0 \text{ & } \bar{\Theta} = 1$   
 $\Theta = 1 \text{ & } \bar{\Theta} = 0$



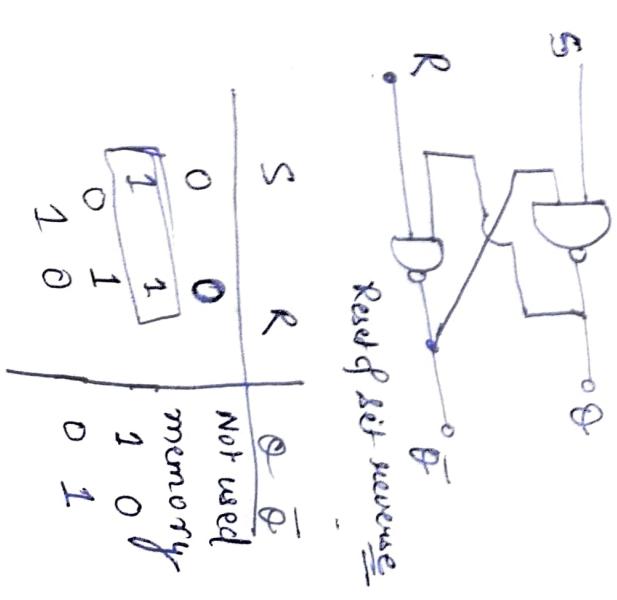
$S$

$\bar{D} = D$

1 Reset = 0  
 1 Set = 1

Table

$S$	$R$	$\Theta$	$\bar{\Theta}$
0	1	0	1
1	0	1	0
1	1	Not used.	
0	0	memory	

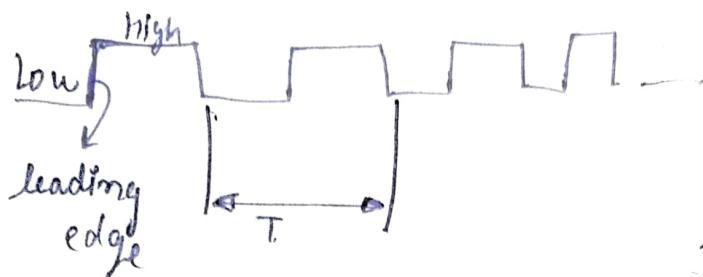


Reset Set reverse

$S$	$R$	$\Theta$	$\bar{\Theta}$
0	0	Not used	
1	1	memory	
1	0		
0	1		

what is clock

$C/K=1$  functional of flip flop.

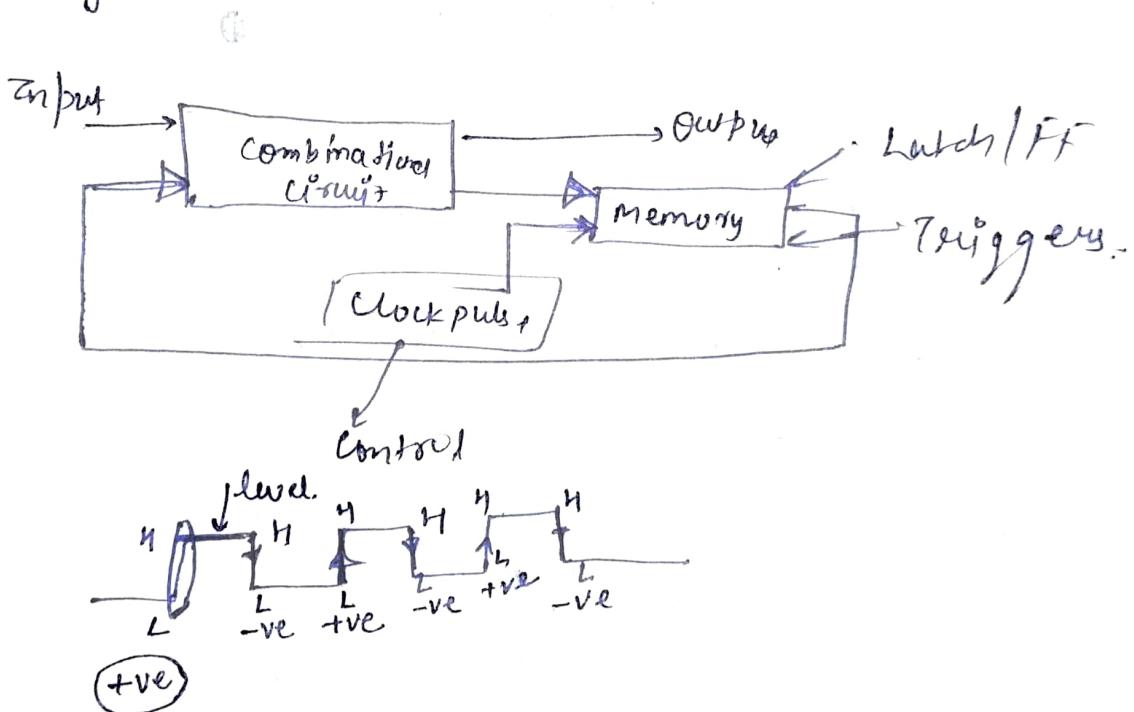


$$f = \frac{1}{T}$$

Duty = Ratio of high ↑

Ratio of low ↓

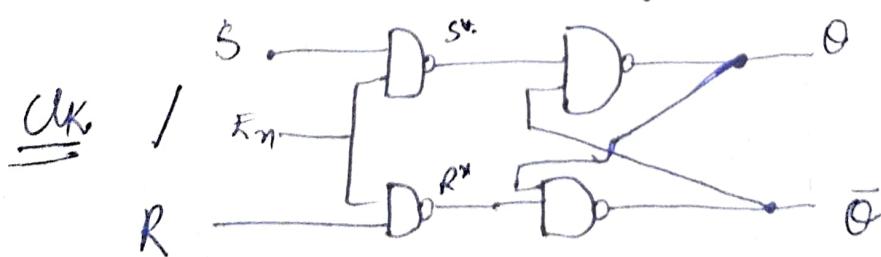
Triggering Method -



Diff b/w latch and Flip Flop.

②

level triggering



Flip - Flop

It become operational  
on edge triggering  
from L to H and  
H to L.

$$E_n = 1$$

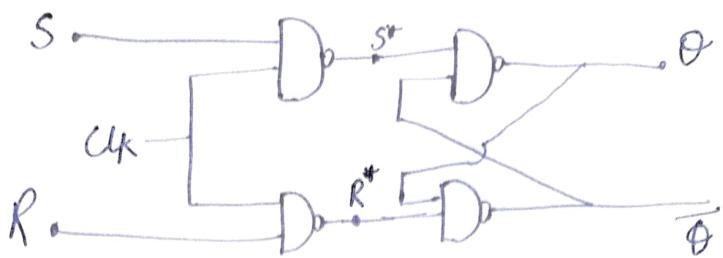
$$S^* = \bar{S}$$

$$E_n = 0$$

$$S^* = 1, R^* = 1$$

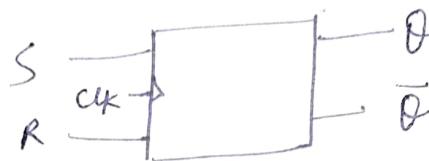
memory.

# SR flip flop



$$S^* = (\bar{S} \cdot \bar{A}_K) = \bar{S} + \bar{A}_K$$

$$R^* = (\bar{R} \cdot \bar{A}_K) = \bar{R} + \bar{A}_K$$



SR latch with NAND

$S^*$	$R^*$	$Q$	$\bar{Q}$
0	0	Not used.	
0	1	1	0
1	0	0	1
1	1	Memory	

CK	S	R	$Q$	$\bar{Q}$	$Q_{n+1}$
0	X	X	Memory		$Q_n$
1	0	0	Memory		$Q_n$
1	0	1	0	1	$S^* = \bar{S}$
1	1	0	1	0	$R^* = \bar{R}$
1	1	1	Not used		Invalid

Truth Table

remember

( $A_K = 1$ ) characteristic Table

$Q_n$	S	R	$Q_{n+1}$
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	X
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1

Excitation table

$Q_n$	$Q_{n+1}$	S	R
0	0	0	X
0	1	1	0
1	0	0	1
1	1	X	0

characteristics Eq<sup>n</sup>

$\Theta_n$	SR	00	01	11	10	
0	0	0	0	X	1	$I = 1$
1	1	0	0	X	1	$I = 1$

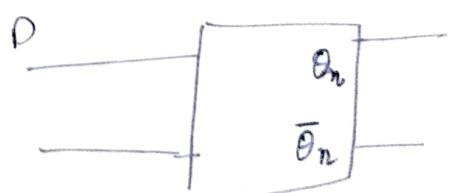
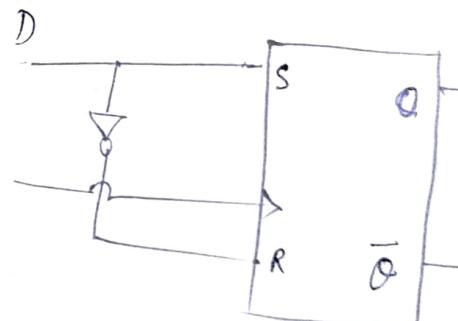
$$Y = SR + \Theta_n \cdot \bar{R}$$

$\Theta_n$	SR	00	01	11	10	
0	0	0	0	X	1	
1	1	0	0	X	1	

$$\Theta_{n+1} \quad Y = S + \Theta_n \cdot \bar{R}$$

D-flip flop       $D \rightarrow D + Q$   
 T<sub>T</sub>      SR flip flop

$\Theta_n$	Clk	S	R	$\Theta_{n+1}$
0		x	x	memory ( $\Theta_n$ )
1	0	0	0	memory ( $\Theta_n$ )
1	0	1	0	0
1	1	0	0	1
1	1	1	1	Invalid



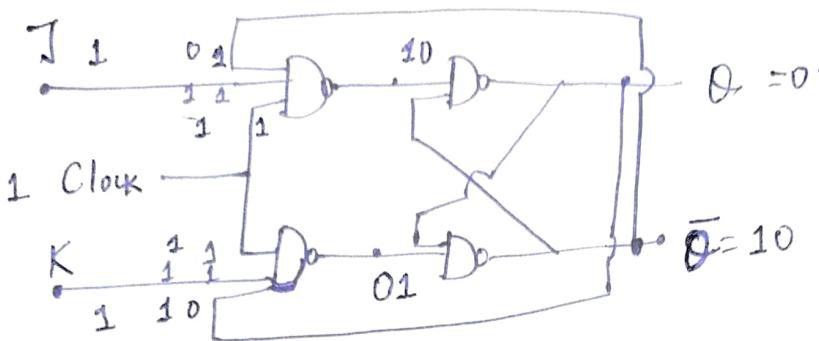
T.T for D-f.f.

Clk	D	$\Theta_{n+1}$
0	x	$\Theta_n$
1	0	0
1	1	1

## JK flip-flop

Advantage -  $S = J, R = K$   
useable form

T.T for SR flip-flop



Clk	S	R	$Q_{n+1}$
0	x	x	on
1	00	00	on
1	01	01	0
1	10	10	1
1	11	11	Invalid

use of invalid case

$JK = 0$  Memory

$Clk = 1$   $J = 1, K = 0, \theta = 1, \bar{\theta} = 0$

$Clk = 1$   $J = 0, K = 1, \theta = 0, \bar{\theta} = 1$

$\Rightarrow Clk = 1$   $J = 1, K = 1$   
assume  $\theta = 0 \neq \bar{\theta} = 1$   
 $\theta = 1 \neq \bar{\theta} = 0$

$\theta = 0, 1, 0, 1, \dots$   $\bar{\theta} = 1, 0, 1, 0, \dots$

$$\bar{\theta}_{n+1} = 0, 1, 0, 1$$

$$= \bar{\theta}_n$$

∴

Clk	J	K	$Q_{n+1}$
0	x	x	memory $Q_n$
1	00	00	on
1	01	01	0
1	10	10	1
1	11	11	$\bar{\theta}_n$ (Joggle)

$\theta_n$	JK	00	01	11	10
0		0	0	1	1
1		1	0	0	1

ch. Table

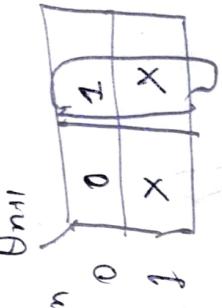
$\theta_n$	J	K	$\theta_{n+1}$
0	0	0	$\theta_n$ 0
0	0	1	0
0	1	0	1
0	1	1	$1 (\bar{\theta}_n)$
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	0

$$\theta_{n+1} = \bar{\theta}_n \cdot J + \theta_n \cdot K$$

### Excitation table

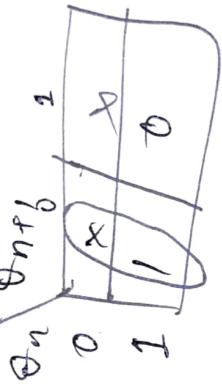
$\theta_n$	$\theta_{n+1}$	$J$	$K$
0	0	0	X
0	1	1	X
1	0	X	1
1	1	X	0

for J



$$J = \theta_{n+1}$$

for K

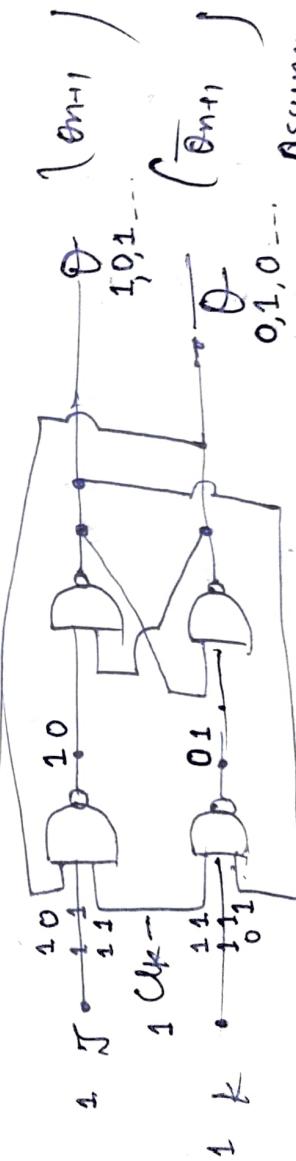


$$K = \overline{\theta_{n+1}}$$

### Race - Around Condition

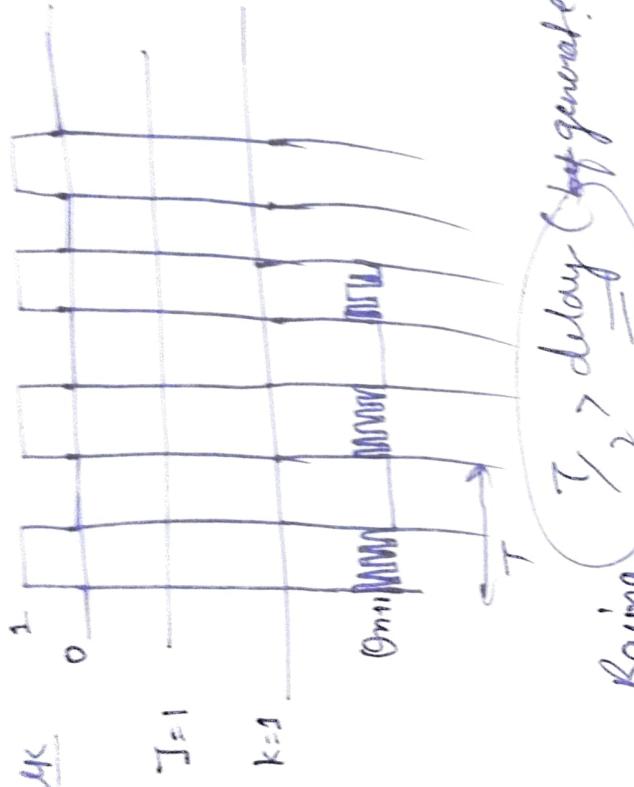
$J_K$  flip flop problem - Race-around condition

101010--  
010101--



$\theta = 1$   
 $\overline{\theta} = 0$

$C/K$	$J$	$K$	$\theta_{n+1}$	$\overline{\theta_{n+1}}$
0	X	X	$\theta_n$	$\overline{\theta_n}$
1	0	0	$\theta_n$	$\overline{\theta_n}$
1	1	0	1	0
1	1	1	0	1



Rising  $T_{1/2} >$  delay (not generate output)  
 Cond'n to over come racing.  
 Cond'n to over come racing. Output same  
 me time  
 ①  $T_{1/2} <$  propagation delay of 1T,  $> T_{1/2}$   
 half time period  
 of clock =

- ② edge triggering
- ③ Master Slave

toggle  
 T-flip flop  
 Controlled change  
 of output from  
 $0 \rightarrow 1$   
 $1 \rightarrow 0$



T-T

Clk	T	Q	$\bar{Q}$
0	X	=	=
1	0	=	=

$\bar{Q}$  =  $\bar{Q}$  (Toggle)