

16/01/2024

## logic gates

i) And  $\rightarrow$  conjunction

Output will be high  
if all the input high.

True = 1 Both input  
high - 5 volt high.

Low = 0  
High = 1

ii) OR  $\rightarrow$  disjunction



iii) not

### Truth table

Output  $31:85T = 2^n \rightarrow$  inputs.

For 2 inputs

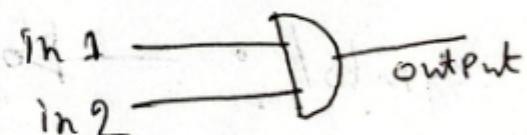
Logic expression:

$$C = A \cdot B$$

And:

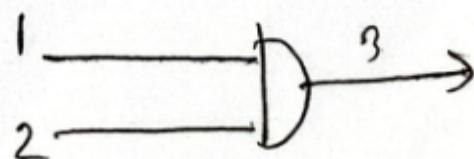
A	B	F
0	0	F
0	1	F
1	1	T

Logic Diagram:



Circuit Diagram:

Comes with pin config



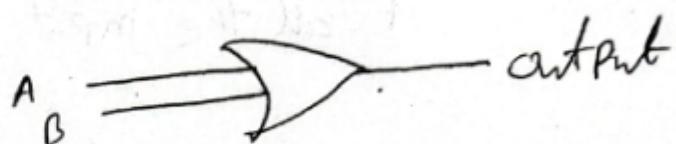
## OR Gate

Output will be low/false when all the inputs are 0 or low.

Truth table

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

symbol:

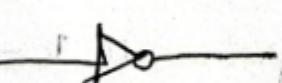


Logic expression:

$$Y = A + B$$

## Not gate

Opposite of input

Symbol: 

expression:  $\bar{A}$

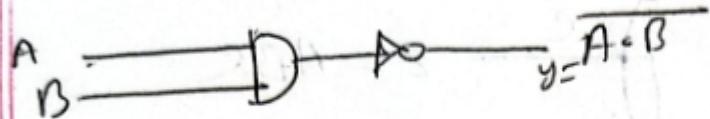
Truth table

a	b	Y
0	1	0
1	1	0

## Compound logic gate:

Nand:

$$\text{not} + \text{Nand} = \text{Nand.}$$

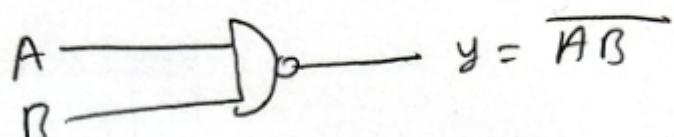


Ic

$$\text{And} = 7408$$

$$\text{OR} = 7432$$

OR,



Truth table

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

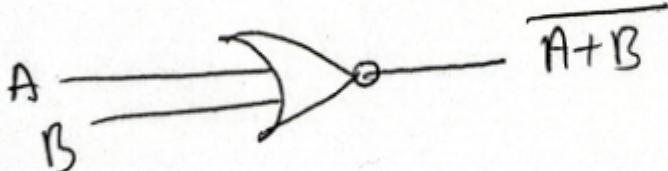
Nor :

$$\text{Op} + \text{Not} = \text{Nor}$$



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

Or,



### Circuit Diagram

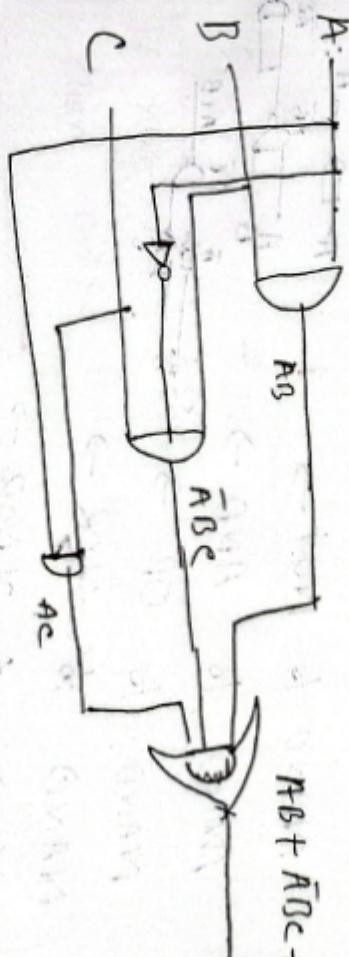
$$Y = AB + \bar{A}Bc + Ac$$

$$Z = ABC$$

$$\begin{array}{c} 1 \\ A \\ \hline 2 \\ B \\ \hline 3 \\ C \\ \hline Z = ABC \end{array}$$

$$\begin{array}{c} 1 \\ A \\ \hline 2 \\ B \\ \hline 3 \\ C \\ \hline Z \end{array}$$

$$\begin{array}{c} 1 \\ A \\ \hline 2 \\ B \\ \hline 3 \\ C \\ \hline Z \\ \hline \text{For OR} \end{array}$$



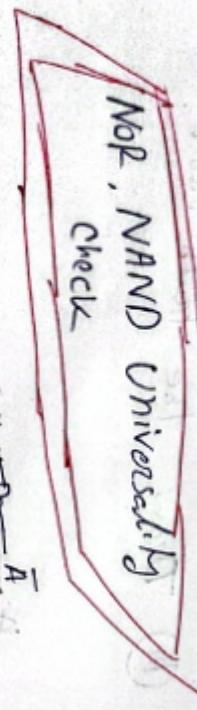
① Draw logic diagram

② Draw truth table.

③ Draw sum of products

$$2) \overline{AB} + \overline{ABC} + BC + D$$

① Draw circuit and truth table.



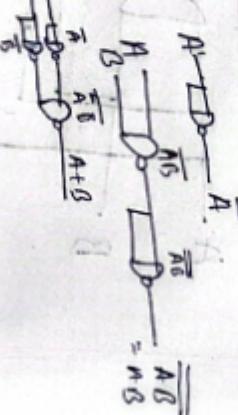
NAND to NOT  $\rightarrow$  1

NAND to AND  $\rightarrow$  2

NAND to OR  $\rightarrow$  3

NAND to NOR  $\rightarrow$  4

NAND to X-NOR  $\rightarrow$  5

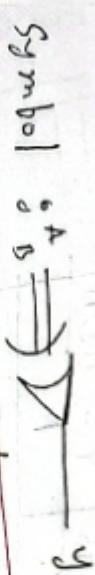


- NOR to NOT  $\rightarrow$  1
- NOR to AND  $\rightarrow$  2
- NOR to OR  $\rightarrow$  3
- NOR to NOR  $\rightarrow$  4
- NOR to X-NOR  $\rightarrow$  5

Exclusive OR (XOR / n-or) gate:



Output will be high / 1 when inputs are different  
Otherwise 0 / low.



If number of inputs ~~are~~ (1) is odd  
then output will be 1 (high)

Truth table

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

Truth table

A	B	C	Y = A ⊕ B ⊕ C
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	0

nor & input same total error

A	B	C
1	1	0

Ex-NOR

Output will be high when inputs are same

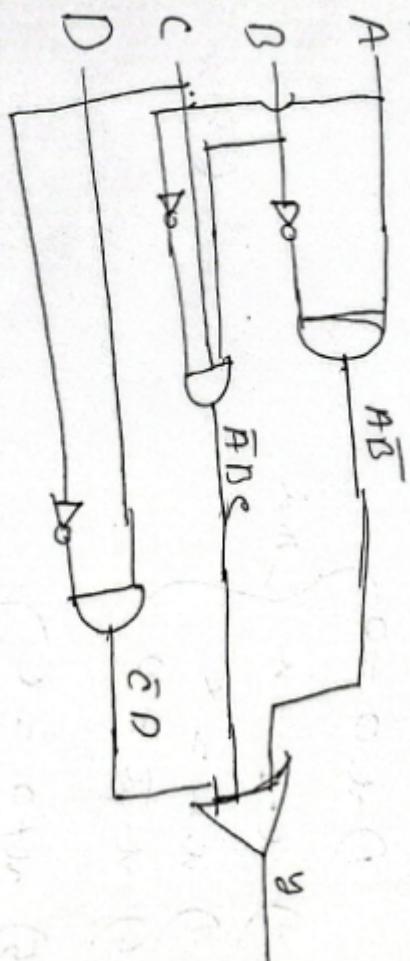
Symbol:  $\mu \Rightarrow \Delta$        $Y = \overline{A + B}$

A	B	$Y = A \oplus B$
0	0	0
0	1	1
1	0	1
1	1	0

It can also be done  
with 2 gate method,

$$Y = A\bar{B} + \bar{A}Bc + \bar{c}D$$

Logic Expression



## Simplified Expressions

### Boolean algebra

Distributive

$$a) w(y+z) = wy + wz$$

$$b) (w+y)(w+z) = ww + wz + yw + yz$$

$$1) w \cdot 0 = 0$$

$$\left. \begin{array}{l} 2) w \cdot 1 = w \\ 3) w \cdot w = w \\ 4) w \cdot \bar{w} = 0 \end{array} \right\} \text{Idemp}$$

$$\left. \begin{array}{l} 5) w+0 = w \\ 6) w+1 = 1 \\ 7) w+w = w \\ 8) w+\bar{w} = 1 \end{array} \right\} \text{Compl}$$

$$9) w+y = y+w$$

$$10) w \cdot y = y \cdot w$$

$$11) w+(y+z) = (w+y)+z$$

$$12) w(yz) = (wy)z = wyz$$

Associative Rule

$$14) w+xy = w \quad 15) a) w+\bar{w}y = w+y \quad b) \bar{w}+wy = \bar{w}+y$$

$$16) \bar{w}+y = \bar{w} \cdot \bar{y} \quad 17) \bar{w}\bar{y} = \bar{w} \cdot \bar{y}$$

De Morgan

Design a LC from truth table

$$n = \text{input} = 2^m$$

① we'll have only max input who has output 1/high

①

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

T <sub>0</sub>	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>
0	1	1	1	1	0	0	0
1	0	0	0	0	1	1	1
1	0	1	0	1	1	0	0
1	1	0	1	0	0	1	1
1	1	1	1	1	1	1	1

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

$$\begin{aligned}
 & \# Y = ABC + A\bar{B} \quad (\bar{A}, C) \\
 &= ABC + A\bar{B} (A + C) \\
 &= ABC + A\bar{B}A + A\bar{B}C \\
 &= ABC + A\bar{B} + A\bar{B}C \\
 &= A(C(\bar{B} + \bar{B})) + A\bar{B} \\
 &= A(C + \bar{B})
 \end{aligned}$$

$$\begin{aligned}
 & \therefore Y = \bar{A}\bar{B}C + A\bar{B}C + AB\bar{C} + A\bar{B}C \\
 & \Rightarrow C(\bar{A}\bar{B} + A\bar{B}) + A\bar{B}(C + C) \\
 &= A\oplus B = A \oplus B
 \end{aligned}$$

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

① Suppose  $a, b, c$  are inputs and  $w$  is the output. Output  $w$  will be high only when majority of inputs are high.

$w = \bar{a}bc + ab\bar{c} + abc + a\bar{b}\bar{c}$

→ Truth table

→ sum of product from TT

→ simplify.

→ direct.

$A$	$B$	$C$	$w$
0	0	0	0
0	0	1	0
0	1	0	0
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

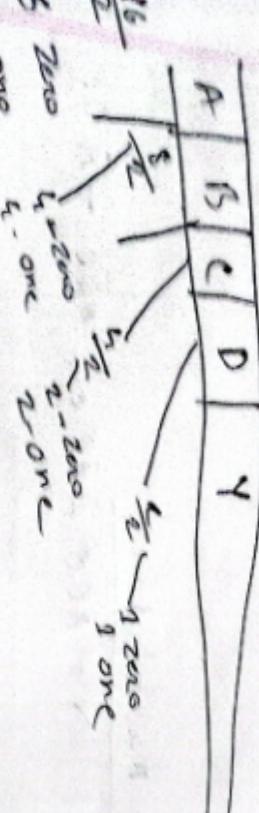
Majority  
circuit

$$\overline{F \text{ or majority low}} = \bar{A}\bar{B}C + A\bar{B}C + AB\bar{C} + ABC$$

If  $S = 1$  then  $Y = 1$

For input 4

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



### Statement:

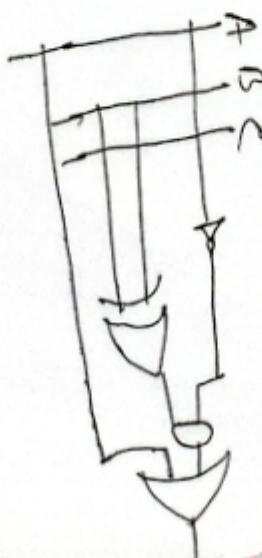
ABc are inputs and n is output, n will be equal to 1 when B and C are same and remain high when B and C are differ.

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

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

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

MSB      LSB  
4      2      1  
1      0      1 - 5



## K-map Algorithm

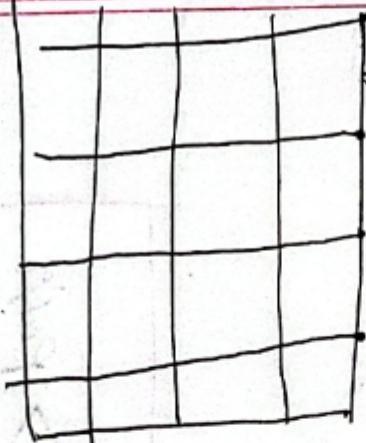
[Step 1] Construct the K-map and place 1 in those squares corresponds to the 1 in the truth table or in other square.

[Step 2] Examine the map for adjacent 1 and loop those 1 that are not adjacent to any other 1 that are called isolated 1.

10 11 12 13  
00 01 02 03

00  
CD'A'

A'B  
0  
11 AB  
A'B



2<sup>in</sup> input stereo k-map  $\rightarrow$  8<sup>in</sup>  
3<sup>in</sup> " " " " " " 16<sup>in</sup>

Isolated  $\rightarrow$  reduce Q input

Pair  $\rightarrow$  reduce T input

Quad  $\rightarrow$  reduce Q input

Octet  $\rightarrow$  reduce Q input.

Column স্বাধীন এবং Column স্বাধীন

C	1	0	0	1
C	1	0	0	0

\* If 2<sup>in</sup> k-map has different Pairs across  
Row Path sum = Stereo stereo ( $A\bar{B} + A\bar{B}$ )

Each Path sum = Stereo stereo

- Step 3:** Next look for those T that are adjacent to only other 1 loop any pairing contain such 1
- Step 4:** Loop any octet even if it contains 1 that have already been looped.

**Step 5:** Find Quads

**Step 6:** Find any pairs

**Step 7:** From the OR sum of all four generators each loop.

isolated > adjacent pair > octate > quad > pair

\* ये एक लॉप है  
क्योंकि बीमा

Q. Design a logic circuit whose output is High only when majority input low.

K-map

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

$\bar{A}\bar{B}$	$A\bar{B}$	$A\bar{B}$	$\bar{A}B$
1	1	0	1
1	0	0	0

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

LSB side  
true or  
false side

### Example - 4:

A 4 bit binary number is represented as D,C,B,A where D,C,B,A represent the individual bits and A is equal to the LSB. Design a logic circuit that will produce a high output whenever the binary number is greater than 0010 and less than 1010.

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

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

$\bar{C}\bar{D}$	$\bar{A}\bar{B}$	$AB$	$AD$
0	0	0	0
1	1	1	1
0	1	0	0
1	0	0	0

### Example 5:

A 4 bit binary represented as D,C,B,A where D,C,B represent the individual bits which are equal to the LSB. A logic circuit that will produce a high output whenever the binary number is greater than 011 and less than 101 (all inputs are high).

Ex 6

Design a logic circuit which has 4 inputs A, B, C, D and output Y. That is only high when input A is high at the same time that at least two other inputs are high.

$\bar{CD}$	$\bar{AB}$	$AB$	$\bar{ACD}$
00	1	1	1
01	1	1	1

$$Y = ABC + ACD + ABD$$

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

## Chapter - 6

Addition:

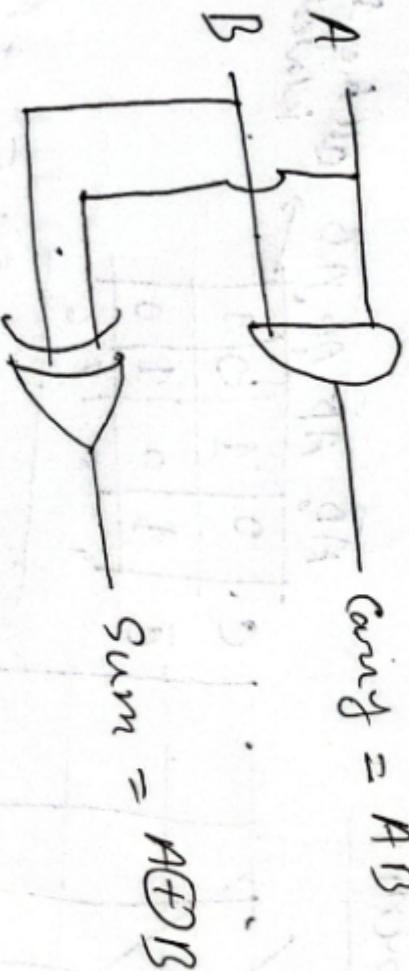
$$\begin{array}{r}
 + \\
 0 \\
 \hline
 0 \\
 \xrightarrow{\text{Carry}} \text{sum}
 \end{array}
 \quad
 \begin{array}{r}
 + \\
 1 \\
 \hline
 0 \\
 \xrightarrow{\text{Carry}} \text{sum}
 \end{array}$$

Carry

Truth table for 2 input sum & carry

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

2 bit is always Half-Adder  
 3 bit is always Full-Adder



Half Adder

$$\text{Ans}$$

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

$$\text{Ans}$$

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

Design full Adder

A, B and C<sub>in</sub>

A	B	C <sub>in</sub>	Sum	C <sub>out</sub>
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

For Sum:

$$\bar{A}\bar{B}, A\bar{B}, AB, \bar{A}B$$

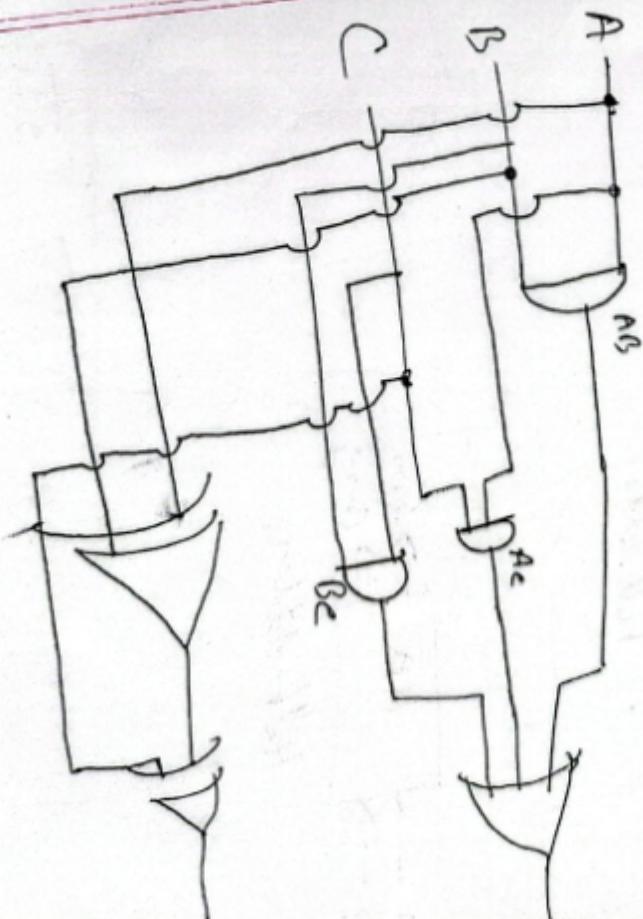
all three are isolated

C	$\bar{A}\bar{B}$	$A\bar{B}$	$AB$	$\bar{A}B$
0	1	0	0	1
1	0	1	1	0

C	0	1	0	1
$\bar{C}$	1	0	1	0

$$\begin{aligned} &= \bar{A}\bar{B} C_{in} + \bar{A}B \bar{C}_{in} + AB \bar{C} + A\bar{B} \\ &= \bar{A}\bar{B} C_{in} + \bar{A}B \bar{C}_{in} + A(\bar{B}C + \bar{B}C) \\ &= A(\bar{B} \oplus C) + A(\bar{B} \oplus C) \\ &\equiv A(\bar{B} \oplus C) \end{aligned}$$

1 bit full adder  
gives input to 2 bit full adder

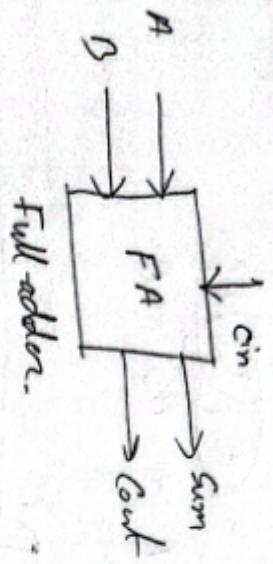


For cont:

C	$\bar{A}\bar{B}$	$A\bar{B}$	$AB$	$\bar{A}B$
0	0	0	0	0
1	1	1	1	1

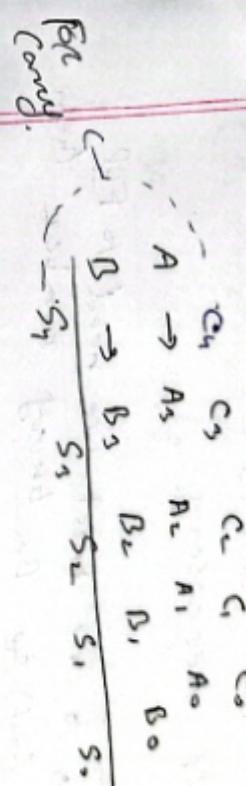
$$AB + AC + BC$$

## Block Diagram

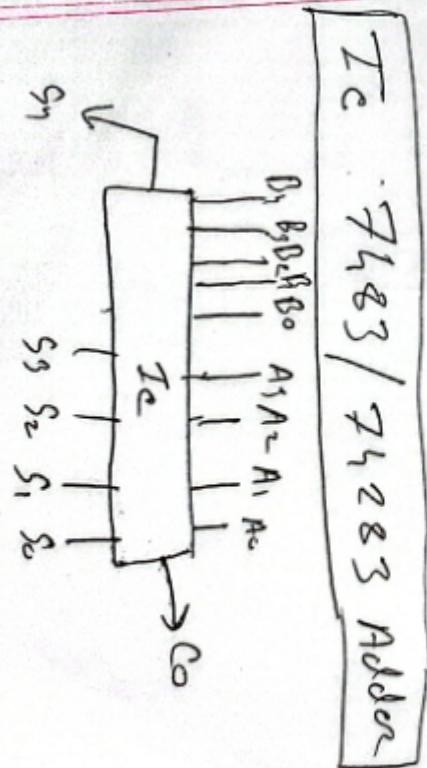
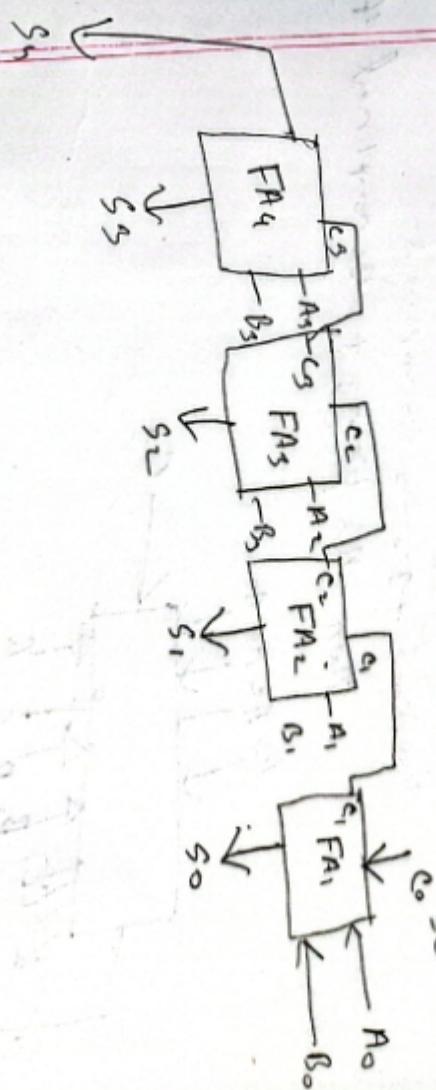


(1 bit)

## 4 bit Parallel Adder:



$A = \text{Input}$   
 $B = \text{Input}$   
 $C_{in} = \text{Carry}$   
 $S = \text{Output}$   
 at first carry = 0 & no



Q Design a 4 bit Subtractor Using IC Nand

Ans you can use other logic  
gate if necessary.

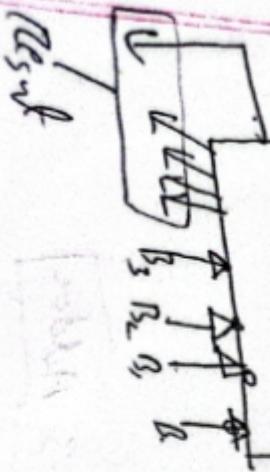
$A - B = A + (-B)$  → To find first we need to flip then

$(\text{complement}) \rightarrow \text{ex complement}$   
 $\text{As complement} \rightarrow \text{ex complement of } B$

Then it's binary +

$A_3 A_2 A_1 A_0$

$B_3 B_2 B_1 B_0$



& Q

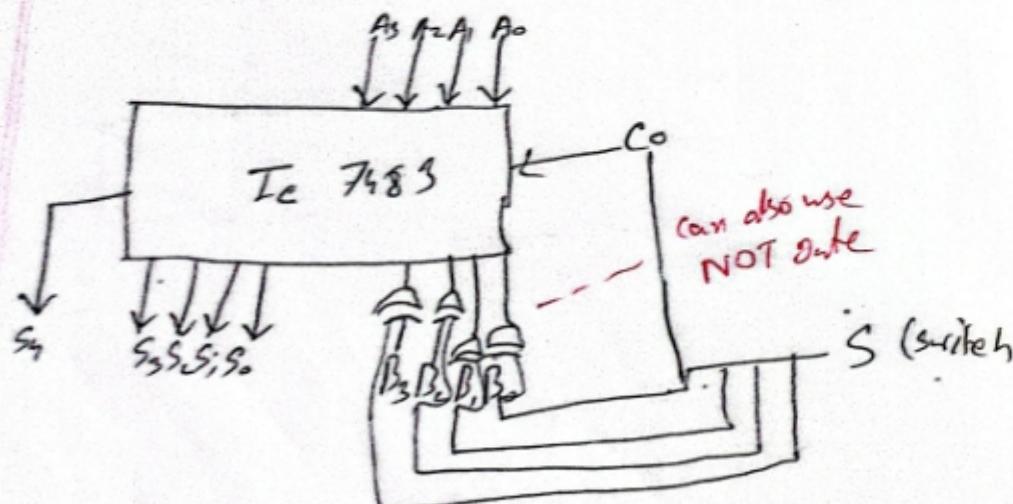
For subtraction To flip the binary number  
(is complement) we can either use NOT gate  
or N-OR gate

⇒ Switch off selected inputs  
 " ON " Subtraction  
 \* Design a 4 bit Adder/Subtractor using  
 IC Number 7483. You can use other  
 logic gate if necessary.

→ (Switch)

$$\text{If switch} = 0 \quad (A + B)$$

$$\text{If switch} = 1 \quad (A - B)$$



7483 = 4 bit Parallel adder

### Operations

If  $S=0$  then 1st input are  $A_3 A_2 A_1 A_0$   
 2nd " "  $B_3 B_2 B_1 B_0$

and  $C=0$

This is addition

If  $(S=1)$  then 1st input  $\overline{A_3} \overline{A_2} \overline{A_1} \overline{A_0}$   
 2nd "  $\overline{B_3} \overline{B_2} \overline{B_1} \overline{B_0}$

and  $C=1$

This is subtraction

BCP Addition

0110 0001 ← 18  
1011 1110 ← 57

### BCD Addition

$\begin{array}{r} 1110 \\ \times 110 \\ \hline 1100 \end{array}$

Hin

$$4+3=7 \leq 9$$

But when the sum of the numbers

does transfer more 9 BCD count.  
be done so we need to  
add 4 bit (0-15) to 9 BCD (0-9)  $\oplus_{BCD}$

If sum is more than add 6

*Zf Sun. 67. Do nothing*

କେବଳ ଏହାରେ କିମ୍ବା ଏହାରେ  
କିମ୍ବା ଏହାରେ କିମ୍ବା ଏହାରେ

Order number	1	0001	0110	0110
	6	0110	0110	0110
	4	0001	0110	0110
	1	0001	0110	0110

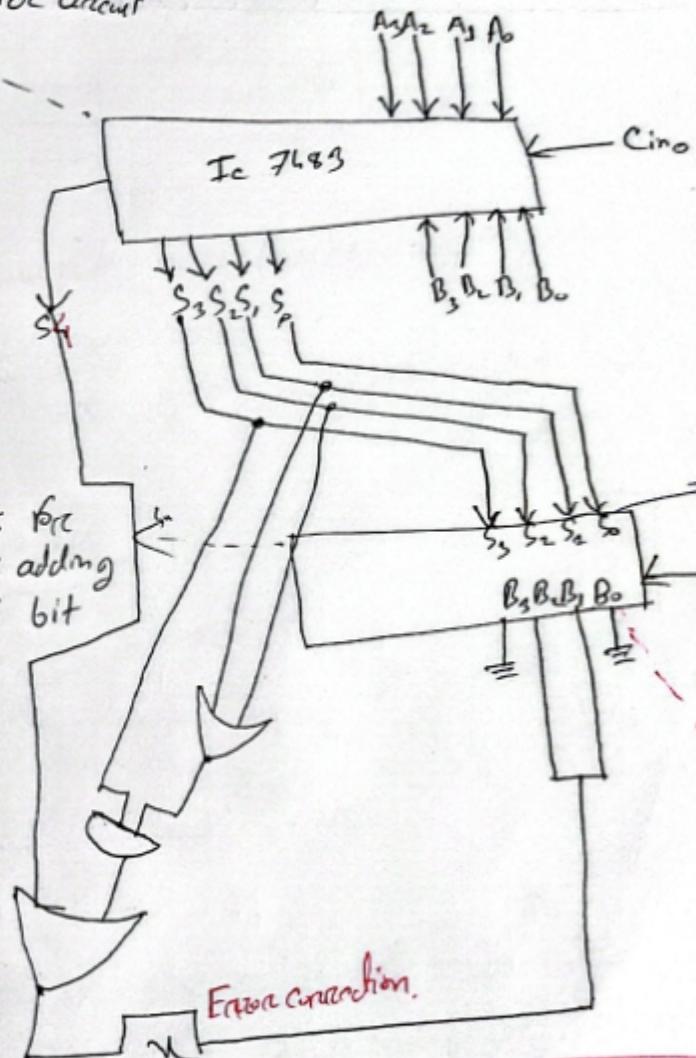
$$6 < 11 = g + \zeta$$

Q1 Design BCD adder using IC 7483  
 Can use other gate if necessary.

$BCD \rightarrow \text{Binary Coded Decimal}$

0 → 0000
1 → 0001
2 → 0010
3 → 0011
4 → 0100
5 → 0101
6 → 0110
7 → 0111
8 → 1000
9 → 1001

Main Adder circuit



Using this for  
 error finding. for adding  
 those extra 6 bit

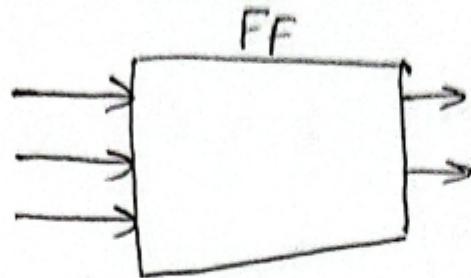
These are  
 input can  
 use  $A_i$   
 of  $S$

→ Here  
 $B_3$  is 2/  
 Because  
 0110

$$x = S_4 + S_3(S_2 + S_1)$$

# FLIP-FLOPS

One or many inputs

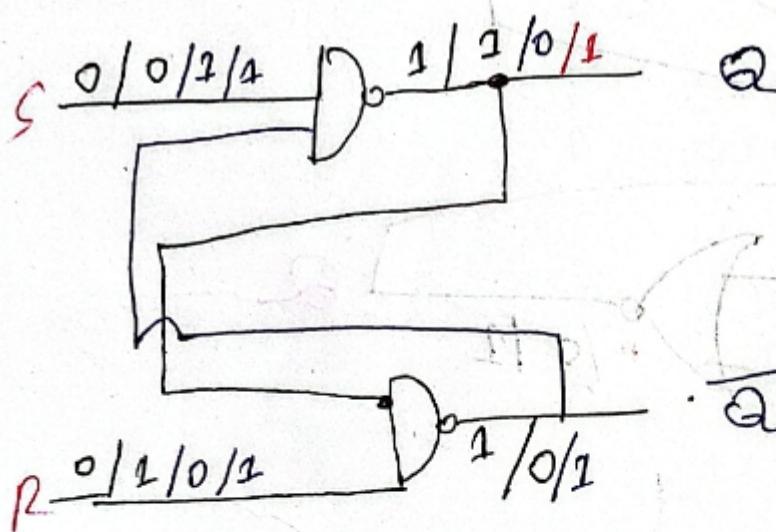


One output is opposite of another output

NAND LATCH: ( $S, R$ ,  $S, C$ )

input  $S = 0$   $R = 1$  output automatically is 2625

Truth table



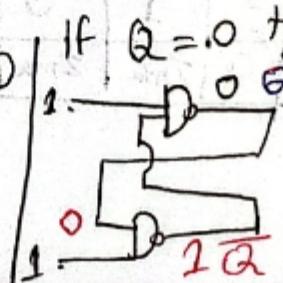
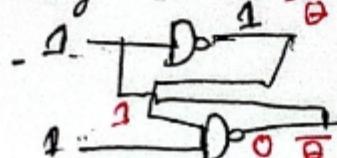
S	C	N	Q	$\bar{Q}$
0	0	1	0	0
0	1	1	1	0
1	0	1	0	1
1	1	0	Ne	Ne

Step by step logic of latch

যখন  $S=1$  হলে  $Q$  এর পরিমাণ পরিমাণ পরিমাণ পরিমাণ পরিমাণ

এ ক্ষেত্রে প্রথম  $Q$  এর value ১ হবে ক্ষেত্রে ১

If initially  $Q=1$  then  $\bar{Q}=0$

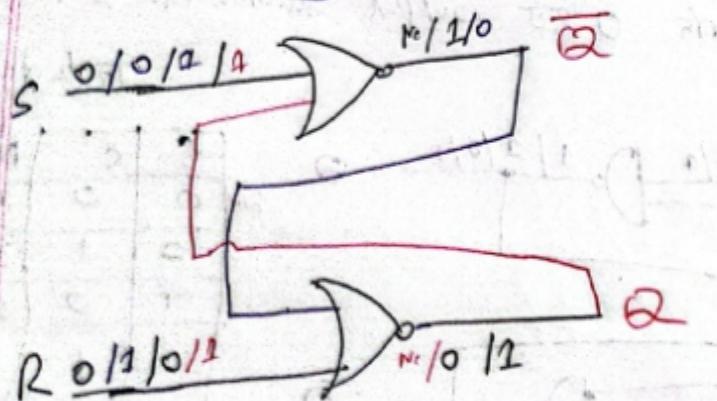


Q	$\bar{Q}$	Invalid
0	0	Invalid
1	0	
.	.	
.	.	

NOR  $1,1 \rightarrow$  Invalid  
NAND  $0,0 \rightarrow$  Invalid



### NOR Latch



S	R	NOR	$\bar{Q}$	Q
0	0	1	1	0
0	1	0	0	1
1	0	0	1	0
1	1	0	1	1

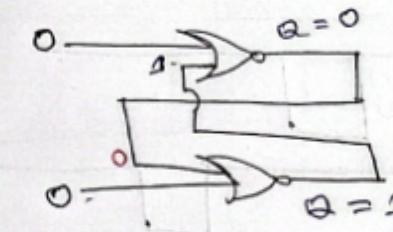
Invalid

NOR  $\Rightarrow$  1  $\oplus$  1  $\oplus$  0  
 $\oplus$  0 = 1

যেহেতু . 2'4 Input  $\Rightarrow$  0 করে Output force 0

যান না তৈরি করে 2'4

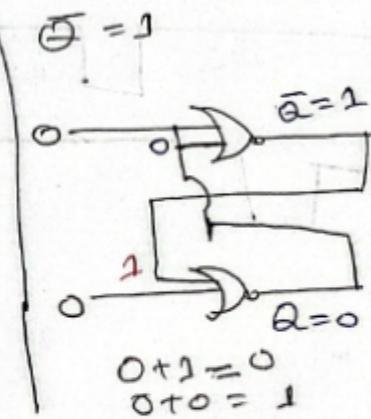
Initially  $Q = 1$  then  $\bar{Q} = 0$



$$0+1=0$$

$$0+0=1$$

again initially  $Q = 0$  then

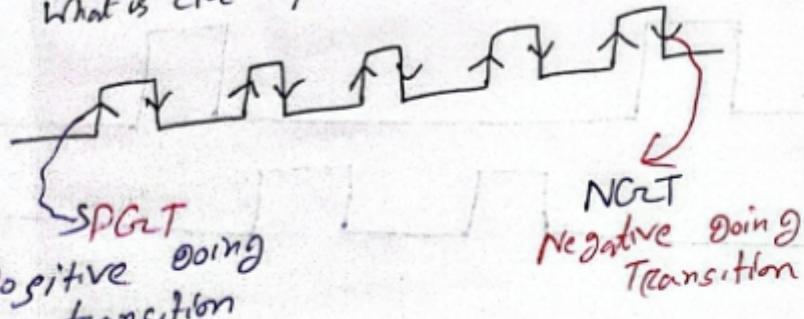


$$0+1=0$$

$$0+0=1$$

## Clock SC FlipFlop

What is clock?



SPGT  
Positive going transition

NCNT  
Negative going Transition

Short cut  
S & R  $\Rightarrow$  J & K Input for  
Output  $\Rightarrow$  ultra fast

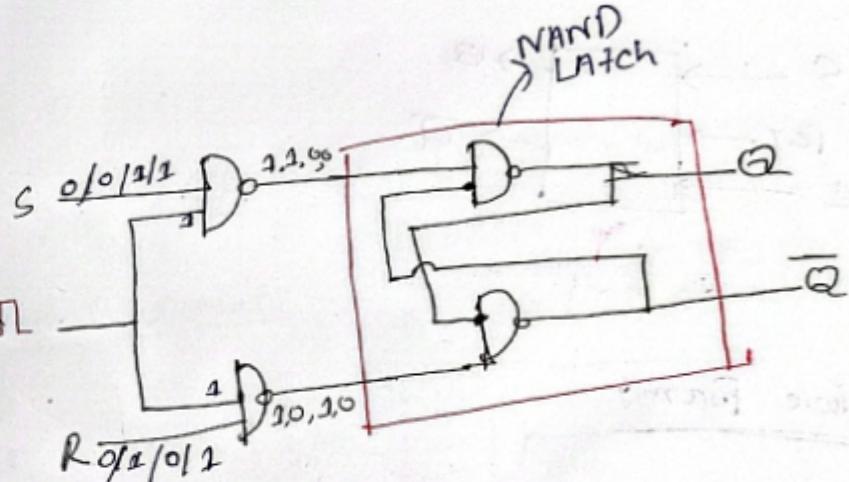
Clocked S-R FlipFlop

J & K output  
Nand gate

S	R	Q	$\bar{Q}$
0	0	1	0
0	1	0	1
1	0	0	1
1	1	nc	nc

SRIN  
Clock  
2 inputs  
2 outputs  
always  
1 error

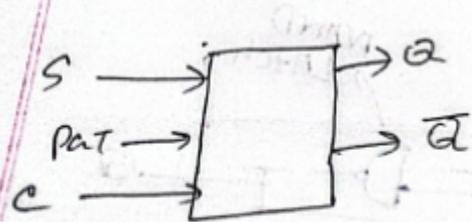
PORT 5L



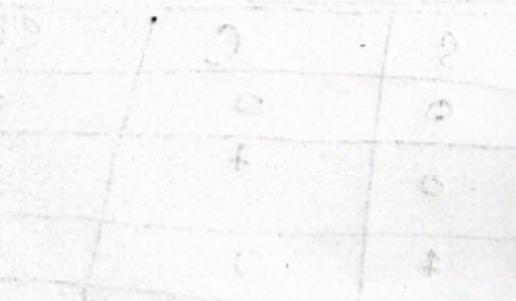
Truth Table:

S	C	CLK	Q	$\bar{Q}$
0	0	$\uparrow$	No change	
0	1	$\uparrow$	0	1
1	0	$\uparrow$	1	0
1	1	$\uparrow$	Invalid	

### Block Diagram:

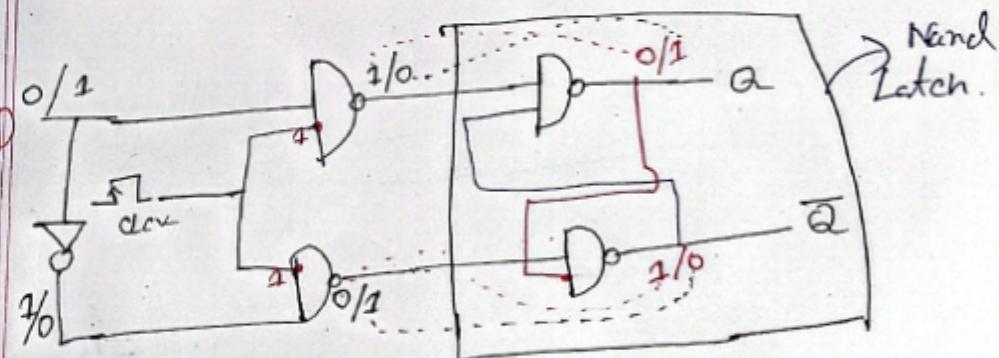


### Wave Forms:



### Clocked D-FlipFlop

DFlipFlop  $\rightarrow$  input  $\rightarrow$  set  
not gate  $\rightarrow$  2M 2M 2P  
clock  $\rightarrow$  off always 1 input



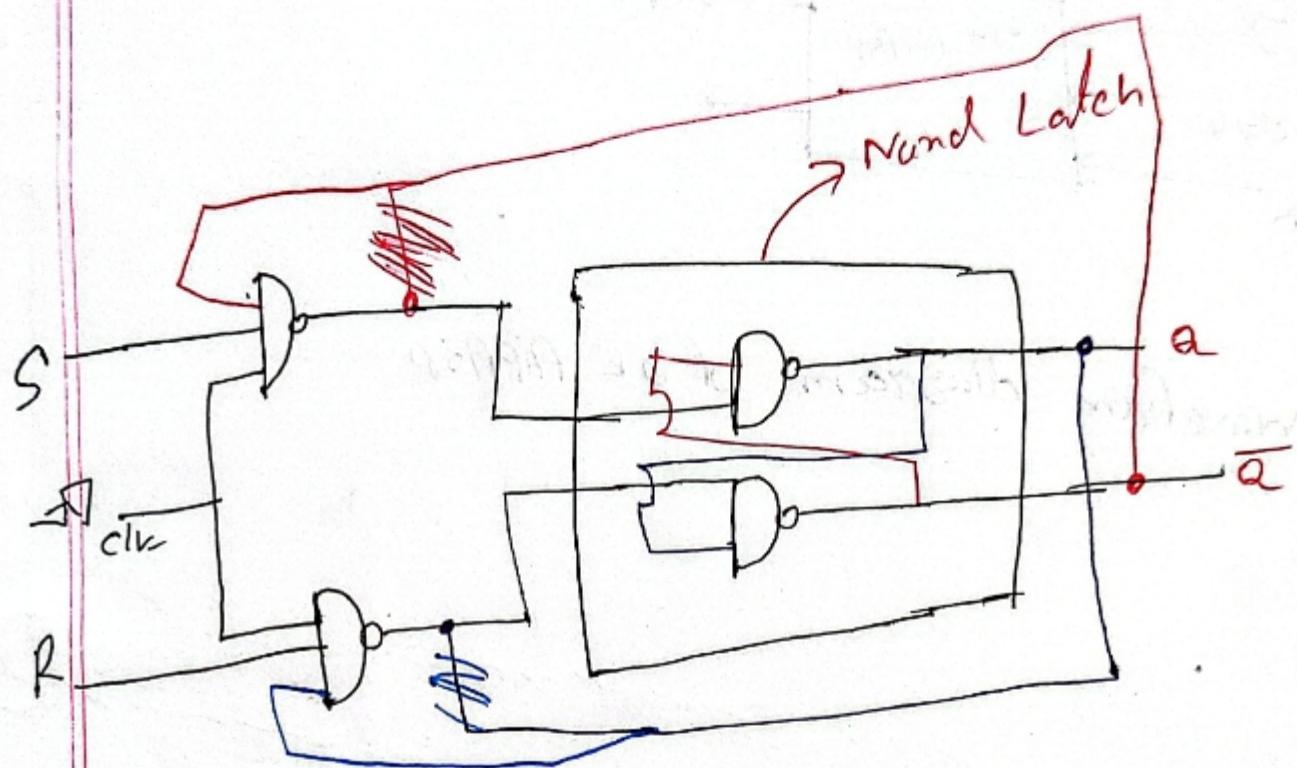
### Truth table:

D	Clk	Q	$\bar{Q}$
0	↑	0	1
1	↑	1	0

### Block Diagram



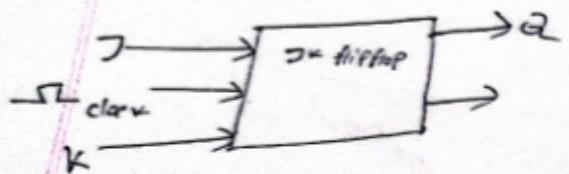
## Jk - Flipflop



Truth Tables

J	K	clk	Q	$\bar{Q}$
0	0	↑	Ne	
0	1	↑	0	
1	0	↑	1	
1	1	↑	Toggle	

## Block Diagrams



waveform diagram of D flip-flop



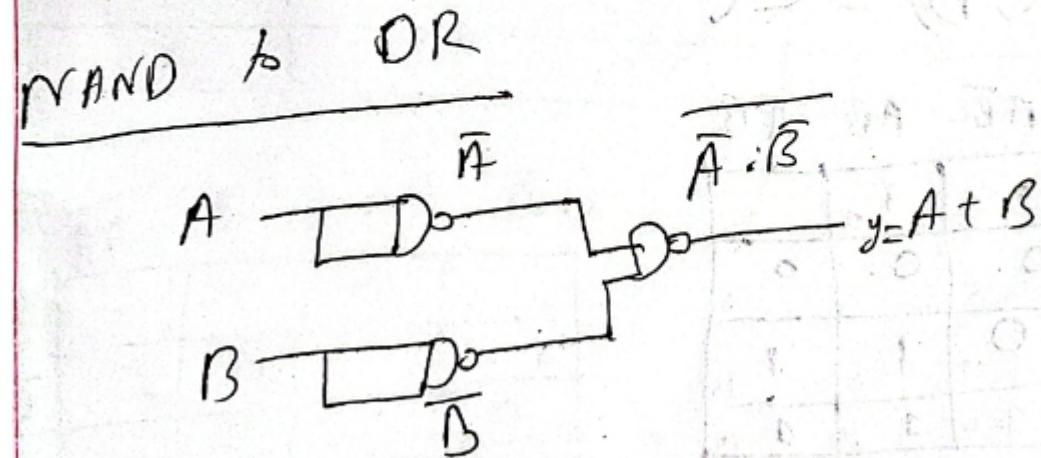
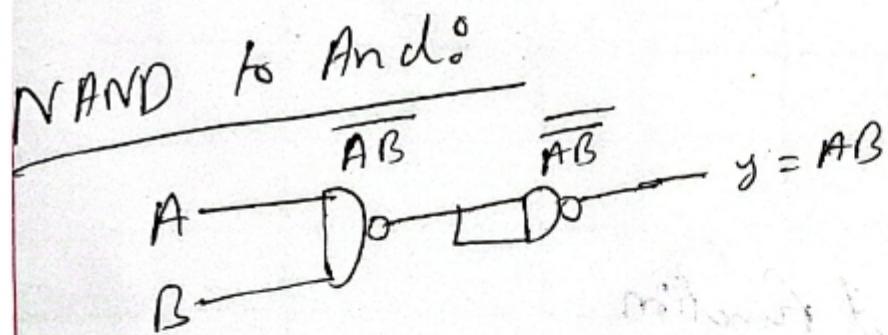
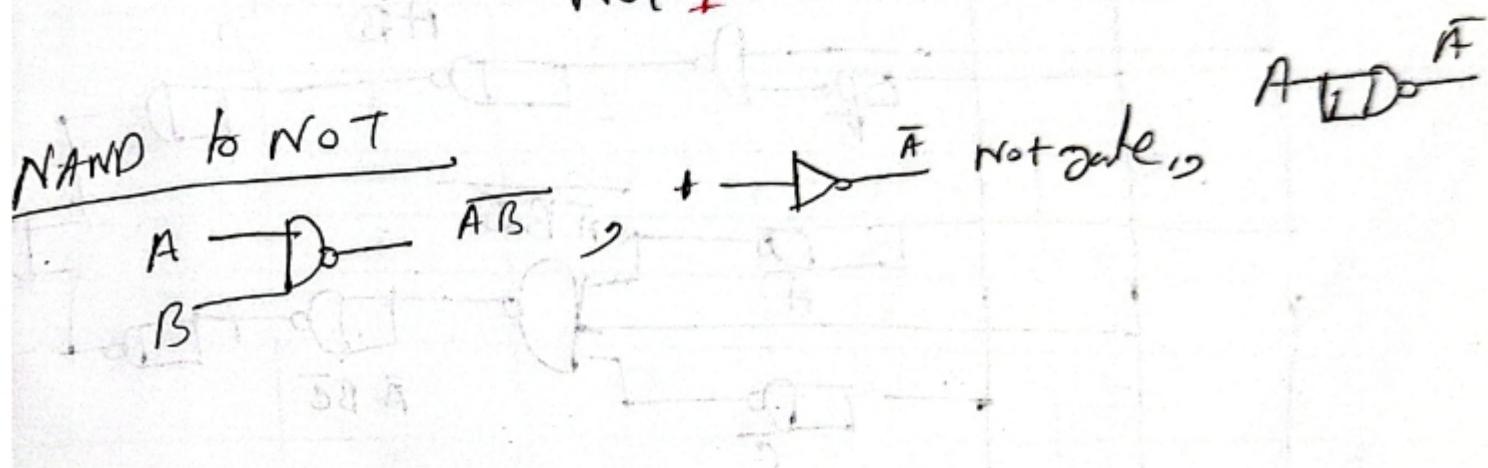
Table - 1

S	C	D	Q	Q-bar
0	0	0	0	1
1	0	0	1	0
0	1	0	0	1
1	1	0	1	0
0	0	1	1	0
1	0	1	0	1
0	1	1	0	1
1	1	1	1	0

Sec C - ct1

# Discuss the Universality of NAND Gate.

Nand gate  $\leftarrow$  And  $\rightarrow$  <sup>3 nand gate</sup> NOR  $\rightarrow$  <sup>2 nand gate</sup> NOT  $\rightarrow$  <sup>1 nand gate</sup>

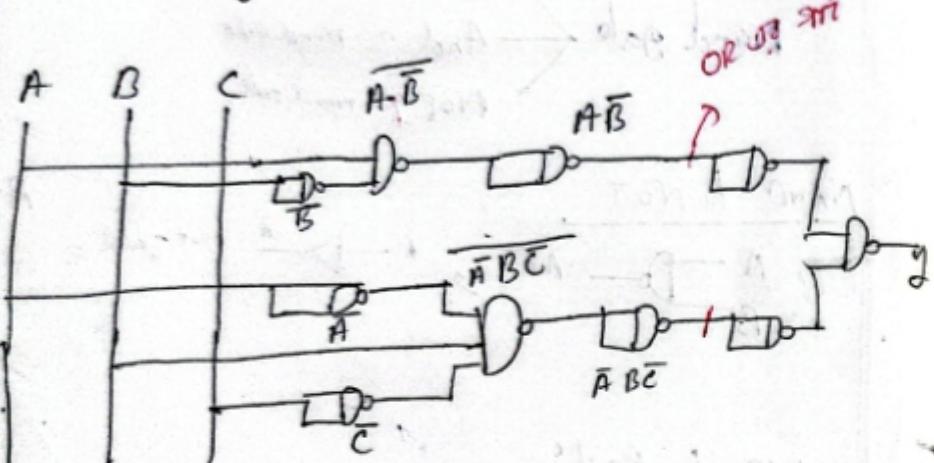


$$\overline{A \cdot B}$$

$$= A\bar{B}$$

2) Implement following function with NOR

Order:  $y = A\bar{B} + \bar{A}B\bar{C}$



OR OR NOT

3) Given, Implement function  
 $F(A, B, C, D) = \sum(0, 1, 3, 4, 8, 9, 10, 11, 12, 14, 15)$

$\bar{CD}$	$\bar{AB}$	$AB$	$\bar{AB}$
$\bar{CD}$	1	1	1
$CD$	1	0	0
$\bar{CD}$	1	1	1
$CD$	1	0	0
$\bar{CD}$	0	1	1
$CD$	0	1	1
$\bar{CD}$	0	0	1
$CD$	0	0	1

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

$\bar{AB}$	$AB$	$\bar{AB}$	$AB$
$\bar{CD}$	1	0	1
$CD$	1	0	0
$\bar{CD}$	0	1	0
$CD$	0	1	1
$\bar{CD}$	0	0	1
$CD$	0	0	0
$\bar{CD}$	1	1	0
$CD$	1	1	1

10.03.94

After tick

~~Transition table for T-V Flip-Flop~~

Output transition	Flip-Flop input		Don't care $\Rightarrow$ next step
	$Q_N \rightarrow Q_{N+1}$	$Q_N \rightarrow \bar{Q}_{N+1}$	
0 → 0	0	x	
0 → 1	1	x	
1 → 0	0	1	
1 → 1	x	1	

Here,  $P_S = 0$  &  $R_S = 0$ , then  $J = 0$  &  $K = x$

Asynchronous or Common clock pulse bit state  
Change occurs,

Counter (synchronous)  $\rightarrow$  Parallel.

T flip-flop or  $\textcircled{T}$  FF timer

TRF timer  $\rightarrow$  short total sum result now

$$\begin{array}{l} \square \quad \square \quad \square \\ \text{3 bit} = 2^3 = 8 \\ \text{Word No: 8} \end{array}$$

3.64 Synchronous Up counter:

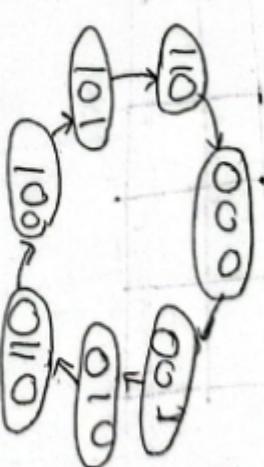
Present State								Next State				
Q <sub>3</sub>	Q <sub>2</sub>	Q <sub>1</sub>	Q <sub>0</sub>	Q <sub>3</sub>	Q <sub>2</sub>	Q <sub>1</sub>	Q <sub>0</sub>	J <sub>A</sub>	K <sub>A</sub>	J <sub>B</sub>	K <sub>B</sub>	
0	0	0	0	1	0	x	1	x	0	0	x	1
0	0	0	1	0	0	x	1	x	1	1	x	1
0	0	1	0	1	0	0	x	x	0	1	x	1
0	1	0	1	1	0	0	x	x	0	1	x	1
0	1	1	0	0	x	x	0	0	1	1	x	1
1	0	0	1	0	1	x	1	x	1	0	x	1
1	0	1	1	1	0	x	1	x	1	1	x	1
1	1	0	0	0	x	x	1	1	1	0	x	1

as it's a 3 bit  
it can only count  
(0-7) & after  
it goes to 0

(Flip-flop)  
(PS > NS)

For J <sub>A</sub>			
Q <sub>3</sub> Q <sub>2</sub> Q <sub>1</sub> Q <sub>0</sub>	Q <sub>3</sub> Q <sub>2</sub> Q <sub>1</sub> Q <sub>0</sub>	Q <sub>3</sub> Q <sub>2</sub> Q <sub>1</sub> Q <sub>0</sub>	Q <sub>3</sub> Q <sub>2</sub> Q <sub>1</sub> Q <sub>0</sub>
0 0 0 0	0 0 0 1	0 0 1 0	0 0 1 1
0 0 1 0	0 0 1 1	0 1 0 0	0 1 0 1

For K <sub>A</sub>			
Q <sub>3</sub> Q <sub>2</sub> Q <sub>1</sub> Q <sub>0</sub>	Q <sub>3</sub> Q <sub>2</sub> Q <sub>1</sub> Q <sub>0</sub>	Q <sub>3</sub> Q <sub>2</sub> Q <sub>1</sub> Q <sub>0</sub>	Q <sub>3</sub> Q <sub>2</sub> Q <sub>1</sub> Q <sub>0</sub>
0 0 0 0	0 0 0 1	0 0 1 0	0 0 1 1
0 0 1 0	0 0 1 1	0 1 0 0	0 1 0 1



State diagram:

For  $J_B^0$

0	1	0	0	0	1	0	0
4	5	6	7	8	9	10	11
0	1	0	0	0	1	0	0
4	5	6	7	8	9	10	11

For  $k_B^0$

0	1	0	0	0	1	0	0
4	5	6	7	8	9	10	11
0	1	0	0	0	1	0	0
4	5	6	7	8	9	10	11

For  $J_C^0$

0	1	0	0	0	1	0	0
4	5	6	7	8	9	10	11
0	1	0	0	0	1	0	0
4	5	6	7	8	9	10	11

$V_C^0$

0	1	0	0	0	1	0	0
4	5	6	7	8	9	10	11
0	1	0	0	0	1	0	0
4	5	6	7	8	9	10	11

3 bit Synchronous UP-Counter

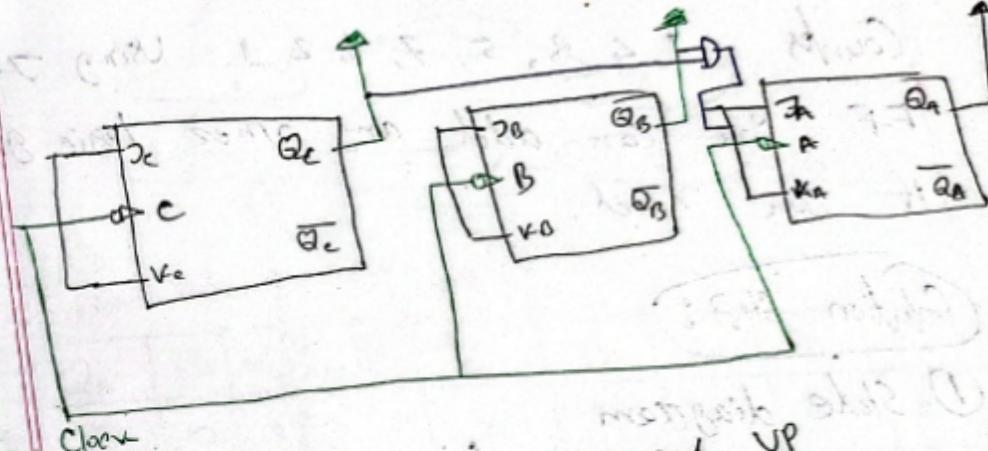


Fig: 3 bit Synchronous UP Counter Octal-7 MOD-8

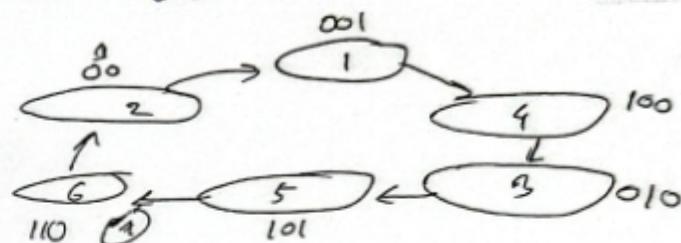
0	0	0	x
0	1	1	x
1	0	x	2
1	1	x	0

# Design a Synchronous Counter that  
Counts 1, 4, 3, 5, 7, 6, 2, 1 Using J-K  
FF - you can add any other logic gate  
if you need.

Solution Step:

- I State diagram
- II Next state table (transition table)
- III FF transition table
- IV Circuit excitation table
- V K-map
- VI Logic expression
- VII Counter implementation.

State diagrams



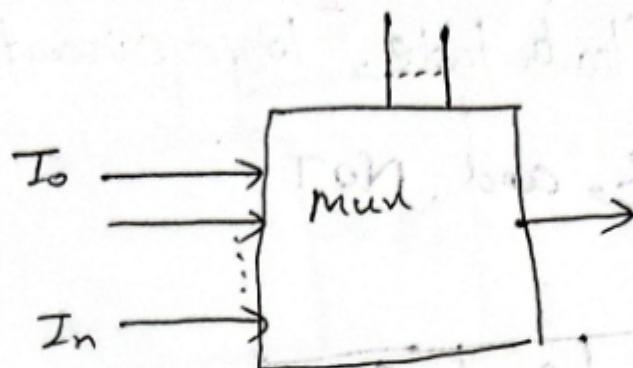
	Present state	next state.	J <sub>A</sub>	K <sub>A</sub>	J <sub>B</sub>	K <sub>B</sub>	J <sub>C</sub>	K <sub>C</sub>
m <sub>0</sub> 1	0 0 1 1 0 0	1 x 0 x	1	x	0	x		
m <sub>1</sub> 4	1 0 0 0 1 1	x 1 1 x			1	1	x	
m <sub>2</sub> 3	0 1 1 1 0 1	1 x x x	1	x	x	x	1	
m <sub>3</sub> 5	1 0 1 1 1 1	x 0 x 1 x			0	x 1		
m <sub>4</sub> 67	1 1 1 1 1 0	x 0 x 0 x			0	x	0	
m <sub>5</sub> 6	1 1 0 0 1 0	x 1 x x 0			1	x	x	0
m <sub>6</sub> 2	0 1 0 0 0 1	0 x x x 1						
m <sub>7</sub> 1	0 0 1 1 0 0	1 x 0 x x						

Reseted to 1

Reapent to start (22)

19-03-29

## Multiplexers

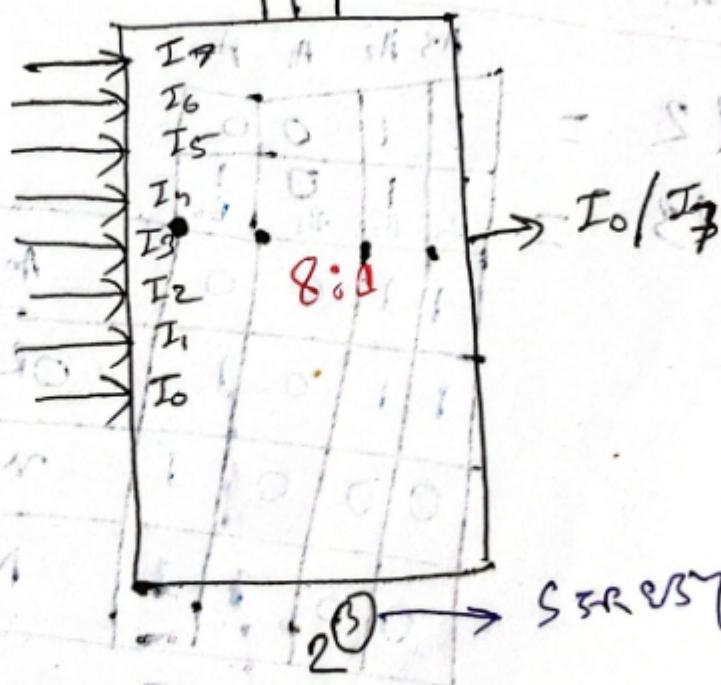
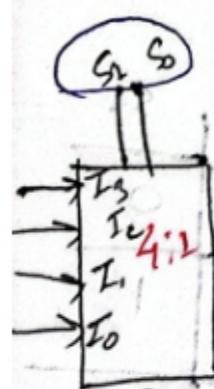
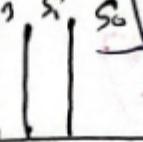


input  $I_0$  (exp)  
Output 1

4:1, 8:1, 16:1, 32:1,  $2^n:1$

$2^1 \quad 2^2$        $2^4 \quad 2^5 \quad 2^n$

(3)



S3R23Y

8:1

4

Design a 4 bit logic Unit for the following functions : (truth table, logic circuit, example)

And, OR, NOR, and, NOT.

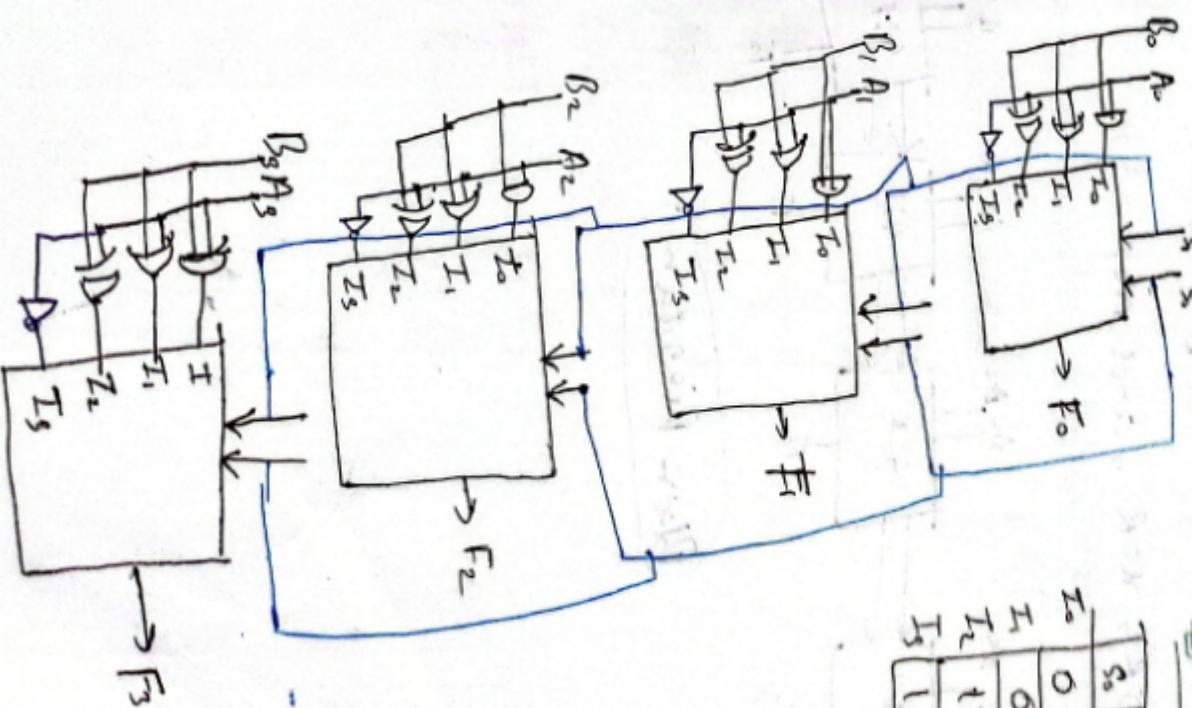
$I_0$	$I_1$	$I_2$	$I_3$	$S_0$	$S_1$	$S_2$	$S_3$
0	0	0	0	0	0	0	0
0	0	0	1	1	0	0	1
0	0	1	0	0	1	0	1
0	1	0	0	1	1	0	1
1	0	0	0	0	0	1	1
1	0	0	1	0	1	1	0
1	0	1	0	1	0	1	0
1	1	0	0	0	1	0	0

4 bit Example

$A = 12$	$B = 13$	$I_0$	$I_1$	$I_2$	$I_3$	$S_0$	$S_1$	$S_2$	$S_3$
1	1	0	0	0	0	0	0	0	0
1	0	1	0	0	0	1	0	0	1
0	1	0	0	0	1	0	1	0	1
0	0	0	1	0	0	1	1	0	0
0	0	1	1	0	0	1	0	1	1
0	0	1	1	1	0	1	1	1	0
0	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1

Truth table

4 bit logic Unit circuit



$I_0$	$I_1$	$I_2$	$I_3$	$S_0$	$S_1$	$S_2$	$S_3$
0	0	0	0	0	0	0	0
0	0	0	1	1	0	0	1
0	0	1	0	0	1	0	1
0	1	0	0	1	1	0	1
1	0	0	0	0	0	1	1
1	0	0	1	0	1	1	0
1	0	1	0	1	0	1	0
1	1	0	0	0	1	0	0

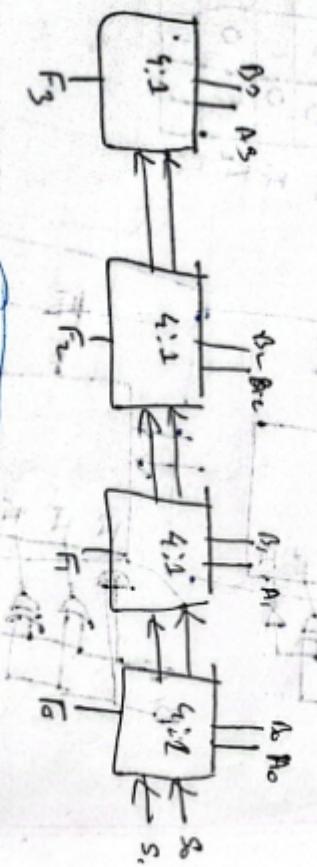
Function table

Q2

Design a 4 bit logic unit using 4:1

max for the following function.

And, OR, XOR, and NOT



Block Diagram

What is SAP?

Simple as possible. 2 operation (Arithmetic operation)

o Addition & subtraction

# The instruction set and OP-code of SAP:

**SAP-1**

1) Load

$\rightarrow 0000$

2) ADD

$\rightarrow 0001$

3) SUB

$\rightarrow 0010$

4) MUL

$\rightarrow 0011$

5) OUT

$\rightarrow 0110$

$$\begin{array}{l}
 \text{Out} - 1110 \\
 \text{ADD} = 0001 \\
 \text{MUL} - 1111 \\
 \text{SUB} = 0010
 \end{array}$$

$LH^n =$

Write the assembly language.

$$10H + 18H + 20H - 14H$$

Assembly lang

Machine language

Address	Contains	Address	Contains
0H	LDA 9H	0000	0000 1001
1H	ADD AH	0001	0001 1010
2H	ADD BH	0010	0001 1011
3H	SUB CH	0011	0010 1100
4H	Out	0100	0100 1110
5H	MUL	0101	0101 1111
6H	**	0110	0110 1002
7H	**	0111	0111 1003
8H	**	1000	1000 X
9H	1001	0001 0000	0001 0000
10H	0001	1000	1000
11H	10	10	10
12H	0010	0000	0000
13H	0100	0100	0100

Create a Syst

$$0F(52 + 28 - 38 + 72 - 12)$$

in Hexa = 34H + 1CH - 26H + 48H - C

Address	Contains	Address	Contains
0H	LDA 8H	0000	0000 1000
1H	ADD 9H	0001	0001 1001
2H	Sub AH	0010	0010 1010
3H	ADD BH	0011	0011 1011
4H	Sub CH	0100	0100 1100
5H	Out	0101	0101 1110
6H	MUL	0110	0110 1111
7H	**	1000	1000 X
8H	**	0001	0001 0000
9H	1001	1000	1000
10H	0001	1000	1000
11H	10	10	10
12H	0010	0000	0000
13H	0100	0100	0100

~~To~~ Create a SAP]

$$18 - 20 + 35 + 37 - 8$$

Address	Contains	Address	Contains
0H	LDA 8H	0000	0000 1000
1H	SUB 9H	0001	0010 1001
2H	ADD AH	0010	0001 1010
3H	ADD BH	0011	0001 1011
4H	SUB CH	0100	0010 1100
5H	Out	0101	1110
6H	HLT	0110	1111
7H	x	0111	0001 0010
8H	12H	1000	0001 0001
9H	14H	1001	00001 00100
AH	23H	1010	0010 00101
BH	25H	1011	0010 0101
CH	8H	1100	1000

8 4 2 1  
1 1 1 1

Final Spring

8421  
0000  
0010  
1001

$$35 - 29 + 32 - 53 - 7$$

$$H = 23H - 18H + 20H - 35H - 7H$$

Address	Containing	Address	Content.
0M	LDA 8M	0000	0000 1000
1M	SUB 9M	0001	0010 1001
2M	ADD A M	0010	0001 1010
3M	SUB B M	0011	0010 1011
4M	SUB C M	0100	0010 1100
5M	Out	0101	1110 x
6M	MUL	0110	1111 x
7M	x	0111	x
8M	29M	1000	0010 001
9M	18 M	1001	0001 100
10M	20 M	1010	0010 00
11M	35 M	1011	0011 010
12M	7M	1100	0111

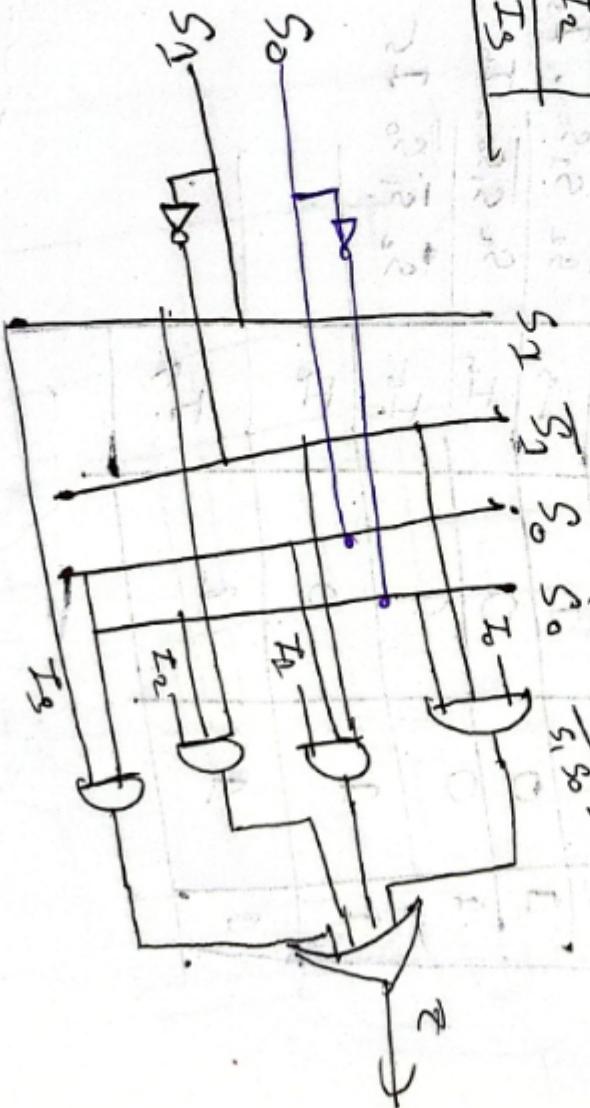
Mux

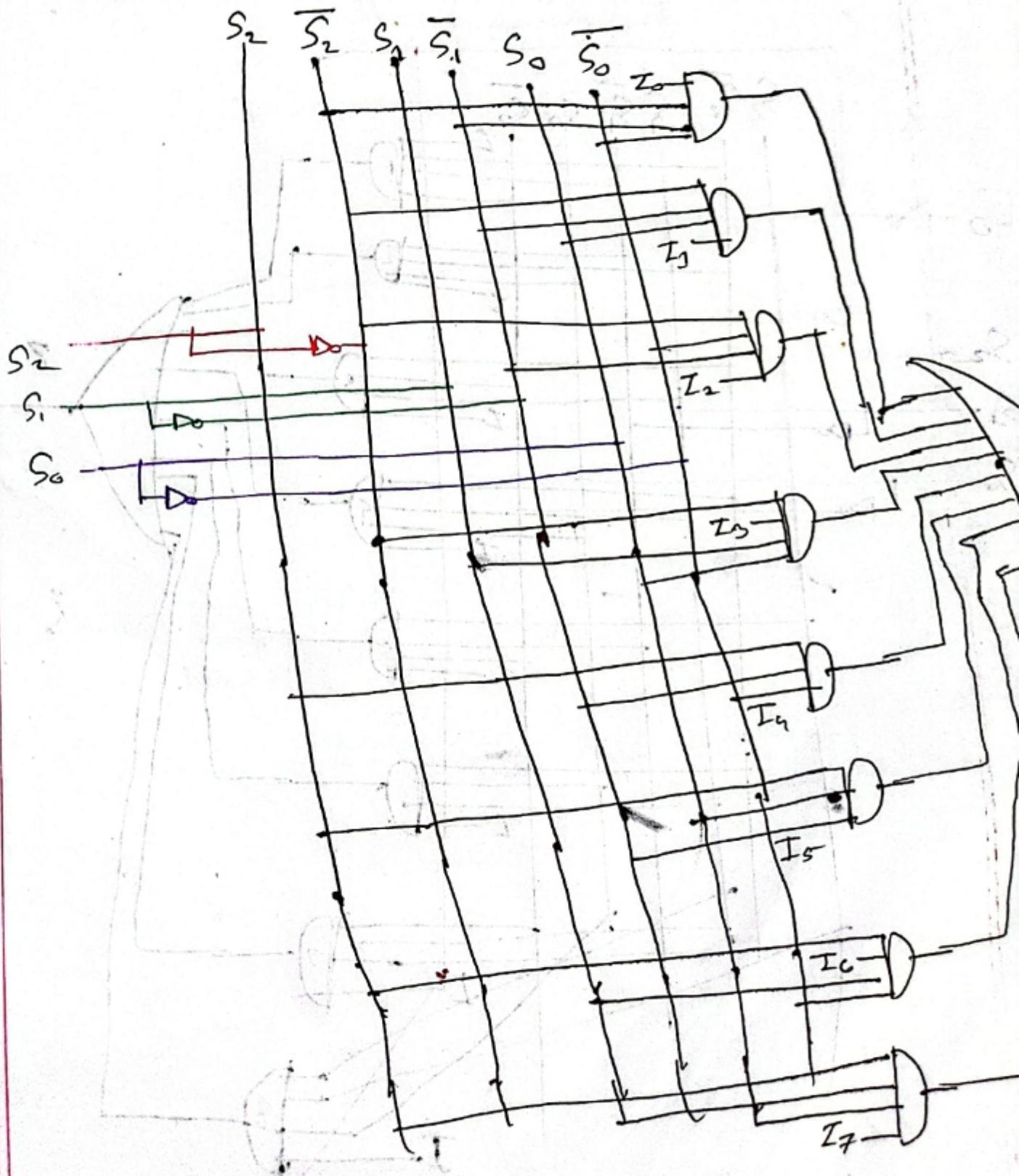
many input one output

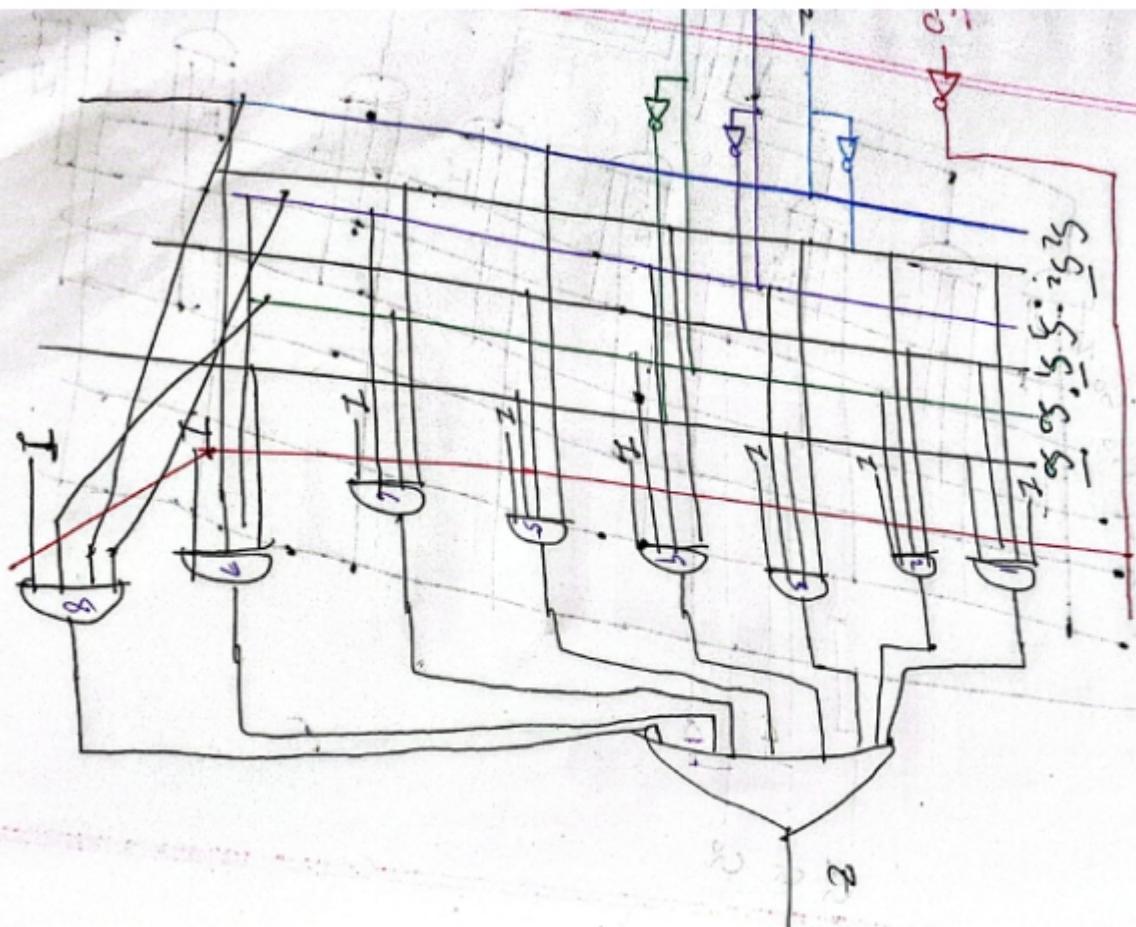
No of input =  $2^{\text{no of selected input}}$

$$\text{4 input mux} \quad Z = \bar{S}_1 \bar{S}_0 \cdot Z_0 + \bar{S}_1 S_0 \cdot Z_1 + S_1 \bar{S}_0 \cdot Z_2 + S_1 S_0 \cdot Z_3$$

1	1	1	1	$Z_1$
1	1	0	0	$Z_0$
1	0	1	0	$I_2$
0	1	0	1	$I_3$
1	0	0	1	$I_1$



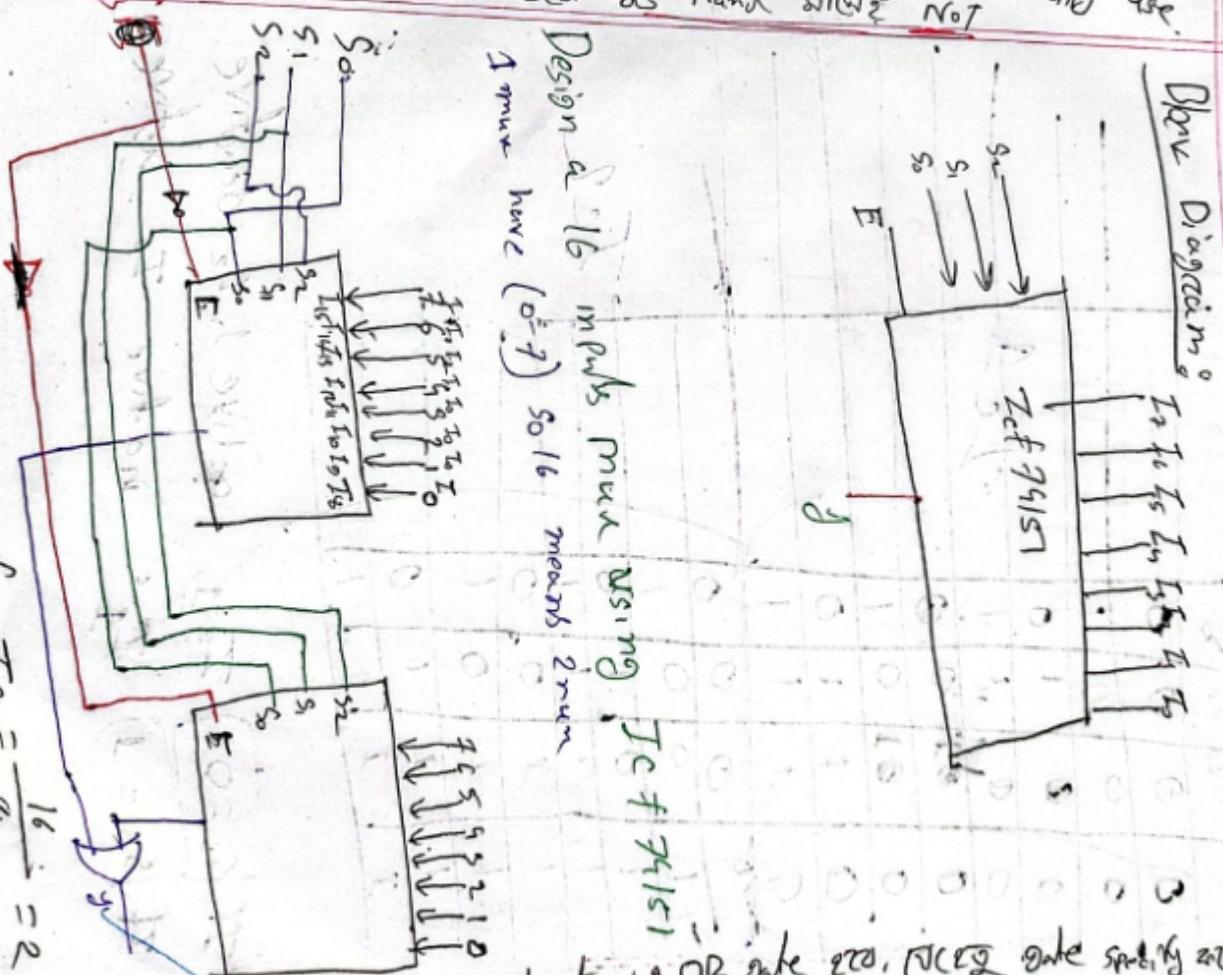




ZC #74151

~~E.g. value not 223 for 16 inputs~~ Specific case can be used as NAND gate universality case.

Design the circuit for selected input ZC#74151



Design a 16 inputs mux using 74151  
1 must have  $(2^5 - 1)$  so 16 means 2 mux

Output OR gate ecc. (IC#74151) OR gate having direct OR gate first, then NAND gate then OR gate then the output OR gate.

$$\text{No of } T_C = \frac{16}{8} = 2$$

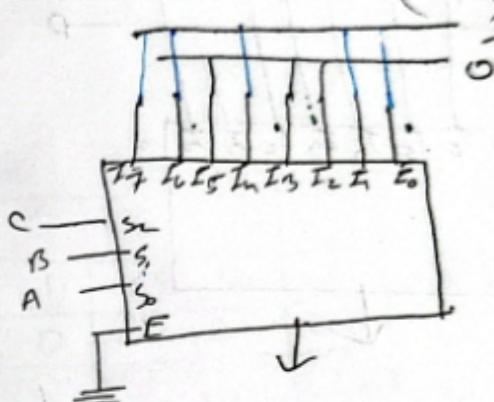
$$\text{No of input} = 2^5 - 1 = 31$$

$$\text{No of select input} = 2^3 = 8$$

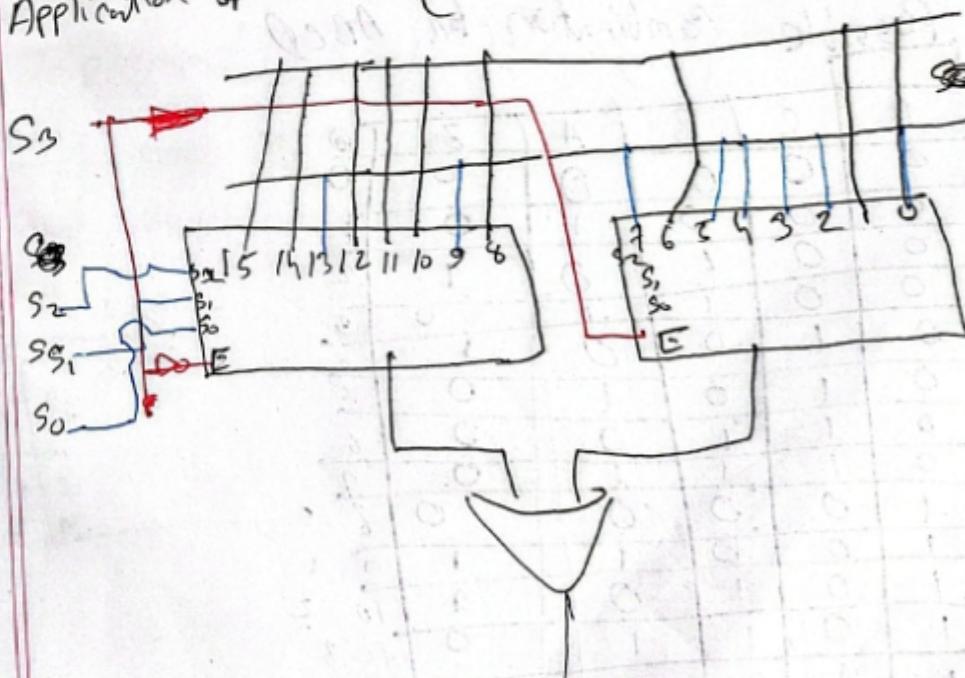
	$S_3$	$S_2$	$S_1$	$S_0$	
0	0	0	0	0	
0	0	0	0	1	
0	0	1	0		
0	0	1	1		
0	0	0	0		
0	0	0	1		
0	1	1	0		
0	1	1	1		
1	0	0	0		
1	0	0	1		
1	0	1	0		
1	0	1	1		
1	1	0	0		
1	1	0	1		
1	1	1	0		
1	1	1	1		

If  $S_3 = 0$ ,  $I_{e1}$  active &  $I_{e2}$  inactive  
If  $S_3 = 1$ ,  $I_{e1}$  inactive &  $I_{e2}$  active

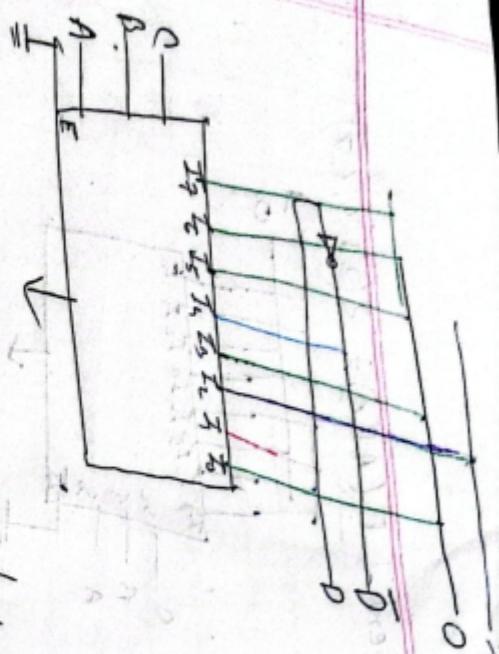
# Implement  $F(A, B, C) = \Sigma(0, 1, 4, 6, 7)$



# Application of mull =  $(0, 1, 6, 8, 10, 11, 12, 14, 15)$



21



- (d) setup a truth table showing output for 16 possible combination for ABCD

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

B)

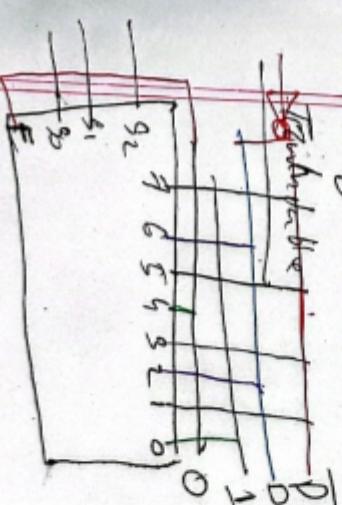
Simplifying

$\bar{AB}$	$\bar{A}\bar{B}$	$AB$	$A\bar{B}$
$\bar{CD}$	0	0	1
$CD$	1	0	0
$c\bar{D}$	0	0	0

$$Y = \bar{AB}\bar{C}D + \bar{AB}c\bar{D}$$

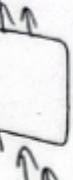
Implement  $F(A, B, C, D) = (0, 1, 3, 5, 7, 8, 10, 14)$

Using one IC and not gate



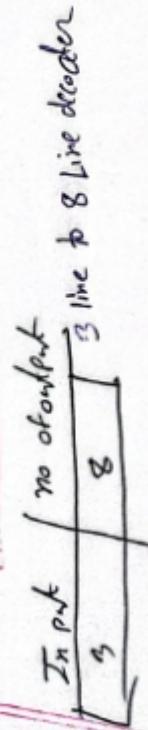
$\bar{D}-Y$	$I_0$	$I_1$	$I_2$	$I_3$	$I_4$	$I_5$	$I_6$	$I_7$
1	1	1	0	1	0	1	0	0
2	0	1	0	0	0	0	0	1
3	1	0	1	0	0	0	1	0
4	0	0	0	1	1	1	1	1
5	1	1	1	0	0	0	0	0
6	0	0	1	1	0	0	0	0
7	1	0	0	0	1	1	1	1
8	0	1	1	1	0	0	0	0
9	1	1	0	0	1	1	1	1
10	0	0	0	1	0	0	1	0
11	1	0	1	0	0	1	0	0
12	0	1	1	0	0	0	1	0
13	1	1	0	0	1	0	0	0
14	0	0	0	0	0	1	1	0
15	1	0	0	1	0	0	0	1

## Decoder

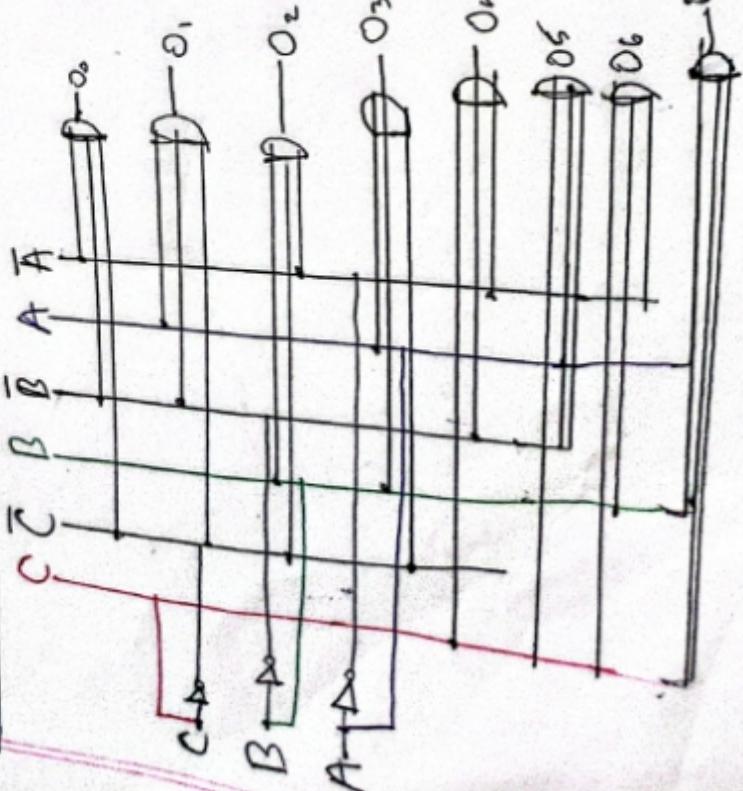


Many input many output  $\Rightarrow$

$$\text{no of output} = 2^{\text{no of input}}$$



## 3 line to 8 line Decoder



## 4 line to 16 line Decoder without design

Input  $D_C B_A$   
Output will be  $2^4 = 16$

$\bar{A}$  (odd number) = 1, 3, 5, 7, 9, 11, 13, 15  
 $A$  (even number) = 2, 4, 6, 8, 10, 12, 14, 16  
 $\bar{B}$  (2 number then jump) = 12, 5, 6, 9, 10, 13, 14  
 $B$  ( ) = 9, 4, 7, 8, 11, 12, 15, 16

$\bar{C}$  (4 number then jump) = 1, 2, 3, 4, 9, 10, 11, 12  
 $C$  ( ) = 5, 6, 7, 8, 13, 14, 15, 16

$\bar{D}$  (8 number then sum) = 1, 2, 3, 4, 5, 6, 7, 8

$\bar{D}(\ ) = \text{off}(9-16)$

4 line to 10 line Decoder

$A = 1, 3, 5, 7, 9$   $\rightarrow$  sum number 2, 4, 6, 8, 10

$A = 2, 4, 6, 8, 10$   $\rightarrow$  10, 29 + sum

$\bar{B} = 1, 2, 3, 5, 6, 9, 10$   $\rightarrow$  sum number 4, 7, 8

$B = 3, 4, 7, 8$   $\rightarrow$  sum number 1, 2, 5, 6

$\bar{C} = 1, 2, 3, 4, 5, 2, 10$   $\rightarrow$  sum number 6, 0

$C = 5, 6, 3, 8$   $\rightarrow$  sum number 1, 9

$\bar{D} = 1, 2, 3, 4, 5, 6, 7, 8$   $\rightarrow$  sum number 9

$D = 0, 10$   $\rightarrow$  sum number 0, 1

$\bar{D} = 0, 10$   $\rightarrow$  sum number 0, 1

$D = 0, 10$   $\rightarrow$  sum number 0, 1

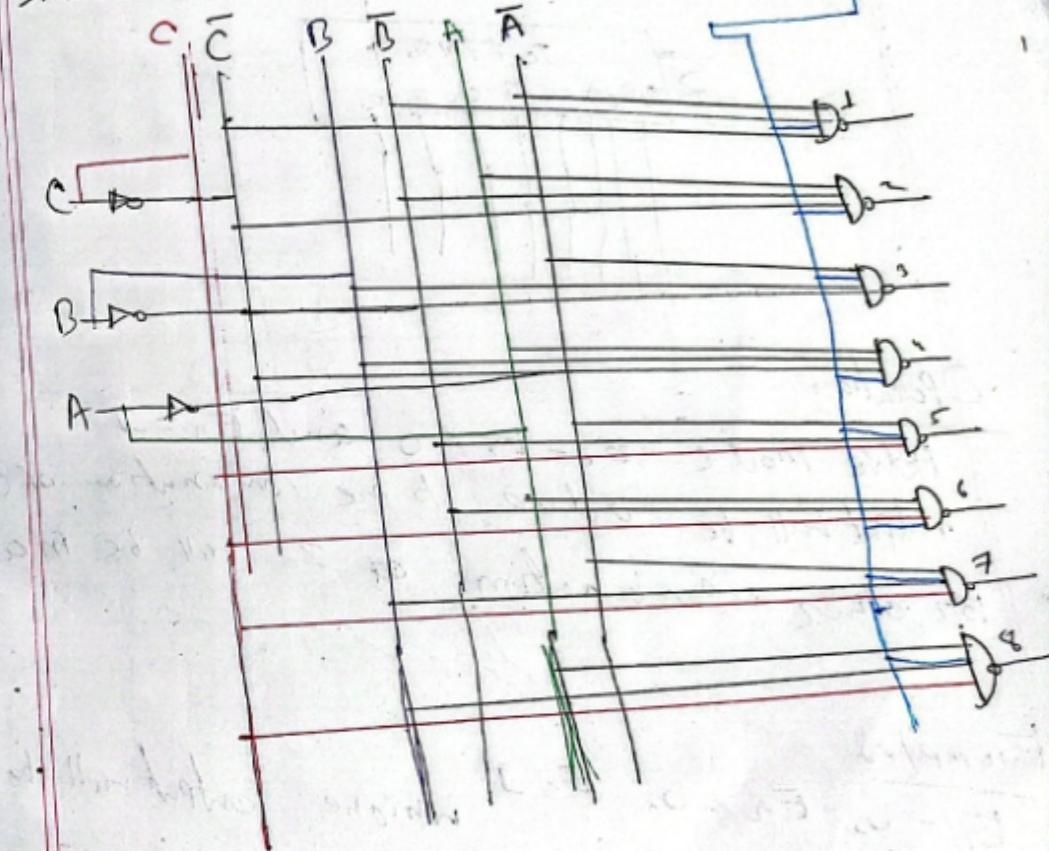
$\bar{D} = 0, 10$   $\rightarrow$  sum number 0, 1

$D = 0, 10$   $\rightarrow$  sum number 0, 1

$\bar{D} = 0, 10$   $\rightarrow$  sum number 0, 1

Decoder IC (74138):

Internal circuit



Enable input

$E_1, E_2, E_3$

Active mode  $E_1 = 0$   $\rightarrow$  sum number 0, 1

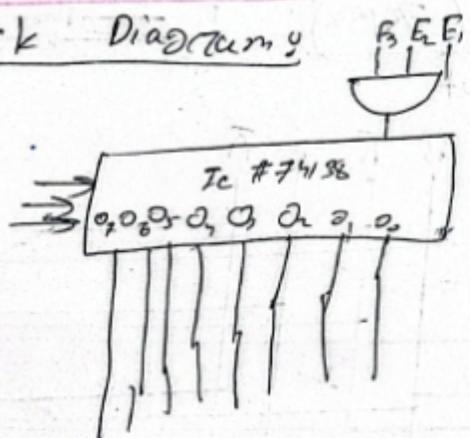
$E_2 = 0$

$E_3 = 1$

001

active

### Block Diagram



### Operations

Active mode E<sub>1</sub> = E<sub>2</sub> = 0 and E<sub>3</sub> = 1  
output will be active to the combination of CBA  
for other combination I<sub>0</sub> will be inactive.

Example  
E<sub>1</sub> = 0, E<sub>2</sub> = 0, E<sub>3</sub> = 1, which the output will be active.

$$A = 0, B = 1, C = 0$$

Now (001) I<sub>0</sub> is active now check input

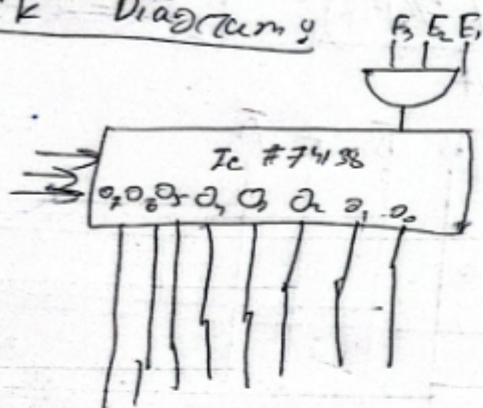
$$\begin{aligned} CBA &= 110 \text{ Decimal is } 6 \\ \text{So } I_{06} \text{ pin is active} \end{aligned}$$

# Design a 4 line to 16 line gate if possible!  
Can use other gate if needed.  
Decoders are always used  
so number of is needed.

$$\begin{aligned} \text{Input} &= CBA \\ \text{Output} &= Q_0 - Q_{15} \end{aligned}$$

D	C	B	A	Noting output
0	0	0	0	Q <sub>0</sub>
0	0	0	1	Q <sub>1</sub>
0	0	1	0	Q <sub>2</sub>
0	0	1	1	Q <sub>3</sub>
0	1	0	0	Q <sub>4</sub>
0	1	0	1	Q <sub>5</sub>
0	1	1	0	Q <sub>6</sub>
0	1	1	1	Q <sub>7</sub>
1	0	0	0	Q <sub>8</sub>
1	0	0	1	Q <sub>9</sub>
1	0	1	0	Q <sub>10</sub>
1	0	1	1	Q <sub>11</sub>
1	1	0	0	Q <sub>12</sub>
1	1	0	1	Q <sub>13</sub>
1	1	1	0	Q <sub>14</sub>
1	1	1	1	Q <sub>15</sub>

### Block Diagram



### Operations

Active mode  $E_1 = E_2 = 0$  and  $E_3 = 1$   
output will be active. To the combination of CSA  
for other combination of  $I_C$  will be inactive.

### Example

$E_1 = 0, E_2 = 0, E_3 = 1$  which the output will be active

$$A = 0, B = 1, C = 0$$

one (001)  $I_C$  is active now check next

$BA = 110$  decimal is 6  
so  $I_C$ 's 6th pin is active

# Design a 4 line to 16 line decoder using  $I_C$   
can you use other gate if needed.

Decoder's  $I_C$  always takes input 0-7  
So number of  $I_C$  needed.  $= \frac{16}{8} = 2$

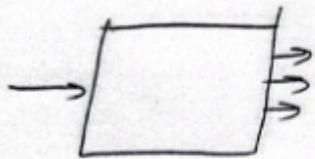
$$\begin{aligned} \text{Input} &= DCBA \\ \text{Output} &= O_0 - O_{15} \end{aligned}$$

D	C	B	A	Active Output
0	0	0	0	$O_0$
0	0	0	1	$O_1$
0	0	1	0	$O_2$
0	0	1	1	$O_3$
0	1	0	0	$O_4$
0	1	0	1	$O_5$
0	1	1	0	$O_6$
0	1	1	1	$O_7$
1	0	0	0	$O_8$
1	0	0	1	$O_9$
1	0	1	0	$O_{10}$
1	0	1	1	$O_{11}$
1	1	0	0	$O_{12}$
1	1	0	1	$O_{13}$
1	1	1	0	$O_{14}$
1	1	1	1	$O_{15}$

#  $I_C$  2

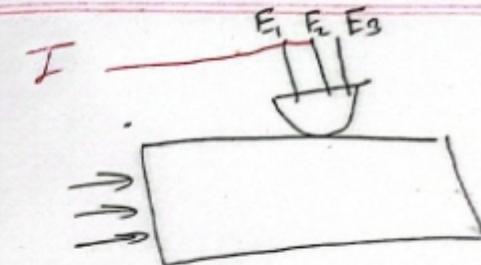
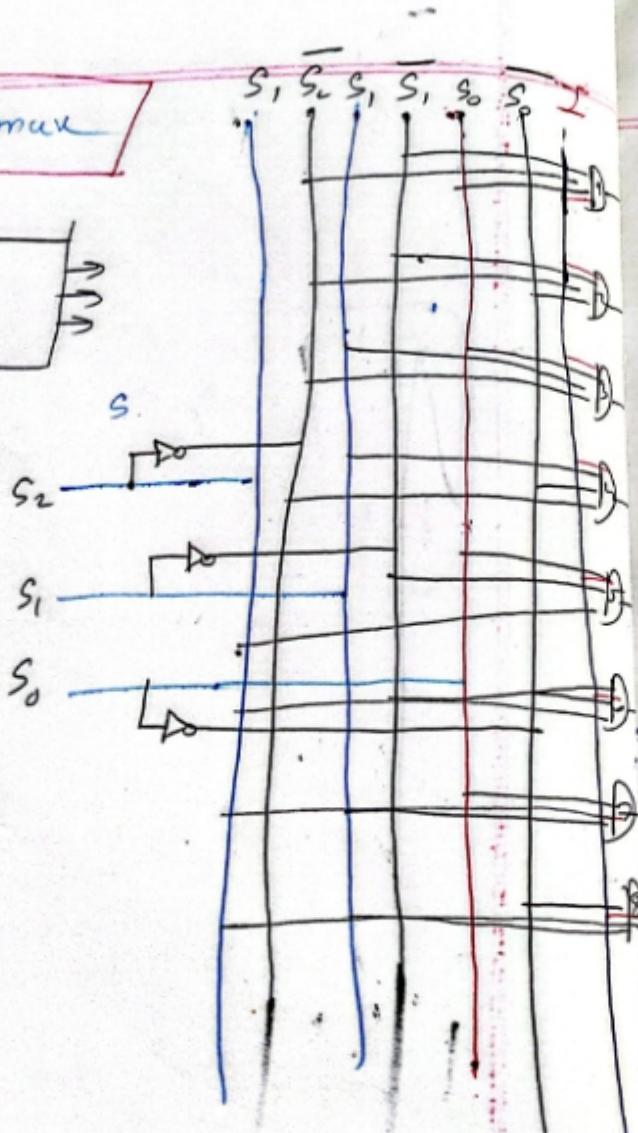
#  $I_C$

Demux



1 of 8 Demux

$S_2$	$S_1$	$S_0$	$I$
0	0	0	$O_0$
0	0	1	$O_1$
0	1	0	$O_2$
0	1	1	$O_3$
1	0	0	$O_4$
1	0	1	$O_5$
1	1	0	$O_6$
1	1	1	$O_7$



Active

$$E_1 = E_2 \text{ } 0$$

$$E_3 = 1$$