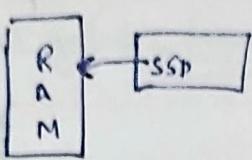
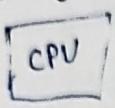


DP CON

Problem
solving.

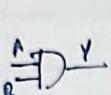


./a.out

* Logic gates:

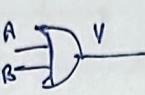
AND gate:

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



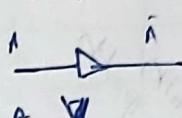
OR

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



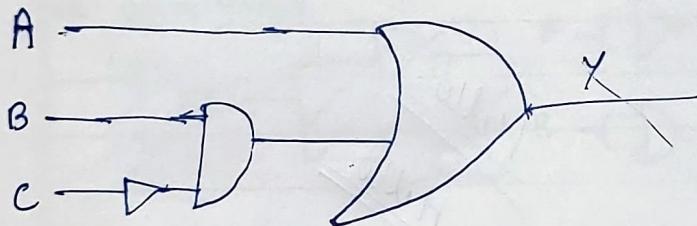
NOT

A	Y
0	1
1	0

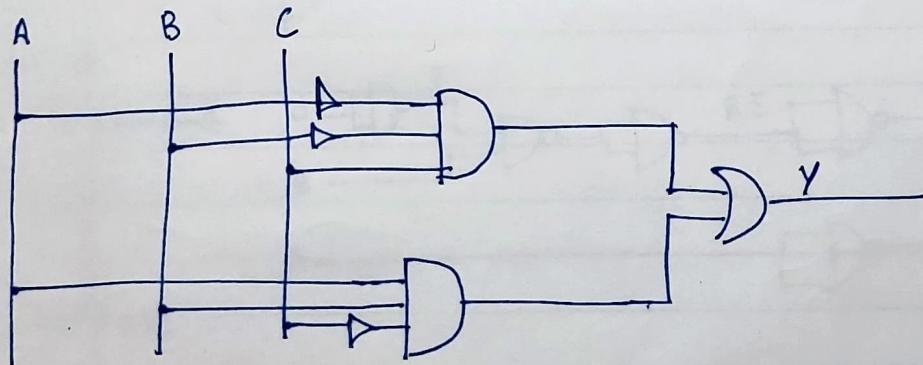


Q) Design a circuit using logic gates.

$$Y = A + BC$$

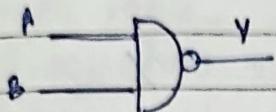


2)
$$Y = \bar{A}\bar{B}C + ABC$$
 [If you want draw ~~an~~ or get directly]



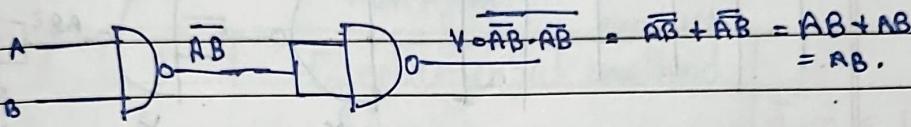
Universal gates: NAND and NOR

i) NAND

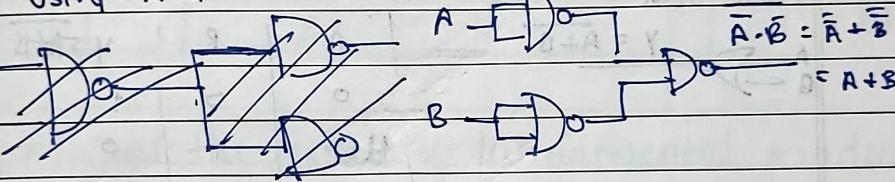


A	B	$Y = \bar{A}\bar{B}$
0	0	1
0	1	0
1	0	0
1	1	0

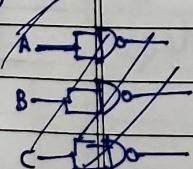
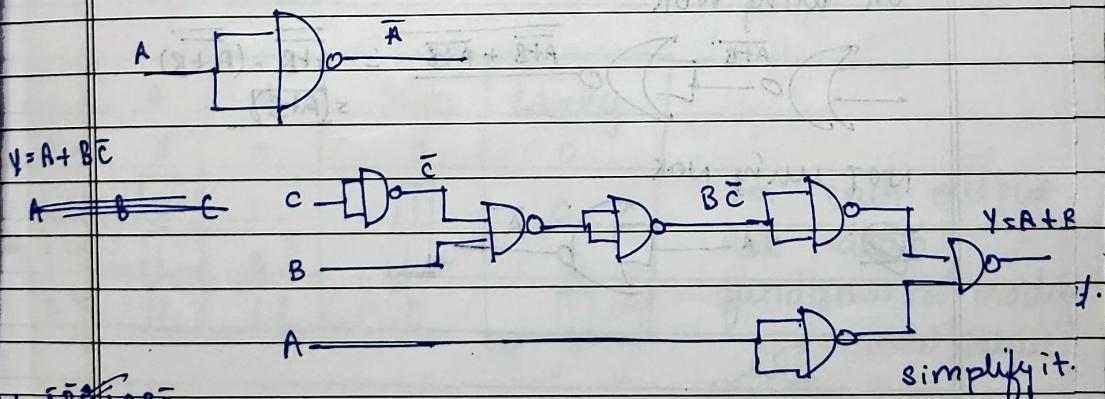
AND using NAND



OR using NAND



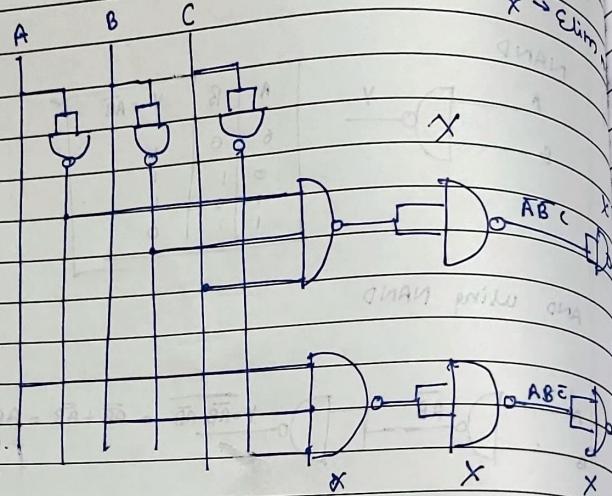
NOT using NAND



pap

Do W

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



NOR logic

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

	A	B	y
1	0	0	1
2	0	1	0
3	1	0	0
4	1	1	0

OR using NOR

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

NOT using NOR

$$\overline{A} = \overline{\overline{A}}$$

papergrid
Date: / /

logic design (Digital Design)

designing a circuit.

Truth table

SOP equation or POS equation

Simplify the above equation.

Write the circuit using basic gates or universal gates.

* sum of product method (SOP method)

Minterm

product

A	B	Fundamental product
0	0	$\bar{A}\bar{B}$
0	1	$\bar{A}B$
1	0	$A\bar{B}$
1	1	AB

$$\bar{0} \cdot \bar{0} = 1 \cdot 1 = 1$$

$$\bar{0} \cdot 1 = 1 \cdot 1 = 1$$

$$1 \cdot \bar{0} = 1 \cdot 1 = 1$$

$$1 \cdot 1 = 1 \cdot 1 = 1$$

Always the result of fundamental product is 1.

→ Half adder

	A	B	Sum	Carry
0	0	0	0	0
1	0	1	1	0
2	1	0	1	0
3	1	1	0	1

For each output note down fundamental products.

↓ decimal values

$$\begin{cases} \text{SOP equation: } \text{Sum: } \bar{A}B + A\bar{B} = \sum_m(1, 2) \\ \text{Carry: } AB = \sum_m(3) \end{cases}$$

pa

Do

$$\begin{array}{r} 0 \quad 0 \\ 0 \quad 1 \quad \frac{0}{1} \quad \frac{-1}{0} \end{array}$$

Date: 13/11/23

Half subtractor:

A	B	Diff	Borrow
0	0	0	0
0	1	1	1
0	0	1	0
1	1	0	0

Difference: $\bar{A}B + A\bar{B} = \Sigma_m(1,2)$

Borrow: $\bar{A}B = \Sigma_m(1)$

② Product of sum method (POS method)

A	B	Maximum fundamental sum	
0	0	$A+B$	$0+0=0$
0	1	$A+\bar{B}$	$0+0=0$
1	0	$\bar{A}+B$	$\bar{1}+0=0=0$
1	1	$\bar{A}+\bar{B}$	$\bar{1}+\bar{1}=0+0=0$

The result of the fundamental sum is 0.

Half adder using POS method:

$A+B$	A	B	Sum	Carry
0	0	0	0	0
0	1	1	1	0
1	0	1	1	0
1	1	0	0	1

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

$$= \Pi_M(0,3)$$

Product

$$\text{Carry} = (A+B)(A+\bar{B})(\bar{A}+B)$$

$$= \Pi_M(0,1,2)$$

Half subtractor using POS:

A	B	Diff	Borrow
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	0

$$\text{Difference} = (A+B)(\bar{A}+\bar{B}) = \Pi_M(0,3)$$

$$\text{Borrow} = (A+B)(\bar{A}+\bar{B})(\bar{A}+B) = \Pi_M(0,1,2)$$

pc

Do

Half adder:

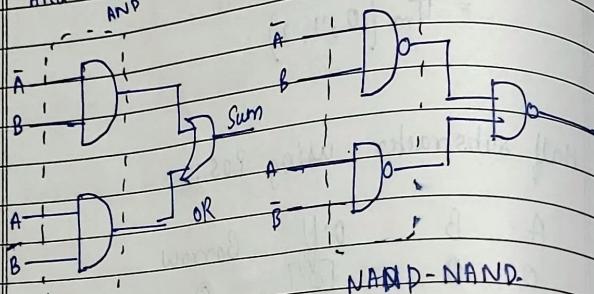
1) SOP method

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

And or circuit

AND

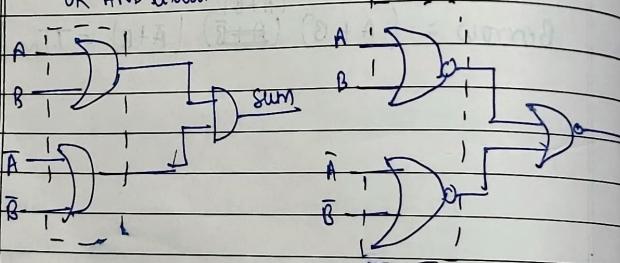
NAND-NAND circuit.



2) POS method:

$$\text{sum} = (A+B)(\bar{A}+\bar{B})$$

OR AND circuit



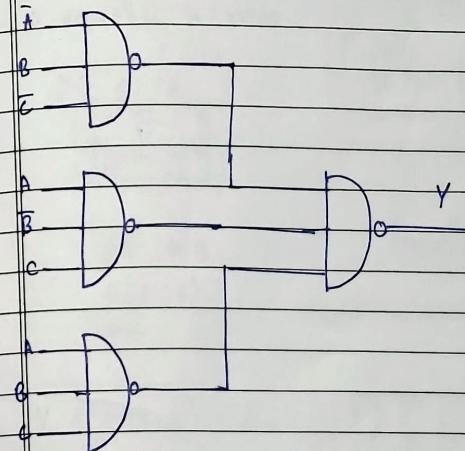
NOR - NOR.

① Design a circuit using NAND gate
for the given equation.

$$Y = f(A, B, C) = \Sigma m(2, 5, 7)$$

o/p = 1

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

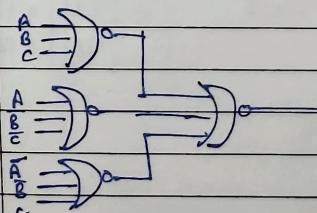


② Design a circuit using NOR gate for the given equation.

$$\neg Y = f(A, B, C) = \overline{\prod M}(0, 3, 6)$$

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

0 3 6



K-map:

$$\begin{aligned}
 Y &= A\bar{B} + AB = \Sigma_m(2, 3) \\
 &= A(\bar{B} + B) \\
 &= A \cdot 1 \\
 &= A
 \end{aligned}$$

It is a visual display of the fundamental product needed for a SOP solution.

1) 2-variable K-map

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

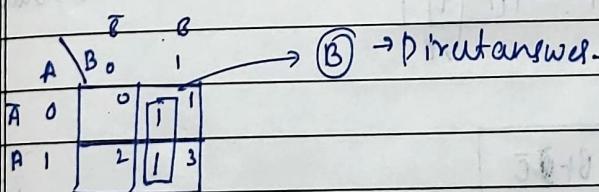
- 1) Put 1 in Σ_m values
- 2) Group them.

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

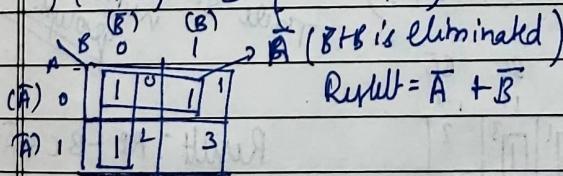
$$A(\bar{B} + B)$$

$$= A.$$

$$2) Y = f(A, B) = \Sigma_m(1, 3)$$



$$3) f(A, B) = \Sigma(0, 1, 2)$$



3 variable K-map:

$$\text{Ex: } f(A, B, C) = \Sigma_m(1, 3, 5, 7)$$

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

Pair $\rightarrow 2$
 Quad $\rightarrow 4$
 Octet $\rightarrow 8$ be descending

$$\text{Result is } \underline{\bar{A}B+C}$$

$$3) f(A, B, C) = \Sigma_m(0, 1, 3, 4, 5, 7)$$

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

$$\text{Result: } \bar{B}+C$$

$$3) f(A, B, C) = \Sigma_m(0, 2, 3, 4, 6, 7)$$

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

folding & overlapping
is allowed.

$$\text{Result: } B+\bar{C}$$

$$4) f(A, B, C) = \Sigma_m(0, 1, 3, 6, 7)$$

There should be minimum
number of groups

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

$$\text{Result: } AB + BC + \bar{AB}$$

12

$$Y = \Sigma_m(1, L, 5)$$

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

$$\text{Result: } \bar{B}C + \bar{A}\bar{B}\bar{C}$$

(Groups)

1) Pair

1	1	1
1		

2) Quad

1	1	1	1
1	1	1	1

3) Octet

1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1

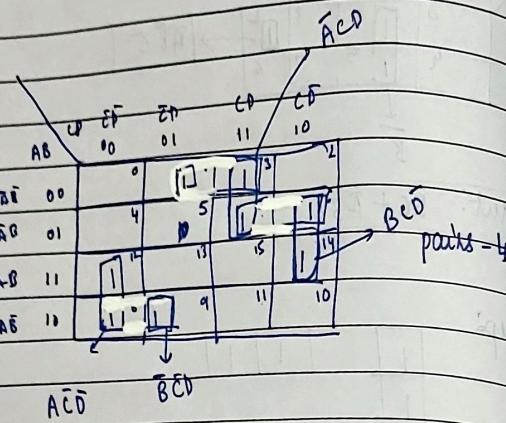
3) 4 variable K-map:

		$\bar{B}\bar{D}$	$\bar{B}D$	$B\bar{D}$	BD
		00	01	11	10
		A 0	1	1	1
		A 1	1	1	1

Ex: $f(A, B, C, D) = \Sigma_m(2, 5, 7, 8, 9, 10, 11, 12, 13, 14, 15)$

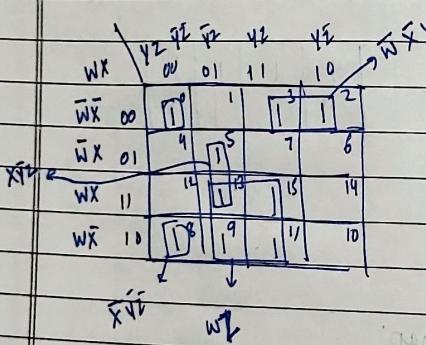
Result: $A + B\bar{D} + \bar{B}C\bar{D}$

$$1) Y = f(A, B, C, D) = \Sigma m(1, 3, 6, 7, 8, 9, 11, 14)$$



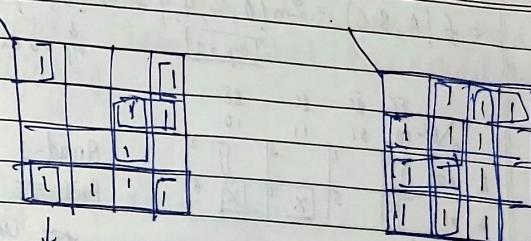
$$\text{Result} = A\bar{C}\bar{D} + \bar{A}CD + \bar{A}C\bar{D} + B\bar{C}D$$

$$2) Y = f(W, X, Y, Z) = \Sigma m(0, 2, 3, 5, 8, 9, 11, 13, 15)$$



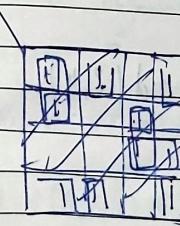
$$\text{Result: } WZ + \bar{W}\bar{Y}\bar{Z} + X\bar{Y}Z + \bar{W}\bar{X}Y$$

3)

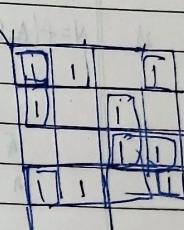


2 quads
2 pairs.

Octet - 1
Quad - 2
pair. - 2.



1 quad
By pair

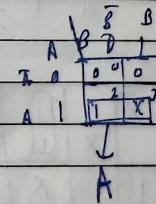


2 quads
3 pair.

* Don't care (x) : The input combination for which output can be 0 or 1.

Example:

A	B	Y
0	0	0
0	1	0
1	0	1
1	1	X (0 or 1)



$$Y = f(A, B, C) = \Sigma_m(0, 2, 4) + d(6, 7)$$

\downarrow
 $O/P = 1$

\downarrow
 $O/P = 0 \text{ or } 1$

		AC	BC	AB	AB
		00	01	11	10
A	00	1	1	3	1
	01	1	X	X	0

Quad-1
Pair

* Consider
quad 1
Consider
quad 6

C

Result: C

$$Y = f(A, B, C) = \Sigma_m(1, 2, 5, 7) + d(3, 4)$$

		AC	BC	BC	BC	BC
		00	01	11	10	
A	00	0	1	X	1	2
	01	X	1	1	0	0

4=0
3=1

Result: C + AB.

3)

1	1	X
X		
X	1	
1	X	X

1 quad
2 pairs
3 quads

1	X	1	X
1	X	1	X
X			
1		X	

quad 3
pairs

X	1	X
X	1	
1		X
X		1

1 quad
2 pairs

If variables exceed 4 then we use tabular method
also called as Tabular method (Quin McClusky for
simplification)

2/11/2022

* Simplify using K-map: pos

$$Y = f(A, B, C) = \prod_m(1, 3, 5, 6, 7)$$

		AC	BC	BC	BC	BC
		00	01	11	10	
A	00	0	1	0	3	2
	01	X	1	1	0	0

C
(A+B)

Result: (C')(A+B)

$$(2) Y = f(A, B, C) = \prod_m(0, 2, 3, 5, 7)$$

		AC	BC	BC	BC
		00	01	11	10
A	00	0	1	0	2
	01	4	0	0	0

Result:
(A+C)(A+B)(B+C)

(A+C)(A+B+C)(B+C)

$$(3) Y = f(A, B, C, D) = \prod_m(1, 3, 6, 7, 9, 10, 11, 13, 15)$$

		CD	CD	CD	CD	CD
		00	01	11	10	
AB	00	0	1	0	3	2
	01	4	5	10	7	0

Result:
(A+B)(B+C)
(AB+C)(A+B+C)

17
Monday

Duine Meckley Method (Tabular Method)

$$1) Y = f(A, B, C) = \Sigma m(0, 2, 3, 6, 7)$$

(condition: Primary grouping)
1) Based on no. of 1's.

$$\begin{array}{ccc} A & B & C \\ 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{array}$$

$$\begin{array}{c} 0 \\ 0 \\ 0 \\ 0 \\ -1 \\ -1 \\ -1 \\ -1 \end{array} \quad \begin{array}{c} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{array} \quad \begin{array}{c} 0 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{array}$$

whatever is not zero is a 1's

0) find prime implicants:
Prime implicants:

$$0 - 0 (0, 2)$$

$$-1 - (2, 3, 6, 7)$$

$$\bar{A} \bar{C}$$

$$B$$

Finding essential prime implicants:

	0	2	3	6	7
0	X	X			
1		X	X	X	X

Essential:

$$(B(2, 3, 6, 7))$$

In the above eg both are essential prime implicants.

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

$$2) Y = f(A, B, C) = \Sigma m(0, 2, 6, 7)$$

②

$$\begin{array}{ccc} A & B & C \\ 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \end{array}$$

$$\begin{array}{l} 0 - 0 (0, 2) \\ -10 (2, 6) \\ 11 - (6, 7) \end{array}$$

All are prime implicants.

prime implicants

$$\bar{A} \bar{C} (0, 2)$$

$$B \bar{C} (2, 6)$$

$$AB (6, 7)$$

	0	2	6	7
0	X		X	
1		X		X

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

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

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

$$y = f(A, B, C) = \Sigma m(1, 2, 5, 7) + d(3, 4)$$

and not
essential p.t.

	0 0 1	$\bar{D} - 1(1, 3) \checkmark$
1	0 1 0 (1)	$01 - (2, 3)$
2	1 0 0	$-01(1, 5) \checkmark$
3	0 1 1 (2)	$10 - (4, 5)$
4	1 0 1 (3)	$-11(3, 7) \checkmark$
5	1 1 1 (4)	$-11(5, 7) \checkmark$
6	1 1 0 (5)	
7	1 1 1 (6)	

prime implicants

$\bar{A}B(2, 3)$

$A\bar{B}(1, 5)$

$\bar{C}(1, 3, 5, 7)$

essential prime implicants:

	1	2	5	7
Don't care	X			
W.R.B. 8/10/2023		X		
W.R.B. 8/10/2023			X	
W.R.B. 8/10/2023			X	X
W.R.B. 8/10/2023				

$$Y = AB + BC$$

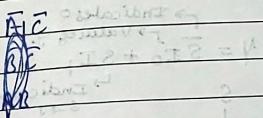
	$\bar{B}C$	$\bar{B}C$	BC	$B\bar{C}$
A_0	00	01	11	10
A_1	X4	1	X5	12

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

$$f(A, B, C) = \Sigma m(0, 2, 4) + d(6, 7)$$

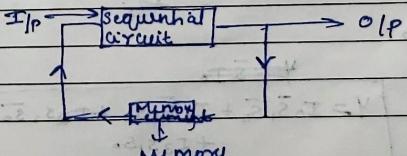
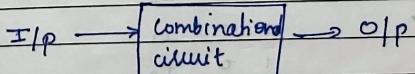
A	B	C	
0	0	0 0	6IV
2	0	1 0	0IV
4	1	0 0	
6	1	1 0 (6) 3V	
7	1	1 1 6IV	-10 (2, 6) V

~~A, B, C~~
~~(0, 2, 4)~~
~~(0, 4, 2)~~
~~(6, 7)~~
~~(4, 6)~~
~~(2, 6)~~



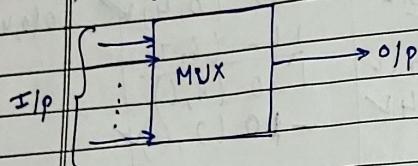
logic circuits:

- combinational circuit: o/p depends on present i/p.
- sequential circuit: o/p depends on present i/p and previous output.



* Combinational circuit:

i) Multiplexer is a combinational circuit with many inputs and one output.



2:1 MUX

Truth Table

I_0	I_1	S	Y
0	0	I_0	I_0
1	1	I_1	I_1

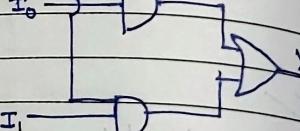
$Y = \bar{S}I_0 + S I_1$

Indicates value of y .
Indicates $S=1$.

selections

When $S=0$, I_0 will be output.

When $S=1$, I_1 will be output.

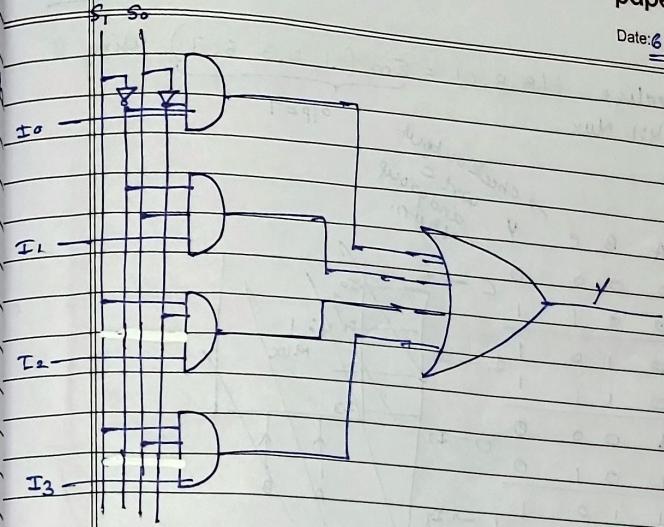


* 4:1 MUX

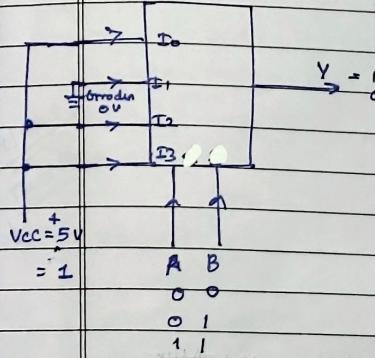
I_0	I_1	I_2	I_3	S_1	S_0	Y
0	0	0	0	0	0	I_0
0	0	0	1	0	0	I_1
0	0	1	0	0	1	I_2
0	0	1	1	0	1	I_3
1	0	0	0	1	0	0
1	0	0	1	1	0	1
1	0	1	0	1	1	0
1	0	1	1	1	1	1

$00, 01, 10, 11$ will be off

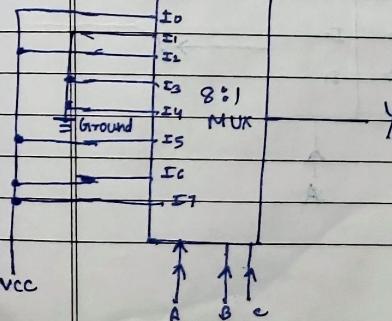
$Y = I_0\bar{S}_1\bar{S}_0 + I_1\bar{S}_1S_0 + I_2S_1\bar{S}_0 + I_3S_1S_0$



* Eg $f(A, B) = \sum m(0, 2, 3)$ using 4:1 MUX



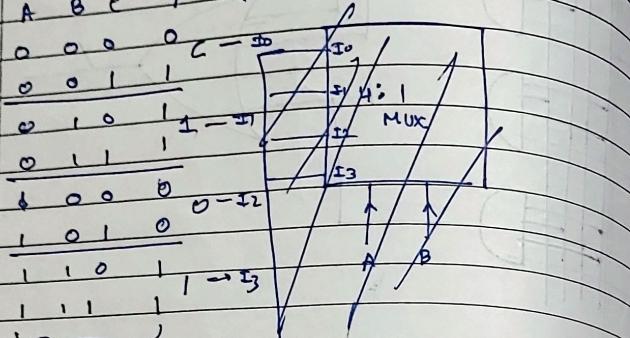
② Realize $f(A, B, C) = \sum m(0, 2, 5, 6, 7)$



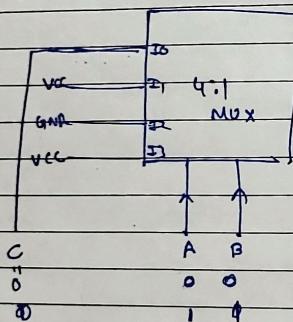
23

A3 Realize $f(A, B, C) = \Sigma m(1, 2, 3, 6, 7)$ using 4:1 Mux

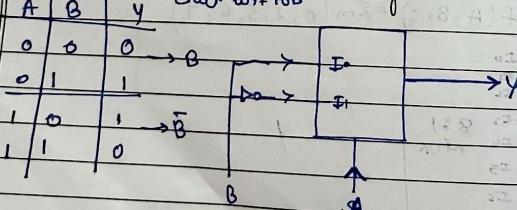
check output
with C
and group
them.



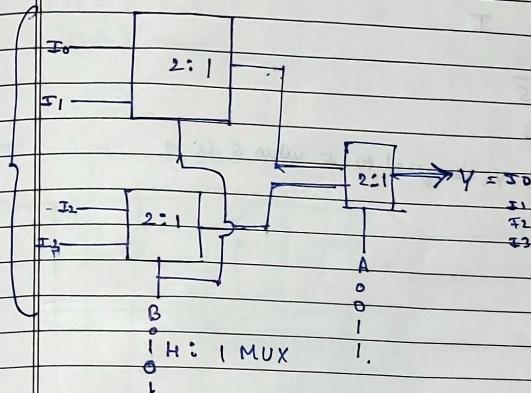
By seeing the
question



④ Realize $f(A, B) = \Sigma m(1, 2)$ using 2:1 MUX.



Q Design 4:1 MUX using 2:1 Multiplexers:



A	B	Y
0	0	I_0
0	1	I_1
1	0	I_2
1	1	I_3

Hence truth table
verified.

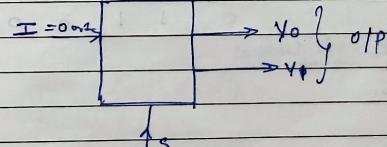
1/10/23
Thursday

2) * Demultiplexer:

Combinational circuit with one input and many outputs.

For example:

1 to 2 Demux



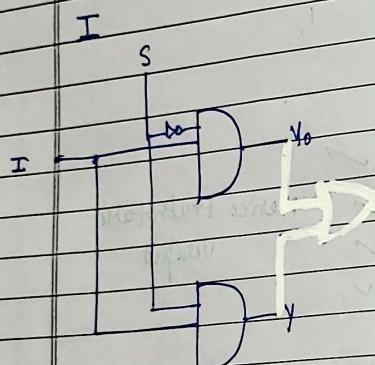
$S=0 \rightarrow Y_0$ will be selected
 $S=1 \rightarrow$ Input will be available through Y_1 .

S	y_0	y_1
0	I	0
1	0	I

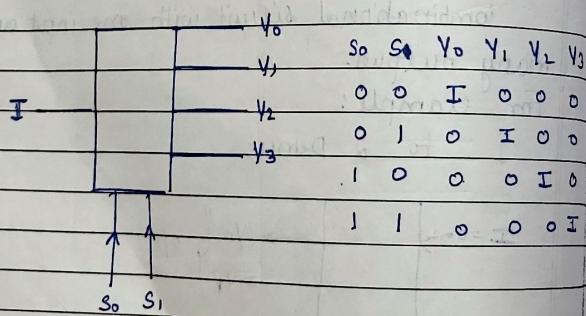
$$y_0 = I \bar{S}$$

$$y_1 = IS$$

more y_1 is equal to I when S is 1.



(2) 1 to 4 Demux:

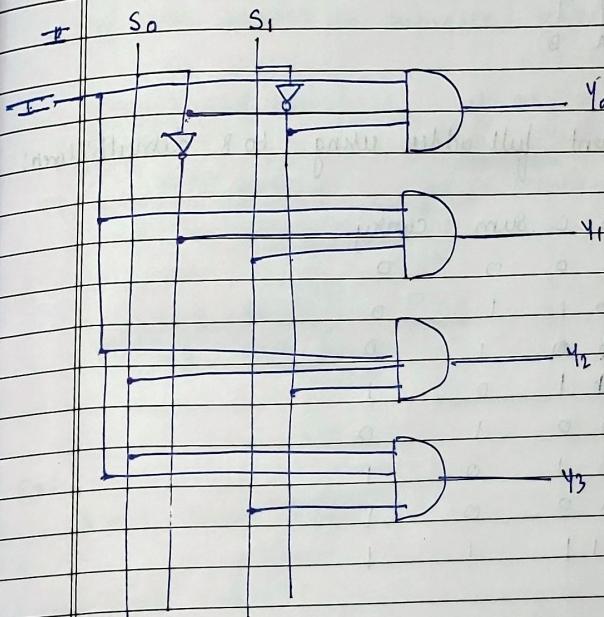


$$y_0 = I \bar{S}_0 \bar{S}_1$$

$$y_1 = I \bar{S}_0 S_1$$

$$y_2 = I S_0 \bar{S}_1$$

$$y_3 = I S_0 S_1$$



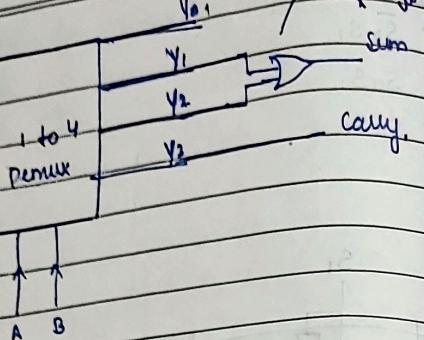
Implement Half adder using 1 to 4 demux.

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

→ Connect these numbers to OR gate.

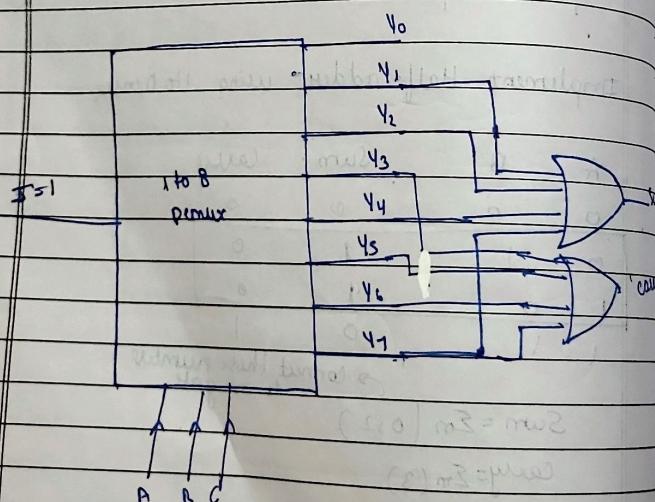
$$\text{Sum} = \Sigma_m (0, 2)$$

$$\text{Carry} = \Sigma_m (3)$$



② Implement full adder using 1 to 8 demultiplexer

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

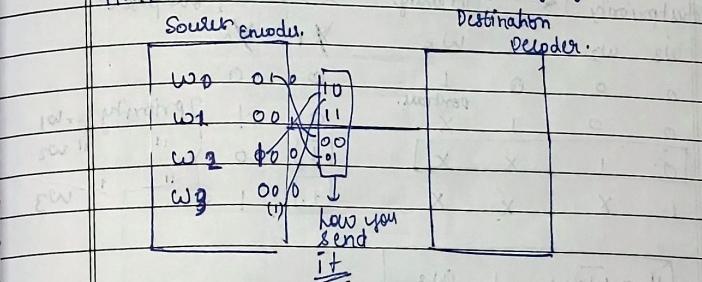


28

3) * Encoder is a combinational circuit that encodes given information into a more compact form.

i) Binary encoder: A binary encoder encodes information from 2^n inputs into a n bit code.
Encoders are used to reduce the number of bits needed to represent given information.

only one of the ip is 1 at any given instant of time.



Ex:

4 to 2 Binary encoder

Truth table

Input				Output
w ₃	w ₂	w ₁	w ₀	y ₁ , y ₀
0	0	0	1	0 0
0	0	1	0	0 1
0	1	0	0	1 0
1	0	0	0	1 1

$$Y_1 = W_2 + W_3 \quad \left\{ \text{Equation from truth table.} \right.$$

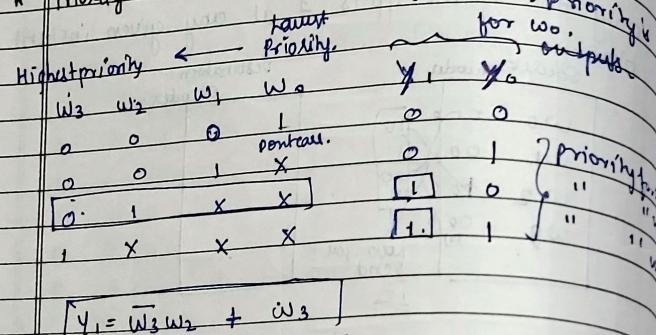
$$Y_0 = W_1 + W_2 \quad \left. \right\}$$

$$Y_1 = \overline{W_3} W_2 \overline{W_1} \overline{W_0} + W_3 \overline{W_2} \overline{W_1} \overline{W_0} \quad \left\{ \text{This also correct.} \right.$$

$$Y_0 = \overline{W_3} \overline{W_2} W_1 \overline{W_0} + W_3 \overline{W_2} \overline{W_1} \overline{W_0} \quad \left. \right\}$$

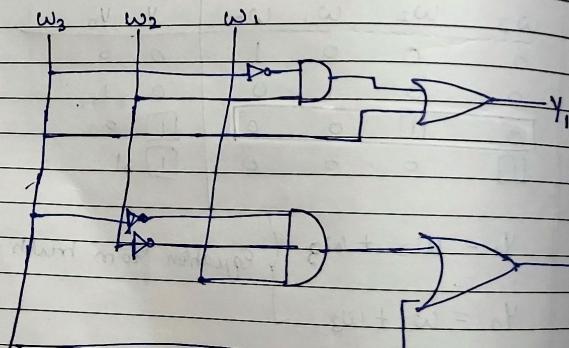
11/10/23
Monday

* Priority encoder:

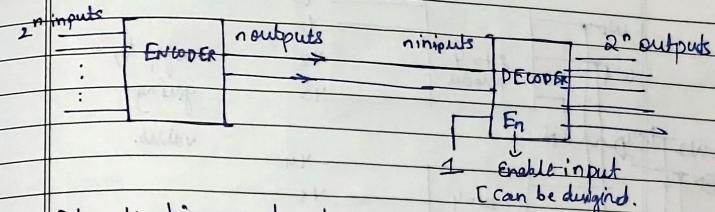


$$y_0 = \bar{w}_3 \bar{w}_2 w_1 + w_3$$

Circuit diagram:



- u) Decoder is a combinational circuit used to decode encoded information.



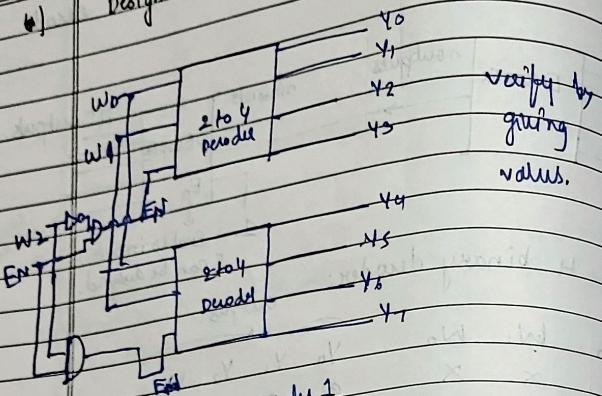
2 to 4 binary decoder: outputs.

E_n	w_1	w_0	y_0	y_1	y_2	y_3
0	X	X	0	0	0	0
1	0	0	1	0	0	0
1	0	1	0	1	0	0
1	1	0	0	0	1	0
1	1	1	0	0	0	1

$$\begin{aligned}y_0 &= E_n \bar{w}_1 \bar{w}_0 \\y_1 &= E_n \bar{w}_1 w_0 \\y_2 &= E_n w_1 \bar{w}_0 \\y_3 &= E_n w_1 w_0\end{aligned}$$



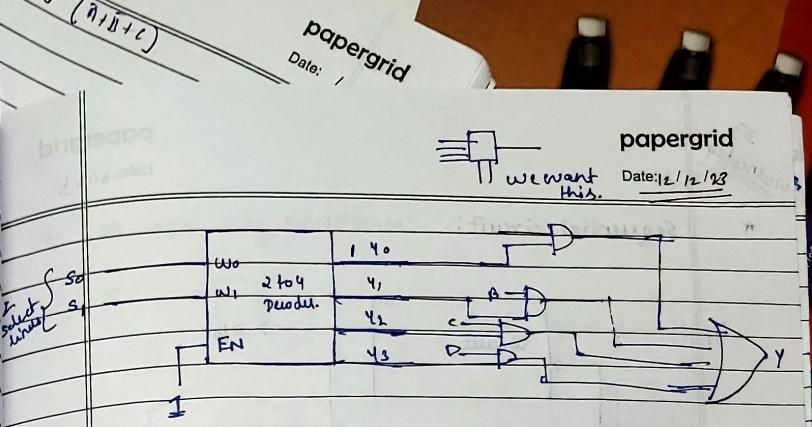
Design a 3 to 8 Decoder using 2 to 4 Decoder



	W2	W1	W0	Y0	Y1	Y2	Y3	Y4	Y5	Y6	Y7
EN	0	0	0	1	0	0	0	0	0	0	0
	0	0	1	0	1	0	0	0	0	0	0
	0	1	0	0	0	1	0	0	0	0	0
	0	1	1	0	0	0	1	0	0	0	0
	1	0	0	0	0	0	0	1	0	0	0
	1	0	1	0	0	0	0	0	1	0	0
	1	1	0	0	0	0	0	0	0	1	0
	1	1	1	0	0	0	0	0	0	0	1
0	x	x	x								

Decoder 2

Design 4 to 1 multiplexer using decoder and basic gates.



MUX:

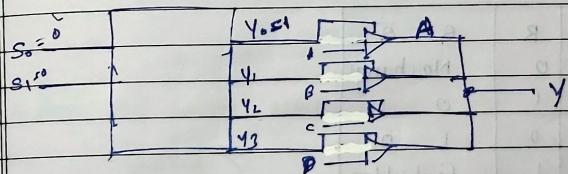
S1	S0	Y
0	0	A
0	1	B
1	0	C
1	1	D

Design 4 to 1 multiplexer using decoder and tristate buffer.

(EN acts as switch)



If $EN=1$ then $out = in$.

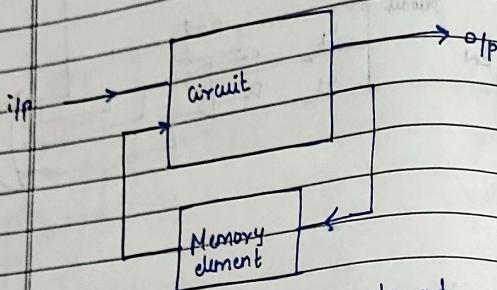


S1 S0

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

93
wednesday

* Sequential circuit:



Memory element (latch/Flipflop) ^{memory element where you can store 0 or 1 if you can}
store 0 or 1 if you can

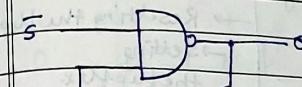
Memory element

1) SR latch
Set Reset

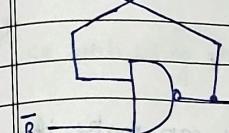
		whatever is given in set will be stored in the memory element	
		Storing 0	Storing 1
S=1	R=0	$Q=0$	$\bar{Q}=1$
R=1	S=0	$Q=1$	$\bar{Q}=0$

S	R	Q	\bar{Q}
0	0	No change	
0	1	0	1
1	0	1	0
1	1	Forbidden	

+ SR Latch using NAND gate:



Active High i/p



\bar{S}	\bar{R}	Q	\bar{Q}
0	0	Forbidden	
0	1	1	0
1	0	0	1
1	1	No change	

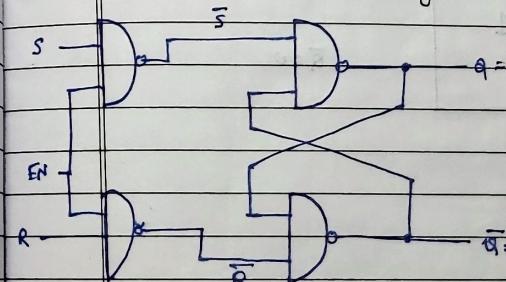
Active High i/p

$\rightarrow S$ should be 1 to make latch 1.

S	R	Q	\bar{Q}
0	0	No change	
0	1	0	1
1	0	1	0
1	1	Forbidden	

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

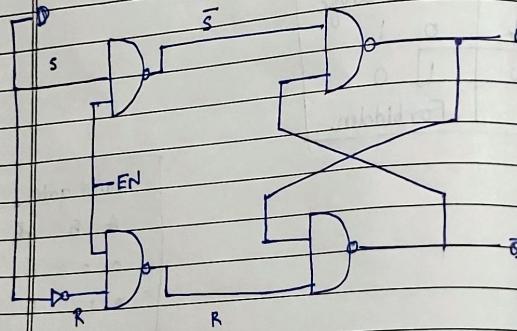
+ gated SR latch using nand gate:



EN	S	R	Q	\bar{Q}	
1	0	0	No change		
1	0	1	1	0	→ Resetting the flip flop.
1	1	0	0	1	→ Setting the flip flop.
1	1	1	Forbidden		
0	X	X	No change		

Enable acts as a switch
when EN=1 then only you can make changes
when EN is 0 then switch is off.

Unlatched D latch using NAND gate:
To avoid forbidden case

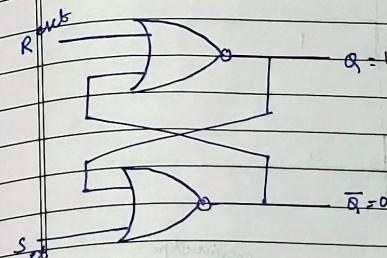


D = 1	S = 1	R = 0	Q = 1	$\bar{Q} = 0$

EN	D	Q	\bar{Q}	
1	0	0	1	
1	1	1	0	
0	X	No change		

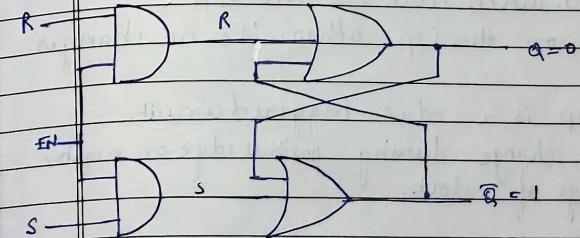
If its gated the it should have enable.

SR latch using NOR gate:



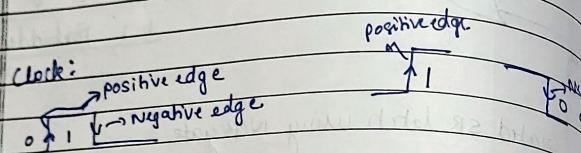
S	R	Q	\bar{Q}
0	0	No change	
0	1	0	1
1	0	1	0
1	1	Forbidden	

Gated SR latch using NOR gate



EN	S	R	Q	\bar{Q}
1	0	0	No change	
1	0	1	0	1
1	1	0	1	0
0	X	X	No change	

Timing Diagrams



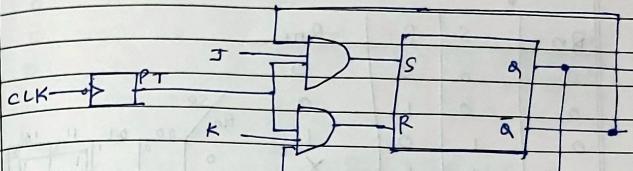
In latch when clock = 1 the o/p depends on the i/p otherwise no change.

Flip flop is a edge triggered circuit.
i.e. o/p change during positive edge or negative edge of clock.

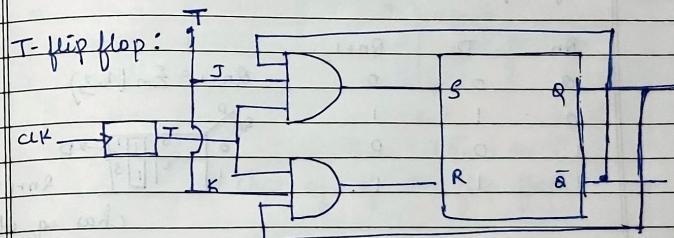
* JK flip flop:

Flip flop is a latch that has clocked input. It overcomes the disadvantages of SR latch (forbidden state) creates a toggle case.

(PT = positive transition pulse; generated during +ve edge of CLK
o/p = PT = 1 during +ve edge)



CLK	J	K	Q	\bar{Q}
↑	0	0	No change	
↑	0	1	0	1
↑	1	0	1	0
↑	1	1	toggle (opp to previous value)	



CLK	T	Q	\bar{Q}
↑	0	no change	
↑	1	toggle	

Characteristic equation of flip flops (used to analyse the circuit)

1. SR flip flop:

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

$\rightarrow S = Q_n \oplus R$

$\therefore Q_{n+1} = S + \bar{R} Q_n$

\Rightarrow characteristic eq for SR-FF

2. D flip flop:

Q_n	D	Q_{n+1}
0	0	0
0	1	1
1	0	0
1	1	1

$\rightarrow D = Q_n$

$\therefore Q_{n+1} = \Sigma m(1, 3)$

\Rightarrow char eq of D flip flop.

Derive the characteristic equation of D and JK flip flop.

3. J-K flip flop:

Q_n	J	R	Q_{n+1}
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1

$$Q_{n+1} = \Sigma m(2, 3, 4, 6)$$

Q_n	J	K	Q_{n+1}
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1

$$Q_{n+1} = J\bar{Q}_n + Q_n\bar{K}$$

\Rightarrow char eq. of JK flip flop.

4. Toggle Flip flop:

Q_n	T	Q_{n+1}
0	0	0
0	1	1
1	0	1
1	1	0

$$Q_{n+1} = \Sigma m(1, 2)$$

Q_n	T	Q_{n+1}
0	0	1
0	1	0
1	0	1

$$Q_{n+1} = T\bar{Q}_n + Q_n\bar{T}$$

\Rightarrow char eq. of JK flip flop.

Excitation table:

SR - FF

S _n	N.S	S _{n+1}	R
0	0	0	0 or 1 X "Don't care."
0	1	1	0
1	0	0	1
1	1	0 or 1	0
		X	

D- Flip flop:

S _n	N.W	D
0	Q _{n+1}	0
0	0	0
0	1	1
1	0	0
1	1	1

JK flipflop:

Q _n	Q _{n+1}	J	K
0	0	0	X
0	1	1	X
1	0	X	1
1	1	X	0

S _n	Q _{n+1}	T
0	0	0
0	1	1
1	0	1
1	1	0

when T is 1 it toggles.

Till next.

* Registers : Group of flipflops

Types:



- 1) SISO
- 2) SIPO
- 3) PISO
- 4) PIPO.

③ Design

28/12/13
* Thursday

Unit 1

Code converter

i) BCD to 7 segment display

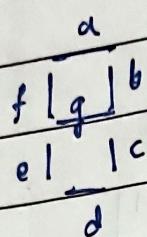
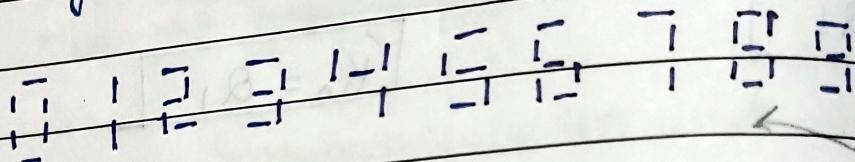
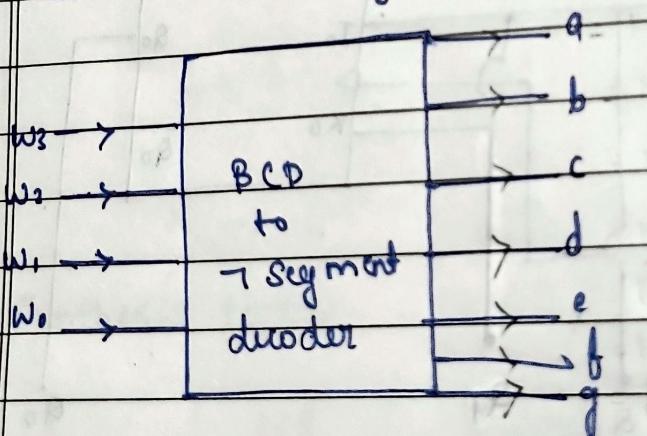


Fig: Seven segment display.



BCD to seven segment decoder:



BCD - Binary Decimal

0 → 0000

1 → 0001

:

:

9 → 1001

w_3	w_2	w_1	w_0	a	b	c	d	e	f	g
0	0	0	0	1	1	1	1	1	1	0
0	0	0	1	0	1	1	0	0	0	0
0	0	1	0	1	1	0	1	0	0	1
0	0	1	1	1	1	1	1	0	0	1
0	1	0	0	0	1	1	0	0	0	0
0	1	0	1	1	0	1	1	0	1	1
0	1	1	0	1	0	1	1	1	1	1
0	1	1	1	1	1	1	0	0	0	0
1	0	0	0	1	1	1	1	1	1	1
1	0	0	1	1	1	1	1	0	1	1
1	0	1	0	x	x	x	x	x	x	x
1	0	1	1	x	x	x	x	x	x	x
1	1	1	1	x	x	x	x	x	x	x
1	1	1	1	x	x	x	x	x	x	x

In exam write for any 1. if notine.

$$a = \sum m \{ 0, 2, 3, 5, 6, 7, 8, 9 \} + \sum d \{ 10, 11, 12, 13, 14, 15 \}$$

$$g = \sum m \{ 2, 3, 4, 5, 6, 8, 9 \} + \sum d \{ 10, 11, 12, 13, 14, 15 \}$$

$$d \{ 10, 11, 12, 13, 14, 15 \}$$

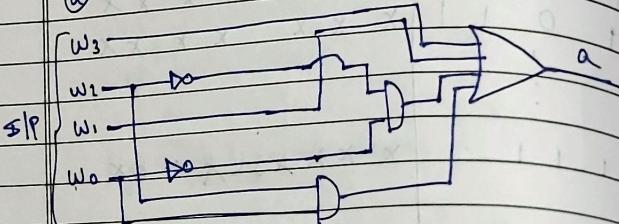
55

	$\bar{w}_1 \bar{w}_0$	$\bar{w}_3 \bar{w}_2$	$\bar{w}_3 w_0$	$w_1 w_0$	$w_3 \bar{w}_2$	$w_3 w_2$	$w_1 \bar{w}_0$	$\bar{w}_1 w_0$	$w_1 w_0$
1	1	1	1	1	1	1	1	1	1
2	1	1	1	1	1	1	1	1	1
3	1	1	1	1	1	1	1	1	1
4	1	1	1	1	1	1	1	1	1
5	1	1	1	1	1	1	1	1	1
6	1	1	1	1	1	1	1	1	1
7	1	1	1	1	1	1	1	1	1
8	1	1	1	1	1	1	1	1	1
9	1	1	1	1	1	1	1	1	1
10	1	1	1	1	1	1	1	1	1

$$a = w_3 + w_1 + \bar{w}_2 \bar{w}_0 + w_2 w_0$$

 $g =$

@



* Gray code: It is a binary numeral system where two successive values differ in only one bit. It reduces the switching operations.

papergrid

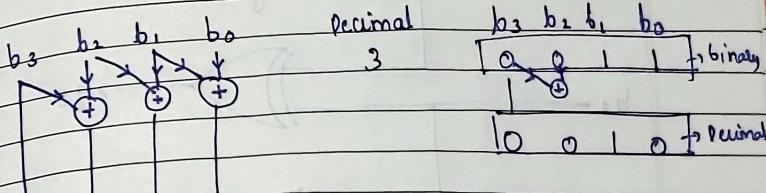
Date:

A	B	X OR Y
0	0	0
0	1	1
1	0	1
1	1	0

papergrid

Date: / /

Binary code to Gray code:

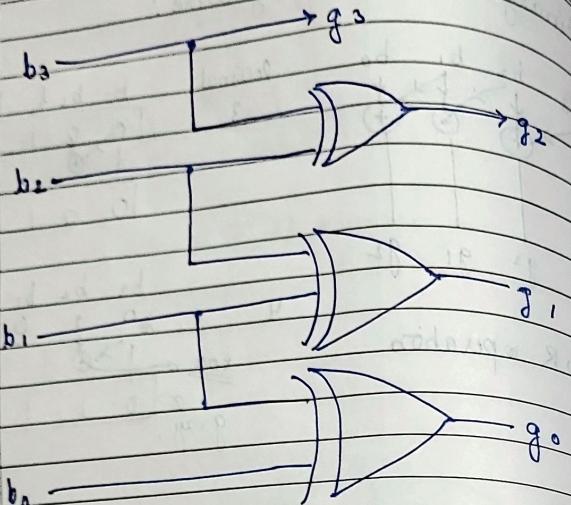
g₃ g₂ g₁ g₀

4

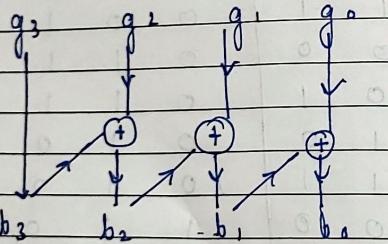
XOR operation

gray.

b ₃	b ₂	b ₁	b ₀	g ₃	g ₂	g ₁	g ₀
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	1
0	0	1	0	0	0	1	0
0	0	1	1	0	0	1	0
0	1	0	0	0	1	1	0
0	1	0	1	0	1	1	1
0	1	1	0	0	1	0	1
0	1	1	1	0	1	0	0
1	0	0	0	1	1	0	0
1	0	0	1	1	1	1	1
1	0	1	0	1	1	1	0
1	0	1	1	1	1	1	1
1	1	0	0	1	0	1	0
1	1	0	1	1	0	1	1
1	1	1	0	1	0	0	1
1	1	1	1	1	0	0	0



* Gray code to Binary code.



0 0 1 0 → gray code.

0 0 1 1 → Binary code

1 0 0 0 → gray code

1 1 1 1 → Binary.

$$b_3 = g_3$$

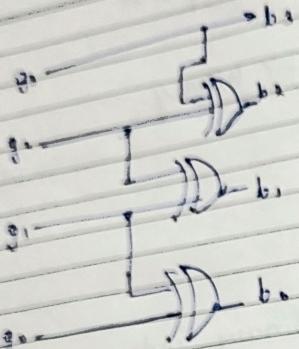
$$b_2 = b_3 \oplus g_2$$

$$b_1 = b_2 \oplus g_1$$

$$b_0 = b_1 \oplus g_0$$

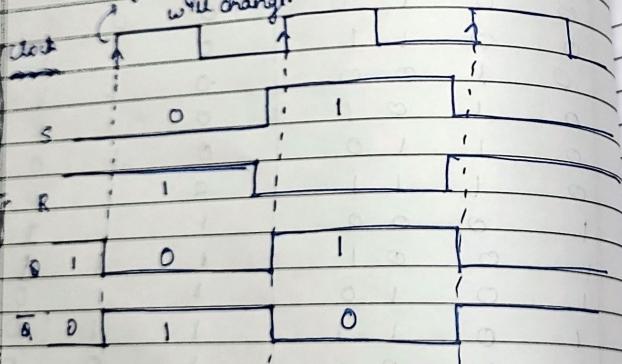
Gray code to Binary code

	g_3	g_2	g_1	g_0		b_3	b_2	b_1	b_0
0	0	0	0	0		0	0	0	0
0	0	0	0	1		0	0	0	1
0	0	0	1	0		0	0	1	1
0	0	0	1	1		0	0	1	0
0	0	1	0	0		0	1	1	1
0	0	1	0	1		0	1	1	0
0	0	1	1	0		0	1	0	0
0	0	1	1	1		0	1	0	1
1	0	0	0	0		1	1	1	1
1	0	0	0	1		1	1	0	0
1	0	0	1	0		1	1	0	0
1	0	0	1	1		1	0	0	0
1	0	1	0	0		1	0	1	0
1	0	1	0	1		1	0	1	1
1	0	1	1	0		1	0	0	0
1	0	1	1	1		1	0	0	1
1	1	0	0	0		1	0	1	0
1	1	0	0	1		1	0	0	1
1	1	0	1	0		1	0	1	0
1	1	0	1	1		1	0	1	1
1	1	1	0	0		1	0	0	0
1	1	1	0	1		1	0	0	1
1	1	1	1	0		1	0	1	0
1	1	1	1	1		1	0	1	1



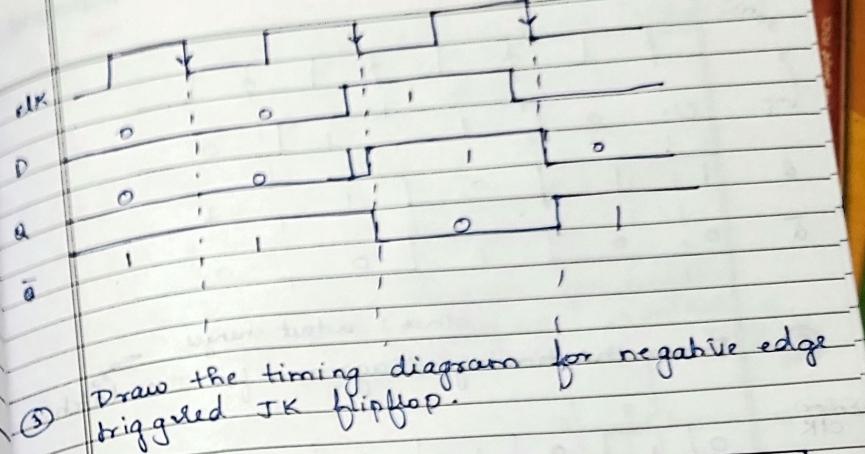
(Q1) Write the timing diagram for positive edge triggered SR flip flop.

During every positive edge output will change.

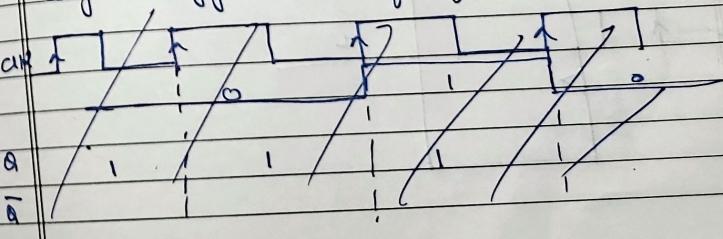


(Q2) Write the timing diagram for negative edge triggered D flip flop.

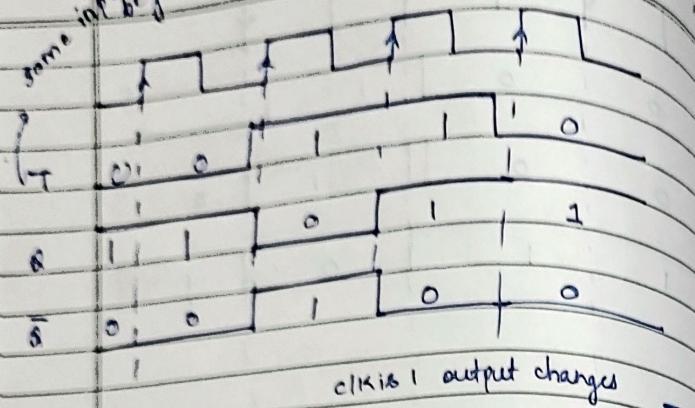
soln:



(Q4) Draw the timing diagram for positive edge triggered T flip flop.

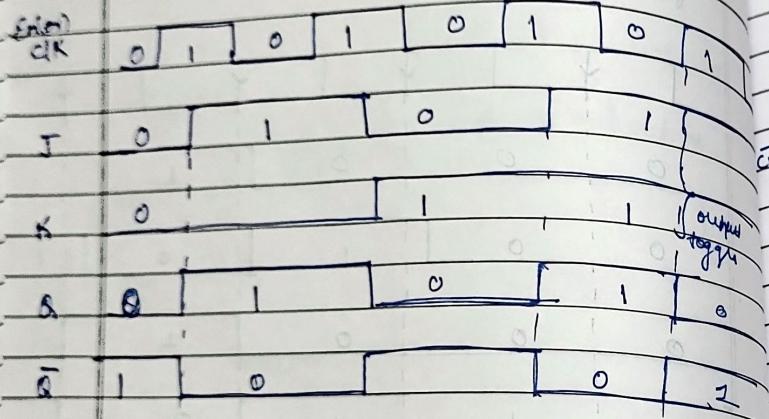


same input should begin

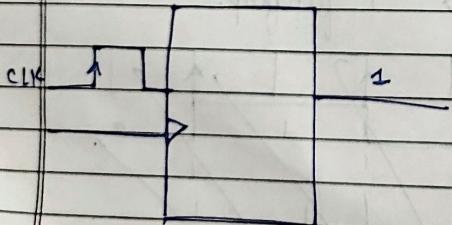


clk is 1 output changes

- ⑤ Draw the timing diagram for JK latch



* clock symbol for positive edged triggered circuit:



Digital
Date:

(A, B, C). (A, E)

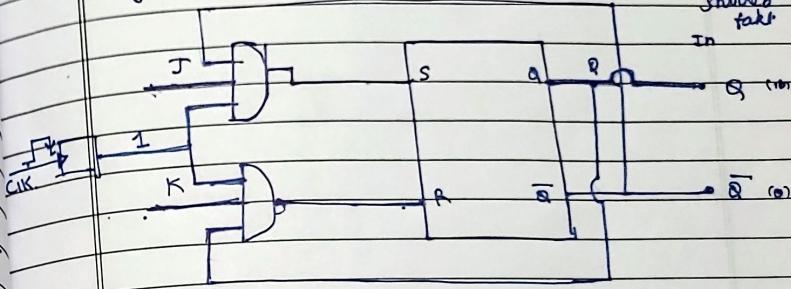
papergrid

Date: / /

Clock symbol for negative edge triggered circuit



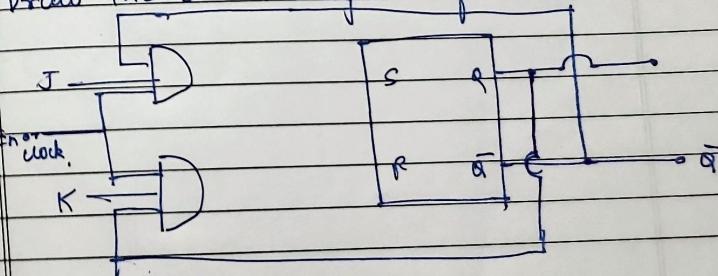
- ⑥ Draw the circuit diagram for negative edge triggered JK flip flop. Initially q value will be shown take



CLK	J	K	Q	\bar{Q}
↓	0	0	0	1
↓	0	1	0	1
↓	1	0	1	0
↓	1	1	1	0

toggle (opp to previous value)

- ⑦ Draw the circuit diagram for JK latch:



EN J K Q \bar{Q}

1 0 0 No change
1 0 1 0 1
1 1 0 1 0
1 1 1 Toggle
0 X X No change.