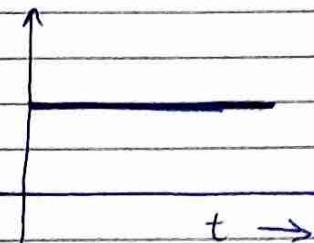


Unit - I
 Reference Book
 → R.P. Jain

→ Digital signals / discrete


 * $V_{max} = 5V$
 $(3.5V - 5V)$
 $\text{max} \leftrightarrow \text{min}$

 * Frequency = 0 Hz
 * Time bounded
Logic Gates:

The combination of either 2 same input or 2 different input which provides a suitable output either in the form of 0 i.e. off/false or in the form of 1 i.e. on/true is known as logic gates.

There are total 7 types of gates among which AND, OR, NOR are known as BASIC GATES.
 ALL GATES / AOI.

NAND, NOR → Universal gates
 X-OR, X-NOR → Exclusive gates

- AND, OR, NAND, NOR → "N" Input Gates
- X-OR, X-NOR → 2 Input gates
- NOT → Single input gate

(a) OR Gate:

$$A + B = Y$$

Truth Table:

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

(b) AND Gate:

$$A \cdot B = Y$$

Truth Table:

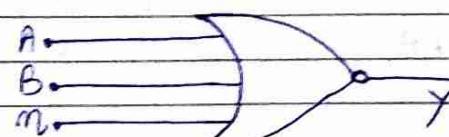
A	B	Y
0	0	0
0	1	0
1	0	0
1	1	1

(c) NOT Gate:

$$\bar{A} = Y$$

Truth Table:

A	Y
0	1
1	0

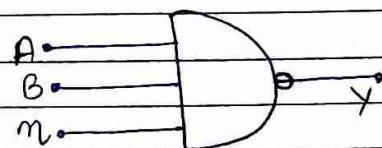
(d) NOR Gate:

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

Truth Table:

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

(e) NAND Gate:

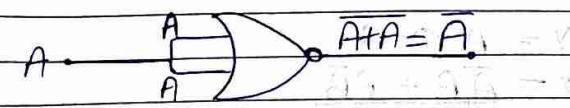


$$\overline{A \cdot B} = Y$$

Truth Table:

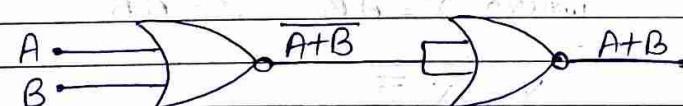
A	B	Y
0	0	1
0	1	1
1	0	1
1	1	0

1. NOT Gate (\overline{A}):



$$\overline{A+A} = \overline{A} \quad \checkmark$$

2. OR Gate ($A+B$):



$$A+B$$

3. AND Gate ($A \cdot B$):

De Morgan's Theorem

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

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

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

(f) X-OR Gate:

Anticoincidence gate \rightarrow on same input, output will be zero



$$Y = A \oplus B$$

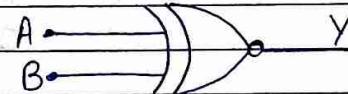
$$Y = \overline{A}B + A\overline{B}$$

Truth Table:

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

(g) X-NOR Gate:

Coincidence gate \rightarrow on different input, output will be zero



$$Y = A \odot B$$

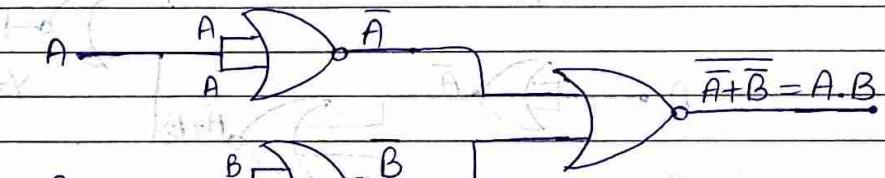
$$Y = \overline{A}B + A\overline{B}$$

Truth Table:

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

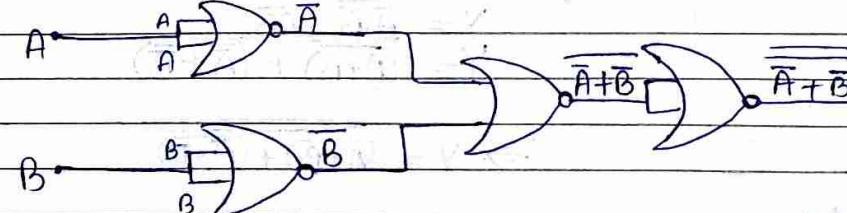
Realizations:

i) Using NOR Gate ($\overline{A} + \overline{B}$):



4. NAND Gate ($\overline{A} \cdot \overline{B}$):

$$Y = \overline{A \cdot B} = \overline{\overline{A} + \overline{B}} \quad (\text{DMT})$$



5. X-OR Gate: $(A \oplus B)$

$$Y = A \oplus B$$

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

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

$$Y = \overline{X} + Z$$

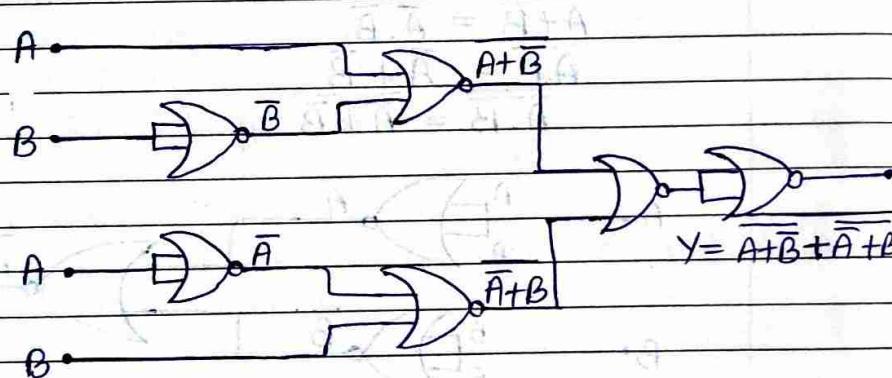
where $X = \overline{A}B$, $Z = A\overline{B}$

$$Y = \overline{X} \cdot \overline{Z}$$

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

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

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



6. X-NOR Gate:

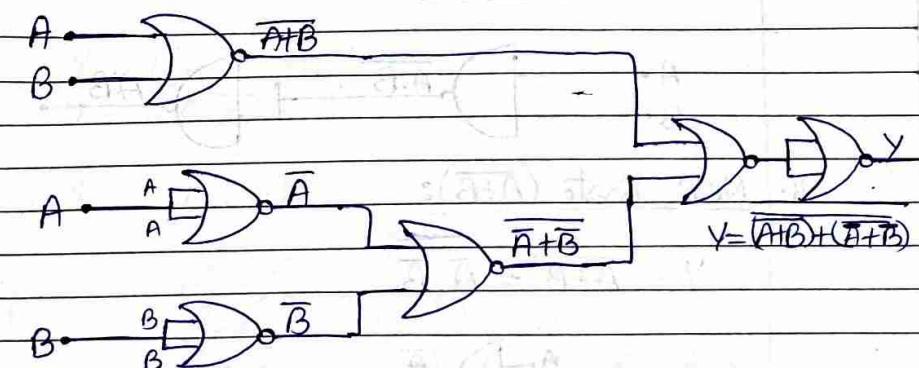
$$Y = \overline{A} \cdot \overline{B} + A \cdot B$$

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

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

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

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



→ Using NAND Gate ($\overline{A} \cdot \overline{B}$):

1. NOT Gate (\overline{A}):



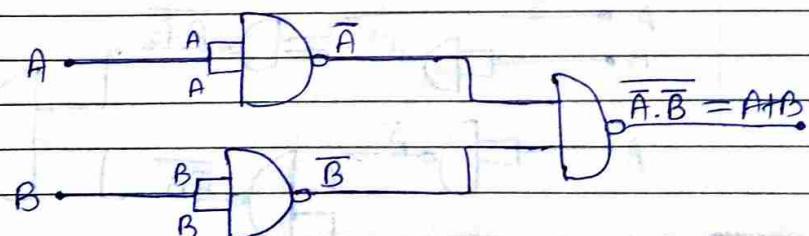
2. OR Gate ($A + B$):

By Morgan's theorem

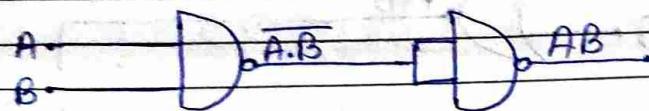
$$A + B = \overline{\overline{A} \cdot \overline{B}}$$

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

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

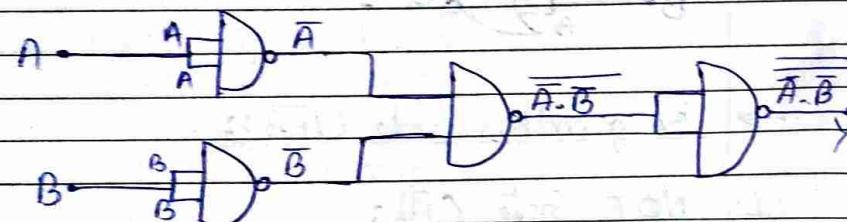


3. AND Gate ($A \cdot B$):



4. NOR Gate ($\overline{A+B}$):

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



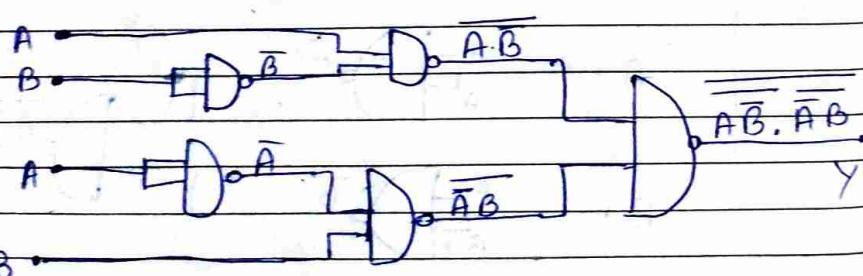
5. X-OR Gate: ($(A \oplus B)$)

$$Y = A \oplus B$$

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

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

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

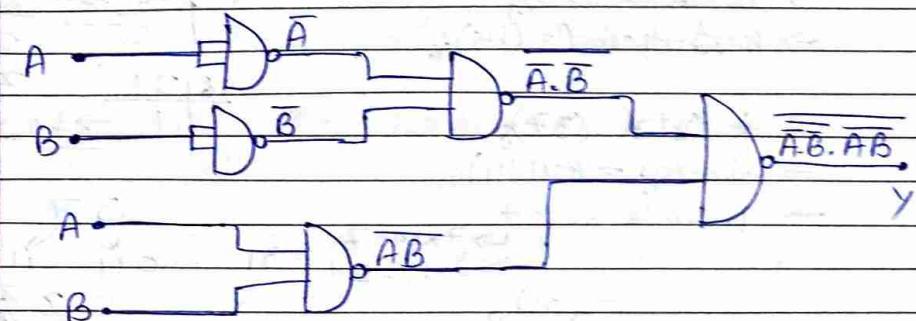


6. X-NOR Gate:

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

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

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



Number System:

Binary (2) (0-1)	Decimal (10) (0-9)	Octal (8) (0-7)	Hexa-Desimal (16) (0-9) (A-F)
------------------------	--------------------------	-----------------------	--

Conversions:

* Binary $(11011)_2$

→ Decimal = 11011

$$\begin{array}{r} \xrightarrow{1 \times 2^0 = 1} \\ \xrightarrow{1 \times 2^1 = 2} \\ \xrightarrow{1 \times 2^3 = 8} \\ \xrightarrow{1 \times 2^4 = 16} \end{array} = (27)_{10}$$

$$\begin{array}{r} \xrightarrow{1 \times 2^0 = 1} \\ \xrightarrow{1 \times 2^1 = 2} \\ \xrightarrow{1 \times 2^3 = 8} \\ \xrightarrow{1 \times 2^4 = 16} \end{array} \rightarrow \text{Octal} = \frac{0}{3} \frac{1}{3} \frac{0}{3} \frac{1}{3} = (33)_8$$

$$\rightarrow \text{Hexa-decimal} = \begin{array}{c} 0\ 0\ 0 \\ \downarrow \quad \downarrow \\ 1\ 0\ 1 \end{array} \\ = (1\ B)_{16}$$

$$\rightarrow \text{Decimal } (21)_{10}$$

$$\rightarrow \text{Binary} = (10101)_2$$

$$\rightarrow \text{Octal} = (25)_8$$

$$\rightarrow \text{Hexa-decimal} = (15)_{16}$$

$$\ast \text{ Octal } (37)_8$$

$$\rightarrow \text{Binary} = (11111)_2$$

$$\rightarrow \text{Decimal} = 37$$

$$\begin{array}{l} \xrightarrow{7 \times 8^0 = 7} \\ \xrightarrow{3 \times 8^1 = 24} = 31 \end{array}$$

$$\begin{array}{r} 8 | 21 & 2 | 21 \\ \downarrow & \downarrow \\ 2 \rightarrow 5 \uparrow & 2 | 0 \rightarrow 1 \\ \downarrow & \downarrow \\ 2 & 5 \rightarrow 0 \\ \downarrow & \downarrow \\ 16 | 21 & 2 | 2 \rightarrow 1 \\ \downarrow & \downarrow \\ 1 \rightarrow 5 \uparrow & 1 \rightarrow 0 \end{array}$$

$$\rightarrow \text{Hexa-decimal} = \begin{array}{c} 0\ 0\ 0 \\ \downarrow \quad \downarrow \\ 1\ 1\ 1 \end{array}$$

Octal \rightarrow Binary \rightarrow Hexa

$$= (1F)_{16}$$

$$\begin{array}{c} 3\ 7 \\ \downarrow \\ 0\ 1\ 1 \end{array}$$

$$\ast \text{ Hexa-decimal - } 2F$$

$$\rightarrow \text{Binary} = (10111)_2$$

$$\rightarrow \text{Decimal} = 2F$$

$$\begin{array}{l} \xrightarrow{15 \times 16^0 = 15} \\ \xrightarrow{2 \times 16^1 = 32} \uparrow \end{array}$$

$$= (47)_{10}$$

$$\rightarrow \text{Octal} = (10111)$$

(Hexa-Binary-Octal) $\xrightarrow[5]{?}$

$$= (57)_8$$

$$\begin{array}{c} 2\ F \\ \downarrow \\ 0\ 0\ 0 \quad 1\ 1\ 1 \end{array}$$

23.7.19

#

Karnaugh Map (K-Map):

Sum of Product

(SOP)

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

\rightarrow Complementary Variables

$$(\bar{A}, \bar{B}) \rightarrow 0$$

\rightarrow Non Complementary Variables

$$(A, B) \rightarrow 1$$

$\rightarrow 1 \cdot 1 + 0 \cdot 0 + 1 \cdot 0 + 0 \cdot 1$

$\rightarrow 0 + 0 \cdot 1 + 1 \cdot 0 + 1 \cdot 1 + 0$

$\rightarrow 0 \cdot 3 \cdot 1 \cdot 2$

$\rightarrow 3 + 0 + 2 + 1$ (Normal form)

\rightarrow In POS all terms

\rightarrow In SOP all variables are

known as Minterms

$\rightarrow \sum m(0, 1, 2, 3)$ [Representation]

\rightarrow Grouping \rightarrow always using 1.

\rightarrow Grouping \rightarrow 0

Cells Grouping:

1) 2 Variables :- (A, B)

$2^m \rightarrow$ no. of variables used

$$\Rightarrow 2^2 = 4 \text{ cells}$$

A	B	0	1
0	0	•	•
1	0	•	•

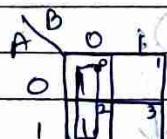
Addition & Subtraction:

$$\begin{array}{r} 1101 \\ + 1011 \\ \hline 11000 \end{array}$$

$$\begin{array}{r} 11011 \\ - 10111 \\ \hline 10000 \end{array}$$

Eg: $\sum m(0, 2)$ reduce this exp. using mapping

$$\text{Ans: } \sum m(0, 2)$$



$\Rightarrow A\bar{B} + A\bar{B}$ & $\bar{B}\bar{B}$ is common & it is the ans.

Rules of Grouping:

1. Make the largest group
2. Second priority is always given to corner group
3. Third priority is given to single bit/independent bit.

Eg: $\Sigma m(1, 2)$ using mapping

A	B	C
0	0	1
1	0	0

$$\Rightarrow (\bar{A}\bar{B})(\bar{A}+B) \text{ (Ans)}$$

2.) 3 Variables: - (A, B, C)

$$2^3 = 8 \text{ cells}$$

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

because of Gray codes
are those codes in which
2 bits are different
only 1 bit

Eg: $\Sigma m(0, 2, 4, 6, 7)$ using mapping

A	B	C
0	0	0
1	1	0

Group 1: (0, 2, 4, 6)
Group 2: (6, 7).

$$\begin{aligned} \text{Group 1: } & \bar{A} \bar{B} \bar{C} + A \bar{B} \bar{C} + A \bar{B} \bar{C} + A \bar{B} \bar{C} \\ & \Rightarrow \bar{C} \end{aligned}$$

$$\text{Group 2: } \bar{A} \bar{B} C + A \bar{B} \bar{C} = AB$$

$$\Rightarrow AB + \bar{C} \text{ (Ans)}$$

Eg: $\Pi M(0, 3, 4, 7)$ using Mapping

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

$$\begin{aligned} \text{Group 1: } & (A + B + C) + (\bar{A} + B + C) \Rightarrow B + C \\ \text{Group 2: } & (A + B + \bar{C}) (\bar{A} + B + \bar{C}) \Rightarrow \bar{B} + \bar{C} \\ & \Rightarrow (B + C)(\bar{B} + \bar{C}) \text{ (Ans.)} \end{aligned}$$

3.) 4 Variables: - (A, B, C, D)

$$2^4 = 16 \text{ cells}$$

A	B	C	D
00	0	0	0
01	0	0	1
11	0	1	0
10	0	1	1

Eg: $\Sigma m(0, 2, 8, 10, 5, 13)$ using mapping

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

$$\rightarrow \text{Grp 1} \rightarrow \bar{B}\bar{D} \quad ? \text{ SOP}$$

$$(\text{Grp 2} \rightarrow B\bar{C}D)$$

$$\Rightarrow \bar{B}\bar{D} + B\bar{C}D \quad (\text{Ans})$$

For POS:

$$\begin{cases} \text{Grp 1 } (1, 3, 9, 11) \rightarrow B + \bar{D} \\ \text{Pos } \begin{cases} \text{Grp 2 } (4, 6, 12, 14) \rightarrow \bar{B} + D \\ \text{Grp 3 } (5, 7, 13, 15) \rightarrow \bar{B} + \bar{C} \end{cases} \\ \Rightarrow (B + \bar{D})(\bar{B} + D)(\bar{B} + \bar{C}) \quad (\text{Ans}) \end{cases}$$

Don't care conditions: (d, ϕ , x)

$$\text{Eg: } \Sigma m(0, 9, 2, 10, 5) + d(7, 11)$$

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

$$\begin{aligned} \text{Grp 1} &\rightarrow (0, 2, 8, 10) \rightarrow \bar{B}\bar{D} \\ \text{Grp 2} &\rightarrow (5, 7) \rightarrow \bar{A}BD \end{aligned}$$

$$\text{Ans} = \bar{B}\bar{D} + \bar{A}BD$$

Note \rightarrow Grouping of only don't care variables only is not valid. Minimum one variable with don't care variables is necessary for grouping

24.7.19

4.) 5 Variables :- (A, B, C, D, E)
 $2^5 \rightarrow 32$ cells

$$\text{Eg: } \Sigma m(0, 8, 2, 10, 16, 24, 18, 26)$$

$\bar{B}C$	$\bar{D}E$	$\bar{D}\bar{E}$	$D\bar{E}$	$D\bar{E}$	$\bar{B}C$	$\bar{D}E$	$\bar{D}\bar{E}$	$D\bar{E}$	$D\bar{E}$	
$\bar{B}\bar{C}$	00	01	11	10	$\bar{B}\bar{C}$	00	11	10	11	10
$B\bar{C}$	00	01	11	10	$B\bar{C}$	00	11	10	11	10
$B\bar{C}$	00	01	11	10	$B\bar{C}$	00	11	10	11	10
BC	00	01	11	10	BC	00	11	10	11	10

$$\begin{aligned} \text{Ans} = & \bar{A}\bar{B}\bar{C}\bar{D}\bar{E} + A\bar{B}\bar{C}\bar{D}\bar{E} + \bar{A}\bar{B}\bar{C}\bar{D}\bar{E} + A\bar{B}\bar{C}\bar{D}\bar{E} + \\ & \bar{A}\bar{B}\bar{C}\bar{D}\bar{E} + \bar{A}\bar{B}\bar{C}\bar{D}\bar{E} + A\bar{B}\bar{C}\bar{D}\bar{E} + A\bar{B}\bar{C}\bar{D}\bar{E} \end{aligned}$$

$$= \bar{C}\bar{E}$$

5.) 6 Variables:- (A, B, C, D, E, F)
 $2^6 \rightarrow 64$ cells

$$\text{Eg: } \Sigma m(33, 17, 1, 49)$$

$\bar{C}D$	$\bar{E}F$	$\bar{E}\bar{F}$	$E\bar{F}$	$E\bar{F}$	$\bar{C}D$	$\bar{E}F$	$\bar{E}\bar{F}$	$E\bar{F}$	$E\bar{F}$	
$\bar{C}\bar{D}$	00	01	11	10	$\bar{C}\bar{D}$	00	11	10	11	10
$C\bar{D}$	00	01	11	10	$C\bar{D}$	00	11	10	11	10
$C\bar{D}$	00	01	11	10	$C\bar{D}$	00	11	10	11	10
CD	00	01	11	10	CD	00	11	10	11	10

$$AB = 01$$

AB = 10				AB = 11			
CD	EF	CD	EF	CD	EF	CD	EF
00	01	11	10	00	01	11	10
00	02	11	33	00	02	11	33
00	36	37	39	00	32	53	55
01	36	37	38	01	32	53	54
11	44	45	47	11	60	61	63
11	44	45	46	11	56	57	59
10	46	47	42	10	58	59	58

	8, 9	1 0 0 -
2	3, 7	0 - 1
	3, 11	- 0 1
	9, 11	1 0 - 1
3	7, 15	- 1 1
	11, 15	1 - 1

$$\text{Ans} = \bar{A}\bar{B}\bar{C}\bar{D}\bar{E}F + \bar{A}\bar{B}\bar{C}\bar{D}\bar{E}F + \bar{A}\bar{B}\bar{C}\bar{D}\bar{E}F + A\bar{B}\bar{C}\bar{D}\bar{E}F$$

$$= \bar{C}\bar{D}\bar{E}F$$

Quine - McClusky (Q-M) Method:

$$\text{Eg: } \Sigma m (0, 1, 3, 7, 8, 9, 11, 15)$$

$\downarrow 0$ $\downarrow 1$ $\downarrow 3$ $\downarrow 7$ $\downarrow 8$ $\downarrow 9$ $\downarrow 11$ $\downarrow 15$
 0000 0001 0011 0111 1000 1001 1011 1111

Group	Minterm	Variables
0	0, 1, 8, 9	A B C D
1	0, 8, 1, 9	- 0 0 -
1	1, 3, 9, 11	- 0 - 1
2	1, 9, 3, 11	- 0 - 1
2	3, 7, 11, 15	-- 1 1
3	3, 11, 7, 15	-- 1 1

Group	Minterm	Variables
0	0	A B C D
1	1	0 0 0 1
1	8	1 0 0 0
2	3	0 0 1 1
2	9	1 0 0 1
3	7	0 1 1 1
3	11	1 0 1 1
4	15	1 1 1 1

→ Prime Implicant Table (P.T)

P.I. Terms:

$$\bar{B}\bar{C} + \bar{B}D + CD$$

Essential Prime Implicant Table:

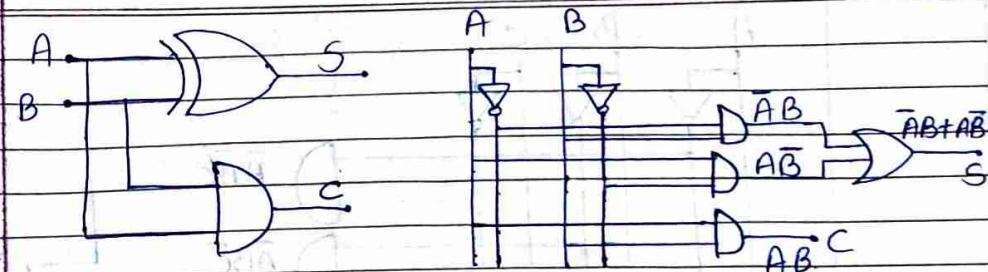
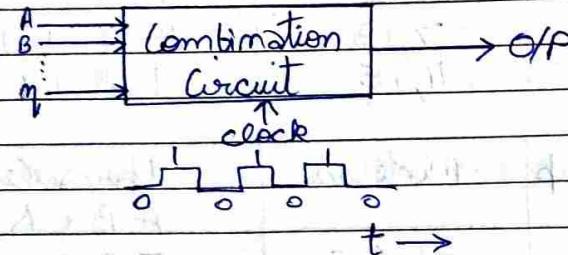
P.I. Terms	Minterms	Total Minterms
$\bar{B}\bar{C}$	0, 1, 8, 9	X X
$\bar{B}D$	1, 3, 9, 11	X X
CD	3, 7, 11, 15	X X

$$Y = \underbrace{\bar{B}\bar{C} + CD}_{\text{EPI Terms}}$$

Group	Minterm	Variables
0	0, 1	A B C D
1	0, 8	- 0 0 0
1	1, 3	0 0 - 1
1	1, 9	- 0 0 1

25-7-19

Combination Circuits:



(b) Full Adder : (A, B, C)

Classification of Combination Circuits:

I Adders → Half Adder (H.A) → 2 bits
→ Full Adder (F.A) → 3 bits
→ Half Subtractor (H.S) → 2 bits

II Subtractor → Full Subtractor (F.S) → 3 bits

III Multiplexers (MUX)

IV De-multiplexers (DMUX / DEMUX)

V Encoders

VI Decoders

VII Comparator / Magnitude comparator → 1 bit
→ 2 bits

I Adders:

(a) Half Adder : (A, B)

A | B | Sum | Carrier

0 | 0 | 0 | 0

0 | 1 | 1 | 0

1 | 0 | 1 | 0

1 | 1 | 0 | 1

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

$$C = AB$$

$$S = A \oplus B$$

$$C = AB$$

A | B | C | Sum | Carry

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

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

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

$$C = BC + AC + AB$$

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

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

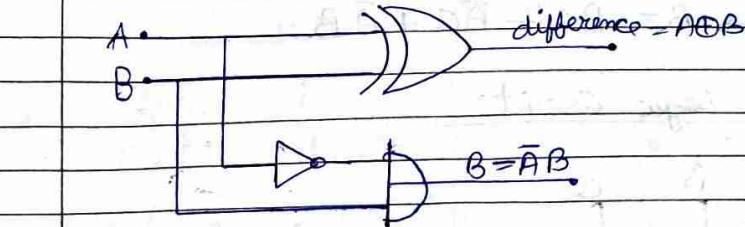
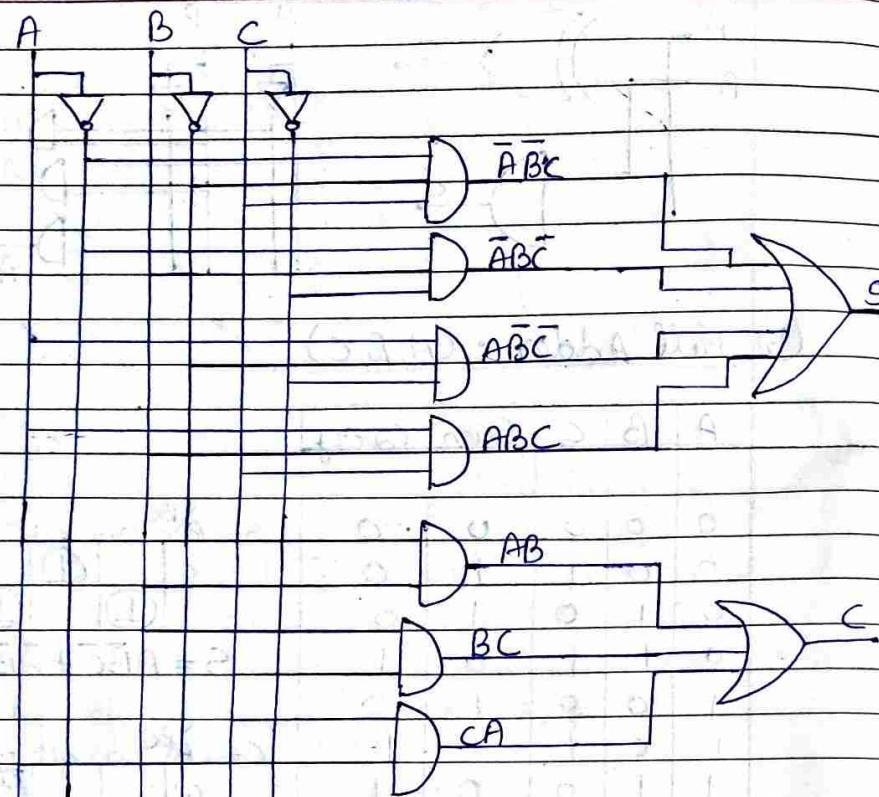
$$S = \bar{A} B \bar{C} + A \bar{B} \bar{C} + C$$

$$\text{Hence, } S = \bar{A} B \bar{C} + A \bar{B} \bar{C} + C$$

$$\& C = BC + AC + AB$$

AOI logic circuit is on the next page

(PTD).



(b) Full Subtractor:

A	B	C	Difference	Borrow
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	0	1
1	0	0	1	0
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1

(AOI Logic)

Difference

II Subtractors:

(a) Half Subtractor:

A B Difference Borrow

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

$$D = \begin{array}{|c|c|} \hline A & B \\ \hline 0 & 0 \\ \hline 0 & 1 \\ \hline 1 & 0 \\ \hline 1 & 1 \\ \hline \end{array} \quad D = \begin{array}{|c|c|} \hline A & B \\ \hline 0 & 1 \\ \hline 1 & 1 \\ \hline 1 & 0 \\ \hline 0 & 1 \\ \hline \end{array}$$

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

$$D = A \oplus B$$

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

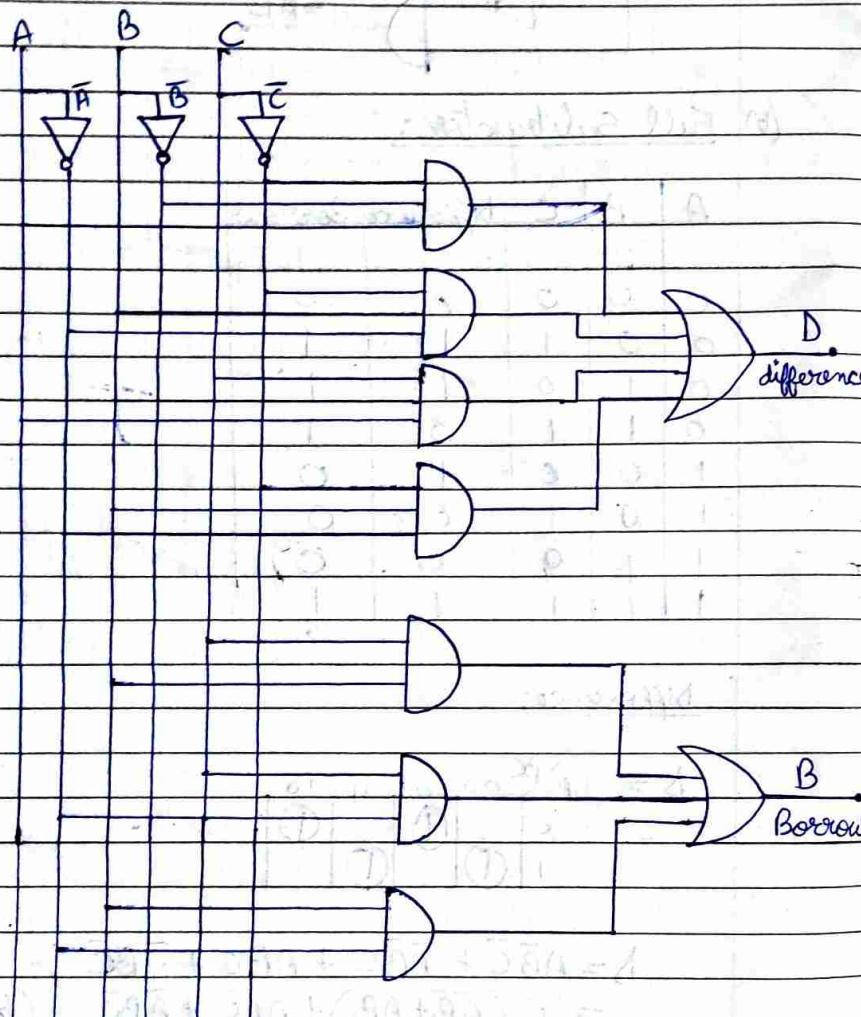
$$= C(\bar{A}\bar{B} + AB) + A\bar{B}\bar{C} + \bar{A}B\bar{C} \quad (\text{because } A\bar{A}=1)$$

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

$$B = \begin{array}{|c|c|} \hline A & B \\ \hline 0 & 0 \\ \hline 0 & 1 \\ \hline 1 & 0 \\ \hline 1 & 1 \\ \hline \end{array}$$

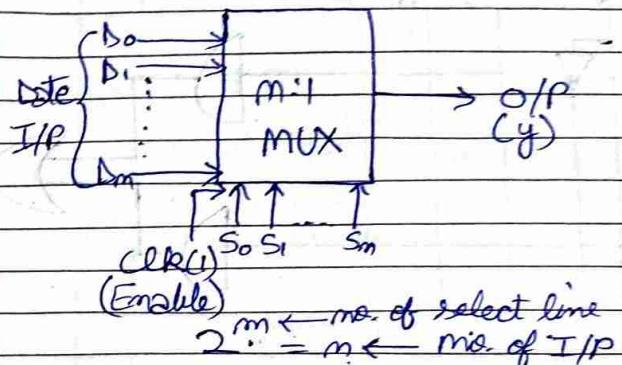
$$\Rightarrow B = BC + \bar{A}C + \bar{A}\bar{B}$$

AOI logic circuit:



(AOI logic)

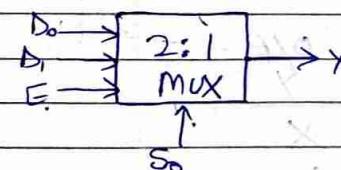
III Multiplexers (MUX)



Classification of Multiplexors:

- 1) $2:1$ MUX ($2^1=2$) \leftarrow I/P
- 2) $4:1$ MUX ($2^2=4$)
- 3) $8:1$ MUX ($2^3=8$)
- 4) $16:1$ MUX ($2^4=16$)
- 5) $32:1$ MUX ($2^5=32$)

i) $2:1$ MUX :-



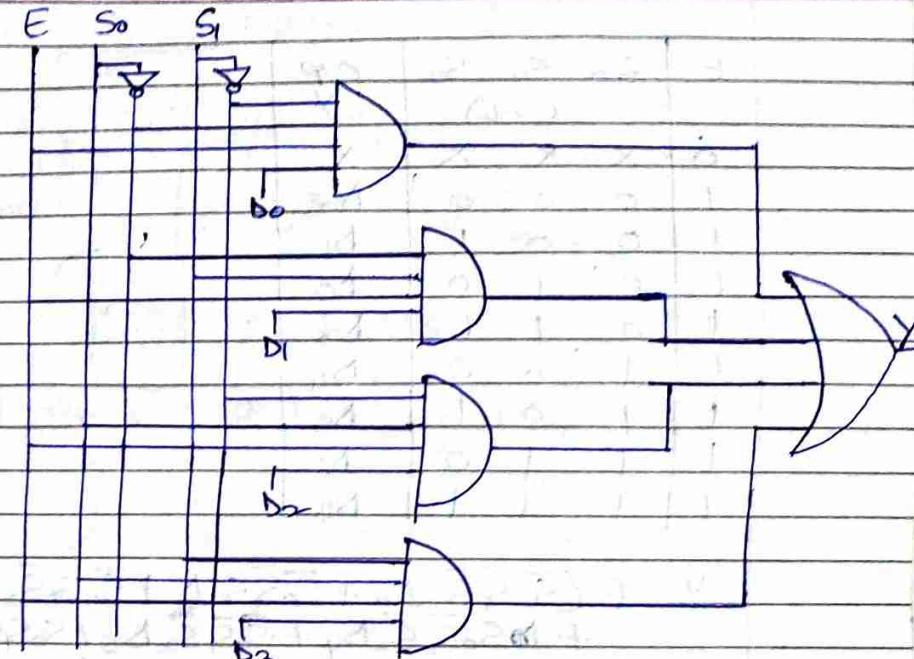
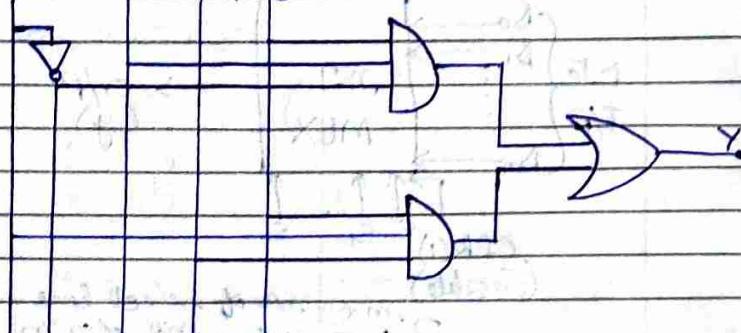
E	S-L	O/P
0	X	X

E	S-L	O/P
1	0	D0

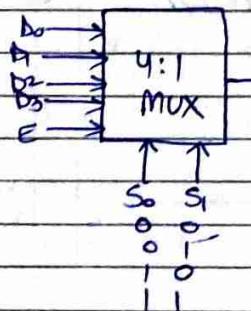
$$Y = E\bar{S}_0D_0 + ES_0D_1$$

$$\Rightarrow Y = E(S_0 \bar{D}_0 + S_0 D_1)$$

$S_0 \quad D_0 \quad D_1 \quad E$



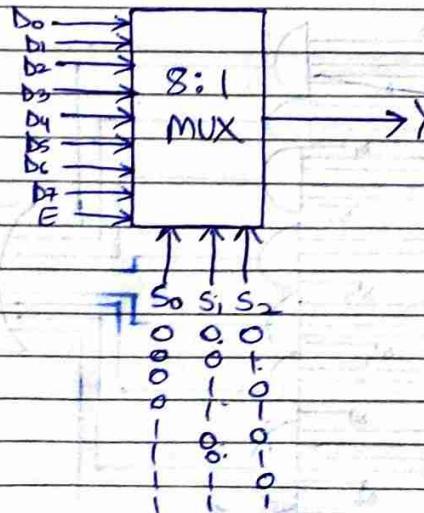
2.) 4:1 MUX :-



E	S_0	S_1	O/P
0	X	X	X
1	0	0	D0
1	0	1	D1
1	1	0	D2
1	1	1	D3

$$Y = E(\bar{S}_0 \bar{S}_1 D_0 + \bar{S}_0 S_1 D_1 + S_0 \bar{S}_1 D_2 + S_0 S_1 D_3)$$

3. 8:1 MUX :-



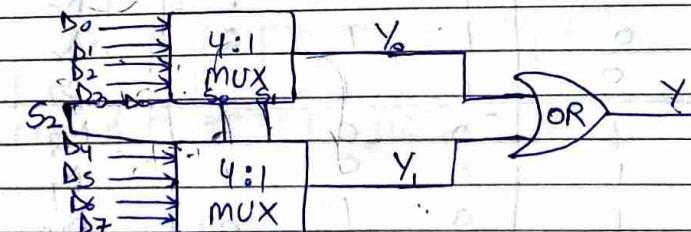
S_0	S_1	S_2
0	0	0
0	0	1
0	1	0
0	1	1
1	0	0
1	0	1
1	1	0
1	1	1

1-8-19

Designing of MUX :-

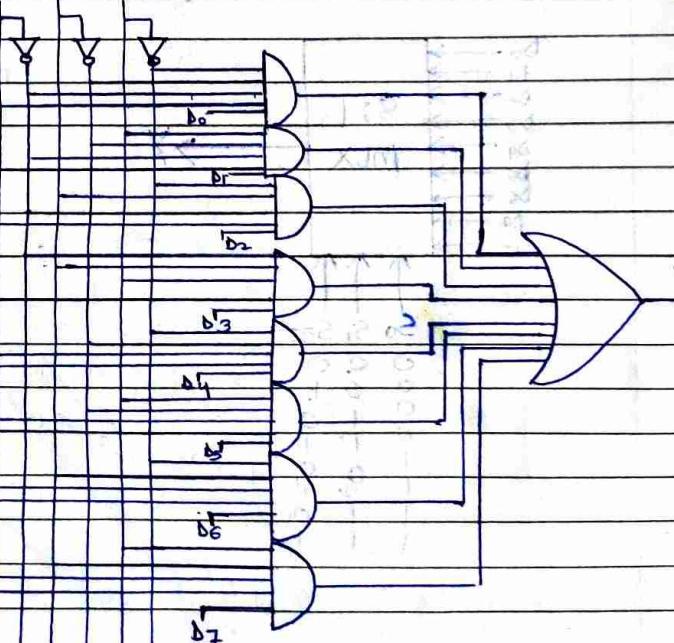
i) Using MUX tree :-

Obtain a 8:1 MUX using 4:1 MUX



$$Y = E(\bar{S}_0 \bar{S}_1 \bar{S}_2 D_0 + \bar{S}_0 \bar{S}_1 S_2 D_1 + \bar{S}_0 S_1 \bar{S}_2 D_2 + \bar{S}_0 S_1 S_2 D_3 \\ + S_0 S_1 S_2 D_4 + S_0 \bar{S}_1 \bar{S}_2 D_5 + S_0 \bar{S}_1 S_2 D_6 + \\ S_0 S_1 S_2 D_7)$$

E S₀ S₁ S₂



(I/P) S₂ | S₀ S₁ | Y

0 0 0 D₀

0 0 1 D₁

0 1 0 D₂

0 1 1 D₃

} MUX 1

} enabled

1 0 0 D₄

1 0 1 D₅

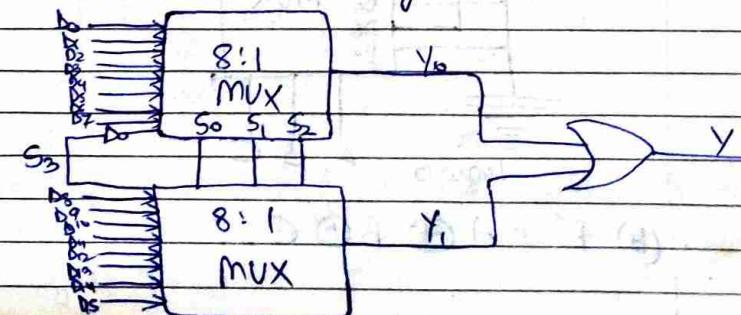
1 1 0 D₆

1 1 1 D₇

} MUX 2

} enabled

Obtain a 16:1 MUX using 8:1 MUX.

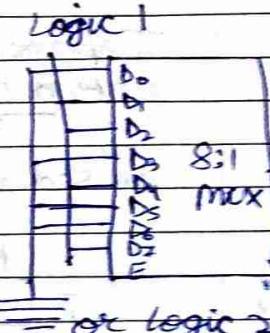


(IP) S_3	S_0	S_1	S_2	Y
0	0	0	0	D_0
0	0	0	1	D_1
0	0	1	0	D_2
0	0	1	1	D_3
0	1	0	0	D_4
0	1	0	1	D_5
0	1	1	0	D_6
0	1	1	1	D_7
1	0	0	0	D_8
1	0	0	1	D_9
1	0	1	0	D_{10}
1	0	1	1	D_{11}
1	1	0	0	D_{12}
1	1	0	1	D_{13}
1	1	1	0	D_{14}
1	1	1	1	D_{15}

MUX 1
enabled

MUX 2
enabled

A	B	C	Y
0	0	0	0
0	0	1	1 $\leftarrow D_1$
0	1	0	1 $\leftarrow D_2$
0	1	1	0
1	0	0	1 $\leftarrow D_4$
1	0	1	0
1	1	0	0
1	1	1	1 $\leftarrow D_7$



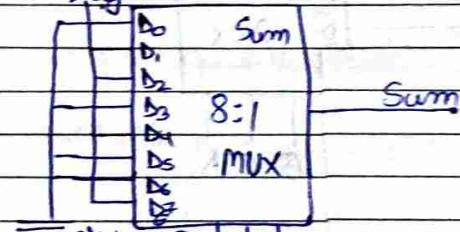
or logic 2

(c) Full Adder

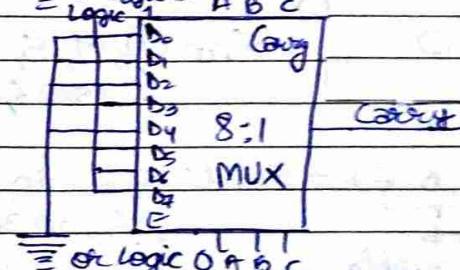
$$\text{Sum} = \Sigma m(1, 2, 4, 7)$$

$$\text{Carry} = \Sigma m(3, 5, 6, 7)$$

Logic 1



Sum

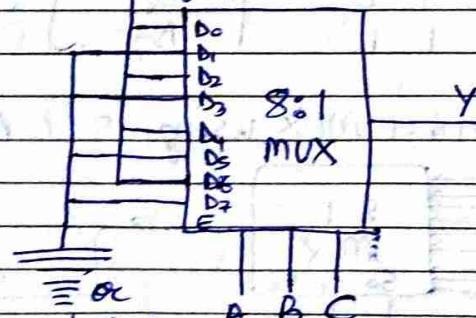


Carry

2) Using Combination circuit / logic :-

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

Logic 1



or logic 0

A B C

$$(b) F = A \oplus B \oplus C$$

⇒ Similarly for half adder we use a 2, 4:1 MUX & then combine their output using OR gate.

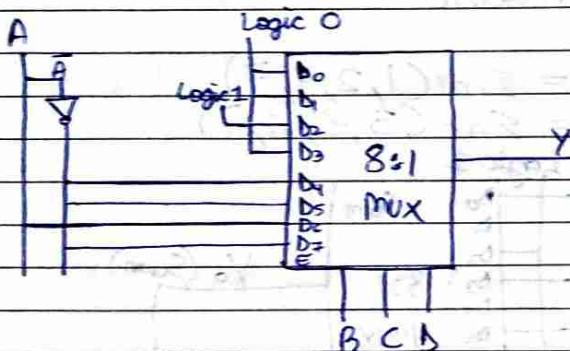
3) Using Design Table:

Implement using 8:1 MUX

$$f(A, B, C, D) = \sum m(2, 4, 5, 7, 10, 14)$$

S.L.
unused variable

	D ₀	D ₁	D ₂	D ₃	D ₄	D ₅	D ₆	D ₇
Ā	0	1	2	3	4	5	6	7
A	8	9	10	11	12	13	14	15
	logic 0	logic 1	logic 1	logic 0	A	A	A	A

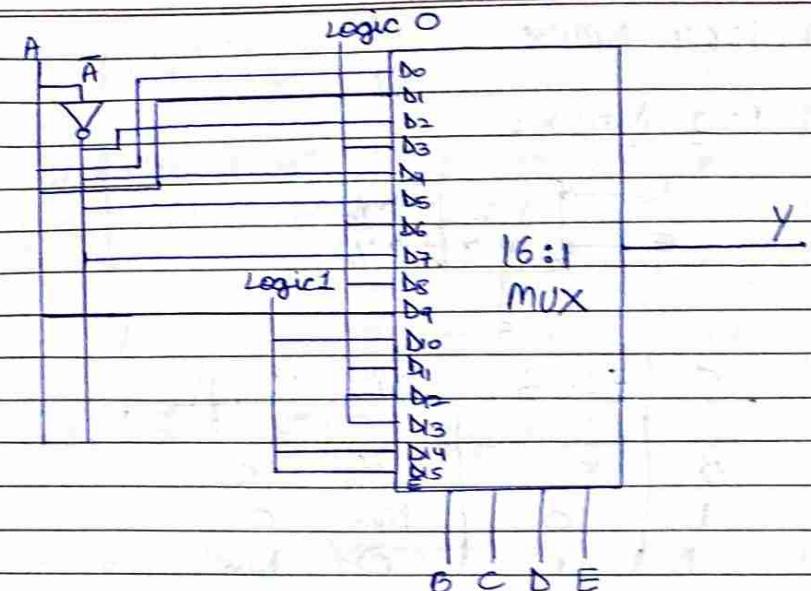


Q1. Obtain a 16:1 MUX.

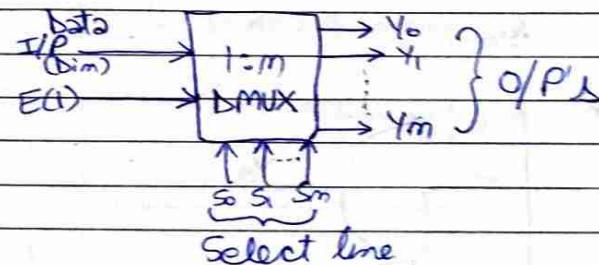
$$f(A, B, C, D, E) = \sum m(2, 4, 5, 7, 10, 14, 15, 16, 17, 25, 26, 30, 31)$$

S.L.
unused variable

	D ₀	D ₁	D ₂	D ₃	D ₄	D ₅	D ₆	D ₇	D ₈	D ₉	D ₁₀	D ₁₁	D ₁₂	D ₁₃	D ₁₄	D ₁₅
Ā	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
A	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
	A	Ā	logic 0	A	logic 0	A	logic 0	A	logic 1	logic 0	logic 0	logic 1	logic 0	logic 1	logic 0	logic 1



7.8.19 IV Demultiplexer (DMUX):



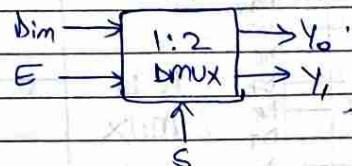
No. of O/P's $\rightarrow m = 2^m$ \leftarrow S.L.

Classification of Demultiplexer (DMUX):

- 1.) 1:2 DMUX
- 2.) 1:4 DMUX
- 3.) 1:8 DMUX
- 4.) 1:16 DMUX
- 5.) 1:32 DMUX

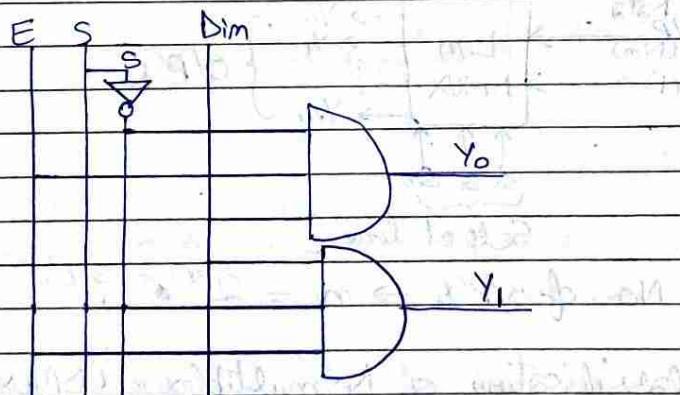
6.) 1:64 DMUX

1.) 1:2 DMUX:

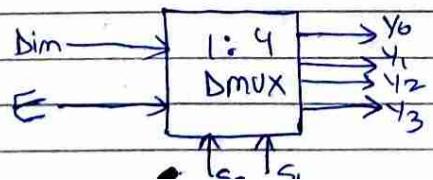


E	S.L.	S	O/P's	Y_0	Y_1
0	x		x	x	x
1	0		Dim	0	0
1	1		0	0	Dim

$$Y = \underbrace{E\bar{S} \text{Dim}}_{Y_0} + \underbrace{ES \text{Dim}}_{Y_1}$$



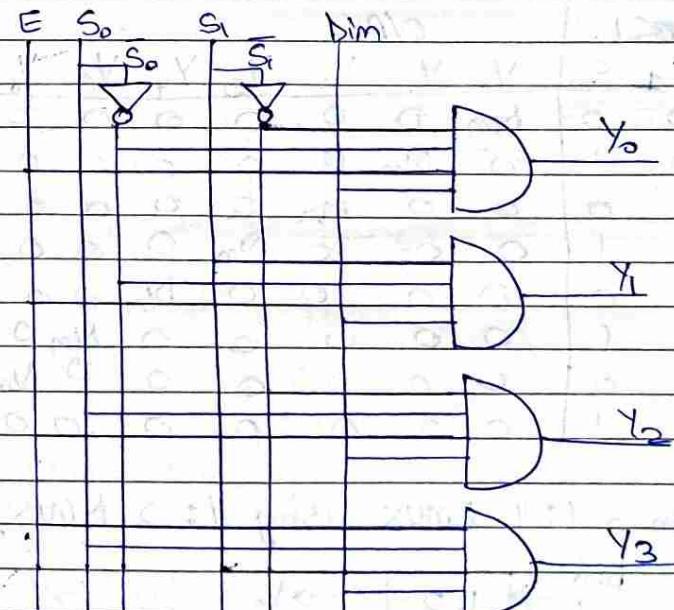
2.) 1:4 DMUX:



E	S.L.	S	O/P's	Y_0	Y_1	Y_2	Y_3
0	x	x	x	x	x	x	x
1	0	0	0	Dim	0	0	0
1	0	1	0	0	Dim	0	0
1	1	0	0	0	0	Dim	00
1	1	1	1	0	0	0	Dim

E	\bar{S}_0	\bar{S}_1	Dim	Y_0	Y_1	Y_2	Y_3
0	1	1	0	0	0	0	0
1	1	0	0	0	0	0	0
1	0	1	0	0	0	0	0
1	0	0	1	0	0	0	0

$$Y = E(\underbrace{\bar{S}_0 \bar{S}_1 \text{Dim}}_{Y_0} + \underbrace{\bar{S}_0 S_1 \text{Dim}}_{Y_1} + \underbrace{S_0 \bar{S}_1 \text{Dim}}_{Y_2} + \underbrace{S_0 S_1 \text{Dim}}_{Y_3})$$



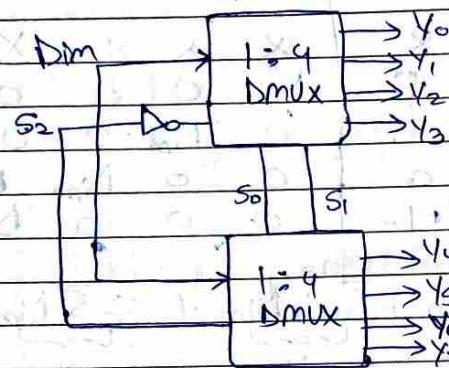
Designing of De-Multiplexer:

(I) Using DMUX Tree:

(II) Using Combinational logic/circuit:

Q1. Design a 1:8 DEMUX using 1:4 DEMUX.

Ans1.

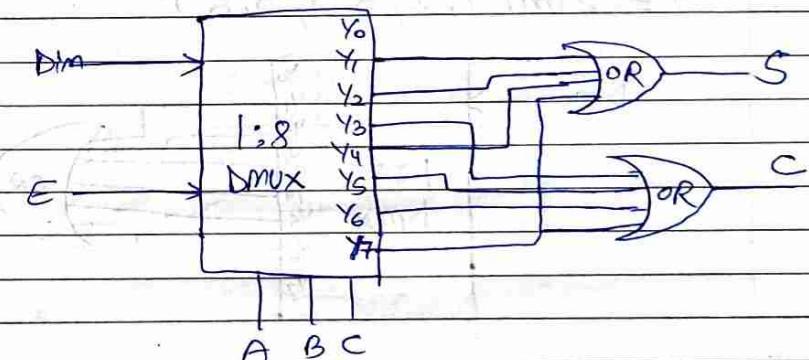


S.L.		O/P's			
S ₁	S ₀	Y ₀	Y ₁	Y ₂	Y ₃
0	0	Dim	0	0	0
0	1	0	Dim	0	0
1	0	0	0	Dim	0
1	1	0	0	0	Dim

Q3. Design Full Adder using suitable De-multiplexer.

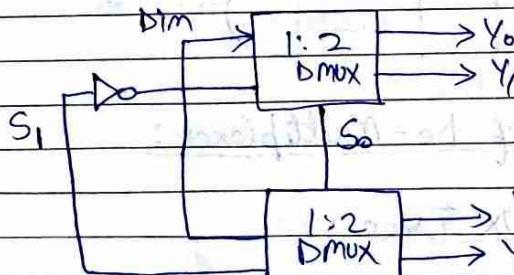
Ans3
 $S = \Sigma m(1, 2, 4, 7)$
 $C = \Sigma m(3, 5, 6, 7)$

S.L.			O/P's							
S ₂	S ₁	S ₀	Y ₀	Y ₁	Y ₂	Y ₃	Y ₄	Y ₅	Y ₆	Y ₇
0	0	0	Dim	0	0	0	0	0	0	0
0	0	1	0	Dim	0	0	0	0	0	0
0	1	0	0	0	Dim	0	0	0	0	0
0	1	1	0	0	0	Dim	0	0	0	0
1	0	0	0	0	0	0	Dim	0	0	0
1	0	1	0	0	0	0	0	Dim	0	0
1	1	0	0	0	0	0	0	0	Dim	0
1	1	1	0	0	0	0	0	0	0	Dim



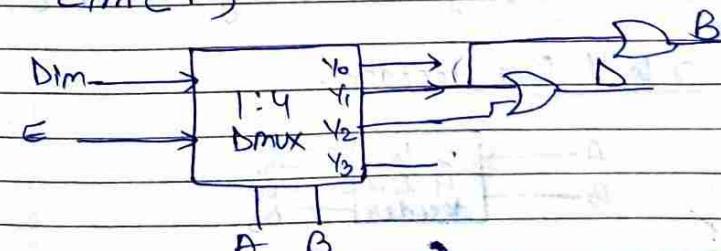
Q2. Design a 1:4 DEMUX using 1:2 DEMUX.

Ans2.



Q4. Before Design a Half Subtractor using suitable DEMUX.

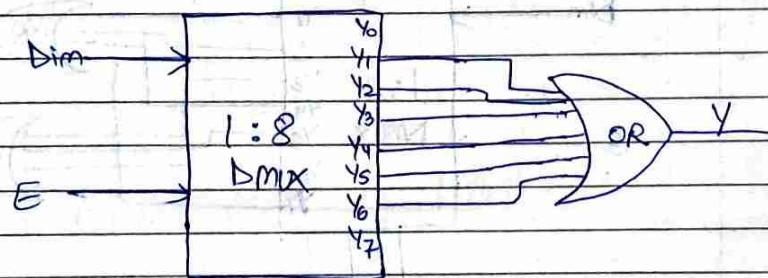
Ans4.
 $D = \Sigma m(1, 2)$
 $B = \Sigma m(1)$



$$Q4. \quad Y = A\bar{B} + B\bar{C} + C\bar{A}$$

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

$$Y = \text{Em}(1, 2, 3, 4, 5, 6)$$



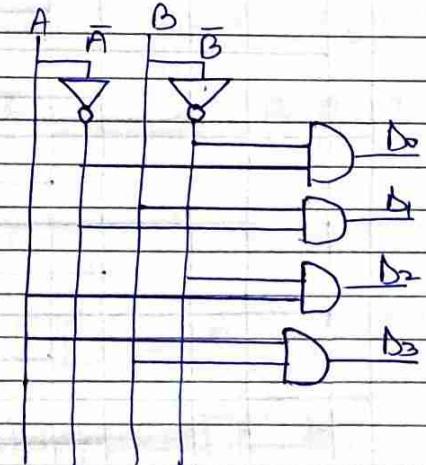
A	B	D_0	D_1	D_2	D_3
0	0	1	0	0	0
0	1	0	1	0	0
1	0	0	0	1	0
1	1	0	0	0	1

$$D_0 = \bar{A}$$

$$D_1 = \overline{A}$$

$$D_2 = A\bar{P}$$

$$D_3 = AB$$

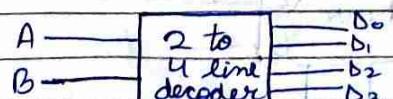


2) 3 to 8 line decoder:

8.8.19 I Decoders:

- 2 to 4 line decoder
 - 3 to 8 line decoder
 - 4 to 10 line (BCD to decimal decoder)
 - BCD to 7 segment display decoder

1-) 2 to 4 line decoder:



```

graph LR
    A((A)) --- I1[ ]
    B((B)) --- I2[ ]
    C((C)) --- I3[ ]
    I1 --- D1[D1]
    I1 --- D2[D2]
    I1 --- D3[D3]
    I1 --- D4[D4]
    I1 --- D5[D5]
    I1 --- D6[D6]
    I1 --- D7[D7]
    I1 --- D8[D8]
    I2 --- D1
    I2 --- D3
    I2 --- D5
    I2 --- D7
    I3 --- D2
    I3 --- D4
    I3 --- D6
    I3 --- D8
  
```

A block diagram showing a 3-to-8 line decoder. On the left, three inputs labeled A, B, and C enter a central block. The output of this block is a 3-to-8 decoder, which produces eight outputs labeled D₁ through D₈. Inputs A and B share a common connection to the first four outputs (D₁, D₃, D₅, D₇). Input C is connected to the last four outputs (D₂, D₄, D₆, D₈).

$$D_0 = \bar{A}\bar{B}\bar{C}$$

$$D_1 = \bar{A}\bar{B}C$$

$$D_2 = \bar{A}B\bar{C}$$

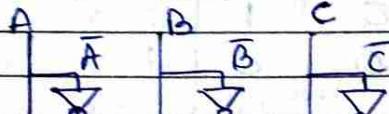
$$D_3 = \bar{A}BC$$

$$D_4 = A\bar{B}\bar{C}$$

$$D_5 = A\bar{B}C$$

$$D_6 = AB\bar{C}$$

$$D_7 = ABC$$



D_0

D_1

D_2

D_3

D_4

D_5

D_6

D_7

4 →	0	1	0	0	D_4
5 →	0	1	0	1	D_5
6 →	0	1	1	0	D_6
7 →	0	1	1	1	D_7
8 →	1	0	0	0	D_8
9 →	1	1	0	0	D_9

$$D_0 = \bar{B}_3 \bar{B}_2 \bar{B}_1 \bar{B}_0$$

$$D_1 = \bar{B}_3 \bar{B}_2 \bar{B}_1 B_0$$

$$D_2 = \bar{B}_3 \bar{B}_2 B_1 \bar{B}_0$$

$$D_3 = \bar{B}_3 \bar{B}_2 B_1 B_0$$

$$D_4 = \bar{B}_3 B_2 \bar{B}_1 \bar{B}_0$$

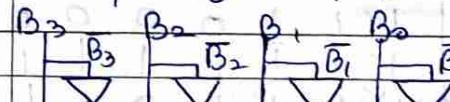
$$D_5 = \bar{B}_3 B_2 \bar{B}_1 B_0$$

$$D_6 = \bar{B}_3 B_2 B_1 \bar{B}_0$$

$$D_7 = \bar{B}_3 B_2 B_1 B_0$$

$$D_8 = B_3 \bar{B}_2 \bar{B}_1 \bar{B}_0$$

$$D_9 = B_3 \bar{B}_2 \bar{B}_1 B_0$$



D_0

D_1

D_2

D_3

D_4

D_5

D_6

D_7

D_8

D_9

3) BCD to Decimal decoder:

↳ Primary coded decimal code (0-9)

B_3	B_2	B_1	B_0	O/P's
0 → 0	0	0	0	D_0
1 → 0	0	0	1	D_1
2 → 0	0	1	0	D_2
3 → 0	0	1	1	D_3

4.) BCD to 7 segment display decoder:

\hookrightarrow Binary coded decimal codes (0-9)

~~$B_3 \ B_2 \ B_1 \ B_0$~~ | $a \ b \ c \ d \ e \ f \ g$

0 \rightarrow	0 0 0 0	1 1 1 1 1 0
1 \rightarrow	0 0 0 1	0 1 1 0 0 0
2 \rightarrow	0 0 1 0	1 0 1 1 0 1
3 \rightarrow	0 0 1 1	1 1 1 1 0 0
4 \rightarrow	0 1 0 0	0 1 1 0 0 1
5 \rightarrow	0 1 0 1	0 1 1 0 1 1
6 \rightarrow	0 1 1 0	0 1 1 1 1 1
7 \rightarrow	0 1 1 1	1 1 1 0 0 0
8 \rightarrow	1 0 0 0	1 1 1 1 1 1
9 \rightarrow	1 0 0 1	1 1 1 1 0 1

$$\begin{array}{l}
 \underline{\underline{a}} \\
 f1 \underline{\underline{g}} \underline{\underline{16}} \quad 0 \rightarrow f1 \underline{\underline{16}} \quad 1 \rightarrow \underline{\underline{16}} \\
 \underline{\underline{el}} \underline{\underline{1c}} \quad \underline{\underline{el}} \underline{\underline{1c}} \\
 \underline{\underline{a}} \quad \underline{\underline{a}} \\
 2 \rightarrow \underline{\underline{16}} \quad 3 \rightarrow \underline{\underline{16}} \quad 4 \rightarrow f1 \underline{\underline{g}} \underline{\underline{16}} \\
 \underline{\underline{el}} \underline{\underline{g}} \quad \underline{\underline{d}}
 \end{array}$$

K-map (make K-map for any two alphabets)

$B_3 \ B_2$	00	01	11	10
00	1	0	1	1
01	1	1	0	0
11	1	0	1	1
10	1	1	0	0

(for a)

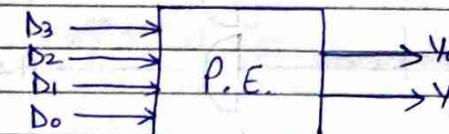
$$\begin{aligned}
 2 = & B_1 \bar{B}_3 + \bar{B}_3 B_2 \bar{B}_0 \\
 & + B_2 \bar{B}_3 \bar{B}_1 + \bar{B}_2 B_1 \bar{B}_0
 \end{aligned}$$

$B_3 \ B_2$	00	01	11	10
00	0	1	1	0
01	1	0	0	1
11	1	1	0	1
10	1	0	1	0

(for b)

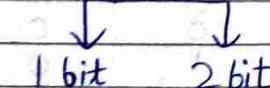
$$\begin{aligned}
 b = & \bar{B}_3 \bar{B}_2 + \bar{B}_3 \bar{B}_1 \bar{B}_0 + \\
 & \bar{B}_3 B_1 B_0 + B_3 \bar{B}_2 \bar{B}_1
 \end{aligned}$$

20-8-19 # Priority Encoder:



D_3	D_2	D_1	D_0	Y_0	Y_1
X	X	X	X	X	X
0	0	0	1	0	0
0	0	1	X	0	1
0	1	X	X	1	0
1	X	X	X	1	1

Comparators:



(a) 1-bit Magnitude Comparator:

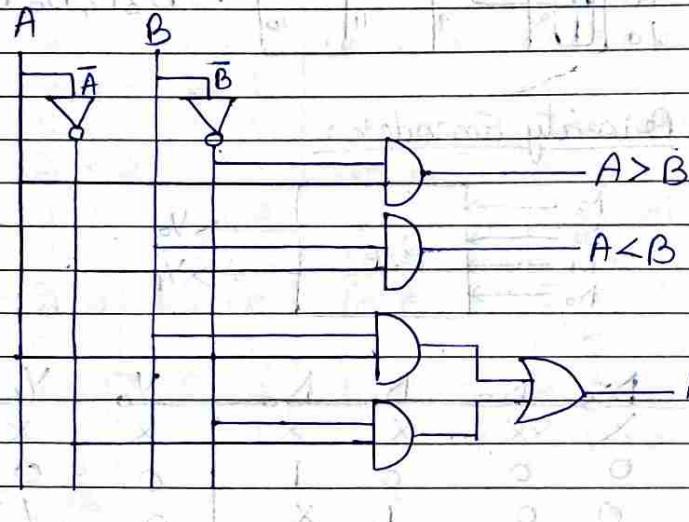
A	B	$A > B$	$A < B$	$A = B$
0	0	0	0	1
0	1	0	1	0
1	0	1	0	0
1	1	0	0	1

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

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

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

$$= A \odot B \quad (\text{x-NOR})$$



A, A _o	B, B _o	00	01	11	10
00					
01		1			
11		1	1		
10		1	1	1	1

For (A > B)

$$[A > B] = A_1\bar{B}_1 + A_o\bar{B}_o\bar{B}_1 + A_oA_1\bar{B}_o$$

A, A _o	B, B _o	00	01	11	10
00					
01		1			
11		1	1	1	1
10		1	1	1	1

for (A < B)

$$[A < B] = \bar{A}_1B_1 + \bar{A}_o\bar{A}_1B_o + \bar{A}_oB_oB_1$$

(b) 2-bit Magnitude Comparator:

B ₁	B ₀	A, A _o	A > B	A < B	A = B
0	0	0 0	0	0 1	1
0	0	0 1	1	0 0	0
0	0	1 0	1	0 0	0
0	0	1 1	1	0 0	0
0	1	0 0	0	1 0	0
0	1	0 1	0	0 0	1
0	1	1 0	1	0 0	0
0	1	1 1	1	0 0	0
1	0	0 0	0	1 0	0
1	0	0 1	0	0 0	1
1	0	1 0	0	0 0	1
1	0	1 1	0	0 0	1
1	1	0 0	0	1 1	0
1	1	0 1	0	0 0	1
1	1	1 0	0	0 0	1
1	1	1 1	0	0 0	1

A, A _o	B, B _o	00	01	11	10
00		1			
01		1	1		
11		1	1	1	
10		1	1	1	1

for (A = B)

$$\begin{aligned}
 [A = B] &= \bar{A}_1\bar{A}_o\bar{B}_o\bar{B}_1 + A_o\bar{A}_1B_o\bar{B}_1 + A_oA_1B_oB_1 + \bar{A}_oA_1\bar{B}_oB_1 \\
 &= \bar{A}_1\bar{B}_1 [\bar{A}_o\bar{B}_o + A_oB_o] + A_1B_1 [\bar{A}_oB_o + \bar{A}_o\bar{B}_o] \\
 &= [\bar{A}_1\bar{B}_1 + A_1B_1][\bar{A}_o\bar{B}_o + A_oB_o]
 \end{aligned}$$

$$\begin{array}{c}
 [A = B] = (A_1 \odot B_1)(A_o \odot B_o) \\
 \downarrow \quad \downarrow \\
 \text{x-NOR} \quad \text{x-NOR}
 \end{array}$$

21/8/19

Codes

Binary	BCD	Excess-3	Gray Code
4-bit (0000-1111)	Binary (add 1 to each digit)	(0-9) BCD+3	(Mirrored reflecting codes)
B ₃ B ₂ B ₁ B ₀	0 → 0 0 0 0	Eg: 0 → 0 0 0 0	0 → 0 0 0 0
0 → 0 0 0 0	1 → 0 0 0 1	1 → 0 0 0 1 (B ₃ → 1 0 0 1 (0-15))	1 → 0 0 0 0
1 → 0 0 0 1	2 → 0 0 1 0	2 → 0 0 1 0 (0011 → BCD → 3 → 0011)	2 → 0 0 0 1
2 → 0 0 1 0	3 → 0 0 1 1	3 → 0 0 1 1 (Excess-3 → 0100 → G ₃ → 0010)	3 → 0 0 1 0
3 → 0 0 1 1	4 → 1 0 1 0	4 → 1 0 1 0 (B ₃ → 9 → 1001 → G ₃ → 0010 (for gray code first))	4 → 0 1 0 1
4 → 0 1 0 0	5 → 0 1 0 1	5 → 0 1 0 1 (Excess-3 → 1100 → write MSB as 0)	5 → 0 1 0 1
5 → 0 1 0 1	6 → 0 1 1 0	6 → 0 1 1 0 (Excess-3 → 1100 → it is 2 times of 5)	6 → 0 1 1 0
6 → 0 1 1 0	7 → 0 1 1 1	7 → 0 1 1 1 (Eg: (78) → 0111 → it is 2 times of 6)	7 → 0 1 1 1
7 → 0 1 1 1	8 → 1 0 0 0	8 → 1 0 0 0 (Excess-3 → 0111 → do XOR of 7 and 8)	8 → 1 0 0 0
8 → 1 0 0 0	9 → 1 0 0 1	9 → 1 0 0 1 (Excess-3 → 0111 → 0 → 1010 → 9 → 1001 → each 2 bits)	9 → 1 0 0 1
9 → 1 0 0 1	Eg: (78) → BCD	Eg: (78) → BCD → 0 → 1010 → 9 → 1001 → 1010 → 8 → 1000 → 1010 → 7 → 0111 → 1010 → 6 → 0110 → 5 → 0101 → 4 → 0100 → 3 → 0101 → 2 → 0100 → 1 → 0101 → 0 → 0100 → 1 → 0101 → 2 → 0100 → 3 → 0101 → 4 → 0100 → 5 → 0101 → 6 → 0100 → 7 → 0101 → 8 → 0100 → 9 → 0101 → 10 → 0100 → 11 → 0101 → 12 → 0100 → 13 → 0101 → 14 → 0110 → 15 → 0111	Eg: (78) → BCD → 0 → 1010 → 9 → 1001 → 1010 → 8 → 1000 → 1010 → 7 → 0111 → 1010 → 6 → 0110 → 5 → 0101 → 4 → 0100 → 3 → 0101 → 2 → 0100 → 1 → 0101 → 0 → 0100 → 1 → 0101 → 2 → 0100 → 3 → 0101 → 4 → 0100 → 5 → 0101 → 6 → 0100 → 7 → 0101 → 8 → 0100 → 9 → 0101 → 10 → 0100 → 11 → 0101 → 12 → 0100 → 13 → 0101 → 14 → 0110 → 15 → 0111
10 → 1 0 1 0	BCD(0111 1000)	Excess-3(8) → 1000 → G ₃ → 15 1000	
11 → 1 0 1 1		0011 (Eg: G ₃ → 15 1000 → 0011)	
12 → 1 1 0 0		1011 (B → 15 1000 → 1011)	
13 → 1 1 0 1		(diagonal XOR)	
14 → 1 1 1 0			
15 → 1 1 1 1			

 $BSD \rightarrow 3 \rightarrow 0011$ $G_7 \rightarrow 3 \rightarrow 0010$ $G_3 \quad G_2 \quad G_1 \quad G_0$

$0 \rightarrow 0 \quad 0 \quad 0 \quad 0$
 $1 \rightarrow 0 \quad 0 \quad 0 \quad -1$
 $2 \rightarrow 0 \quad 0 \quad 1 \quad 1$
 $3 \rightarrow 0 \quad 0 \quad 1 \quad 0$

$\} (1\text{-bit})$
 $\Rightarrow 0\ 1 \text{ then we put}$
 a mirror 8-bit
 $\text{will be the mirror}$
 image of 0 i.e.
 10^2 to cm.

Conversions:

1) $(1011010)_B = (?)_A$

2) $(01101110)_B = (?)_A$

3) $(792)_{10} = (?)_{BCD}$

4) $(83)_{10} = (?)_{Ex-3}$

Ans 1) $B \rightarrow 1011010$

$G_7 \rightarrow 1110111$

Ans 2) $G_7 \rightarrow 01101110$

$B \rightarrow 010000010101$

01001011 (Ans)

Ans 3) $(792)_{10}$

$BCD \rightarrow 7 = 0111$

$BCD \rightarrow 9 = 1001$

$BCD \rightarrow 2 = 0010$

$\Rightarrow (011110010010)_{BCD} \text{ (Ans)}$

Q1. Why gray codes are called mirror reflecting codes?

$BSD \rightarrow 0 \quad 0000$

$G_7 \rightarrow 0 \quad 0000$

$BSD \rightarrow 1 \quad 0001$

$G_7 \rightarrow 1 \quad 0001$

$BSD \rightarrow 2 \quad 0010$

$G_7 \rightarrow 2 \quad 0011$

Ans 4) $(83)_{10} = (?)_9 \div ?$

~~BCD $\rightarrow 8$~~ 1000

~~BCD $\rightarrow 3$~~ 0011

1000

0011

0011

0011

1011

0110

• $(10110110)_9$ (Ans)

Gray Codes to Binary Codes Table:

Number	Gray Codes	Binary Codes
	G ₃ G ₂ G ₁ G ₀	B ₃ B ₂ B ₁ B ₀
0	0 1 0 0 0	0 0 0 0
1	0 1 0 0 1	0 0 0 1
2	0 0 1 1	0 0 1 0
3	0 0 1 0	0 0 1 1
4	0 1 1 0	0 1 0 0
5	0 1 1 1	0 1 0 1
6	0 1 0 1	0 1 1 0
7	0 1 0 0	0 1 1 1
8	1 1 0 0	1 0 0 0
9	1 1 0 1	1 0 0 1
10	1 1 1 1	1 0 1 0
11	1 1 1 0	1 0 1 1
12	1 0 1 0	1 1 0 0
13	1 0 1 1	1 1 0 1
14	1 0 0 1	1 1 1 0
15	1 0 0 0	1 1 1 1

Binary to Gray Codes Table:

Number	Binary Code	Gray Code
	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
1	0 0 0 1	0 0 0 1
2	0 0 1 0	0 0 1 1
3	0 0 1 1	0 0 1 0
4	0 1 0 0	0 1 1 0
5	0 1 0 1	0 1 1 1
6	0 1 1 0	0 1 0 1
7	0 1 1 1	0 1 0 0
8	1 0 0 0	1 1 1 0
9	1 0 0 1	1 1 1 0 1
10	1 0 1 0	1 1 1 1
11	1 0 1 1	1 1 1 0
12	1 1 0 0	1 0 1 0
13	1 1 0 1	1 0 1 1
14	1 1 1 0	1 0 0 1
15	1 1 1 1	1 0 0 0

K-Map for Binary to Gray Codes:

→ For G₀:

		B ₃ B ₂	
		00 01 11 10	
		00 01 11 10	
00	00	0	1
01	01	1	1
11	11	1	1
10	10	1	1

$$G_0 = \overline{B}_3 B_0 + B_1 \overline{B}_0$$

$$G_0 = B_1 \oplus B_0$$

22/8/19

→ For G_1 :

$B_3 B_2$	00	01	11	10
00	1	1		
01	1	1		
11			1	1
10			1	1

$$G_1 = B_2 \bar{B}_1 + \bar{B}_2 B_1$$

$$(G_1 = B_2 \oplus B_1)$$

Similarly, we draw for G_2 & G_3

K-Map for Gray to Binary Codes:

→ For B_0 :

$G_3 G_2$	00	01	11	10
00	1			
01		1		
11			1	1
10			1	1

$$\Rightarrow B_0 = G_3 \bar{G}_1 \bar{G}_2 \bar{G}_3 + \bar{G}_3 G_1 \bar{G}_2 \bar{G}_3 + \bar{G}_3 \bar{G}_1 G_2 \bar{G}_3 + \\ G_3 \bar{G}_1 \bar{G}_2 G_3 + \bar{G}_3 G_1 G_2 \bar{G}_3 + \bar{G}_3 \bar{G}_1 G_2 G_3 + G_3 \bar{G}_1 \bar{G}_2 G_3 + \\ G_3 \bar{G}_1 G_2 G_3$$

$$\Rightarrow B_0 = \bar{G}_2 \bar{G}_3 (\bar{G}_3 G_1 + G_3 \bar{G}_1) + \bar{G}_3 G_2 (\bar{G}_3 \bar{G}_1 + G_3 G_1) + \\ G_{12} G_3 (G_3 G_1 + \bar{G}_3 \bar{G}_1) + \bar{G}_2 G_3 (\bar{G}_3 \bar{G}_1 + G_3 G_1)$$

$$\Rightarrow B_0 = \bar{G}_2 \bar{G}_3 (G_3 \oplus G_1) + \bar{G}_3 G_2 (G_3 \odot G_1) + \\ G_{12} G_3 (G_3 \oplus G_1) + \bar{G}_2 G_3 (G_3 \odot G_1)$$

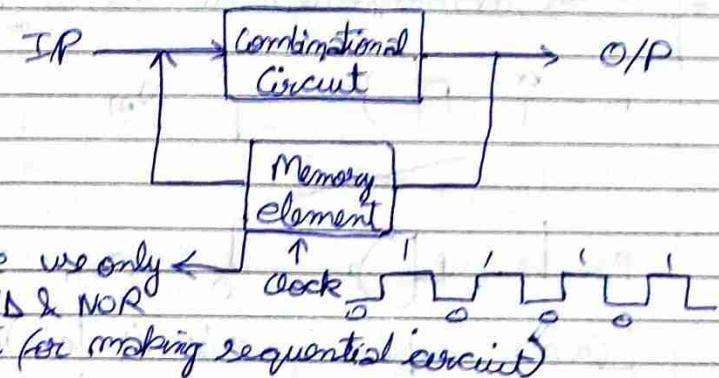
$$\Rightarrow B_0 = (G_3 \oplus G_1) (G_{12} \odot G_3) + (G_3 \odot G_1) (G_2 \oplus G_3)$$

$$\Rightarrow B_0 = G_3 \oplus G_{12} \oplus G_1 \oplus G_2$$

Similarly, we draw K-maps for B_1, B_2 & B_3

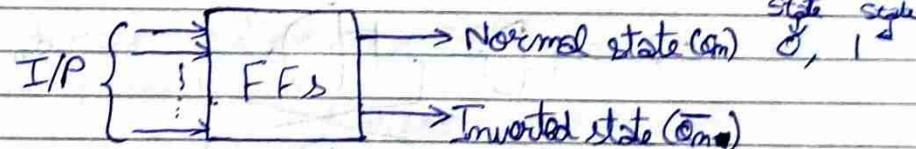
Unit - II Sequential Circuit

* Sequential Circuit:



Classification of Sequential Circuits:

1) Flip-Flops: (Stores only 1-bit)



(Used to check the value of normal state)

Present state (Q_n) [It is the last stored value in FF]

Next state (Q_{n+1}) [It is the current value in the FF]

* Flip-Flops are of 4 types:

(a) S-R FF

Set Reset

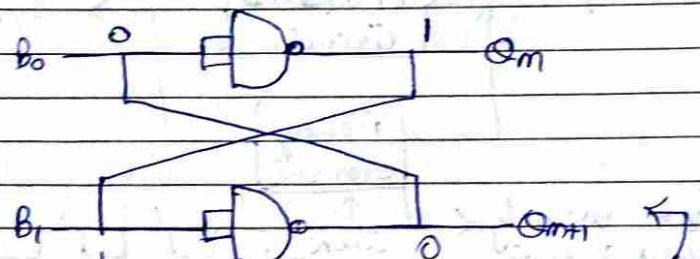
(b) J-K FF

J F F

Delay

(A) T FF
Toggle

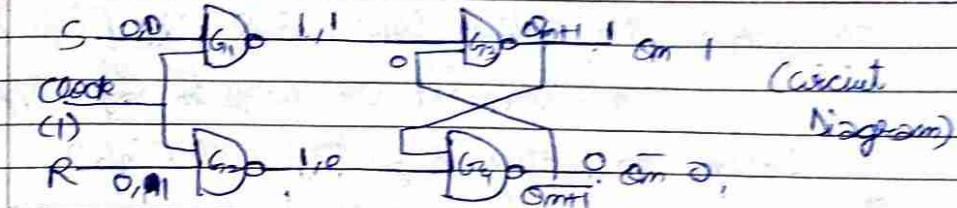
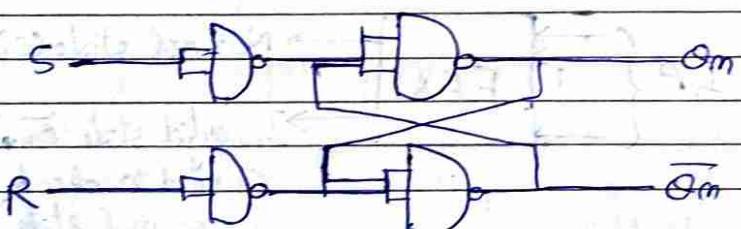
* 1-bit memory element :-



[Cross coupled inverter (Latch):]

Q1. Difference b/w Latch & Flip-Flops.

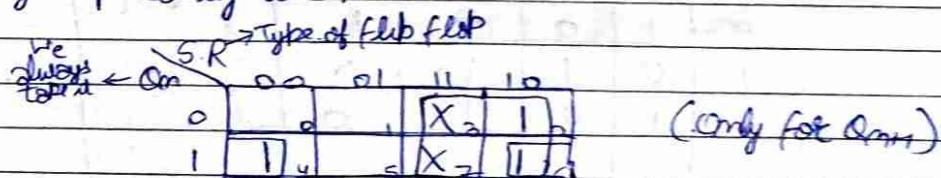
* Modified Cross Coupled Inverter / S-R Latch:



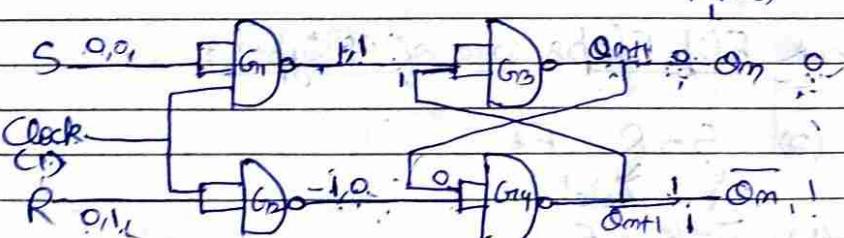
Truth Table →

S	R	Q_m	\bar{Q}_m	Q_{m+1}	\bar{Q}_{m+1}	Comments
0	0	0	1	0	1	No change
0	0	1	0	1	0	No change
0	1	0	1	0	1	{ Reset
0	1	1	0	0	1	{ Set
1	0	0	1	1	0	{ Invalid case
1	0	1	0	1	0	
1	1	0	1	x	x	
1	1	1	0	x	x	

* Whenever we design circuit using NAND gate we give priority to 0.



~~25/8/19~~ * S-R Flip-Flop: (Draw only 1 circuit diagram in paper)

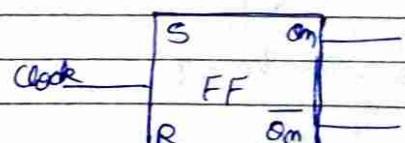


(S-R Latch & S-R Flip-Flop have only the difference of a clock.)

$$Q_{m+1} = \overline{S} \cdot \overline{R} \cdot Q_m$$

Characteristic eqn

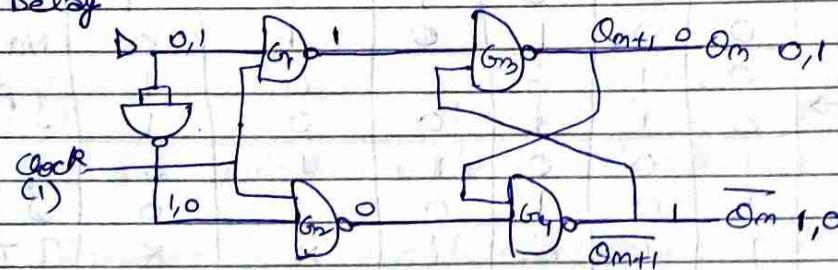
Logic Symbol:



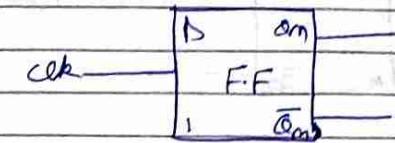
Q1. Design the active-low S-R flip flop?
 ↓
 then we use NAND gate always for all flip flops

Q2. Design the active-high S-R flip flop?
 ↓
 then we use NOR gate always for all flip flops

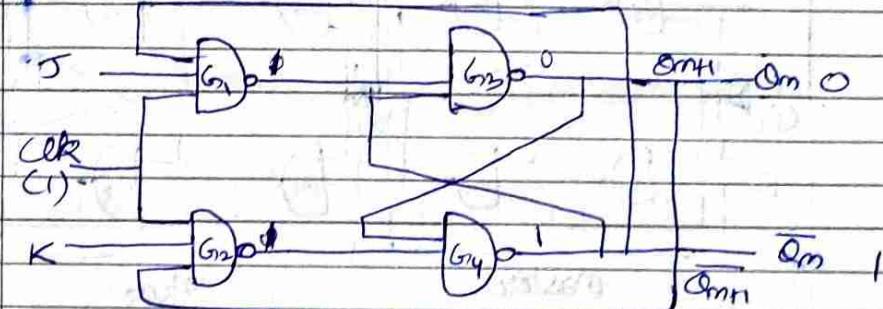
* D-Flip Flop:
 ↓ Delay



Logic Symbol:

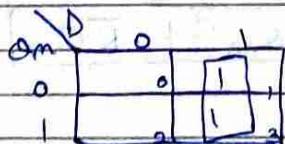


J-K Flip-Flop:



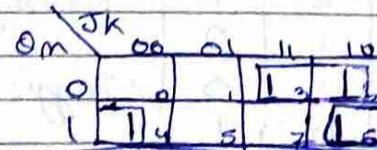
Truth Table:

D	Qm	$\bar{Q}m$	Q_{m+1}	\bar{Q}_{m+1}	Comment
0	0	1	0	1	{ Reset
0	1	0	0	1	
1	0	1	1	0	{ Set
1	1	0	1	0	



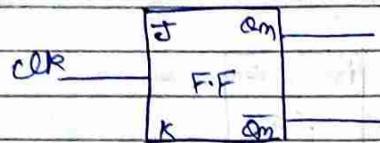
$Q_{m+1} \Rightarrow D$
 ↓
 characteristic eqⁿ

J	K	Qm	$\bar{Q}m$	Q_{m+1}	\bar{Q}_{m+1}	Comments
0	0	0	1	0	1	{ No change
0	0	1	0	1	0	
0	1	0	1	0	1	{ Reset
0	1	1	0	0	1	
1	0	0	1	1	0	{ Set
1	0	1	0	1	0	
1	1	0	1	1	0	
1	1	1	0	0	1	{ Toggle (inverted)



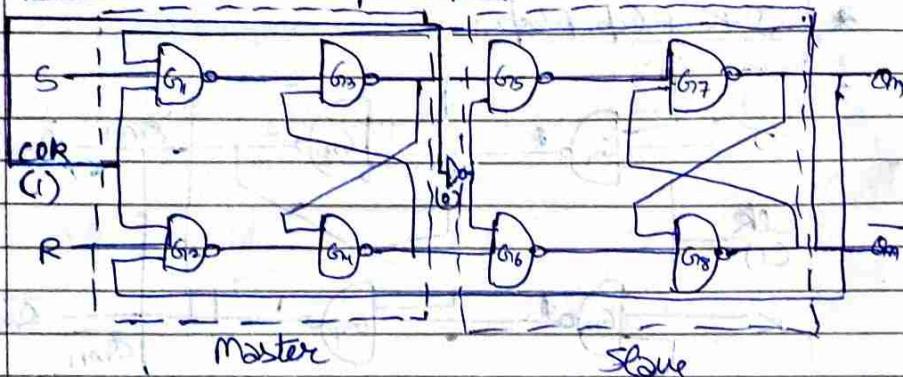
(K-map)

$Q_{m+1} = J Q_m + \bar{K} Q_m \Rightarrow$ characteristic eqⁿ



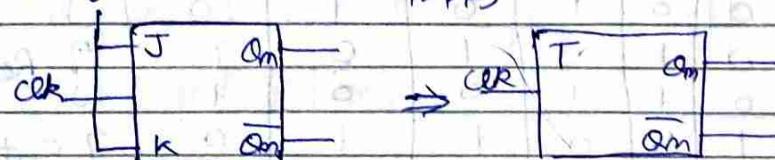
(Logic Symbol)

* Master Slave Flip Flop:

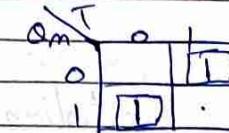


* It provides delay so that the present state will be held for some nano seconds & we can find the next state using it.

* Toggle Flip Flop: (Combine both inputs of J-K FF to draw T-FF)



T	Q_m	\bar{Q}_m	Q_{m+1}	\bar{Q}_{m+1}	Comments
0	0	1	0	1	No change
0	1	0	1	0	
1	0	1	1	0	Toggle
1	1	0	0	1	



$$Q_{m+1} = T \bar{Q}_m + \bar{T} Q_m$$

$$Q_{m+1} = T \oplus Q_m$$

29.8.19

Conversion of one type FF to another type FF:

1) S-R Flip Flop:

Truth Table:

S	R	Q_m	\bar{Q}_m
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	X
1	1	1	X

Excitation Table:

Q_m	Q_{m+1}	S	R
0	0	0	0
0	0	0	X
0	1	1	0
1	0	0	1
1	0	X	0
1	1	X	0

The input which is changed then represent it with a X & rest as it is.

2) D-Flip Flop:

Truth Table:

D	Q_m	Q_{m+1}
0	0	0
0	1	0
1	0	1
1	1	1

Excitation Table:

Q_m	Q_{m+1}	D
0	0	0
0	1	1
1	0	1
1	1	0

3) J-K Flip Flop:

Truth Table:

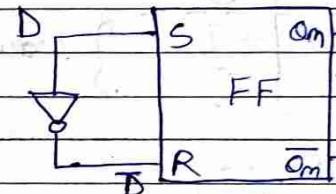
J	K	Qm	Qm+1	Qm	Qm+1	J	K
0	0	0	0	0	0	0	X
0	0	1	1	0	1	1	X
0	1	0	0	1	0	X	1
0	1	1	0	1	1	X	0
1	0	0	1	1	1	1	1
1	0	1	1	1	1	1	1
1	1	0	1	1	1	1	0

Excitation Table:

D	0	1	0	1
0	0	0	1	X
1	1	1	X	1
1	1	1	1	0

$$S = D$$

$$R = \bar{D}$$



4) T-Flip Flop:-

Truth Table:

T	Qm	Qm+1	Qm	Qm+1	T
0	0	0	0	0	0
0	1	1	0	0	1
1	0	1	1	0	1
1	1	0	1	1	0

Excitation Table:

D	Qm	Qm+1	DT
0	0	0	0
0	1	0	1
1	0	1	01
1	1	1	0

D	Qm	Qm+1
0	0	0
0	1	1
1	1	0
1	0	1

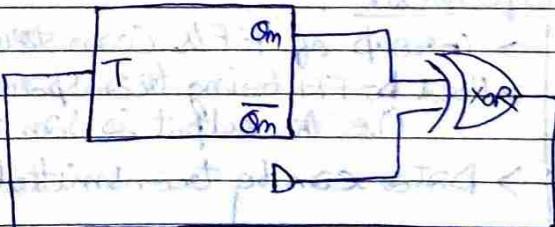
$$T = D \oplus Q_m$$

$$T = D \oplus Q_m$$

Convert S-R FF to D-FF

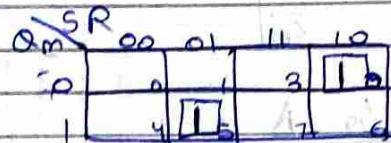
~~Truth Table~~ ~~Excitation Table~~

D	Qm	Qm+1	S	R
0	0	0	0	X
0	1	0	0	1
1	0	1	1	0
1	1	1	X	0

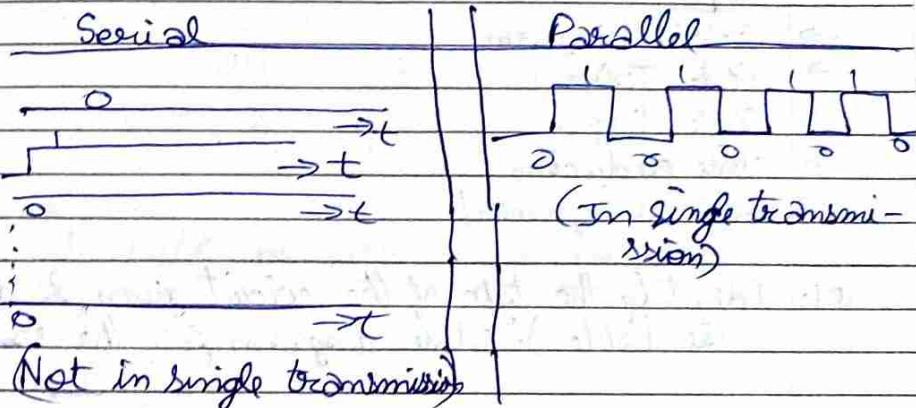
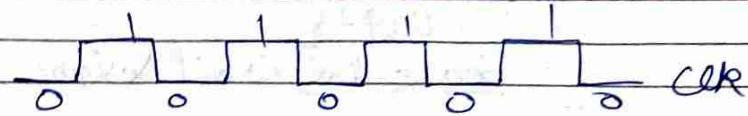
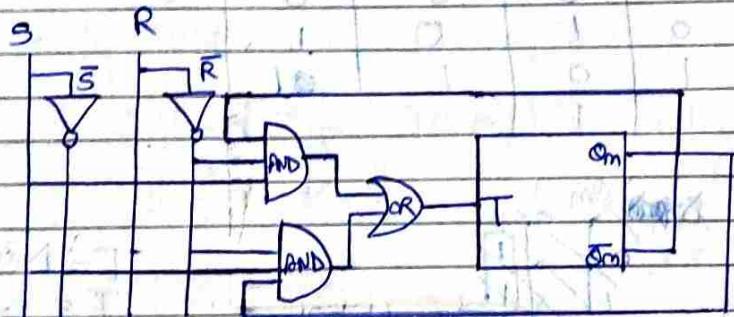


Convert T-FF to SR FF

S	R	Q_m	Q_{m+1}	T
0	0	0	0	0
0	0	1	1	0
0	1	0	0	0
0	1	1	0	1
1	0	0	1	1
1	0	1	1	0
1	1	0	X	?
1	1	1	X	?



$$T = \bar{Q}_m S R + \bar{S} R Q_m$$



Classification of Registers:

- 1.) SISO (Serial I/P Serial O/P)
- 2.) SIPO (Serial I/P Parallel O/P)
- 3.) PIPO (Parallel I/P Serial O/P)
- 4.) PTPO (Parallel I/P Parallel O/P)
- 5.) Universal S.R (Shift Register) [It contains all the above registers]

30.8.19

Registers:

- Group of FF's can store multiple bits
- Used D-FF being transparent (01110...)
(i.e. the output is same as the input)
- Data can be transmitted:-

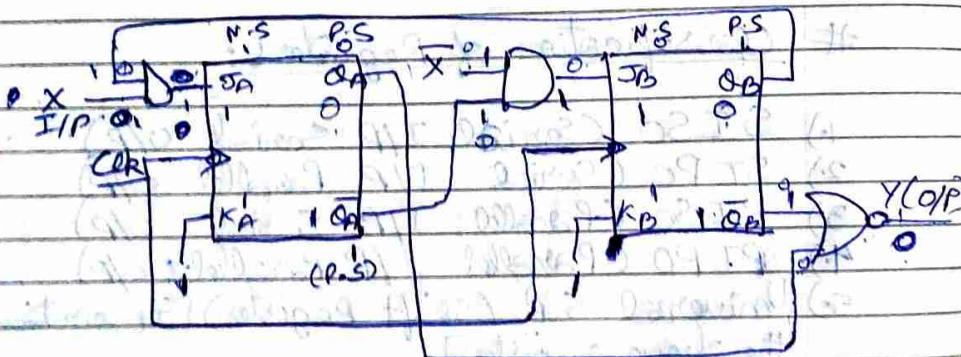
↓ ↓
Serial form at Parallel form

Unit - 3
Sequential Circuit Design:
Steps:

- Circuit Diagram
- State Table
- State diagram
- State reduction
- State assignment

- Q1. Identify the type of the circuit given & draw its state table & state diagram for the same

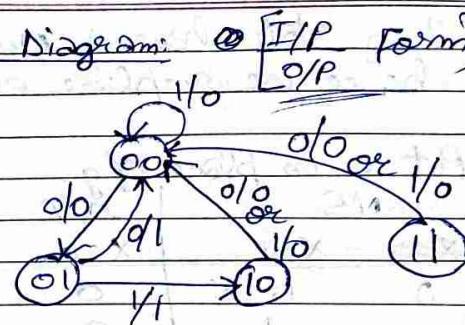
Ans.



Note:- If all flip-flops are connected to clock pulse then it is Mealy circuit otherwise Moore.

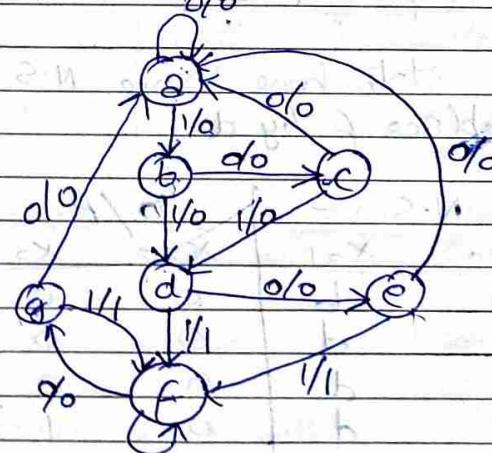
State Table:

P.S	N.S	O/P (Y)			
Q _A	Q _B	X=0	X=1	X=0	X=1
0	0	01	00	0	0
0	1	00	10	1	1
1	0	00	00	0	0
1	1	00	00	0	0

State Diagram: (I/P Form)


- Q2. For the given State diagram plot the reduced state table or state reduction & state assignment.

Ans.



P.S	X=0	N.S	X=1	O/P
a	a	b	0	0
b	c	d	0	0
c	a	d	0	0
d	e	f	0	1
e	a	f	0	1
f	g	f	0	1
g	a	f	0	1

Since e & g state have same next state & output hence we replace either of the one.

Here we put e in place of g.

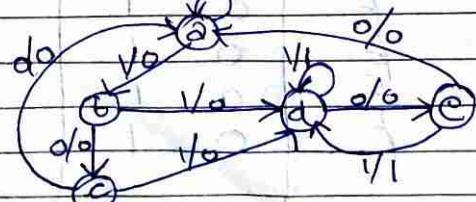
P.S	NS	O/P
	$x=0$ $x=1$	$x=0$ $x=1$
a	a b	0 0
b	c d	0 0
c	a d	0 0
d	e f	0 1
e	a f	0 1
f	e f	0 1

Again d & f state have same NS & O/P
hence we replace f by d

P.S	N.S	O/P
	$x=0$ $x=1$	$x=0$ $x=1$
a	a b	0 0
b	c d	0 0
c	a d	0 0
d	e d	0 1
e	a d	0 1

Reduced state table

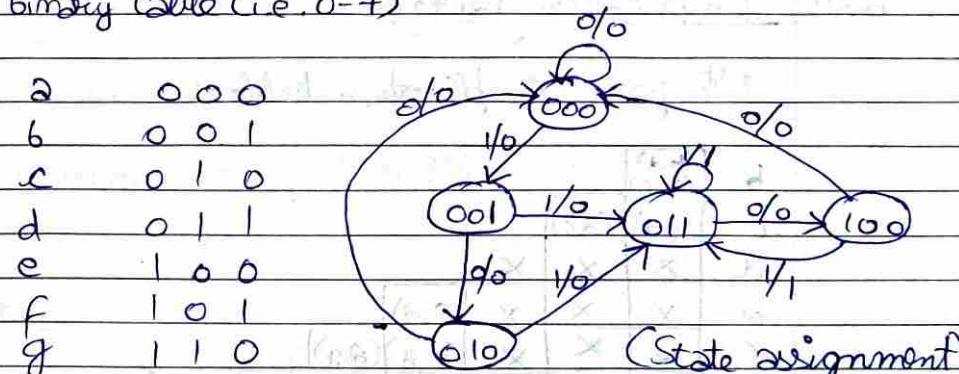
18/10/19 State Diagram:



Initial State

State Assignment: Replacing alphabets with their respective binary bits in State Diagram is known as State assignment.

For eg: for above state diagram as we have alphabets from a to g = 7 \Rightarrow We use 3-bit binary table (i.e. 0-7)



Finite State Machines: (Steps for State Reduction)

Minimization of
completely specified
sequential machines

Maximization of incomplete
specified sequential machines
(N.S or O/P might be
P.S, N.S, I/P, O/P given) missing)

- Q1: For given State Table determine the implications chart & find merger diagram & max. combinable pair for the same. (Completely Specified)

P.S	N.S	O/P
	$x=0$ $x=1$	$x=0$ $x=1$
a	a b	0 0
b	c d	0 0

e	a	d	0	0
d	e	f	0	1
e	a	f	0	1
f	g	F	0	1
g	a	F	0	1

Ans1. Implication Table:

1st pass Implication table:

b	(b,a)					
c	(c,c)					
d	(b,d)	(a,d)				
e	x	x	x	(e,a)		
f	x	x	x	(e,g)	(a,g)	
g	x	x	x	(e,a)	✓	(g,a)

a b c d e f

2nd Pass Implication Table:

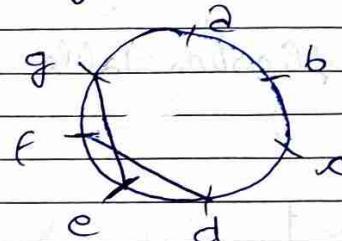
b	x					
c	x	x				
d	x	x	x			
e	x	x	x	x		
f	x	x	x	✓	x	
g	x	x	x	x	✓	x

a b c d e f

Compatible pair: (Ticked one in 2nd pass T.T)

→ (e,g) & (d,f)

Merger diagram:



Rhombus > triangle > straight line

Maximum Compatible Pairs:

e=g & d=f

Reduced State Table:

a, b, c, d, e

→ Again draw the state table but upto e

P.S	0 0	0 1	1 1	1 0
a	c, -	a, 0	b, -	→ -
b	-, -	a, -	b, 1	c, -
c	c, 0	a, -	, -	d, -
d	c, -	, -	b, -	d, 0
e	f, -	, -	b, -	e, 1
f	f, 1	a, -	, -	e, -

→ Next state

→ output

(Incompletely Specified)

- Ans2.
- Step-1: 1st check o/p if same or not
 - Step-2: Blank spaces & any o/p & N/S & (0,1) are treated as same.
 - Step-3: After checking, for o/p check N-S.

Implication Table:

1st pass Implication Table

b	✓			
c	✓	(d,e)		
d	✓	(d,e)	✓	
e	(c,f)	✓	(c,f)	X
f	(c,f)	✓	X	(c,f)

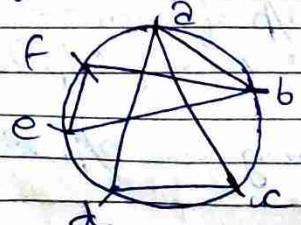
2nd pass Implication Table:

b	✓			
c	✓	X		
d	✓	X	✓	
e	X	✓	X	X
f	X	✓	X	X

Compatible pair: (Ticked one in 2nd pass I.T.):

- (a,b) (b,c) (c,d)
- (a,c) (b,f) (e,f)
- (a,d)

Merger diagram:



Maximum Compatible Pair:

$$b = e = f$$

$$a = c = d$$

$$a = b$$

Reduced State Table:

State Table (after Reduction)

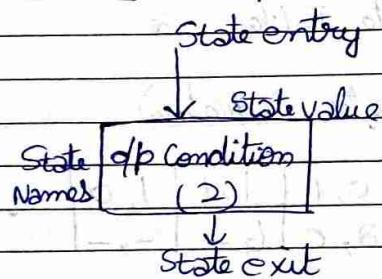
P.S	0,0	0,1	1,1	1,0
a	c,-	a,0	b,-	-, -

23/10/19

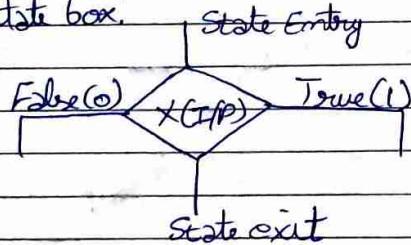
Unit - 4 Algorithmic State Machines

Components:

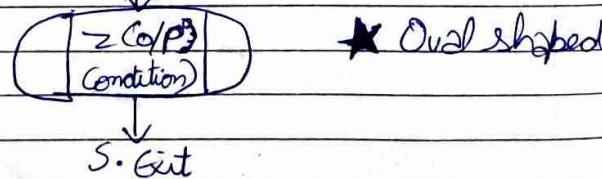
1. State box: It represents the state of the system.



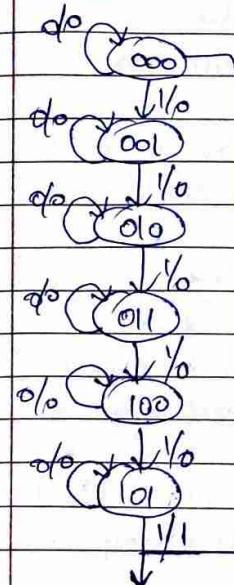
2. Decision box: It should be followed & associated with a state box.



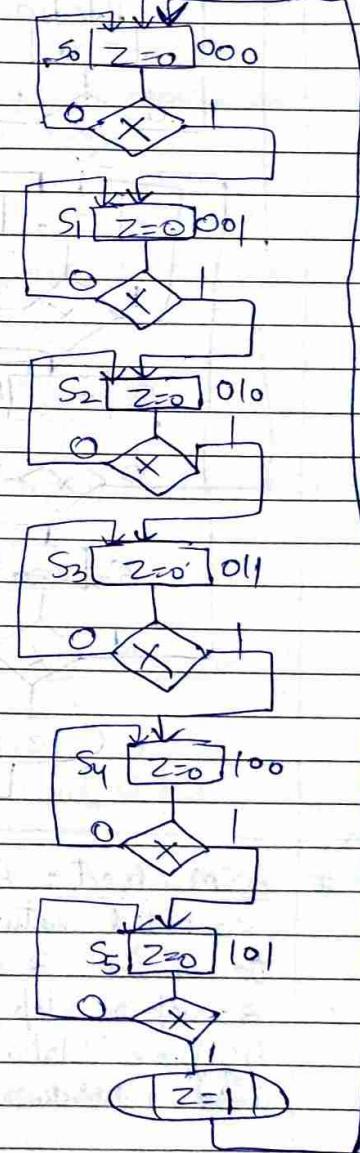
3. Condition Box: Conditional output depends on both the state & inputs S-E also known as Mealy op's



Ans. 1.

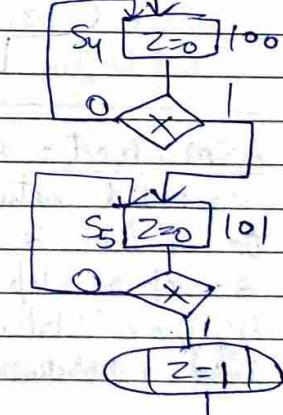
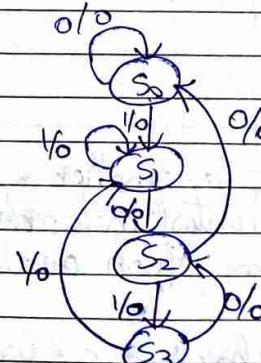


Asm Chart: (Q1)



Q2.

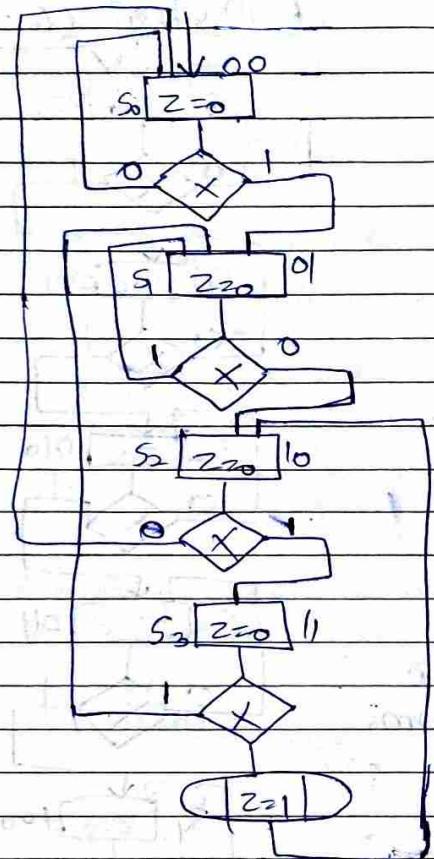
Derive Asm for the given state diagram.



$(Z = 1 \rightarrow \text{condition box})$

- Q1. Design ASM chart for the given state diagram.

Ans 2:



2. ASM charts are equal to state graphs & leads to hardware realization.
3. ASM charts can describe the operation of both combinational & sequential circuits.

Parity Generators / Parity Checkers: A parity bit is an additional bit which is added to a binary word in order to make no. of 1's in a new word formed an even parity or odd parity.

Parity Generators are logic circuits which generates the parity bit for an even parity or an odd parity.

Truth Table of Parity Checkers:

D	C	B	A	P	K-Map (For P):
0	0	0	0	0	00 00 01 11 10
0	0	0	1	1	00 00 01 11 10
0	0	1	0	1	00 00 01 11 10
0	0	1	1	0	00 00 01 11 10
0	1	0	0	1	11 00 11 00 11
0	1	0	1	0	10 10 10 10 10
0	1	1	0	0	11 11 11 11 11
0	1	1	1	1	00 01 10 11 10
1	0	0	0	1	00 01 10 11 10
1	0	0	1	0	00 01 10 11 10
1	0	1	0	0	00 01 10 11 10
1	0	1	1	1	00 01 10 11 10
1	1	0	0	0	00 01 10 11 10
1	1	0	1	1	00 01 10 11 10
1	1	1	0	1	00 01 10 11 10
1	1	1	1	0	00 01 10 11 10

$\Rightarrow P = \bar{A}\bar{B}\bar{C}\bar{D} + \bar{A}\bar{B}\bar{C}D + \bar{A}\bar{B}C\bar{D} + A\bar{B}\bar{C}D + \bar{A}\bar{B}CD + A\bar{B}C\bar{D} + ABC\bar{D}$

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

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

$P = (\bar{A}\bar{B} + AB)(C\oplus D) + (CD + \bar{C}D)(A\oplus B)$

$P = (A \odot B)(C \oplus D) + (A \oplus B)(C \odot D)$

$P = A \oplus B \oplus C \oplus D$

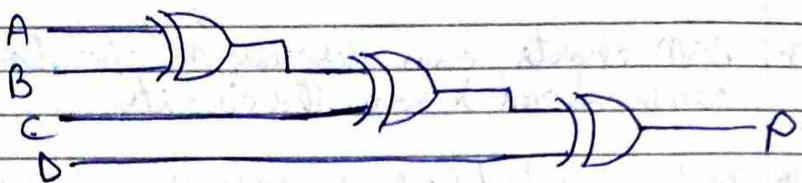
ASM chart: ASM chart is another name for sequential network. Sequential network is used to control a digital system which carries out a step by step procedure.

ASM or static machine charts are flow charts, used in hardware design of digital systems.

Advantages of ASM charts:

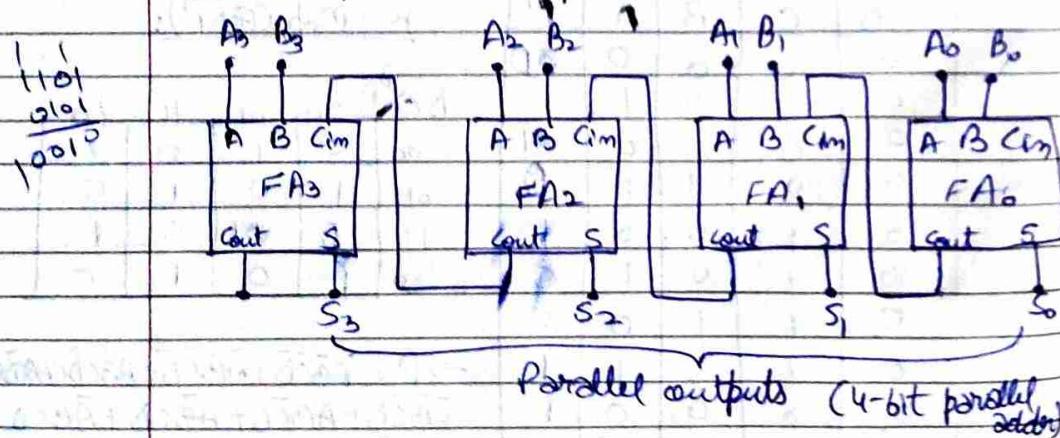
- Using ASM charts the operation of digital system is made easy.

Circuit Diagrams



Parallel Adders: A full-adder is capable of adding two 1-bit binary numbers and a carry-in. When two n-bit binary numbers are to be added, then the no. of full adders required will be equal to the no. of bit m in each number.

A parallel adder is used to add two n-bit no.'s in parallel form & to produce the sum bits as parallel output.



$$\text{Ex: } A_3 A_2 A_1 A_0 = 1101 \quad \& \quad B_3 B_2 B_1 B_0 = 0101$$

$$\begin{array}{cccc} A_0 & B_0 & C_{in} & S_0 \quad (\text{out}) \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 \\ \hline 1 & + & 1 & + \\ 0 & & 1 & \\ \hline 0 & & 1 & \end{array}$$

$$\begin{array}{cccc} A_1 & B_1 & C_{in} & S_1 \quad (\text{out}) \\ 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 \\ \hline 0 & + & 1 & = \\ 0 & & 1 & \end{array}$$

$$\begin{array}{cccc} A_2 & B_2 & C_{in} & S_2 \quad (\text{out}) \\ 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 \\ \hline 1 & + & 1 & = \\ 0 & & 1 & \end{array}$$

$$\begin{array}{cccc} A_3 & B_3 & C_{in} & S_3 \quad (\text{out}) \\ 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 \\ \hline 1 & + & 0 & = \\ 0 & & 1 & \end{array}$$

$$\begin{array}{c} \text{Resulting O/P} = S_3 S_2 S_1 S_0 \\ \hline 1 & 0 & 0 & 1 \end{array}$$

Fan-in :- It is defined as the number of inputs that a logic circuit can handle.

Fan-out (loading factor) : The fan-out or loading factor of a logic gate is defined as the maximum no. of standard loads that the output of a gate can drive without impairing its normal operation.

Transistor Transistor Logic (TTL)

TTL or T²L family is so named because of its dependence on transistors alone to perform basic logic operations. It is the most popular & widely used ~~bipolar~~ digital IC family.

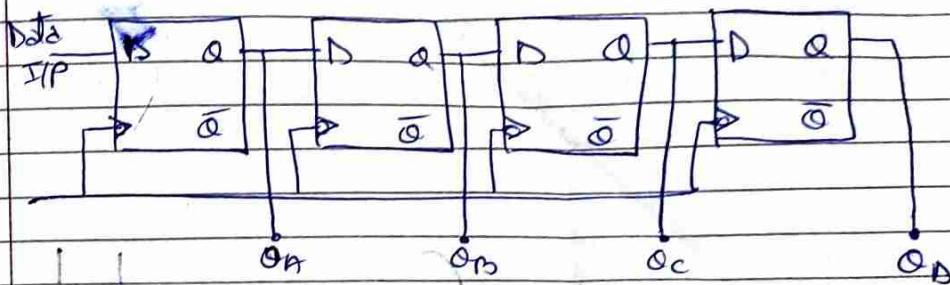
Advantages:

- Good speed
- Low manufacturing cost
- Wide range of circuit
- Availability in SSI (Small Scale Integration) & MSI (Medium Scale Integration)

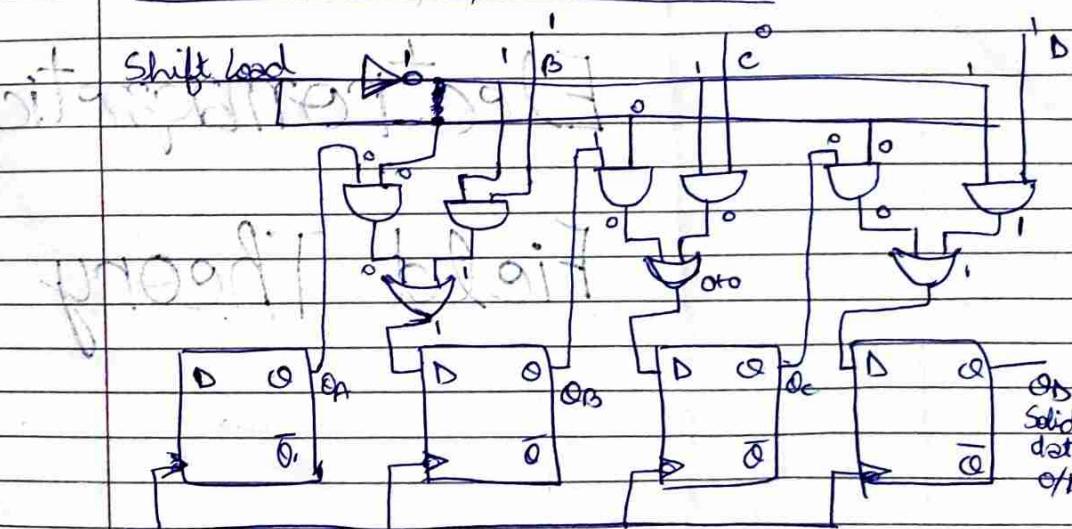
Disadvantages:

- Relatively high power consumption.
- Generation of noise spikes.
- Moderate packing density.
- Tight V_{cc} tolerance.

2. SIPO (Serial I/P Parallel O/P):

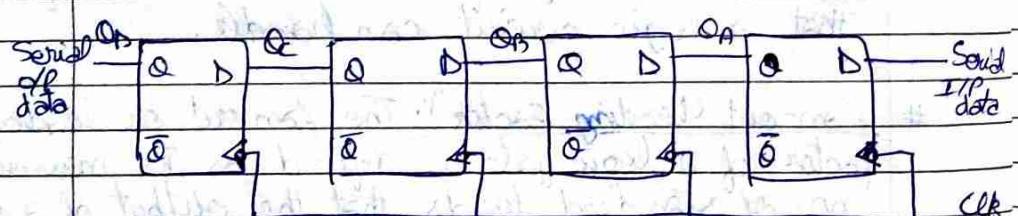


3. PISO (Parallel I/P Serial O/P):



Classification of Registers:

1. SISO (Serial I/P Serial O/P): (shift left operation)



	(L.O)				(R.O)			
Shift Pulse	Q ₁	Q ₂	Q ₃	Q ₄	Q ₁	Q ₂	Q ₃	Q ₄
0	0	0	0	0	0	0	0	0
1	0	0	0	1	1	0	0	0
2	0	0	1	1	1	1	0	0
3	0	1	1	1	1	1	1	0
4	1	1	1	1	1	1	1	1

4. PI PO (Parallel I/P Parallel O/P):

