

UNIT - 1

Boolean Algebra

1. Idempotent Law

$$x + x = x$$

$$x \cdot x = x$$

2. Complementary Law

$$x + \bar{x} = 1$$

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

3. Associative Law

$$(x + y) + z = x + (y + z)$$

$$(x \cdot y) \cdot z = x \cdot (y \cdot z)$$

4. commutative law

$$x + y = y + x$$

$$x \cdot y = y \cdot x$$

5. Distributive law

$$x(y + z) = xy + xz$$

6. De Morgan's Law

$$\overline{x + y} = \bar{x} \cdot \bar{y}$$

$$\overline{x \cdot y} = \bar{x} + \bar{y}$$

7. Absorption

$$x + xy = x$$

$$x(x + y) = x$$

8. Elimination

$$x + \bar{x}y = x + y$$

$$x(\bar{x} + y) = xy$$

9. Uniting Theorem

$$xy + x\bar{y} = x$$

$$(x+y)(x+\bar{y}) = x$$

Q:

$$\begin{aligned}
 & \overline{(AB+C)\bar{D}+E} \\
 &= \overline{(AB+C)\bar{D}\bar{E}} \\
 &= \overline{[(AB+C)+\bar{D}]} \bar{E} \\
 &= \overline{(AB+C)\bar{E}} + \bar{D}\bar{E} \\
 &= \bar{A}\bar{B}\bar{C}\bar{E} + \bar{D}\bar{E} \\
 &= (\bar{A}+\bar{B})\bar{C}\bar{E} + \bar{D}\bar{E} \\
 &= \bar{E} [\bar{A}\bar{C} + \bar{B}\bar{C} + \bar{D}] \\
 &= \bar{E} [(\bar{A}+\bar{B})\bar{C} + \bar{D}]
 \end{aligned}$$

2. XOR \rightarrow XNOR

$$\begin{aligned}
 f &= \bar{a}\bar{b} + \bar{a}b \\
 \bar{f} &= \bar{a}\bar{b} + \bar{a}b \\
 &= \bar{a}\bar{b} \cdot \bar{a}b \\
 &= (\bar{a}+b) \cdot (a+\bar{b}) \\
 &= a\bar{a} + \bar{a}\bar{b} + ab + b\bar{b} \\
 &= ab + \bar{a}\bar{b}
 \end{aligned}$$

* Minterms and Maxterms

Minterms or standard product - product terms

Maxterms or standard sum - sum terms.

x	y	z	Minterms	Maxterms
0	0	0	$\bar{x}\bar{y}\bar{z}$	$x+y+z$
0	0	1	$\bar{x}\bar{y}z$	$x+y+\bar{z}$
0	1	0	$\bar{x}yz$	$x+\bar{y}+z$
0	1	1	$\bar{x}y\bar{z}$	$x+\bar{y}+\bar{z}$
1	0	0	$x\bar{y}\bar{z}$	$\bar{x}+y+z$
1	0	1	$x\bar{y}z$	$\bar{x}+y+\bar{z}$
1	1	0	$xy\bar{z}$	$\bar{x}+y+z$
1	1	1	xyz	$\bar{x}+\bar{y}+\bar{z}$

Minterm : zero is prime (complement)

one is unprime

Maxterm : one is prime (complement)

zero is unprime

* Radix complement: $r-2$'s complement

r - base

N - given number

n - number of digits.

$(r^n - N)$ gives the complement.

$$X = 1010100 \quad Y = 1000011$$

$$X - Y : \quad \begin{array}{r} 1010100 \\ - 1000011 \\ \hline \end{array}$$

$$\Rightarrow 0000\ 0111100 + 1 = 0111101$$

$$\Rightarrow \begin{array}{r} 1010100 \\ + 0111101 \\ \hline \end{array}$$

2's complement

$$\cancel{\text{discarded}} \quad \begin{array}{r} 10010001 \\ \hline \end{array}$$

$$Y - X : \quad \begin{array}{r} 1000011 \\ - 1010100 \\ \hline \end{array}$$

$$\Rightarrow 0101011 + 1 = 0101100 \quad \text{2's complement}$$

$$\Rightarrow \begin{array}{r} 1000011 \\ + 0101100 \\ \hline \end{array}$$

$$\begin{array}{r} 1101111 \\ \Rightarrow 0010000 + 1 = \underline{\underline{0010001}} \end{array}$$

* Positive logic

Logical LOW = 0

Logical HIGH = 1

Negative logic

Logical low = 1

Logical HIGH = 0

* Sequential circuits:

The output depends on previous output. Hence has memory.

* combinational circuits:

The output is independent of previous output. Hence no memory.

- * Literals:
The variable in direct or complemented form.
Ex: $a + \bar{a}b$ - 3 literals.

Q: $f = AB + BC + \bar{A}C$

$$\begin{aligned} &= AB + BC(A + \bar{A}) + \bar{A}C \\ &= AB + ABC + \bar{A}BC + \bar{A}C \\ &= AB(1 + C) + \bar{A}C(1 + B) \\ &= AB + \bar{A}C \end{aligned}$$

- * Duality theorem:

To convert the positive logic and negative logic into a positive logic.

STEP 1: Keep the literal as it is. Replace 1 by 0 and 0 by 1.

STEP 2: Replace OR by AND (i.e., '+' by '·') and AND by OR (i.e., '·' by '+').

Q: Find the dual of:

1. $AB + CD \rightarrow (A+B)(C+D)$
2. $(A+B)(B+C)(C+A) \Rightarrow AB + BC + AC$

- * SOP: Sum of Products (minterms)

POS: Product of sum (maxterms)

minterms: outputs which are high are considered

maxterms: outputs which are low are considered

Representation

minterms (m_0, m_1, m_2, \dots)

maxterms (M_0, M_1, M_2, \dots)

- * XOR - odd-parity detector (odd number of ones)
EVEN-parity generator (even number of ones)
ODD function
Inequality detector
- XNOR - Equivalence detector
ODD-parity generator
EVEN-parity detector
EVEN function.
Coincidence detector.

* canonical Expression:

All the literals are present in the terms of the expression.

* K-Map: collection of cells.

1. Number of cells depends on number of variables
i.e., number of cells = m
 $m = 2^n$
2. Naming of cells should satisfy adjacency property.
The distance between two cells should one and then these are said to be adjacent cells.
3. The grouping can be done in pairs, quads or octets.
4. Don't care terms (x) can be treated as zero or one.
5. While grouping SOP '1's must be covered.
6. While grouping POS '0's must be covered
7. Pair eliminates one variable, quad eliminates two variables and octet eliminates three variables.

* distance - change in bits.

Ex 01 - 11 - change in bits is 1

01 - 10 - change in bits is 2

	A	B	C	D	00	01	11	10	AB	0	1
0	0	0	0	0	0	0	1	0	00	0	1
1	1	0	0	0	1	1	1	0	01	2	3
									11	6	7
									10	4	5

2 variables3 variables

	A	B	00	01	11	10
00	00	0	0	1	3	2
01	01	4	5	7	6	8
11	11	12	13	15	14	16
10	10	9	11	10	18	19

4 variables

* Grouping of SOP:

Q1: $f(x, y, z) = \sum(2, 3, 4, 5)$

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

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

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

$$= \bar{A}B$$

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

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

$$= A\bar{B}$$

~~AB + A\bar{B} + B\bar{C} + \bar{A}B\bar{C}~~

$$\therefore f(x, y, z) = \bar{A}B + A\bar{B} //$$

Q2: Verify:

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

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

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

$$= \bar{A}B(C + 1) + \bar{B}C(\bar{A} + A) + BC$$

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

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

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

K map:

$$\begin{aligned}
 F &= \bar{A}\bar{B} + \bar{A}B + A\bar{B}C + BC \\
 &= \bar{A}C(B+\bar{B}) + \bar{A}B(L+\bar{L}) + A\bar{B}C + BC(A+\bar{A}) \\
 &= \bar{A}BC + \bar{A}\bar{B}C + \bar{A}BL + \bar{A}\bar{B}\bar{C} + A\bar{B}C + ABC + \bar{A}BC \\
 &= \bar{A}BC + \bar{A}\bar{B}C + \bar{A}BC + \bar{A}\bar{B}C + ABC \\
 &\Rightarrow 011, 001, 010, 101, 111
 \end{aligned}$$

$$F(A,B,C) = \sum m(1, 2, 3, 5, 7)$$

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

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

prime implicants

NOTE: The reduced form of expression is prime implicants.

$$Q3: F(P,Q,R,S) = \sum m(0,2,3,7,11,13,14,15)$$

PQ	RS	00	01	11	10
00	1	0	1	1	1
01	0	0	1	0	
11	0	1	1	1	
10	0	0	1	0	

$$F(P,Q,R,S) = \underline{\bar{P}\bar{Q}\bar{S}} + RS + PQS + PQT$$

Prime implicants: $\bar{P}\bar{Q}\bar{S}$, PQS , PQR , RS
 1, 3, 4 are essential prime implicant
 that is it can be grouped only
 in one possible way.

$$Q4: F(P,Q,R,S) = \sum m(0,1,2,4,5,6,8,9,12,13,14)$$

Find the minimal SOP expression.

PQ	RS	00	01	11	10
00	1	1	0	1	1
01	1	1	0	1	
11	1	1	0	1	
10	1	1	0	0	

$$F(P,Q,R,S) = \underline{\bar{R}} + \underline{\bar{P}\bar{S}} + \underline{Q\bar{R}\bar{S}}$$

Q5: $F = \sum m(2, 6, 7, 9, 13, 15)$, find the prime implicants

AB\CD	00	01	11	10
00	0	0	0	1
01	0	0	1	1
11	0	1	1	0
10	0	1	0	0

Prime implicants

$\bar{A}C\bar{D}$, $A\bar{B}D$, $B\bar{C}D$, $\bar{A}\bar{B}C$, $\bar{A}C\bar{D}$

Essential Prime Implicants

$\bar{A}C\bar{D}$, $\bar{A}C\bar{D}$

Non essential prime implicants

$A\bar{B}D$, $B\bar{C}D$, $\bar{A}B\bar{C}$

Q6:

$F = \sum m(0, 2, 3, 5, 7, 8, 9, 10, 11, 13, 15)$

AB\CD	00	01	11	10
00	1	0	1	1
01	0	1	1	0
11	0	1	1	0
10	1	1	1	1

Prime Implicants

$\bar{B}\bar{D}$, CD , $\bar{A}\bar{B}$, BD , AD , $\bar{B}C$

Essential Prime Implicants

$\bar{B}\bar{D}$, BD

Non essential Prime Implicants

CD , AD , $\bar{A}\bar{B}$, $\bar{B}C$

$$SOP = \bar{B}\bar{D} + CD + A\bar{B} + BD$$

$$\text{or } (\bar{B}\bar{D} + BD + A\bar{B} + \bar{B}C) \text{ or } (\bar{B}\bar{D} + BD + AD + \bar{B}C)$$

Q7: $F = \sum m(0, 2, 4, 5, 6, 7, 8, 10, 13, 15)$

AB\CD	00	01	11	10
00	1	0	0	1
01	1	1	1	1
11	0	1	1	0
10	1	0	0	1

Prime implicants

$\bar{B}\bar{D}$, BD , $\bar{A}B$, $\bar{A}\bar{D}$

Essential Prime Implicants

$\bar{B}\bar{D}$, BD

Non Essential Prime Implicants

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

$$SOP = \bar{B}\bar{D} + \bar{A}\bar{D} + BD$$

$$\text{or } (\bar{B}\bar{D} + \bar{A}B + BD)$$

Q8: Find all the prime implicants for the Boolean function and determine the essential prime implicants.

$$F = \Sigma m(2, 3, 4, 5, 6, 7, 9, 11, 12, 13)$$

AB\CD	00	01	11	10
00	0	0	1	1
01	1	1	1	1
11	1	1	0	0
10	0	1	1	0

Essential Prime Implicants

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

Prime Implicants

$$\bar{A}C, \bar{B}C, A\bar{B}D, \bar{A}B, A\bar{C}D, \bar{B}CD$$

Q9: Find the essential and non essential prime implicants.

$$F = \Sigma m(0, 2, 3, 5, 7, 8, 10, 11, 14, 15)$$

Wx\Yz	00	01	11	10
00	1	0	1	1
01	0	1	1	0
11	0	0	1	1
10	1	0	1	1

Essential Prime Implicants.

$$\bar{X}\bar{Z}, \bar{W}XZ, WY$$

Non Essential Prime Implicants.

$$YZ, \bar{X}Y$$

$$SOP: \bar{X}\bar{Z} + \bar{W}XZ + WY + YZ$$

$$OR (\bar{X}\bar{Z} + \bar{W}XZ + WY + \bar{X}Y)$$

* 5 Variables:

BC\DE	00	01	11	10
00	0	1	3	2
01	4	5	7	6
11	12	13	15	14
10	2	9	11	10

BC\DE	00	01	11	10
00	16	14	18	15
01	20	21	23	22
11	28	29	31	25
10	24	25	27	26

$$A = 0$$

$$A = 1$$

The maps are placed on one another. Hence 0 and 16 are adjacent - similarly 12 and 28, 26 and 10.

Q1: $F(A, B, C, D, E) = \sum m(0, 2, 4, 6, 9, 13, 21, 23, 25, 29, 31)$

		BC				DE				00	01	11	10
		00	01	11	10	00	01	11	10	00	01	11	10
		00	1	0	0	1	A=0	01	0	1	1	1	0
		01	1	0	0	1		11	0	1	1	0	A=1
		11	0	1	0	0		10	0	1	0	0	
		10	0	1	0	0							

$$F(A, B, C, D, E) = \overline{ABE} + A\overline{CE} + B\overline{DE}$$

Q2: Solve:

$$A=0$$

		BC				DE				00	01	11	10
		00	01	11	10	00	01	11	10	00	01	11	10
		00	1	0	1	1				00	1	0	0
		01	0	1	1	0				01	0	0	0
		11	0	0	1	1				11	0	1	1
		10	1	0	1	1				10	1	0	1

$$A=1$$

		BC				DE				00	01	11	10
		00	01	11	10	00	01	11	10	00	01	11	10
		00	1	0	0	1				00	1	0	0
		01	0	1	1	0				01	0	0	0
		11	0	0	1	1				11	0	1	1
		10	1	0	1	1				10	1	0	1

$$F(A, B, C, D, E) = \overline{CE} + \overline{B}\overline{D} + BD + \overline{A}\overline{D}E + \overline{A}\overline{B}\overline{C}E + ABC$$

* Product of Sums: (POS)

Q1: $F = A + \overline{B}\overline{C}$

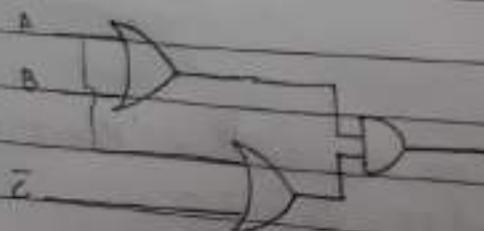
		BC				DE			
		00	01	11	10	00	01	11	10
0	0	0	0	0	1				
1	1	1	1	1	1				

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

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

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

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



Inverters are not used.

Q2: $F = \sum m(1, 3, 4, 5, 9, 11, 14, 15)$, write the MPOS and MSOP.

AB		CD		00	01	11	10
00	01	00	1	1	0		
01	1	1	0	0			
10	0	0	1	1			
10	0	1	1	0			

AB		CD		00	01	11	10
00	01	00	0	1	1	0	
01	1	1	0	0			
11	0	0	0	0			
10	0	1	1	0			

$$\bar{F} = \overline{BD} + AB\bar{C} + \bar{A}BC$$

$$\bar{F} = \overline{BD} \cdot \overline{ABC} \cdot \overline{ABC}$$

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

$$F = \overline{BD} + \overline{ABC} + ABC$$

Q3: $F(A, B, C, D) = \sum m(0, 1, 2, 5, 8, 9, 10)$

AB		CD		00	01	11	10
00	01	00	1	1	0	1	
01	0	1	0	0			
11	0	0	0	0			
10	1	1	0	1			

AB		CD		00	01	11	10
00	01	00	1	1	0	1	
01	0	1	0	1	0	0	
11	0	0	0	0	0	0	
10	1	1	1	0	1	0	

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

$$F = \overline{BD} + \overline{BC} + \overline{ACD}$$

Q4: $F = \pi M(1, 3, 5, 7, 13, 15)$

AB		CD		00	01	11	10
00	01	00	1	0	0	1	
01	1	0	0	1			
11	1	0	0	1			
10	1	1	1	1			

$$F = (A + \bar{B})(\bar{B} + \bar{D})$$

Q5: Simplify the following expression to POS and SOP.

$$\bar{x}\bar{z} + \bar{y}\bar{z} + y\bar{z} + xy$$

$$F = (y + \bar{y})\bar{x}\bar{z} + (\bar{x} + x)y\bar{z} + (x + \bar{x})xy$$

$$F = \bar{x}y\bar{z} + \bar{x}\bar{y}\bar{z} + x\bar{y}\bar{z} + \bar{x}\bar{y}z + x\bar{y}z + xyz$$

$$F = \bar{x}y\bar{z} + \bar{x}\bar{y}z + x\bar{y}\bar{z} + \bar{x}\bar{y}z + xyz$$

$$F = \Sigma m(0, 2, 4, 6, 7)$$

	$\bar{w}\bar{z}$	00	01	11	10
0	1	0	0	1	
1	1	0	1	1	

$w\bar{y}\bar{z}$	00	01	11	10
0	1	0	0	1
1	1	0	1	1

$$POS: (\bar{x} + z)(y + \bar{z})$$

$$SOP: \bar{z} + xy$$

Q6: Don't care condition:

$$F(w, x, y, z) = \Sigma(1, 3, 7, 11, 15)$$

$$d(w, x, y, z) = \Sigma(0, 2, 5)$$

$$SOP: \bar{w}z + yz //$$

$w\bar{x}\bar{y}\bar{z}$	00	01	11	10
00	x	1	1	x
01	0	x	1	0
11	0	0	1	0
10	0	0	1	0

$$\bar{w}wx\bar{y}x\bar{z} + \bar{w}\bar{x}x\bar{y}x\bar{z}$$

$wx\bar{y}\bar{z}$	00	01	11	10
00	x	1	1	x
01	0	x	1	0
11	0	0	1	0
10	0	0	1	0

$$POS: z(\bar{w} + \bar{y}) //$$

Q7: $F = \sum(1, 3, 4, 5, 9, 11, 14, 15) + \sum d(2, 6, 7, 8)$

sol:

AB	CD	00	01	11	10
00		0	1	1	X
01		1	1	X	X
11		0	0	1	1
10		X	1	1	0

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

AB	CD	00	01	11	10
00		0	1	1	X
01		1	1	X	X
11		0	0	1	1
10		X	1	1	0

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

Q8: $F = \sum(0, 6, 8, 13, 14) + \sum d(2, 4, 10)$

sol:

AB	CD	00	01	11	10
00		1	0	0	X
01		X	0	0	1
10		0	1	0	1
11		1	0	0	X

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

AB	CD	00	01	11	10
00		1	0	0	X
01		X	0	0	1
11		0	1	0	1
10		1	0	0	X

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

* Tabulation Method:

Quine McCluskey Technique is a tabulation method used when the number of variables is high.

Q1: $f(x_1, x_2, x_3, x_4) = \Sigma(0, 5, 7, 8, 9, 10, 11, 14, 15)$

STEP 1: Write all the minterms in binary form

$$m_0 = 0000 \quad m_{10} = 1010$$

$$m_5 = 0101 \quad m_{11} = 1011$$

$$m_7 = 0111 \quad m_{14} = 1110$$

$$m_8 = 1000 \quad m_{15} = 1111$$

$$m_9 = 1001$$

STEP 2: Arrange these binary numbers into different groups in terms of numbers of ones the number contains.

$x_1 \ x_2 \ x_3 \ x_4$

- 0 0 0 0 (0) Zero number of ones

- 1 0 0 0 (1) One number of ones

- 0 1 0 1 (5)

- 1 0 0 1 (9) Two number of ones

- 1 0 1 0 (10)

- 0 1 1 1 (11)

- 1 0 1 1 (12) Three number of ones

- 1 1 1 0 (14)

- 1 1 1 1 (15) Four number of ones

STEP 3: Compare each binary number with every term in the adjacent next higher category and if they differ only by one position put a check mark and copy the term in the next table with a symbol (-) in the position that they differ.

First reduction $\bar{0} \ 0 \ 0 \ (0,8)$ $1 \ 0 \ 0 \ (8,9)$ $1 \ 0 \ - \ 0 \ (8,10)$ $0 \ 1 \ - \ 1 \ (5,4)$ $1 \ 0 \ - \ 1 \ (9,11)$ ~~$1 \ 0 \ 1 \ (10,11)$~~ $1 \ - \ 1 \ 0 \ (10,14)$ $- \ 1 \ 1 \ 1 \ (4,15)$ $1 \ - \ 1 \ 1 \ (11,15)$ ~~$1 \ 1 \ 1 \ (14,15)$~~

STEP 4: Apply step 3 procedure till you get the reduced form.

Second reduction $1 \ 0 \ - \ - \ (8,9,10,11)$ $1 \ - \ 1 \ - \ (10,14,14,15)$

STEP 5: List the minterms which are unchecked which are the prime implicants.

 $1 \ - \ 1 \ - : x_1, x_3$ $1 \ 1 \ 0 \ - : x_1, \bar{x}_2$ $- \ 1 \ 1 \ 1 : x_2, x_3, x_4$ $0 \ 1 \ - \ 1 : \bar{x}_1, x_2, x_4$ $- \ 0 \ 0 \ 0 : \bar{x}_2, \bar{x}_3, \bar{x}_4$

STEP 6: Write the implicant selection chart

Prime Implicants	0	5	4	8	9	10	11	14	15
x_1, x_3						x	x	(x)	x
x_1, \bar{x}_2				x	(x)	x	x		
x_2, x_3, x_4			x						x
\bar{x}_1, x_2, x_4		(x)	x						
$\bar{x}_2, \bar{x}_3, \bar{x}_4$	(x)			x					

STEP 1: Write the minimal expression.

$$f = x_1 x_3 + x_1 \bar{x}_2 + \bar{x}_1 x_2 x_4 + \bar{x}_2 x_3 \bar{x}_4$$

Verification using K Map:

$x_1 \backslash x_2$	00	01	11	10
00	1	0	0	0
01	0	1	1	0
11	0	0	1	1
10	1	1	1	1

$$\text{SOP: } x_1 \bar{x}_2 + x_1 x_3 + \bar{x}_1 x_2 x_4 + \bar{x}_2 \bar{x}_3 \bar{x}_4$$

Q2: $f = \Sigma m (0, 2, 3, 6, 7, 8, 9, 10, 13)$

First Reduction Second Reduction

<u>sd:</u>	$m_0 = 0000 - 0000 (0)$	$-00-0 (0,2)$	$-0-0 (0,2,4)$
	$m_2 = 0010 - 0010 (2)$	$-0000 (0,2)$	$0-1- (2,5,7)$
	$m_3 = 0011 - 1000 (3)$	$-001- (0,5)$	
	$m_6 = 0110 - 0011 (6)$	$-0-10 (3,6)$	
	$m_7 = 0111 - 0110 (7)$	$-010 (3,10)$	
	$m_8 = 1000 - 1001 (8)$	$100- (8,9)$	
	$m_9 = 1001 - 1010 (10)$	$-10-0 (8,10)$	
	$m_{10} = 1010 - 0111 (10)$	$-0-11 (3,4)$	
	$m_{13} = 1101 - 1101 (13)$	$-011- (6,7)$	
			$1-01 (9,13)$

Prime implicants

$$0-1 : \bar{x}_1 x_3$$

$$-0-0 : \bar{x}_2 \bar{x}_4$$

$$1-01 : x_1 \bar{x}_3 x_4$$

$$100- : x_1 \bar{x}_2 \bar{x}_3$$

Prime Implicants	0	2	3	6	7	8	9	10	13
$\bar{x}_1 \bar{x}_3$		x	(x)	(x)	(x)				
$\bar{x}_2 \bar{x}_4$	(x)	x				x			
$x_1 \bar{x}_3 x_4$						x	x		
$x_1 \bar{x}_2 \bar{x}_3$						x	x		(x)

$$SOP = \bar{x}_1 x_3 + \bar{x}_2 \bar{x}_4 + x_1 \bar{x}_3 x_4 //$$

Q3: $f = \sum m(3, 4, 5, 7, 10, 12, 14, 15) + \sum d(2)$

*ABCD First Reduction

Second Reduction

$m_2 = 0010$	0010 (2)	001_ (2,3)
$m_3 = 0011$	0100 (4)	-010 (2,10)
$m_4 = 0100$	0011 (3)	010_ (4,5)
$m_5 = 0101$	0101 (5)	-100 (1,12)
$m_7 = 0111$	1010 (10)	0-11 (3,7)
$m_{10} = 1010$	-1100 (12)	01-1 (5,9)
$m_{12} = 1100$	-0111 (7)	1-10 (10,14)
$m_{14} = 1110$	-1110 (14)	11-0 (12,14)
$m_{15} = 1111$	-1111 (15)	-111 (7,15)
		111- (14,15)

Prime Implicants

$\bar{A}B\bar{C}$ 001_ : $\bar{A}\bar{B}C$

$\bar{D}010$: $\bar{B}CD$

$$SOP = \bar{A}CD + A\bar{C}D + \bar{A}B\bar{C} + B\bar{C}D + BCD$$

$D10_-$: $\bar{A}BC$

-100 : $B\bar{C}D$

AB CD
00 00 01 11 10

00 0 0 1 x

00 0 1 1 0

01 1 1 1 0

11 1 0 1 1

10 0 0 0 1

111 : BCD

$$SOP = B\bar{C}D + \bar{A}BD + \bar{A}CD + ABC$$

+ $A\bar{C}D$

111- : ABC

Prime Implicants	2	3	4	5	7	10	12	14	15
$A\bar{B}C$	x								
$\bar{B}CD$		x							
$\bar{A}BC$			x					x	
$B\bar{C}D$			x			x			
$\bar{A}CD$				x	x				
$\bar{A}BD$					x			x	
$AC\bar{D}$				x	x			x	x
$AB\bar{D}$					x	x			x
BCD						x			x
ABC							x	x	

Q4: $F = \sum(1, 2, 3, 5, 9, 12, 14, 15) + \sum d(4, 8, 11)$

	ABCD	First Reduction	Second Reduction
sol:	$m_1 = 0001$	-0001 (1)	-0-1 (1, 3)
	$m_2 = 0010$	-0010 (2)	0-01 (0, 5)
	$m_3 = 0011$	-0100 (4)	-001 (0, 9)
	$m_4 = 0100$	-1000 (8)	001- (2, 3)
	$m_5 = 0101$	-0011 (3)	010- (4, 5)
	$m_6 = 1000$	-0101 (5)	-100 (4, 12)
	$m_7 = 1001$	-1001 (9)	100- (3, 9)
	$m_8 = 1011$	-1100 (12)	1-00 (3, 12)
	$m_9 = 1100$	-1011 (11)	-011 (3, 11)
	$m_{10} = 1110$	-1110 (14)	11-0 (12, 14)
	$m_{11} = 1111$	-1111 (15)	1-11 (11, 15)
			111- (14, 15)

Prime Implicants

$$\begin{array}{lll}
 -0-1 : \bar{B}D & 1-00 : A\bar{C}\bar{D} & 001- : \bar{A}\bar{B}C \\
 111- : A\bar{B}C & 100- : \bar{A}\bar{B}\bar{C} & 0-01 : \bar{A}\bar{C}D \\
 1-11 : ACD & -100 : \bar{B}\bar{C}D & 00-1 : \bar{A}\bar{B}D \\
 11-0 : A\bar{B}\bar{D} & 010- : \bar{A}\bar{B}\bar{C} &
 \end{array}$$

Prime Implicants	1	2	3	4	5	8	9	11	12	14	15
\overline{BD}	x		x				x	d			
ABC									x	x	
ACD								d			x
$A\overline{B}\overline{D}$									x	x	
$A\overline{C}D$							d		x		
$A\overline{B}\overline{C}$							d	x			
$\overline{B}\overline{C}\overline{D}$							x				
$\overline{A}\overline{B}\overline{C}$							d	x			
$\overline{A}\overline{B}\overline{D}$	x										
$\overline{A}\overline{C}\overline{D}$											

$$SOP = \overline{A}\overline{B}C + \overline{B}D + ABC + A\overline{B}\overline{D} + \overline{A}\overline{C}D$$

SOP/F

Q5: $F = \Sigma(0, 5, 7, 8, 9, 10, 11, 14, 15)$

	ABCD	First Reduction	Second reduction
Sol:	$m_0 = 0000$	- 0000 (w)	- 000 (0, 8)
	$m_5 = 0101$	- 1000 (z)	- 100- (8, 9)
	$m_7 = 0111$	- 0101 (z)	- 10-0 (8, 10)
	$m_8 = 1000$	- 1001 (y)	01-1 (5, y)
	$m_9 = 1001$	- 1010 (y)	- 10-1 (9, 11)
	$m_{10} = 1010$	- 0111 (y)	- 101- (10, 11)
	$m_{11} = 1011$	- 1011 (y)	- 1-10 (10, 14)
	$m_{14} = 1110$	- 1110 (y)	- 111 (y, 15)
	$m_{15} = 1111$	- 1111 (y)	- 1-11 (11, 15)
			- 111- (14, 15)

Prime Implicants

 $1-1-$: AC $01-1$: $\overline{A}BD$ $10--$: $A\overline{B}$ -000 : $\overline{B}\overline{C}\overline{D}$ -111 : BCD

Prime Implicants	0	5	7	8	9	10	11	14	15
AC						x	x	(x)	x
$\bar{A}B$				x	(y)	x	x		x
BCD			x						
$\bar{A}BD$		(x)	x						
$\bar{B}\bar{C}\bar{D}$	(x)			x					

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

* MAP ENTERED VARIABLES:

Q1:	A	B	C	F	MEV	
	0	0	0	0	0	A \ B
	0	0	1	0	0	0 \ C
	0	1	0	1	1	1 \ X
	0	1	1	1		
	1	0	0	1	1	
	1	0	1	0		
	1	1	0	x	x	
	1	1	1	x		

Q2:	A	B	C	F	MEV	
	0	0	0	1	1	A \ B
	0	0	1	0	0	0 \ C
	0	1	0	0	1	1 \ 0
	0	1	1	1		
	1	0	0	0	1	
	1	0	1	0		
	1	1	0	1	1	
	1	1	1	1		

Q3:

$$g = A\bar{B}\bar{C}D + A\bar{B}\bar{C}\bar{D} + A\bar{B}CD + A\bar{B}C\bar{D} + \bar{A}B\bar{C}D + ABCD + ABC\bar{D} + AB\bar{C}D + ABC\bar{D}$$

$$g = A\bar{B}\bar{C}(D+\bar{D}) + A\bar{B}C(D+\bar{D}) + \bar{A}B\bar{C}(D) + ABC(D+\bar{D}) + AB\bar{C}(D+\bar{D})$$

$$= m_4(1) + m_5(1) + m_6(D) + m_7(1) + m_8(1)$$

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

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

AB	$\bar{c}\bar{d}$	$\bar{c}d$	$c\bar{d}$	cd
$\bar{A}\bar{B}$	0	0	0	0
$\bar{A}B$	0	1	0	0
AB	1	1	1	1
A \bar{B}	1	1	1	1

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

Q4: $f = \Sigma(0, 1, 3, 4, 5, 9, 11, 13, 15)$

AB	$\bar{c}\bar{d}$	00	01	11	10
00	1	1	1	0	
01	1	1	0	0	
11	0	1	1	0	
10	0	1	1	0	

$$f = \bar{a}\bar{c} + ad + \bar{b}d$$

$$f = \bar{a}\bar{b}\bar{c}\bar{d} + \bar{a}\bar{b}\bar{c}d + \bar{a}\bar{b}c\bar{d} + \bar{a}b\bar{c}\bar{d} + \bar{a}b\bar{c}d + ab\bar{c}\bar{d}$$

$$+ ab\bar{c}d + ab\bar{c}d + abc\bar{d}$$

$$f = \bar{a}\bar{b}\bar{c}(\bar{d}+d) + \bar{a}\bar{b}c(d) + \bar{a}b\bar{c}(\bar{d}+d) + ab\bar{c}(d)$$

$$+ a\bar{b}c(d) + ab\bar{c}(d) + abc(d)$$

Q5
Truth table

A	B	C	D	F	a	b	c	d	F
0	0	0	0	1	1	1	0	0	0
0	0	0	1	1	1	1	0	1	1
0	0	1	0	0	1	1	1	0	0
0	0	1	1	1	1	1	1	1	1
0	1	0	0	1	1	1	0	0	1
0	1	0	1	1	1	1	0	1	1
0	1	1	0	0	1	1	1	0	0
0	1	1	1	0	1	1	1	1	0
1	0	0	0	0	0	0	0	0	0
1	0	0	1	1	0	0	0	1	1
1	0	1	0	0	0	1	0	0	0
1	0	1	1	1	0	1	1	1	1
1	1	0	0	0	1	0	0	0	0
1	1	0	1	1	1	0	0	1	1
1	1	1	0	0	1	1	1	0	0
1	1	1	1	1	1	1	1	1	1

a	$\bar{b}c$	$b\bar{c}$	$\bar{b}c$	bc	$b\bar{c}$
\bar{a}	1	d	0	1	
a	d	d	d	d	d

$$f = \bar{a}\bar{c} + ad + \bar{b}d$$

Q5:

$$y = \text{TRUE} \Leftrightarrow (0, 1, 8, 9, 13, 15, 14)$$

ab	cd	00	01	11	10
00		1	1	1	0
01		0	0	0	1
11		0	0	1	1
10		1	1	0	0

$$y = \bar{b}\bar{c} + \bar{a}\bar{b}d + abc + bcd$$

Truth table

a	b	c	d	y
0	0	0	0	1
0	0	0	1	1
0	0	1	0	0
0	0	1	1	1
0	1	0	0	0
0	1	0	1	0
0	1	1	0	1
0	1	1	1	0
1	0	0	0	1
1	0	0	1	1
1	0	1	0	0
1	0	1	1	0
1	1	0	0	0
1	1	0	1	0
1	1	1	0	1
1	1	1	1	1

a	b	00	01	11	10
0		1	d	d	0
1		1	0	1	0

$$y = \bar{b}\bar{c} + abc + \bar{a}\bar{b}d + bcd$$

Q6: $\Sigma(2,3,4,5,13,15) + d(8,9,10,11)$

Sol: Truth table

a	b	c	d	y	MEN
0	0	0	0	0	0
0	0	0	1	0	
0	0	1	0	1	1
0	0	1	1	1	
0	1	0	0	1	1
0	1	0	1	1	
0	1	1	0	0	0
0	1	1	1	0	
1	0	0	0	x	x
1	0	0	1	x	
1	0	1	0	x	x
1	0	1	1	x	
1	1	0	0	0	d
1	1	0	1	1	
1	1	1	0	0	d
1	1	1	1	1	

a	b	c	00	01	11	10
0	0	1	0	1	0	1
1	x	x	d	d	d	d

$$y = ad + \bar{b}c + \bar{a}\bar{b}\bar{c}$$

ab	cd	00	01	11	10
00	0	0	1	1	
01	1	1	0	0	
11	0	1	1	0	
10	x	x	x	x	

$$y = ad + \bar{b}c + \bar{a}\bar{b}\bar{c}$$

Q7: Expression

$$\begin{aligned}
 & \bar{a}\bar{b}c\bar{d} + \bar{a}\bar{b}cd + \bar{a}b\bar{c}\bar{d} + \bar{a}b\bar{c}d + ab\bar{c}\bar{d} + abc\bar{d} \\
 & \quad + ab\bar{c}x + ab\bar{c}x + abcx + abcx \\
 = & \bar{a}\bar{b}\bar{c}(\bar{d}+d) + \bar{a}b\bar{c}(\bar{d}+d) + ab\bar{c}(d) + abc(d) - \\
 & \quad + ab\bar{c}(x) + ab\bar{c}(x) \\
 = & \bar{a}\bar{b}\bar{c}(1) + \bar{a}b\bar{c}(1) + ab\bar{c}(d) + abc(d) + ab\bar{c}(x) + ab\bar{c}(x)
 \end{aligned}$$

UNIT - 2

Design of Combinational Circuits

* Combinational circuits:

Output depends on only the present input irrespective of previous output. It does not have memory or feedback.

* Code converters:

Helps in reducing hardware complexity and hence reduces the power consumption.

Excess 3 : Encryption, security. ($BCD + 3$)

Gray code For high switching purposes (varies by one bit change)

ASCII : Every symbol is converted into 7-bit number

Conversion of BCD to Excess 3:

	BCD	Excess-3
0	0000	0011
1	0001	0100
2	0010	0101
3	0011	0110
4	0100	0111
5	0101	1000
6	0110	1001
7	0111	1010
8	1000	1011
9	1001	1100
10	1010	xxxx
11	1011	xxxx
12	1100	xxxx
13	1101	xxxx
14	1110	xxxx
15	1111	xxxx

BCD - Binary coded Decimal: 0-9

10-15 - Don't care condition since they are not BCD.

Gray code is usually used for complimenting operations.

To design a circuit:

	b_3, b_2	b_1, b_0	00	01	11	10	b_3, b_2	b_1, b_0	00	01	11	10
0	00	00	1	0	0	1	00	00	1	0	1	0
1	01	01	1	0	0	1	01	01	1	0	1	0
2	11	x	x	x	x	x	11	x	x	x	x	x
3	10	1	0	x	x	x	10	1	0	x	x	x
4	1111	xxxx										
5	1011	xxxx										
6	1101	xxxx										
7	1110	xxxx										
8	1111	xxxx										

$$e_0 = \bar{b}_0$$

$$e_1 = \bar{b}_1 \bar{b}_0 + b_1 b_0$$

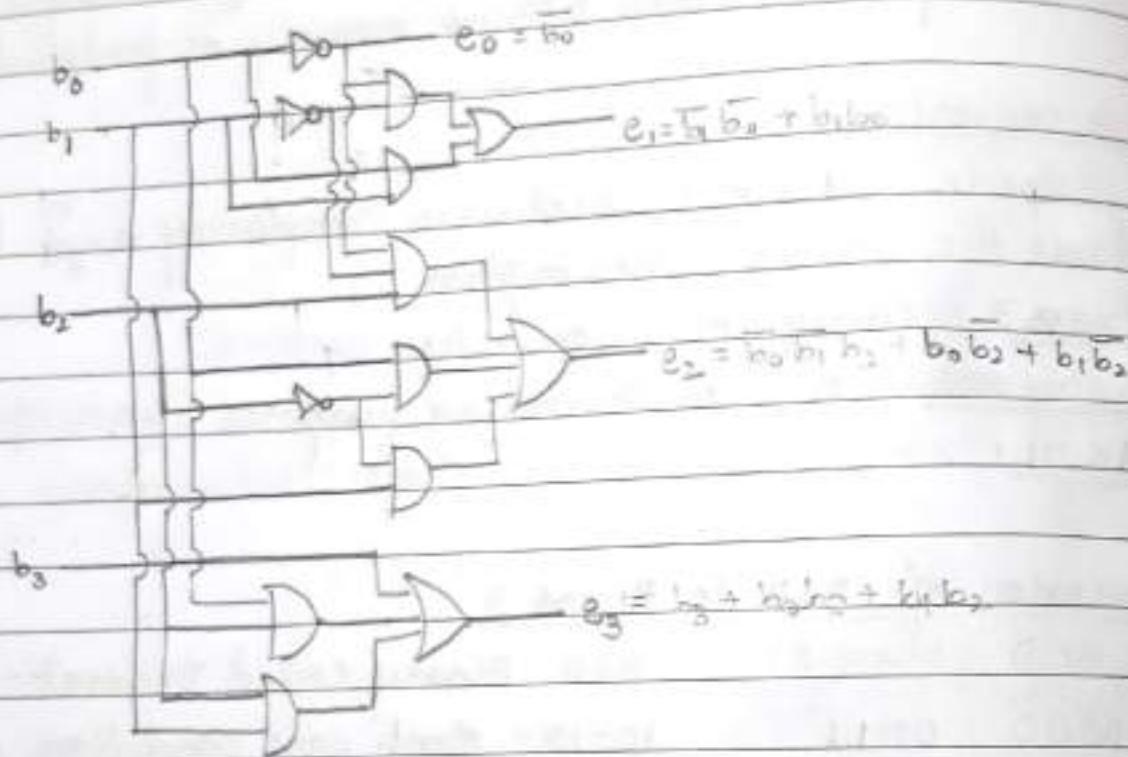
$$e_1 = b_0 \oplus b_1$$

$b_3\ b_2$	00	01	11	10
00	0	1	1	1
01	1	0	0	0
11	x	x	x	x
10	0	1	x	x

$$e_2 = \overline{b_0} \overline{b_1} b_2 + b_0 \overline{b_2} + b_1 \overline{b_2}$$

$b_3\ b_2$	00	01	11	10
00	0	0	0	0
01	0	1	1	1
11	x	x	x	x
10	1	1	x	x

$$e_3 = b_3 + b_0 b_2 + b_1 b_2$$



- Conversion of Binary to Gray code:

We can have a n -bit binary convertor.

Gray code it varies by one bit change.

$b_3\ b_2$	00	01	11	10
00	0	1	0	1
01	0	1	0	1
11	0	1	0	1
10	0	1	0	1

$b_3\ b_2$	00	01	11	10
00	0	0	1	1
01	1	1	0	0
11	1	1	0	0
10	0	0	1	1

$$g_0 = b_0 \overline{b}_1 + \overline{b}_0 b_1$$

$$g_0 = b_0 \oplus b_1$$

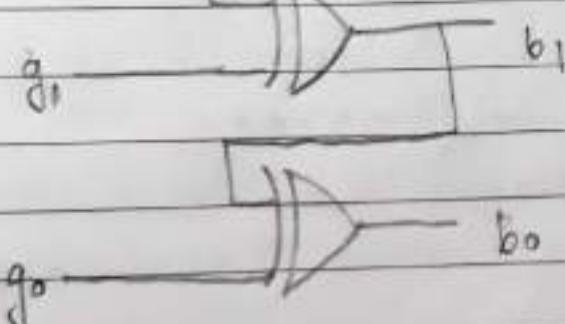
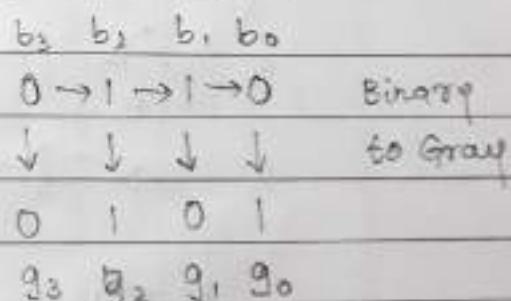
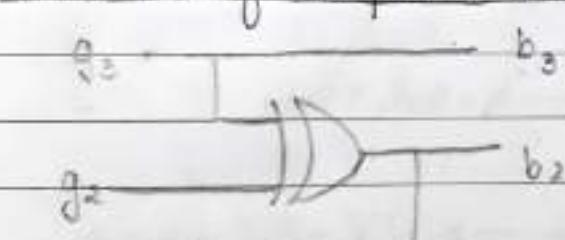
$$g_1 = \overline{b}_1 b_2 + b_1 \overline{b}_2$$

$$g_1 = b_1 \oplus b_2$$

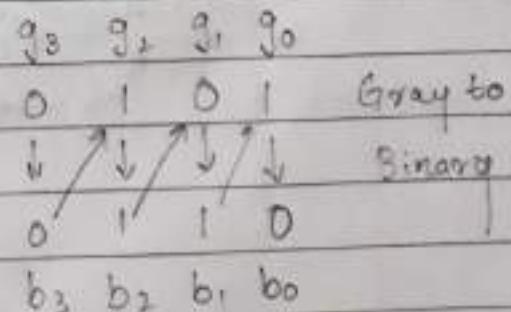
Binary	Gray code	$b_3 b_2 b_1 b_0$	00 01 11 10
$b_3 b_2 b_1 b_0$	$g_3 = b_3 \oplus b_2 \oplus b_1 \oplus b_0$	00	0 0 0 0
0 0 0 0	0 0 0 0	01	1 1 1 1
0 0 0 1	0 0 0 1	11	0 0 0 0
0 0 1 0	0 0 1 1	10	1 1 1 1
0 0 1 1	0 0 1 0		
0 1 0 0	0 1 1 0		
0 1 0 1	0 1 1 1		
0 1 1 0	0 1 0 1		
0 1 1 1	0 1 0 0		
1 0 0 0	1 1 0 0		
1 0 0 1	1 1 0 1	b_3	g_3
1 0 1 0	1 1 1 1	b_2	g_2
1 0 1 1	1 1 1 0	b_1	g_1
1 1 0 0	1 0 1 0	b_0	g_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		

- conversion of Gray code to binary:

Binary to gray code



Gray to binary



- Conversion of Excess-3 to BCD

Excess-3	BCD	$e_3 e_2$	$e_1 e_0$	$b_3 b_2 b_1 b_0$	$b_3 b_2 b_1 b_0$	$e_3 e_2$	$e_1 e_0$	$b_3 b_2 b_1 b_0$
0 0000	0000	$x \ x \ x \ x$	$x \ x \ x \ x$	00 00 01 11 10	00 00 01 11 10	$x \ x \ x \ x$	$x \ x \ x \ x$	00 00 01 11 10
1 0001	xxxx	00 00 00 00	00 00 00 00	01 00 00 00	01 00 00 00	00 00 00 00	00 00 00 00	01 00 00 00
2 0010	xxxx	01 00 00 00	01 00 00 00	11 1 x x x x	11 1 x x x x	01 00 00 00	01 00 00 00	11 0 x x x x
3 0011	0000	11 0 0 0 1 0	11 0 0 0 1 0	10 0 0 1 0	10 0 0 1 0	11 0 x x x x	11 0 x x x x	10 1 1 0 !
4 0100	0001	10 0 0 1 0	10 0 0 1 0	00 0 0 0 1	00 0 0 0 1	10 1 1 0	10 1 1 0	01 0 0 1 0
5 0101	0010	00 0 0 1 0	00 0 0 1 0	00 0 0 1 0	00 0 0 1 0	00 0 0 1 0	00 0 0 1 0	00 0 0 1 0
6 0110	0011	00 0 1 0 0	00 0 1 0 0	00 0 1 0 0	00 0 1 0 0	00 0 1 0 0	00 0 1 0 0	00 0 1 0 0
7 0111	0100	00 0 1 0 0	00 0 1 0 0	00 0 1 0 0	00 0 1 0 0	00 0 1 0 0	00 0 1 0 0	00 0 1 0 0
8 1000	0101	01 0 1 0 1	01 0 1 0 1	01 0 1 0 1	01 0 1 0 1	01 1 0 0 1	01 1 0 0 1	01 1 0 0 1
9 1001	0110	11 0 x x x x	11 0 x x x x	11 0 x x x x	11 0 x x x x	11 1 x x x x	11 1 x x x x	11 1 x x x x
10 1010	0111	10 0 1 0 1	10 0 1 0 1	10 0 1 0 1	10 0 1 0 1	10 1 0 0 1	10 1 0 0 1	10 1 0 0 1
11 1011	1000	10 0 1 0 1	10 0 1 0 1	10 0 1 0 1	10 0 1 0 1	10 1 0 0 1	10 1 0 0 1	10 1 0 0 1
12 1100	1001	00 1 0 0 1	00 1 0 0 1	00 1 0 0 1	00 1 0 0 1	00 1 0 0 1	00 1 0 0 1	00 1 0 0 1
13 1101	xxxx	00 1 0 0 1	00 1 0 0 1	00 1 0 0 1	00 1 0 0 1	00 1 0 0 1	00 1 0 0 1	00 1 0 0 1
14 1110	xxxx	00 0 1 0 0	00 0 1 0 0	00 0 1 0 0	00 0 1 0 0	00 0 1 0 0	00 0 1 0 0	00 0 1 0 0
15 1111	xxxx	00 0 1 0 0	00 0 1 0 0	00 0 1 0 0	00 0 1 0 0	00 0 1 0 0	00 0 1 0 0	00 0 1 0 0

$$b_3 = e_2 e_3 + e_0 e_1 e_3$$

$$b_2 = \bar{e}_1 \bar{e}_2 + \bar{e}_0 \bar{e}_2 + \bar{e}_0 e_1$$

$$b_1 = e_0 \bar{e}_1 + \bar{e}_0 e_1$$

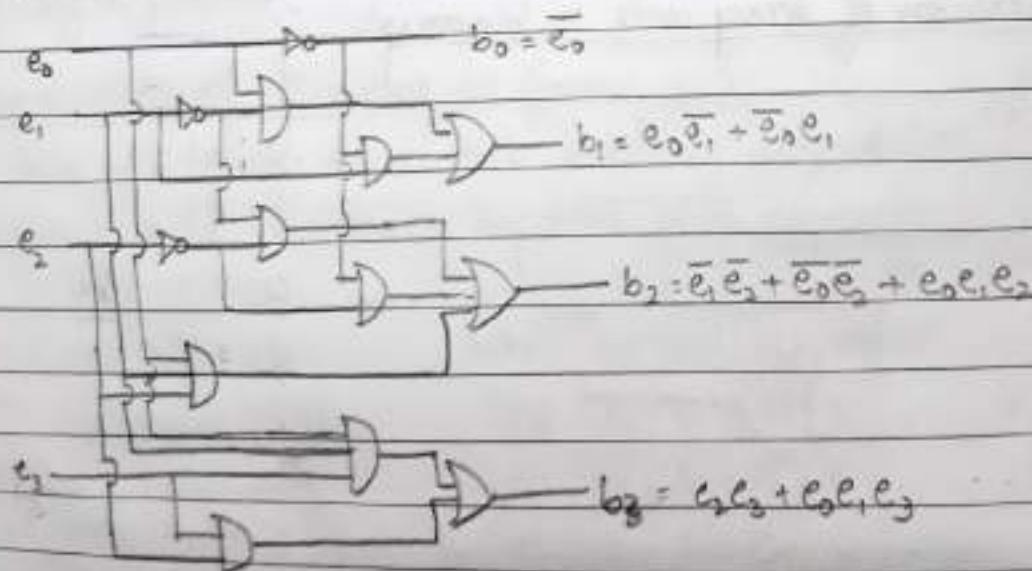
$$b_0 = \bar{e}_0$$

$$b_1 = e_0 \bar{e}_1 + \bar{e}_0 e_1$$

$$b_0 = \bar{e}_0$$

$$b_2 = \bar{e}_1 \bar{e}_2 + \bar{e}_0 \bar{e}_2 + e_0 e_1 e_2$$

$$b_3 = e_2 e_3 + e_0 e_1 e_3$$



* Binary Adder and Subtractor:

- HALF ADDER:

- Usually not preferred
- cascading of half adders is not possible since the carry from one stage cannot be involved in next stage.

- FULL ADDER:

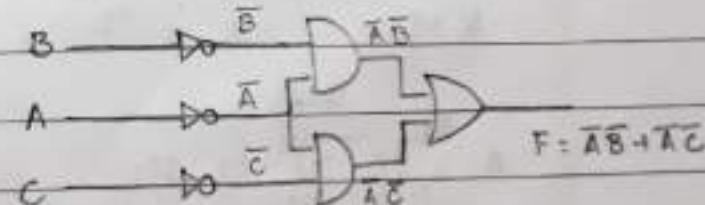
- cascading of full adders: First stage full adder always has Cin as zero. The carry of one stage is given as the one of the input to the next stage.

Design the combination circuit with 3 input and one output. i] The output is 1 when the binary value of the input is less than three, the output is 0 otherwise.

Truth table:

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

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



ii] The output is 1 when the binary value of the input is an even number.

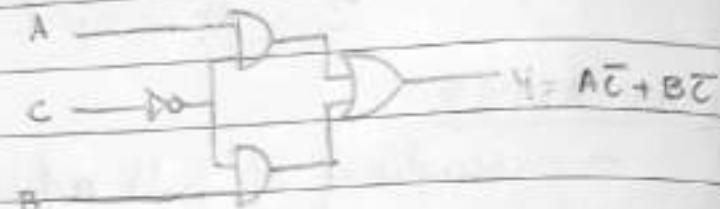
Truth table

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

K-Map

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

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



* FULL ADDER:

Truth table

A	B	Cin	S	Cout
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

Block Diagram



K Map

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

$$S = \bar{A}\bar{B}C_{in} + \bar{A}B\bar{C}_{in} + A\bar{B}\bar{C}_{in} + ABC_{in}$$

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

$$S = \bar{A} \oplus B \oplus C_{in}$$

A	Carry			
	00	01	11	10
0	0	0	1	0
1	0	1	1	1

$$Cout = AB + AC + BC$$

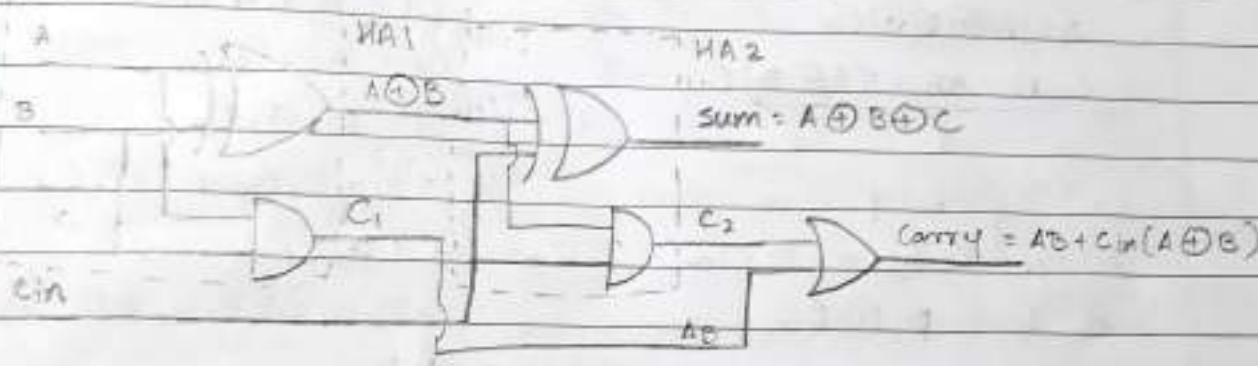
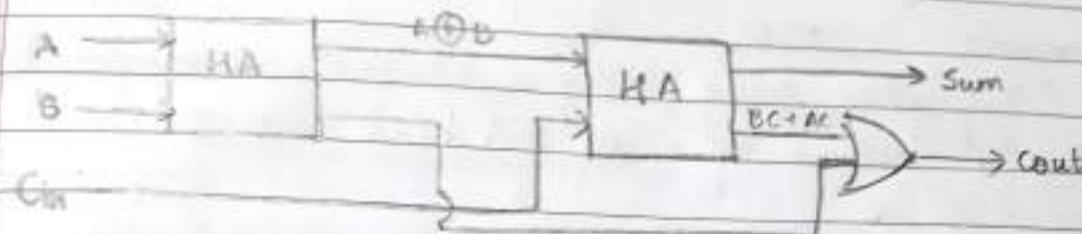
Carry with XOR

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

$$\begin{aligned} C_{out} &= BC_{in} + A\bar{B}C_{in} + ABC_{in} \\ C_{out} &= BC_{in} + A(BC_{in} + B\bar{C}_{in}) \\ C_{out} &= BC + A(B \oplus C) \end{aligned}$$

6a

Full adder using Half Adders:



- Sign magnitude conversion (manual)

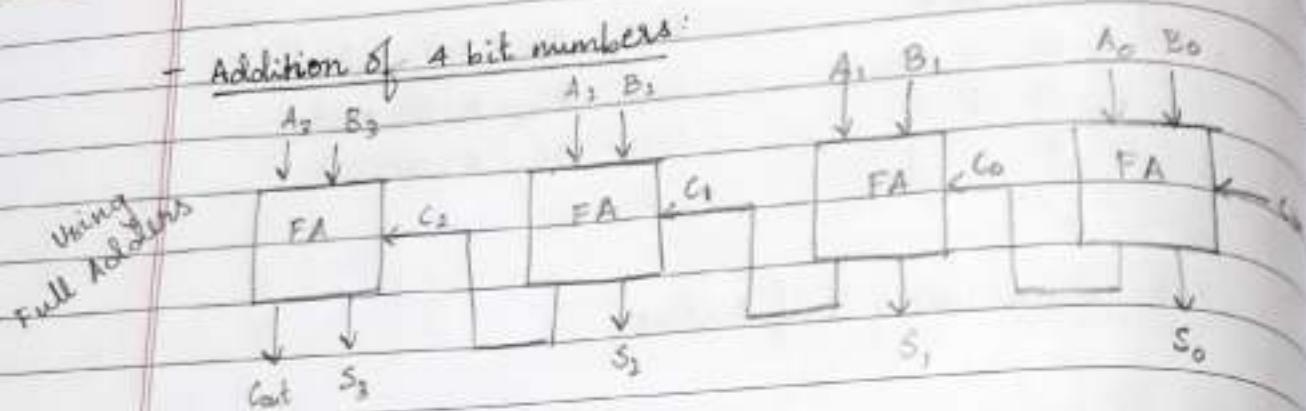
Here the signed number can be misunderstood as its binary equivalent.

i.e., $-9 \Rightarrow 11001$ which is also 25.

- Sign complementing conversion

$$\begin{array}{r}
 -8 \quad 11000 \\
 + -7 \quad + 11001 \\
 \hline
 -15 \quad 110001
 \end{array}
 \qquad
 \begin{array}{r}
 01000 \quad 00111 \\
 \Rightarrow 10111 \quad \Rightarrow 11000 \\
 + 1 \quad + 1 \\
 \hline
 1000 \quad 11001
 \end{array}$$

↓
2's complement



Demerits

- Propagation delay (carry is delayed when compared to sum).
- This is overcome by carry lookahead adders.

$$S_i = A_i \oplus B_i \oplus C_{in}$$

$$C_{out} = AB_i + (A_i \oplus B_i) C_{in}$$

Carry generator *Carry propagator*

$$\therefore C_{out} = G_i + P_i C_{in}$$

$$\text{and } S_i = P_i \oplus C_{in}$$

$$C_i = G_i + P_i C_{i-1}$$

$$C_0 = G_0 + P_0 C_{in}$$

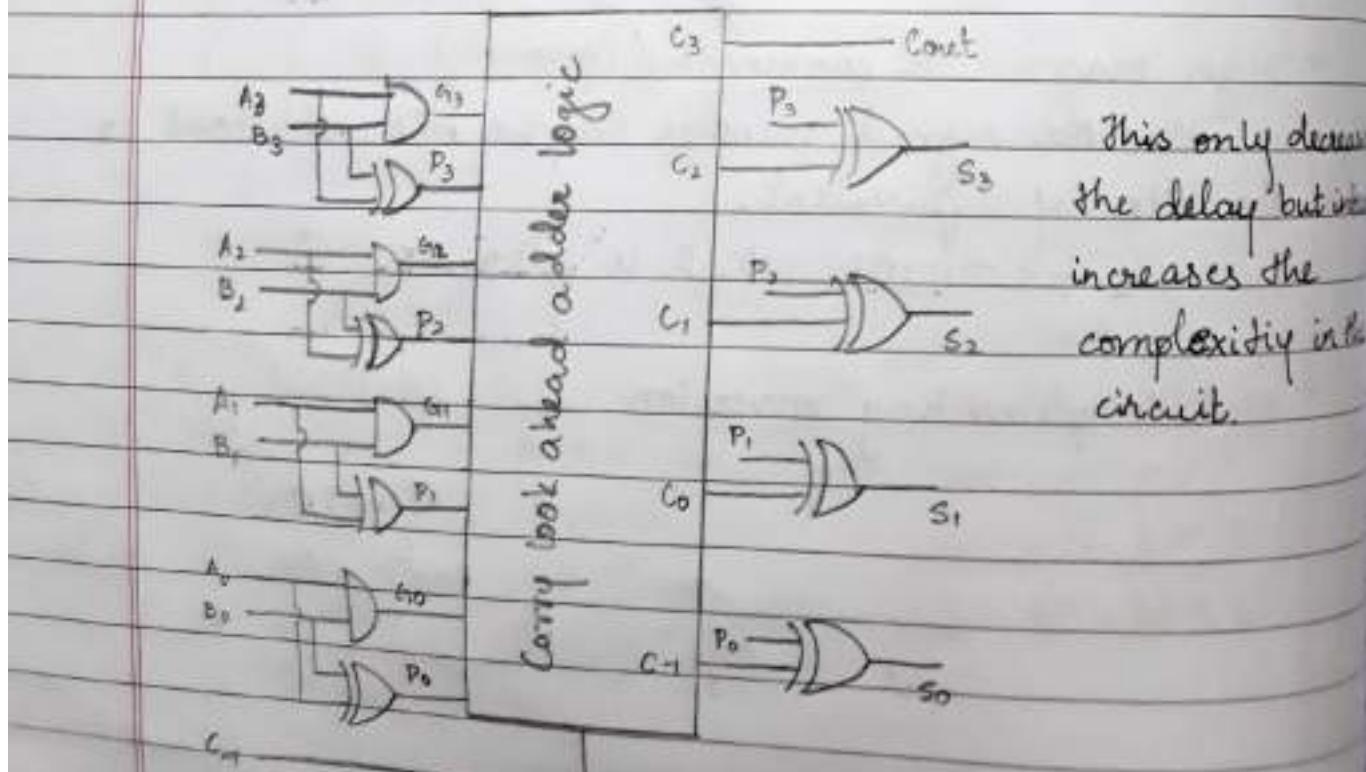
$$C_1 = G_1 + P_1 C_0$$

$$\therefore C_1 = G_1 + P_1 G_0 + P_0 P_1 C_{in}$$

$$C_2 = G_2 + P_2 C_1$$

$$C_3 = G_3 + P_3 G_2 + P_2 P_3 G_1 + P_1 P_2 G_0 + P_0 P_1 P_2 C_{in}$$

$$C_{out} = G_3 + P_3 C_2$$



* Fan in and Fan out:

Fan in - The number of inputs that can be taken without changing its functionality.

Fan out - The number of outputs that can be given without changing its functionality.

* HALF SUBTRACTOR:

Truth table

A Bin	B Bin	D	Bout
0 0	0 0	0	0
0 1	1 1	1	1
1 0	0 1	1	0
1 1	1 0	0	0

K Map:

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

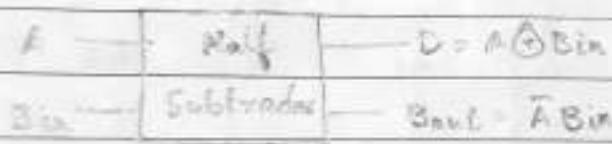
A	B	0	1
0	0	0	1
1	0	1	0

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

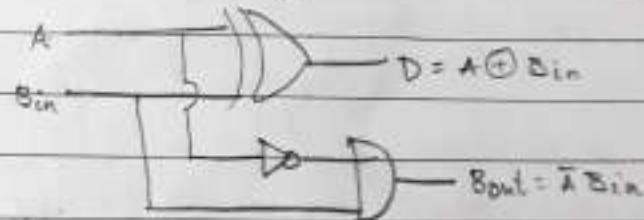
$$B_{out} = \bar{A}B_{in}$$

$$B_{out} = A\bar{B}_{in}$$

Block Diagram



Circuit Diagram



* FULL SUBTRACTOR:

Truth table

A	B	C	D	Bout
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

K-Map:

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

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

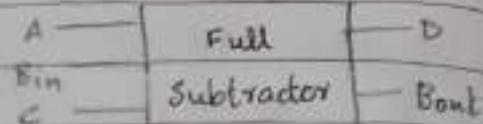
$$D = \bar{A}(\bar{B}C + B\bar{C}) + A(\bar{B}C + BC)$$

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

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

Block diagram

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



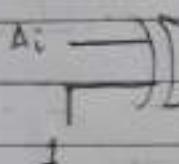
$$\text{Bout} = \overline{A}C + \overline{A}B + BC$$

constructing subtractor using adder only:

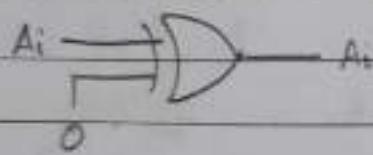
This is done by controlled complementing using XOR.

When $A_i = 0$ Output: $\overline{A}_i = 1$

When $A_i = 1$ Output = 0 = \overline{A}_i



Subtractor

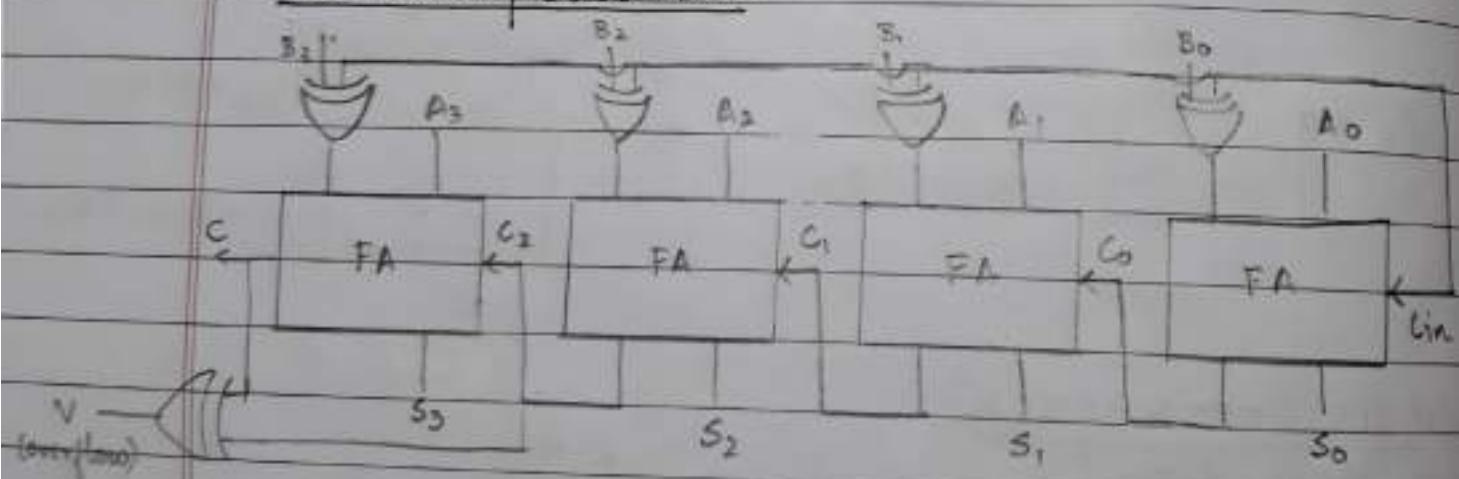


When $A_i = 0$ Output = 0 = A_i

When $A_i = 1$ Output = 1 = A_i

Adder

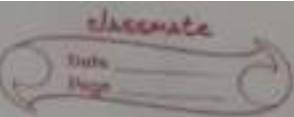
4 bit adder / Subtractor:



$Cin = 0$ acts as adder

$Cout = 1$ acts as subtractor

overflow flag is set only when there is carry, that is when an extra bit is generated.



* BCD ADDITION:

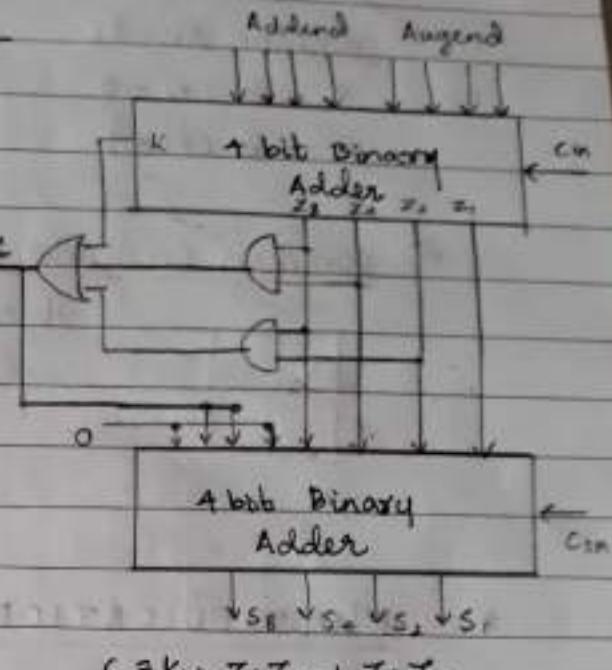
Truth table :

Binary Sum

K Z ₈ Z ₄ Z ₂ Z ₁	Binary Sum
0 0 0 0 0 0	0 0 0 0 0 0
0 0 0 0 0 1	0 0 0 0 0 1
0 0 0 1 0 0	0 0 0 1 0 c
0 0 0 1 0 1	0 0 0 1 1
0 0 1 0 0 0	0 0 1 0 0
0 0 1 0 0 1	0 0 1 0 1
0 0 1 1 0 0	0 0 1 1 0
0 0 1 1 0 1	0 0 1 1 1
0 1 0 0 0 0	0 1 0 0 0
0 1 0 0 0 1	0 1 0 0 1
0 1 0 1 0 0	1 0 0 0 0
0 1 0 1 0 1	1 0 0 0 1
0 1 1 0 0 0	1 0 0 1 0
0 1 1 0 0 1	1 0 0 1 1
0 1 1 1 0 0	1 0 1 0 0
0 1 1 1 0 1	1 0 1 0 1
1 0 0 0 0 0	1 0 1 1 0
1 0 0 0 0 1	1 0 1 1 1
1 0 0 1 0 0	1 1 0 0 0
1 0 0 1 0 1	1 1 0 0 1

BCD Sum

C S₈ S₄ S₂ S₁



4 bit Binary Adder

$$C = K + Z_8 Z_4 + Z_8 Z_2$$

- When C = 1, the number is greater than 9 than 0110 i.e., 6 is added.
- When C = 0, the number is less than 9 than 0000 i.e., 0 is added.

* SEVEN SEGMENT DISPLAY:

- Common Anode
 - \Rightarrow To be ON = 1
 - \Rightarrow To be OFF = 0

- Common Cathode
 - \Rightarrow To be ON = 0
 - \Rightarrow To be OFF = 1

\Rightarrow common cathode seven segment display:

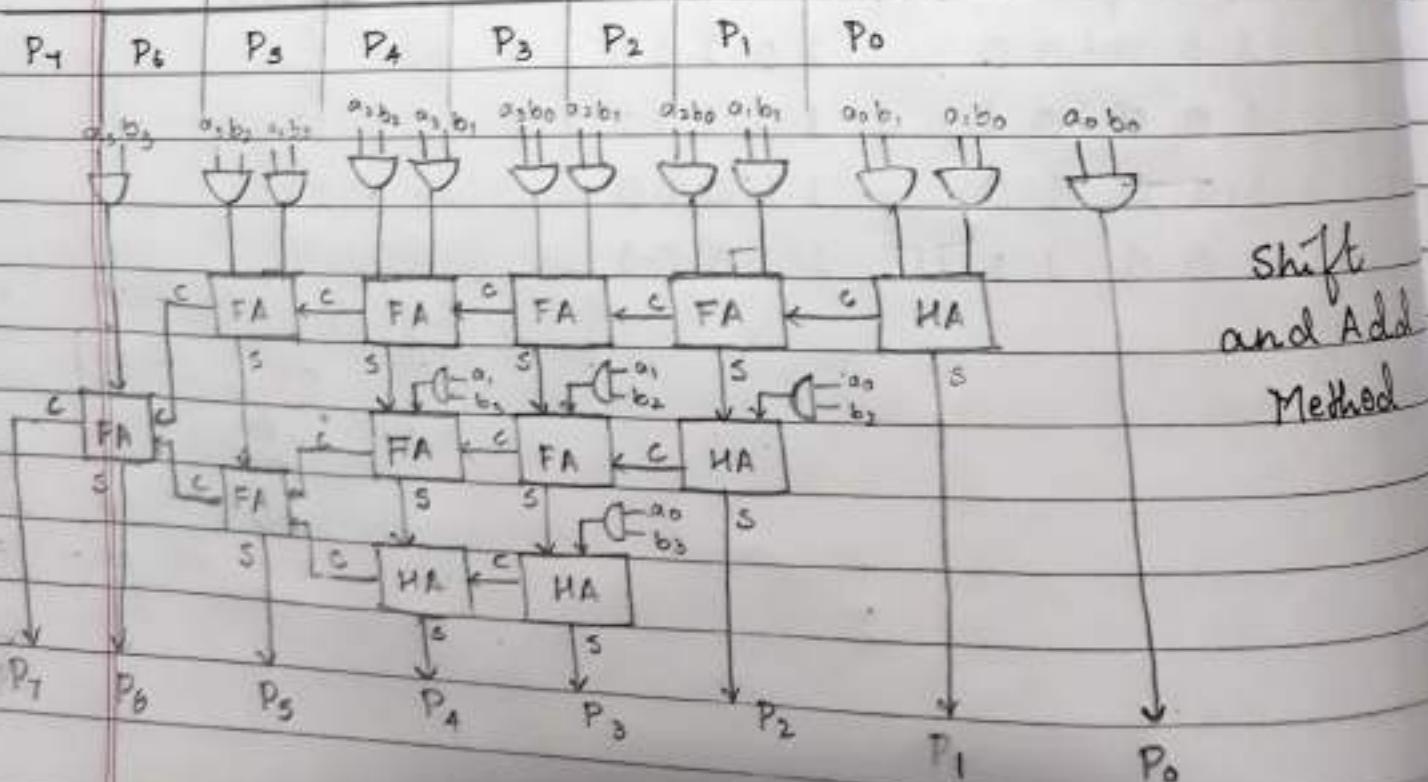


It requires two seven segment displays to display BCD sum for 4 bit addition.

* MULTIPLICATION:

$$a_3 a_2 a_1 a_0 \times b_3 b_2 b_1 b_0$$

$a_3 b_0$	$a_2 b_0$	$a_1 b_0$	$a_0 b_0$	4 bits \Rightarrow 16 AND
$a_3 b_1$	$a_2 b_1$	$a_1 b_1$	$a_0 b_1$	8 Full Adders
$a_3 b_2$	$a_2 b_2$	$a_1 b_2$	$a_0 b_2$	A Half Adder
$a_3 b_3$	$a_2 b_3$	$a_1 b_3$	$a_0 b_3$	



Magnitude comparator:

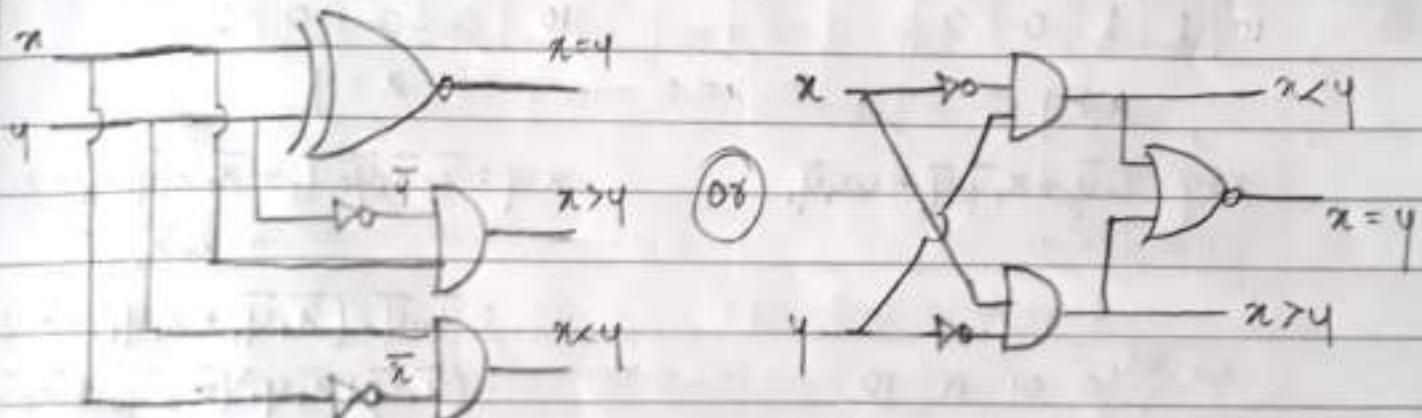
- 1 bit comparator

x	y	$x > y$	$x = y$	$x < y$
0	0	0	1	0
0	1	0	0	1
1	0	1	0	0
1	1	0	1	0

$$x > y : x \bar{y}$$

$$\begin{aligned} x = y &= \bar{x} \bar{y} + x \bar{y} \\ &= \bar{x} \oplus y \end{aligned}$$

$$x < y = \bar{x} y$$



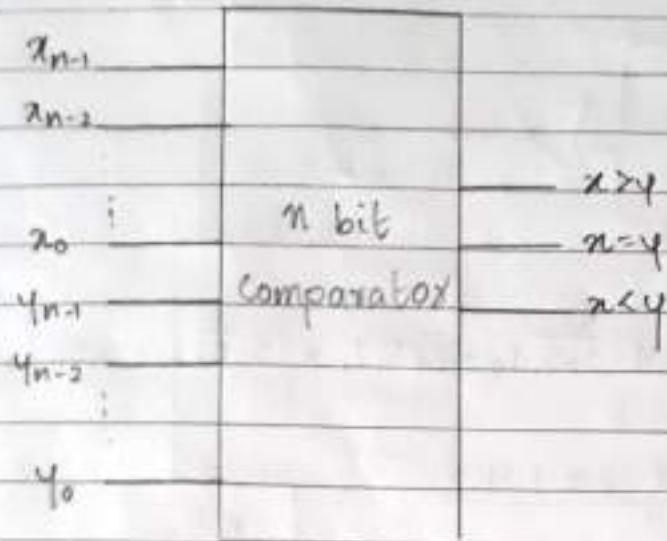
- 2 bit comparator

$x_0 x_1$	$y_0 y_1$	$x > y$	$x = y$	$x < y$
0 0	0 0	0	1	0
0 0	0 1	0	0	1
0 0	1 0	0	0	1
0 0	1 1	0	0	1
0 1	0 0	1	0	0
0 1	0 1	0	1	0

$x_0 x_1$	$y_0 y_1$	$\pi \geq 4$	$\pi = 4$	$\pi < 4$
0 1	1 0	0	0	1
0 1	1 1	0	0	1
1 0	0 0	1	0	0
1 0	0 1	1	0	0
1 0	1 0	0	1	0
1 0	1 1	0	0	1
1 1	0 0	1	0	0
1 1	0 1	1	0	0
1 1	1 0	1	0	0
1 1	1 1	0	1	0

$$\begin{aligned} \pi \geq 4 &= \bar{x}_0 x_1 \bar{y}_0 \bar{y}_1 + x_0 \bar{x}_1 \bar{y}_0 \bar{y}_1 + x_0 x_1 \bar{y}_0 \bar{y}_1 + x_0 x_1 y_0 \bar{y}_1 \\ &\quad + x_0 x_1 y_0 y_1 + x_0 x_1 \bar{y}_0 y_1 \\ &= \bar{y}_0 \bar{y}_1 (x_0 \oplus x_1) + x_0 x_1 (y_0 \oplus y_1) - y_0 x_1 \bar{y}_0 \bar{y}_1 + y_0 x_1 y_0 \bar{y}_1 \end{aligned}$$

Block diagram of an n -bit comparator



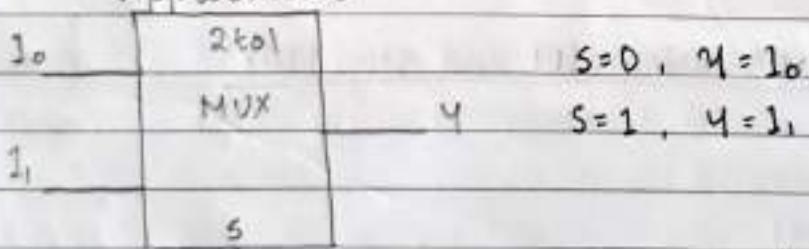
* MUXPLEXERS (Selector)

When there is only one transmission line but multiple user, only one transmission can take place at a time then a multiplexer is used to select the lines.

n - Selection line m to 1 MUX

m - input lines $2^n = m$

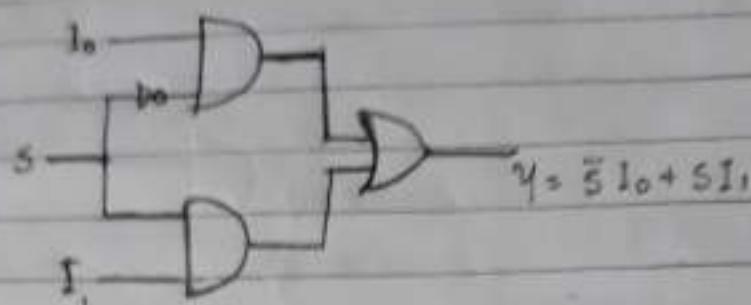
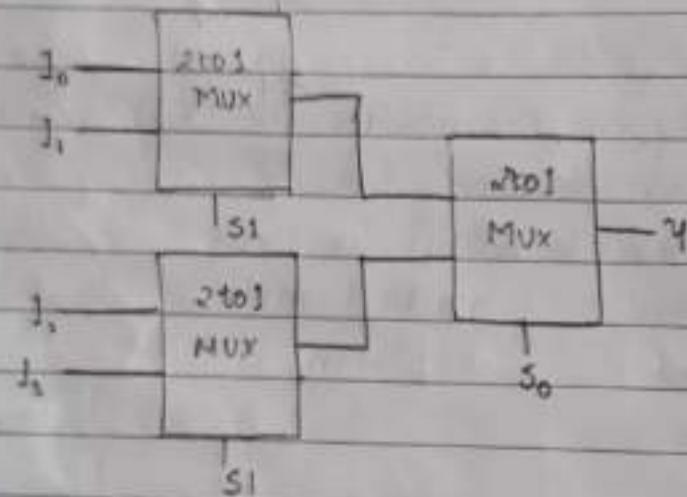
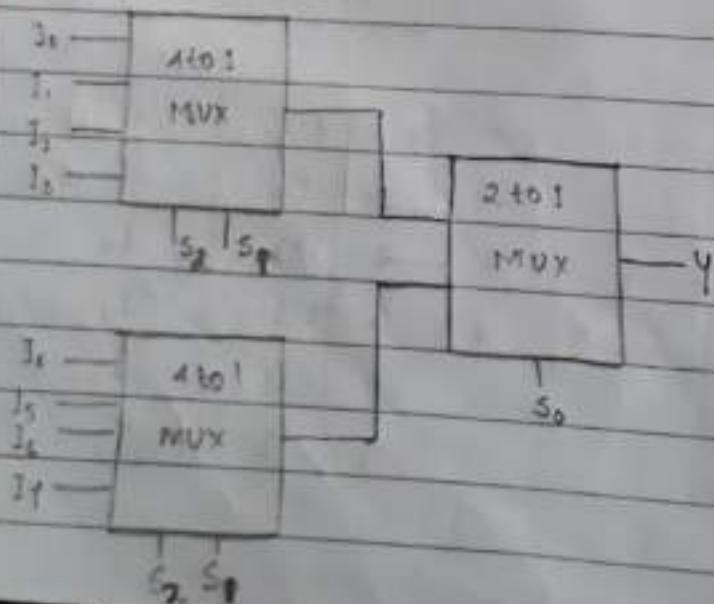
switching
Application

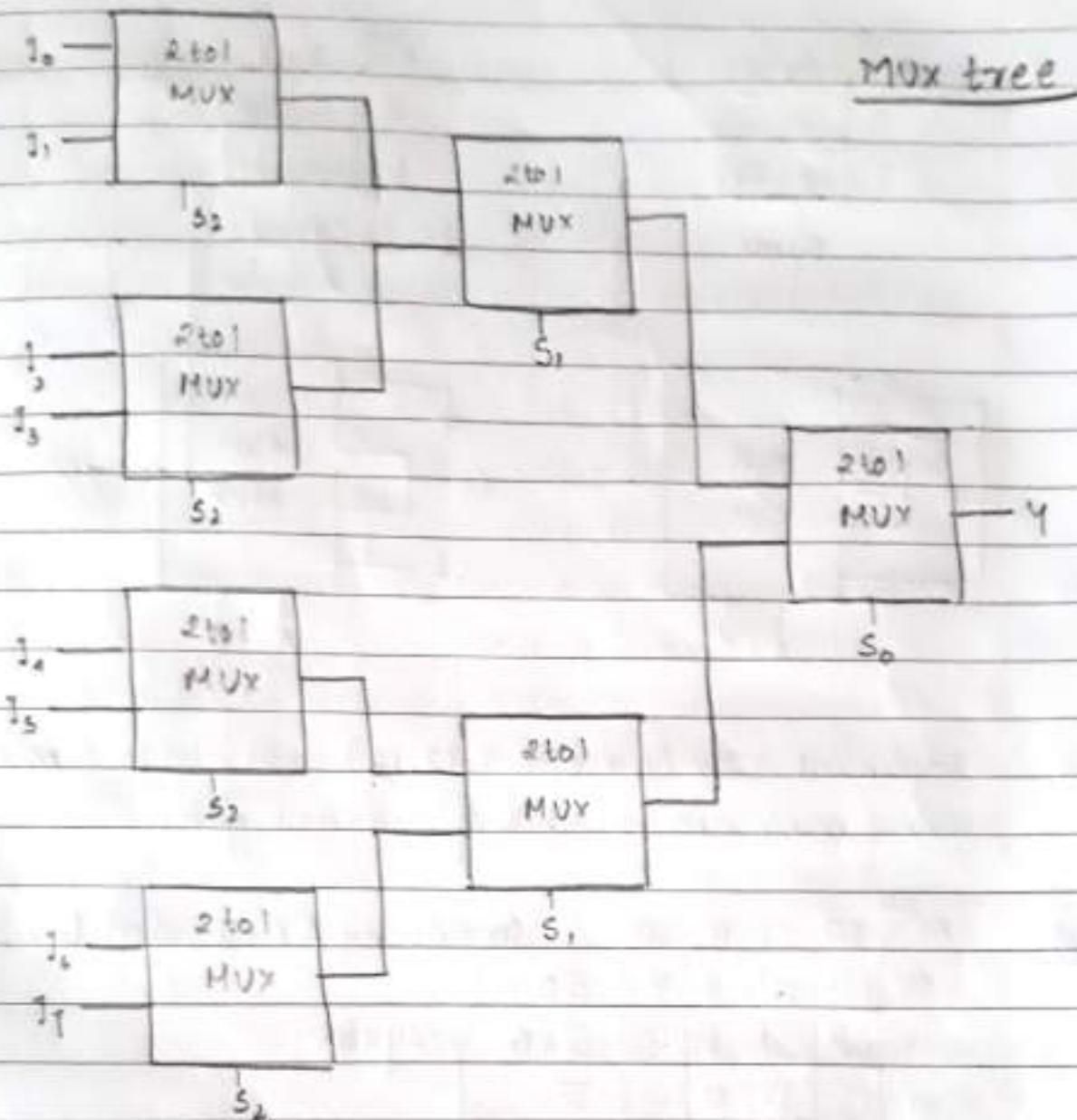


		select line	Y
		$S_1 \quad S_0$	
I_0	4 to 1 MUX	0 0	I_0
I_1	4 to 1 MUX	0 1	I_1
I_2		1 0	I_2
I_3	$S_1 \quad S_0$	1 1	I_3

2 to 1 MUX

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

4 to 1 MUX $Y = \bar{S}_0 \bar{S}_1 I_0 + \bar{S}_0 S_1 I_1 + S_0 \bar{S}_1 I_2 + S_0 S_1 I_3$ 4 to 1 MUX using 2 to 1 MUX:8-to-1 MUX using 2to1 and 4to1 MUX:



Full Adder using 4 to 1 MUX:

Sum:

A	B	Cin	S	MEX
0	0	0	0	C_{in}
0	0	1	1	
0	1	0	1	\bar{C}_{in}
0	1	1	0	\bar{C}_{in}
1	0	0	1	\bar{C}_{in}
1	0	1	0	
1	1	0	0	C_{in}
1	1	1	1	$\bullet 1$

Carry:

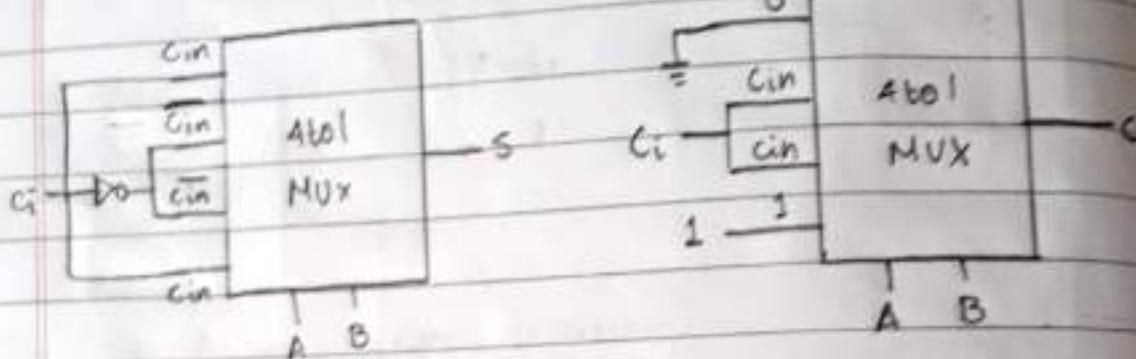
A	B	Cin	C	MEX
0	0	0	0	0
0	0	1	0	
0	1	0	0	C_{in}
0	1	1	1	
1	0	0	0	C_{in}
1	0	1	1	
1	1	0	1	$\bullet 1$
1	1	1	1	1

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

Sum

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

Carry



Q1: Implement $\Sigma_m(1, 4, 5, 7, 9, 12, 13)$ using 4-to-1 MUX using minimum number of external gates.

sol:

AB	CD	00	01	11	10	Considering AB as select line
00	00	0	1	0	0	$\bar{C}D$
01	01	1	1	1	0	$\bar{C} + D$
11	11	1	1	0	0	\bar{C}
10	10	0	1	0	0	$\bar{C}D$

considering CD as select line

AB	CD	00	01	11	10	
00	00	0	1	0	0	
01	01	1	1	1	0	
11	11	1	1	0	0	
10	10	0	1	0	0	

gates

2-to-1 MUX

4-to-1 MUX

AB

* Three State Gates:

0 - Low state

1 - High state

Σ - High Impedance state

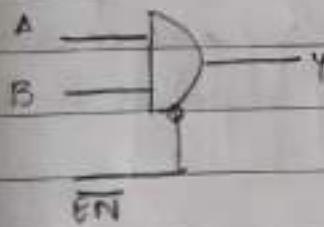
(Physical connected but not electrically connected)



Active high

$$\overline{EN} = 1, Y = A$$

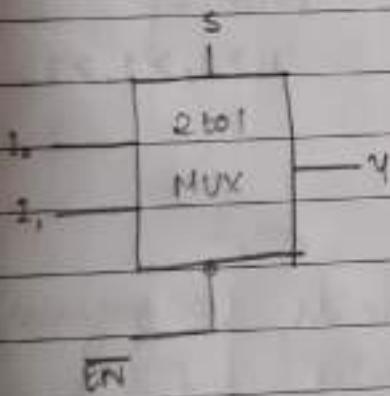
$$\overline{EN} = 0, Y = \Sigma$$



Active low

$$\overline{EN} = 0, Y = AB$$

$$\overline{EN} = 1, Y = \Sigma$$



Active low

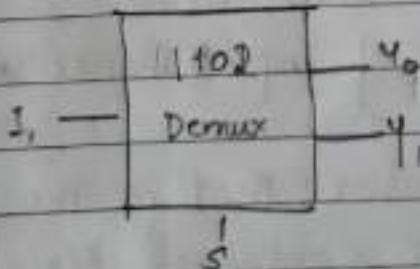
$$\overline{EN} = 0 \quad \text{MUX is active}$$

$$\overline{EN} = 1 \quad \text{MUX is inactive} \Rightarrow \Sigma$$

* DEMUX:

One input multiple outputs

Distributor - input distributed as many outputs.



* Encoders:

1 to 2 Encoder



Encoder has 2^n input lines
n output lines.

I_3	I_2	I_1	I_0	Y_1	Y_0
			1	0	0
			0	1	0
			0	0	1
			0	0	0

Q2: An encoder has 2^n input lines and n output lines if more than one input is there then they are considered based on the priority and priority encoder is designed.

Priority Encoder

I_3	I_2	I_1	I_0	Y_1	Y_0
1	x	x	x	1	1
0	1	x	x	1	0
0	0	1	x	0	1
0	0	0	1	0	0

Priority

$I_3 > I_2 > I_1 > I_0$

- NOTE:

MSIC - Medium Scale Integrated circuits. (100 transistors)

- to perform particular task only

- non programmable

LSIC - Large Scale Integrated circuits (1000 transistors)

- programmable

MUX:

- 2^n input lines ; $m = 2^m$ select lines

- output is same as that of input hence the data is transferred as it is.

- Parallel to serial convertor ; that is the input data is given parallelly and output is obtained serially.

- Output is based on the select lines
- Applications: communication

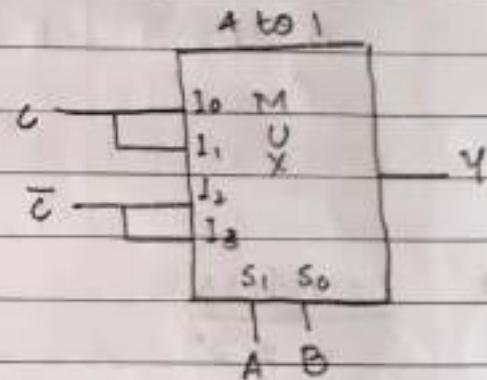
ENCODER:

- 2^n input lines; n output line
- Output is coded form of input
- Output is compressed form of input.
- No select line is used
- Applications: coders.

Q1: Write a logic realised by circuit shown.

Sol: Truth Table :

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



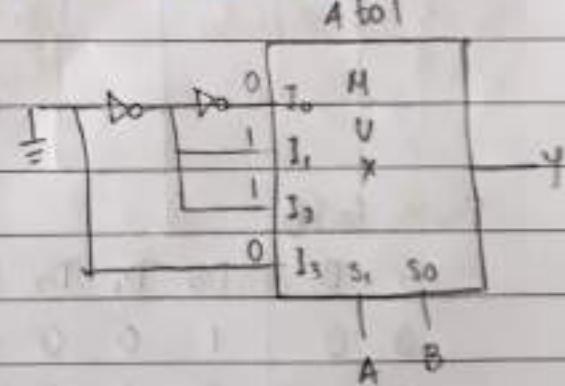
$$Y = \bar{A}C + A\bar{C}$$

$$\underline{Y = A \oplus C}$$

Q2: write a logic realised by circuit shown.

Sol: Truth Table :

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



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

$$\underline{Y = A \oplus B}$$

* Map entered variables using don't cares:

Q1.

A	B	C	F	
0	0	0	$x \bar{x} + c$	
0	0	1	\bar{x}	
0	1	0	$x \bar{x} + c$	
0	1	1	\bar{x}	
1	0	0	1	
1	0	1	1	
1	1	0	c	
1	1	1	1	

A B 0 1

0	$\bar{x} + c$	$\bar{x} + c$
1	1	c

when $x = 0$

A B 0 1

0	c	c
1	1	c

$A\bar{B} + c$

when $x = 1$

A B 0 1

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

$\bar{A}\bar{B} + c$ by \bar{B}

Main min term v. given

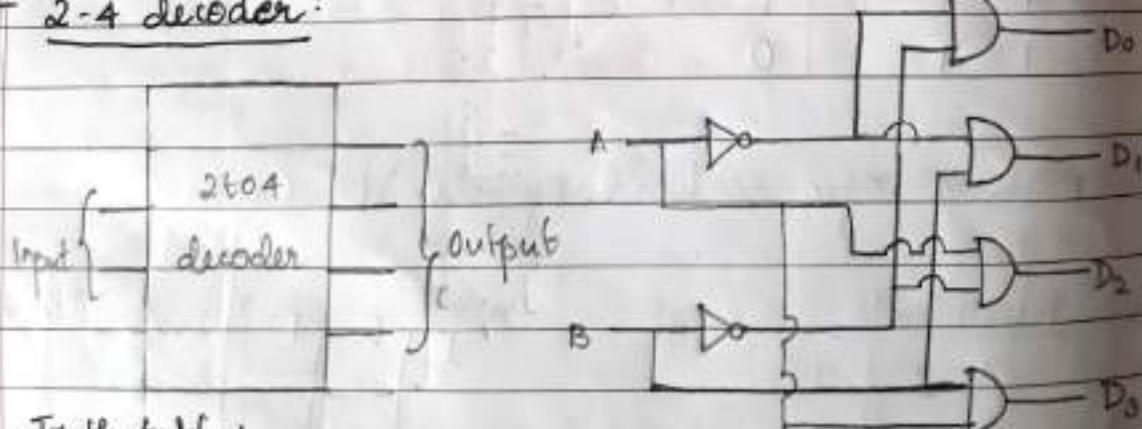
* Decoders:

$$m \leq 2^n$$

n - number of input lines

m - output lines.

- 2-4 decoder:



Truth table:

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

Q1:

Implement a full adder with a decoder.

Sol:

Full adder

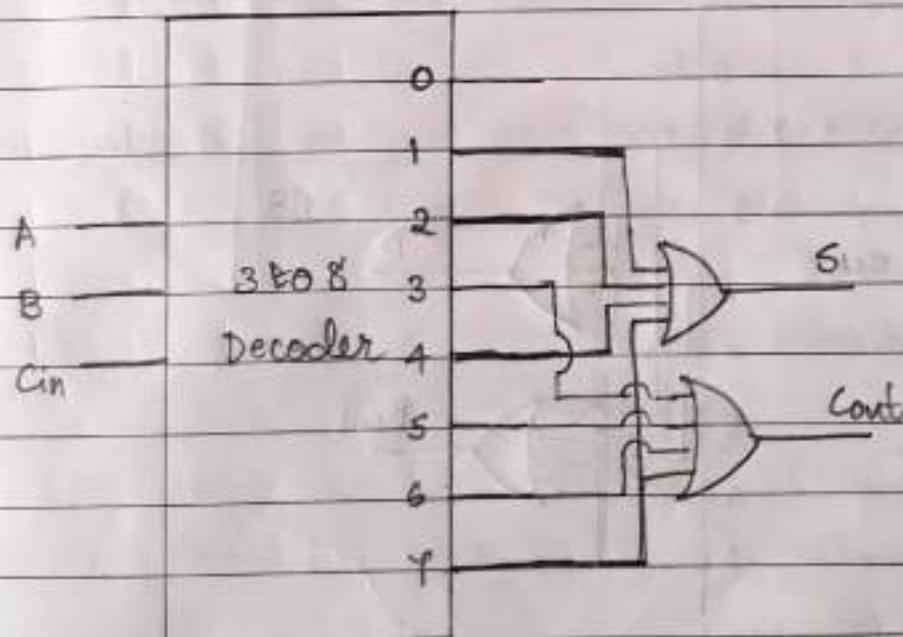
Input = 3 Hence 8 output lines.

Truth table

	A	B	Cin	S	Cout
0:	0	0	0	0	0
1:	0	0	1	1	0
2:	0	1	0	1	0
3:	0	1	1	0	1
4:	1	0	0	1	0
5:	1	0	1	0	1
6:	1	1	0	0	1
7:	1	1	1	1	1

$$S = \sum m(1, 2, 4, 7)$$

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



Q.2: Implement a full subtractor using decoder

Sol: Full subtractor

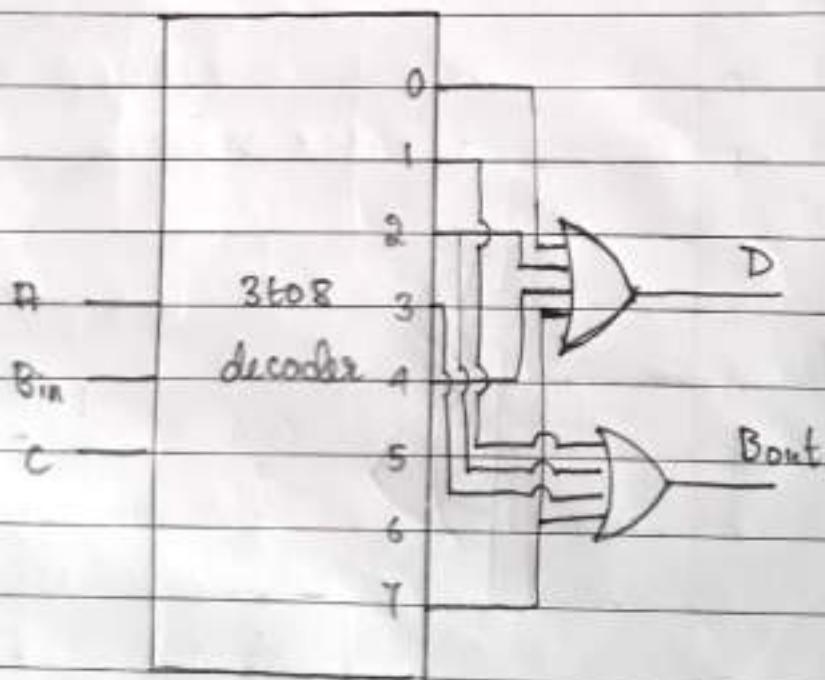
3 inputs \Rightarrow 8 output lines.

Truth table:

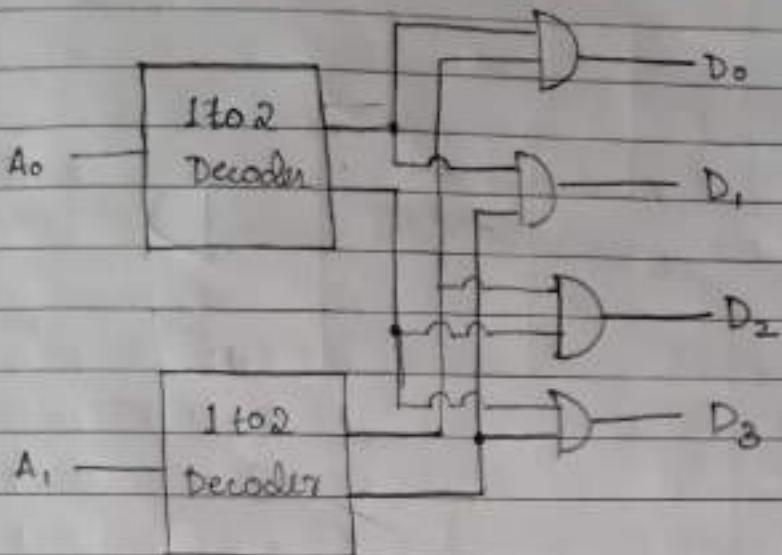
A	B _{in}	C	D	B _{out}
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	0	1
1	0	0	1	0
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1

$$\textcircled{2} \quad D = \sum m(1, 2, 4, 7)$$

$$B_{out} = \sum m(1, 2, 3, 7)$$

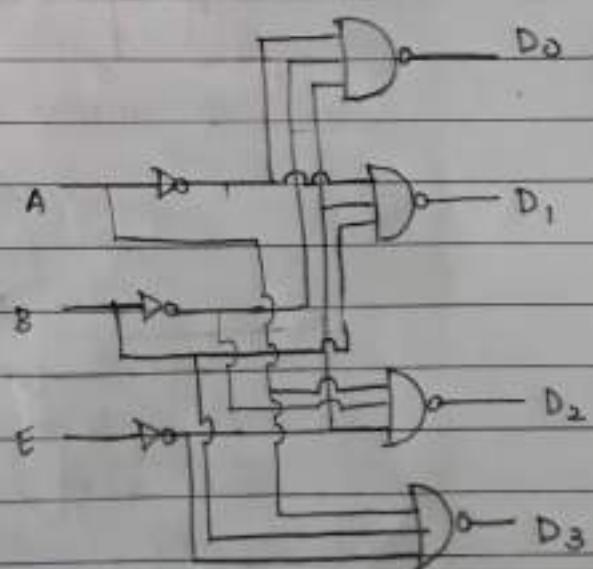


Q3: Find the minimum number of 1 to 2 decoder to design 2 to 4 decoder. Use any external hardware if required



* Complemented output and complemented enable:

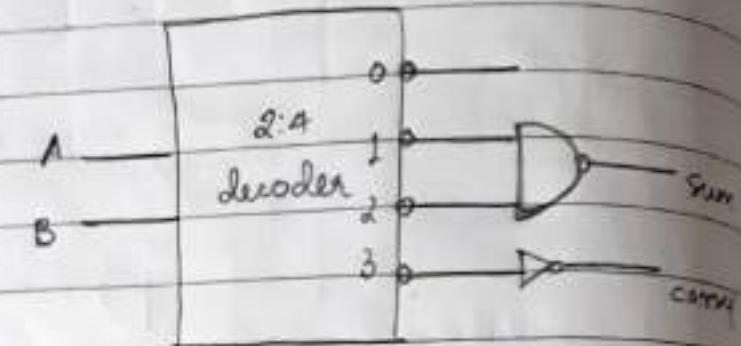
E	A	B	D ₀	D ₁	D ₂	D ₃
1	x	x	x	x	x	x
0	0	0	0	1	1	1
0	0	1	1	0	1	1
0	1	0	1	1	0	1
0	1	1	1	1	1	0



Q1: Design half adder using 2 to 4 complemented decoder.

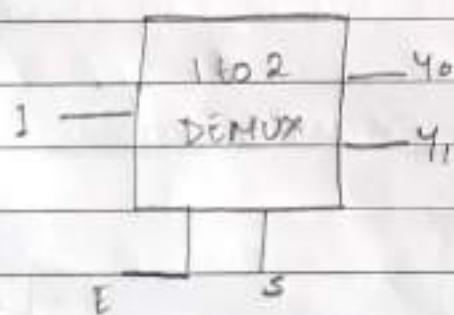
Sol: Truth table

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



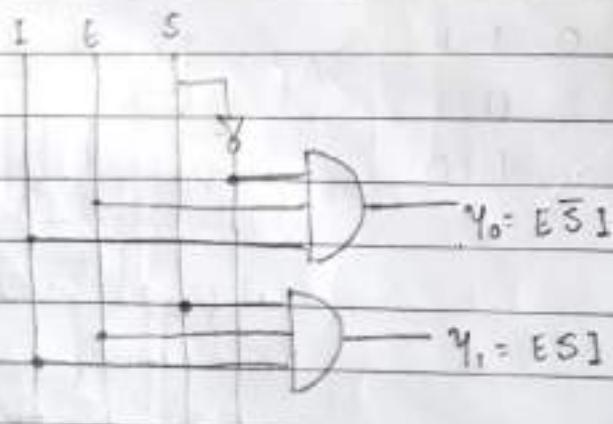
* DEMUX:

- 1 to 2 DEMUX:



Truth table

E	S	y_0	y_1
0	x	1	1
1	0	0	1
1	1	0	0

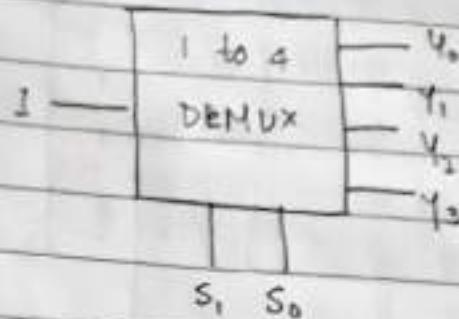


1 to 4 DEMUX:

$$m = 2^n$$

$$A = 2^n$$

$$n = \log_2 4$$



Truth table

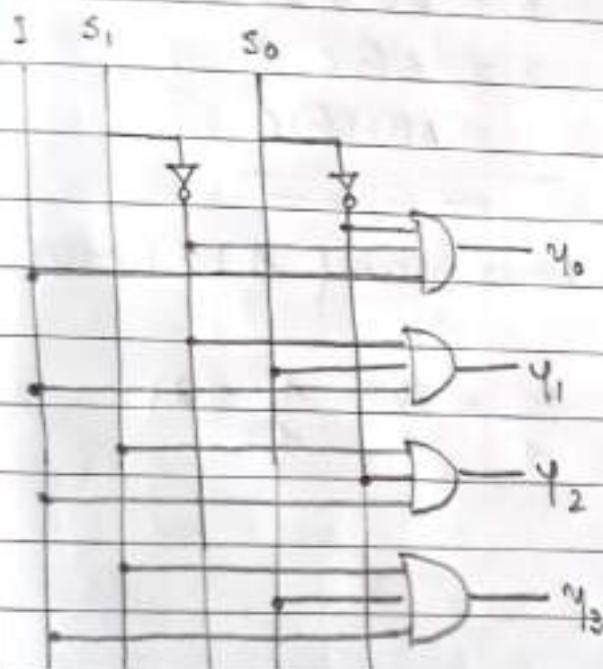
S_1	S_0	Y_0	Y_1	Y_2	Y_3
0	0	I	0	0	0
0	1	0	I	0	0
1	0	0	0	I	0
1	1	0	0	0	I

$$Y_0 = \bar{S}_1 \bar{S}_0 I$$

$$Y_1 = \bar{S}_1 S_0 I$$

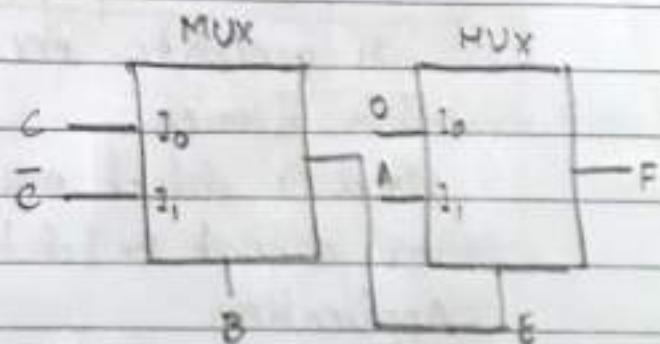
$$Y_2 = S_1 \bar{S}_0 I$$

$$Y_3 = S_1 S_0 I$$



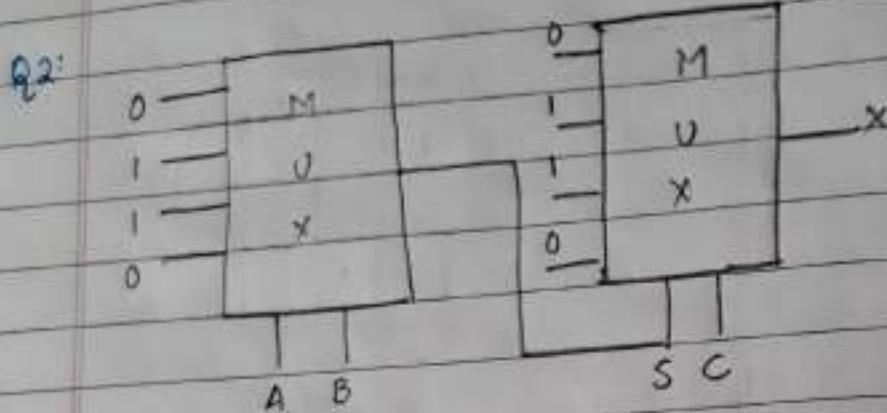
Q1: Find the final expression F :

<u>s1:</u>	B	E	F
0	C	O	
1	C	A	



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

$$F = EA = (\bar{B}C + B\bar{C})A = A(B \oplus C)$$



Truth table

A	B	S	C	X
0	0	0	0	0
0	1	1	0	1
1	0	1	1	0
1	1	0	1	1

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

$$S = A \oplus B$$

$$X = S\bar{C} + \bar{S}C$$

$$X = S \oplus C$$

$$X = A \oplus B \oplus C$$

Q3: To design 64 to 1 MUX, how many 4 to 1 MUX are needed.

21: 4 to 1 MUX

$$\frac{64}{4} + \frac{16}{4} + \frac{4}{4} = 21$$

* Parity generators:

- 4 bit Odd Parity generator:

It generates one if the input contains odd number of ones.

Used to detect one bit errors. but a two bit error cannot be detected.

Application

Error detector.

b_3	b_2	b_1	b_0	P	$b_3 b_2$	$b_1 b_0$	00	01	11	10
0	0	0	0	0	00	00	0	1	0	1
0	0	0	1	1	01	01	1	0	1	0
0	0	1	0	1	11	11	0	1	0	1
0	0	1	1	0	10	10	1	0	1	0
0	1	0	0	1						
0	1	0	1	0						
0	1	1	0	0						
0	1	1	1	1						
1	0	0	0	1						
1	0	0	1	0						
1	0	1	0	0						
1	0	1	1	1						
1	1	0	0	0						
1	1	0	1	1						
1	1	1	0	1						
1	1	1	1	0						

$$P = \bar{b}_3 \bar{b}_2 \bar{b}_1 b_0 + \bar{b}_3 \bar{b}_2 b_1 \bar{b}_0$$

$$+ \bar{b}_3 b_2 \bar{b}_1 \bar{b}_0 + \bar{b}_3 b_2 b_1 b_0$$

$$+ b_3 b_2 \bar{b}_1 b_0 + b_3 b_2 b_1 \bar{b}_0$$

$$+ b_3 \bar{b}_2 \bar{b}_1 \bar{b}_0 + b_3 \bar{b}_2 b_1 b_0$$

$$P = \bar{b}_3 \bar{b}_2 (\bar{b}_1 b_0 + b_1 \bar{b}_0)$$

$$+ \bar{b}_3 b_2 (\bar{b}_1 \bar{b}_0 + b_1 b_2)$$

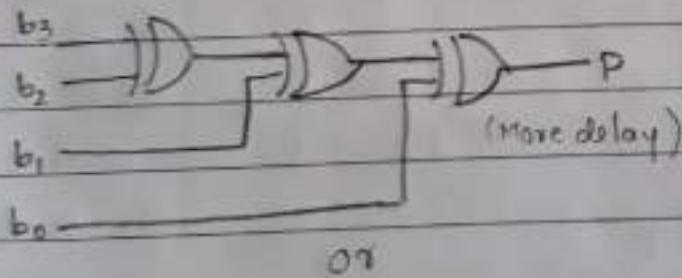
$$+ b_3 b_2 (\bar{b}_1 b_0 + b_1 \bar{b}_0)$$

$$+ b_3 \bar{b}_2 (\bar{b}_1 \bar{b}_0 + b_1 b_0)$$

$$P = \bar{b}_3 \bar{b}_2 (b_1 \oplus b_0) + \bar{b}_3 b_2 (\bar{b}_1 \oplus \bar{b}_0)$$

$$+ b_3 b_2 (b_1 \oplus b_0) + b_3 \bar{b}_2 (\bar{b}_1 \oplus \bar{b}_0)$$

①



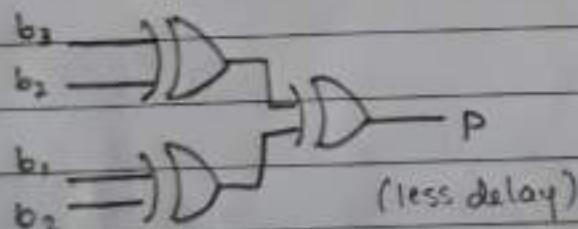
$$P = b_1 \oplus b_0 (\bar{b}_3 \bar{b}_2 + b_3 b_2)$$

$$+ \bar{b}_1 \oplus \bar{b}_0 (\bar{b}_3 b_2 + b_3 \bar{b}_2)$$

$$P = (b_1 \oplus b_0) (\bar{b}_2 \oplus \bar{b}_3)$$

$$+ (\bar{b}_1 \oplus \bar{b}_0) (b_2 \oplus b_3)$$

②



$$P = b_0 \oplus b_1 \oplus b_2 \oplus b_3$$

① is preferred as in some cases the data may not be available simultaneously.

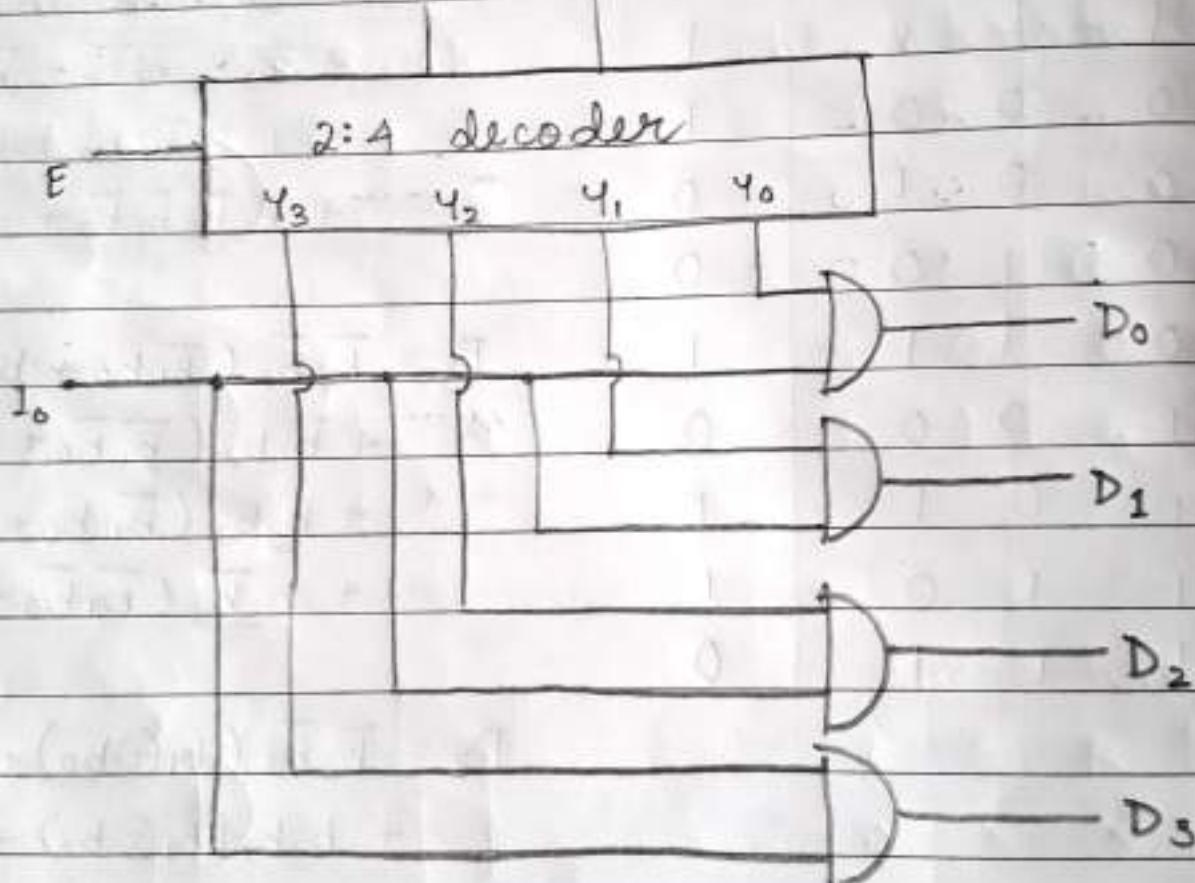
Q: Decoder to DEMUX: (with enable)

Truth table (Decoder)

A	B	y_0	y_1	y_2	y_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

Truth table (DEMUX)

E	s_0	s_1	y_0	y_1	y_2	y_3
1	0	0	1	0	0	0
1	0	1	0	1	0	0
1	1	0	1	1	0	0
1	1	1	1	1	1	1
	s_1	s_0	0	X	X	0
						0



Unit 2 : continued

Q: Write the memory structure in the form of ~~memory~~ logic gates to realise $f_1 = \Sigma m(0, 1, 4, 6)$, $f_2 = \Sigma m(2, 3, 5, 6, 7)$, $f_3 = \Sigma m(0, 1, 2, 6)$.

Sol: 3:8 decoder is used

	000	1	0	1
	001	1	0	1
3:8 decoder	010	0	1	1
	011	0	1	0
	100	1	0	0
	101	0	1	0
	110	1	1	1
	111	0	1	0
		f_1	f_2	f_3

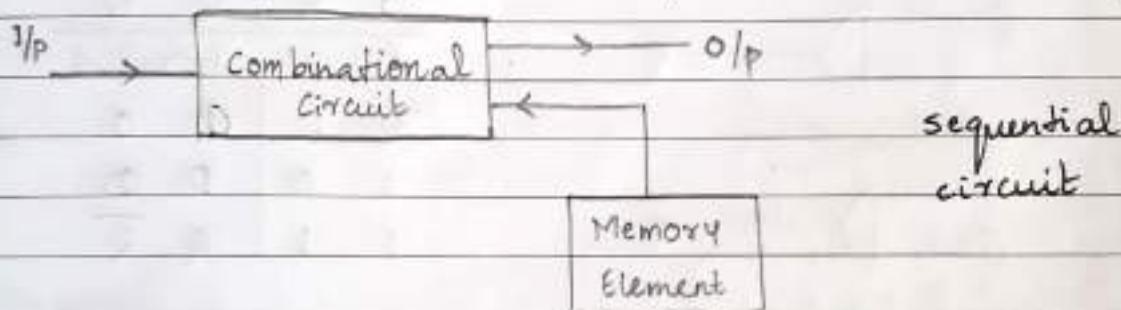
UNIT - 3

Sequential Circuits* sequential circuits:

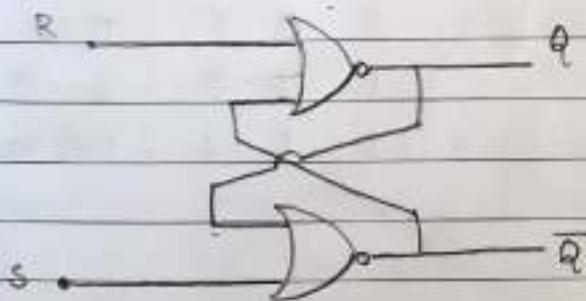
sequential circuit is a combination of combinational circuit along with memory.

Previous output has role in the computation.

- present output depends on input and previous output

* Storage or Memory Elements:SR

① Latch: Holds a certain value until another value is entered



S	R	Q	\bar{Q}
0	0	Q	\bar{Q}
0	1	0	1
1	0	1	0
1	1	Not Valid	

when $S=1, R=1$ then $Q=\bar{Q}=0$
but that is invalid

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

When $S=0$ and $R=0$

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

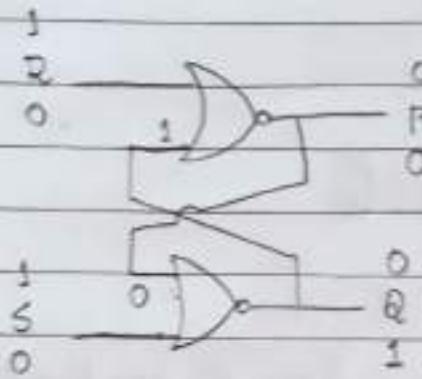
then the latch holds

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

on to the previous

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

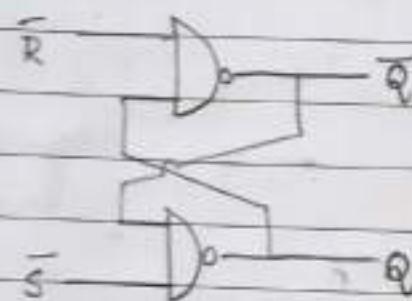
value.



When $R=1, S=1$ then $P=0, Q=0$
 When $R=0, S=0$ then $P=0, Q=1$
 but when $R=0, S=0$ then the previous value has to hold on but
 For $R=1, S=1$ this is invalid.
 There will be small delay between the two gates.

Using NAND gates.

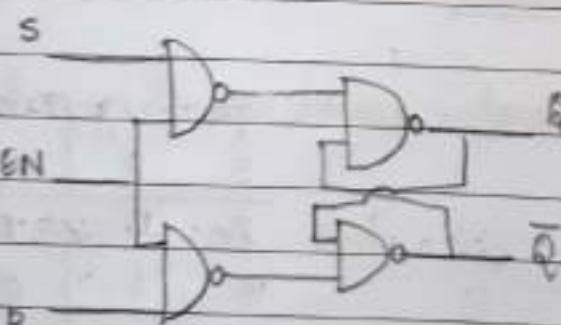
$\bar{S}\bar{R}$ Latch



Characteristic Table

\bar{S}	\bar{R}	Q	\bar{Q}
0	0	Invalid	
0	1	0	1
1	0	1	0
1	1	Q	\bar{Q}

With enable



Characteristic Table

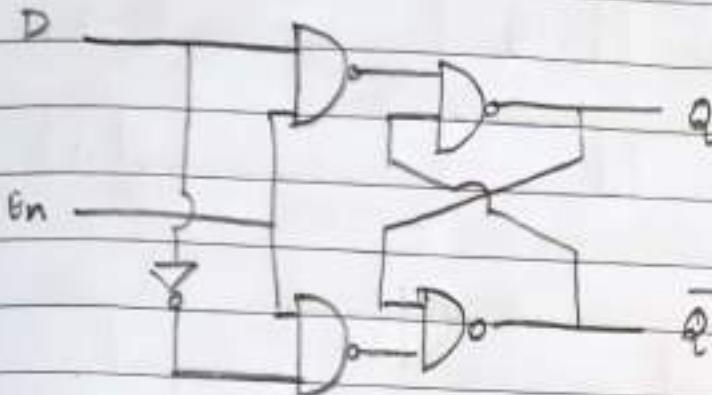
E	S	R	Q	\bar{Q}
0	x	x	Q	\bar{Q}
1	0	0	Q	\bar{Q}
1	0	1	0	1
1	1	0	1	0
1	1	1	Not valid	

SR latch



gated latch - with enable

2. D latch:

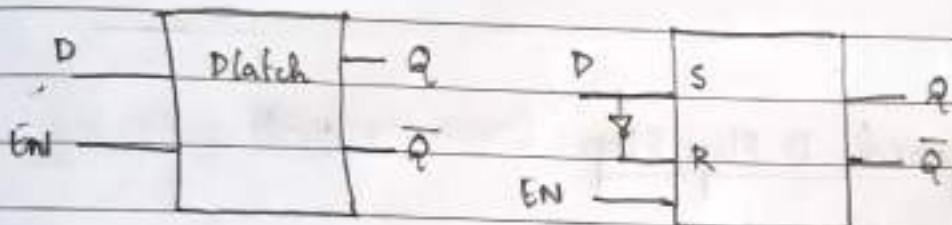


Characteristic Table

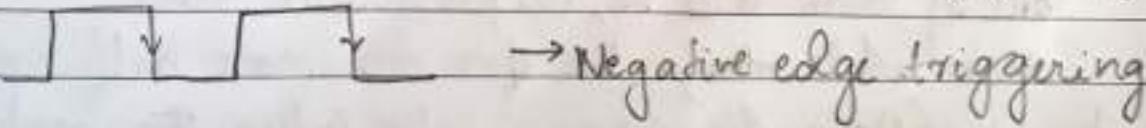
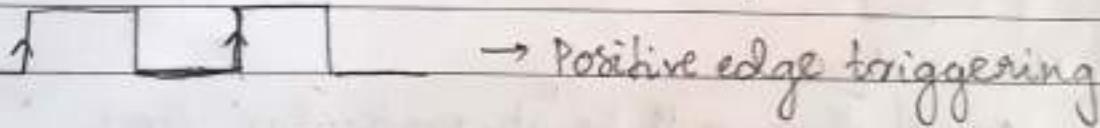
EN	D	Q	\bar{Q}
0	x	Q	\bar{Q}
1	0	0	1
1	1	1	0

reset ..

set



* Triggering:

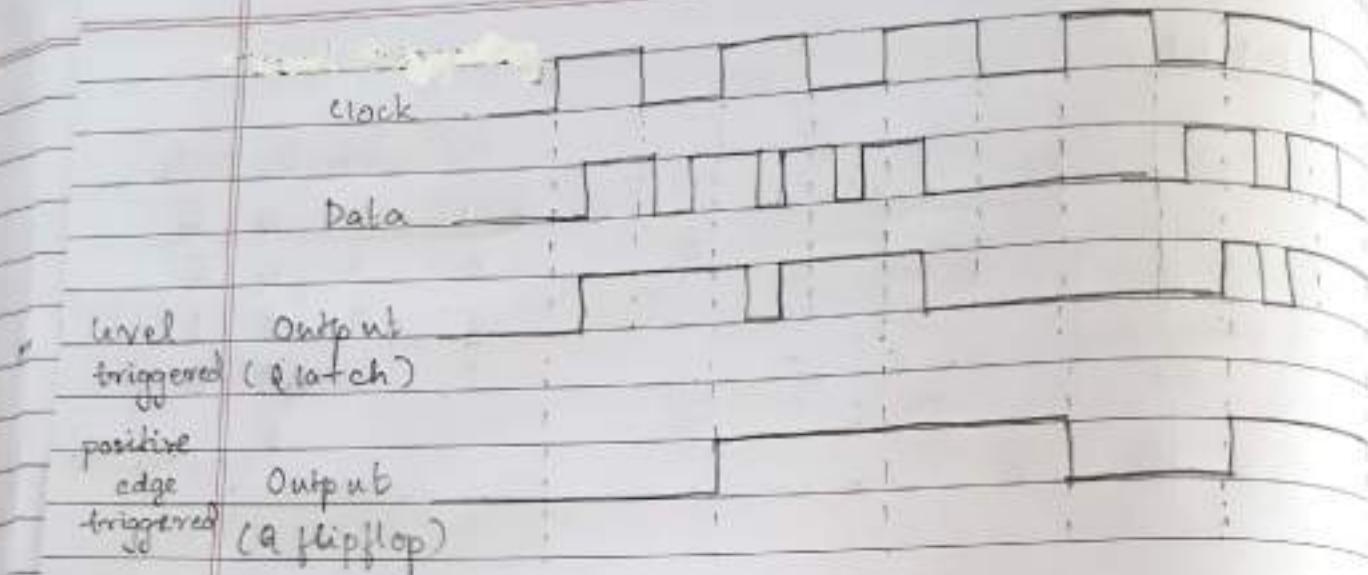


Edge triggering is employed in flip flops.

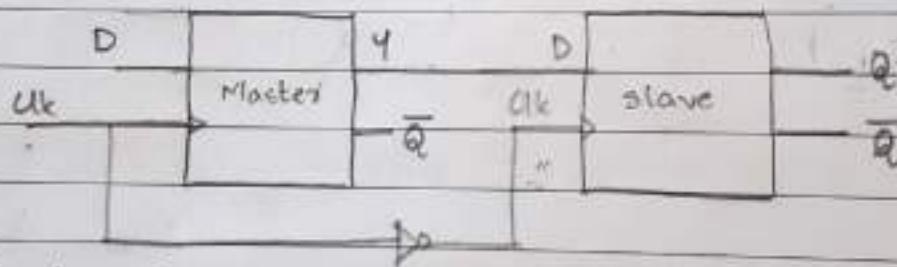
From one positive edge to the next the output remains the same in positive edge triggering.

~~latch~~ is level triggered.

The output changes only at the level otherwise the previous output is retained in level triggering.



* Edge Triggered D Flip Flop : (data storage)



Master Slave

One of them will be ^{ON} at a particular time.
slave holds the data for one more half cycle i.e.,
it will retain the same data when the master
flip flop is ON when itself is OFF.

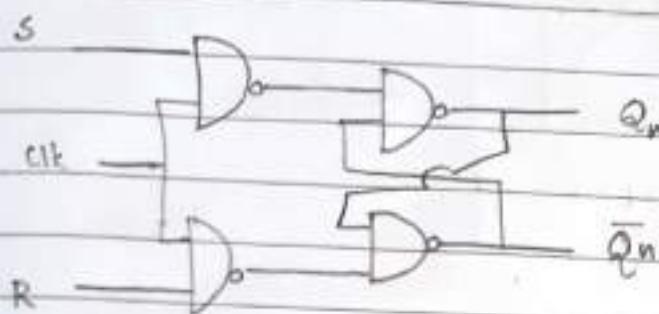
Setup time - Input must be maintained at a constant value prior to the occurrence of the clock transition.

Hold time - The input must not change after the application of the clock transition.

Propagation delay time - The interval between the triggered edge and the stabilisation of the output

to a new state.

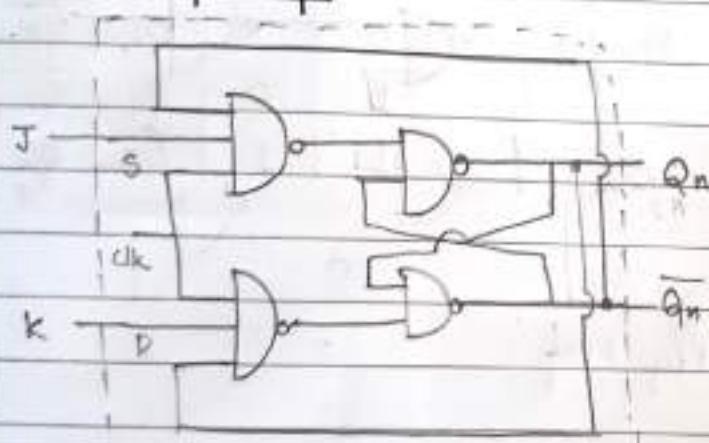
* SR Flip Flop:



Characteristic Table

clk	S	R	Q_n	\bar{Q}_n
0	x	x	Q_{n-1}	\bar{Q}_{n-1}
1	0	0	Q_{n-1}	\bar{Q}_{n-1}
1	0	1	0	1
1	1	0	1	0
1	1	1	Not valid.	

* Jk Flip Flop:



Characteristic Table

clk	J	K	Q_n	\bar{Q}_n
0	x	x	Q_{n-1}	\bar{Q}_{n-1}
1	0	0	Q_{n-1}	\bar{Q}_{n-1}
1	0	1	0	1
1	1	0	1	0
1	1	1	\bar{Q}_n	Q_n

toggles the
previous output

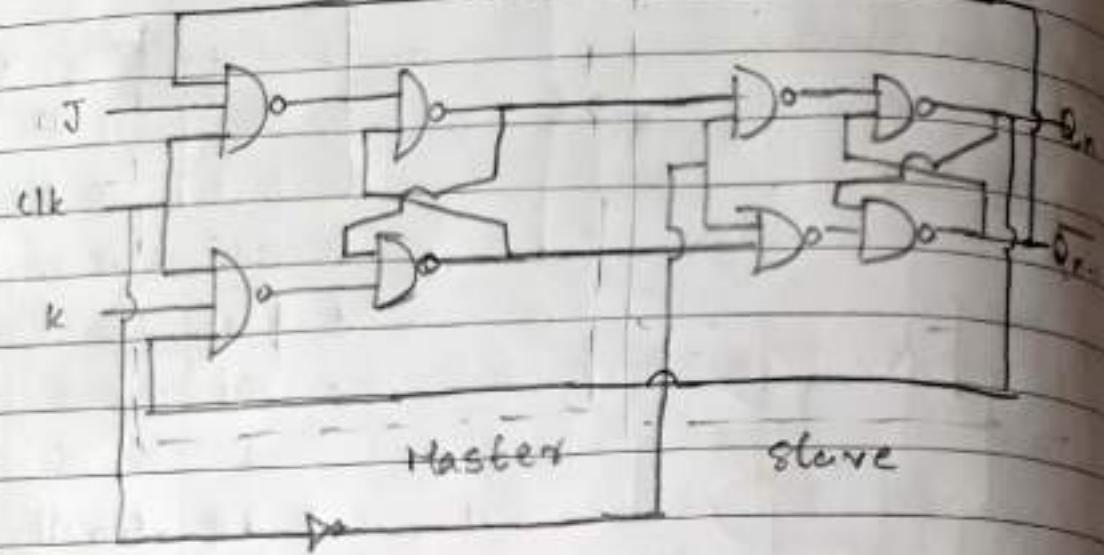
Racing - uncontrollable toggle

This racing condition is overcome by considering master slave flip flop.

* Master-Slave JK Flip Flop:

This is used to avoid racing condition and data is stored for longer period.

But due to more hardware complexity it leads to more delay.



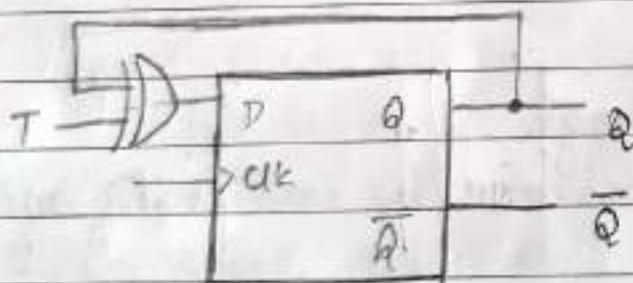
* Toggle Flip Flop / T-Flip Flop:

Characteristic Table

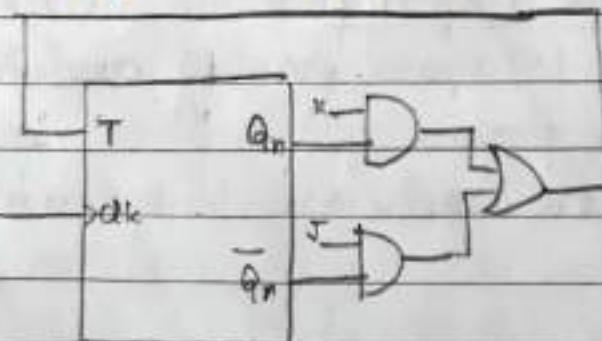
clk	T	Q_n, \bar{Q}_n
0	x	Q_{n-1}, \bar{Q}_{n-1}
1	0	\bar{Q}_{n-1}, \bar{Q}_n
1	1	\bar{Q}_{n-1}, Q_n



Q1: T Flip Flop from D Flip Flop



Q2: JK Flip Flop from T Flip Flop



- * Characteristic Equation: (Gives the expression for the next output)

1. JK Flip Flop:

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

J	Q_n	00	01	11	10
0	0	0	1	0	0
1	1	1	1	0	1

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

2. SR Flip Flop:

S	R	Q_n	Q_{n+1}
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

S	RQ_n	00	01	11	10
0	0	0	1	0	0
1	1	1	1	x	x

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

3. T Flip Flop:

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

T	Q_n	0	1
0	0	0	1
1	1	1	0

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

<u>D Flip Flop:</u>		
D	Q _n	Q _{n+1}
0	0	0
0	1	0
1	0	1
1	1	1

D	Q _n	1
0	0	0
1	1	1

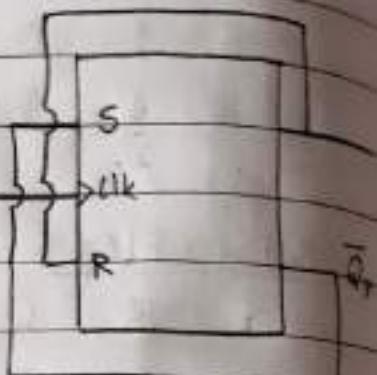
$$Q_{n+1} = D$$

Q1: Determine the Q output relation to the clock given.
What is the specific function that is being performed?

clk	Q _n	\bar{Q}_n	S	R
1	0	1	1	0
1	1	0	0	1

Inverted input of S is given to R as input.

Here the output Q_n is getting toggled. Therefore it is acting as a T flip flop.



$$S = \bar{Q}_n$$

$$R = Q_n$$

* Applications of Flip Flop:

1. Counters:

Counts from 0 to $2^n - 1$ or $2^n - 1$ to 0.

Each flip flop deals only with one bit hence the number of flip flops required is n.

Ex: 0 to 15 i.e., 0 to $2^4 - 1 \Rightarrow 4$ flip flops
Usually JK flip flops are used.

PRESET - Output $Q=1$ irrespective of clock and input

CLEAR - Output $Q=0$ irrespective of clock and input
* only at that instant of time.

Modulus counter : $0-n$, once it counts till n , it comes back to 0 and starts counting again.

- a. Asynchronous counters: (ripple counter) (delay gets added up)

The output of one flip flop triggers the next flip flop, i.e., all the flip flops are not triggered at a time.

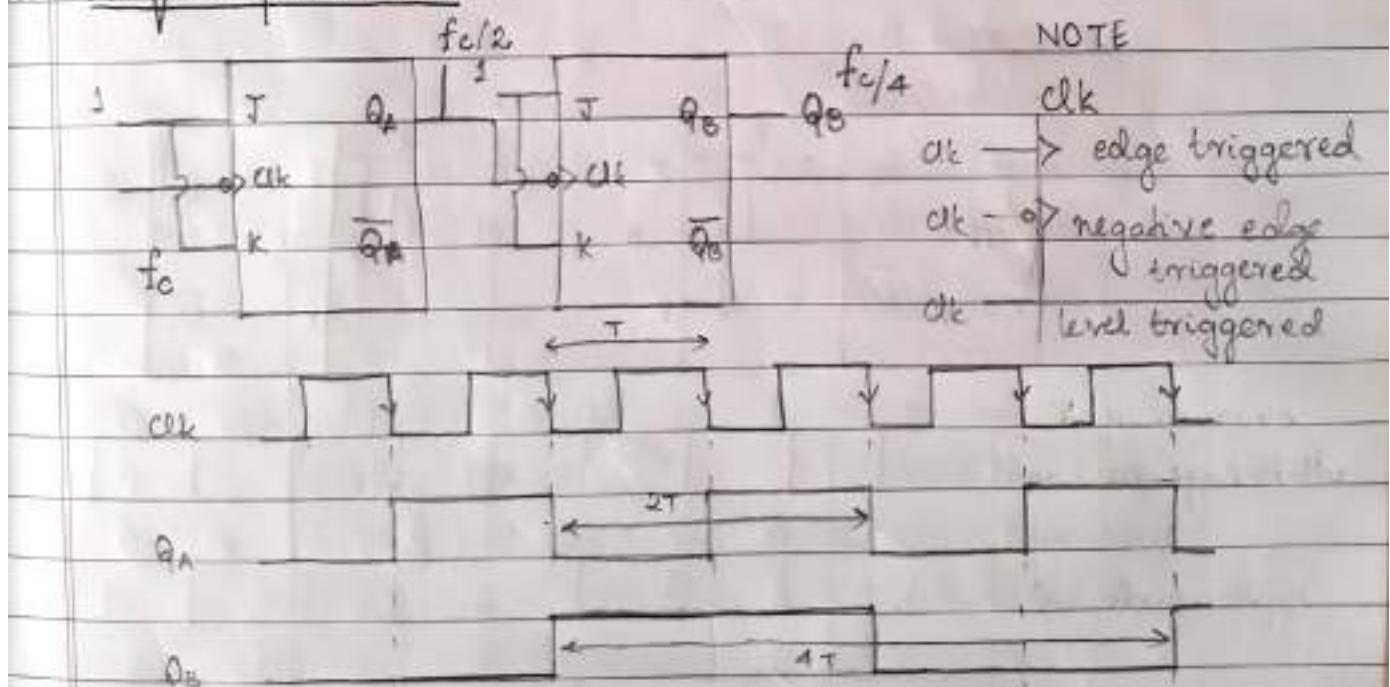
- b. Synchronous Counters: (constant delay)

All the flip flops are triggered simultaneously.

Thus a common clock is given to all the flip flops whereas in asynchronous, clock is given only to the first flip flop.

As the counting required increases synchronous counter is preferred as the delay gets added up in asynchronous counter as the number of flip flop increases.

2. Frequency divider:



This is asynchronous as the output of the first flip flop triggers the second flip flop.

clock period : T

Q_1 - negative triggered (clk)

Time period : $2T$

Q_2 - negative transition of Q_1

Time period : $4T$

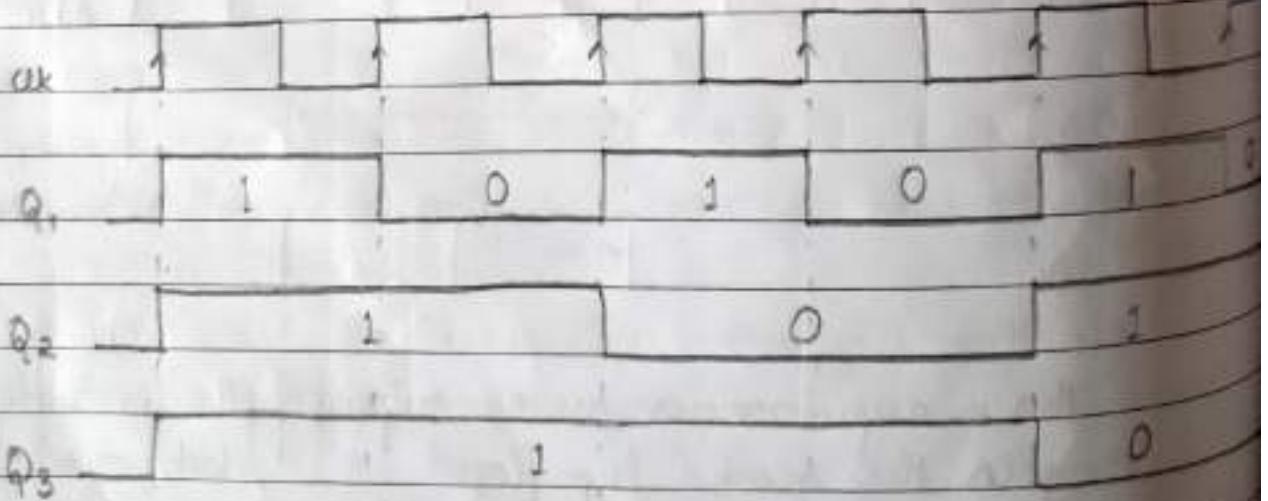
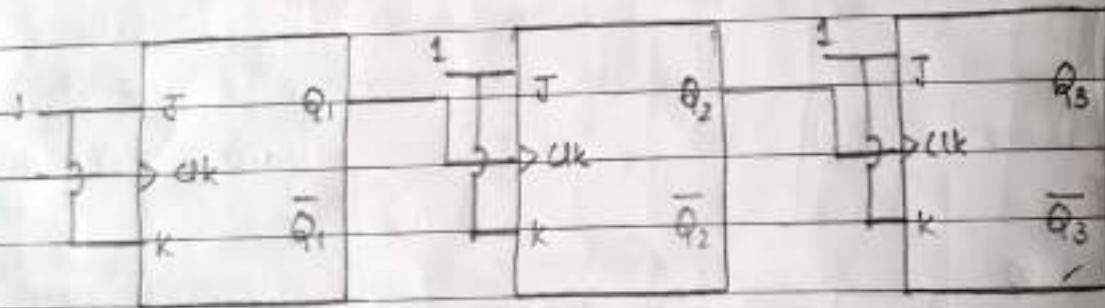
With increase in number of flip flop, frequency divides further.

$$n \text{- Flip Flops} \quad \text{frequency} = f_c / 2^n$$

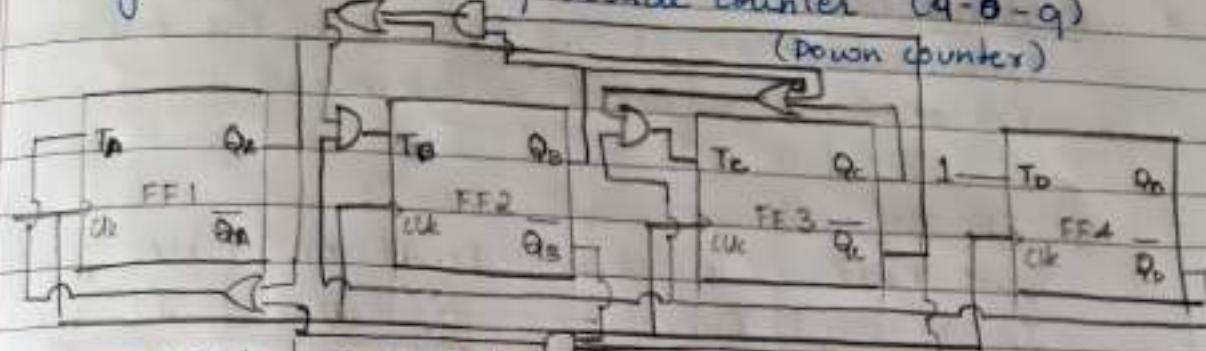
- Maximum Clock Frequency
The highest rate at which a flip flop can be reliably triggered.
- Pulse Width
Minimum pulse width is usually specified by manufacturer. It is mainly the minimum high time and minimum low time.
- Power Dissipation
Total power consumed by the flip flop.

* 3 bit Ripple Counter:

NOTE: Latency : Time taken by the device to get set to its stable state.



Q1: Design a BCD counter / Decade counter ($Q_3-Q_2-Q_1-Q_0$)
 (Down counter)



Present State $Q_3\ Q_2\ Q_1\ Q_0$	Next State $Q_3'\ Q_2'\ Q_1'\ Q_0'$	T_A	T_B	T_C	T_D
1 0 0 1	1 0 0 0	0	0	0	1
1 0 0 0	0 1 1 1	1	1	1	1
0 1 1 1	0 1 1 0	0	0	0	1
0 1 1 0	0 1 0 1	0	0	1	1
0 1 0 1	0 1 0 0	0	0	0	1
0 1 0 0	0 0 1 1	0	1	1	1
0 0 1 1	0 0 1 0	0	0	0	1
0 0 1 0	0 0 0 1	0	0	1	1
0 0 0 1	0 0 0 0	0	0	0	1
0 0 0 0	1 0 0 1	1	0	0	1

$T_A\ Q_3Q_2Q_0$				$T_B\ Q_3Q_2Q_0$					
$Q_3Q_2Q_1$	00	01	11	10	$Q_3Q_2Q_1$	00	01	11	10
00	1	0	0	0	00	0	0	0	0
01	0	0	0	0	01	1	0	0	0
11	X	X	X	X	11	X	X	X	X
10	1	0	X	X	10	1	0	X	X

$$T_A = \bar{Q}_3 \bar{Q}_2 \bar{Q}_1 \bar{Q}_0 + Q_3 \bar{Q}_0$$

#EQA

$$T_B = Q_3 \bar{Q}_2 \bar{Q}_1 \bar{Q}_0 + Q_3 \bar{Q}_0$$

$$T_C = Q_3 \bar{Q}_2 Q_1 \bar{Q}_0 + Q_3 \bar{Q}_2 \bar{Q}_1 + Q_3 Q_2 \bar{Q}_0$$

$$T_D = 1$$

$T_C\ Q_3Q_2Q_0$				
$Q_3Q_2Q_1$	00	01	11	10
00	0	0	0	1
01	1	0	0	1
11	X	X	X	X
10	1	0	X	X

* Excitation Table:

For a certain output what will be the minimum number of inputs required is determined.

- JK Flip Flop:

Truth Table

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

Excitation Table

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

- SR Flip Flop:

Truth Table

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

- D Flip Flop:

Truth Table

D	Q _n	Q _{n+1}
0	0	0
0	1	1
1	0	1
1	1	1

Excitation Table

Q _n	Q _{n+1}	D
0	0	0
0	1	1
1	0	0
1	1	1

- T Flip Flop:

Truth Table

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

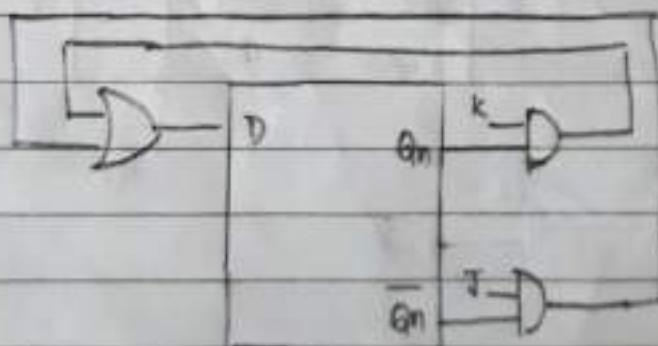
Excitation Table

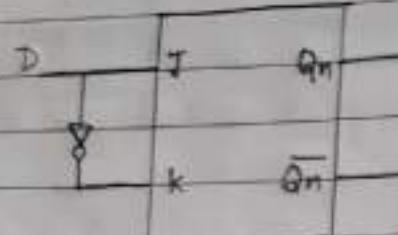
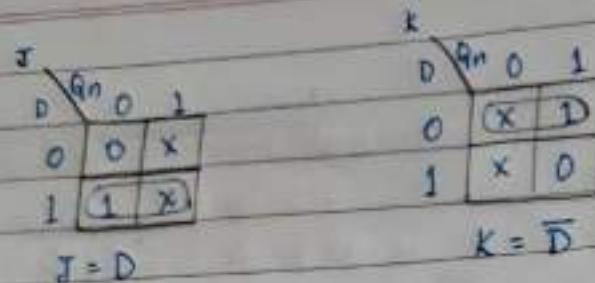
Q _n	Q _{n+1}	T
0	0	0
0	1	1
1	0	1
1	1	0

Q: Convert D flip flop to JK flip flop and JK flip flop to D flip flop.

Excitation Table

D	Q _n	Q _{n+1}	J K		D = J Q _n + K Q _n	D = J Q _n + K Q _n
			J	K		
0	0	0	0	X	0 0 1	0 0
0	1	0	X	1	1 1 1	0 1
1	0	1	1	X	100, 110	100, 110
1	1	1	X	0	101, 001	101, 001





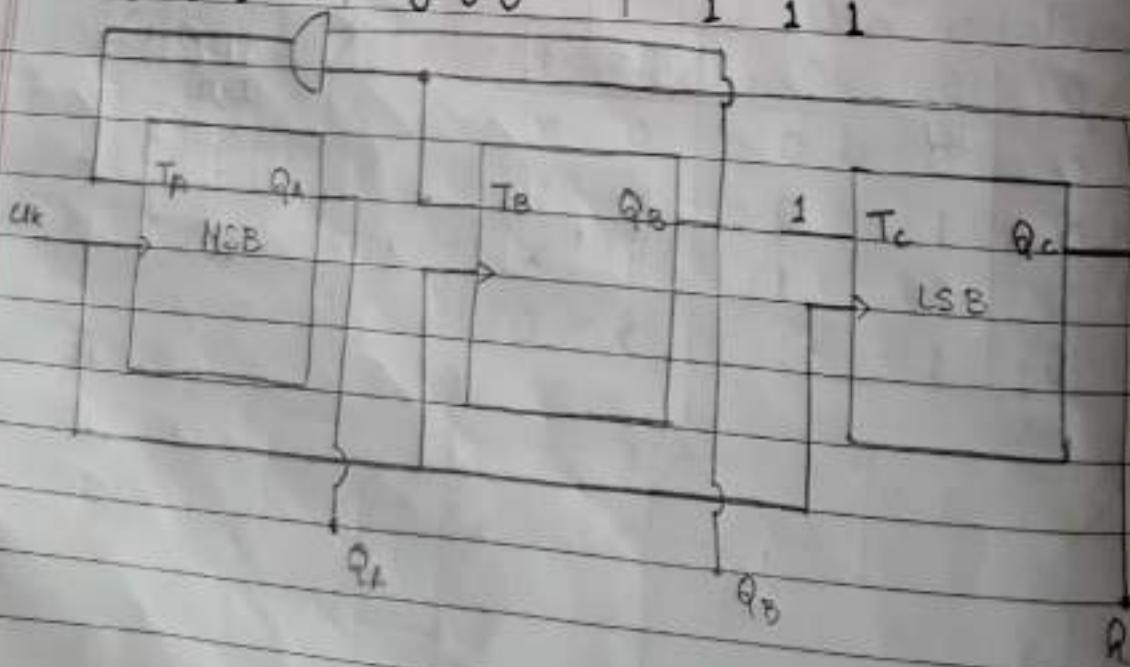
Any combination
of this kind can be done
for all the 4 flip flops

* Synchronous counters:

Present State	Next State	Flip Flop Inputs
0 0 0	0 0 1	0 0 1
0 0 1	0 1 0	0 1 1
0 1 0	0 1 1	0 0 1
0 1 1	1 0 0	1 1 1
1 0 0	1 0 1	0 0 1
1 0 1	1 1 0	0 1 1
1 1 0	1 1 1	0 0 1
1 1 1	0 0 0	1 1 1

Using

T flip flop



		T_A				T_B				T_C			
		$Q_A \setminus Q_B Q_C$				$Q_A \setminus Q_B Q_C$				$Q_A \setminus Q_B Q_C$			
		00	01	11	10	00	01	11	10	00	01	11	10
0	0	0	0	1	0	0	0	1	1	0	0	1	1
1	0	0	0	1	0	1	0	1	1	0	1	1	0

$$T_A = Q_B Q_C$$

$$T_B = Q_C$$

$$T_C = 1$$

Q: Design a synchronous counter for the sequence
 $0 \rightarrow 4 \rightarrow 1 \rightarrow 2 \rightarrow 6 \rightarrow 0$

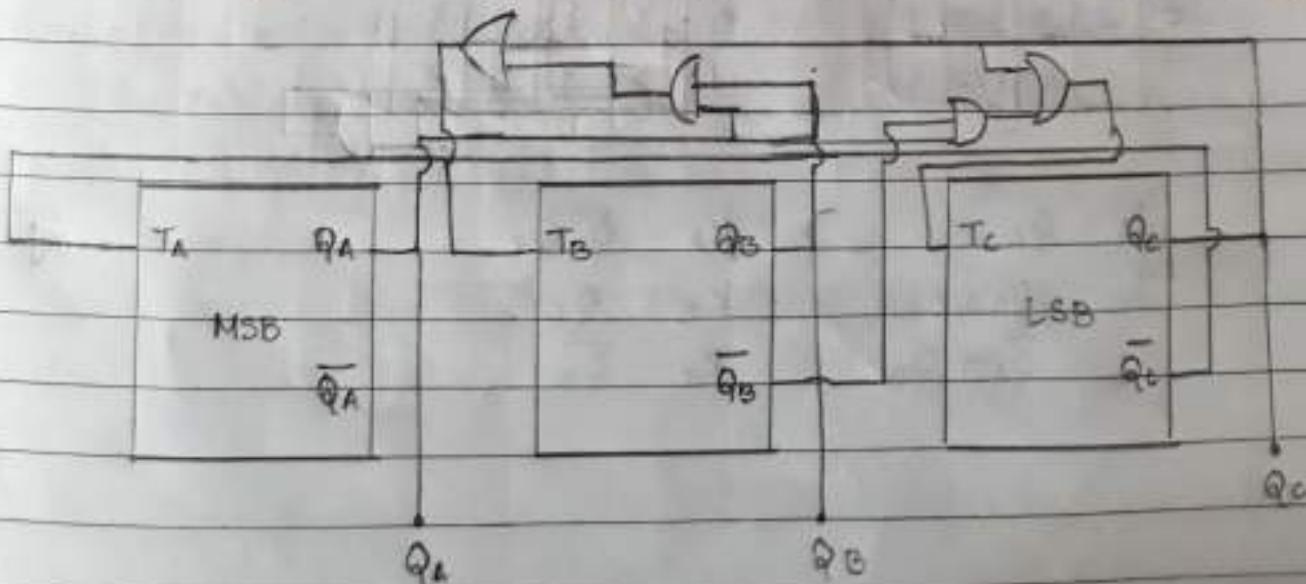
sd:	Present State	Next State	T Flip Flop Inputs
	$Q_A \quad Q_B \quad Q_C$ 0 0 0	$Q_A^+ \quad Q_B^- \quad Q_C^+$ 1 0 0	$T_A \quad T_B \quad T_C$ 1 0 0
	0 0 1	0 1 0	0 1 1
	0 1 0	1 1 0	1 0 0
	0 1 1	x x x	x x x
	1 0 0	0 0 1	1 0 1
	1 0 1	x x x	x x x
	1 1 0	0 0 0	1 1 0
	1 1 1	x x x	x x x

T_A				T_B				T_C						
$Q_A \setminus Q_B Q_C$				$Q_A \setminus Q_B Q_C$				$Q_A \setminus Q_B Q_C$						
00 01 11 10				00 01 11 10				00 01 11 10						
0	1	0	x	1	0	0	1	x	0	0	0	1	x	0
1	1	x	x	1	1	0	x	x	1	1	1	x	x	0

$$T_A = \bar{Q}_C$$

$$T_B = \bar{Q}_C + Q_A Q_B$$

$$T_C = Q_C + Q_A \bar{Q}_B$$



Q: Design a synchronous counter using JK flip flop for the sequence
 $0 \rightarrow 1 \rightarrow 4 \rightarrow 6 \rightarrow 7 \rightarrow 5 \rightarrow 0$

Present Stage $Q_A\ Q_B\ Q_C$	Next Stage $Q_A'\ Q_B'\ Q_C'$	$J_A\ K_A\ J_B\ K_B\ J_C\ K_C$
0 0 0	0 0 1	0 * 0 x 1 x
0 0 1	1 0 0	1 x 0 x x 1
0 1 0	x x x	x x x x x x
0 1 1	x x x	x x x x x x
1 0 0	1 1 0	x 0 1 x 0 x
1 0 1	0 0 0	x 1 0 * x 1
1 1 0	1 1 1	x 0 x 0 1 x
1 1 1	1 0 1	x 0 * 1 x 0

J_A	00 01 11 10	K_A	00 01 11 10
0 0	1 x x	0 x x	x x x
1 x	x x x	1 0 1	0 0 0

J_B	00 01 11 10	K_B	00 01 11 10
0 0	0 x x	0 x x	x x x
1 1	0 x x	1 x 1	0 0 0

J_C	00 01 11 10	K_C	00 01 11 10
0 1	x x x	0 x 1	x x x
1 0	x x 1	1 x 0	0 x

$$J_A = Q_C$$

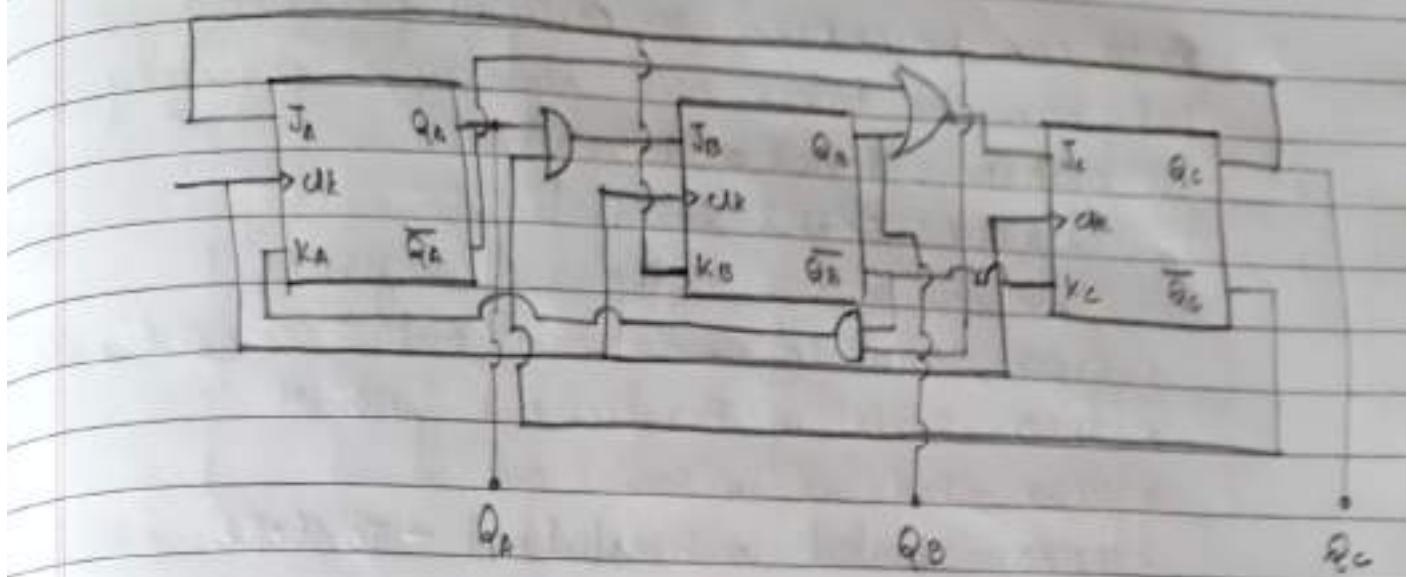
$$J_B = \bar{Q}_A \bar{Q}_C$$

$$J_C = \bar{Q}_A + \bar{Q}_B$$

$$K_A = \bar{Q}_B Q_C$$

$$K_B = Q_C$$

$$K_C = \bar{Q}_B$$



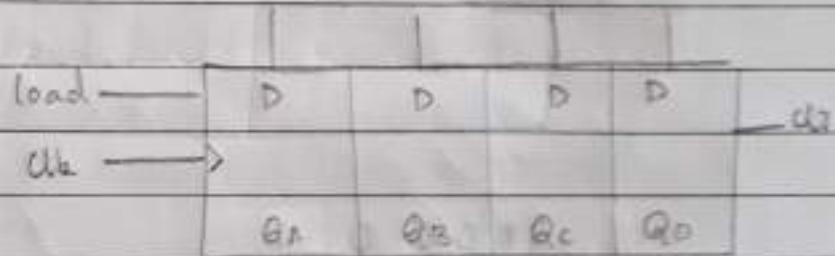
3. Registers:

Multiple flip flops connected together.

Operations of registers:

- Shift registers (Used to shift data)

Load pin is used to have an initial data.



Synchronous load / clear

- with respect to clock (when clk is high and load is high then the data is loaded)

Asynchronous load / clear

- irrespective of clock the data is loaded when the load is high and data is cleared when the clear is high
advantage:

Load keeps the data as it is until it receives another data

Every flip flop with a slower clock helps to have delay.

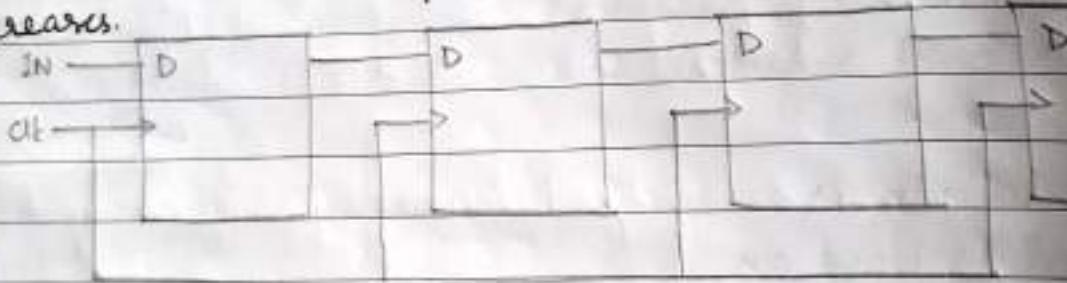
Data can be entered serially or parallelly
 Data can be taken out serially or parallelly
 DATA IN : serial or parallel
 DATA OUT: serial or parallel.

Hence we can design 4 types of registers.

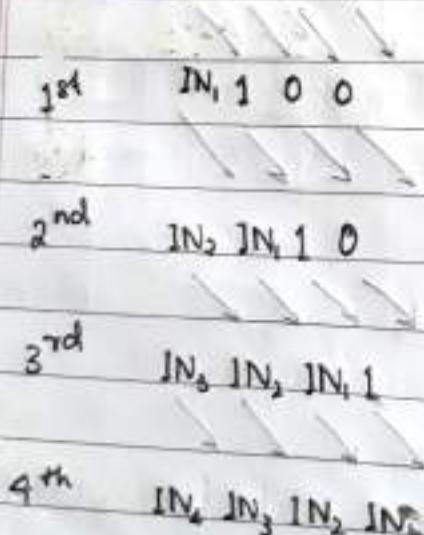
1. SISO - Serial in and serial out - Delay production
2. SIPO - Serial in Parallel out } MODEN
3. PIPO - Parallel in Serial out
4. PIPO - Parallel in Parallel out - TO STORE

* Serial in serial Out:

The time to obtain output increases of number of bits loaded increases.



Clk 1 0 0 0

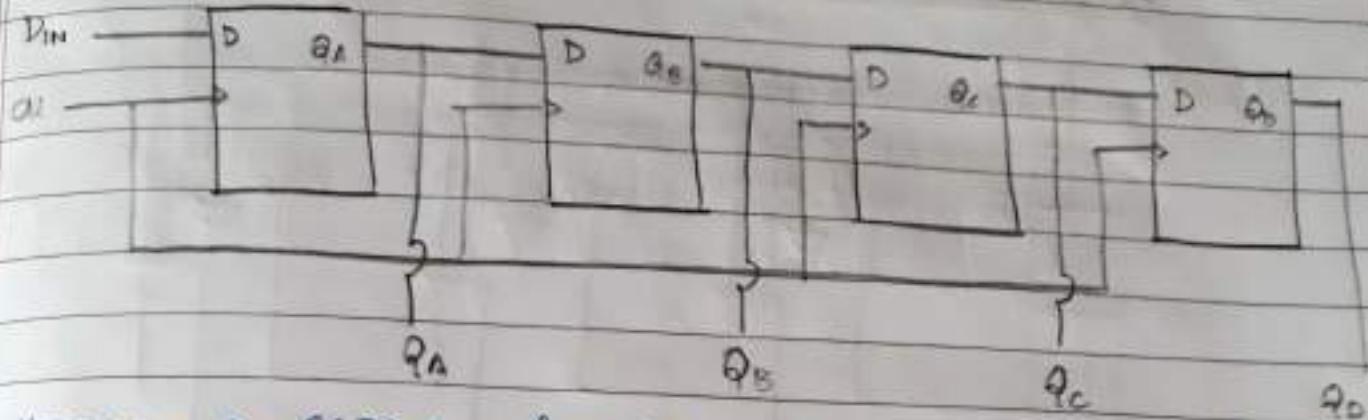


This type of registers can be used to have delay

Ex: To store a 4 bit data the output is obtained after 4 clock pulses.

i.e., n flip flop $\Rightarrow n$ clock pulse delay

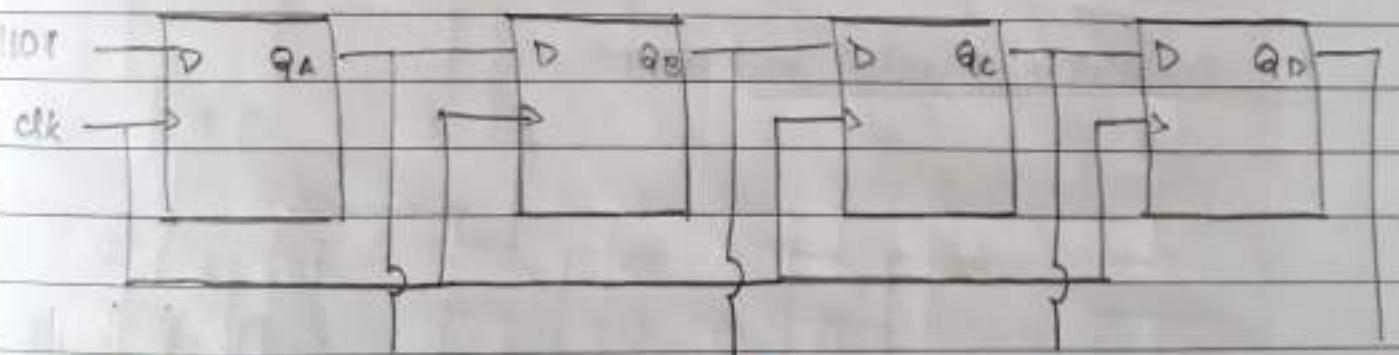
* Serial in Parallel Out



Q: How can a SIPO register be used as a SISO register?

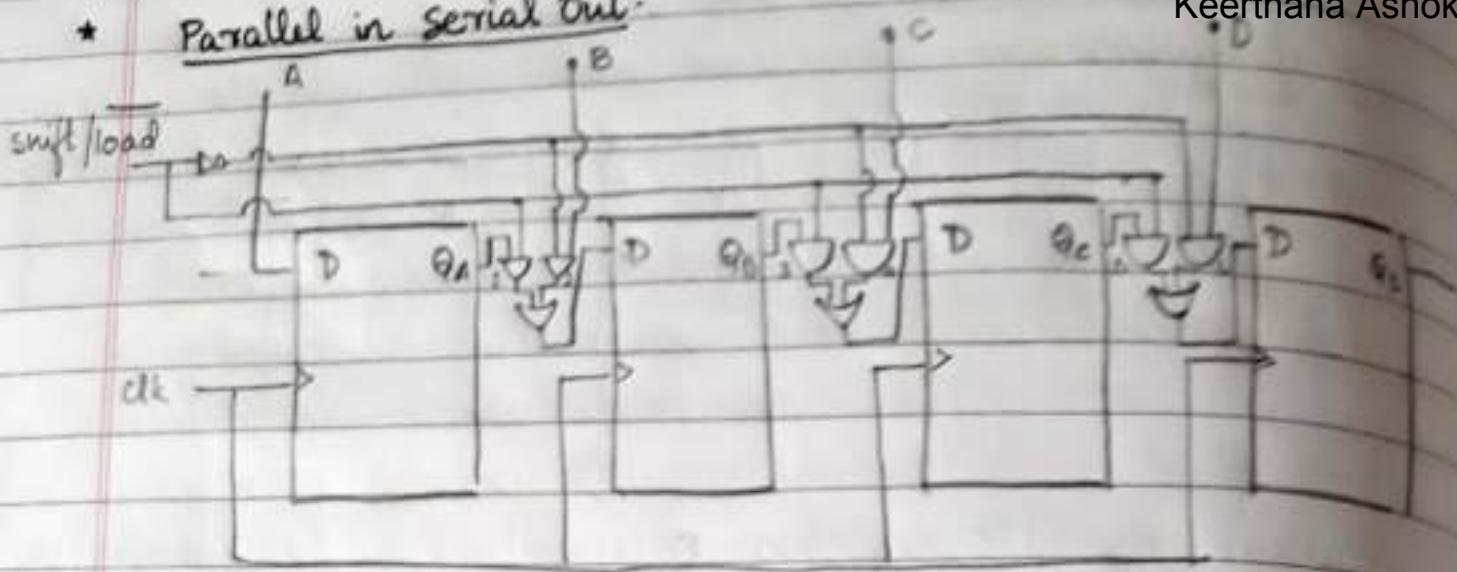
Sol: Q_A, Q_B, Q_C are not considered only Q_D is used to obtain output.

Q: The binary number 1101 is serially entered into a 4 bit parallel out shift register, i.e., initially cleared. What are the Q outputs after 2 clock pulses?



Initially	0	0	0	0
1st clock	1	0	0	0
2nd clock	0	1	0	0

- * Parallel in serial Out:



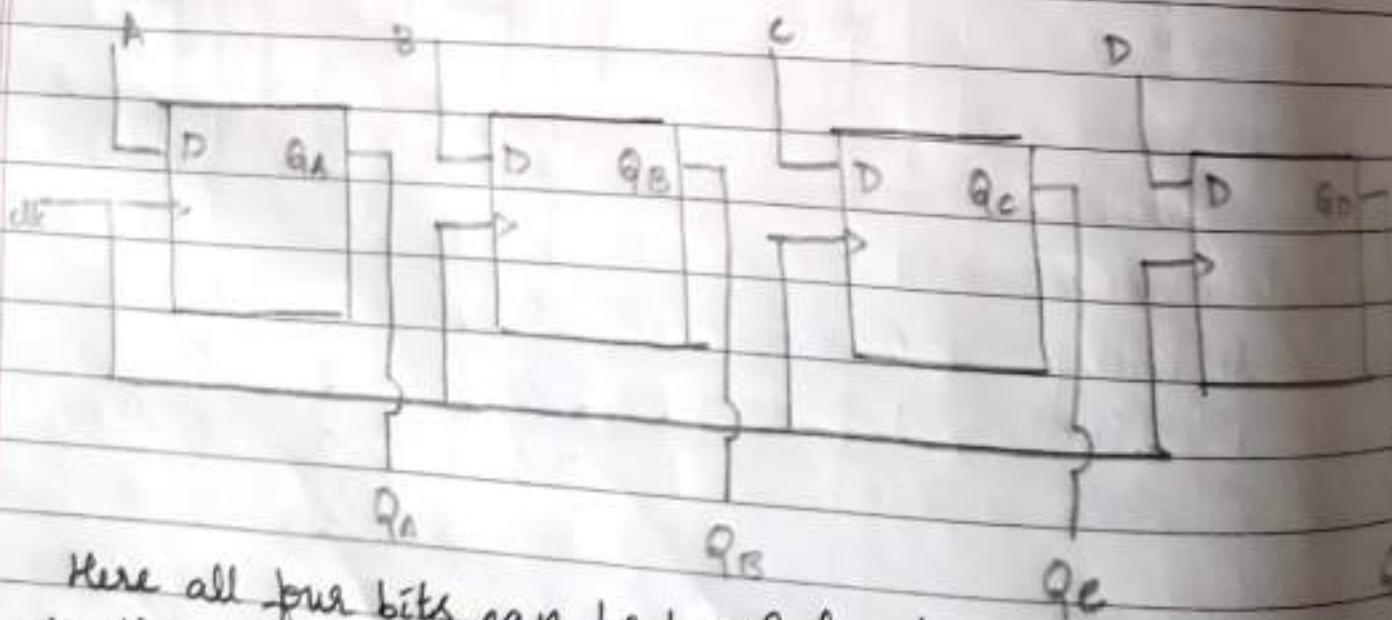
When shift / load = 0 then load operation is activated
then gates 2, 4, 6 are activated.

When shift / load = 1 then shifting of data takes place
then gates 1, 3, 5 are activated.

Drawback

Requires many external hardwares.

- * Parallel in Parallel out:

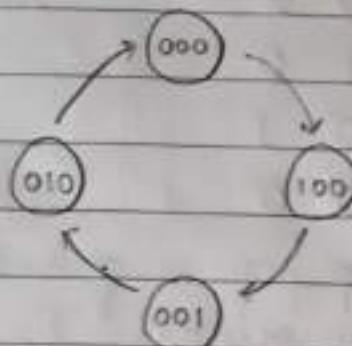


Here all four bits can be loaded at the same time and all the bits are read out at the same time.

* State diagram: (synchronous sequence counters)
 Each circle indicates one state, and the flow is indicated by arrows.

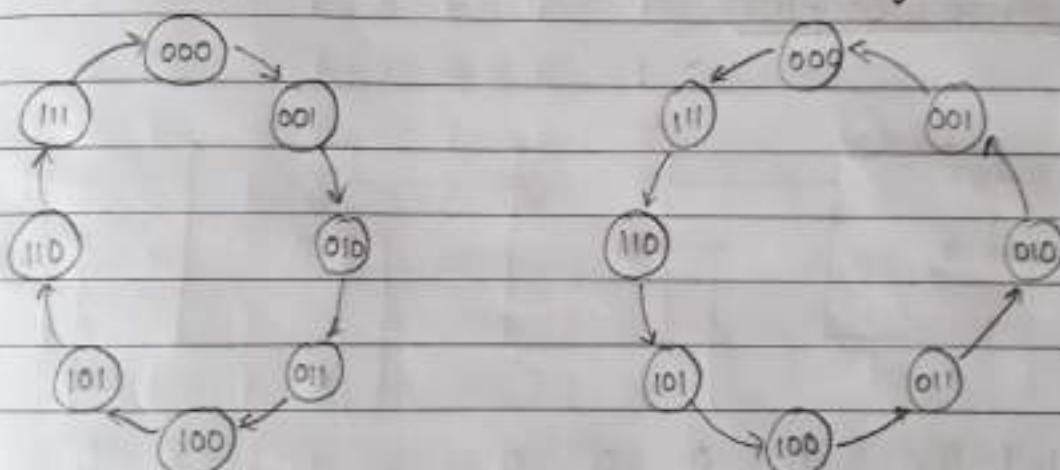
Ex : $0 \rightarrow 1 \rightarrow 2 \rightarrow 0$

There are 4 states

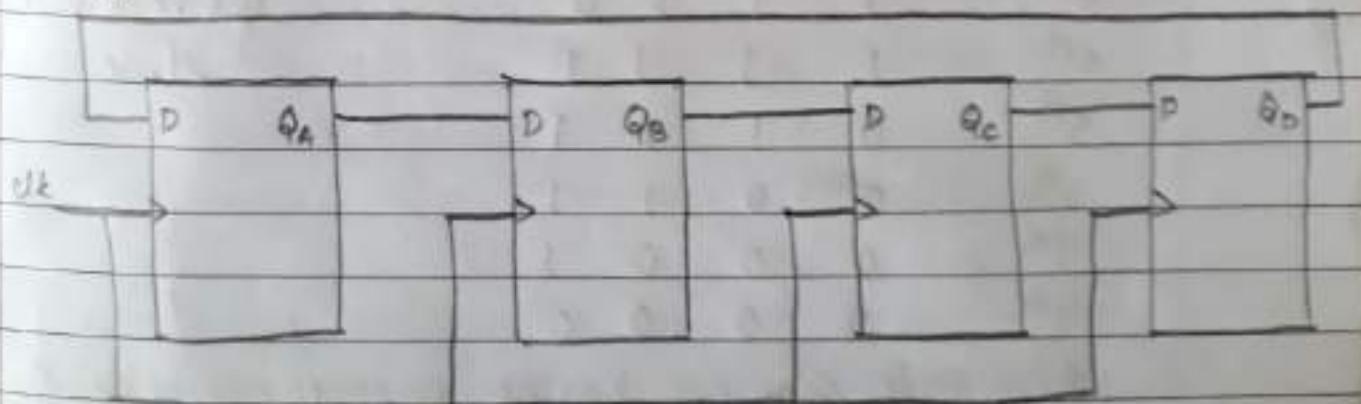


upcounting

Down counting



* Ring Counter:



Data is rotated and not taken out at any stage.
 Hence no data is lost.

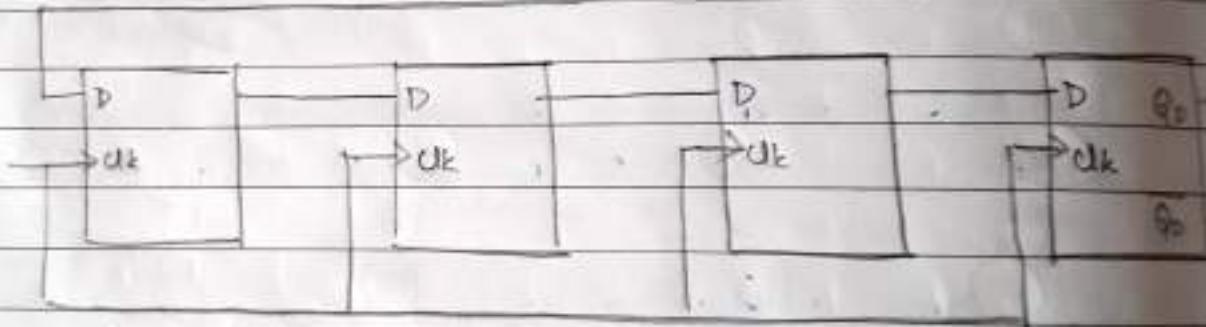
At least one of the flip flop should be loaded with 1, if all of them are zero then rotating cannot be noticed used in control sequence, i.e., time taken for one complete rotation is considered.

initial	1	0	0	0
1 st cycle	0	1	0	0
2 nd cycle	0	0	1	0
3 rd cycle	0	0	0	1
4 th cycle	1	0	0	0

application of shift register.

Loading one to one flip flop is a drawback which is overcome by Johnson counter.

* Johnson Counter:



Initially 0 0 0 0

1st 1 0 0 0

2nd 1 1 0 0

3rd 1 1 1 0

4th 1 1 1 1

5th 0 1 1 1

6th 0 0 1 1

7th 0 0 0 1

8th 0 0 0 0

N - bits

⇒ 2N clock cycles to get back to initial state

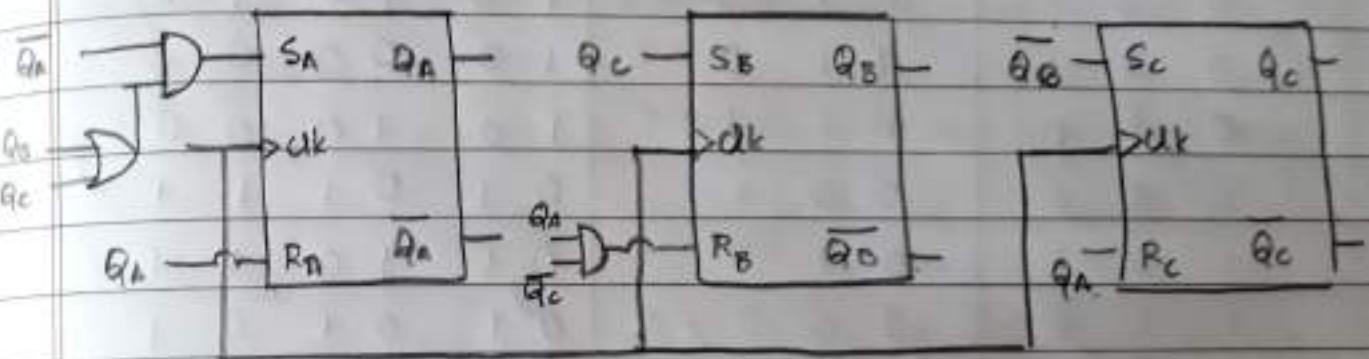
Intermediate sequence has no importance only the time taken to get back to initial stage is only considered.

classmate

Date _____
Page _____

Q:

Find the sequence.



Q_A	Q_B	Q_C	Q_A^+	Q_B^+	Q_C^+	S_A	R_A	S_B	R_B	S_C	R_C
0	0	0	0	0	1-1	0	0	0	0	1	0
0	0	1	1	1	1-1	1	0	1	0	1	0
0	1	0	1	1	0-6	1	0	0	0	0	0
0	1	1	1	1	1-1	1	0	1	0	0	0
1	0	0	0	0	x	0	1	0	1	1	1
1	0	1	0	1	x	0	1	1	0	1	1
1	1	0	0	0	0-0	0	1	0	1	0	1
1	1	1	0	1	0-2	0	1	1	0	0	1

Hence the given sequence is $1 \rightarrow 7 \rightarrow 2 \rightarrow 6 \rightarrow 0 \rightarrow 4$

* Universal Shift Register:

4 operations:

- retain previous output
- shift right
- shift left
- load

It has four modes

00 - store

01 - shift right

10 - shift left

11 - load

application specific integrated circuits (ASICs)

They are designed for a particular application only.
cannot be used in any other application.

Disadvantages:

- Not programmable (confined to one application only)

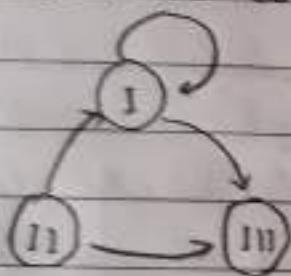
- Expensive when manufactured in small scale.

Advantage:

- Faster as once programmed reducing damage and delay.

* State Machine:

It is used to have a state with multiple decisions
it helps to have conditions for it



Here state I has two decisions to make hence state machine helps in having condition for which it has to choose which path of flow.

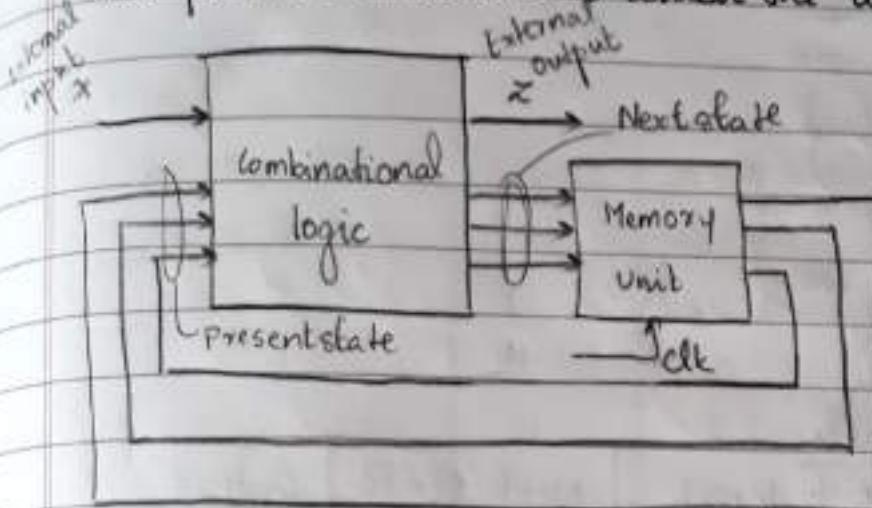
UNIT - 4 and 5

Analysis and Design of Clocked Sequential Circuits

Derivation of State Graphs and Tables

* states:

All possible conditions at which the device works.



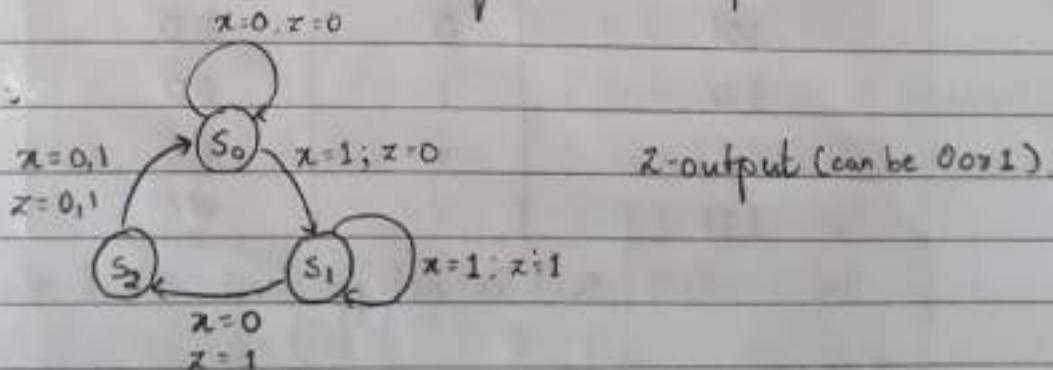
Generalised Model
of state Machine
Diagram

State machine decides the output based on the input, that is deciding the state based on certain conditions.

Designing can be done in two ways:

1. Mealy model

2. Moore model: has a certain sequence so independent of current input



General State Diagram

Mealy model is same as general model.

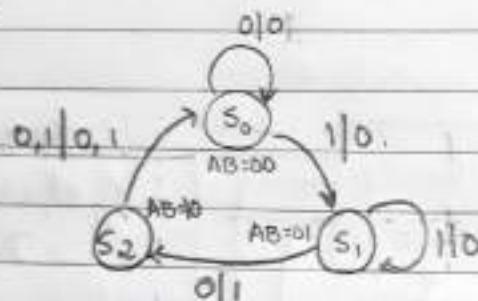
When the state itself is the output it is a Moore model.
(counters)

Any design with only one state - combinational logic

Any design with more than one state - state machine

For a Mealy model the output is given for the present state or the given state and input conditions
 Moore model, the output is defined for a state irrespective of the input conditions.

* Mealy State Graph



Mealy State Table

Present State AB	Input x	Next State $A^+ B^+$	Output z
00	0	00	0
00	1	01	0
01	0	10	1
01	1	01	0
10	0	00	0
10	1	00	1
11	x	xx	0
11	x	xx	0

For a state that is not present there should be no output.

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

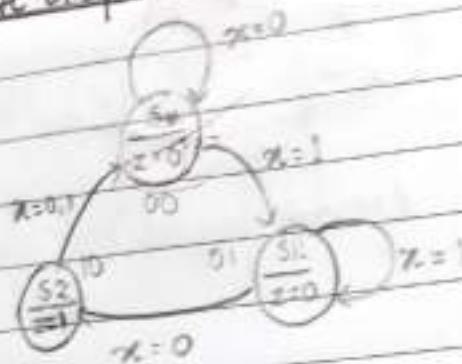
$$z = \bar{A} \bar{B} \bar{x} + A \bar{B} x$$

$$z = \bar{A} B \bar{x} + A \bar{B} x$$

$$z = (\bar{A} B + A \bar{B}) x$$

$$z = (A \oplus B) x$$

* Moore State Graph:



Moore State Table

Present State AB	Input x	Next State $A^+ B^+$	Output Z
00	0	00	0
00	1	01	0
01	0	10	0
01	1	01	a
10	0	00	1
10	1	00	1
11	x	xx	0. \downarrow False triggering
11	x	xx	0

$Z = Bx$	00	01	11	10
A	0	0	0	0
0	0	1	1	0
1	1	0	0	0

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

(independent of present input once the sequence is carried out).

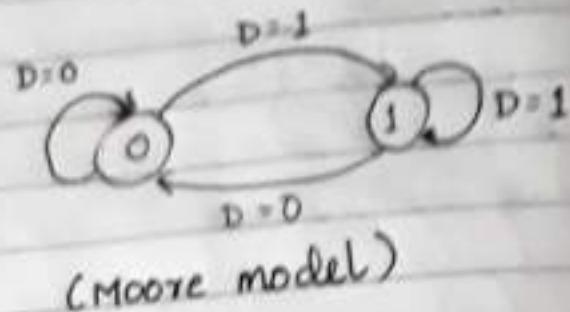
* State diagram of Flip Flops:

All flip flops have only two states.
States are 0 or 1.

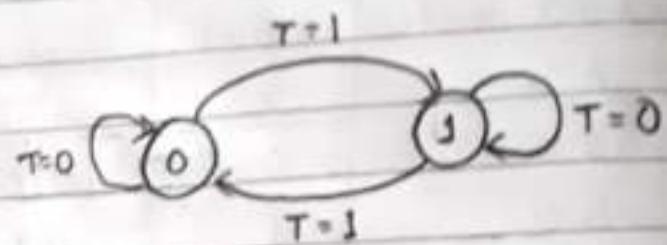
All flip flops are Moore model only as the states are output itself.

D Flip Flop:

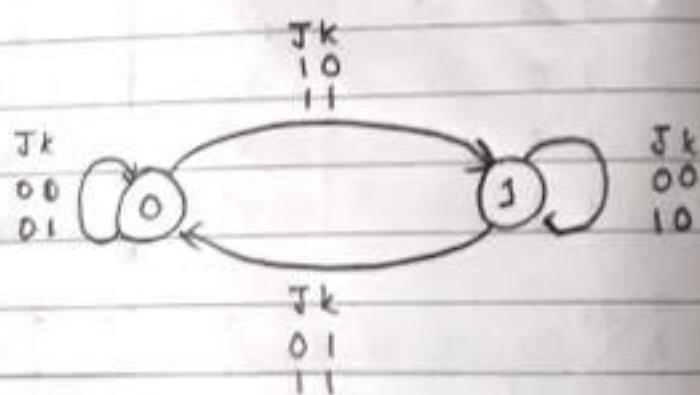
Q_n	Q_{n+1}	D
0	0	0
0	1	1
1	0	0
1	1	1

T Flip Flop:

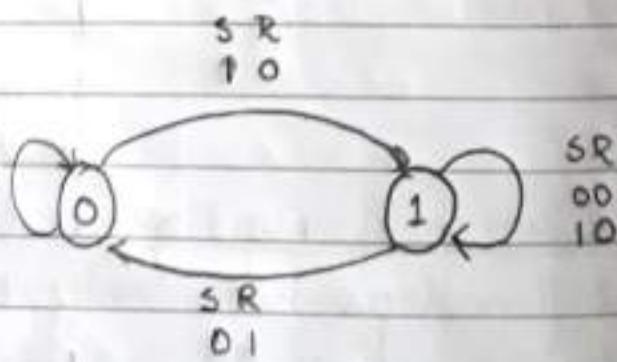
Q_n	Q_{n+1}	T
0	0	0
0	1	1
1	0	1
1	1	0

JK Flip Flop:

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

SR Flip Flop:

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



Q:

A sequential circuit has one input and one output, the state diagram is given below. Design a sequential circuit.

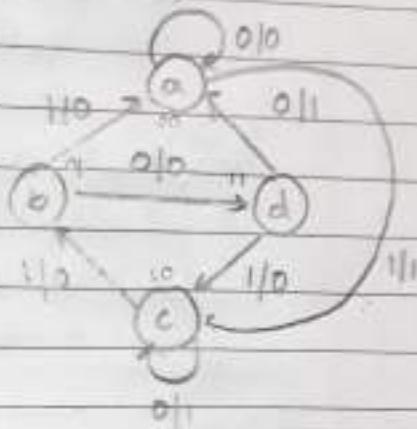
- Using T flip flop
- Using SR flip flop

Sol:~~State transition~~

Mealy sequential circuit

State Table:

Present State	Next State	Output
$Q_1 \ Q_2$	$Q_1^+ \ Q_2^+$ when $x=0$	$Q_1^+ \ Q_2^+$ when $x=1$
00 \rightarrow a	a	c 0 1
01 \rightarrow b	d	a 0 0
10 \rightarrow c	c	b 1 0
11 \rightarrow d	a	c 1 0



4 states

 $4 = 2^{(2)} - \text{Number of flip flops}$

i. Transition Table: (using T flip flop)

Present state	Input	Next state	Flip Flop input	Output
$Q_1 \ Q_2$	x	$Q_1^+ \ Q_2^+$	$T_1 \ T_2$	z
00	0	00	00	0
00	1	10	10	1
01	0	11	10	0
01	1	00	01	1
10	0	10	00	0
10	1	01	11	1
11	0	00	11	0
11	1	10	01	1

\bar{Q}_1	\bar{Q}_2	00	01	11	10
0	0	1	0	1	1
1	0	1	0	1	1

$$T_1 = \bar{Q}_2 z + Q_2 \bar{z}$$

$$T_1 = Q_2 \oplus z$$

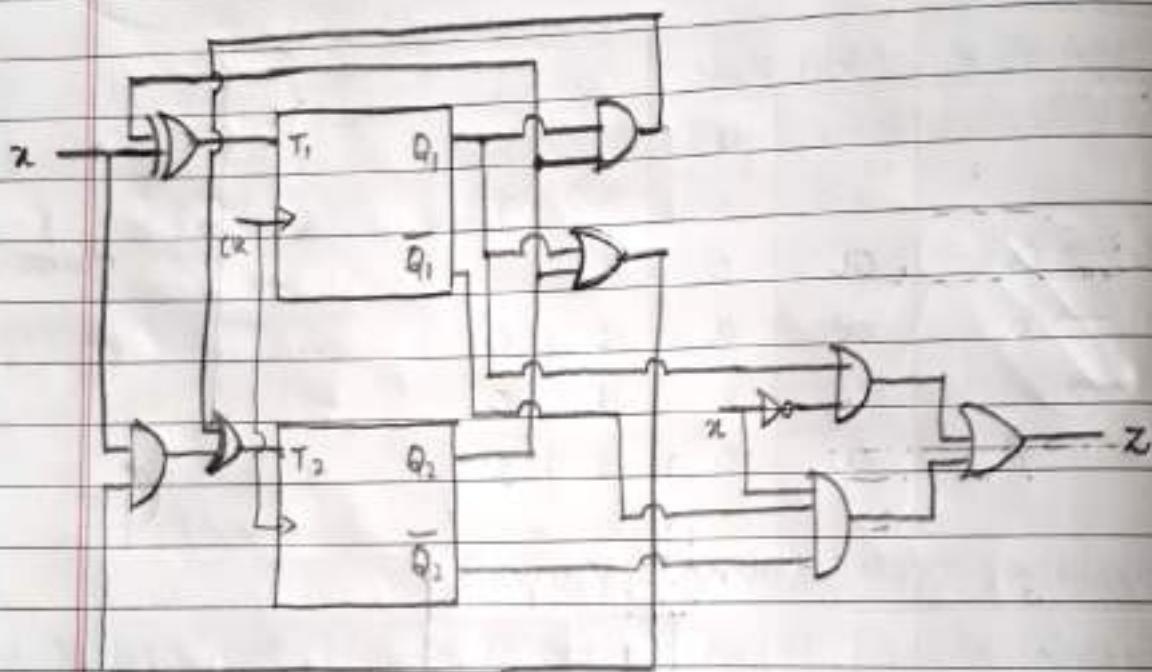
\bar{Q}_1	\bar{Q}_2	00	01	11	10
0	0	0	1	0	0
1	0	1	0	1	1

$$T_2 = Q_1 z + Q_1 Q_2 + Q_2 z$$

$$T_2 = Q_1 Q_2 + z(Q_1 + Q_2)$$

\bar{Q}_1	\bar{Q}_2	00	01	11	10
0	0	1	0	0	0
1	1	0	0	1	1

$$z = \bar{Q}_1 \bar{Q}_2 z + Q_1 \bar{z}$$



ii) Transition Table (using SR flip flop):

Present State Q_1, Q_2	Input x	Next State Q_1^+, Q_2^+	Flip Flop Inputs S, R, S_2, R_2	Output z
0 0	0	0 0	0 x 0 x	0
0 0	1	1 0	1 0 0 x	1
0 1	0	1 1	1 0 x 0	0
0 1	1	0 0	0 x 0 1	0
1 0	0	1 0	x 0 0 x	1
1 0	1	0 1	0 1 1 0	0
1 1	0	0 0	0 1 0 1	1
1 1	1	1 0	x 0 0 1	0

S_1	$Q_2 \oplus x$	00	01	11	10
Q_1		0	1	0	1
0		0	1	0	1
1		x	0	x	0

$$S_1 = \bar{Q}_1 \bar{Q}_2 x + \bar{Q}_1 Q_2 \bar{x}$$

$$S_1 = \bar{Q}_1 (Q_2 \oplus x)$$

R_1	$Q_2 \oplus x$	00	01	11	10
Q_1		0	x	0	x
0		0	1	0	0
1		0	1	0	1

$$R_1 = Q_1 \bar{Q}_2 x + \bar{Q}_1 Q_2 \bar{x}$$

$$R_1 = Q_1 (Q_2 \oplus x)$$

S_2	$Q_1 \oplus x$	00	01	11	10
Q_1		0	0	0	x
0		0	0	0	x
1		0	1	0	0

$$S_2 = Q_1 \bar{Q}_2 x$$

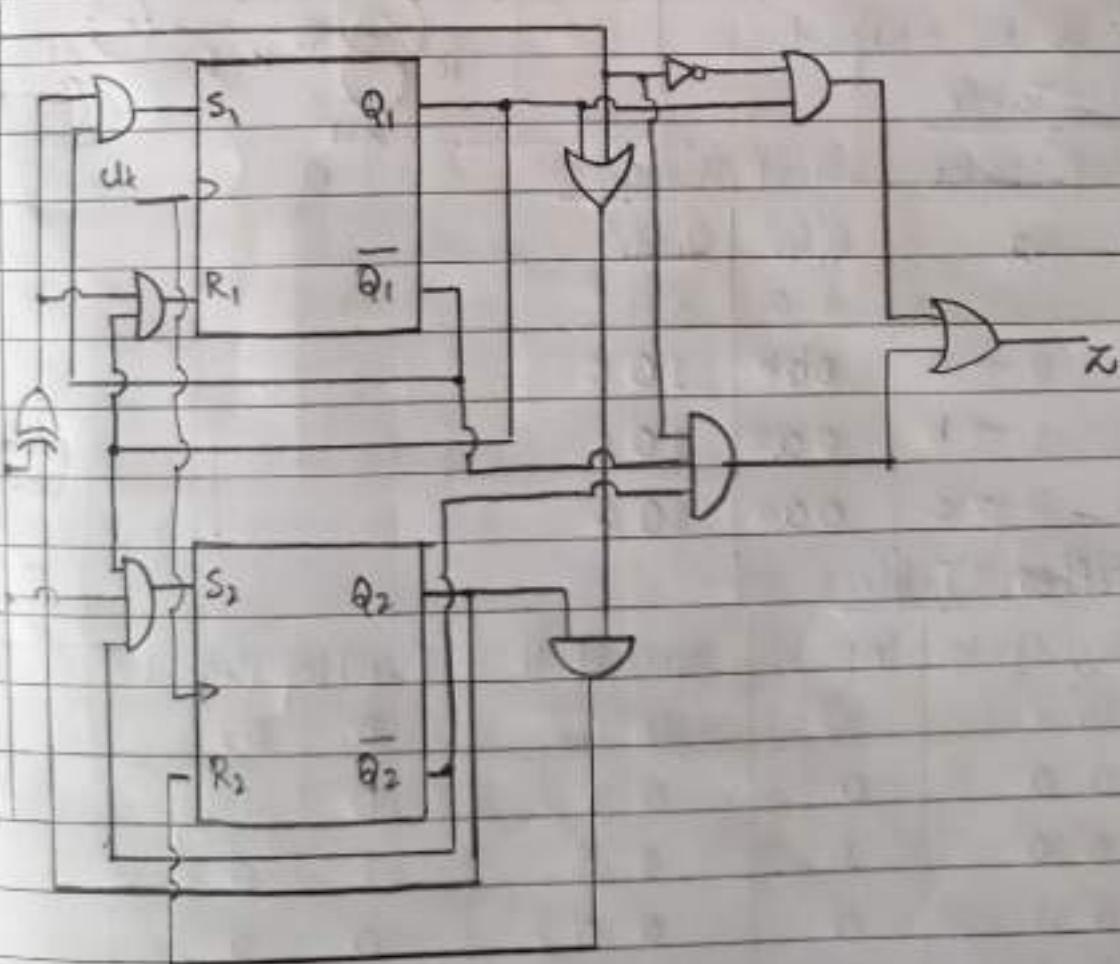
$$R_2 = Q_2 x + \bar{Q}_1 Q_2$$

$$R_2 = Q_2 (Q_1 + x)$$

Z	$Q_1 \oplus x$	00	01	11	10
Q_1		00	01	11	10
0		0	1	0	0
1		1	0	0	1

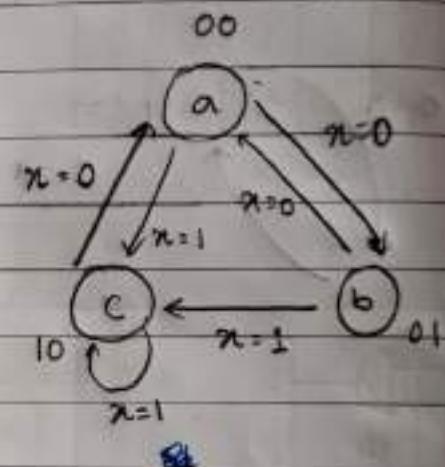
$$x = \bar{Q}_1 \bar{Q}_2 x + Q_1 \bar{x}$$

x



Q: A clocked sequential circuit has three states, a , b , and c . As long as $x = 0$, the circuit alternates between states A and B . If x becomes 1 either its in state A or state B , the circuit goes to state C and remains in state C as long as $x = 1$. The circuit goes to state A if x becomes 0 again and then repeats the behaviour. Assume state assignments as $A = 00$, $B = 01$, and $C = 10$. Draw the state diagram and prepare state table and draw the circuit using D flip flop.

State diagram

Sol:

Moore Mealy Sequential circuit

State Table:

Present ~~110~~ output is nowhere mentioned hence the state itself is the output.

State Table:

Present States		Next State	
Q_1 Q_2		$Q_1^+ Q_2^+$	$Q_1^+ Q_2^+$
	x	$x=0$	$x=1$
0 0	0	b	c
0 1	1	a	c
1 0	0	a	c

Transition Table:

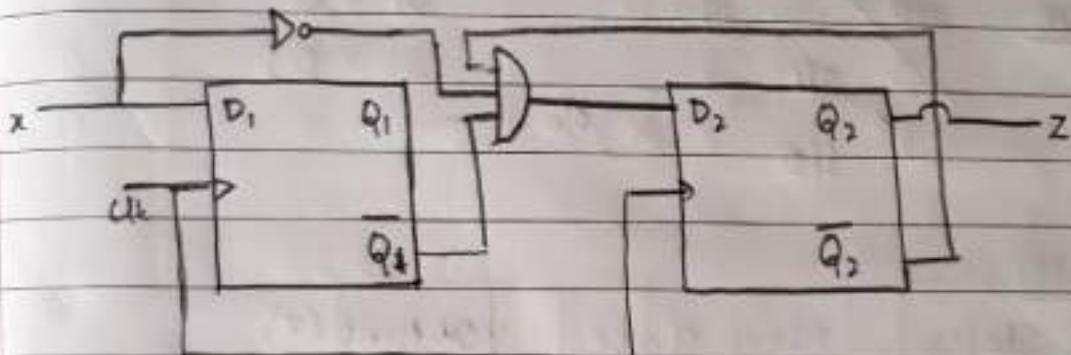
Present State	Input	Next State	Flip Flop Input
$Q_1 Q_2$	x	$Q_1^+ Q_2^+$	$D_1 D_2$
0 0	0	0 1	0 1
0 0	1	1 0	1 0
0 1	0	0 0	0 0
0 1	1	1 0	1 0
1 0	0	0 0	0 0

Q_1, Q_2	x	Q_1^+, Q_2^+	D_1, D_2
1 0	• 1	1 0	1 0
1 1	x	x x	x x
1 1	x	x x	x x

$D_1, Q_2 x$	00 01 11 10	$D_2, Q_2 x$	00 01 11 10
Q ₁	0 0 1 1 0	Q ₁	0 (1) 0 0 0
1 0	1 x x x x	1 0 0 x x x	

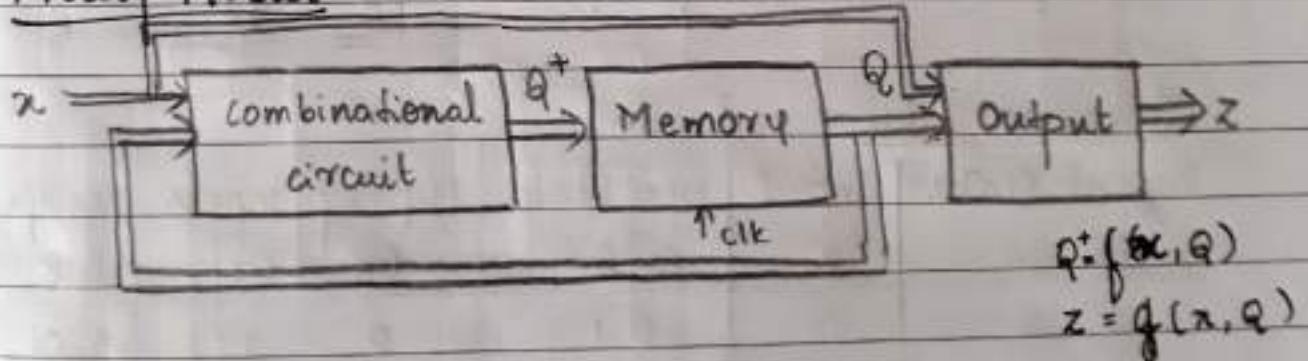
$D_1 = x$

$D_2 = \bar{Q}_1 \bar{Q}_2 \bar{x}$

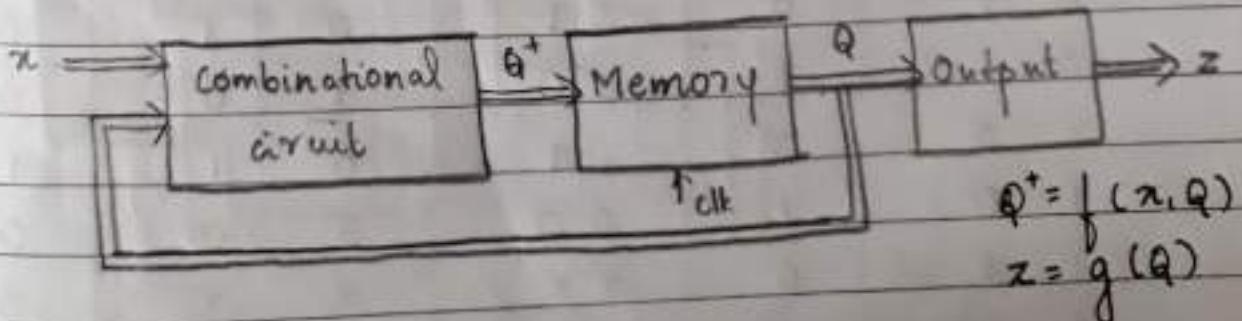


* Block Diagram:

Mealy Model



Moore Model



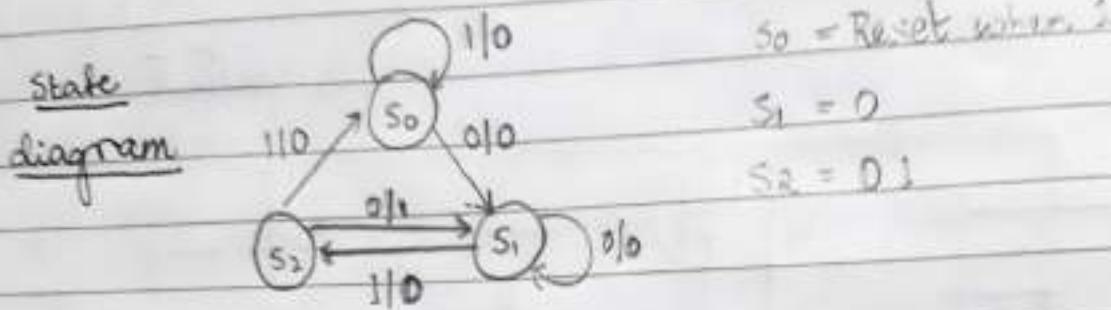
* Pattern Detection | Sequence Detection:

To detect : 010

$x = 00011101000111010$

$z = 0000000010100000001$

Number of states = Number of bits to be detected.
Here number of states = 3 (S_0, S_1, S_2)



State Table:

Present States	Next State		Output (z)	
	$x=0$	$x=1$	$x=0$	$x=1$
S_0	S_1	S_0	0	0
S_1	S_1	S_2	0	0
S_2	S_1	S_0	1	0

Transition Table:

Present states $Q_A\ Q_B$	Input x	Next State $Q_A^+\ Q_B^+$	FlipFlop Inputs $D_A\ D_B$	Outputs. z
0 0	0	0 1	0 1	0
0 0	1	0 0	0 0	0
0 1	0	0 1	0 1	0
0 1	1	1 0	1 0	0
1 0	0	0 1	0 1	0
1 0	1	0 0	0 0	1
1 1	0	$\times\ \times$	$\times\ \times$	0
1 1	1	$\times\ \times$	$\times\ \times$	0

D_A	$Q_B \bar{x}$	00	01	11	10
0	0	0	1	0	
1	0	0	x	x	

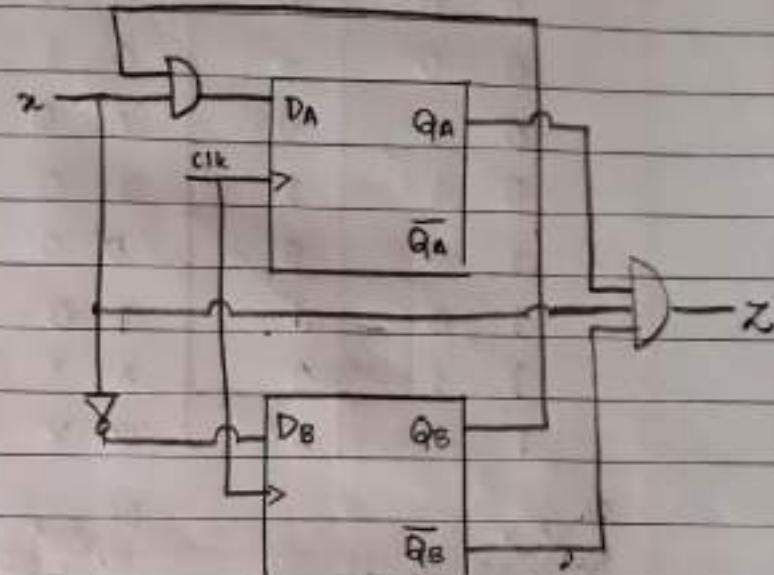
$$D_A = Q_B \bar{x}$$

D_B	$Q_B \bar{x}$	00	01	11	10
0	1	0	0		1
1	1	0	x	x	

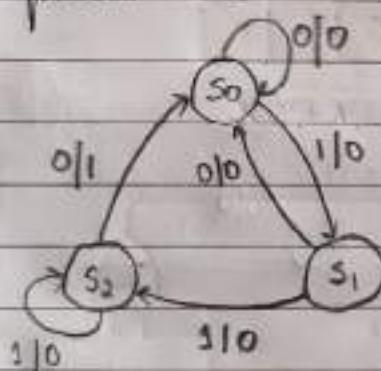
$$D_B = \bar{x}$$

$Q_A \bar{x}$	00	01	11	10
0	0	0	0	0
1	1	0	0	0

$$\bar{x} = Q_A \bar{Q}_B \bar{x}$$



Sequence: 110



State Diagram

State table:

Present State	Next State		Output (z)	
	$\bar{x}=0$	$\bar{x}=1$	$\bar{x}=0$	$\bar{x}=1$
S ₀	S ₀	S ₁	0	0
S ₁	S ₀	S ₂	0	0
S ₂	S ₀	S ₂	1	0

Transition Table:

Present States	Input	Next State	Flip flop inputs	Outputs
$Q_A \ Q_B$	x	$Q_A' \ Q_B'$	$D_A \ D_B$	z
0 0	0	0 0	0 0	0
0 0	1	0 1	0 1	0
0 1	0	0 0	0 0	0
0 1	1	1 0	1 0	0
1 0	0	0 0	0 0	1
1 0	1	1 0	1 0	0
1 1	0	x x	x x	0
1 1	1	x x	x x	0

D_A	$Q_A \bar{Q}_B x$	00 01 11 10
0	0 0 0 1 0	00 01 11 10
1	0 1 x x	x x x x

D_B	$\bar{Q}_A \bar{Q}_B x$	00 01 11 10
0	0 0 1 0 0	00 01 11 10
1	0 0 x x	x x x x

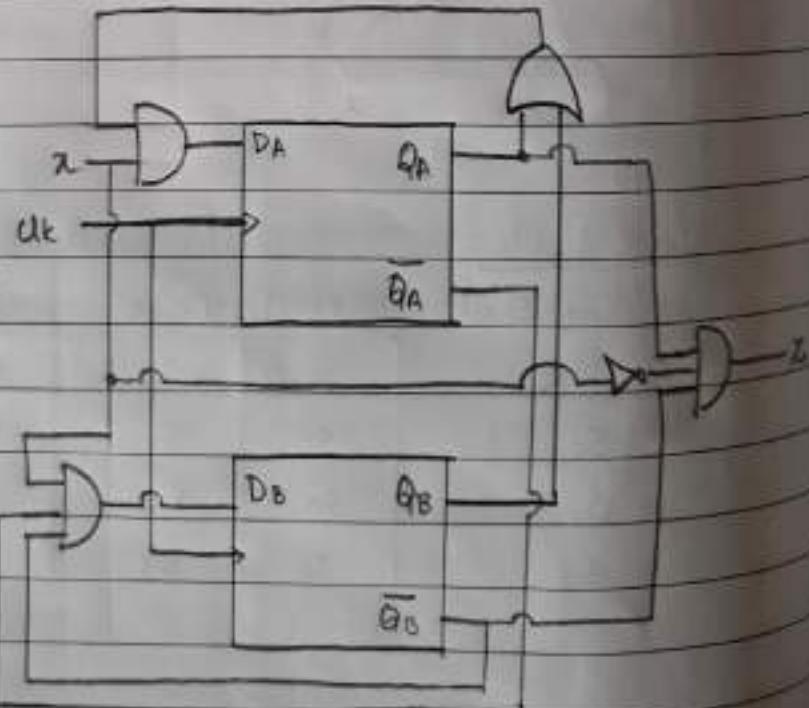
$$D_A = Q_A x + Q_B x$$

$$D_B = \bar{Q}_A \bar{Q}_B x$$

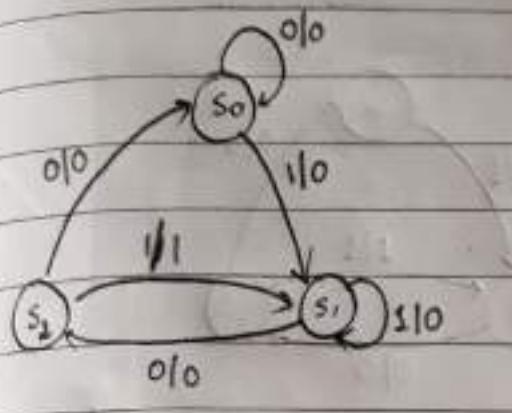
$$D_A = (Q_A + Q_B) x$$

Z	$Q_A \bar{Q}_B x$	00 01 11 10
0	0 0 0 0 0	00 01 11 10
1	1 0 0 0 0	x x x x

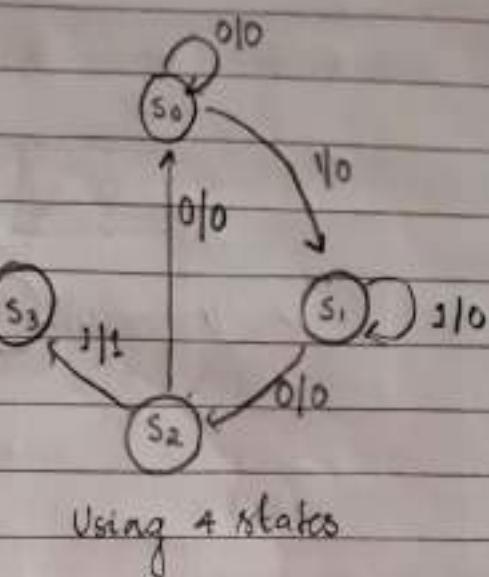
$$Z = Q_A \bar{Q}_B x$$



- Sequence: 101

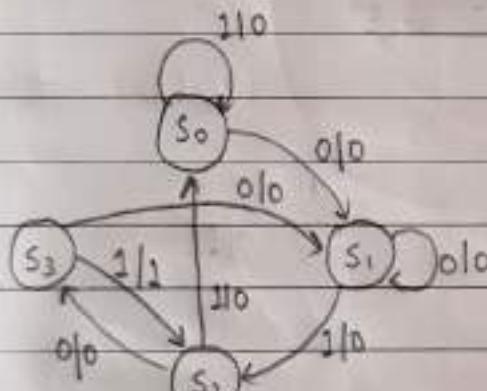


using 3 states

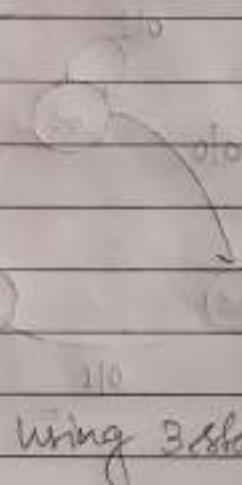


Using 4 states

- Sequence: 0101



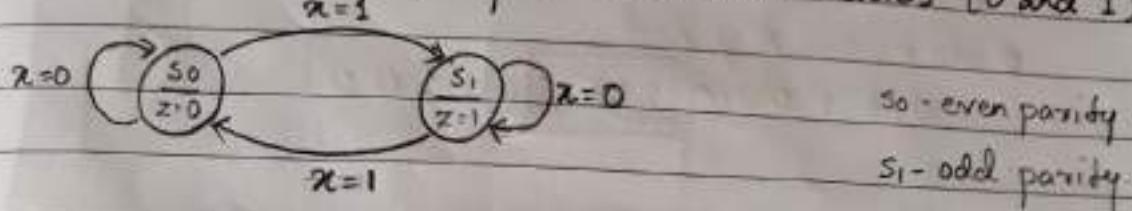
Using 4 states



Using 3 states

Parity Checker:

- Error detector : It can detect only one bit error.
- Moore model is employed. Hence two states (0 and 1).

State table:

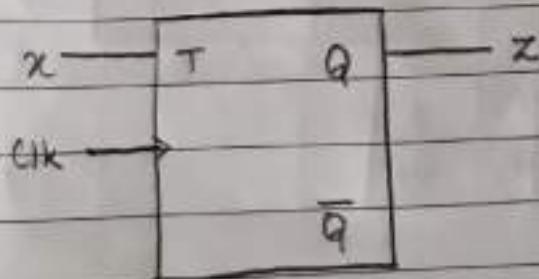
Present State	Next State		Output	
	$x=0$	$x=1$	$x=0$	$x=1$
S_0	S_0	S_1	0	0
S_1	S_1	S_0	1	1

Present state output is considered in Moore's model

Transition table:

Present State	Input	Next State	FlipFlop Input	Output
Q	x	Q^+	T	z
0	0	0	0	0
0	1	1	1	0
1	0	1	0	1
1	1	0	1	1

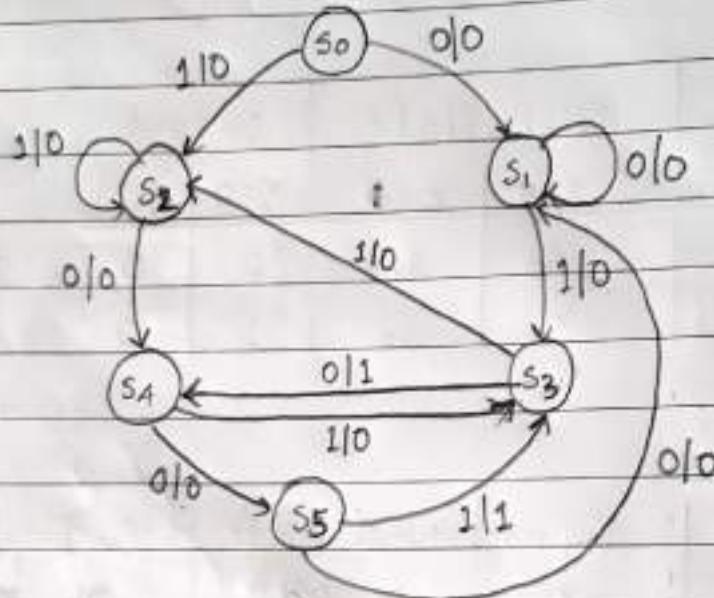
T	x	Q	z
0	0	0 1	$T = x$
1	0 1	0 0 0	$z = Q$



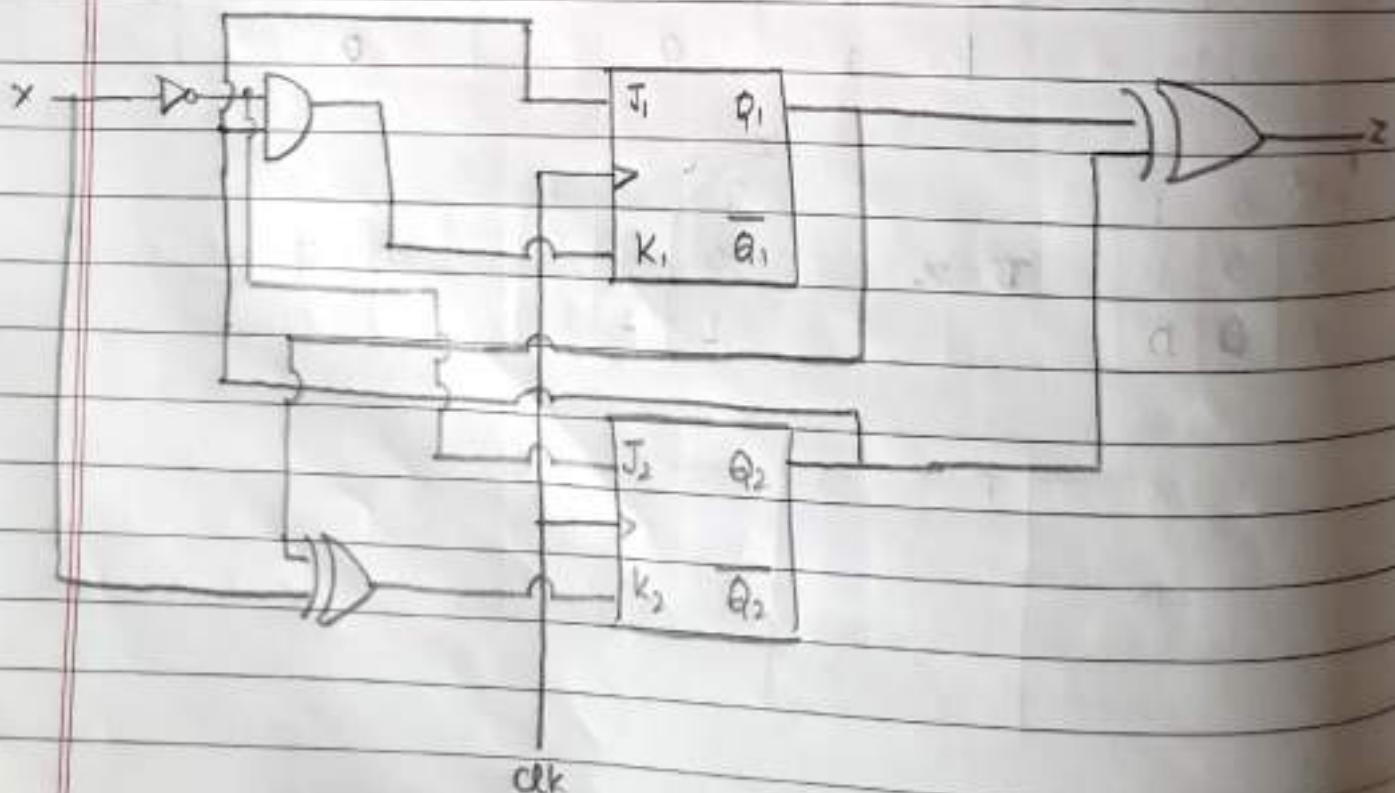
* Overlapping sequence

To detect 1001 and 010 simultaneously
6 states are required.

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



Q: Analyse the given circuit.



$$\begin{array}{lll} J_1 = Q_2 & K_1 = Q_2 \bar{x} & z = Q_1 \oplus Q_2 \\ J_2 = \bar{x} & K_2 = Q_1 \oplus x \end{array}$$

x independent of z — Moore model
Characteristic equation

$$\begin{aligned} Q_1^+ &= J_1 \bar{Q}_1 + \bar{K}_1 Q_1 \\ &= Q_2 \bar{Q}_1 + (\bar{Q}_2 \bar{x}) Q_1 \\ &= Q_2 \bar{Q}_1 + (\bar{Q}_2 + x) Q_1 \\ &= Q_2 \bar{Q}_1 + Q_1 \bar{Q}_2 + Q_1 x \\ &= Q_1 \oplus Q_2 + Q_1 x \end{aligned}$$

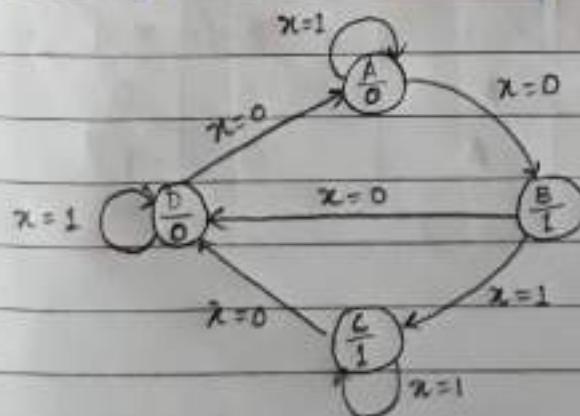
$$\begin{aligned} Q_2^+ &= J_2 \bar{Q}_2 + \bar{K}_2 Q_2 \\ &= \bar{x} \bar{Q}_2 + (\bar{x} \bar{Q}_1 + x \bar{Q}_1) Q_2 \\ &= \bar{x} \bar{Q}_2 + (\bar{x} \bar{Q}_1 + x Q_1) Q_2 \\ &= \bar{x} \bar{Q}_2 + \bar{x} \bar{Q}_1 Q_2 + x Q_1 Q_2 \end{aligned}$$

Transition Table

Present state	Next state	Output
$Q_1 \quad Q_2$	$Q_1^+ Q_2^+$	z
$z=0$	$z=0$	
0 0	0 1	0
0 1	1 1	1
1 0	1 1	1
1 1	0 0	0

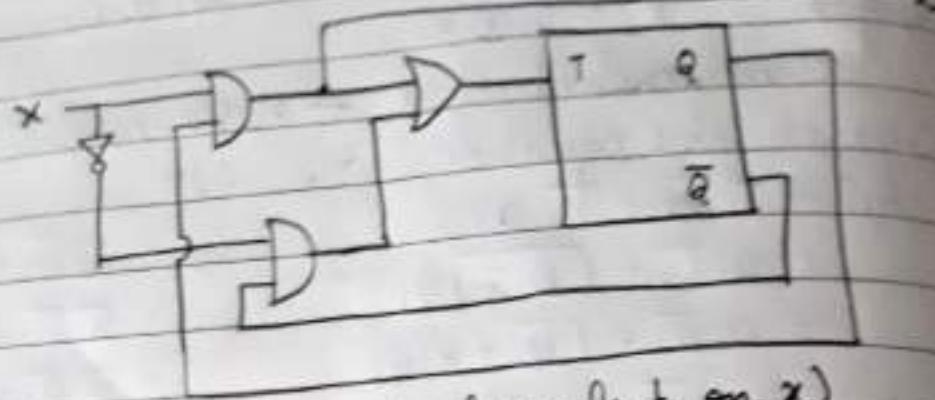
State table

Present state	Next state	Output
$Q_1 \quad Q_2$	$Q_1^+ Q_2^+$	z
• •	$z=0 \quad z=1$	
A \rightarrow 0 0	B A	0
B \rightarrow 0 1	D C	1
C \rightarrow 1 0	D C	1
D \rightarrow 1 1	A P	0



state diagram

Q:



Mealy model (because Z is dependent on x)
 Input expression Output expression
 $T = xQ + \bar{x}\bar{Q}$ $Z = xQ$

$$Q^+ = T \oplus Q$$

$$Q^+ = x \oplus Q \oplus Q$$

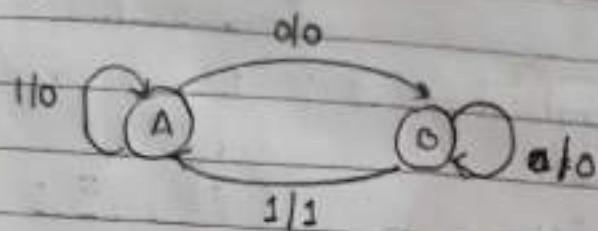
Transition table

Present State	Next State		Output	
	Q	Q^+	$x=0$	$x=1$
0	1	0	0	0
1	1	0	0	1

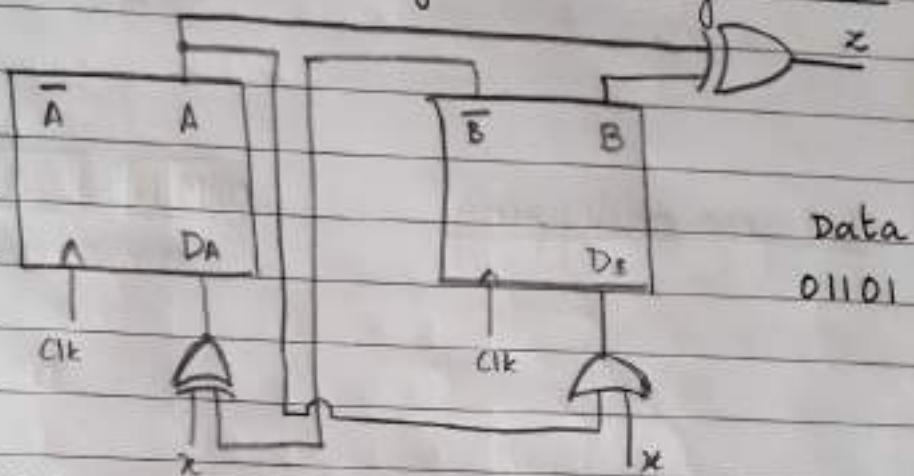
State table

Present State	Next State		Output	
	Q	Q^+	$x=0$	$x=1$
A $\rightarrow 0$	B	A	0	0
B $\rightarrow 1$	B	A	0	1

state diagram:



Analysis of Signal Tracing and Timing Charts:

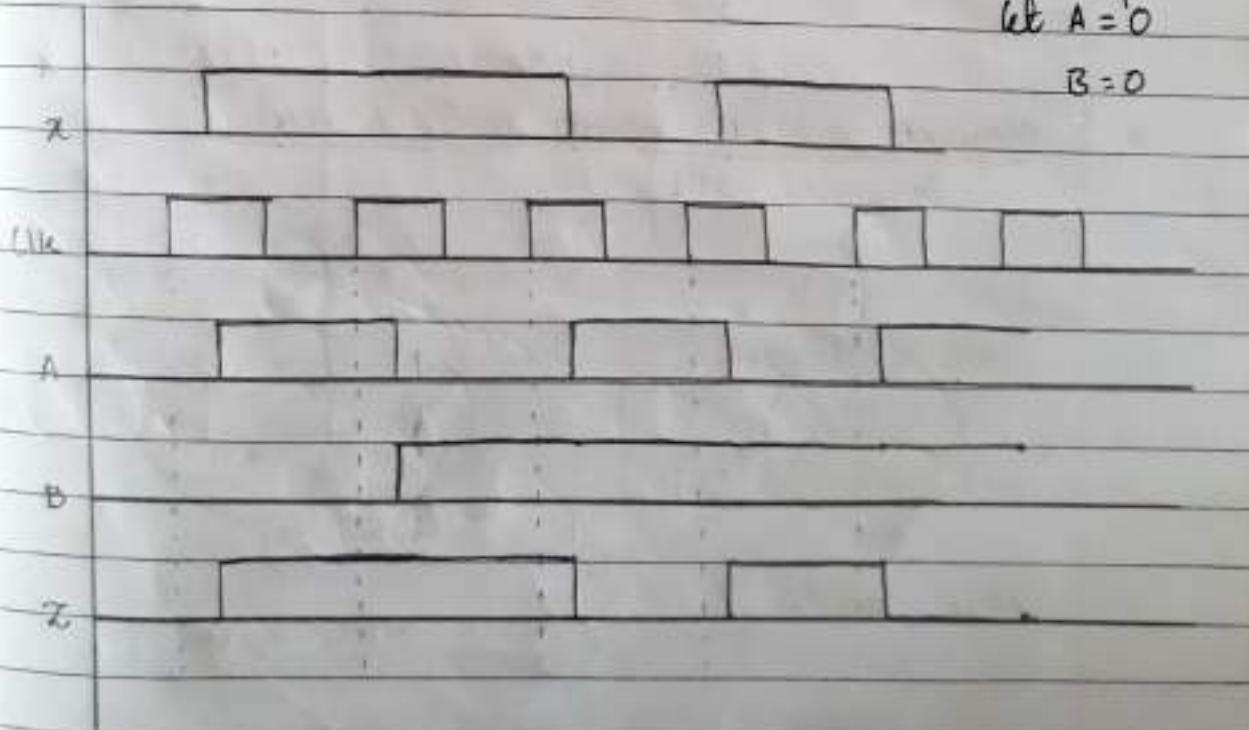


Moore model : $z = A \oplus B$

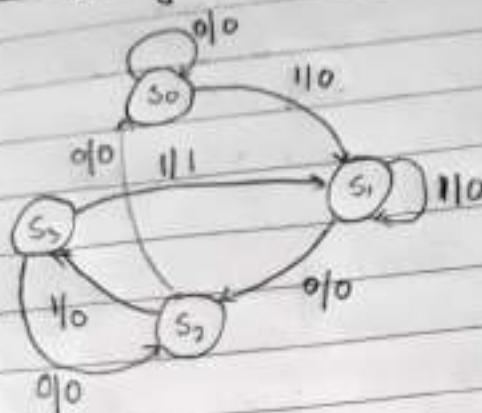
Timing diagram

Initially
let $A = 0$

$B = 0$

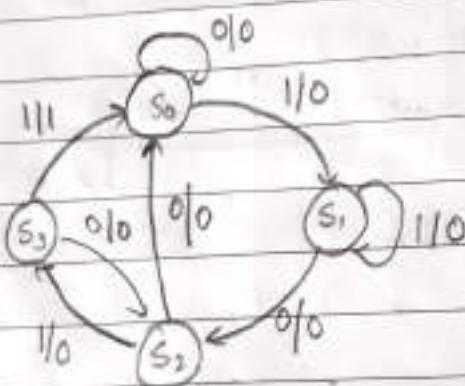


* 1011 - overlapping:



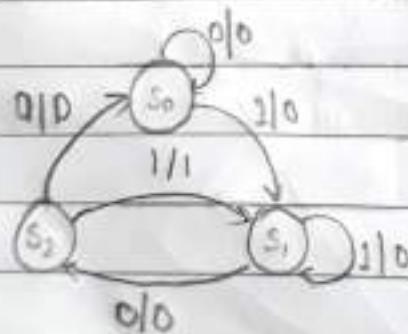
$01\boxed{1011}01100101$
overlapping
(usually considered)

* 1011 - non overlapping:

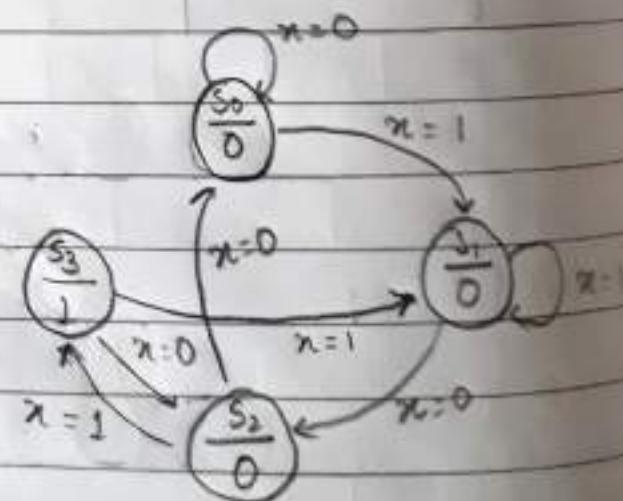


$01\boxed{1011}011001011$
non overlapping

* Sequence detection using moore's model:
Sequence 101

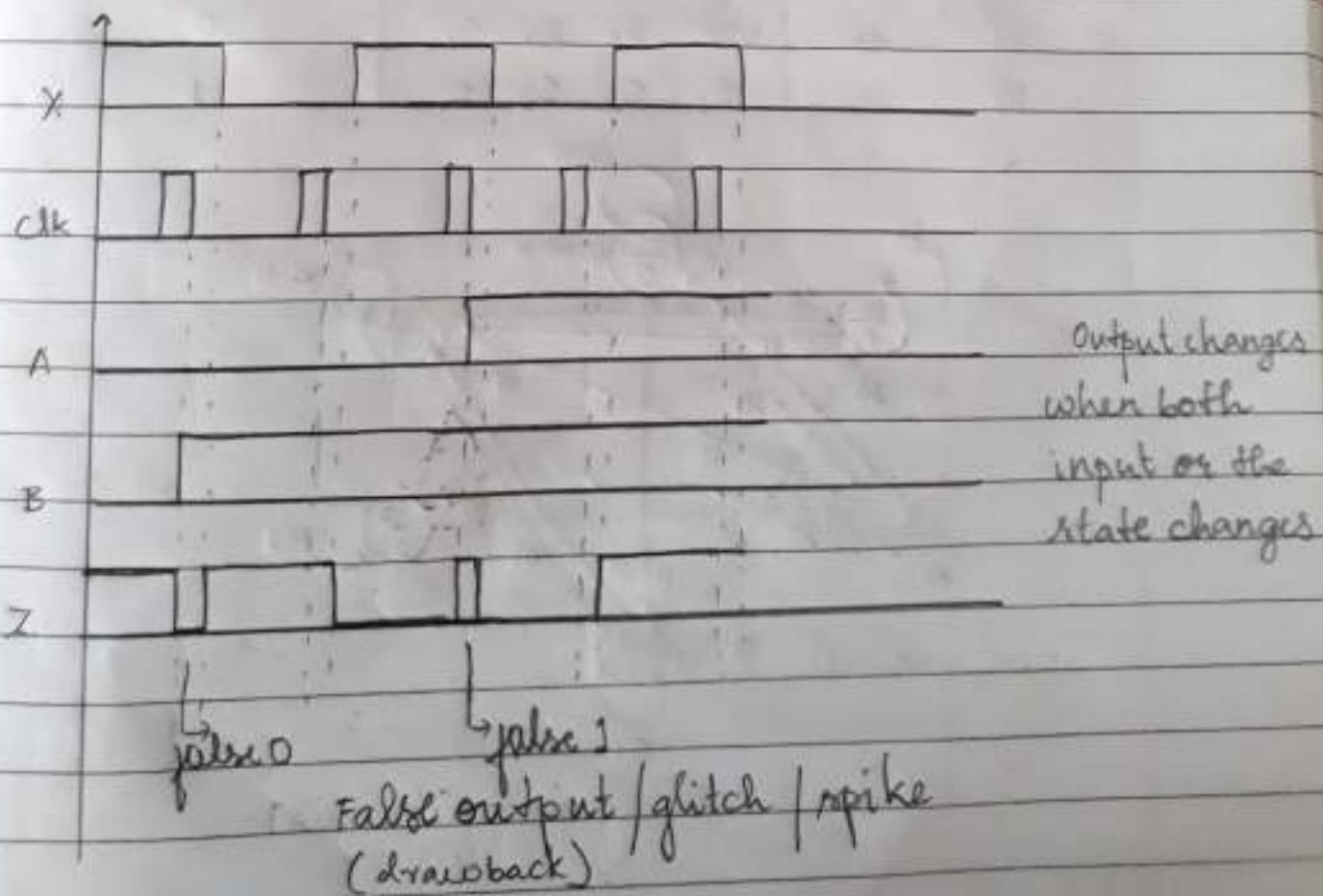
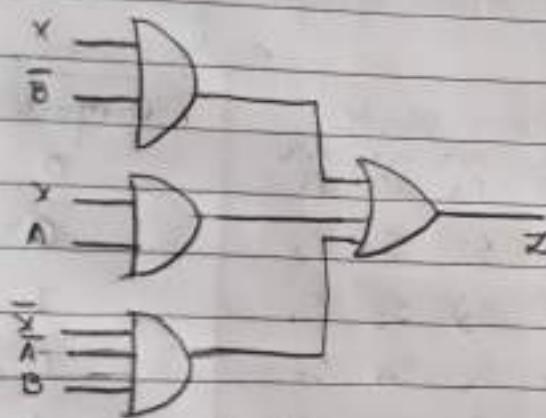
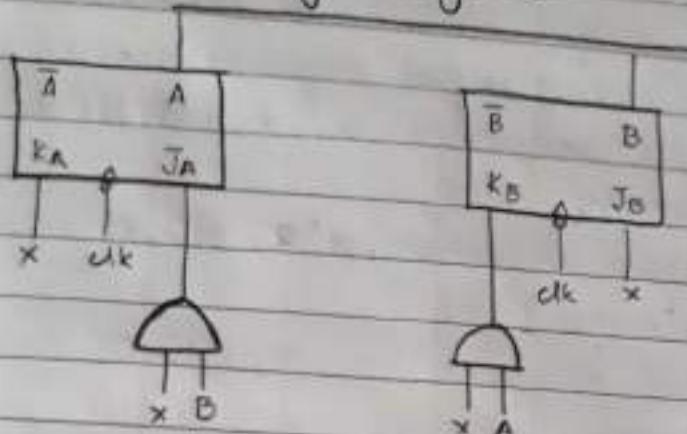


Mealy model



Moore model

Mealy circuit timing diagram:



IMP

Q:

A sequential circuit has two inputs x_1 and x_2 and one output z . The output remains a constant value unless one of the following input sequence occurs.

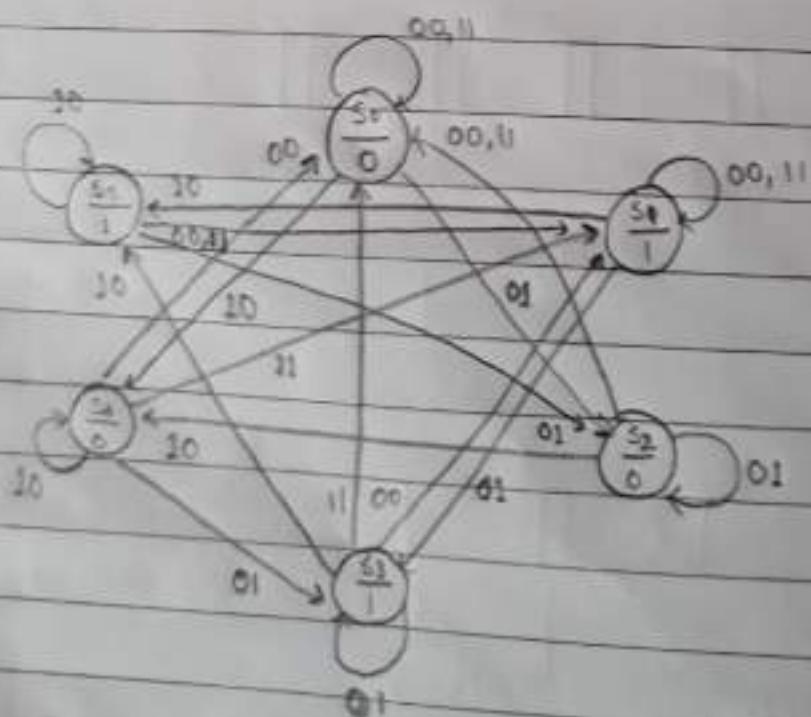
a. the input sequence $x_1, x_2 = 01, 11$ causes the output to become zero.

b. the input sequence $x_1, x_2 = 10, 11$ causes the output to become 1.

c. the input sequence $x_1, x_2 = 10, 01$ causes the output to change value.

Derive a Moore state graph for the circuit

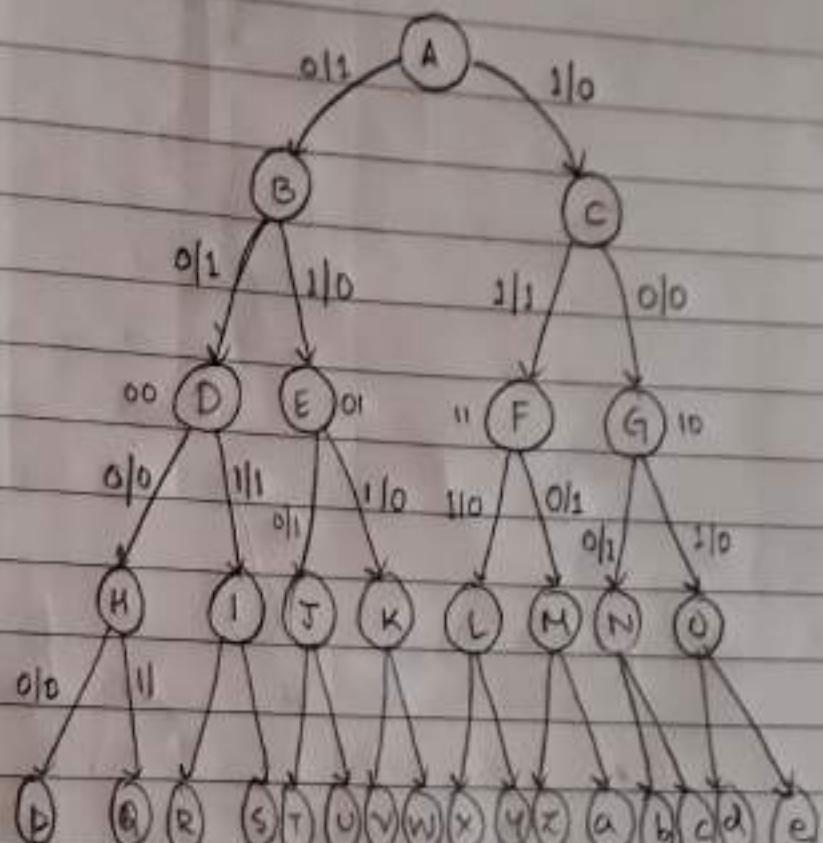
sd1:	Present State	Next State				Output (z)
		x_1, x_2	x_1, x_2	x_1, x_2	x_1, x_2	
Non-combination leads with 0 or 1 no 10 or 11	'00' or '11' s_0	s_0	s_2	s_4	s_0	0
watched initial state	'00' or '11' s_1	s_1	s_3	s_5	s_1	1
	'01'	s_2	s_0	s_2	s_4	0
	'10'	s_3	s_1	s_3	s_5	1
	'10'	s_4	s_0	s_3	s_4	0
	'10'	s_5	s_1	s_2	s_5	1



BCD
0000
0001
0010
0011
0100
0101
0110
0111
1000
1001

Excess-3
0011
0100
0101
0110
0111
1000
1001
1010
1011
1100

Mealy model



Incomplete

UNIT - 6

Reduction of State Tables and State Assignments

State reduction

- reduces number of gates used (ie hardware)
- reduces number of flipflops.

It is done by two methods :-

1. State reduction table
2. Implication table

+ State Reduction Table:

Present State	Next State		Output (z)	
	$x=0$	$x=1$	$x=0$	$x=1$
a	f	b	0	0
b	d	c	0	0
c	f	e	0	0
d	g	a	1	0
e	d	c	0	1
f	f	b	1	1
g	g	h	0	1
x h	g	a	1	0

Check for common next state and common output.

$d \equiv h$ now replace h by d

Now there are 7 states from 8 states

Present State	Next State		Output(x)	
	$x=0$	$x=1$	$x=0$	$x=1$
a	f	b	0	0
b	d	c	0	0
c	f	e	0	0
d	g	a	1	0
e	d	c	0	1
f	f	b	1	1
g	g	d	0	1

* Implication Table:

Present State	Next State		Output
	$n=0$	$n=1$	
a	d	c	0
b	f	h	0
c	e	d	1
x	d	a	0
x	e	c	1
f	f	b	1
g	b	h	0
h	c	g	1

→ Step 1

b	d-f c-h					
c	x	x				
d	a-d c-e	a-f e-h	x			
e	x	x	e-e a-d	x		
f	x	x	e-f b-d	x	c-f a-b	
	b-g c-h	b-f	x	a-b e-h	x	x
	x	x	c-e d-g	x	a-g b-g	x
a	b	c	d	e	f	g

check for same
common output, then
the blocks are filled
with implied pairs
don't write self
implied pairs (i.e., $\text{1} \times \text{1}$)

→ Step 2 :

b	x					
c	x	x				
d	c-e	x	v			
e	x	x	a-d	x		
f	x	x	e-f b-d	x	c-f a-b	
g	c-h	x	x	a-b e-h	x	x
h	x	x	d-g	x	a-g	c-f b-g
	a	b	c	d	e	f

Since a-d is present in a-d it is not considered
similarly c-e is present in c-e hence not considered

→ Step 3 :

b	x					
c	x	x				
d	c-e	x	x			
e	x	x	a-d	x		
f	x	x	x	x	x	
g	x	x	x	x	x	x
h	x	x	c-e d-g	x	a-g	x
	a	b	c	d	e	f

→ Step 4 :

b	x					
c	x	x				
d	c-e	x	x			
e	x	x	a-d	x		
f	x	x	x	x	x	
g	x	x	x	x	x	x
h	x	x	x	x	x	x
	a	b	c	d	e	f

Hence $c \equiv e$ and $a \equiv d$
Therefore e is replaced by c and d is replaced by a.

Present state	Next state		Output
	$n=0$	$n=1$	
a	a	c	z
b	f	h	0
c	c	a	1
f	f	b	1
g	b	h	0
h	g	g	1

Now it is reduced to 6 states from 8 states