

Digital

Logic

Number System, Operations and code.

① Number System:

A number system is a set of symbols used to represent values derived from a common base or radix (r). Number system can be classified into four major categories.

- i) decimal number system
- ii) Binary number system
- iii) Octal number system
- iv) Hexadecimal number system

Number with different bases:

Decimal (base 10)	Binary (base 2)	Octal (base 8)	Hexadecimal (base 16)
0	0000	0	0 0 10 A
1	0001	1	1 1 11 B
2	0010	2	2 2 12 C
3	0011	3	3 3 13 D
4	0100	4	4 4 14 E
5	0101	5	5 5 15 F
6	0110	6	6 6
7	0111	7	7 7
8	1000	10	8 8
9	1001	11	9 9

1) Decimal Number System:

The term decimal is derived from a latin prefix deci which means ten. Decimal number system has ten digits ranging from 0-9. Because this system has ten digits it is also called a base ten number system or denary number system. A decimal number should always be written with a subscript \downarrow 10.

Eg. $(X)_{10}$.

2) Binary number System:

It uses two digits namely 1 and 0 to represent numbers. Unlike in decimal numbers where the place value goes up in factors of ten, in binary system the place values increase by the factor of 2. Binary number are written as $(X)_2$ such as $(1011)_2$.

3) Octal Number System:

Consist of eight digits ranging from 0-7. The place value of octal numbers goes up in factors of eight from right to left.

It is written as $(X)_8$. Example: $(1012)_8$.

4) Hexadecimal number system:

This is a base 16 number system that consists of sixteen digits ranging from 0-9 and letters A-F where A is equivalent to 10, B to 11 up to F which is equivalent to 15 in base ten system. The place value of hexadecimal number goes up in factors of sixteen.

A hexadecimal number can be denoted using 16 as a subscript or capital letter H to the right of the number.

For example: 94B can be written as $(94B)_{16}$ or 94BH

Conversions among different Number System

Perform the following conversions.

$$a) (101)_2 = N_{10}$$

$$\begin{aligned} &= 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 \\ &= 4 + 0 + 1 \\ &= (5)_{10} \text{ or } 5 \end{aligned}$$

$$(b) (11001)_2 = N_{10}$$

$$\begin{aligned} &= 1 \times 2^4 + 1 \times 2^3 + 0 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 \\ &= 16 + 8 + 0 + 0 + 1 \\ &= (25)_{10} \end{aligned}$$

$$c) (10011110)_2 = N_{10}$$

$$\begin{aligned} &\Rightarrow 1 \times 2^7 + 0 \times 2^6 + 0 \times 2^5 + 1 \times 2^4 + 1 \times 2^3 + 1 \times 2^2 + 1 \times 2^1 + 0 \times 2^0 \\ &\Rightarrow 128 + 0 + 0 + 16 + 8 + 4 + 2 + 0 \\ &\Rightarrow (158)_{10} \end{aligned}$$

d) $(101101)_2 = N_{10}$

$$\Rightarrow 1 \times 2^5 + 0 \times 2^4 + 1 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 \\ \Rightarrow 32 + 0 + 8 + 4 + 0 + 1 \\ \Rightarrow (45)_{10}$$

e) $(100)_10 = N_2$

$$\begin{array}{r} 100 \\ 2 | \quad 50 \quad - 0 \\ 2 | \quad 25 \quad - 0 \\ 2 | \quad 12 \quad - 1 \\ 2 | \quad 6 \quad - 0 \\ 2 | \quad 3 \quad - 0 \\ \hline 1 \quad - 1 \end{array}$$

$$(100)_{10} = (1100100)_2$$

f) $(225)_{10} = N_2$

$$\begin{array}{r} 225 \\ 2 | \quad 112 \quad - 1 \\ 2 | \quad 56 \quad - 0 \\ 2 | \quad 28 \quad - 0 \\ 2 | \quad 14 \quad - 0 \\ 2 | \quad 7 \quad - 0 \\ 2 | \quad 3 \quad - 1 \\ \hline 1 \quad - 1 \end{array}$$

$$(225)_{10} = (11100001)_2$$

g) $(129)_{10} = N_2$

$$\begin{array}{r} 129 \\ 2 | \quad 64 \quad - 1 \\ 2 | \quad 32 \quad - 0 \\ 2 | \quad 16 \quad - 0 \\ 2 | \quad 8 \quad - 0 \\ 2 | \quad 4 \quad - 0 \\ 2 | \quad 2 \quad - 0 \\ \hline 1 \quad - 0 \end{array}$$

$$(129)_{10} = (10000001)_2$$

* $(137)_{10} = (?)_2$

$$\begin{array}{r}
 2 | 137 \\
 2 | 68 - 0 \\
 2 | 34 - 0 \\
 2 | 17 - 0 \\
 2 | 8 - 1 \\
 2 | 4 - 0 \\
 2 | 2 - 0 \\
 1 - 0
 \end{array}$$

$$(137)_{10} = (10001001)_2$$

* $(28A)_{16} = (?)_{10}$

$$\Rightarrow 2 \times 16^2 + 8 \times 16^1 + A \times 16^0$$

$$\Rightarrow 2 \times 256 + 8 \times 16 + 10$$

$$\Rightarrow 512 + 128 + 10$$

$$\Rightarrow (650)_{10}$$

*

$$(1AB)_{16} = (?)_2$$

$$\Rightarrow (1AB)_{16} = (110101011)_2$$

8	4	2	1
1	A	B	
1	10	11	
0001	1010	1011	

$$= 1 \times 16^2 + A \times 16^1 + B \times 16^0$$

$$= 256 + 10 \times 16 + 11$$

$$= 256 + 160 + 11$$

$$= (427)_{10}$$

$$\begin{array}{r}
 2 | 427 \\
 2 | 213 - 1 \\
 2 | 106 - 1 \\
 2 | 53 - 0 \\
 2 | 26 - 1 \\
 2 | 13 - 0 \\
 2 | 6 - 1 \\
 2 | 3 - 0 \\
 \hline
 1 - 1
 \end{array}$$

$$(1A3)_{16} = (110101011)_2$$

(*) $(197)_{10} = (?)_{16}$

$$\begin{array}{r}
 16 | 197 \\
 12 - 5 \uparrow
 \end{array}$$

$$(197)_{10} = (C5)_{16} \text{ or } C5H$$

(*) $(65)_{10} = (?)_8 = (?)_2 = (?)_{16}$

$$\begin{array}{r}
 8 | 65 \\
 8 | 8 - 1 \\
 1 - 0
 \end{array}$$

$$(65)_{10} = (101)_8 = (1000001)_2 = (81)_{16}$$

③ $(371)_8 = (?)_{10}$

$$3 = 011$$

$$7 = 111$$

$$1 = 001$$

$$\text{So; } (371)_8 = (1111001)_2$$

④ $(255.35)_{10} = (?)_8$

$$\begin{array}{r} 255 \\ 8 \boxed{31} - 7 \\ 3 - 7 \\ \hline 377 \end{array}$$

$$0.35 \times 8 = 2.8 = 2 + 0.8$$

$$0.8 \times 8 = 6.4 = 6 + 0.4$$

$$0.4 \times 8 = 3.2 = 3 + 0.2$$

$$0.2 \times 8 = 1.6 = 1 + 0.6$$

$$(255.35)_{10} = (377.2631)_8$$

⑤ $(\cancel{377.} 367.237)_8 = (?)_{10}$

$$\Rightarrow 3 \times 8^2 + 6 \times 8^1 + 7 \times 8^0 + 2 \times 8^{-1} + 3 \times 8^{-2} + 7 \times 8^{-3}$$

$$\Rightarrow 128 + 48 + 7 + 0.25 + 0.0468 + 0.013$$

$$\Rightarrow (183.3098)_{10}$$

Complement

Complements are used in the digital computers in order to simplify the subtraction operation and for the logical manipulations. For each radix- r system (radix r represents base of number system). There are two types of complements.

S.N	Complement	Description
1	Radix Complement	The radix complement is referred to as the r 's complement
2	Diminished Radix Complement	The diminished radix complement is referred to as the $(r-1)$'s complement

Binary System Complements

As the binary system has base $r=2$. So, the two types of complements for the binary system are 2's complement and 1's complement

1's complement

The 1's complement of a number is found by changing all 1's to 0's and all 0's to 1's. This is called taking complement or 1's complement. Example for 1's complement.

Given number: 1 0 1 0 1
 1's complement: 0 1 0 1 0

2's complement

The 2's complement of binary number is obtained by adding 1 to the Least Significant Bit (LSB) of 1's complement of the number.

$$2's \text{ complement} = 1's \text{ complement} + 1$$

Example of 2's complement

Given number: 1 0 1 0 1

1's complement: 0 1 0 1 0
 + 1

2's complement 01011

$$\begin{array}{r} \cancel{\text{Add}} \quad 11011 \\ -01010 \\ \hline \end{array}$$

$$\begin{array}{r} 1's \text{ complement of } 01010 = 10101 \\ + 1 \\ \hline 10110 \end{array}$$

$$\begin{array}{r} 11011 \\ 10110 \\ \hline \end{array}$$

end-carry 1] 10001 Ans = 10001

10's complement

By the given formula we find the 10's complements:

$$10's \text{ complement} = r^n - N ; r = 10 \\ = 10^n - N$$

$$10's \text{ complement of } 23 = 10^2 - 23 \\ = 100 - 23 \\ = 77$$

9's complement

By the given formula we find the 9's complement:

$$9's \text{ complement} = (r^n - 1) - N ; r = 10 \\ = (10^n - 1) - N$$

$$9's \text{ complement of } 23 = (10^2 - 1) - 23 \\ = 99 - 23 \\ = 76$$

Subtraction using 10's complement

$$\begin{array}{r} 100 \\ - 023 \\ \hline \end{array}$$

$$\begin{aligned} \text{10's complement of } 023 &= 10^3 - 023 \\ &= 1000 - 023 \\ &= 977 \end{aligned}$$

$$\begin{array}{r} 100 \\ + 977 \\ \hline \end{array}$$

end carry
= 077
= 77

Subtraction using 9's complements

$$\begin{array}{r} 100 \\ - 023 \\ \hline \end{array}$$

$$\begin{aligned} \text{9's complement of } 023 &= (10^3 - 1) - 023 \\ &= (1000 - 1) - 023 \\ &= 999 - 023 \\ &= 976 \end{aligned}$$

$$\begin{array}{r} 100 \\ + 976 \\ \hline \end{array}$$

→ ← + 1
= 77

10's complement

$$\begin{array}{r} 23 \\ - 100 \\ \hline \end{array}$$

$$\begin{aligned} \text{10's complement of } 100 &= 10^3 - 100 \\ &= 1000 - 100 \\ &= 900 \end{aligned}$$

$$\begin{array}{r} 023 \\ - 900 \\ \hline \end{array}$$

9 23 If no comes carry then

$$\begin{aligned} &= -(10\text{'s complement of } 923) \\ &= -(10^3 - 923) \\ &= -(1000 - 923) \\ &= -77 \end{aligned}$$

Logic Gates

Binary logic deals with variables that assume discrete values and with operators that assume logical meaning. While each logical element or condition must always have a logic value of either "0" or "1".

A) Basic Gates:

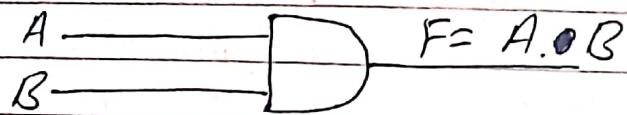
The basic building blocks of a computer are called logical gates or just gates. Gates are basic circuit that have at least one (and usually more) input and exactly one output. Input and output values are the logical values true or false.

We usually consider three basic unit of gates:

- i) AND Gate
- ii) OR Gate
- iii) NOT Gate (Inverter)

1) AND GATE

The AND Gate produces a high output when all inputs are high, otherwise, the output is low.

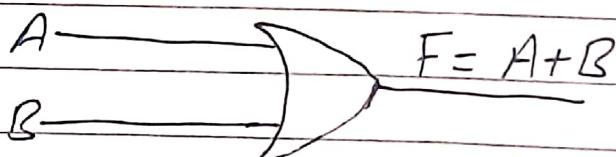


Truth Table

Input	Output	
A	B	$F = AB$
0	0	0
0	1	0
1	0	0
1	1	1

2) OR Gate:

The OR Gates performs the Boolean NOT addition operation. OR Gate produces a high ~~input~~ output if any input is high, if all input is low, output is also low.



Truth Table:

Input		Output
A	B	$F = A + B$
0	0	0
0	1	1
1	0	1
1	1	1

3) NOT Gate:-

The NOT Gate performs the Boolean NOT operation. When the input is low, the output is high, when the input is high the output is low.

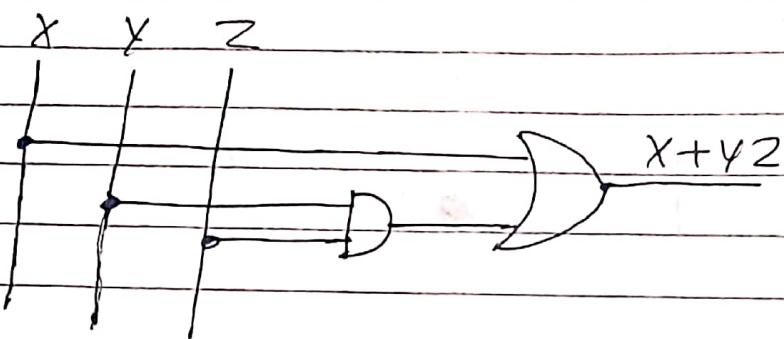
$$A \rightarrow F = A'$$

Truth Table

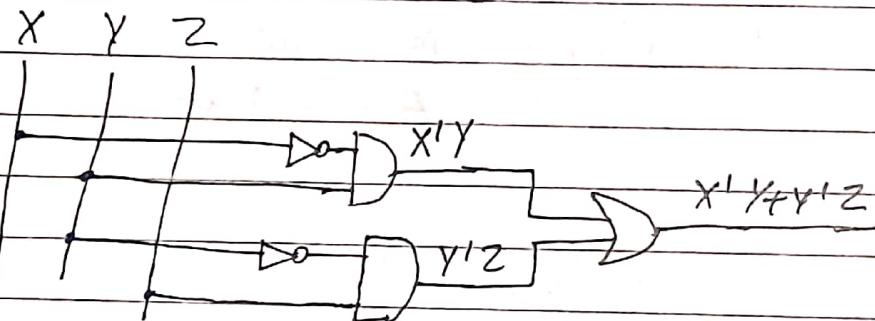
Input		Output
A		$F = A'$
0		1
1		0

Some Logic Circuit

i) $X + YZ$



ii) $X'Y + Y'Z$



④

Universal Gates:

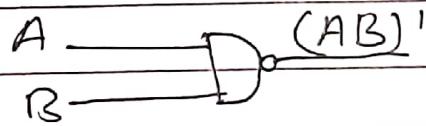
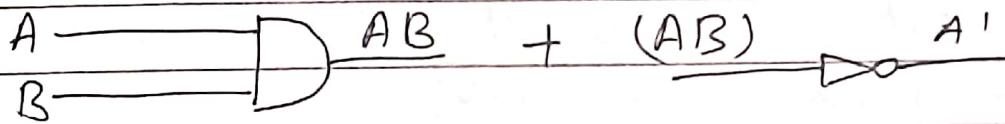
A universal gate is a gate which can implement only Boolean (AND, OR, NOT) function without need to use any other gate type.

The universal gates are:-

- I) NAND Gate
- II) NOR Gate

I) NAND Gate:

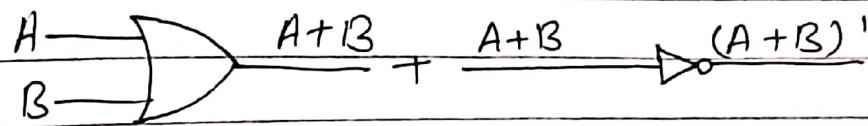
It is the combination of AND gate followed by a NOT gate. It produces output high if any of the input is low.



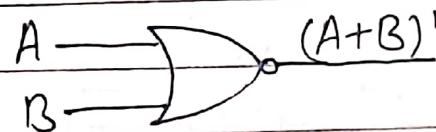
Input		Output
A	B	$F = (AB)'$
0	0	1
0	1	1
1	0	1
1	1	0

2) NOR Gate:-

It is the combination of OR gate followed by a NOT gate. It produces output low if any one of the input is high.



Symbol:



Truth table

Input		Output
A	B	$F = (A+B)'$
0	0	1
0	1	0
1	0	0
1	1	0

(*) Boolean Algebra:

It is also known as switching algebra. It was introduced by George Boole in 1854.

(#) Postulates of Boolean Algebra:

1) Closure:

Boolean Algebra is closed under the AND, OR and NOT gate operator.

2) Commutativity:-

The *(AND) and +(OR) operators are commutative ie $A+B = B+A$
 $A*B = B*A$

3) Distribution:-

AND(*) and OR(+) operators are distribution to one other.

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

or,

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

4) Identity:-

$$A + 0 = A$$

$$A * 1 = A$$

5) Inverse:-

For every value of x there exist a value x^{-1} such that $x \cdot x^{-1} = 0$
 $x + x^{-1} = 1$

Theorems of Boolean Algebra:

1)

$$(a) x + x = x \quad (b) x \cdot x = x$$

② Existence:

$$(a) x + 1 = 1$$

$$(b) x \cdot 0 = 0$$

③ Involution:

$$(x')' = x$$

④ Associative:

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

$$(b) x(yz) = (xy)z$$

⑤ Demorgan:

$$\textcircled{a} \quad (x+y)' = x'y'$$

$$\textcircled{b} \quad (xy)' = x'+y'$$

⑥ Absorption:

$$\textcircled{a} \quad x+xy = x(1+y) \\ = x$$

$$\textcircled{b} \quad x(x+y) = x \cdot x + x \cdot y \\ = x+xy \\ = x(1+y) \\ = x$$

State and prove Demorgan's law for two variables?

⇒ Demorgan's theorem is mainly used to solve the various Boolean algebra expressions. The Demorgan's theorem defines the uniformity between the gate with same inverted input and output. It is used for implementing the basic gate operation like NAND gate and NOR Gate. The demorgan's theorem mostly used in digital programming and for making digital circuit diagrams.

$$\textcircled{1} \quad (x+y)' = x'y'$$

$$\textcircled{2} \quad (x'y)' = x'+y'$$

For ① $(x+y)' = x'y'$

x	y	$x+y$	$(x+y)'$	x'	y'	$x'y'$
0	0	0	1	1	1	1
0	1	1	0	1	0	0
1	0	1	0	0	1	0
1	1	1	0	0	0	0

For ② $(xy)' = x'y' + xy'$

x	y	xy	$(xy)'$	x'	y'	$x'y' + xy'$
0	0	0	1	1	1	1
0	1	0	1	1	0	1
1	0	0	1	0	1	1
1	1	1	0	0	0	0

Simplyfy the following:-

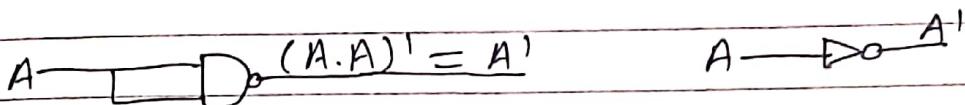
$$\begin{aligned} ① \quad & x + x'y \\ &= (x \cancel{x}) + (x \cancel{x})y \\ &= (x+y) \end{aligned}$$

$$\begin{aligned} ② \quad & \cancel{xy} + \cancel{xz} + \cancel{yz} \quad x(x'y + yz) \\ &= xx' + xy \\ &= xy \end{aligned}$$

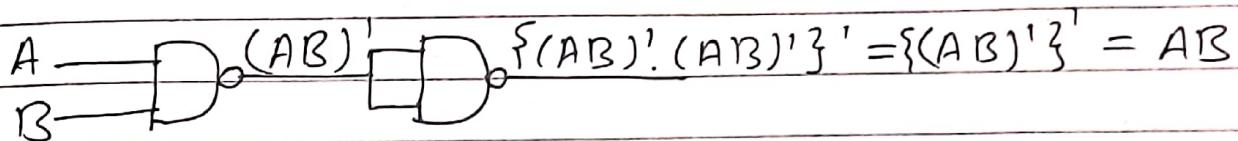
- ⑨ Show that NAND Gate & NOR Gate are universal gate.

NAND Gate

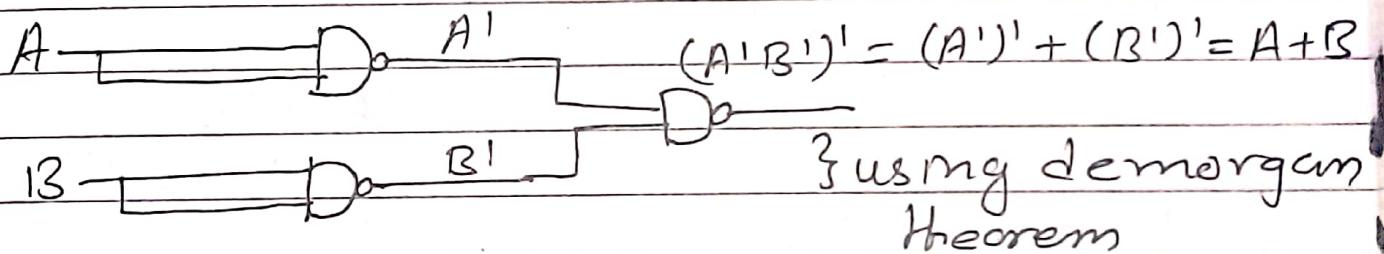
NAND Gate as NOT Gate



NAND Gate as AND Gate

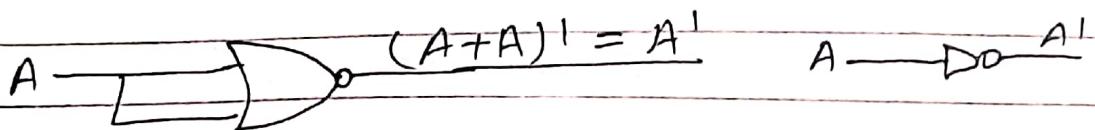


NAND Gate as OR Gate



NOR Gate

a) NOR Gate as NOT Gate



b) NOR Gate as OR Gate

A circuit diagram showing two NOR gates. The first NOR gate has inputs A and B, and its output is labeled $(A+B)'$. This output is connected to one input of a second NOR gate, whose other input is also labeled $(A+B)'$. The output of the second NOR gate is labeled $\{(A+B)'+(A+B)'\}'$, which simplifies to $\{(A+B)'\}'$, and finally to $A+B$.

c) NOR Gate as AND Gate

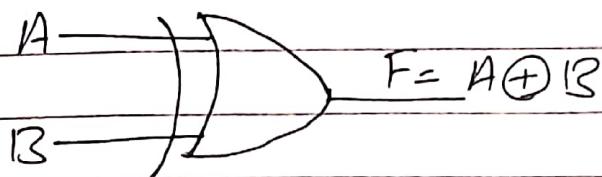
A circuit diagram showing two NOR gates. The first NOR gate has inputs A and B, and its output is labeled A' . The second NOR gate has inputs $(A')'$ and B' , and its output is labeled $(A')'+B'$, which simplifies to $(A')'.(B')'$, and finally to $A.B$.

? using demorgan
theorem

Hence, proved.

Ex-Or Exclusive Gate

① Exclusive OR (XOR)



$$A \oplus B = A'B + AB'$$

A	B	$F = A \oplus B = A'B + AB'$
0	0	$1.0 + 0.1 = 0$
0	1	$0.1 + 0.0 = 1$
1	0	$0.0 + 1.1 = 1$
1	1	$0.1 + 1.0 = 0$

It is a logic gate which produces output high if any two input are different or, it produces output low if the inputs are same.

② Exclusive NOR (X-NOR)



Truth table:

A	B	$A \oplus B = AB + A'B'$
0	0	$0 \cdot 0 + 1 \cdot 1 = 0 \cdot 1$
0	1	$0 \cdot 1 + 1 \cdot 0 = 0$
1	0	$1 \cdot 0 + 0 \cdot 1 = 0$
1	1	$1 \cdot 1 + 0 \cdot 0 = 1$

It is a logic gate which produces output high if any two input are same or it produces output low if the inputs are different.

Dual (Duality Principle)

It states that "Every algebraic expression deducible from the postulates of Boolean Algebra remains valid. If the operator and identity elements are interchanged."

Find the complement

$$F_1 = x'y'z + x'y'z$$

$$F_1' = (x'y'z + x'y'z)'$$

The dual of $F_1 = (x'+y+z)*(x'+y'+z)$

Complement $F_1' = (x+y+z)*(x+y+z)'$

$$(AB)' = A'B'$$

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

$$(AB)' = A'+B'$$

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

$$\begin{aligned}
 F &= XY'Z' + XYZ \\
 &= (X+Y'+Z') * (X+Y+Z) \\
 &= (X'+Y+Z) * (X'+Y'+Z')
 \end{aligned}$$

Max terms

- => A max terms is a sum (ORing of each term) of literals, in which each input variable appears exactly ones.
- => A function with 'n' variable has 2^n max terms.
- => Each max terms is false for exactly one combination of inputs

Min terms

- => A min terms is a special product (ANDing of terms) of literals, in which each input variable appears exactly once's.
- => A function with 'n' variable has 2^n min terms
- => Each min terms is true for exactly one combination of inputs

X	Y	Z	Min term	Max term
0	0	0	$m_0 X'.Y'.Z'$	$M_0 X+Y+Z$
0	0	1	$m_1 X'.Y'.Z$	$M_1 X+Y+Z'$
0	1	0	$m_2 X'.Y.Z'$	$M_2 X+Y'+Z$
0	1	1	$m_3 X'.Y.Z$	$M_3 X+Y'+Z'$
1	0	0	$m_4 X.Y'.Z'$	$M_4 X'+Y+Z$
1	0	1	$m_5 X.Y'.Z$	$M_5 X'+Y+Z'$
1	1	0	$m_6 X.Y.Z'$	$M_6 X'+Y'+Z$
1	1	1	$m_7 X.Y.Z$	$M_7 X'+Y'+Z'$

SOP = Sum of Product

POS = Product of Sum

Sum of Products

$$f(x, y, z) = \{x'y'z + xy'z + x'y'z'\} \quad \text{canonical form}$$

$$= m_1 + m_7 + m_0$$

$$f(x, y, z) = \Sigma(0, 1, 7) \quad \text{standard form}$$

$$F(x, y, z) = (x+y+z)(x'+y'+z')(x+y'+z')$$

$$= M_0 \cdot M_1 \cdot M_2$$

$$= \Pi(0, 2, 7)$$

What are ^{on} canonical form?

⇒ Boolean functions expressed as a sum of min terms or product of max terms are said to be in canonical form.

Standard form:-

A standard form is modified form of canonical forms. There are two types of standard form they are

- i) SOP
- ii) POS

Express the following in terms of sum of minterm.

A, B, C

$$\textcircled{1} \quad A$$

$$\Rightarrow A(B+B')$$

$$\Rightarrow AB + AB'$$

$$\Rightarrow AB(C+C') + AB'(C+C')$$

$$\Rightarrow ABC + ABC' + AB'C + AB'C'$$

$$\textcircled{2} \quad A + B'C$$

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

$$\Rightarrow AB + AB' + AB'C + A'B'C$$

$$\Rightarrow AB(C+C') + AB'(C+C') + AB'C + A'B'C$$

$$\Rightarrow ABC + ABC' + AB'C + ABC' + A'B'C$$

$$\Rightarrow ABC + ABC' + AB'C + AB'C' + A'B'C$$

$$\Rightarrow M_7 + M_6 + M_5 + M_4 + M_1$$

$$\Rightarrow \Sigma(1, 4, 5, 6, 7)$$

POS

$$\textcircled{*} \quad x' + y$$

$$\Rightarrow x' + y + 0$$

$$\Rightarrow x' + y + z z'$$

$$\Rightarrow (x' + y + z)(x' + y + z')$$

$$\Rightarrow M_4 \text{ & } M_5$$

$$\Rightarrow \pi(4, 5)$$

K-Map

Karnaugh Map (Veitch Diagram)

In digital circuit we need to find boolean expressions with minimum variables. For this karnaugh map can be used. Karnaugh map is the pictorial representation of simplification of Boolean expression.

K-map can take two forms in SOP & POS according to the need. We feel k-map has square grid representing min-term or max term. K-map can be a two, three or four variable.

Rules:-

- i) Group should be made that must contain one, two, four, eight or in terms in 2^n .
- ii) Group may be horizontal or vertical but not diagonal.
- iii) Each should be as large as possible.
- iv) Groups may overlap.
- v) The group should be as large as possible & no of group should be small.

K-map for 2 variables

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

A \ B	0	1
0	1 0	1 1
1	1 2	1 3

$$= A' \cdot B + A \cdot B'$$

$$F(A, B) = \Sigma(1, 2)$$

A \ B	0	1
0	0	1 1
1	1 2	3

$$= AB' + A'B$$

$$= A \oplus B$$

K-map for 3 variables

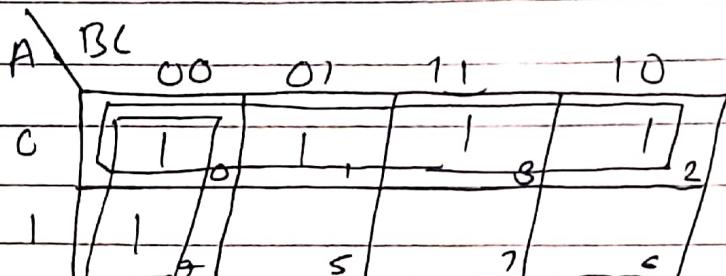
$$F(A, B, C) = \Sigma(0, 1_2, 2, 3)$$

A \ BC	00	01	10	11
0	1 0	1 1	1 3	1 2
1	4	5	7	6

$$= A' \cdot B \cdot C$$

$$= A'$$

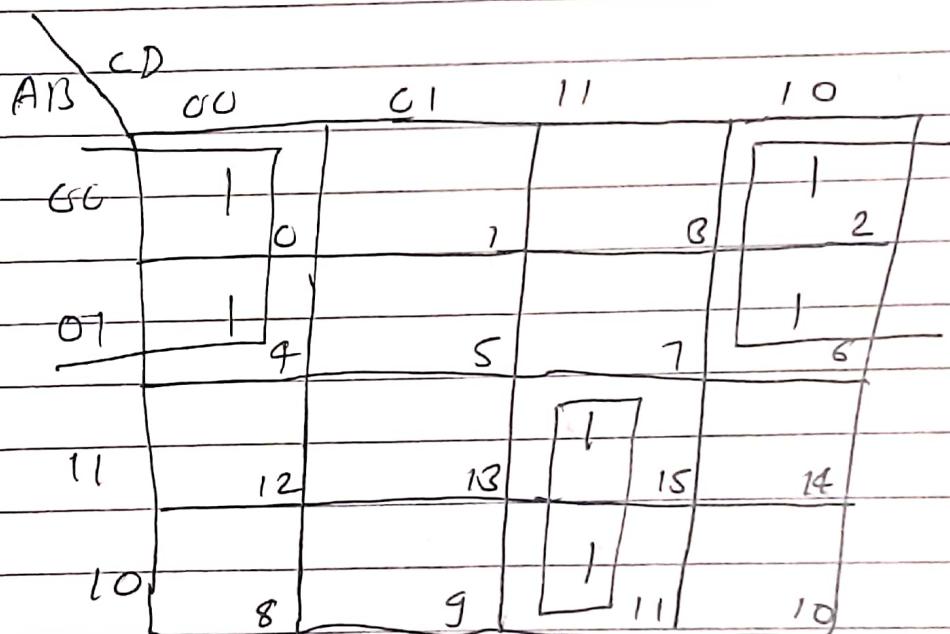
$$F(A, B, C) = \Sigma(0, 1, 2, 3, 4)$$



$$= B'C' + A'$$

K-map for 4 variable

$$F(A, B, C, D) = \Sigma(0, 2, 4, 6, 11, 15)$$



$$= A'D' + ACD$$

POS

$$F(A, B, C) = \pi(0, 2, 4, 6)$$

A	\	BC	00	01	11	10
0		0	0	1	3	0 ₂
1		0	4	5	7	0 ₆

$$= C$$

$$F(A, B, C) = \pi(0, 1, 7, 3)$$

A	\	BC	00	01	11	10
0		0	0	1	3	2
1		4	5	0 ₇	6	

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

Don't Care Condition

$$F(A, B, C) = \Sigma(0, 2, 4, 6)$$

$$d(F)(A, B, C) = \Sigma(7, 1, 3)$$

A	\	BC	00	01	11	10
0		0	1	x ₁	x ₃	1 ₂
1		1	7	5	x ₇	1 ₆

$$= 1 \cdot 1 \cdot C'$$

$$= C'$$

$$F(A, B, C, D) = \Sigma (1, 3, 7, 11, 12, 14)$$

$$J(A, B, C, D) = \Sigma (5, 9, 13, 15)$$

AB	CD	00	01	11	10
00		1	1	3	2
01	X		1		6
11	1	X	X	1	
10	12	13	15	14	
		8	9	11	10

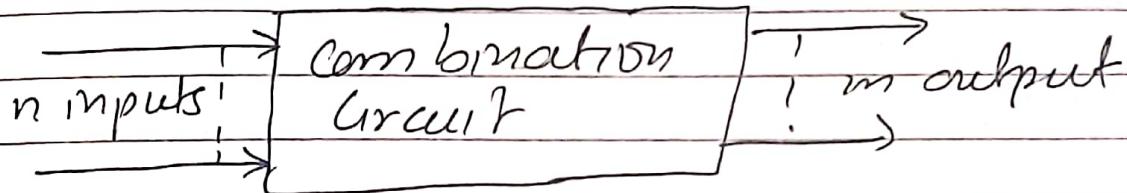
$$= D + AB$$

$$= AB + D$$

Combinational logic Circuit

Combinational logic is a type of digital logic which is implemented by Boolean circuits, where the output is a pure function of the present input only.

A combinational circuit consists of 'n' input variables, logic gates, 'm' output variables. For 'n' input variables there are 2^n possible combination of binary output values. For each possible input combinations there is one and only one output combination.



Types of combinational circuit

① Adder:

The combinational circuit which performs addition operation is called adder.

ⓐ Half-Adder:

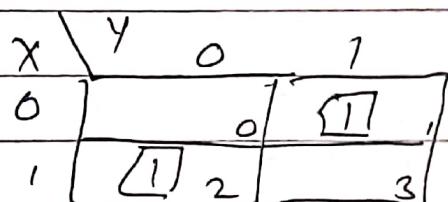
The combinational circuit which performs addition of 2 bit is called half adder. The circuit needs two inputs and 2 output. The input variables can be x & y .

as augend and addend respectively. The output variables produce sum and carry.

Truth table for half-Adder

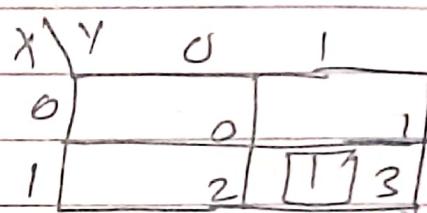
X	Y	Carry Sum(S)	Carry(C)
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

K-map for Sum



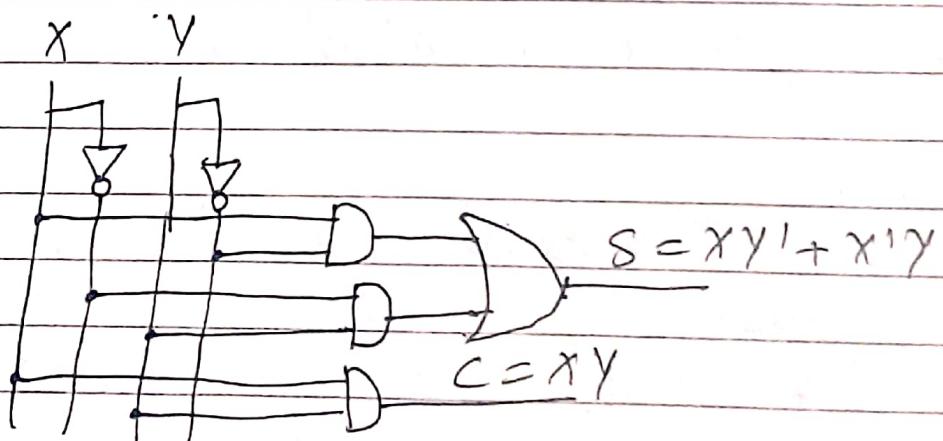
$$\begin{aligned} \text{Sum} &= x'y + xy' \\ &= x \oplus y \end{aligned}$$

K-map for Carry

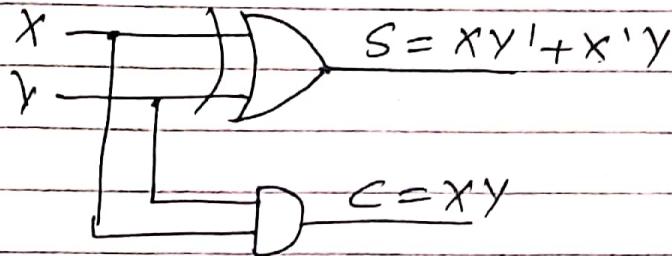


$$C = xy$$

Circuit diagram



Standard logic Circuit



⑥ Full Adder :-

A full adder is a combinational circuit that forms the arithmetic sum of 3-bits. It consists of 3 inputs bit & 2 outputs. Following is the true table for full adder.

X	Y	Z	S	C
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

12k-map for Sum

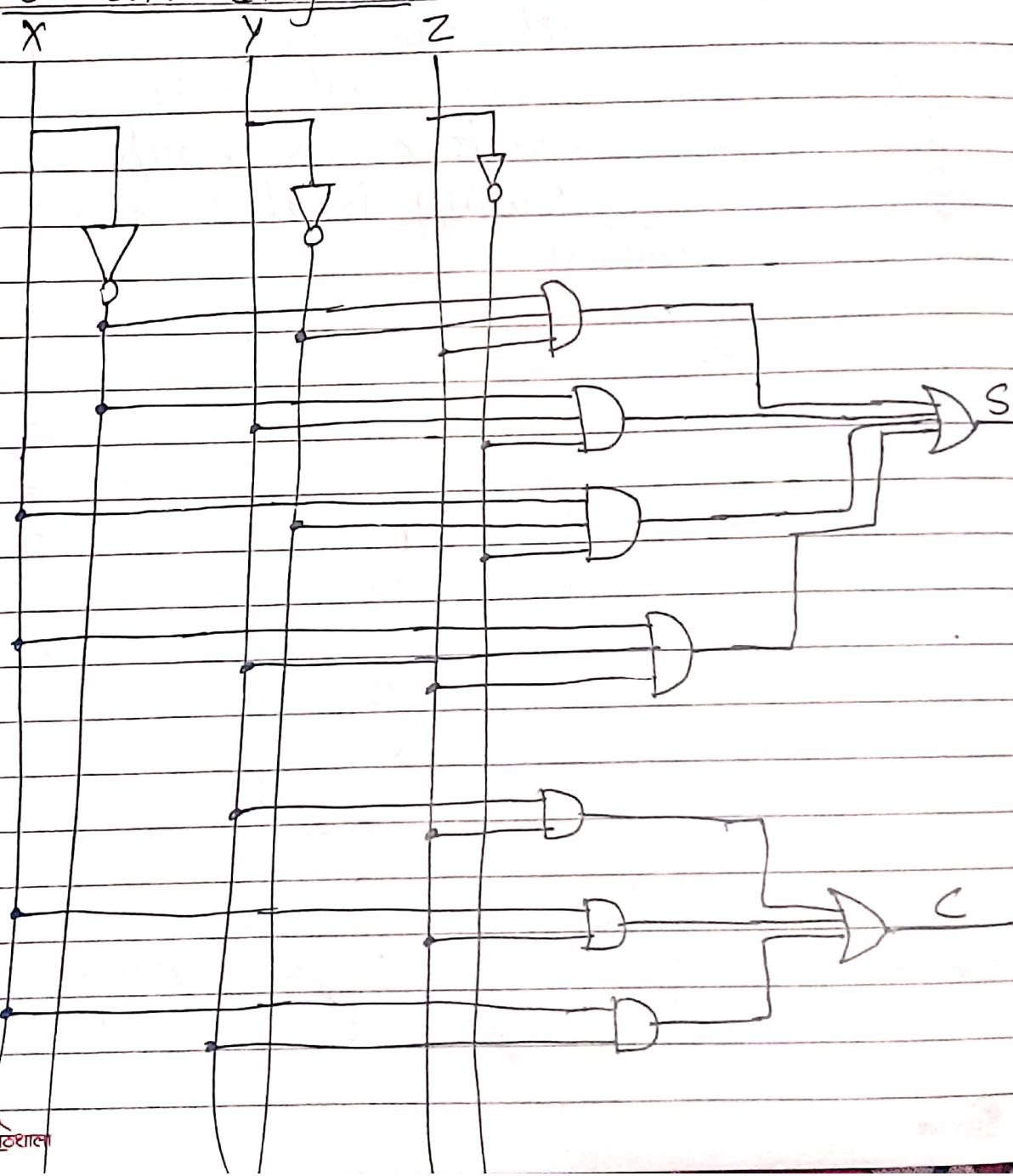
X\Y	00	01	11	10	Sum
0	0	1	1	0	$x'y'z + x'y'z + xy'z + xy'z$
1	1	0	0	1	$x'y'z + xy'z + xy'z + xy'z$

K-map for carry

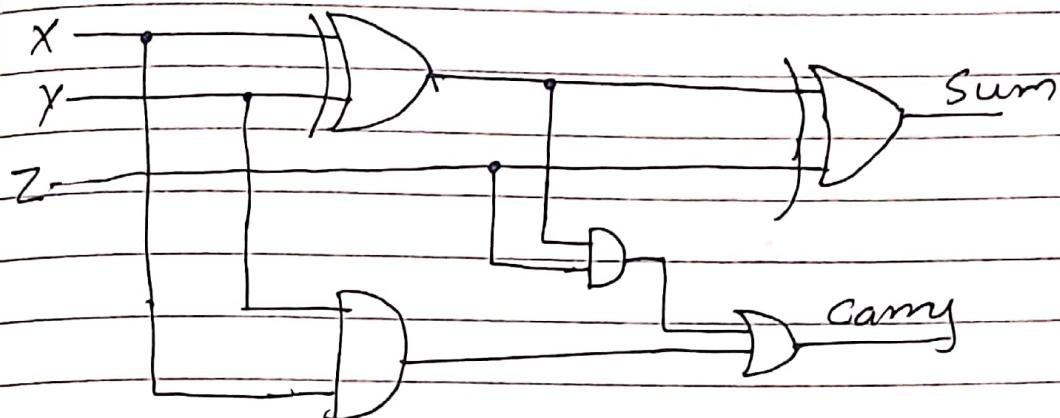
x \ y\z	00	01	11	10
0	0	0	1	1
1	1	1	1	0

$$= y_2 + xz + xy$$

Circuit diagram



Standard Circuit



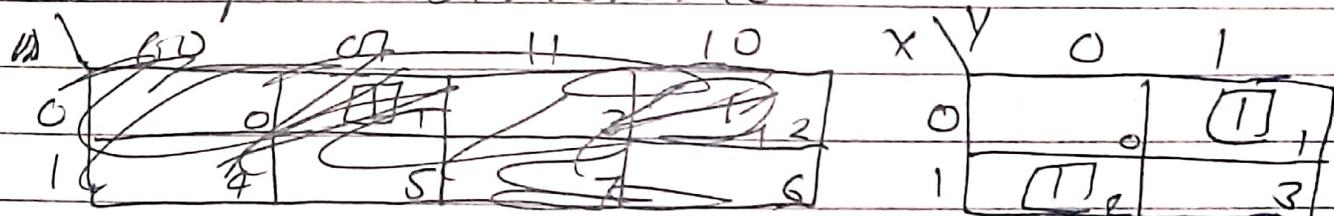
④ Half Subtractor

It is a combinational logic circuit that subtracts two inputs and has two output A & B.

Truth table

X	Y	Difference (D)	Borrow (B)
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0

K-map for Difference



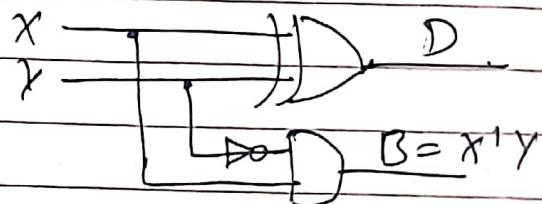
$$D = X'Y + XY'$$

K-map for borrow

x	y	0	1
0		0	1
1		2	3

$$B = X'Y$$

standard diagram



④ Full Subtractor

A full subtractor is a combinational logic circuit which subtracts three numbers and has outputs as difference and borrow.

X	Y	Z	Difference (D)	Borrow (B)
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

K-map for difference

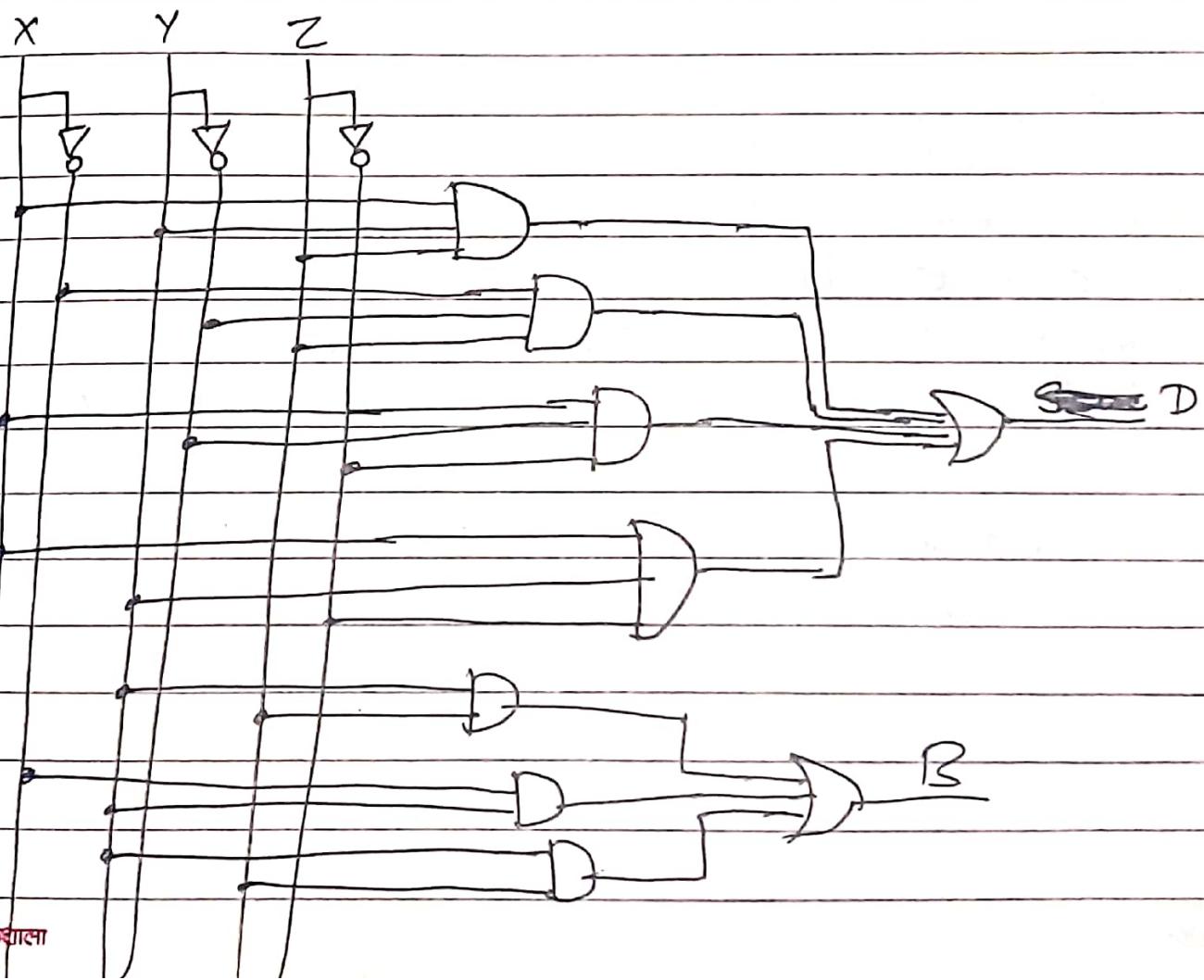
x \ yz	00	01	11	10
0	0	1	3	12
1	4	5	7	6

$$D = X'Y'Z + XY'Z' + XYZ + X'YZ'$$

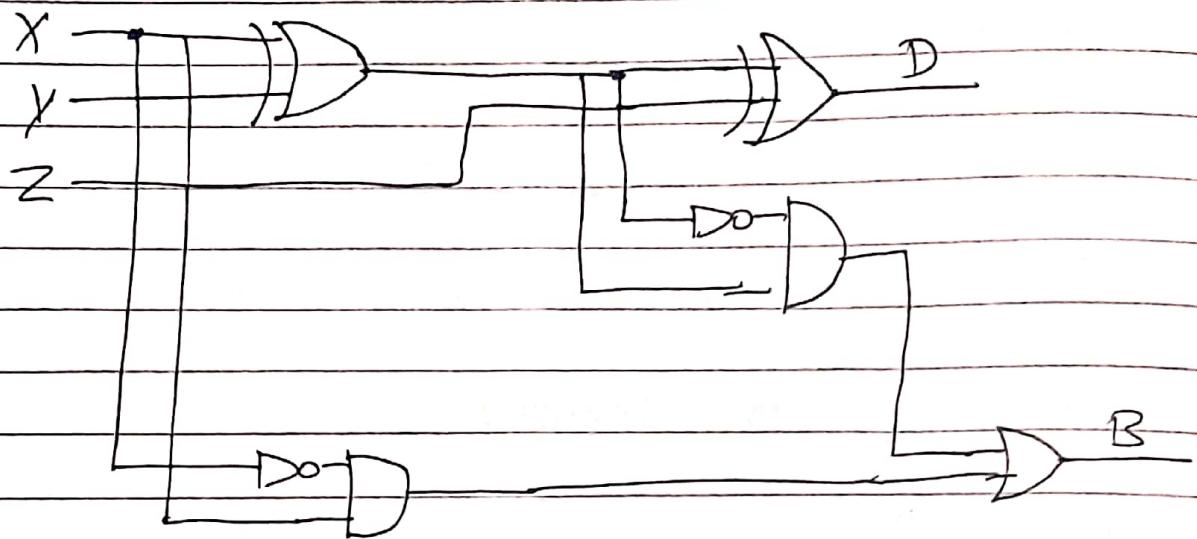
K-map for Borrow

x \ yz	00	01	11	10
0		1	1	1
1			1	

$$B = X'Z + X'Y + YZ$$



Standard Diagram of half Subtractor



Encoders:

Encoder is a combinational circuit that perform the reverse operation of decoder.

It has maximum of 2^n input lines and n output line. It will produce a binary scale equivalent to the input which is active high.

Eg: 4 to 2 lines Encoders:

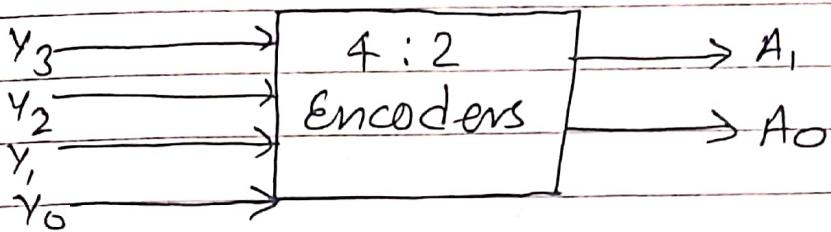
A 4 to 2 encoders has 4 inputs

y_3, y_2, y_1 , and y_0

and 2 output

A_1 , and A_0

The block diagram of 4 to 2 encoder is shown below:-



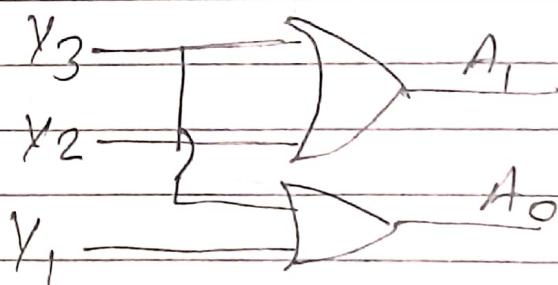
Truth Table

Input				Output	
Y_3	Y_2	Y_1	Y_0	A_1	A_0
0	0	0	1	0	0
0	0	1	0	0	1
0	1	0	0	1	0
1	0	0	0	1	1

From the truth table, we can write boolean for each output as:

$$A_1 = Y_3 + Y_2$$

$$A_0 = Y_2 + Y_3$$



Octal to Binary Encoder

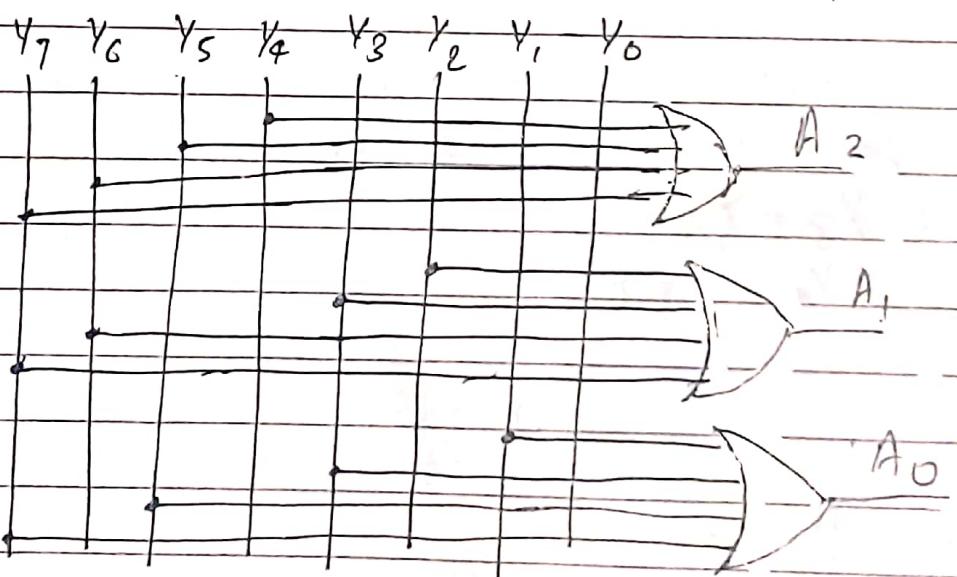
(8 to 3)

Y_7	Y_6	Y_5	Y_4	Y_3	Y_2	Y_1	Y_0	A_2	A_1	A_0
0	0	0	0	0	0	0	1	0	0	0
0	0	0	0	0	0	1	0	0	0	0
0	0	0	0	0	1	0	0	0	0	1
0	0	0	0	1	0	0	0	0	1	0
0	0	0	1	0	0	0	0	1	0	0
0	0	1	0	0	0	0	0	1	0	0
0	1	0	0	0	0	0	0	1	0	1
1	0	0	0	0	0	0	0	1	1	0

$$A_2 = Y_4 + Y_5 + Y_6 + Y_7$$

$$A_1 = Y_2 + Y_3 + Y_6 + Y_7$$

$$A_0 = Y_1 + Y_3 + Y_5 + Y_7$$

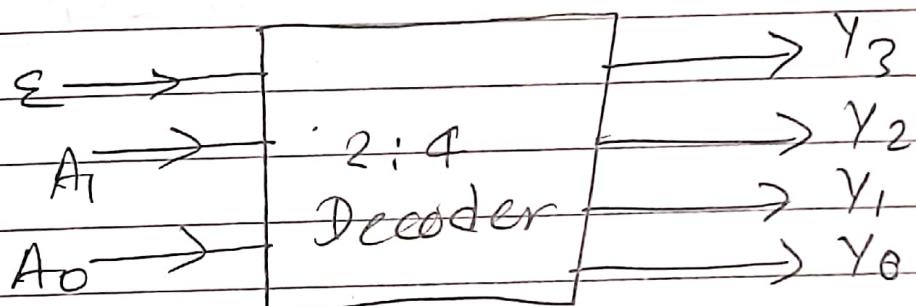


Decoders

Decoder is a combinational circuit that has 'n' input line and maximum of 2^n output lines. One of these outputs will be active high based on the combination of inputs present, when the decoder is enabled. The outputs of the decoder are nothing but the minterms of n inputs for 1 lines (~~one~~ variable) when it is enabled.

2 to 4 Decoder

A 2 to 4 decoder has 2 inputs A_1 & A_0 and 4 outputs Y_3, Y_2, Y_1 & Y_0 .



E	A ₁	A ₀	Y ₃	Y ₂	Y ₁	Y ₀
0	x	x	0	0	0	0
1	0	0	0	0	0	1
1	0	1	0	0	1	0
1	1	0	0	1	0	0
1	1	1	1	0	0	0

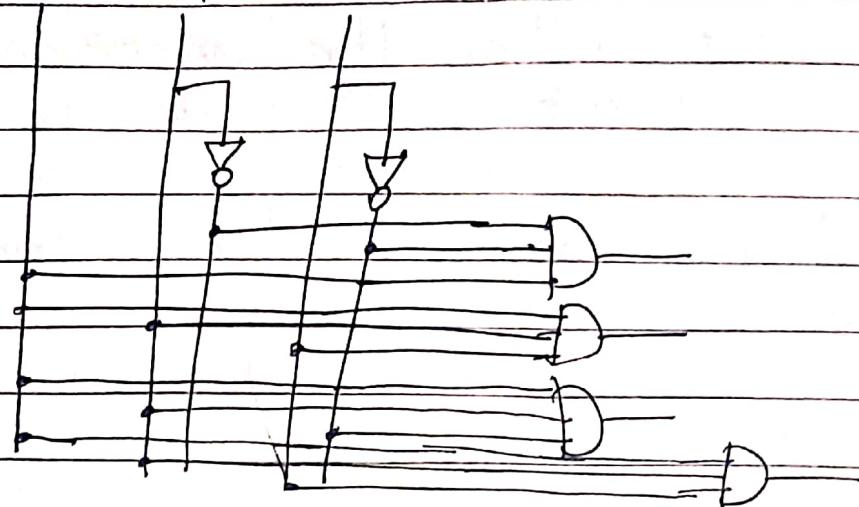
$$Y_0 = EA_1 A_0'$$

$$Y_1 = EA_1' A_0$$

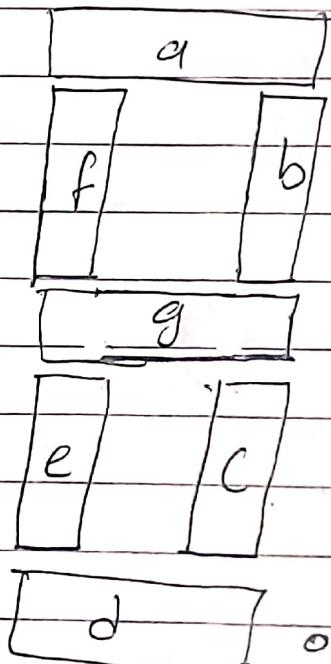
$$Y_2 = EA_1 A_0'$$

$$Y_3 = EA_1 A_0$$

\sum $E \quad A_1 \quad A_0$



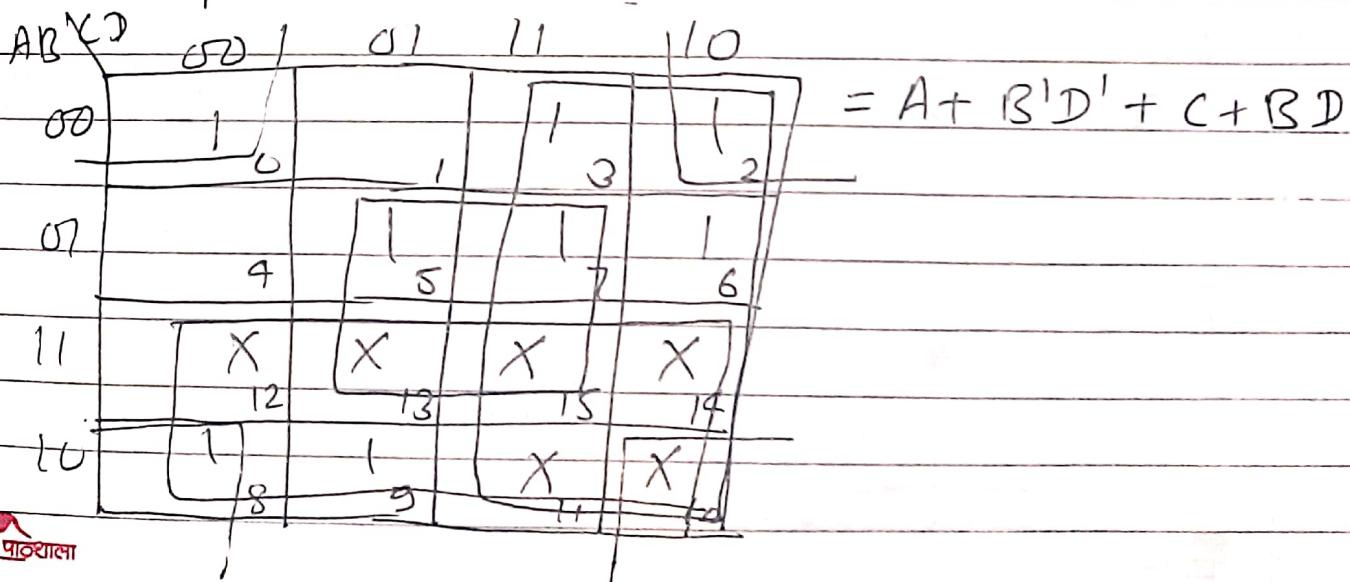
BCD to Seven Segment Decoder



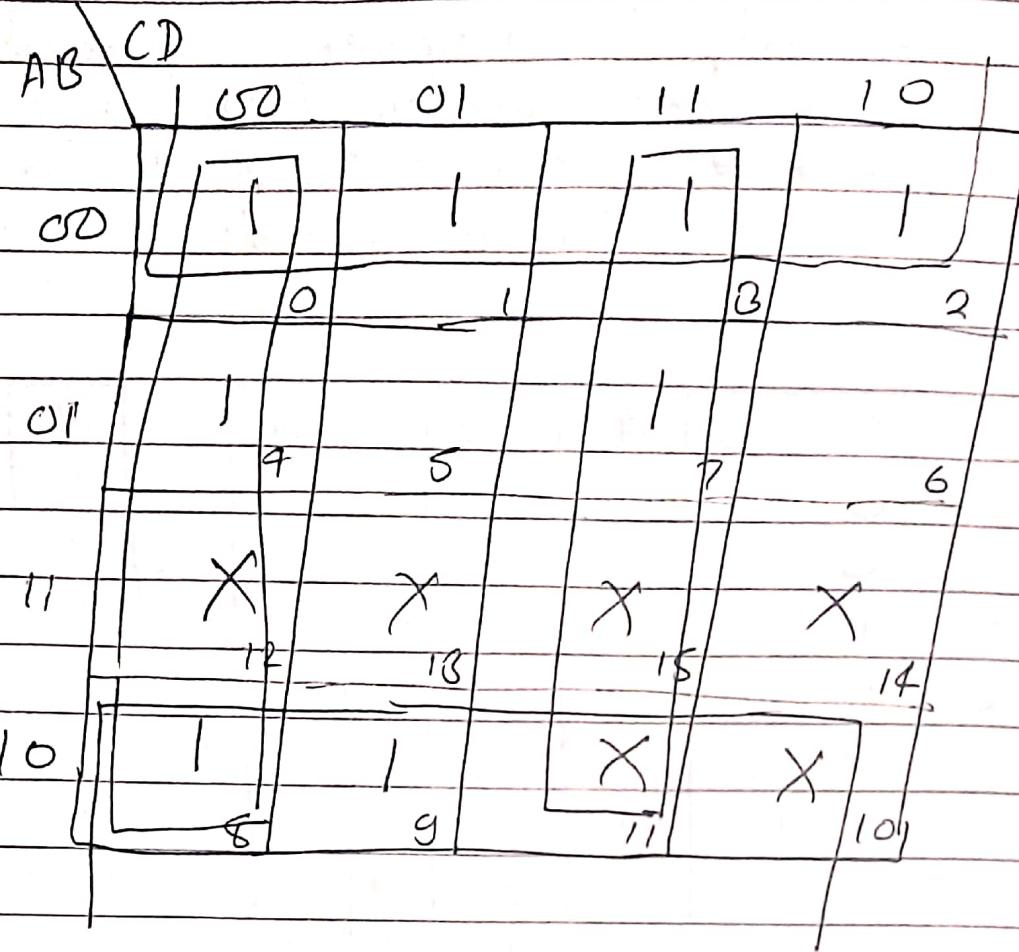
A seven segment decoder is the combinational circuit that takes BCD inputs and gives the output in seven segment display is the combination of 7 LED (Light ~~emit~~ Emitting Diode). The following is the table for the segment decoder.

A	B	C	D	a	b	c	d	e	f	g
0	0	0	0	1	1	1	1	1	1	0
0	0	0	1	0	1	1	0	0	0	0
0	0	1	0	1	1	0	1	1	0	1
0	0	1	1	1	1	1	1	0	0	1
0	1	0	0	0	1	1	0	0	1	1
0	1	0	1	1	0	1	1	0	1	1
0	1	1	0	1	0	1	1	1	1	1
0	1	1	1	1	1	1	0	0	0	0
1	0	0	0	1	1	1	1	1	1	1
1	0	0	1	1	1	1	1	0	1	1

K-map for a

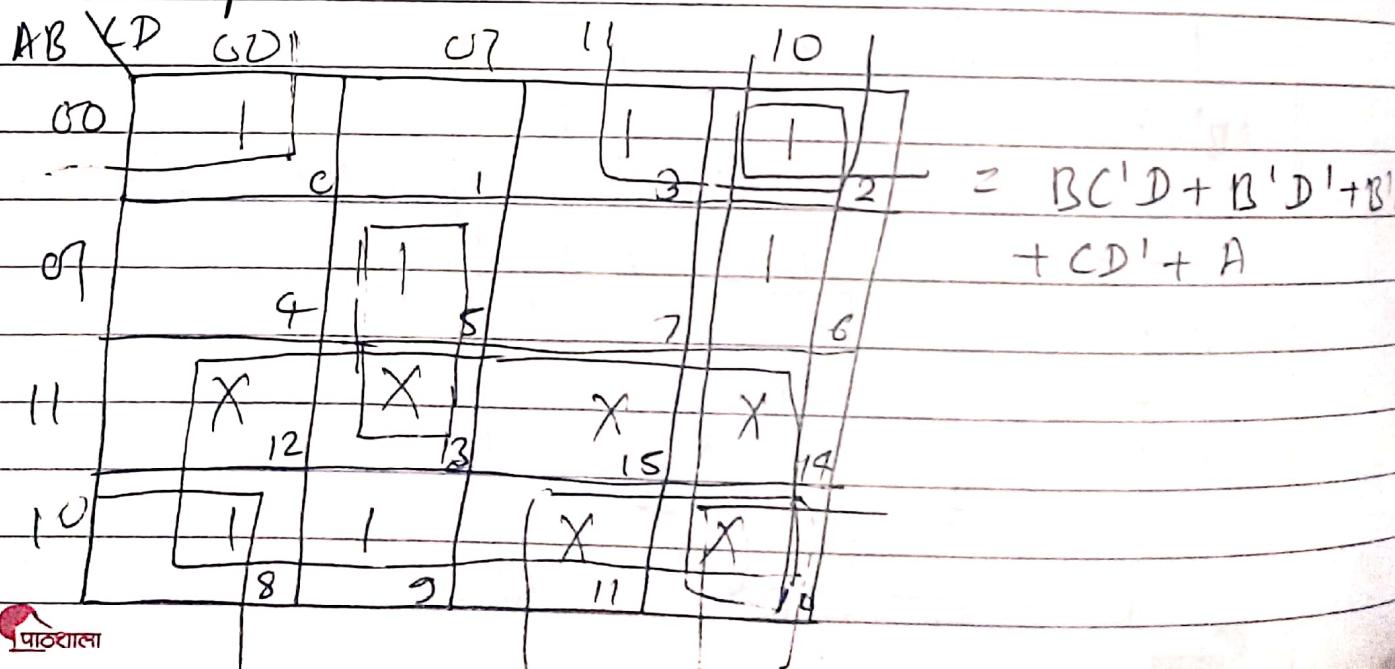


K-map for C

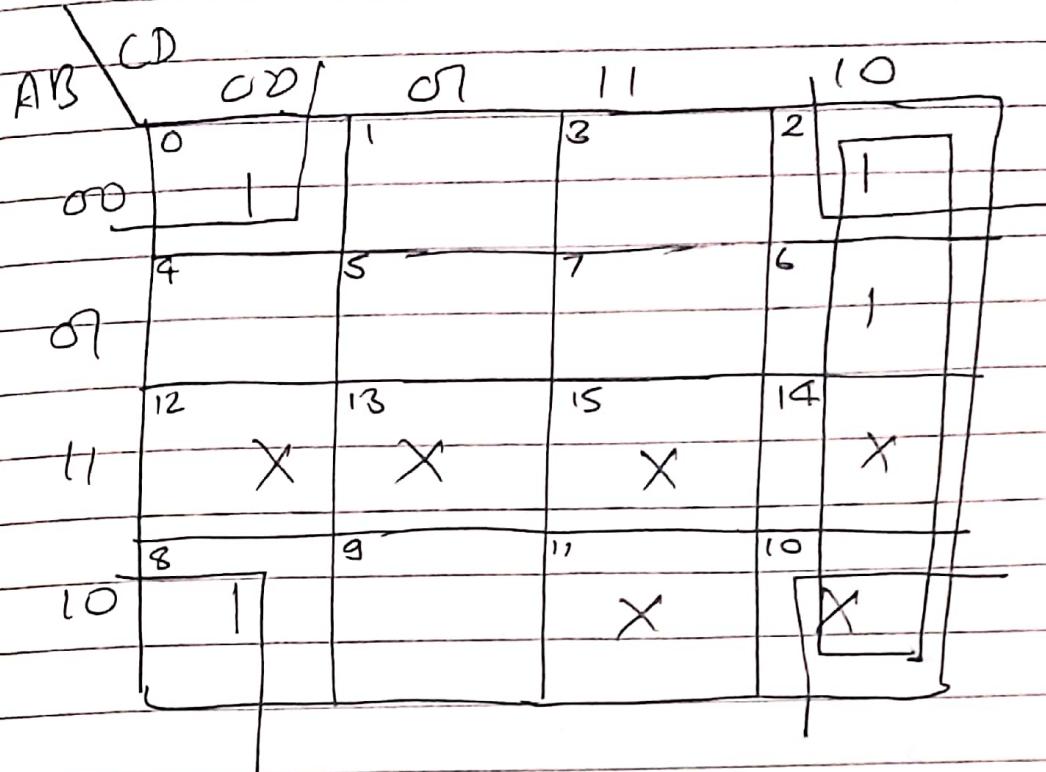


$$C @ = C'D' + CD + B'$$

K-map for D

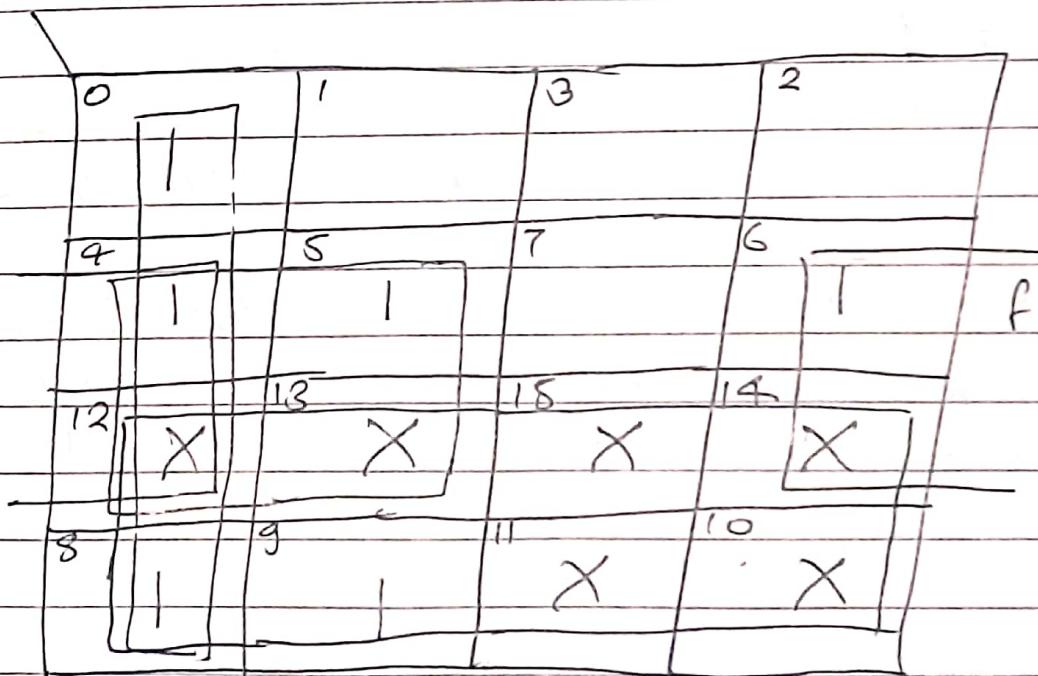


K-map for e



$$e = B'D' + CD'$$

K-map for f



K-map for g

AB\CD	00	01	01	11	10
00	0	1	0	1	1
01	4	5	7	6	11
01	12	13	15	14	
11	X	X	X	X	X
10	8	9	11	10	

$$g = B'C + BC' + BD' + A$$

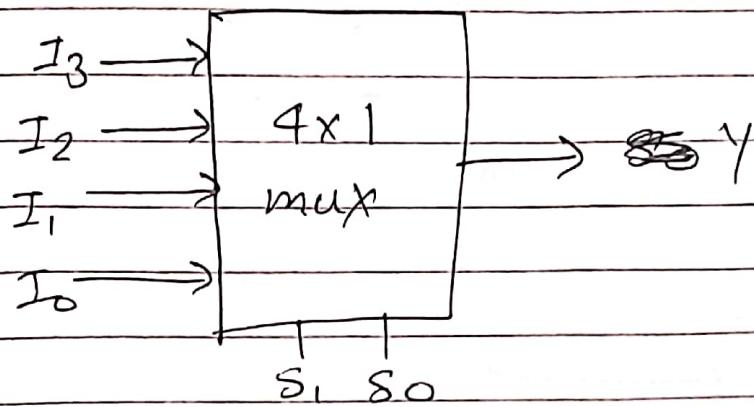
Multiplexer

A multiplexer is a combinational circuit that has maximum of 2^n data inputs, n selection line and a single output line. One of these data inputs will be connected to the output based on the values of selection ~~line~~ line.

Since there are ' n ' selection lines, there will be 2^n possible combinations of zeros & ones. So each combination will select only one data input. Multiplexer is also called mux.

① 4x1 multiplexer:

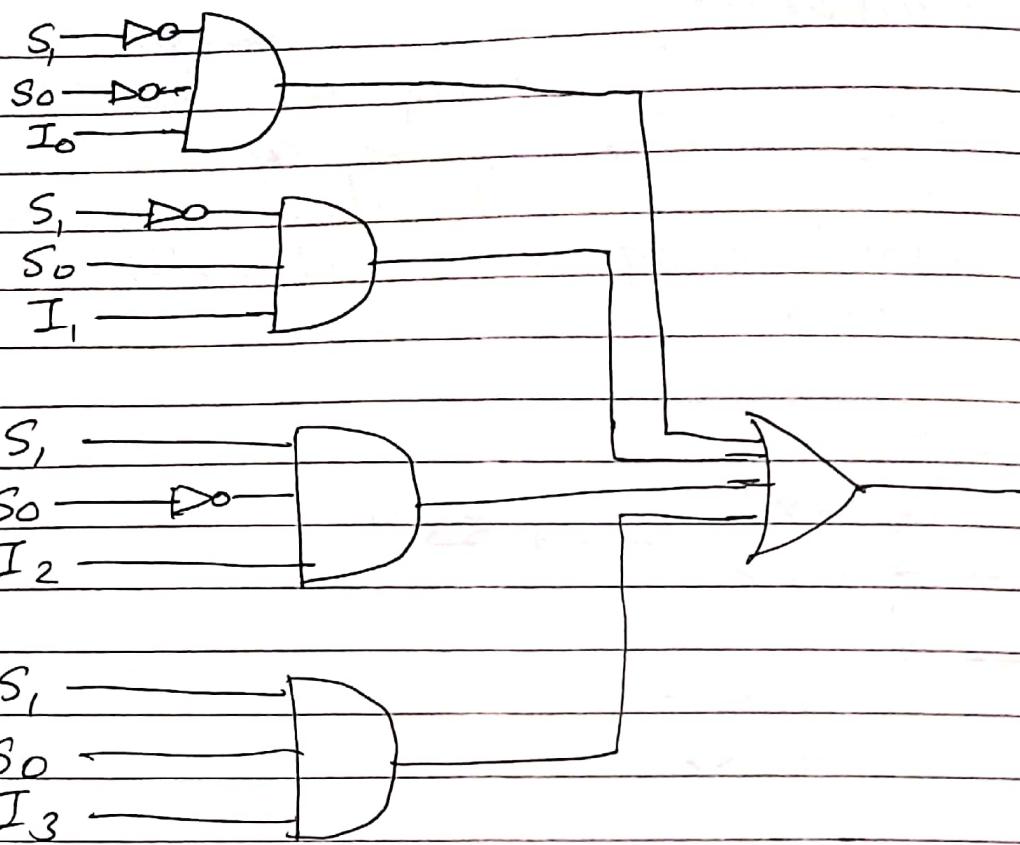
A 4x1 multiplexer has 4 data inputs I_3, I_2, I_1 , and I_0 and 2 select from lines S_1 and S_0 with output one output y . The block diagram of 4x1 multiplexer is given below:



Truth table for multiplexer is given below:

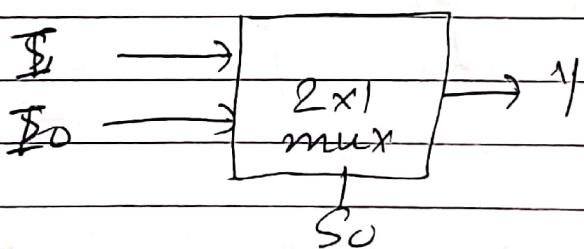
Selection line		Output
S_1	S_0	y
0	0	I_0
0	1	I_1
1	0	I_2
1	1	I_3

$$y = S_1 S_0' I_0 + S_1' S_0 I_1 + S_1 S_0 I_2 + S_1' S_0 I_3$$



Design 2x1 multiplexer

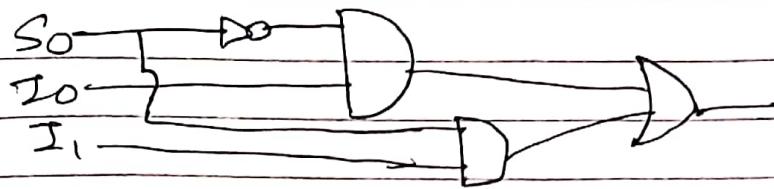
Input I_0 & I_1
Selection line S_0



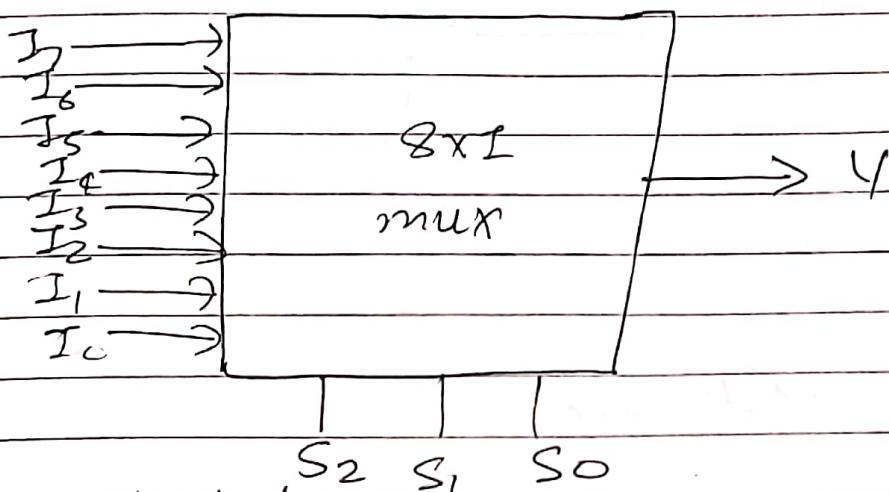
Truth table

$$Y = S_0' I_0 + S_0 I_1$$

Selection	Output
0	I_0
1	I_1



8x1 multiplexer

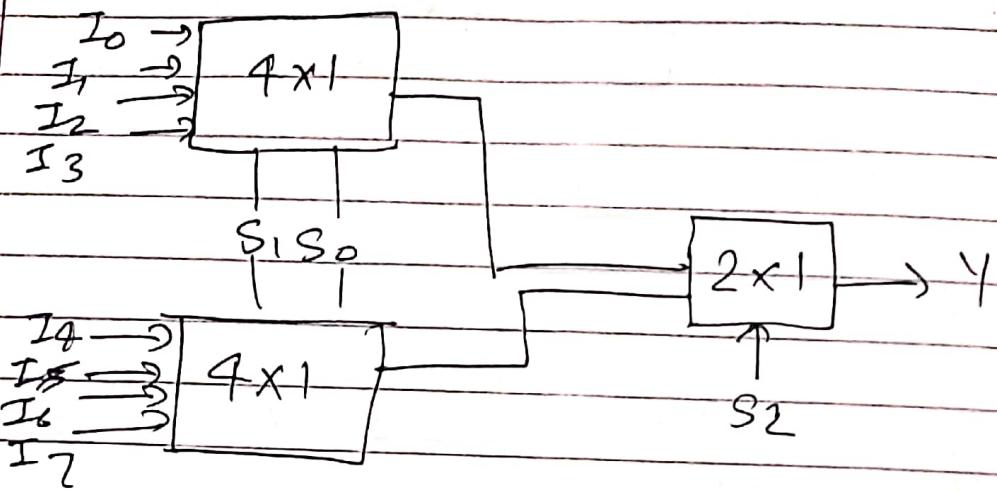


Truth table

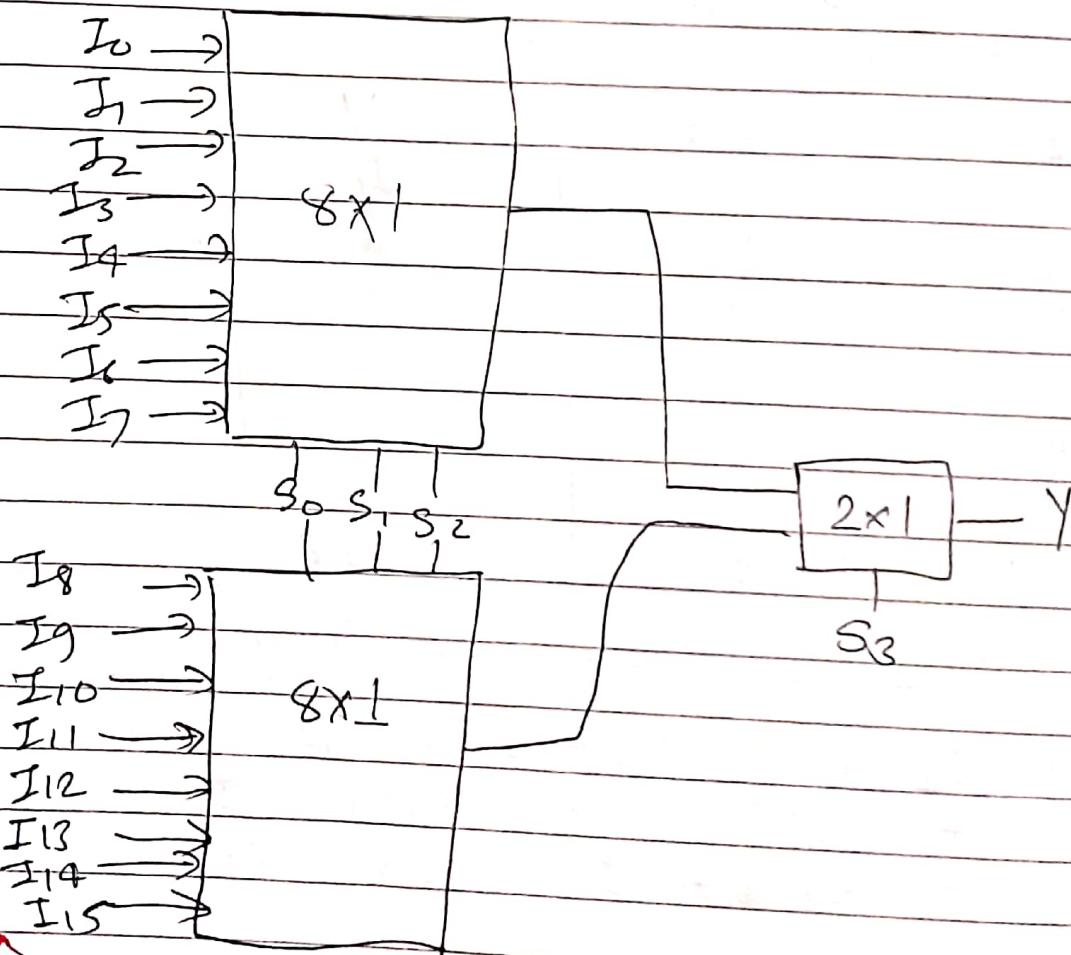
S ₂	S ₁	S ₀	Y
0	0	0	I ₀
0	0	1	I ₁
0	1	0	I ₂
0	1	1	I ₃
1	0	0	I ₄
1	0	1	I ₅
1	1	0	I ₆
1	1	1	I ₇

$$\begin{aligned}
 Y &= S_2' S_1' S_0' I_0 + S_2' S_1' S_0 I_1 + S_2' S_1 S_0' I_2 \\
 &+ S_2' S_1 S_0 I_3 + S_2 S_1' S_0' I_4 + S_2 S_1' S_0 I_5 \\
 &+ S_2 S_1 S_0' I_6 + S_2 S_1 S_0 I_7
 \end{aligned}$$

8 x 1 Multiplexer



16 x 1 Multiplexer



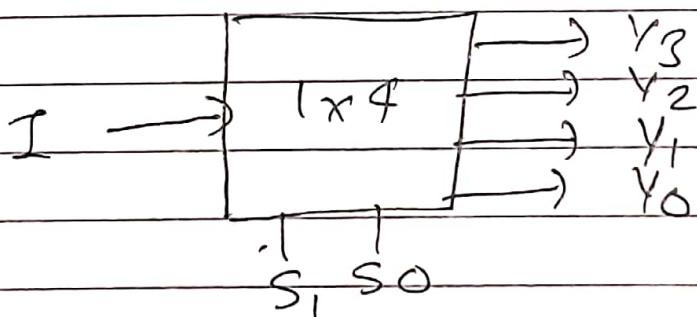
De-Multiplexer (DMUX)

De-multiplexer is a combinational circuit that performs the reverse operation of multiplexer. It has single input, n selection lines and maximum number of 2^n output. The input will be connected to one of these outputs based on the values of selection lines.

Since, there are ' n ' selection lines, there will be 2^n possible combination of 0 & 1 so each combination can select only one output. Demultiplexer is also called D-MUX.

1x4 De-Multiplexer

In this de-multiplexer, there is one input I , 2 selection lines S_1 & S_0 and 4 outputs Y_3 , Y_2 , Y_1 & Y_0 .



The single input I will be connected to one of the four outputs based on the values of selection lines.

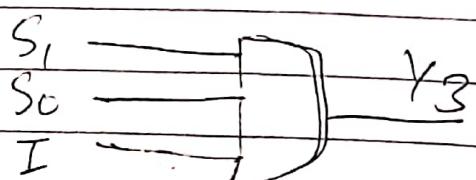
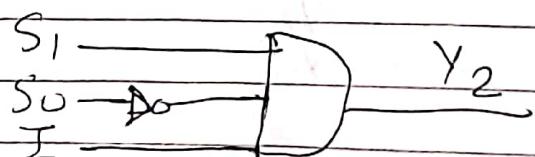
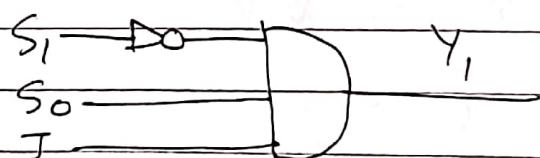
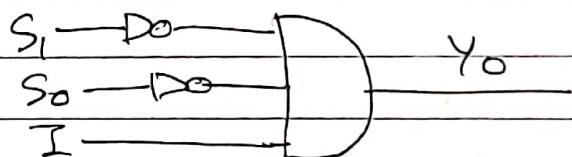
S_1	S_0	Y_3	Y_2	Y_1	Y_0
0	0	0	0	0	I
0	1	0	0	I	0
1	0	0	I	0	0
1	1	I	0	0	0

$$Y_0 = S_1' S_0' I$$

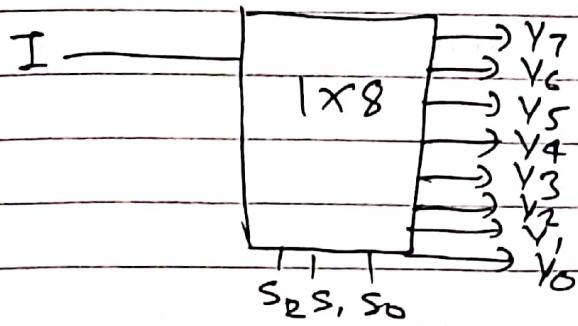
$$Y_1 = S_1' S_0 I$$

$$Y_2 = S_1 S_0' I$$

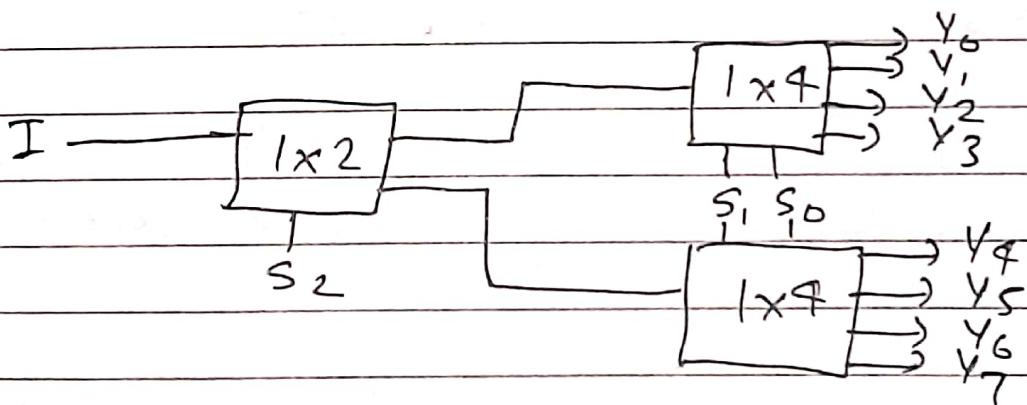
$$Y_3 = S_1 S_0 I$$



1x8 De-mux



S_0	S_1	S_0	Y_7	Y_6	Y_5	Y_4	Y_3	Y_2	Y_1	Y_0
0	0	0	0	0	0	0	0	0	0	I
0	0	1	0	0	0	0	0	0	I	0
0	1	0	0	0	0	0	0	I	0	0
0	1	1	0	0	0	0	I	0	0	0
1	0	0	0	0	0	I	0	0	0	0
1	0	1	0	0	I	0	0	0	0	0
1	1	0	0	I	0	0	0	0	0	0
1	1	1	I	0	0	0	0	0	0	0



Combinational

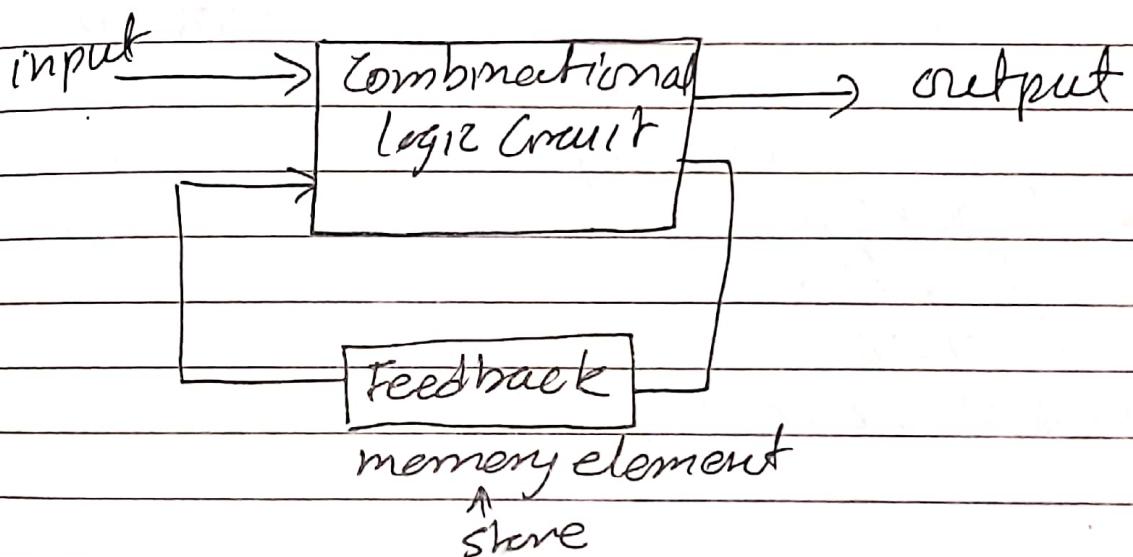
- 1) In this output depends upon present input.
- 2) Speed is fast.
- 3) The design is easy.
- 4) There is no feedback between input & output.
- 5) This is time independent.
- 6) Logic gates are the elementary building blocks.
- 7) It is used for arithmetic as well as boolean operations.
- 8) Combinational circuit does not have capability to store any state.

Sequential

- 1) In this output depends upon present as well as past input.
- 2) Speed is slow.
- 3) The design is difficult.
- 4) There exists a feedback path between input & output.
- 5) This is time dependent.
- 6) Flip-flops are the elementary building blocks.
- 7) Mainly used for storing data.
- 8) Sequential circuit can store any state.

- 9) Combinational donot have clock pulse because they don't require triggering.
- 10) This circuit donot have memory element.
- 11) Example: Encoder, Decoder, multiplexer etc.
- 9) Sequenched circuit have clock pulse.
- 10) This circuit have memory element.
- 11) Example: flipflop, counter

Sequential logic Circuit



Two types of memory element

- Latch
- Flip-Flop

clock pulse - 



Latch:-

Latch is a memory storage device which has no clock pulse but has two state.

Flip-flop:-

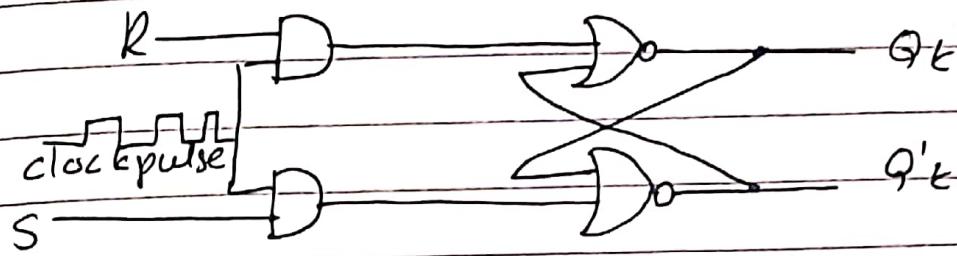
Flip-flop are memory elements for storing binary information. It has two output levels.

- one normal
- other complement

It stores & maintain the binary information. It has clock pulse and needs triggering. There are following types of flip-flops.

- ① Clocked RS flip flop
- ② D- flip flop
- ③ JK flip flop
- ④ T- flip flop
- ⑤ Master slave flip flop

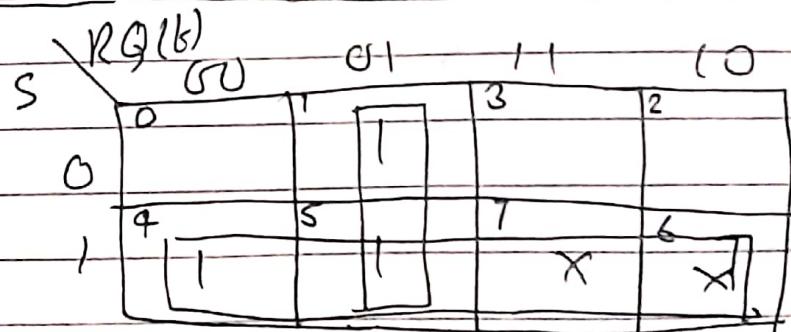
① Clocked RS flipflop



This circuit has two inputs S and R and two outputs Q_E and $Q'E$. This circuit works when the clock pulse is high (1).

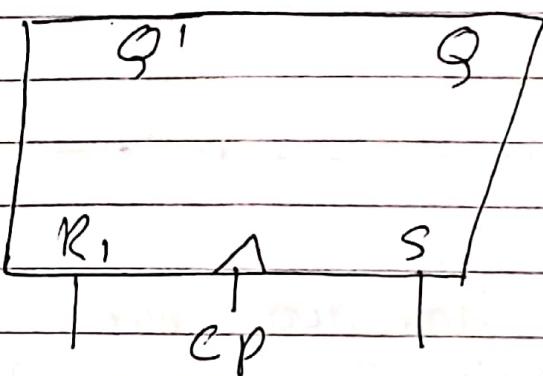
Following is the characteristics table for RS flip-flop

S	R	Present state Q_E	Next State $Q(t+1)$
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



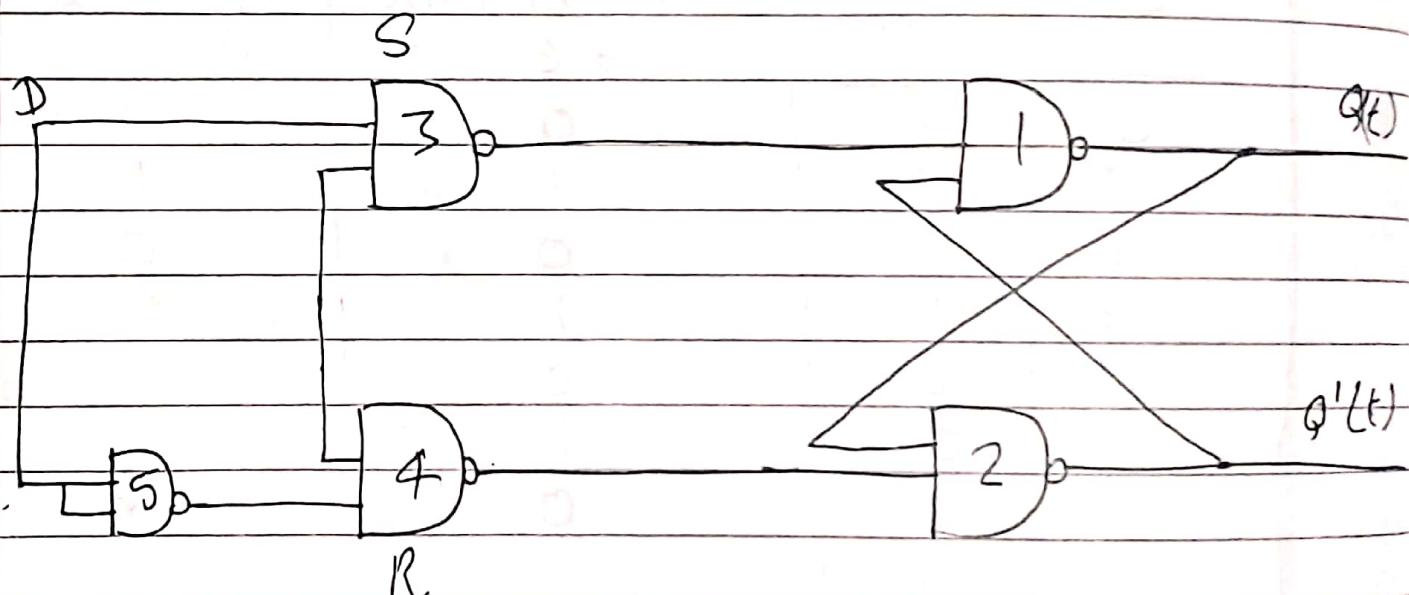
$$Q(t+1) = S + R' Q(t)$$

Symbol:



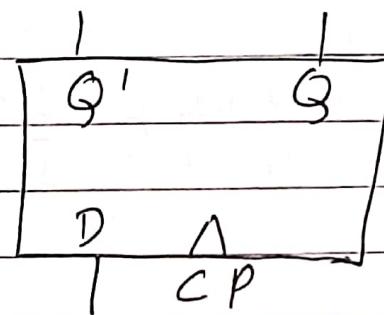
* D-Flipflop

Following is the logic circuit for D-Flipflop



Here, the NAND gate 1 & 2 form a basic flip flop and gate 3 & 4 modify it to a clocked RS flip flop. The D input goes directly to the S input and its complement through gate 5 to R input when CP is equal zero. Create 3 and 4 have one in their outputs.

The following is the symbol of D-flip flop



Now;

Characteristics table, when CP=1

D	Q _t	Q _{t+1}
0	0	0
0	1	0
1	0	1
1	1	1

Characteristic Equation

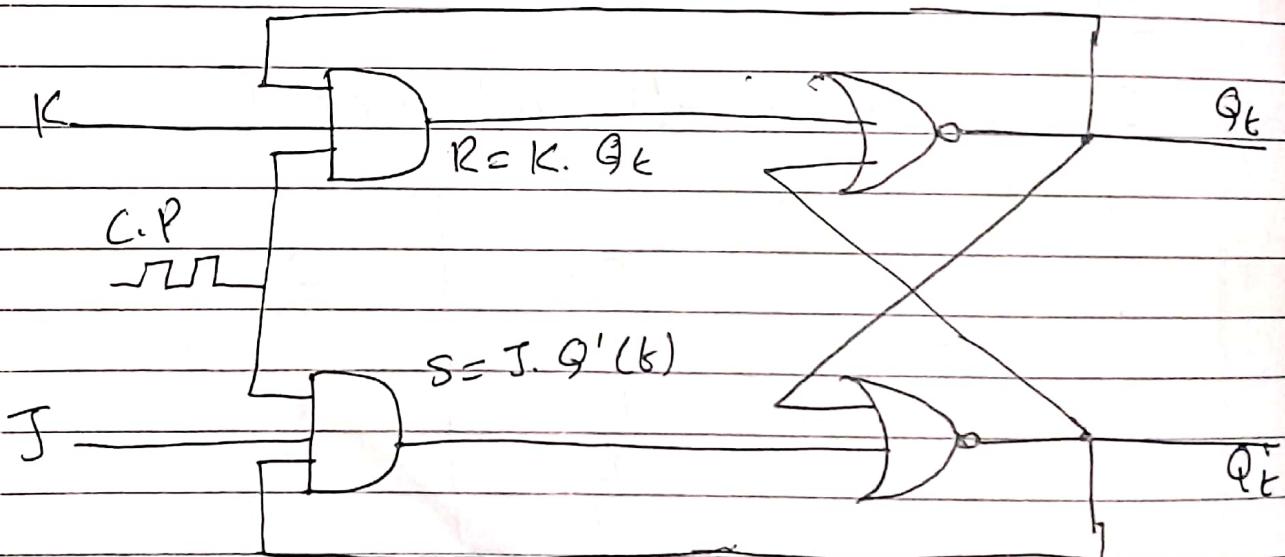
D	Q _t	Q	1
0	0	1	
1	2	1	3

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

JK-Flipflop

JK-Flipflop is the modified version of RS flipflop.
It operates on positive clock pulse. The circuit has two input J & K and two output Q_t & Q'_t

Following is the circuit diagram of JK-Flipflop



Logic circuit of JK flipflop

Characteristics table of JK flipflop

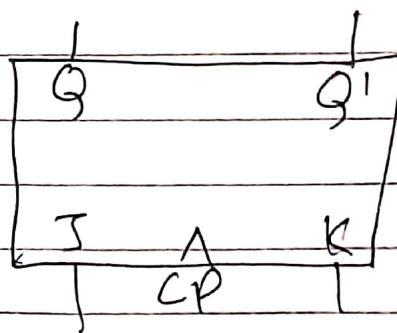
P. S

N. 8454

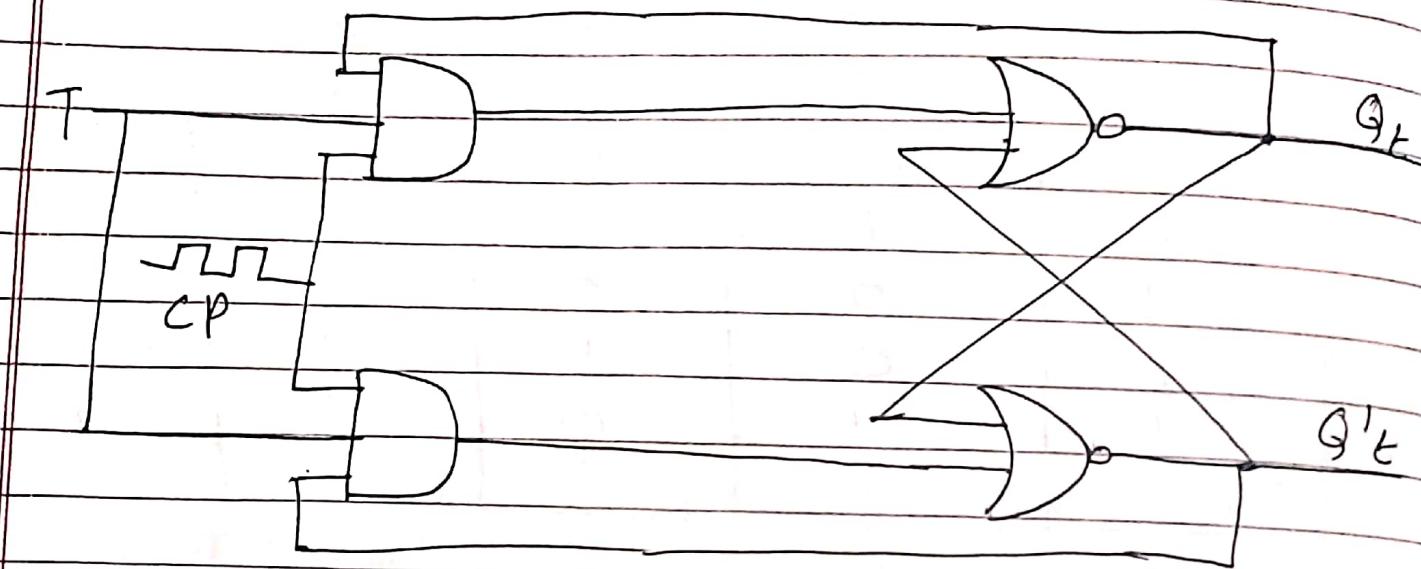
J	K	Q_E	Q_{E+1}
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	0

JK		Q _E	Q _{E+1}	
J	K	00	01	11
0	0	0	1	1
0	1	1	0	0
1	0	1	0	1
1	1	0	1	1

$$Q_{E+1} = K' Q_E + J Q_E'$$



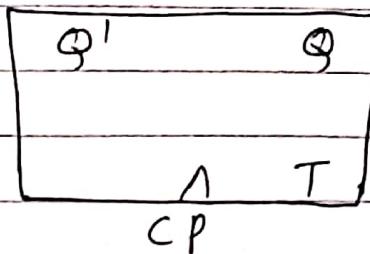
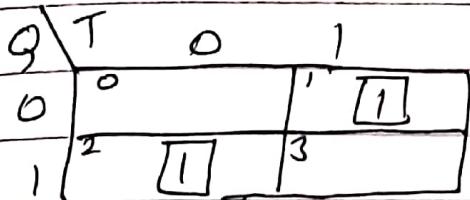
T-Flip Flop (Toggle flip flop)



The designation of T-Flip Flop comes from the ability of flip flop to toggle or change the state. The T-Flip Flop is the single input version of JK flip flop where both inputs J & K are tied together to obtain T-Flip Flop. Here, regardless of the flip flop present state it assume the complement state when clock pulse occurs while input T is at logic high.

Characteristic table

Q_T	T	Q_{T+1}
0	0	0
0	1	1
1	0	1
1	1	0

K-mqPsymbol

$$Q_T + 1 = Q T' + Q' T$$

Master-Slave flipflop

A master slave flipflop is constructed from two separate flipflop, where in one flipflop ~~serves~~ serves / acts as master and the other act as slave. An R.S master slave flipflop logic circuit is shown in the figure where the first is master & second is slave.

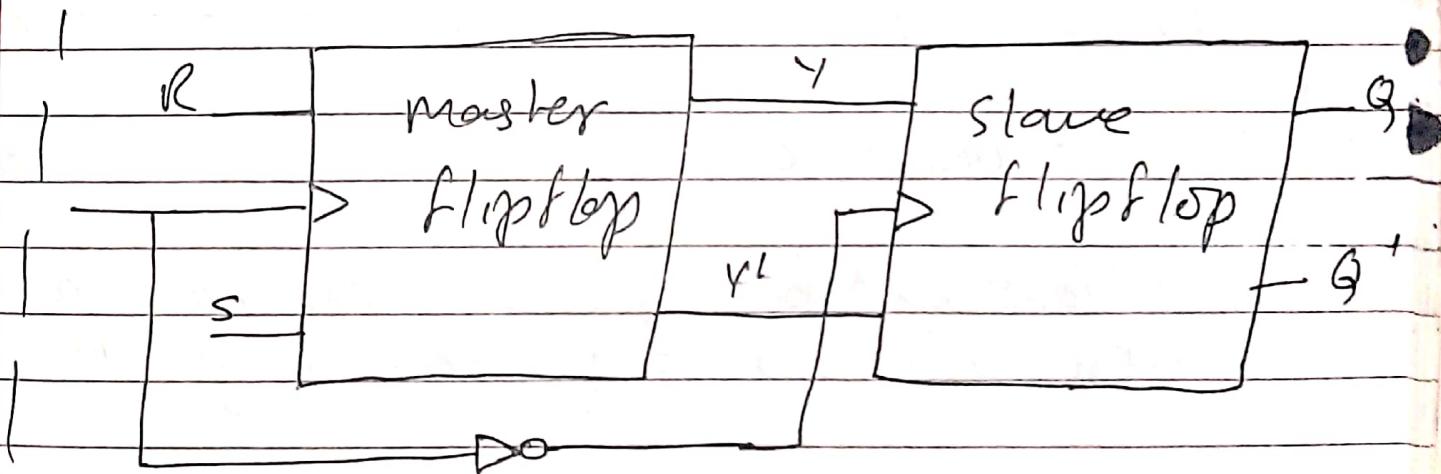


Fig: Master Slave flipflop

Here, when CP or clock is zero, the inverter (NOT gate) output becomes one. Since the slave is one. $Q = Y$ & $Q' = Y'$. Here, the CP is zero, so, the master is disable.

When the clock pulse is one by the external RS inputs is transmitted to the master flip flop. The slave flip flop however is isolated as long as the pulse is at level 1. Master slave flip flop is used for trigger in the logic circuit.

Register

Application of Flipflop

- Used for storage purpose.
- Flipflops are used to construct registers.
- They are also used in sequential circuits and other computing devices such as timers, counters
- In the design of CPU, flip-flops are necessary.
- flip-flops can be used as flags such as carry flag, zero flag, sign flag etc.

Registers

Register consists of group of binary storage cells suitable for holding binary information. A register includes a set of flipflops and 'n' bit register has ' n ' ~~bit~~ flipflop. Therefore, a group of flipflops constitute ~~is~~ a register.

(*) Shift Register:

A register capable of shifting its binary information either to right or to the left is called shift register.

A shift register logical configuration consists of a chain of flipflop. In flipflop are connected in cascade.

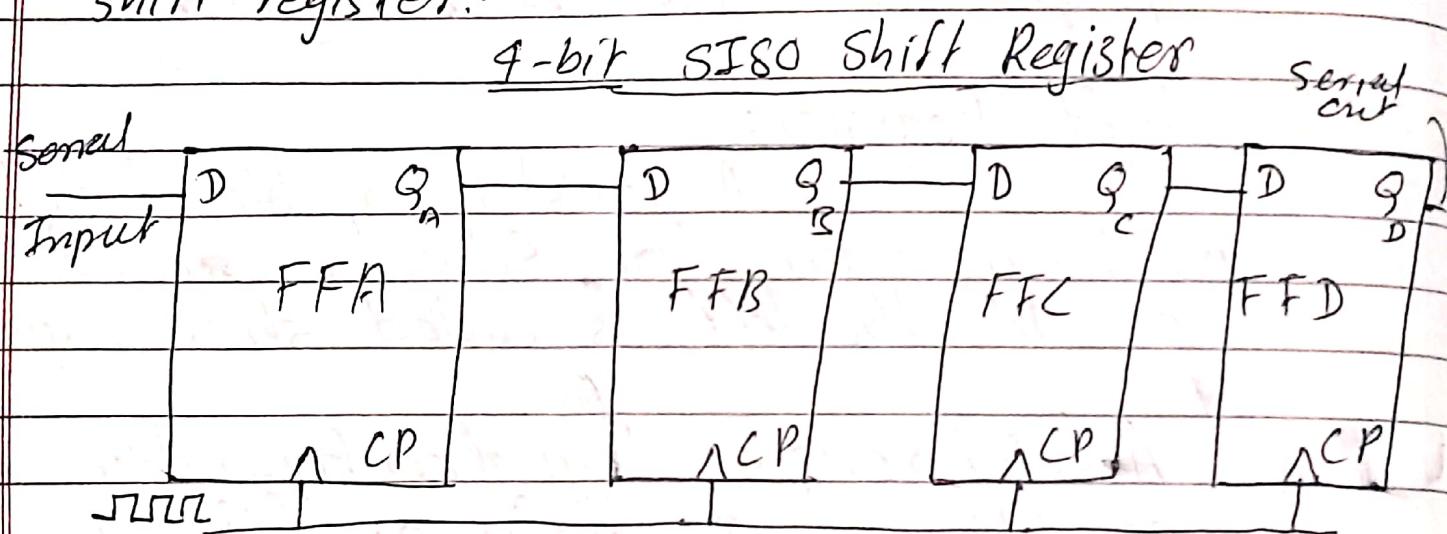
There are 4 types of shift register according to their different modes:

- ① SISO (Serial in Serial Out)
- ② SIPO (Serial in Parallel Out)
- ③ PISO (Parallel in Serial Out)
- ④ PIPO (Parallel in Parallel out)

① SISO (Serial In Serial out) Shift register

The shift register which allows serial input and produces serial output is known as SISO shift register.

The following is the block diagram of SISO shift register:-



11001

Testing

clock pulse A B C D

0 0 0 0 0

1 1 0 0 0

2 0 1 0 0

3 1 0 1 0

4 1 1 0 1

5 0 1 1 0

6 0 0 1 1

7 0 0 0 1

8 0 0 0 0

} storing

} removing

In the ~~figure~~ block diagram, there are 4 ~~bit~~
D flipflops which are cascaded. All
these flipflops are synchronous with each other.
Since, the same clock pulse is applied
to all the flipflop. Here, the input is
serially given to the ~~left~~ left most
flipflop. The input is also called serial
input.

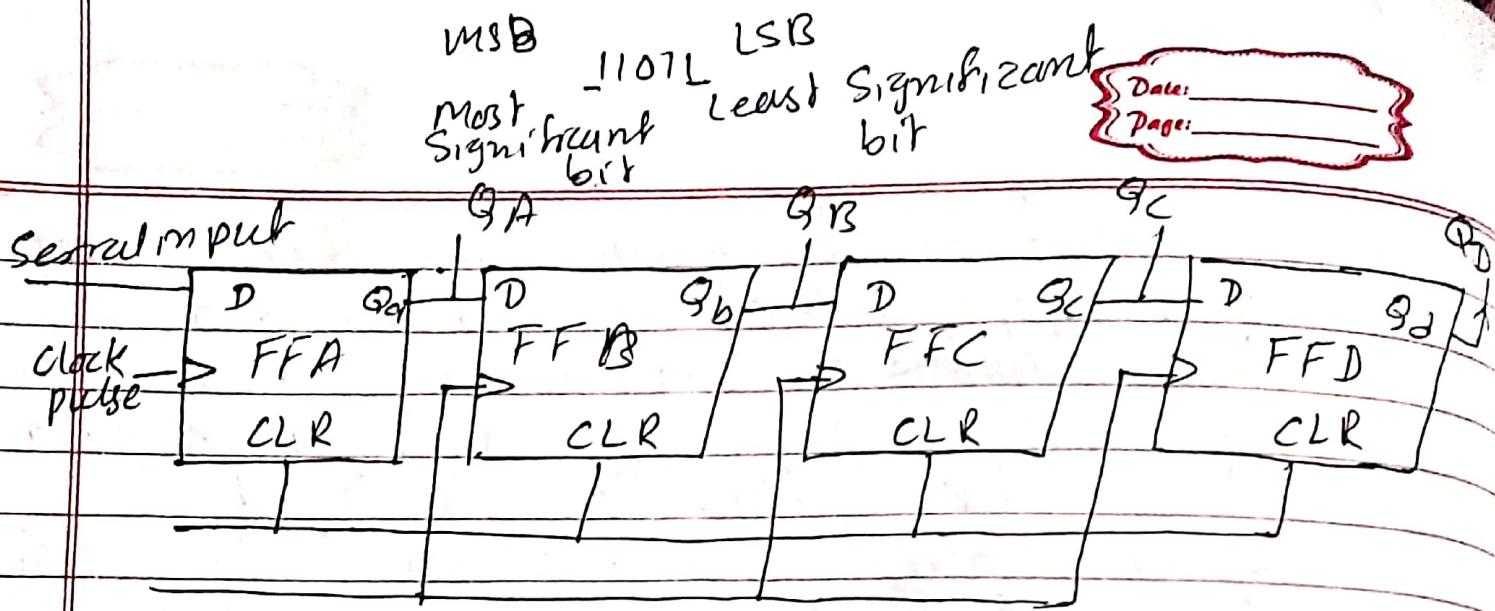
For every positive edge triggering
of clock pulse, the data shift from
one flip flop to other. We can receive the
bits serially from the output of
the right most flip flop. The output is
also called serial output.

SISO (Serial In Parallel Out)

The shift register which allows serial input and
produces parallel output is known as
SISO shift register.

The following is ~~had~~ the block diagram of
SISO shift register

4-bit SISO register



In this shift register, the register is loaded with serial data one bit at a time with the stored data being available in parallel form. In the above figure, all the flipflops have just been reset using clear signal and all the outputs (Q_A, Q_B, Q_C, Q_D) are at logic zero.

Testing

Clock Pulse	Q_A	Q_B	Q_C	Q_D
0	0	0	0	0
1	1	0	0	0
2	0	1	0	0
3	1	0	1	0
4	1	1	0	1

If we give the serial input to the SIPO shift register, the LSB ~~is~~ 1 is given input to A flip flop in the clock pulse. In the second clock pulse, the value from the flip flop A shifts to the

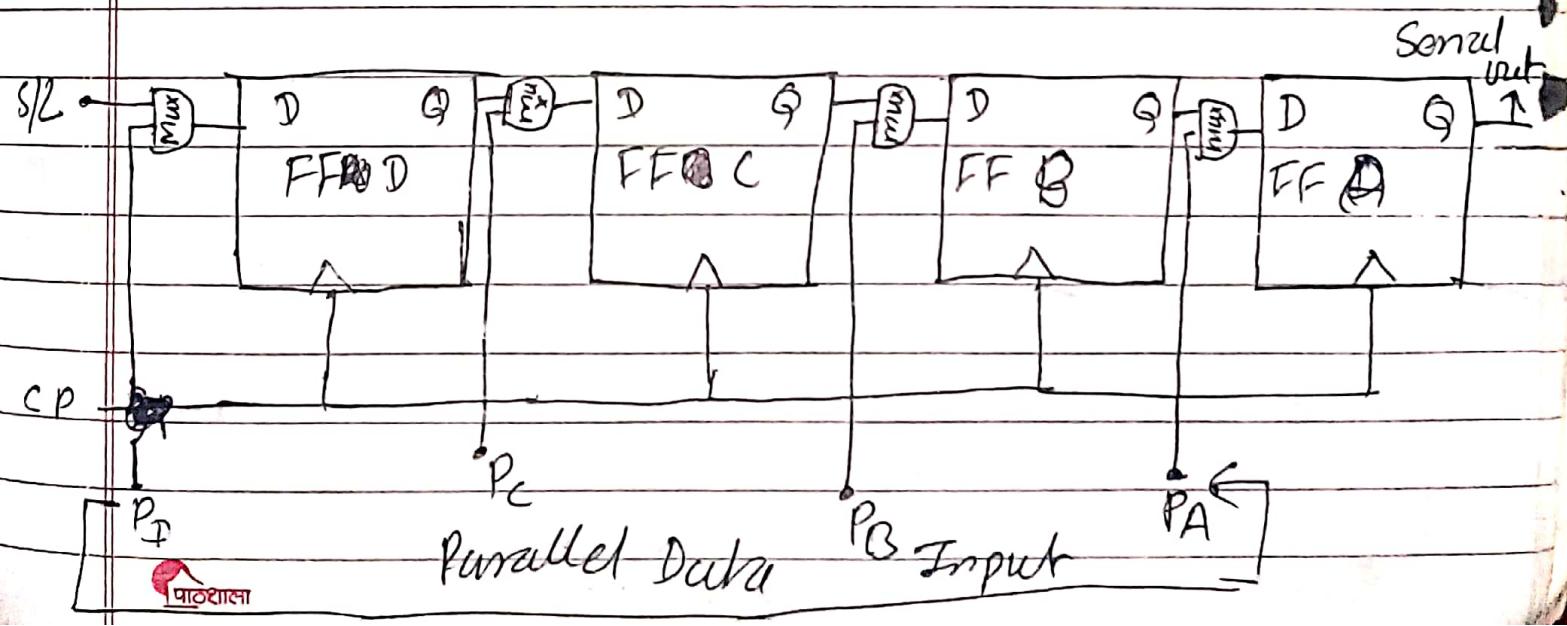
flip flop B and the second input enters the first flip flop. the process continues unless the MSB is seen in the first flip flop.

The process of shifting and storing the data in parallel and shows as parallel to output form which is clearly showing in the above table.

~~Uses~~ SIPO Shift Register is used for serial to parallel conversion of data.

PISO (Parallel In Serial Out) Shift register

In PISO (Parallel in serial out) shift register, ~~the~~ the parallel data is loaded into the register simultaneously and it shifted out of the register serially one bit at a time under clock control. The following is 4 bit PISO register:-



Here, the data is loaded simultaneously to the parallel input pins P_1 to P_4 of the register.

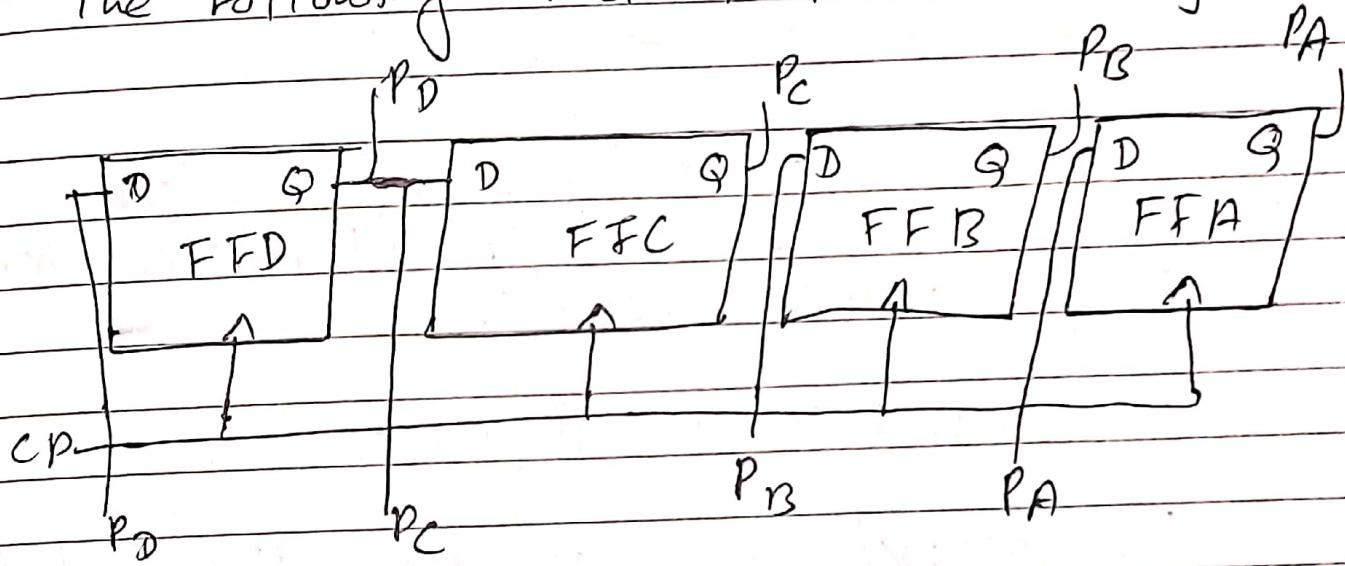
The data is read out sequentially as the serial output. This is the 4 bit register so 4 clock pulse are required to shift the data serially. This kind of shift register is used for parallel to serial conversion.

0	0	0	0	0
1	1	1	0	1
2	0	1	1	0
3	0	0	1	1
4	0	0	0	1
5	0	0	0	0

PIPO (Parallel in Parallel Out)

PIPO shift register is where the parallel data is loaded simultaneously in the register and transferred together to their respective output with the same clock register.

The following 4 bit PIPo shift register



simplest

PIP0 is the ~~shiftless~~ mode of shift register. It has only 3 connections which determine the output of the shift register. This type of register is used as temporary storage device or time delay device. There is no interconnection between the flip flops.

~~Counters~~ Counters

A counter is essentially a register that goes through a pre-determined sequence of states upon the application of input pulses. It is obvious from its name a counter is used to count the pulses. They are used in digital watches and frequency counters.

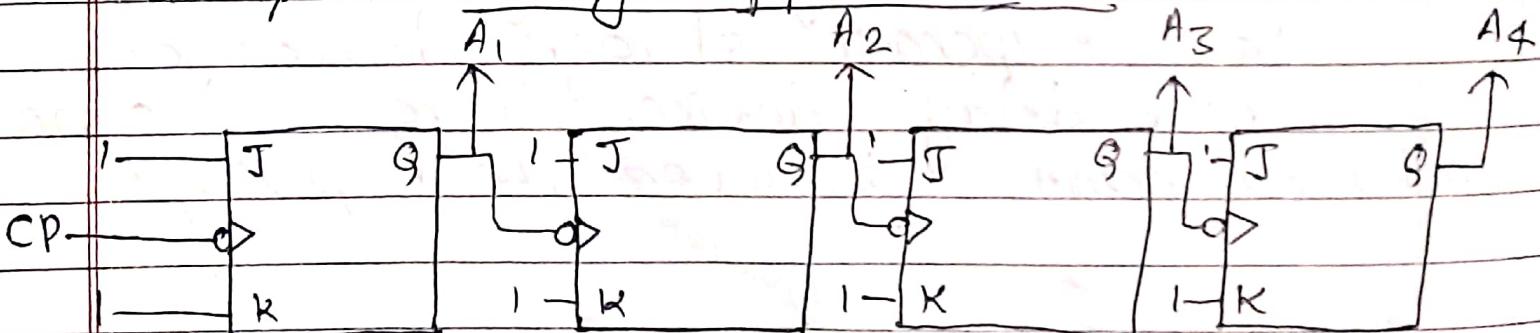
There are two types of Counters:-

- i) Asynchronous counter (Ripple counter)
- ii) Synchronous counter

i) Asynchronous counter (Ripple counter)

In asynchronous counter, the first flip flop is clocked by the external clock pulse and then each successive flip flop is clocked by the output Q or Q' of previous flip flop.

Example: Binary ripple counter



The above figure is 4-bit, ripple counter. Here, the output of each flip flop is connected to the clock input of next ~~flipflop~~ higher order flip flop.

As shown in the figure, the flipflops holding LSB receives the incoming count pulses. The J & K inputs are high, the small circle in clock input indicates that the flipflop is active when CP is zero.

(0)CP	A ₀	A ₃	A ₂	A ₁
0	0	0	0	0
1.	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0
9	1	0	0	1
10	1	0	1	0
11	1	0	1	1
12	1	1	0	0
13	1	1	0	1
15	1	1	1	1

A₁ will complement on every count pulse.
A₂ will be complement if A₁ goes from 1 to 0
A₃ will be complement if A₂ goes from 1 to 0

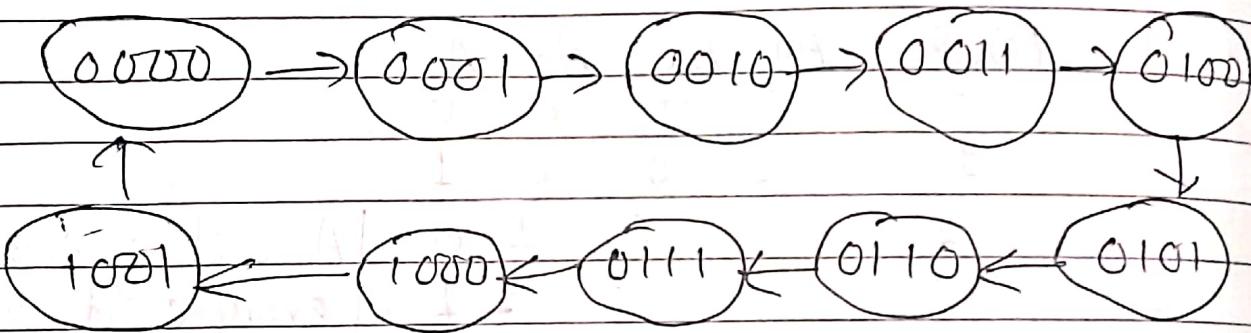
A₀ will be complement if A₃ goes from 1 to 0.

BCD Ripple

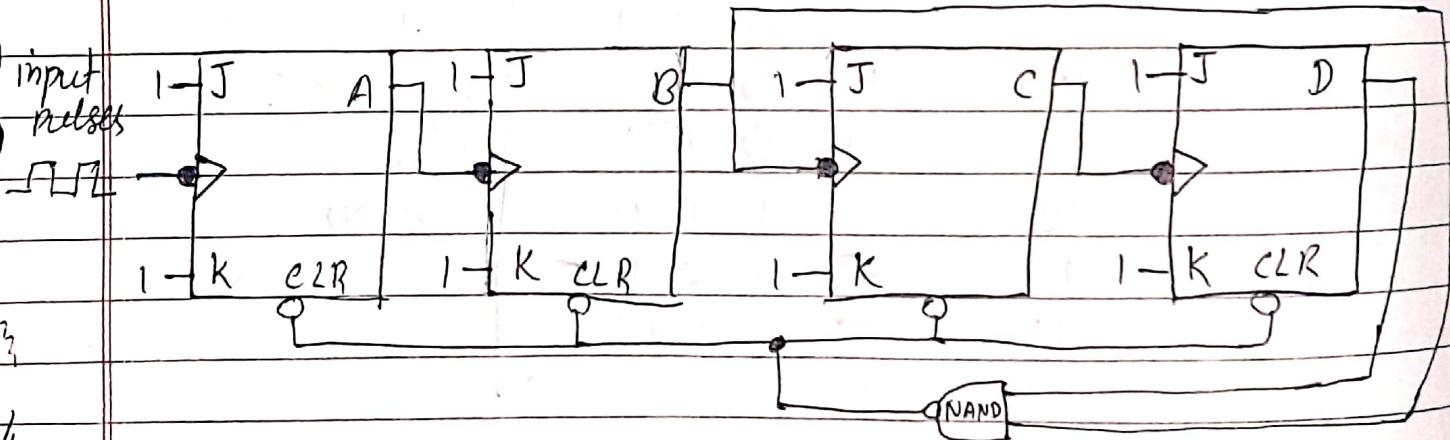
Ex:- 2) Decaded / Decimal Counter

A ~~BCD counter~~

A decaded counter follows the sequence of ~~of~~ 10 states. Such a counter must have 4 flipflops to represent a decimal digits.



state diagram



A 4 bit binary counter will act as decoded counter by skipping any 6 outputs of the 16 outputs. This is an asynchronous decoded counter. Given figure is the asynchronous decoded counter.

The above figure shows the decoded counter constructed with JK-flipflop. The J output and K output are connected to logic one. The clock input of every flipflop is connected to output of next flipflop. The output of NAND is connected in parallel to the clear input to all the flipflop.

When the decoded counter is at rest. The count is equal to ~~0000~~^{A B C D} 0000. This is first stage of the counter cycle. When we connect a clock signal input to the counter circuit, then the circuit will count the binary sequence. The first clock pulse can make the circuit to a count upto 0 to 9. The next clock pulse advances to count 10. When the parts B & D will be high the NAND gate output is low and all the flipflop input will be cleared.

* Synchronous counter:-

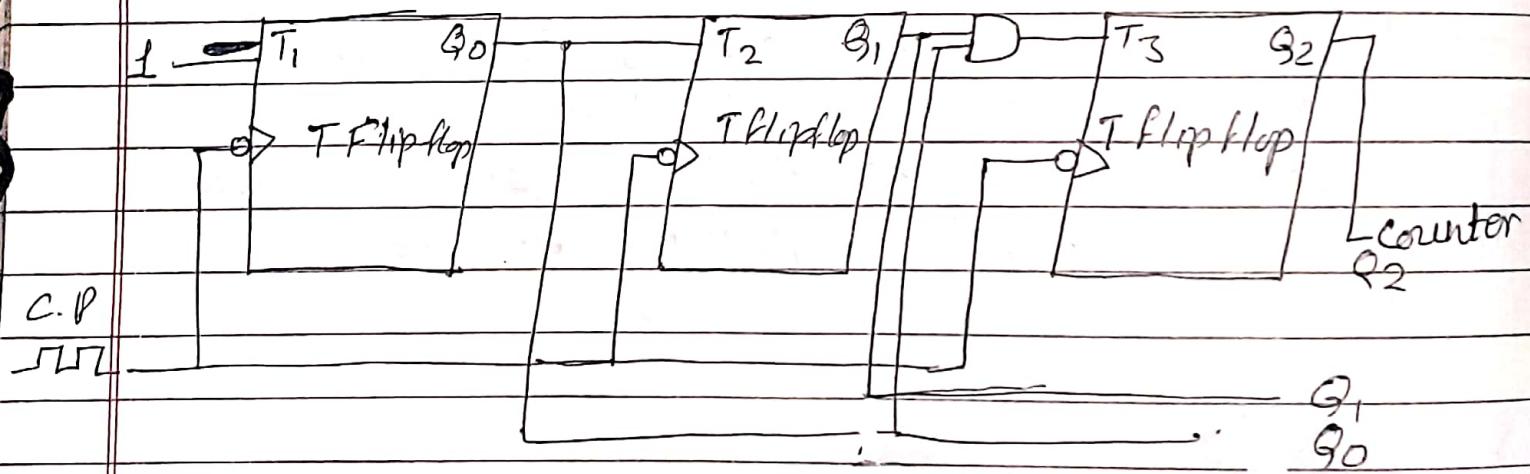
In synchronous counter the clock pulse input is connected to all of the flipflop and therefore they are clocked simultaneously. They are faster than asynchronous counter.

There are ~~too~~ 3 types of synchronous counter:-

- 1) Synchronous Binary up counter
- 2) Synchronous Binary down counter
- 3) Synchronous Binary up/down counter

I) Synchronous Binary up Counter:-

An 'N' bit of synchronous binary up counter consists of 'N' T flip-flops. It counts from 0 to $2^N - 1$. The block diagram of 3-bit synchronous binary up counter is given below:-



The 3-bit synchronous binary up counter contains 3 T-flipflops and one, ~~two~~-input AND Gate. All these flipflops are negative edge triggered (1 to 0 works) and the outputs of the flip flop change synchronously.

The T input of first, second and ~~third~~ third flipflop are \bar{I} , $Q_0 \& Q_1$, Q_0 respectively.

The output of first T-flipflop changes for every negative edge of clock pulse.

The output of second T-flipflop changes if Q_0 is equal to 1 on every negative edge of clock pulse.

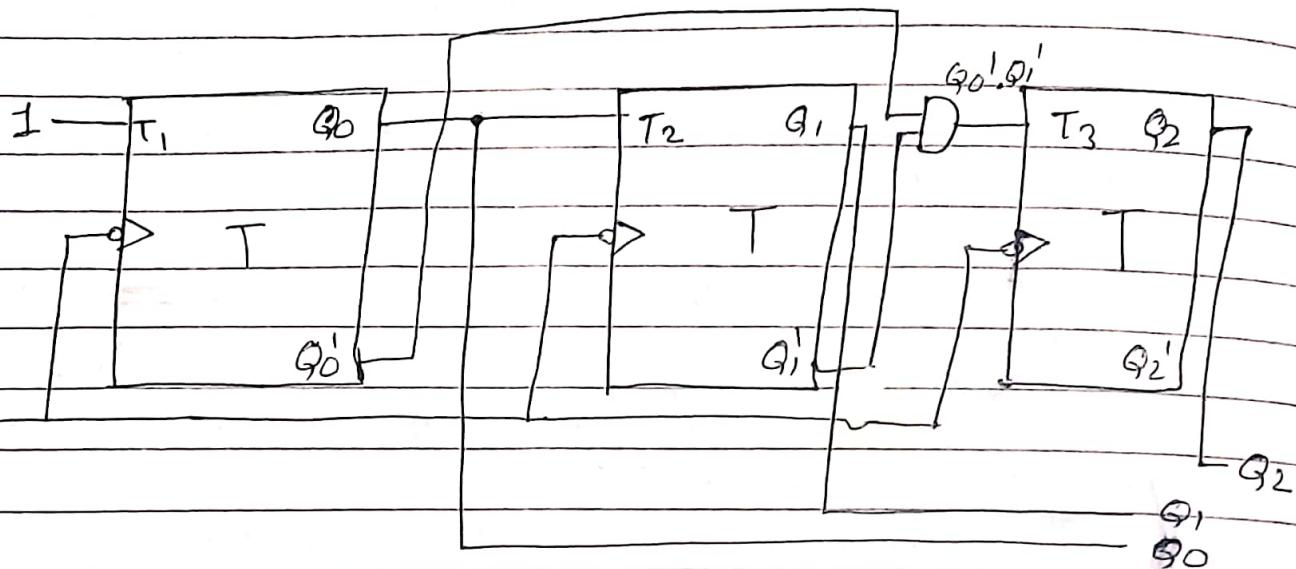
The output of third T-flipflop changes if both Q_0 and Q_1 are one.

Testing:

$CP=0$	Q_2	Q_1	Q_0
0	0	0	0
0	0	0	1
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	1

Date: _____
Page: _____

2) Synchronous Binary down counter



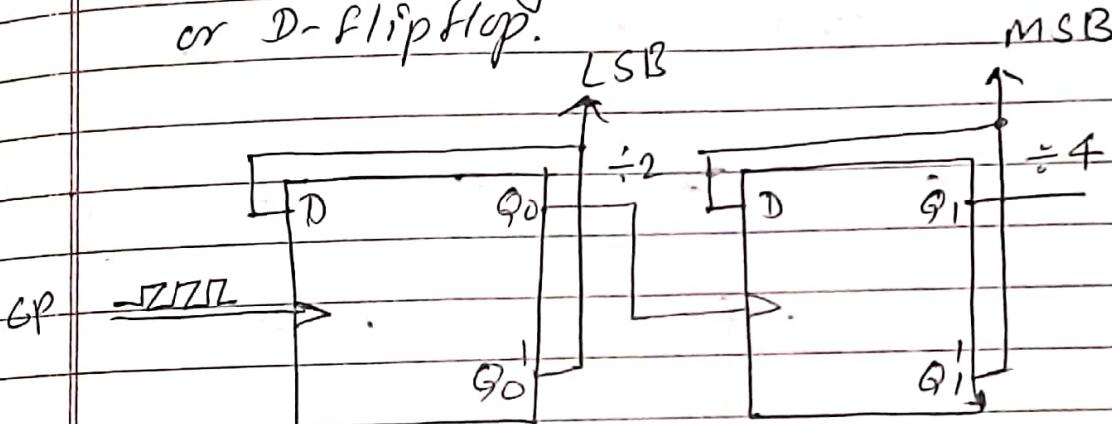
Mod counter / m- counter / Modulo-counter :-

Modulus counter a simply mod counter one defined based on the number of states that the counter will sequence through before returning back to its original value.

For example:-

A 2-bit counter that counts from 00_2 to 11_2 . In binary has modulus value of 4 ($00 \rightarrow 01 \rightarrow 10 \rightarrow 11$ and back to 00) so therefore it is called mod-4 counter.

When designing mod counter we use T flip flop or D flip flop.

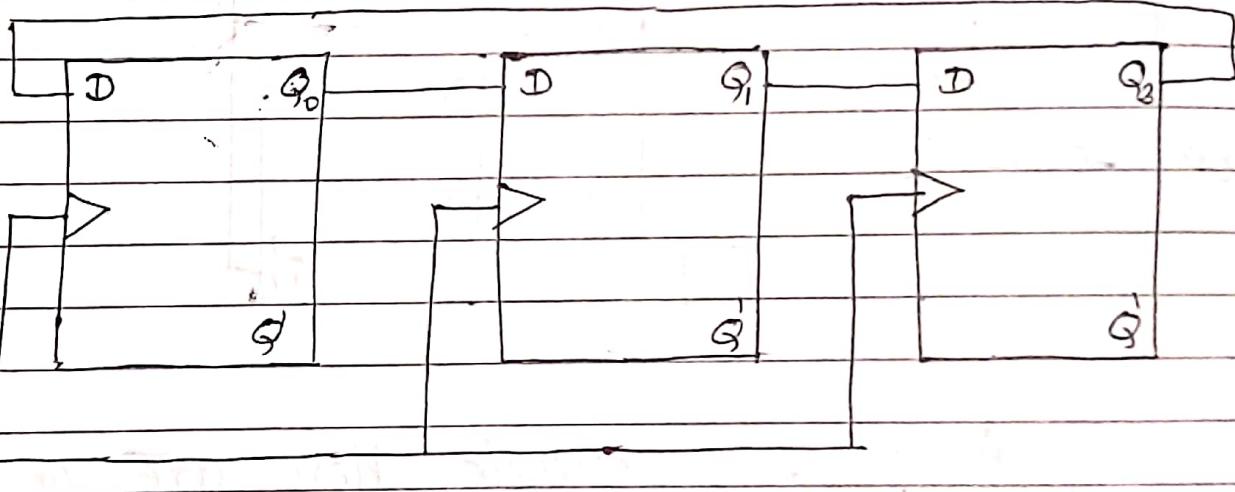


As in this example, here are only 2 bits then the maximum numbers of possible output states for the counter is $2^n = 2^2 = 4$. However, counters can be designed to count to any numbers of 2^n states. In their sequence by cascading multiple counting stages to produce a single modulus or

MOD-N counter.

Ring Counter:-

A ring counter is a shift register (cascade connection of flip-flops) with the out of the last flipflop connected to input of the first flipflop. It is initialized such that only one of the flipflop output is 1 and the remainder is 0. The one bit is circulated so the states repeats every n clock cycle. if n flipflops are used. It can be designed using D or JK flipflop.



The truth table for given counter is as follows:-

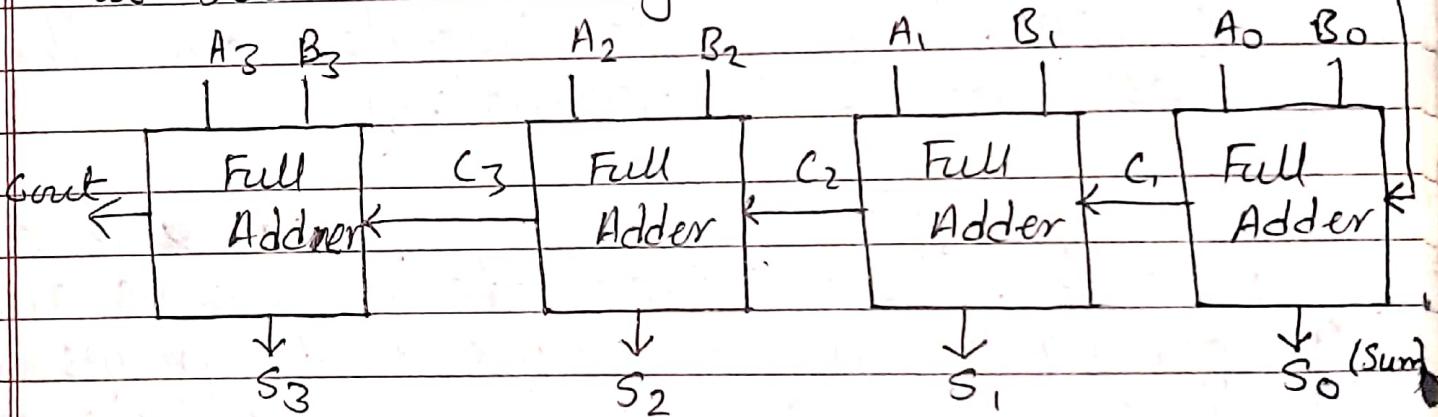
state	Q_0	Q_1	Q_2
0	1	0	0
1	0	1	0
2	0	0	1

From the following truth table, it is clear that the one output is seen at only output of the flip flop and it transfer to other flip flop at every clock pulse.

4 bit

(*) Explain a parallel adder :-

⇒ The parallel adder uses 4 full adders as shown in the figure below:-



Here, the 4 full adder are cascaded. Each full adder is getting two parallel inputs from A & B . Carry output of one full adder will carry input of higher order full adder. The 4 bit binary adder produces

the resultant sum having at most 5 bits.

~~Lecture 9~~

Display

- ① LCD
- ② LED
- ③ Gas Display
(Plasma display)
- ④ Display

① LCD Display (Liquid Crystal Display)

Liquid Crystal Display (LCD) is a type of display technology that makes use of liquid crystals that open or close when stimulated by an electric current. These liquid crystals are the basis of LCD technology.

The liquid crystals are made of complex molecules. (Just like water) crystals are arranged in a matrix with groups of three crystals Red, Green and Blue, forming a segment known as pixel. Groups of pixels can form numbers, letters or shapes and are arranged in column and rows.

LCD is considered a major innovation in display devices and are used in electronic

like:- laptops, computer, smartphone and televisions.

② LED Display (Light Emitting Diode)

LED display is a screen display technology that uses a panel of LEDs as the light source. The biggest advantage of LED display is efficiency and low energy consumption which is necessary for handle devices like phones and other electronic gadgets. An LED display consists of no of LED panels that in turn consists of several LEDs.

LED have many advantages like:- it produces more brilliance and great light intensity. Different types of LED are found in market today such as QLED, AMOLED, QLED etc.

③ Gas Display (Plasma Display)

Gas charge display or plasma display was an earlier flat screen technology that use tiny cells lined with phosphor that were full of ~~inert~~ inert ionized gas (Xenon & Neon). This technology was used in flat panel display in televisions.

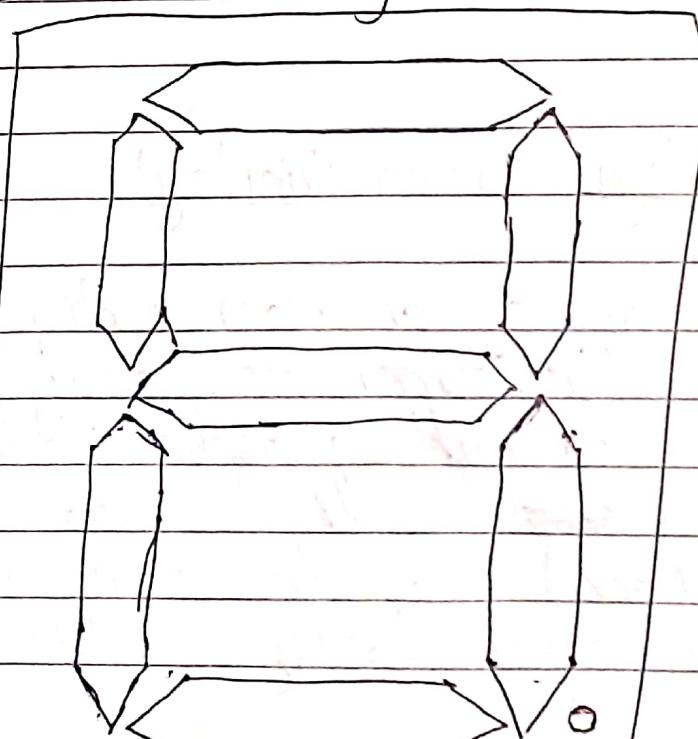
In plasma display, each pixel is comprised of 3 ions gas cells lined with red, green & blue phosphor. The amount of charge determines the intensity and the combination of different intensity of red, green and blue provides all the color.

* Seven Segment Display:-

LED (Light Emitting Diode)

LED is a solid state optical PN-Junction diode which emits light energy in the forms of photons.

The seven segment display also written as 7-segment display, consists of 7 LEDs arranged in rectangular fashion as shown in figure.



Symbol of led
~~actual shape~~

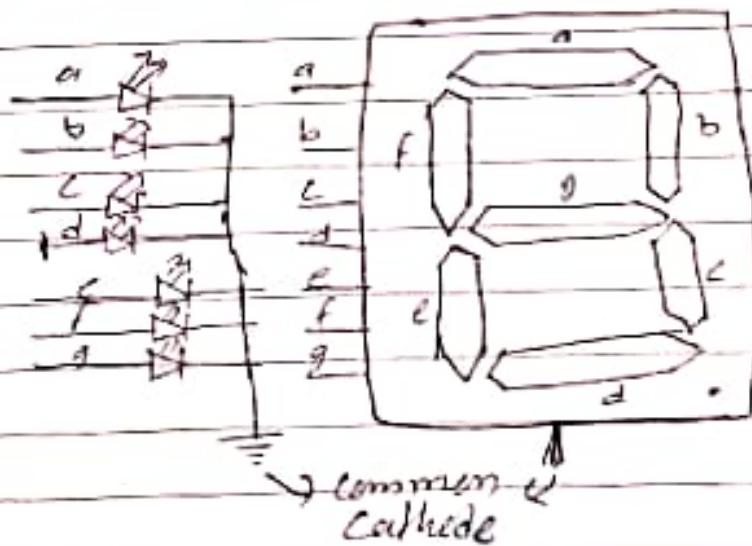
Each of the 7-LEDs is called a segment because when ~~illuminated~~ illuminated. The segment forms part of a numerical digits (both decimal & Hex) to be displayed. An additional 8th LED is sometimes used.

Here are two types of 7-segment LED display:

- ① Common Cathode
- ② Common Anode

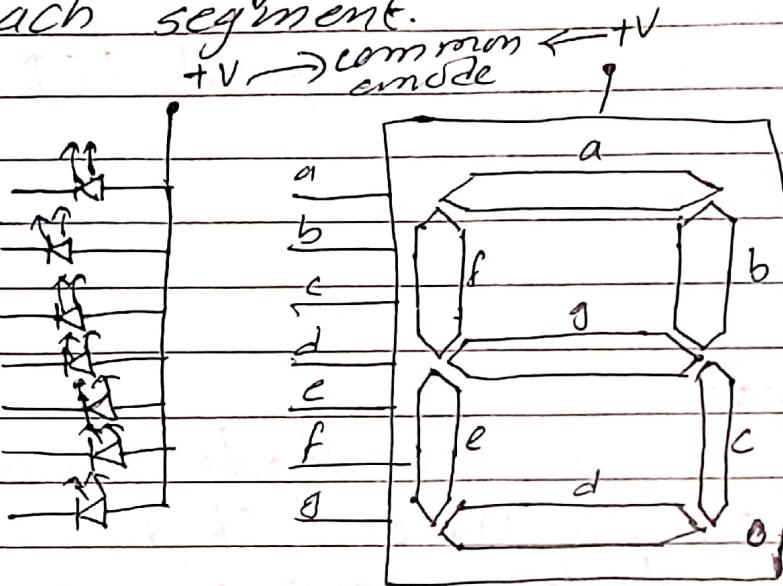
Common Cathode

In common cathode display all the cathode connections of LED segments are joint together to logic zero or ground. Individual segments are illuminated by logic one or high signal on each anode.



Common Anode

In common anode display, all the anode connections are joined together to logical one. The individual segments are illuminated by applying logic zero to cathode of each segment.



Alpha numeric display :-

The alpha numeric displays are used to display alphabets and numbers. These can be implemented by using LCD or LED displays. Alpha numeric display are easier to implement.

They are fabricated (made) in standard configurations. Alpha numeric display has been accepted as recognized technology for many years. They are designed in standard configurations such as LCD display (16x2, 8x1, 40x4) ~~for~~ for example:-

In 16×2 display where there are 16 character in each row and there are two rows of these character.

Note: A character can be letter (Capital or small, any number and other symbol like:- , * # etc.)

Difference between

Analog System

- ① It uses analog waves which are continuous periodic signals.
- ② Analog system can have different values on different levels.
- ③ It has usually large circuits & covers more area.
- ④ The measurement from analog system is less accurate.
- ⑤ It is used in analog filters, radio transmission.
- ⑥ It is old method of data transmission and is being replaced by digital system.

Digital System

- ① It uses digital signals which are not continuous but are discrete.
- ② Digital System uses binary system whose value can either be one or zero.
- ③ It has small circuits and covers small area.
- ④ The result of digital system is more accurate.
- ⑤ It is used in computers, mobiles etc.
- ⑥ It is new technology and is largely used in computers, communication and all other fields.

⑦ Analog devices are difficult to repair.

⑧ It consumes high energy. to operate.

⑦ Digital devices are easy to repair.

⑧ It consumes less energy for operation.

Application of digital System

- Used in data communication to send and receive data.
- All mobile communication systems are equipped with digital devices.
- In hospitals, digital system is used in X-rays, MRI and other medical devices.
- Computers are based on digital system and constructed using digital devices.
- Image processing, data science, computer networks and all other fields of information, science and technology use digital system and digital devices.

Date: _____
Page: _____

→ In short, all the systems in the world implement digital system for better results and comfort.