



# Government Engineering College

## Sec-28 Gandhinagar

Sem: - 3

Subject: - Digital Fundamental

Subject Code: - 3130704



# Government Engineering College

## Sector-28 Gandhinagar

### Certificate

This is to certify that

Mr./Ms. ...kasadiya.....khushi.....P..... Of class

....A.... Division ....2.., Enrollment No. 2103010704..... Has  
satisfactorily completed his/her term work in  
....Digital.....Fundamental..... Subject for the term  
ending in ....Jan....2022-23.

Date:- 6/11/2023

Head of Department

Signature of Teacher

# Contents

<b>1. Institute Vision/Mission .....</b>	<b>4</b>
Vision: .....	4
Mission:.....	4
<b>2. Computer Engineering Department Vision/Mission .....</b>	<b>5</b>
Vision: .....	5
Mission:.....	5
<b>3. Program Educational Outcome (PEO) .....</b>	<b>6</b>
<b>4. PSO.....</b>	<b>6</b>
<b>5. POs .....</b>	<b>6</b>
<b>6. Submission Instructions:.....</b>	<b>Error! Bookmark not defined.</b>
<b>7. Assignment Index.....</b>	<b>8</b>
<b>8. Practical Index .....</b>	<b>9</b>
<b>9. Assignment 1 .....</b>	<b>10</b>
<b>10. Assignment 2 .....</b>	<b>11</b>
<b>11. Assignment 3 .....</b>	<b>12</b>
<b>12. Assignment 4 .....</b>	<b>13</b>
<b>13. Assignment 5 .....</b>	<b>14</b>
<b>14. Practical 1 .....</b>	<b>15</b>
<b>15. Practical 2 .....</b>	<b>16</b>
<b>16. Practical 3 .....</b>	<b>24</b>
<b>17. Practical 4 .....</b>	<b>28</b>
<b>18. Practical 5 .....</b>	<b>32</b>
<b>19. Practical 6 .....</b>	<b>36</b>
<b>20. Practical 7 .....</b>	<b>39</b>
<b>21. Practical 8 .....</b>	<b>41</b>
<b>22. Practical 9 .....</b>	<b>45</b>
<b>23. Practical 10 .....</b>	<b>49</b>
<b>24. Practical 11 .....</b>	<b>55</b>

# **Computer Engineering Department**

## **Vision/Mission**

### **Vision:**

- To achieve excellence for providing value based education in computer engineering through innovation, teamwork and ethical practices.

### **Mission:**

- To produce computer science and engineering graduates according to the needs of industry, government, society and scientific community.
- To develop partnership with industries, government agencies and R & D organization.
- To motivate student /graduates to be entrepreneurs.
- To motivate students to participate in reputed conferences, workshops, symposiums, seminar and related technical activities.

# **Program Educational Outcome (PEO)**

- To provide students with a strong foundation in the mathematical, scientific and engineering fundamentals necessary to formulate, solve and analyze engineering problems and to prepare them for graduate studies, R&D, consultancy and higher learning.
- To develop an ability to analyze the requirements of the software, understand the technical specifications, design and provide novel engineering solutions and efficient product designs.
- To provide exposure to emerging cutting edge technologies, adequate training & opportunities to work as teams on multidisciplinary projects with effective communication skills and leadership qualities.
- To prepare the students for a successful career and work with values & social concern bridging the digital divide and meeting the requirements of Indian and multinational companies.
- To promote student awareness on the life-long learning and to introduce them to professional ethics and codes of professional practice

## **PSO**

By the completion of Computer Engineering program the student will have following Program specific outcomes.

- Design, develop, test and evaluate computer-based systems by applying standard software engineering practices and strategies in the area of algorithms, web design, data structure, and computer network
- Apply knowledge of ethical principles required to work in a team as well as to lead a team

## **POs**

**Engineering Graduates will be able to:**

1. **Engineering knowledge:** Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.
2. **Problem analysis:** Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of

mathematics, natural sciences, and engineering sciences.

3. **Design/development of solutions:** Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.
4. **Conduct investigations of complex problems:** Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.
5. **Modern tool usage:** Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modelling to complex engineering activities with an understanding of the limitations.
6. **The engineer and society:** Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.
7. **Environment and sustainability:** Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.
8. **Ethics:** Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.
9. **Individual and team work:** Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.
10. **Communication:** Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.
11. **Project management and finance:** Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.
12. **Life-long learning:** Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

# Digital Fundamental (3130704)

## Course Outcomes (COs)

CO-1	Solve the given Problem using fundamentals of number system and Boolean algebra.
CO-2	Analyze working of logic families and logic gates and design simple circuits using various gates for a given problem
CO-3	Design and implement combinations and sequential logic circuits and verify its working
CO-4	Examine the process of Analog to Digital conversion and Digital to Analog conversion
CO-5	Implement PLDS for the given logical problem

## 7. Assignment Index

Sr. No	Assignment	Date	Page No.	Sign
1	Assignment 1	30/09/22		
2	Assignment 2	17/10/22		
3	Assignment 3	30/11/22		
4	Assignment 4	14/12/22		
5	Assignment 5			

## 7. Assignment Index

Sr. No	Assignment	Date	Page No.	Sign
1	Assignment 1	30/09	1	<del>10/10/10</del>
2	Assignment 2	17/10	8	<del>10/10/10</del>
3	Assignment 3	30/11	11	<del>10/10/10</del>
4	Assignment 4	14/12	14	<del>10/10/10</del>
5	Assignment 5	30/12	15	<del>10/10/10</del>

# ASSIGNMENT : 01

Page No.:

Date:



Module: 1

A) De Morgan suggested two theorems that form an important part of Boolean algebra.

$$1. \overline{AB} = \bar{A} + \bar{B}$$

The complement of a product is equal to sum of the complements.

Truth table:

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

$$2. \overline{A+B} = \bar{A} \cdot \bar{B}$$

The complement of a sum is equal to the product of the complements.

Truth table:

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

$$A) F = \overline{ABC} + \overline{BC} + A\bar{B}$$

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

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

$$[\because B + \bar{B} = 1]$$

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

Q.3

A.3 Logic families are divided in two parts

i) Bipolar

a) saturated:

i) Register Transistor Logic (RTL)

ii) Diode Transistor Logic (DTL)

iii) Direct coupled Transistor (DCTL)

iv) Integrated Injection Logic ( $I^2L$ )

v) High threshold logic (HTL)

vi) Transistor Transistor Logic (TTL)

b) unsaturated:

i) Schottky TTL

ii) Emitter Coupled Logic (ECL)

ii) unipolar

a) P-channel MOSFET (PMOS)

b) N-channel MOSFET (NMOS)

c) complementary MOSFET (CMOS)

It is a group of compatible ICs with the logic levels and supply voltages for performing logic functions. They are fabricated using specific circuit configuration which is as a logic family. The circuit design of gate of each logic family is the same.

Transistor - Transistor Logic (TTL):

It is more dependency on transistor alone to perform basic logic operations.

Schottky TTL:

A Schottky transistor is a combination of a transistor and a Schottky diode that prevents the transistor from saturating by diverting the excessive input current. It is also called Schottky transistor.

CMOS: EO

CMOS: EO  
Complementary metal-oxide-semiconductor is a type of metal oxide semiconductor field-effect transistor fabrication process that uses complementary and symmetrical pairs of p-type and n-type MOSFET's for logic function.

4. Describe error detecting and correcting code.

→ To maintain the data integrity between transmitter & receiver, extra bit or more than one bit are added in the data. These extra bits allow the detection & sometimes correction of errors in the data. The data along with the extra bit forms the code. Codes allowing only error detection are called error detecting codes & codes which allow error detection and correction are called error correcting code.

→ Parity code:

Even parity code: The value of even bit should be zero, if even number of present in the binary code. otherwise, it be one. so that, even number of present in even parity code.

Binary code	Even parity bit	Even P
000	0	0
001	1	00
010	1	01
011	0	01
100	1	10
101	0	10
110	0	11
111	1	11

Odd parity code: The value of odd parity should be zero, if odd number of ones present in the binary code. otherwise, should be one. number of ones present in odd parity

Binary code	odd parity bit	odd P
000	1	0001
001	0	0010
010	0	0100
011	1	0111
100	0	1000
101	1	1011
110	1	1101
111	0	1110

→ Hamming code :-

Hamming code is useful for both detection and correction of errors present in the received data. This code uses multiple parity bits and we have to place these parity bits in the position of power of 2.

Bit designation

D<sub>7</sub> D<sub>6</sub> D<sub>5</sub> P<sub>4</sub> D<sub>3</sub> P<sub>2</sub> P<sub>1</sub>

Bit Location

7 6 5 4 3 2 1

Binary location num

111 110 101 100 101 010 001

→ Parity P<sub>7</sub> checks bit locations 1, 3, 5, 7 and assign P<sub>7</sub> according to even or odd parity.

→ Parity P<sub>2</sub> checks bit locations 2, 3, 6, 7 and assign P<sub>2</sub> according to even or odd parity.

→ Parity P<sub>1</sub> checks bit locations 4, 5, 6, 7 and assign P<sub>1</sub> according to even or odd parity.

Q.5  
A.5

Differentiate TTL, schottky TTL, and

Parameter	CMOS	TTL	Schottky TTL
Device used	N-channel & P-channel MOSFET	Bipolar junction transistor	Schottky diode
$V_{TH}(\text{min})$	3.5V	2.4V	1.2V
$V_{TH}(\text{max})$	1.8V	0.8V	0.4V
$V_{OH}(\text{min})$	4.95V	2.7V	2.7V
$V_{OL}(\text{max})$	0.005V	0.4V	0.5V
Power dissipation per state	0.1 mW	10 mW	1 mW
Fan out	50	10	10
Application	Portable instrument where battery supply is used	Laboratory instrument	Voltage clamping application prevent saturation

ref

## ASSIGNMENTS : 2

## MODULE : 2

The map method gives us a systematic approach for simplifying a Boolean expression.

The basis of this method is a graphical chart known as Karnaugh map (K-map)

It contains boxes called cells. Each of the cell represents one of the  $2^n$  possible products that can be formed from  $n$  variables. Thus a 2-variable map contains  $2^2 = 4$  cells, a 3 variable map contains  $2^3 = 8$  cells so on.

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

(1-variable map)      (2-variable map)      (3-variable map)      (4-variable map)  
 (2 cells)      (4 cells)      (8 cells)      (16 cells)

A.  $F(x, y, z) = \Sigma(2, 3, 5, 7)$

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

$F = y$

$$2.2 F(A, B, C, D) = \Sigma (4, 6, 7, 15)$$

AB\CD	00	01	101	10
00	0	1	3	2
01	1	4	5	6
11	12	13	15	14
10	8	9	11	10

$$F = BCD + \bar{A}BD$$

Q3

### A3. Adder:

- The logic families circuit which performs addition of two bits (sum and carry) is adder.
- The logic circuit which performs addition three bits (two significant bits and a carry) is a full adder.

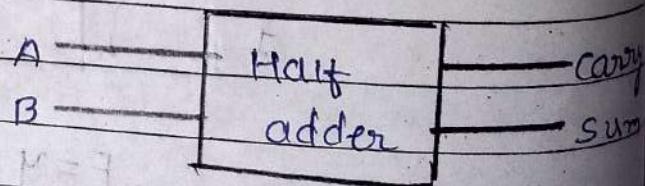
### Half Adder:

- Two binary inputs: augend and addend
- Two binary outputs: sum and carry.

Truth table:

A	B	carry	sum
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

Block diagram:



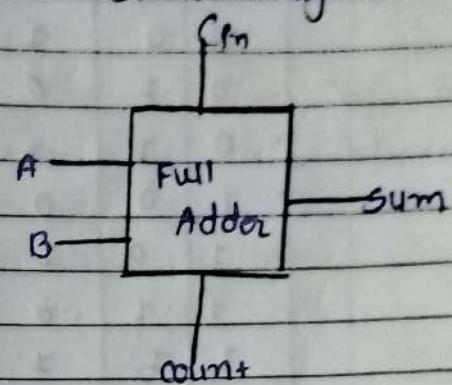
Full Adders:

- There binary inputs, (augend, addend and previous carry).
- Two binary outputs; sum and carry.

Truth table

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

Block diagram



Subtractors:

- A half subtractor is a combinational circuit that subtracts two bits and produces their difference.
- A full Subtractor is a combinational circuit that subtracts between two bits, taking into account borrows of lower stage.

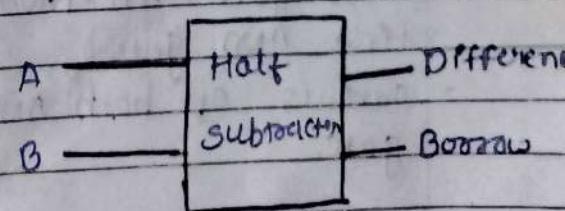
Half subtractor:

- Two inputs and two outputs which is difference & borrow.

Truth table:

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

Block diagram:

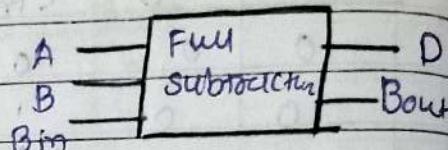


### Full Subtractor:

- Three inputs: minuend, subtrahend and borrow in.
- Two outputs: Difference and borrow out.

Truth table:

A	B	Bin	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



Q4

### A4. Multiplexer:

To select single data line from several data lines, and the data from the selected data line should be available on the output. The circuit which does this task is a multiplexer.

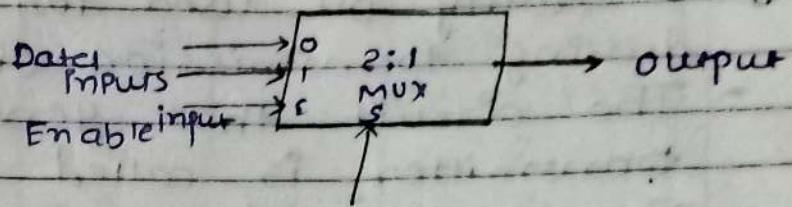
#### 2:1 multiplexer:

- D0 is applied as an input to one AND gate
- D1 is applied as an input to another AND gate
- Enable input is applied to both gates as one input.
- Selection lines is connected as second input to AND gate. An inverted S is to second input first AND gate
- outputs of both AND gate applied as inputs to OR gate

Function table

E	S	Y
1	0	P <sub>0</sub>
1	1	P <sub>1</sub>
0	X	0

Block diagram



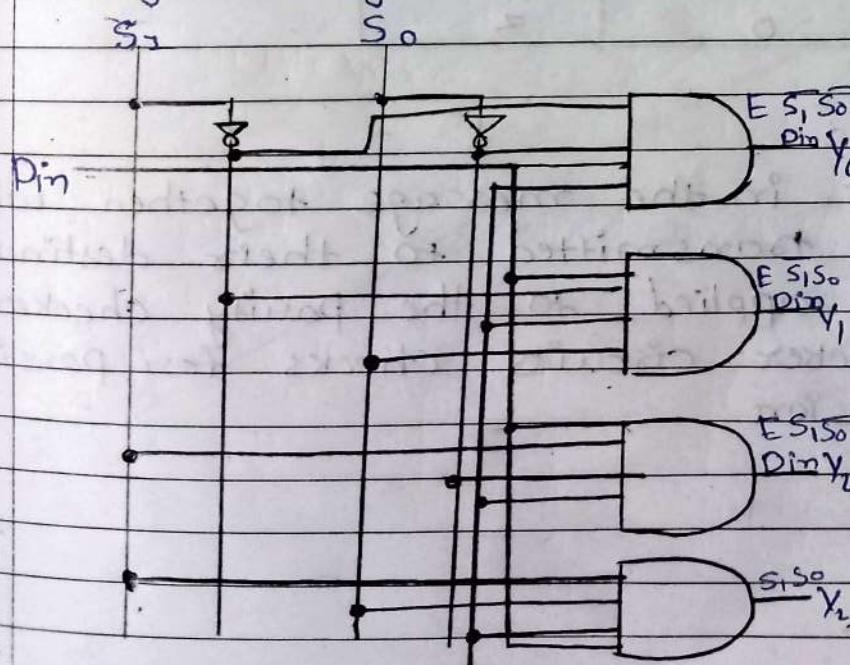
### Demultiplexers:

A demultiplexer is a circuit that receives information on a single line and transmits this information on one of  $2^n$  possible output lines. The selection of specific output line is controlled by the values of n selection lines.

### 1:4 Demultiplexer:

The single input variable D<sub>in</sub> has a path to all four outputs, but the input information is directed to only one of the output lines depending on the select inputs. Enable input should be high to enable demultiplexer.

Logic diagram:



Function Table:

E	S <sub>1</sub>	S <sub>0</sub>	D <sub>in</sub>	Y <sub>0</sub>	Y <sub>1</sub>	Y <sub>2</sub>	Y <sub>3</sub>
0	X	X	X	0	0	0	0
1	0	0	0	0	0	0	0
1	0	0	1	1	0	0	0
1	0	1	0	0	0	0	0
1	1	0	1	1	0	1	0
1	1	0	0	0	0	0	0
1	1	1	0	0	0	0	0
1	1	1	1	0	0	0	1

Q5

- As - A parity bit used for the purpose of detection during transmission of binary information.
- The circuit that generates the parity bit in transmitter is called a parity generator or circuit that checks parity in the receiver is called parity checker.
- \* Parity Generator truth Table for even and parity.

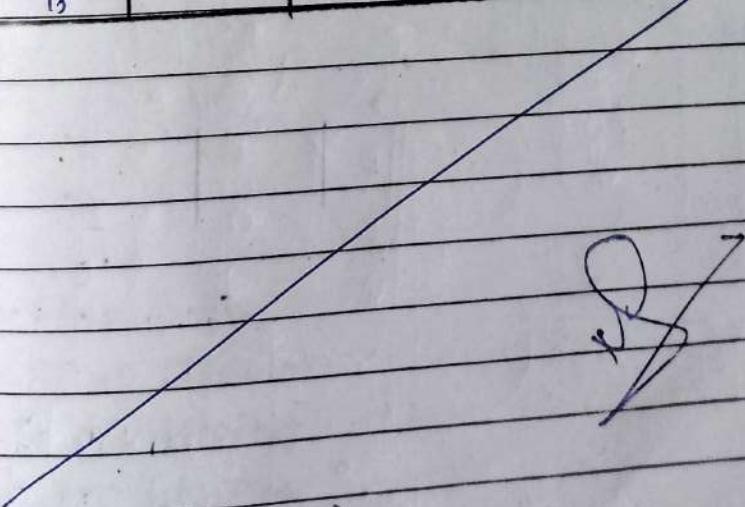
A	B	C	odd parity bit	even parity bit
0	0	0	1	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	0	1

### Parity Checker:

- The three bits in the message together with parity bit are transmitted to their destination where they are applied to the parity checker. The parity checker circuits checks for errors in the transmission.

\* Truth table for even parity checks.

Decimal equivalent	P	A	B	C	Parity error check (PEC)
0	0	0	0	0	0
1	0	0	0	1	1
2	0	0	1	0	1
3	0	0	1	1	0
4	0	1	0	0	1
5	0	1	0	1	0
6	0	1	1	0	0
7	0	1	1	1	1
8	1	0	0	0	1
9	1	0	0	1	0
10	1	0	1	0	0
11	1	0	1	01	1
12	1	1	0	0	0
13	1	1	0	1	1
14	1	1	1	0	1
15	1	1	1	1	0


 A hand-drawn arrow starts near the bottom-left corner of the page and points upwards and to the right, passing over the truth table.

# Assignment : 3

## Module : 3

### Combinational circuit

### Sequential circuit

1. In combinational circuit, the output variables are at all times dependent on the combination of input variables.	1. In sequential circuit, the output variables depend not only on the present input variables but they also depend upon the past history of these input variables
2. Easy to design	2. Harder to design
3. parallel adder is a combinational circuit	3. serial adder is a sequential circuit.
4. Faster in speed	4. slower than combinational circuits

### A2. Flip Flop are:

- 1] JK flipflop
- 2] D flipflop
- 3] T flipflop
- 4] SR flipflop

#### i) JK flipflop:

The data inputs are J and K which are AND with  $\bar{Q}$  and  $\hat{Q}$  respectively, to obtain S and R.

Truth Table:

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

### 2) D flip flop:

→ Input conditions can be avoided by making complement of each other. This modified flip flop is known as D flip flop.

Truth Table:

CP	D	$Q_{n+1}$
↑	0	0
↑	1	1
0	X	$Q_n$

### 3) T Flip-flop:

- T flip flop is also known as "Toggle flip-flop".
- Modification of JK flip flop.

Truth Table:

T	$Q_{n+1}$
0	$Q_n$
1	$\bar{Q}_n$

### 4) SR Flip-Flop:

- The circuit is similar to SR latch. enable signal is replaced by the clock (CP) followed by the positive edge detector.
- The edge detection circuit is a differ-

## Truth Table:

Cp	S	R	Q <sub>n+1</sub>
0	X	X	Q <sub>n</sub>
↑	0	0.	Q <sub>n</sub>
↑	0	1	Hold
↑	1	0	0.
↑	1	1	Invalid

- 3  
 i) Buffer register  
 ii) Controlled Buffer register  
 iii) Shift Register.

### i) Buffer Register:

- Constructed using four D-flipflops. This register is called buffer register.
- Each D flipflop is triggered with a common negative edge clock pulse.
- The input bits set up the flipflop for loading.

### ii) Controlled Buffer Register:

- we can control input and output of the register by connecting tri-state devices at the input and output sides of register. So this register is called controlled Buffer register.
- Tri-state switches are used to control the operation.

### iii) Shift Register:

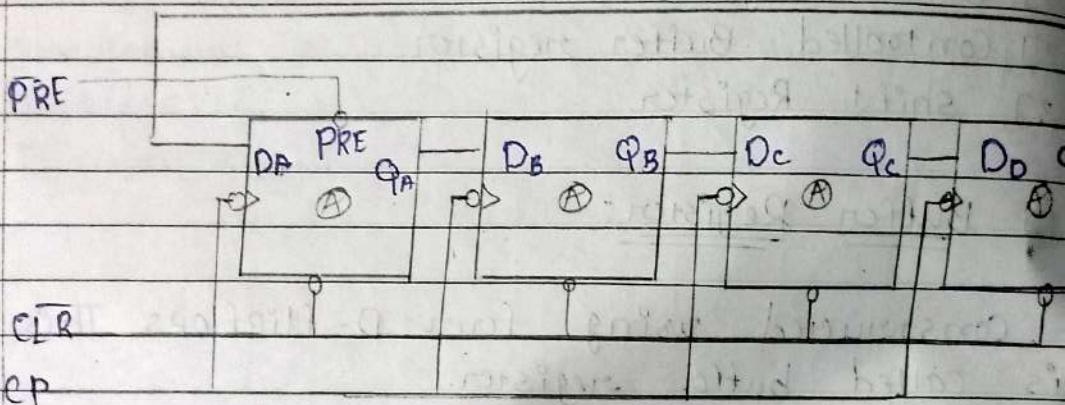
- The binary information in a register can be moved from stage to stage within the

register or into or out of the register application of clock pulses.

- This type of bit movement or shift essential for certain arithmetic and operations used in microprocessors

Q4.

A4. Ring Counter:



- The  $Q$  output of each stage is connected to the input of the next stage and the last stage is fed back to the input of the first stage.
- The CLR followed by PRE makes the output of the first stage to '1' and remaining output i.e.  $Q_A$  is one and  $Q_B$ ,  $Q_C$ ,  $Q_D$  are zero.
- The first clock pulse produces  $Q_B=1$  and remaining output are zero.
- The first clock pulse produces  $Q_B=1$  and remaining output are zero.
- According to the clock pulse applied at the inputs CP, a sequence of four states is

Steps involved in the design of asynchronous counter

- i] Determine the number of flip-flops needed.
- ii] choose the types of flip flops to be used: T or JK. If T flip flop are used connect both J and K inputs of all flip flop are used connected both. toggles the flip flop output on each clock transition.
- iii] write the truth table for the counter.
- iv] derive the reset logic by K-map Simplification
- v] Draw the logic diagram

2/2

## Assignment : 4

### Module : 4

weighted Register converter: A weighted resistor DAC produces an analog output, which is almost equal to the digital (binary) input by using binary weighted resistors in the inverting adder circuit. In short, a binary weighted resistor DAC is called as weighted resistor DAC.

→ The binary weighted resistor DAC uses an op-amp to sum  $n$  binary weighted currents derived from a reference voltage  $V_R$  via current scaling resistor  $2R, 4R, 8R, 2^n R$ .

→ For ON switch;  $I = \frac{V_R}{R}$  and

For OFF switch;  $I = 0$

→ Due to high input impedance of op-amp,

$$I_T = I_1 + I_2 + I_3 + \dots + I_n$$

when  $R_f = R$ ,  $V_o$  is given as

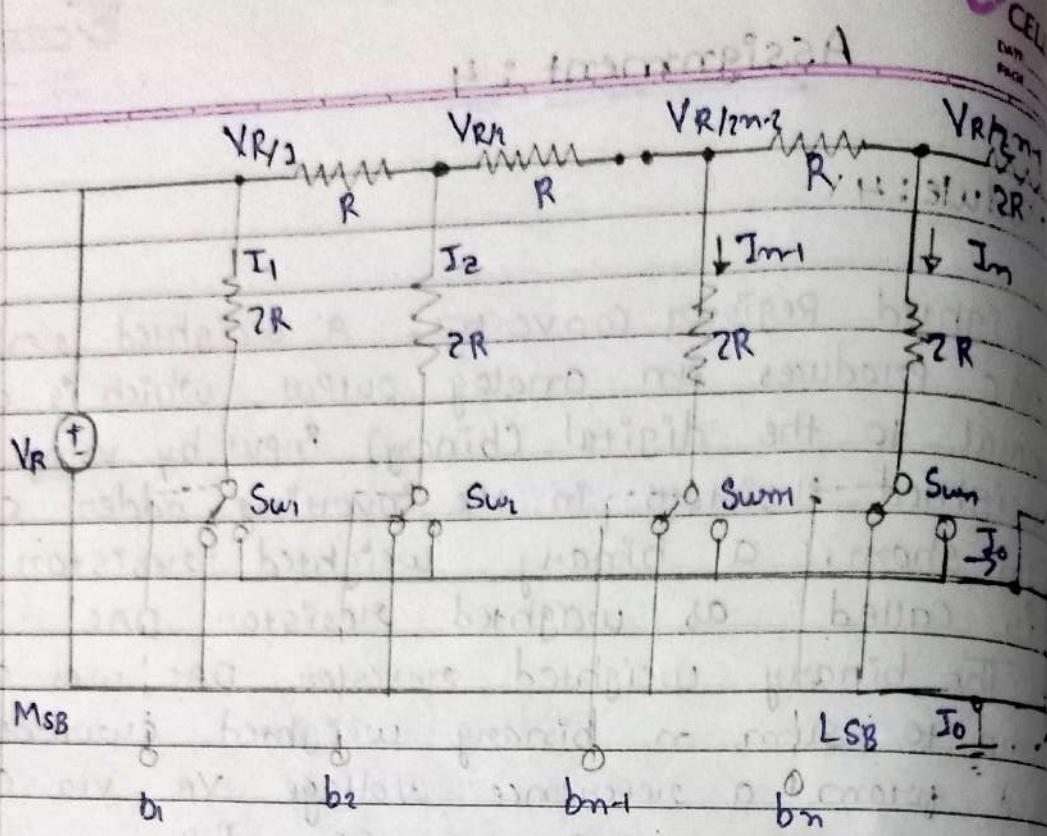
$$V_o = -V_R (b_1 2^1 + b_2 2^2 + b_3 2^3 + \dots + b_n 2^n)$$

1. R/2R ladder D/A converter uses only two resistor values. This avoids resistance spread drawback of binary weighted D/A converter.

- Easier to build accurately as only two precision metal film resistors are required.

- Number of bits can be expanded by adding more sections of same R/2R values.

- In inverted R/2R ladder DAC, node voltage remain constant with changing input binary words. This avoids any slowdown effects by stay capacitors.



Q3

A3 Specification of DA converter:

1] Resolution:

→ Smallest change that occurs in an analog output as a result of a change in the input.

→ % resolution = Step size / full Scale  $\times 100$

→ Full Scale = No. of steps  $\times$  Step size

% resolution = 1 / No. of steps  $\times 100$

2] Accuracy:

- Specified in terms of full-scale and linearity error.

3] Setting time:

- The time required for the analog output to settle to within  $\pm \frac{1}{2}$  LSB of the final value after a change in the digital input.

### 4) Monotonicity :

→ This means that the stair case output will have no downward steps as the binary input is incremented from 0 to full scale value.

### 5) Temperature Sensitivity :

→ The analog output voltage for any fixed digital input varies with temperature.

## Specifications of A/D converter:

### 1) Range of input Voltage

### 2) Input impedance

### 3) Accuracy

### 4) Conversion time

### 5) format of digital output.

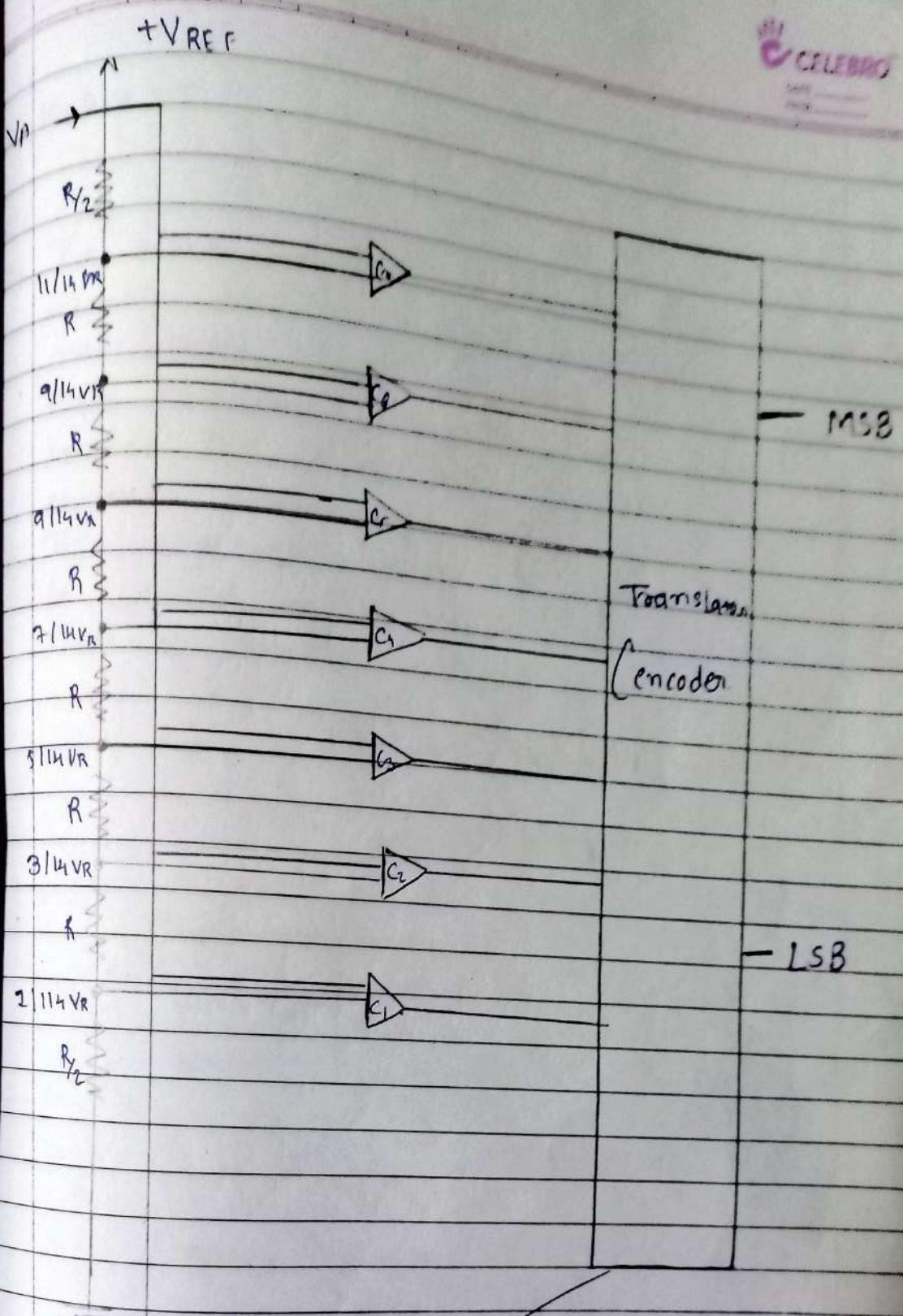
**A4**  
**Quantizing / Encoding :** The process of mapping the sampled analog voltage values to discrete voltage levels, which are then represented by binary numbers. This is needed because the analog sample values are real numbers that occur on a continuum.

$$q = \frac{V_{max} - V_{min}}{2^n} = \frac{1 - (-1)}{4} = 0.25V$$

The value of  $q$  is more formally called the quantizer's resolution.

This circuit is formed of a series of comparators, each one comparing the input signal to a unique reference voltage.

- Based on the principle of comparing input voltage with a set of reference voltages.
- To convert the analog input voltage into digital signal of  $n$ -bit output,  $(2^n - 1)$  comparators are required.
- It is the fastest type of ADC because the conversion is performed simultaneously through a set of comparators; hence, referred as flash type.
- Construction is simple and easier to design.



[ 3-bit flash | Parallel Comparison ]

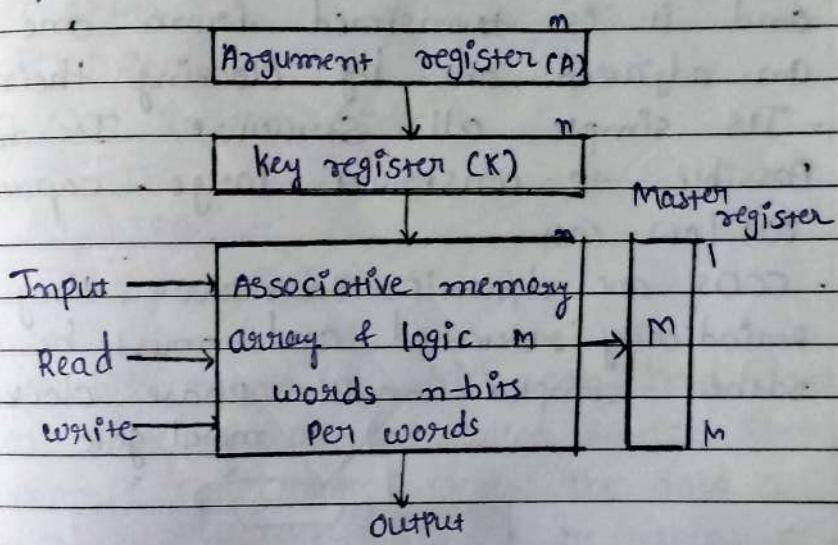
SD

## Assignment : 5

Module : 5

The time required to find an object stored in memory can be reduced considerably if objects are selected based on their contents, not on their locations. A memory unit accessed by the content is called an associative memory or content Addressable memory (CAM).

This type of memory is accessed simultaneously in parallel on the basis of data content rather than by specific address or location.

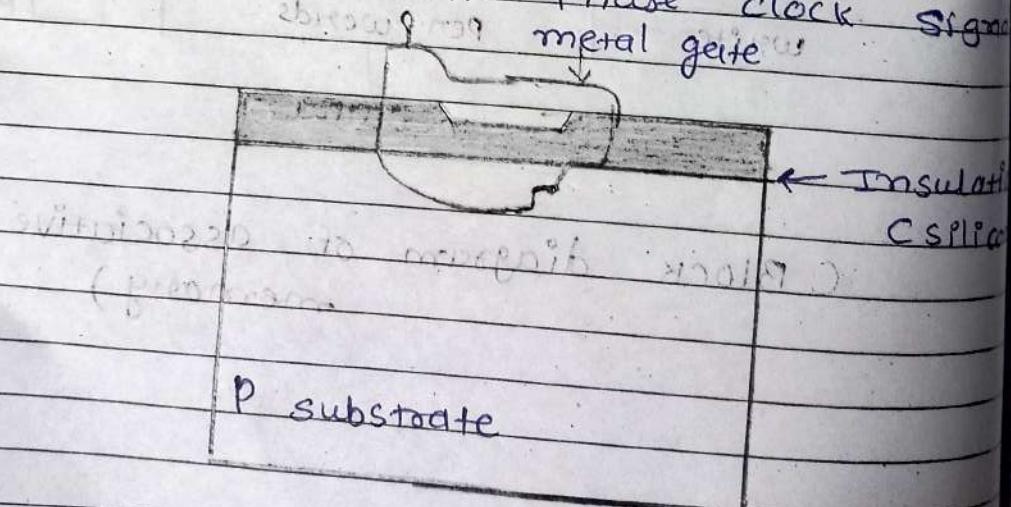


( Block diagram of associative memory )

Q2

A2 Charge-coupled device (CCD) memory is a dynamic memory in which packets of data are continuously transferred from one place to another.

- The structure of a MOS charge coupled device is quite simple as shown in fig.
- When a high voltage is applied to the gate, holes are repelled from a region below the gate in the p-type substrate. This region, called a potential well, is then capable of accepting a packet of negative charged particles. Therefore, data is stored in a CCD and it is transferred from one device to an adjacent one by clocking their gates.
- Its simple cell structure makes it possible to construct large-capacity memories at low cost.
- CCDs are dynamic in nature, they must be periodically refreshed and must be driven rather complex multi-phase clock signals.



## Non volatile memory

- i) Read only memory (ROM)
  - i) Mask Programmable ROM
  - ii) Programmable ROM.
- ii) Read / write memory (NVRAM)
  - i) EEPROM
  - ii) EEPROM
  - iii) FLASH

## Volatile memory

- i) Read / write Memory (RAM)
  - a) Random Access
    - i) SRAM
    - ii) DRAM
  - b) Non Random Access
    - i) FIFO
    - ii) LIFO
    - iii) Shift Register.

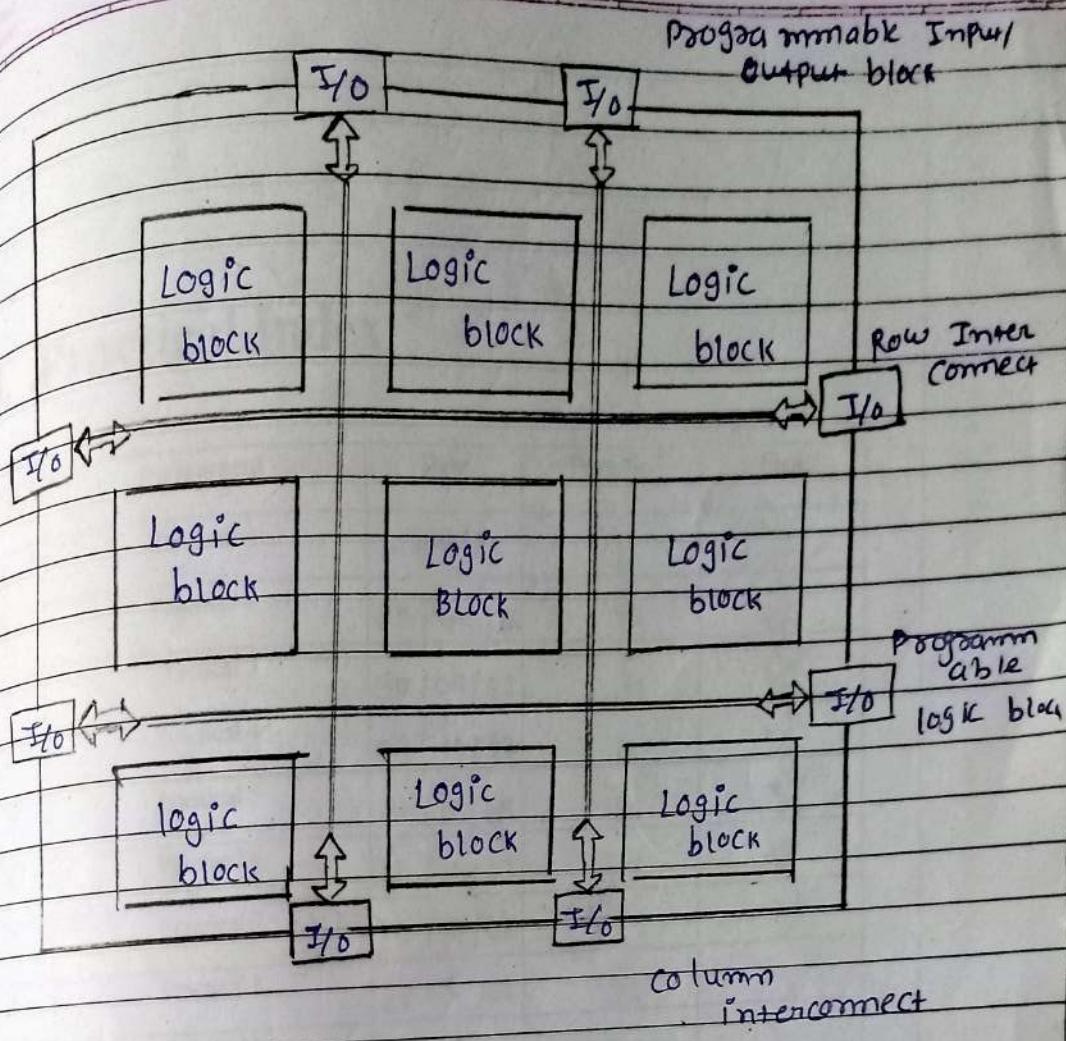
- The volatile memories which can hold data as long as power is on are called static RAMS (SRAMs)
- The dynamic RAM (DRAMs) stores the data as a charge on the capacitor and they need refreshing of charge on the capacitor after every few milliseconds to hold the data even if power is on.
- EEPROM and EEPROM are erasable memories in which the stored data can be erased and new data can be stored.

Q4.

- A4. A Semiconductor is a substance that has some electrical properties that enable it to serve a foundation for computers and other electronic devices.
- It is typically a solid chemical element that conducts electricity under certain conditions but not others.
  - Semiconductors are materials which have conductivity between conductors (generally metals) and non-conductors or insulators.
  - Semiconductors can be pure elements, such as silicon or germanium or compounds such as gallium arsenide or cadmium selenide.

Q5

- A5 Field programmable gate array (FPGA) is the next generation in the programmable logic devices.
- The word field in the name refers to the ability of the gate arrays to be programmed by a specific function by the user instead of the manufacturer of the device.
  - The word array is used to indicate a series of columns and rows of that can be programmed by the user.
  - The programmable logic blocks in FPGA are called logic blocks or Configurable Cells (CLBs).



C Basic architecture of FPGAs

2/2

## 8. Practical Index

Sr. No	Assignment	Date	Page No.	Sign
1	Practical 1	16/09/22		
2	Practical 2	23/09/22	1	
3	Practical 3	30/09/22	8	
4	Practical 4	07/10/22	11	
5	Practical 5	14/10/22	14	
6	Practical 6	11/11/22	18	
7	Practical 7	18/11/22	19	
8	Practical 8	25/11/22	23	
9	Practical 9	02/12/22	24	
10	Practical 10	09/12/22	27	
11	Practical 11	16/12/22	31	

# 14. Practical 1

CO2: Analyze working of logic families and logic gates and design the simple circuits using various gates for a given problem.

## Module 1

Aim: Getting familiar with Logisim, Study and implement all basic logic gates. Implement NAND and NOR logic gates as universal gates.

### THEORY:

Logic gates are devices that can combine multiple inputs at independent logic levels and come up with an output accordingly. There are many kinds of logic gates, and the distinction lies in that each kind processes the inputs differently, and may give different outputs for the same inputs.

The way the logic gate processes different inputs is given in a truth table for that gate, which lists all the possible combinations of inputs next to their outputs. An example is given for a simple one-input gate with the function of giving the opposite logic level at the output to the input. The inputs are given on the left, and the outputs are on the right. Generally, the inputs are called A, B, C, etc., and the output is labeled Q. In this case, there are only two possible inputs, 1 or 0, but logic gates can have any number of inputs.

### THE LOGIC AND FUNCTION:

- The AND gate is an electronic circuit that gives a true output (1) only if all its inputs are true. A dot (.) is used to show the AND operation i.e. A·B.
- Note that the dot is sometimes omitted i.e. AB

### THE LOGIC OR FUNCTION:

- The OR gate is an electronic circuit that gives a true output (1) if one or more of its inputs are true. A plus (+) is used to show the OR operation.

### THE LOGIC NOT FUNCTION:

- The Logic NOT Function is simply a single input inverter that changes the input of a logic level "1" to an output of logic level "0" and vice versa.

### THE LOGIC EXCLUSIVE-OR FUNCTION:

- The 'Exclusive-OR' gate is a circuit which will give a true output if either, but not both, of its two inputs are true.
- An encircled plus sign ( $\oplus$ ) is used to show the E-XOR.

### **THE LOGIC EXCLUSIVE-NOR FUNCTION:**

- The 'Exclusive-NOR' gate circuit does the opposite to the EXOR gate. It will give a false output if either, but not both, of its two inputs are true.
- The symbol is an EXOR gate with a small circle on the output. The small circle represents inversion.

### **THE LOGIC NAND FUNCTION:**

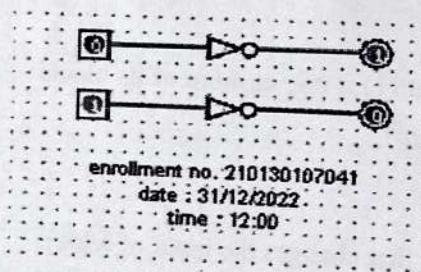
- This is a NOT-AND gate which is equal to an AND gate followed by a NOT gate. It is a UNIVERSAL gate.
- The outputs of all NAND gates are true if any of the inputs are false. The symbol is an AND gate with a small circle on the output. The small circle represents inversion.

### **THE LOGIC NOR FUNCTION:**

- This is a NOT-OR gate which is equal to an OR gate followed by a NOT gate. It is a UNIVERSAL gate.
- The outputs of all NOR gates are false if any of the inputs are true. The symbol is an OR gate with a small circle on the output. The small circle represents inversion.

#### **Snapshots:**

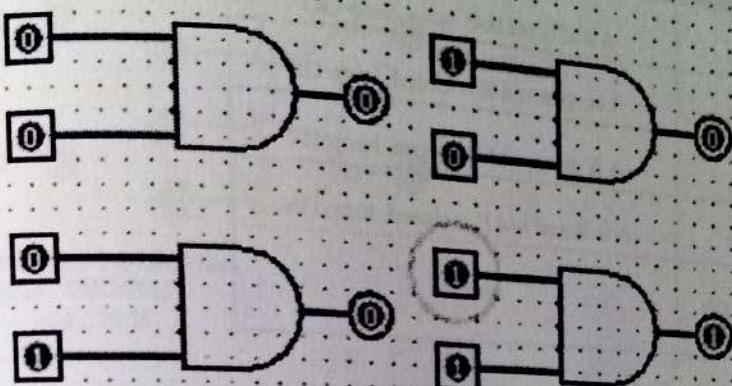
##### **1. NOT GATE**



Switch	Output
1	0
0	1
Boolean Expression	not-A or A

## 2. AND GATE

AND logic circuit



Name: kasdiya khushi

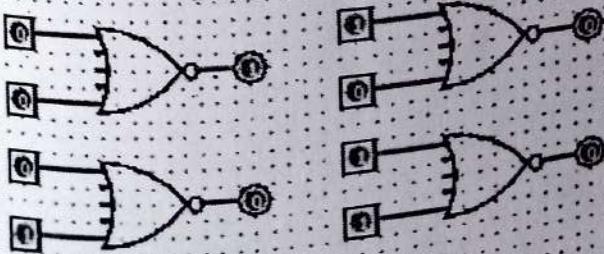
Enroll.No: 210130107041

Time: 10:42 2/1/2031

Switch A	Switch B	Output	Description
0	0	0	A and B are both open, lamp OFF
0	1	0	A is open and B is closed, lamp OFF
1	0	0	A is closed and B is open, lamp OFF
1	1	1	A is closed and B is closed, lamp ON
Boolean Expression (A AND B)			A . B

## 3. OR GATE

NOR function circuit

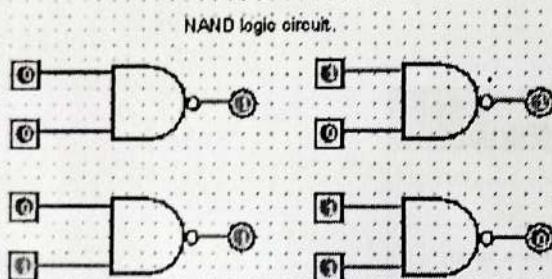


enrollment no. 210130107041  
date : 31/12/2022  
time : 10:15

Switch A	Switch B	Output	Description
0	0	0	A and B are both open, lamp OFF
0	1	1	A is open and B is closed, lamp ON
1	0	1	A is closed and B is open, lamp ON
1	1	1	A is closed and B is closed, lamp ON
Boolean Expression (A OR B)		$A + B$	

## Universal Gates:

### 1. NAND GATE



enrollment no. 210130107040

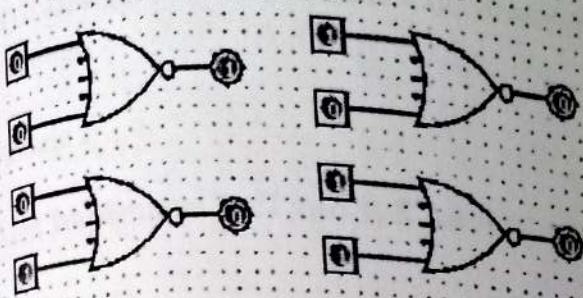
Date : 31/12/2022

time.: 16:46

Truth Table		
B	A	Q
0	0	1
0	1	1
1	0	1
1	1	0

# NOR GATE:

NOR function circuit.

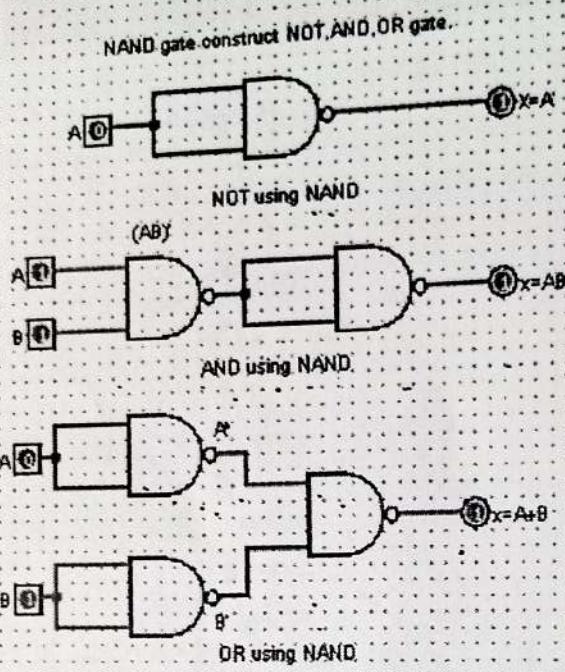


enrollment no. 210130107041  
date : 31/12/2022  
time : 10:15

Truth Table

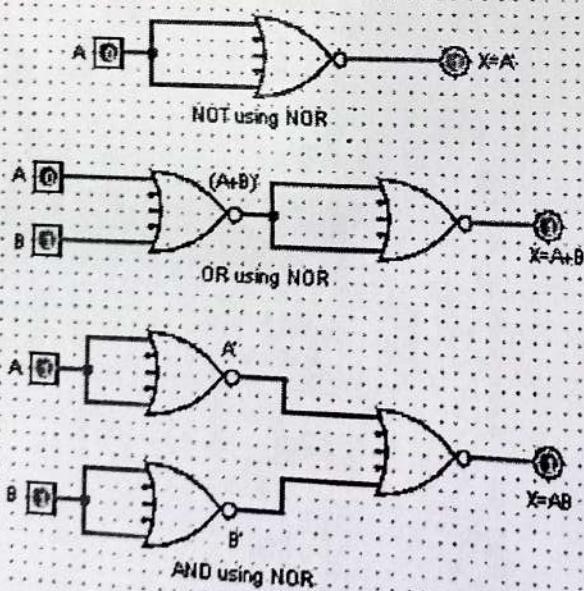
B	A	Q
0	0	1
0	1	0
1	0	0
1	1	0

## NAND gate as Universal gate



Enrollment no. 210130107041  
Date : 31/12/2022  
time : 8:07

## NOR as Universal gates



Enrollment no. 210130107041  
Date : 31/12/2022  
Time : 8:00

# 15.Practical 2

CO3: Design and implement Combinational and Sequential logic circuits and verify its working.

## Module 2

Aim: Implement half and full Adders using logic gates.

### Theory:

An adder is circuiting electronics that implements addition of numbers. In many computers and other types of processors, adders are used to calculate addresses, similar operations and table indices in the ALU and also in other parts of the processors. These can be built for many numerical representations like excess-3 or binary coded decimal. Adders are classified into two types: half adder and full adder. The half adder circuit has two inputs: A and B, which add two input digits and generate a carry and sum. The full adder circuit has three inputs: A and C, which add the three input numbers and generate a carry and sum. An adder is a digital circuit that performs addition of numbers. The half adder adds two binary digits called as augends and addend and produces two outputs as sum and carry; XOR is applied to both inputs to produce sum and AND gates applied to both inputs to produce carry. The full adder adds 3 one-bit numbers, where two can be referred to as operands and one can be referred to as bit carried in. And produces 2-bit output, and these can be referred to as output carry and sum.

### Half Adder:

Half adder is a combinational arithmetic circuit that adds two numbers and produces a sum bit (S) and carry bit (C) as the output. If A and B are the input bits, then sum bit (S) is the X-OR of A and B and the carry bit

(C) will be the AND of A and B. From this it is clear that a half adder circuit can be easily constructed using one X-OR gate and one AND gate.

Half adder is the simplest of all adder circuit, but it has a major disadvantage. The half adder can add only two input bits (A and B) and has nothing to do with the carry if there is any in the input. So, if the input to a half adder have a carry, then it will be neglected it and adds only the A and B bits. That means the binary addition process is not complete and that's why it is called a half adder.

Boolean Expression: Difference =  $X'Y + XY' = X \oplus Y$   
Borrow C =  $X' \cdot Y$

**Full Adder:**  
 The full adder is a little more difficult to implement than a half adder. The main difference between a half adder and a full adder is that the full adder has three inputs and two outputs. The two inputs are A and B, and the third input is a carry input CIN. The output carry is designated as COUT, and the normal output is designated as S. The output S is an EX-OR between the input A and the half adder SUM output B. The COUT will be true only if any of the two inputs out of the three are HIGH or at logic 1.

Thus, a full adder circuit can be implemented with the help of two half adder circuits. The first half adder circuit will be used to add A and B to produce a partial sum. The second half adder logic can be used to add CIN to the sum produced by the first half adder circuit. Finally, the output S is obtained.

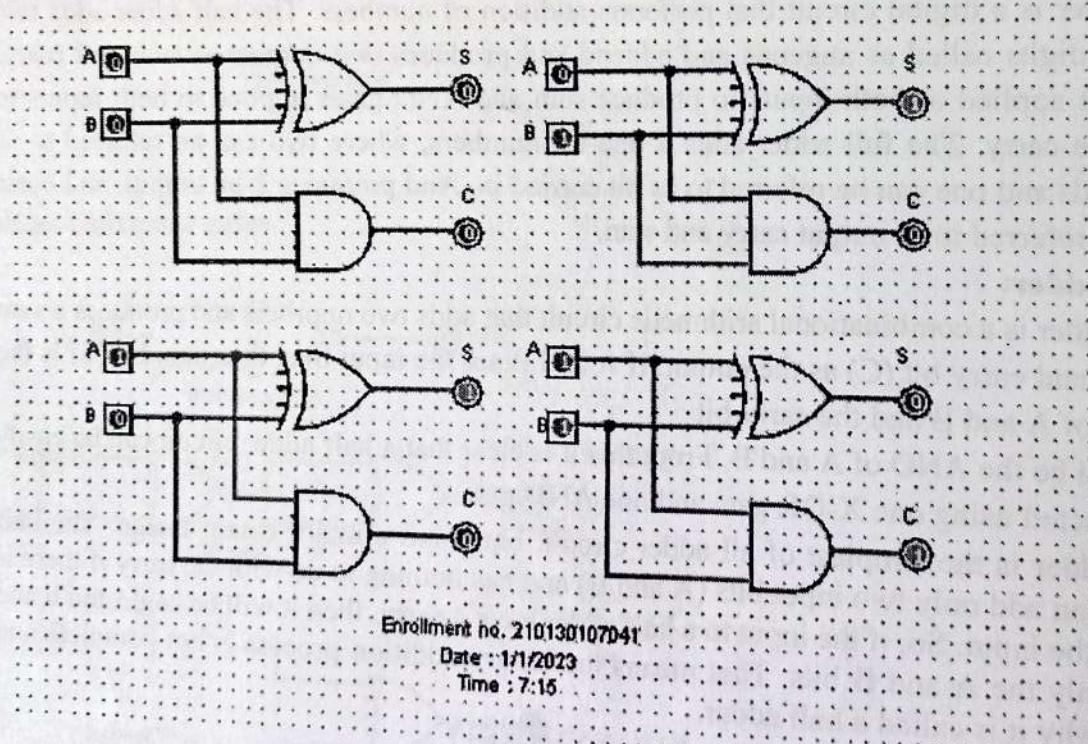
If any of the half adder logic produces a carry, there will be an output carry. Thus, COUT will be an OR function of the half adder CARRY outputs.

Boolean Expression: Difference  $S = X \oplus Y \oplus Z$

Borrow  $Bout = X' (Y \oplus Z) + YZ$

### Snapshots:

#### 1. Half Adder



Truth Table:

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

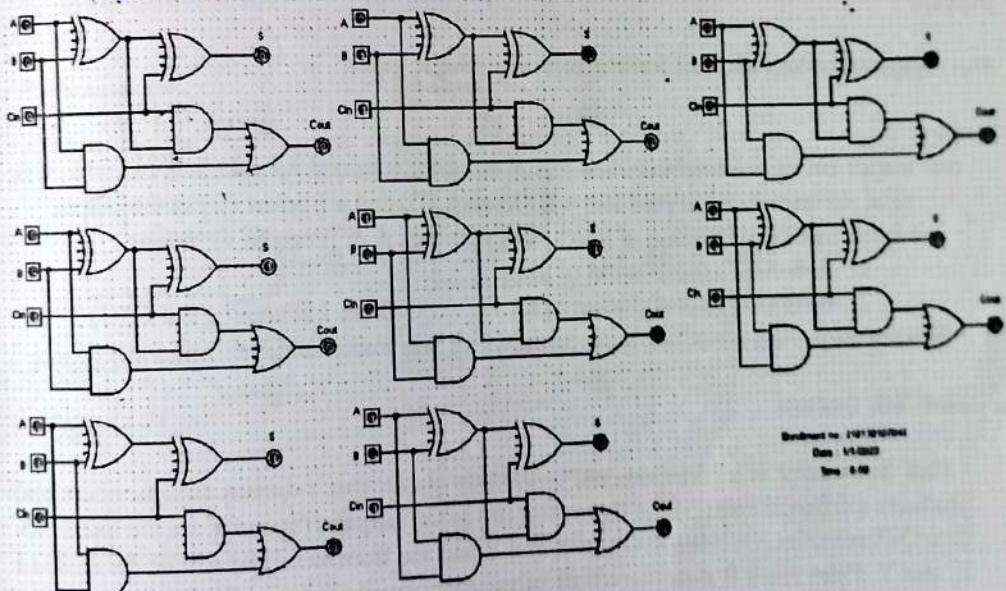
A → 1 bit half adder → S  
B → C

Schematic

## 2. Full Adder

$$AB + C(C(A \oplus B))$$

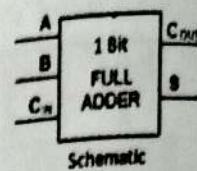
AIM: To construct full adder with 2 half adder circuit.



$$x \oplus y \oplus z$$

Truth Table:

X	Y	Z(C <sub>in</sub> )	S	C <sub>out</sub>
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
1	0	0	0	1
0	1	1	0	1
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1



## 15. Practical 3

CO3: Design and implement Combinational and Sequential logic circuits and verify its working.

### Module 2

Aim: Implement half and full Subtractors using logic gates.

#### Theory:

Subtractor circuit is a combinational digital circuit that is used for subtracting two numbers.

A typical subtractor circuit produces a Difference (D) and a Borrow (B) as the output.

We know that the subtraction of two binary numbers A & B can be written as,

1.  $A - B = A + (2^k \text{ complement of } B)$
  2.  $A - B = A + (1's \text{ complement of } B) + 1$  (here 1 is nothing but EAC)
- Subtractor circuits are of two types namely Full Subtractor and Half Subtractor.

#### Half Subtractor:

Half Subtractor is a combinational arithmetic circuit that subtracts two numbers and produces a difference bit (D) and borrow bit (B) as the output. If X and Y are the input bits then Difference bit (D) is the X-OR of X and Y and the Borrow bit (B) will be the AND of one X-OR gate, one NOT gate and one AND gate.

Half Subtractor is the simplest of all Subtractor circuit, but it has a major disadvantage. The half Subtractor can subtract only two input bits (X and Y) and has nothing to do with the borrow if there is any in the input. So, if the input to a half Subtractor have a borrow then it will be neglected it and subtracts only the A and B bits.

That means the binary addition process is not complete and that's why it is called a half Subtractor. Boolean Expression:

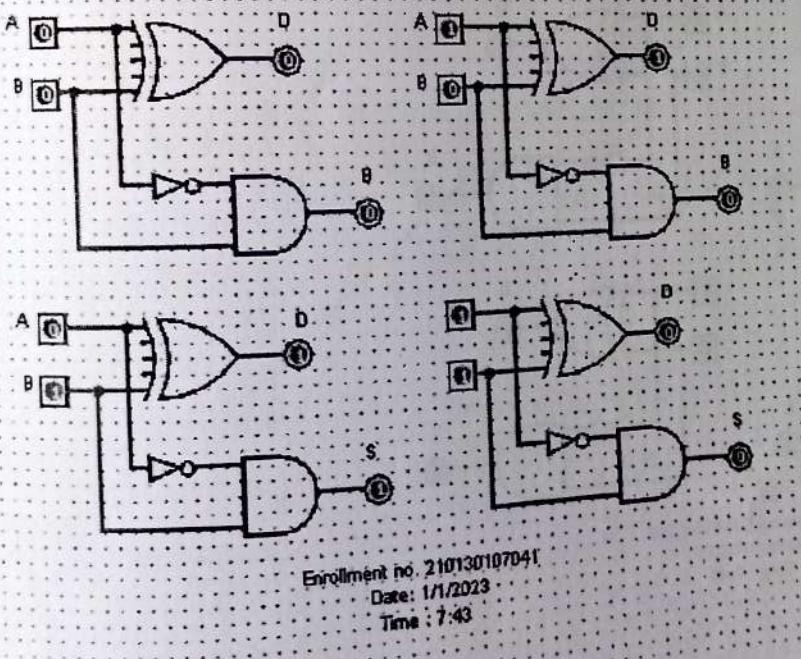
$$\text{Difference} = X'Y + XY' = X \oplus Y, \text{ Borrow } C = X' \cdot Y$$

#### Full Subtractor:

- Full Subtractor is little more difficult than a half-Subtractor circuit. The main difference in both Subtractor is that the full-Subtractor has three inputs and half-Subtractor has only two inputs.
- We have used three input variables X, Y and Z (Bin) which refers to the term minuend(X), subtrahend(Y) and borrow (Z or Bin) bit respectively. The output borrow is designated as  $B_{out}$  and the normal output is designated as D.
- Though the implementation of larger logic diagrams is possible with the below full adder logic a simpler symbol is mostly used to represent the operation. Given above is a simpler schematic representation of a one-bit full adder.
- Boolean Expression: Difference  $S = X \oplus Y \oplus Z$   
Borrow  $B_{out} = X' (Y \oplus Z) + YZ$

Snapshots:

## 1. Half Subtractor

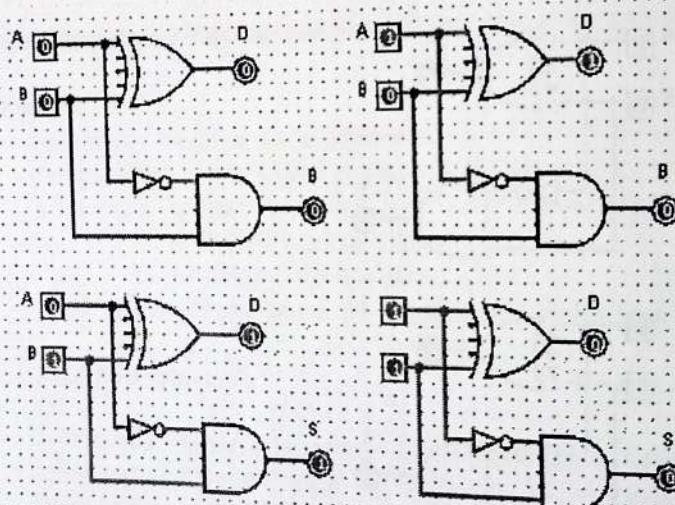


**TRUTH TABLE:**

Input X	Input Y	Difference e	Borrow	
0	0	0	0	
0	1	1	1	
1	0	1	0	
1	1	0	0	

Schematic

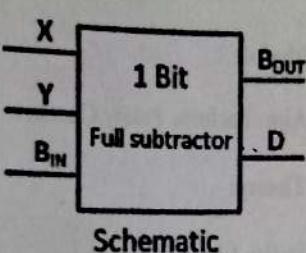
## 2. Full Subtractor :



Enrollment no. 210130107041  
Date: 1/1/2023  
Time : 7:43

**Truth Table:**

X	Y	Z( $B_{in}$ )	D	$B_{out}$
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	0	0	1	0
1	1	1	0	1
0	1	1	0	0
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1



# 16. Practical 4

CO3: Design and implement Combinational and Sequential logic circuits and verify its working.

## Module 2

Aim: Perform Parity Checker.

### Theory:

#### Parity Checker:

The combinational circuit at the receiver is the parity checker. This checker takes the received message including the parity bit as input. It gives output '1' if there is some error found and gives output '0' if no error is found in the message including the parity bit. There are two types of Parity Generator Even parity checker and Odd parity checker.

#### Even Parity Checker:

In even parity checker if the error bit (E) is equal to '1', then we have an error. If error bit E=0 then indicates there is no error. Error Bit

(E)=1, error occurs

Error Bit (E)=0, no error

#### Odd Parity Checker:

In odd parity checker if an error bit (E) is equal to '1', then it indicates there is no error. If an error bit E=0 then indicates there is an error.

Error Bit (E)=1, no error

Error Bit (E)=0, error occurs

### For Even Parity Checker:

K-map as shown below

		CP	00	01	11	10
		AB	00	01	11	10
AB	CP	00	0	1	0	1
		01	1	0	1	0
AB	CP	11	0	1	0	1
		10	1	0	1	0

$$\begin{aligned}
 PEC &= \bar{A} \bar{B} (\bar{C}D + \bar{C}\bar{D}) + \bar{A}B (\bar{C}\bar{D} + CD) + AB (\bar{C}D + \bar{C}\bar{D}) + A\bar{B} (\bar{C}\bar{D} + CD) \\
 &= \bar{A} \bar{B} (C \oplus D) + \bar{A}B (\bar{C} \oplus \bar{D}) + AB (C \oplus D) + A\bar{B} (\bar{C} \oplus \bar{D}) \\
 &= (\bar{A} \bar{B} + AB)(C \oplus D) + (\bar{A}B + A\bar{B})(\bar{C} \oplus \bar{D}) \\
 &= (A \oplus B) \oplus (C \oplus D)
 \end{aligned}$$

For Even parity checker:

4-bit received message				Parity error check $C_p$
A	B	C	P	
0	0	0	0	0
0	0	0	1	1
0	0	1	0	1
0	0	1	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

For Odd Parity Checker:

4-bit received message				Parity error check $C_p$
A	B	C	P	
0	0	0	0	1
0	0	0	1	0
0	0	1	0	0
0	0	1	1	1
0	1	0	0	0
0	1	0	1	1
0	1	1	0	1
0	1	1	1	0
1	0	0	0	0
1	0	0	1	1
1	0	1	0	1
1	0	1	1	0
1	1	0	0	1
1	1	0	1	0
1	1	1	0	0
1	1	1	1	1

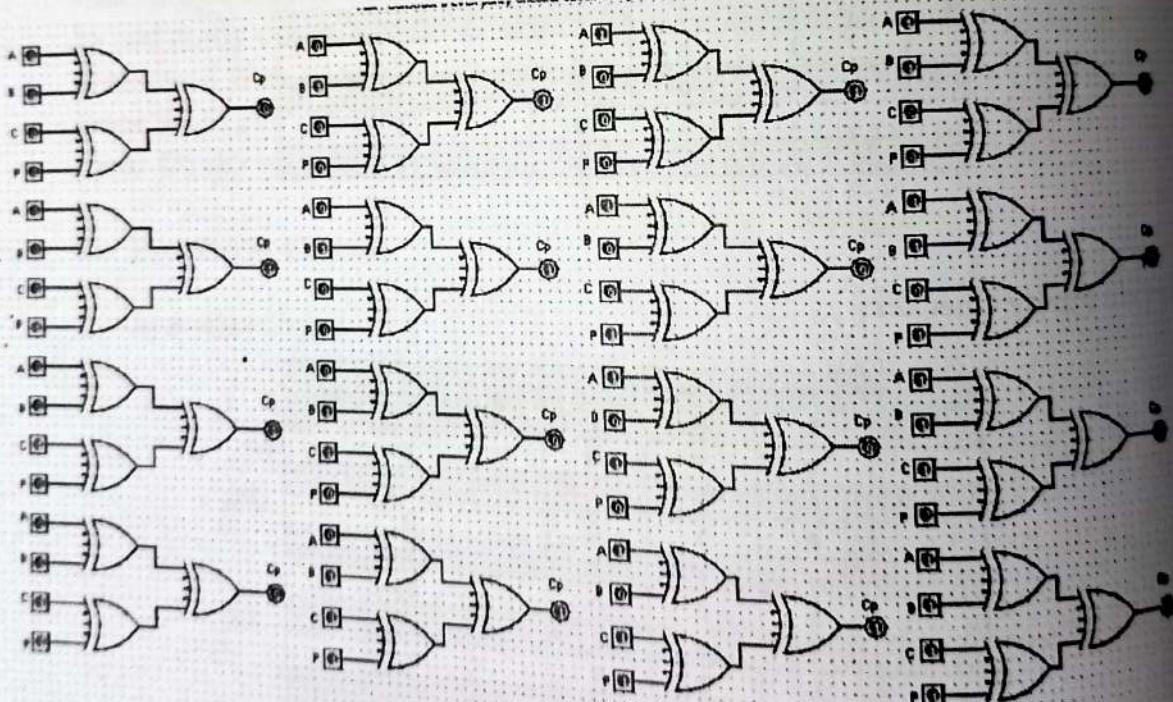
The expression for the PEC in the above truth table can be simplified by K-map as shown below.

		AB	00	01	11	10
		CP	0	1	3	2
		00	1	0	1	0
		01	0	1	0	1
		11	1	0	1	0
		10	0	1	0	1

After simplification, the final expression for the PEC is obtained as  
 $\text{PEC} = (\overline{A} \text{ Ex-NOR } \overline{B}) \text{ Ex-NOR } (\overline{C} \text{ Ex-NOR } \overline{D})$

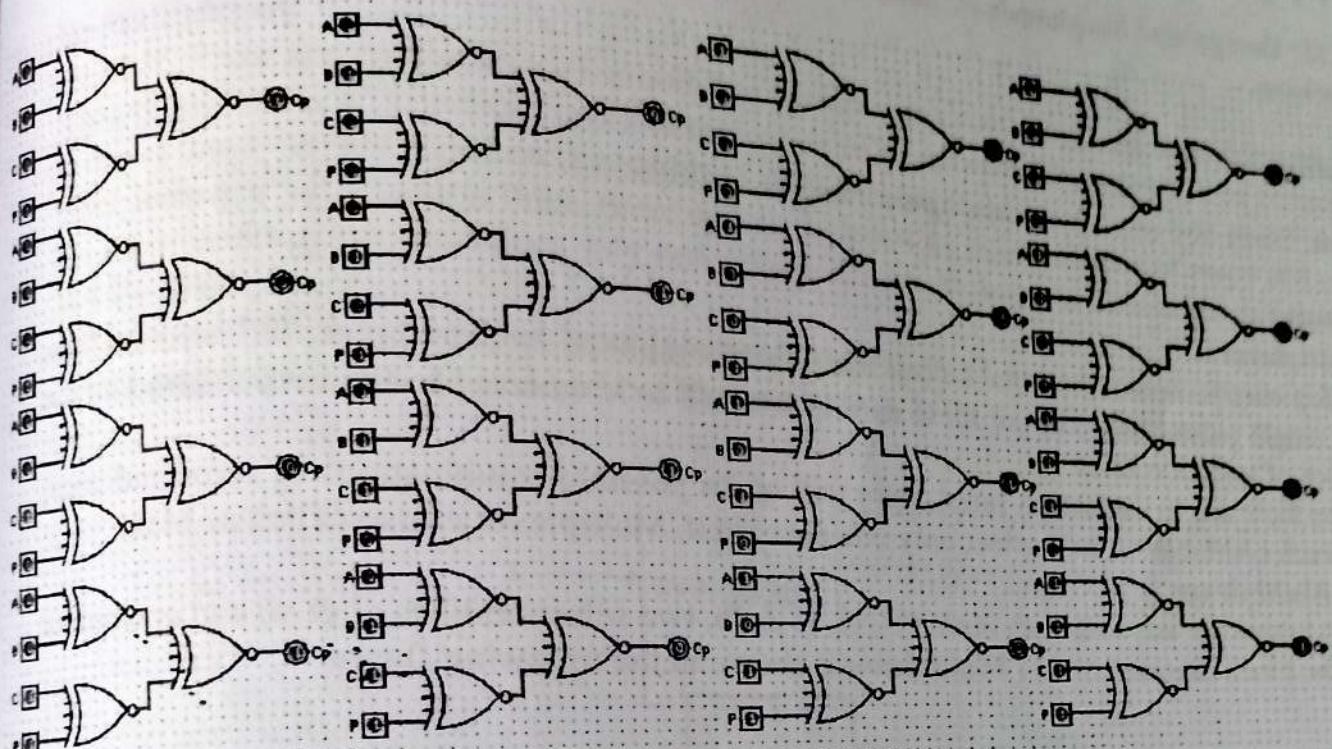
Snapshots:

## 1. EVEN Parity Checker



Enrollment no. 210130107041  
 Date : 2/1/2023  
 Time : 12:00

## 2. ODD Parity Checker



Enrollment no. 210130107041  
Date : 2/1/2023  
Time : 5:45

2/1/23

# 17. Practical 5

CO3: Design and implement Combinational and Sequential logic circuits and verify its working.

## Module 2

Aim: Study and implement Multiplexer and Demultiplexer.

### Theory:

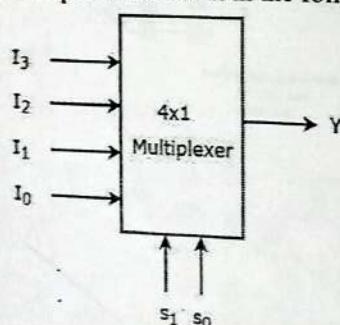
#### Multiplexer:

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

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

#### 4x1 Multiplexer

4x1 Multiplexer has four data inputs  $I_3, I_2, I_1$  &  $I_0$ , two selection lines  $s_1$  &  $s_0$  and one output Y. The block diagram of 4x1 Multiplexer is shown in the following figure.



One of these 4 inputs will be connected to the output based on the combination of inputs present at these two selection lines. Truth table of 4x1 Multiplexer is shown below.

Selection Lines		Output
$s_1$	$s_0$	Y
0	0	$I_0$
0	1	$I_1$
1	0	$I_2$
1	1	$I_3$

From Truth table, we can directly write the Boolean function for output, Y as  
 $Y = S_1'S_0'I' + S_1'S_0I + S_1S_0'I + S_1S_0I'$

We can implement this Boolean function using Inverters, AND gates & OR gate.

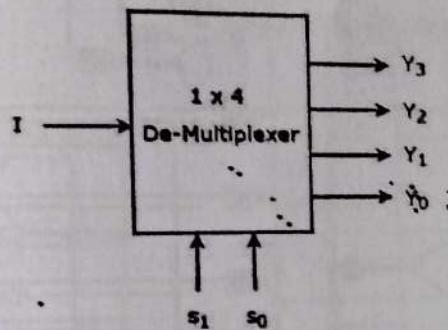
### De-Multiplexer:

De-Multiplexer is a combinational circuit that performs the reverse operation of Multiplexer. It has single input, 'n' selection lines and maximum of  $2^n$  outputs. 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 combinations of zeros and ones. So, each combination can select only one output. De-Multiplexer is also called as De-Mux.

### 1x4 De-Multiplexer:

1x4 De-Multiplexer has one input I, two selection lines,  $s_1$  &  $s_0$  and four outputs  $Y_3$ ,  $Y_2$ ,  $Y_1$  &  $Y_0$ . The block diagram of 1x4 De-Multiplexer is shown in the following figure.



The single input 'I' will be connected to one of the four outputs,  $Y_3$  to  $Y_0$  based on the values of selection lines  $s_1$  &  $s_0$ . The Truth table of 1x4 De-Multiplexer is shown below.

Selection Inputs		Outputs			
$s_1$	$s_0$	$Y_3$	$Y_2$	$Y_1$	$Y_0$
0	0	0	0	0	1
0	1	0	0	1	0
1	0	0	1	0	0
1	1	1	0	0	0

From the above Truth table, we can directly write the Boolean functions for each output as

$$Y_3 = s_1 s_0 I$$

$$Y_2 = s_1 s_0' I$$

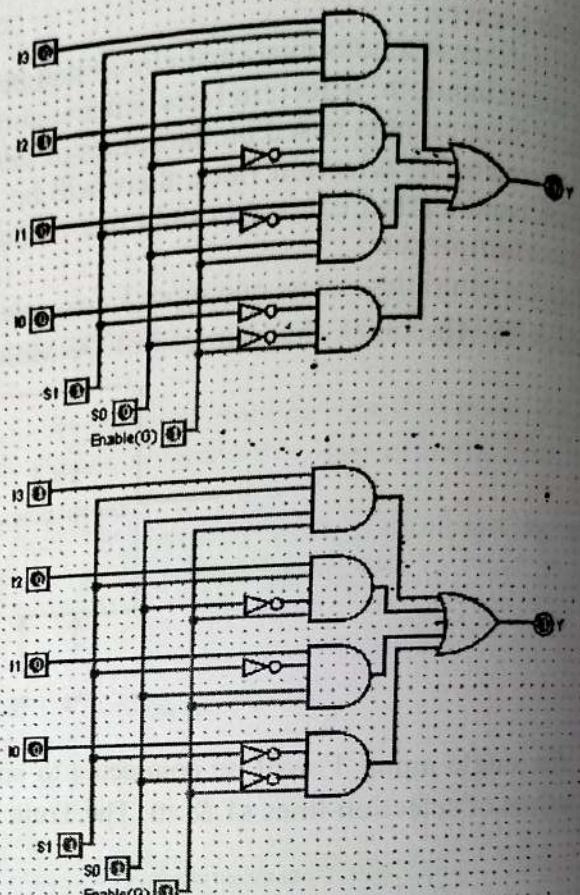
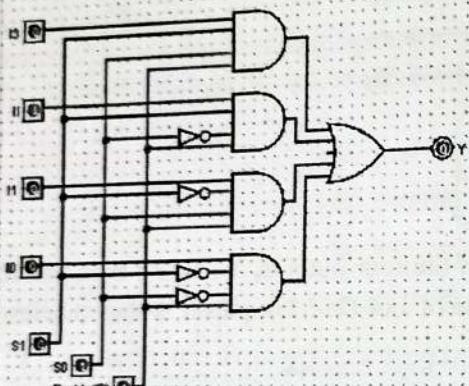
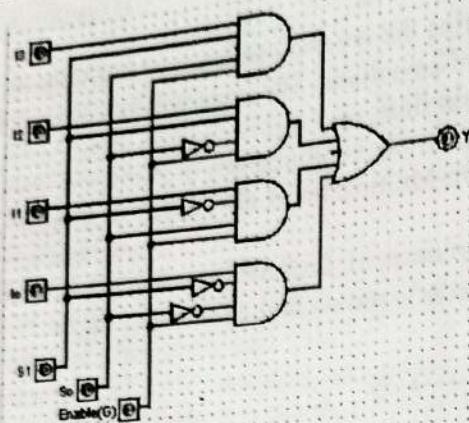
$$Y_1 = s_1' s_0 I$$

$$Y_0 = s_1' s_0' I$$

We can implement these Boolean functions using Inverters & 3-input AND gates.

Snapshots:

## 1. 4X1 MULTIPLEXER

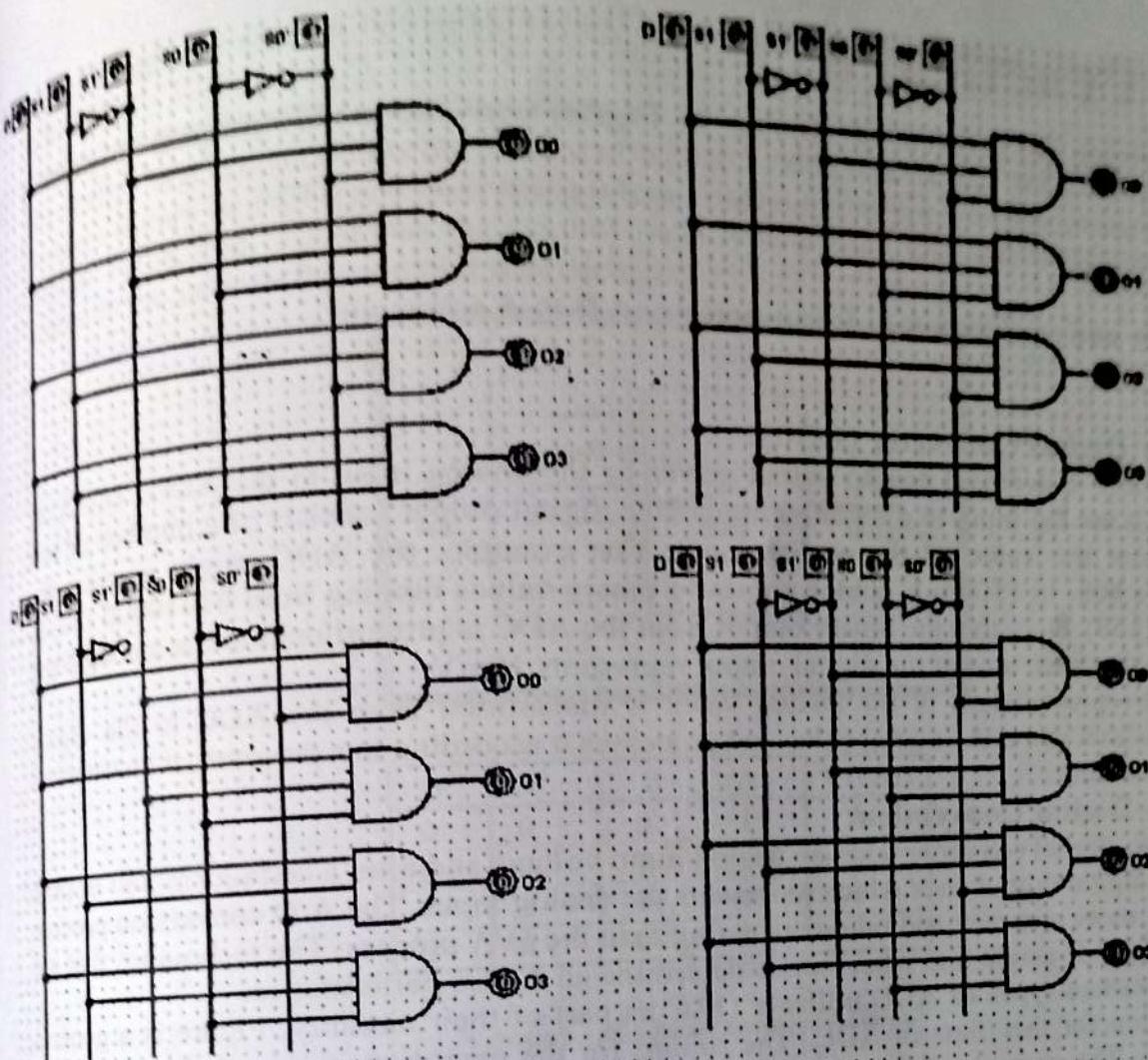


enrollment no: 210130107041  
date : 1/1/2023  
time: 12:00

## Truth Table:

Selection Lines		Output
S <sub>1</sub>	S <sub>0</sub>	Y
0	0	I <sub>0</sub>
0	1	I <sub>1</sub>
1	0	I <sub>2</sub>
1	1	I <sub>3</sub>

## 2. 1X4 DEMULTIPLEXER



enrollment no. 21D130107041  
date: 2/1/2023  
time : 12:30

### Truth Table:

Selection Inputs		Outputs			
$S_1$	$S_0$	$Y_0$	$Y_2$	$Y_1$	$Y_3$
0	0	0	0	1	0
0	1	0	0	0	0
1	0	0	1	0	0
1	1	1	0	0	0

1/2

# 18. Practical 6

CO4: Examine the process of Analog to Digital conversion and Digital to Analog conversion

## Module 4

Aim: Study and configure A to D convertor and D to A convertor.

### Theory:

#### Analog-to-Digital Converter (ADC):

The transducer's electrical analog output serves as the analog input to the ADC. The ADC converts this analog input to a digital output. This digital output consists of a number of bits that represent the value of the analog input. For example, the ADC might convert the transducer's 800- to 1500-mV analog values to binary values ranging from 01010000 (80) to 10010110 (150). Note that the binary output from the ADC is proportional to the analog input voltages so that each unit of the digital output represents 10mV.

The digital representation of the analog values is transmitted from the ADC to the digital computer, which stores the digital value and processes it according to a program of instructions that it is executing.

#### Analog-to-Digital Conversion:

An analog-to-digital converter takes an analog input voltage and after a certain amount of time produces a digital output code which represents the analog input. The A/D conversion process is generally more complex and time-consuming than the D/A process. The techniques that are used provide and insight into what factors determine an ADCs performance.

Several important types of ADC utilize a DAC as part of their circuitry. Figure is a general block diagram for this class of ADC. The timing for the operation is provided by the input clock signal. The control unit contains the logic circuitry for generating the proper sequence of operations. The START COMMAND, initiates the conversion process, The op-amp compactor has two analog inputs and a digital output that switches states, depending on which analog input is greater.

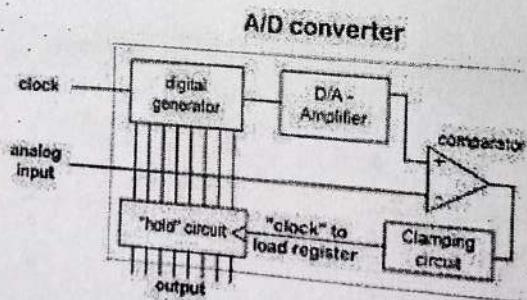


Figure 18.3 The A/D circuit layout.

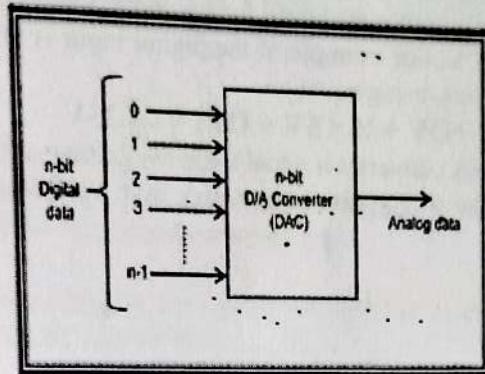
#### Digital-to-Analog Converter (DAC)

This digital output from the computer is connected to a DAC, which converts it to a proportional analog voltage or current. For example, the computer might produce a digital

output ranging from 0000000 to 1111111, which the DAC converts to a voltage ranging from 0 to 10V.

### Digital to Analog (D to A) Conversion:

Basically, D/A conversion is the process of taking a value represented in digital code (such as straight binary or BCD) and converting it to a voltage or current which is proportional to the digital value. Fig. 7.2 shows the symbol for a typical 4-bit D/A converter. Now, we will examine the various input/output relationships.



D	C	B	A	V(out)
0	0	0	1	1
0	0	1	0	2
0	1	0	0	4
1	0	0	0	8

D/A Converter With Binary Weighted Resistors

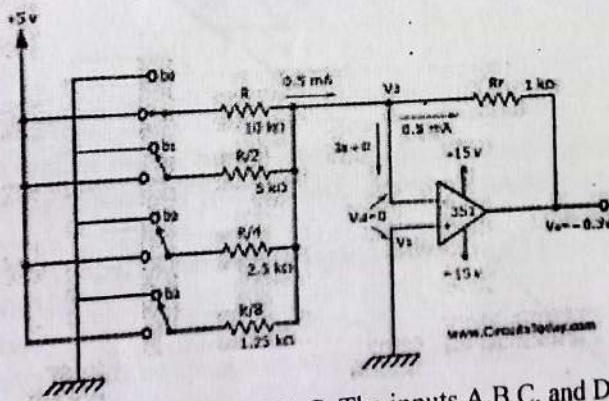


Figure shows the basic circuit of 4-bit DAC. The inputs A, B, C, and D are binary inputs which are assumed to have values of either 0 V or 5 V. The operational amplifier is employed as a summing amplifier, which produces The purpose of a digital-to-analog converter is to convert a binary word to a proportional current or voltage. The weighted sum of these input voltages. The summing amplifier multiplies each input voltage by the ratio of the feedback resistor  $R_f$  to the corresponding input resistor  $R_{IN}$ . In this circuit  $R_f = 1\text{ k}\Omega$  and the input resistors range from 1 to  $8\text{ k}\Omega$ . The D input has  $R_{IN} = 11.7\text{ k}\Omega$ .

$R_{IN} = 1K\ \Omega$ , so the summing amplifier passes the voltage at D with no attenuation. The C input has  $R_{IN} = 2\ k\ \Omega$ , so that it will be attenuated by. Similarly, the B input will be attenuated by  $\frac{1}{4}$  and the A input by  $1/8$ . The amplifier output can thus be expressed as

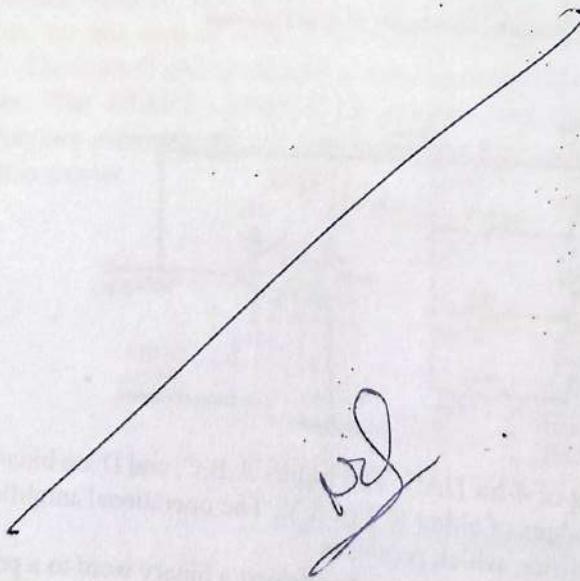
$$V_{OUT} = - (V_D + \frac{1}{2} V_C + \frac{1}{4} V_B + \frac{1}{8} V_A)$$

The negative sign is present because the summing amplifier is a polarity-inverting amplifier, but it will not concern us here.

Clearly, the summing amplifier output is an analog voltage which represents a weighted sum of the digital inputs. The output is evaluated for any input condition by setting the appropriate inputs to either 0 V or 5 V. For example, if the digital input is 1010, then  $V_D = V_B = 5V$  and  $V_C = V_A = 0V$ . Thus, using equation

$$V_{OUT} = - (5V + 0V + \frac{1}{4} \times 5V + 0V) = - 6.25V$$

The resolution of this D/A converter is equal to the weighting of the ISB, which is  $\frac{1}{8} \times 5V = 0.625\ V$ . The analog output increases by 0.625 V as the binary input number advances one step.



# 19. Practical 7

CO3: Design and implement Combinational and Sequential logic circuits and verify its working

## Module 3

Aim: Study and implement a shifter.

### Theory:

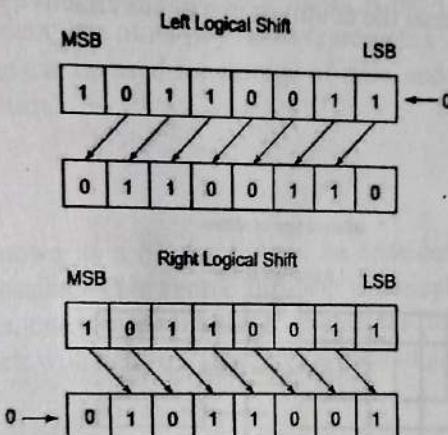
#### Shifters: -

Shifters move bits and multiply or divide by powers of 2. As the name implies, a shifter shifts a binary number left or right by a specified number of positions.

There are several kinds of commonly used shifters: -

#### 1) Logical Shifter:

shifts the number to the left (LSL) or right (LSR) and fills empty spots with 0's: Ex: 11001 LSR 2=00110; 11001 LSL 2=00100



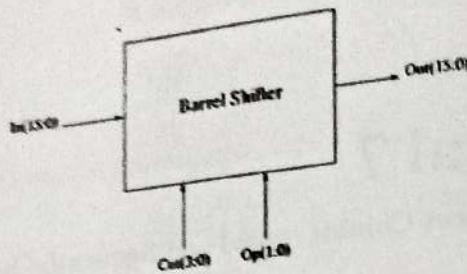
#### 2) Arithmetic shifter:

is the same as a logical shifter, but on right shifts fills the most significant bits with a copy of the old most significant bit (msb). This is useful for multiplying and dividing signed numbers

Arithmetic shift left (ASL) is the same as logical shift left (LSL). Ex: 11001 ASR 2=11110; 11001 ASL 2=00100

#### 3) Barrel Shifters

The people most affected by change on a personal level seem in many cases to be blame-shifters. The usual laments include, "The boss had it in for me. The company deceived me. Lady Luck was against me." They are self-doubters who end up failing because they fear success. They tend to undervalue their talents and destroy their chances of getting ahead. The losers in change are sometimes people who are so locked into their own identity they cannot tolerate working differently. Change to the job is unacceptable because the job and the person are one and the same. "They wanted me to be the fleet manager when I was already the head of security. They can't do that to me." But they did.

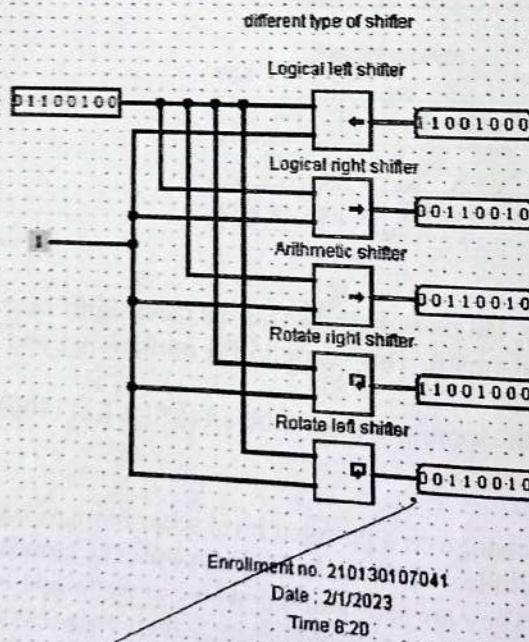


#### 4) Frequency Shifter

As demonstrated in Figure the conversion from LP01 to LP11 occurs with a frequency shift equal to the acoustic frequency. Thus, the setup in Figure acts as an optical frequency shifter. The fact that the frequency-shifted light is in the LP11 mode is inconvenient for coupling into a single-mode fiber. However, this problem can be solved by means of a static mode coupler that converts the light back to the LP01 mode. The basic principle of two different static couplers is illustrated in Figure. The mechanical mode coupler consists of two corrugated plates that impose a periodic bending on the fiber, with a period equal to the beat length at the optical center wavelength. The photorefractive mode coupler is a two-mode fiber where a permanent photorefractive grating has been written. For coupling between the LP01 and LP11 modes to occur, the grating must be written at a skew angle as indicated in Figure. The mechanical mode coupler can easily provide complete coupling between the modes, whereas the coupling in a photorefractive grating is relatively weak.

Code:

**SHIFTER:**



# 20. Practical 8

CO3: Design and implement Combinational and Sequential logic circuits and verify its working

## Module 3

**Aim:** Study and implement Flip-flops.

### Theory:

In electronics, a flip-flop or latch is a circuit that has two stable states and can be used to store state information – a bistable multivibrator. The circuit can be made to change state by signals applied to one or more control inputs and will have one or two outputs. It is the basic storage element in sequential logic. Flip-flops and latches are fundamental building blocks of digital electronics systems used in computers, communications, and many other types of systems. Flip-flops and latches are used as data storage elements. A flip-flop is a device which stores a single bit (binary digit) of data; one of its two states represents a "one" and the other represents a "zero". Such data storage can be used for storage of state, and such a circuit is described as sequential logic in electronics.

### SR flip flop:

The SR flip-flop, also known as a SR Latch, can be considered as one of the most basic sequential logic circuit possible. This simple flip-flop is basically a one-bit memory bistable device that has two inputs, one which will "SET" the device (meaning the output = "1"), and is labelled S and one which will "RESET" the device (meaning the output = "0"), labelled R.

### JK flip flop:

The J-K flip-flop is the most versatile of the basic flip-flops. It has the input- following character of the clocked D flip-flop but has two inputs, traditionally labeled J and K. If J and K are different then the output Q takes the value of J at the next clock edge.

### D flip flop:

The D-type flip-flop is a modified Set-Reset flip-flop with the addition of an inverter to prevent the S and R inputs from being at the same logic level. One of the main disadvantages of the basic SR NAND Gate Bistable circuit is that the indeterminate input condition of SET = "0" and RESET = "0" is forbidden.

This state will force both outputs to be at logic "1", over-riding the feedback latching action and whichever input goes to logic level "1" first will lose control, while the other input still at logic "0" controls the resulting state of the latch.

But in order to prevent this from happening an inverter can be connected between the "SET" and the "RESET" inputs to produce another type of flip flop circuit known as a Data Latch,

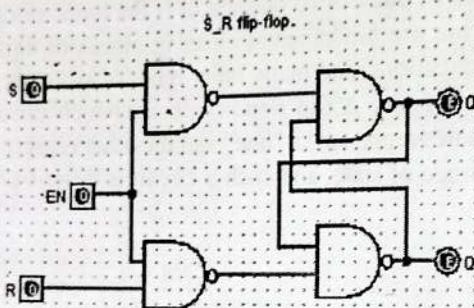
Delay flip flop, D-type Bistable, D type Flip Flop or just simply a D Flip Flop as it is more generally called.

### T flip flop:

T flip-flop is an edge triggered device i.e. the low to high or high to low transitions on a clock signal of narrow triggers that is provided as input will cause the change in output state of flip-flop.

### Snapshots:

## SR FLIP FLOP

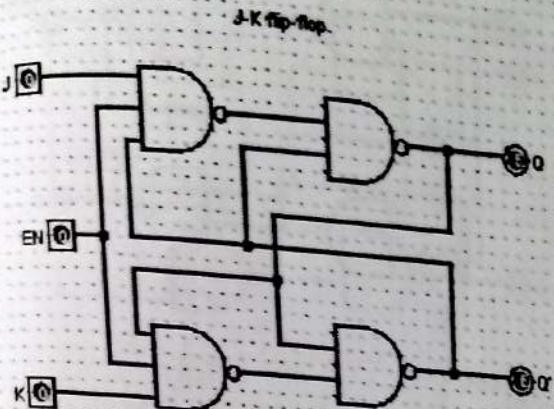


enrollment no. 210130107041  
DATE : 31/12/2023  
Time : 8:40

### Truth Table:

En	S	R	$Q_n$	$Q_{n+1}$	State
1	0	0	0	0	No Change (NC)
1	0	0	1	1	
1	0	1	0	0	
1	0	1	1	0	Reset
1	1	0	0	1	
1	1	0	1	1	Set
1	1	1	0	X	
1	1	1	1	X	Indeterminate (Invalid)
0	X	X	0	0	
0	X	X	1	1	No Change (NC)

## JK FLIP FLOP



Enrollment no. 210130107041

Date : 31/12/2022

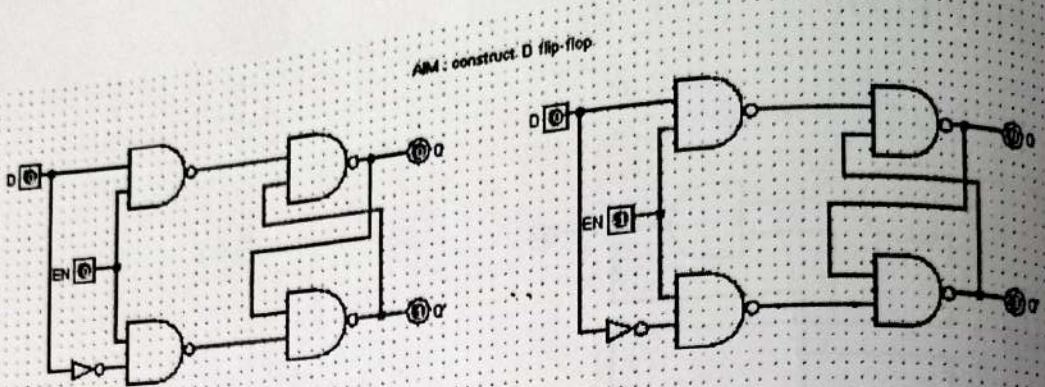
Time : 10:30

## Truth Table:

En	J	K	$Q_n$	$Q_{n+1}$	State
1	0	0	0	0	No Change (NC)
1	0	0	1	1	
1	0	1	0	0	
1	0	1	1	0	Reset
1	1	0	0	1	
1	1	0	1	1	
1	1	1	0	1	Set
1	1	1	1	0	
0	X	X	0	0	No Change (NC)
0	X	X	1	1	

## D FLIP FLOP

ED

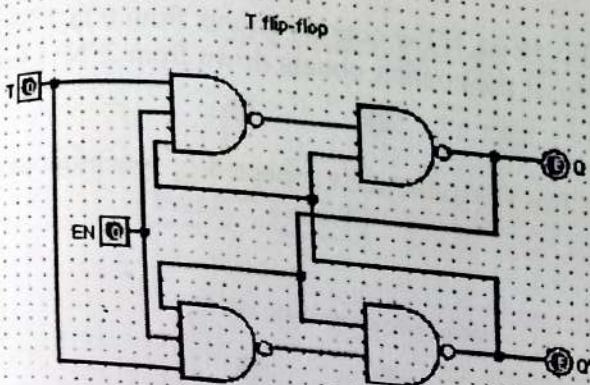


enrollment no. 210130107041  
 Date : 1/1/2023  
 Time : 11:00

Truth Table:

En	D	$Q_n$	$Q_{n+1}$	State
1	0	0	0	Reset
1	0	1	0	
1	1	0	1	Set
1	1	1	1	
0	X	0	0	No Change (NC)
0	X	1	1	

## T FLIP FLOP



Enrollment no. 210130107041

Date : 31/12/2022

Time : 8:23

**Truth Table:**

En	T	$Q_n$	$Q_{n+1}$	State
1	0	0	0	No Change (NC)
1	0	1	1	
1	1	0	1	Toggle
1	1	1	0	
0	X	0	0	No Change (NC)
0	X	1	1	

## 21. Practical 9

**CO3: Design and implement Combinational and Sequential logic circuits and verify its working**

**Module 3**

**Aim: Study and implement Counter.**

**Theory:**

### **Counters**

An 'N' bit binary counter consists of 'N' T flip-flops. If the counter counts from 0 to  $2^N - 1$ , then it is called as binary up counter. Similarly, if the counter counts down from  $2^N - 1$  to 0, then it is called as binary down counter.

There are two types of counters based on the flip-flops that are connected in synchronous or not.

- Asynchronous counters
- Synchronous counters

#### **Asynchronous counters:**

The logic diagram of a 2-bit ripple up counter is shown in figure. The toggle(T) flip-flop are being used. But we can use the JK flip-flop also with J and K connected permanently to logic 1. External clock is applied to the clock input of flip-flop A and Q A output is applied to the clock input of the next flip-flop.

#### **Synchronous counters**

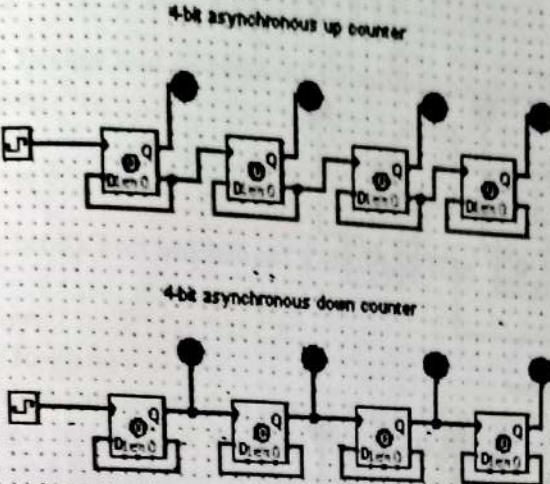
If the "clock" pulses are applied to all the flip-flops in a counter simultaneously, then such a counter called asynchronous counter.

#### **2- bit Synchronous up counter:**

The JA and KA inputs of FF-A are tied to logic1. So, FF-A will work as a toggle flip-flop. The JB and KB inputs are connected to QA.

#### **Snapshots:**

#### **1. Asynchronous counters:**



Enrollment no: 210130107041

Date : 1/1/2023

Time : 10:00

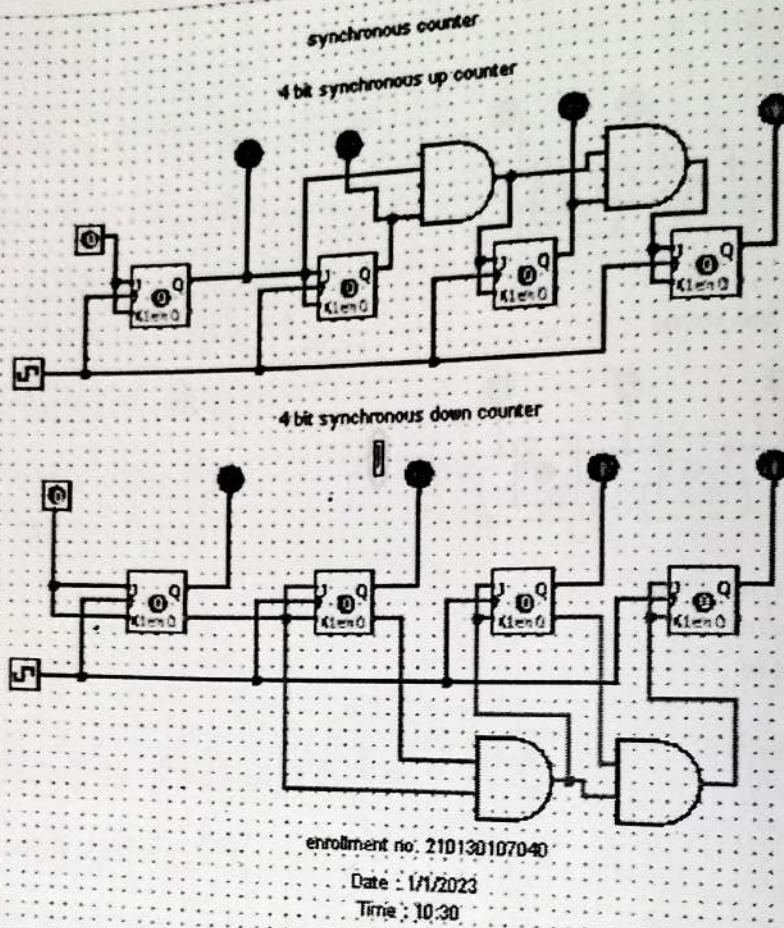
## Truth Table:

D3	D2	D1	D0
0	0	0	0
0	0	0	1
0	0	1	0
0	0	1	1
0	1	0	0
0	1	0	1
0	1	1	0
0	1	1	1
1	0	0	0
1	0	0	1
1	0	1	0
1	0	1	1
1	1	0	0
1	1	0	1
1	1	1	0
1	1	1	1

For up counter

For down counter

## Synchronous counters:



### Truth Table:

D3	D2	D1	D0
0	0	0	0
0	0	0	1
0	0	1	0
0	0	1	1
0	1	0	0
0	1	0	1
0	1	1	0
1	0	1	1
1	0	0	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	0	1
1	1	1	0
1	1	1	1

For up counter

For down counter

## 22. Practical 10

CO3: Design and implement Combinational and Sequential logic circuits and verify its working

### Module 3

Aim: Study and implement a shift register.

#### Theory:

##### Shift Register:

Flip flops can be used to store a single bit of binary data (1 or 0). However, in order to store multiple bits of data, we need multiple flip flops. N flip flops are to be connected in an order to store n bits of data. A **Register** is a device which is used to store such information. It is a group of flip flops connected in series used to store multiple bits of data.

The information stored within these registers can be transferred with the help of **shift registers**. Shift Register is a group of flip flops used to store multiple bits of data. The bits stored in such registers can be made to move within the registers and in/out of the registers by applying clock pulses. An n-bit shift register can be formed by connecting n flip-flops where each flip flop stores a single bit of data.

The registers which will shift the bits to left are called "Shift left registers". The registers which will shift the bits to right are called "Shift right registers".

Shift registers are basically of 4 types. These are:

1. Serial in Serial Out shift register
2. Serial in parallel Out shift register
3. Parallel in Serial Out shift register
4. Parallel in parallel Out shift register

##### 1) Serial-In Serial-Out Shift Register (SISO):

The shift register, which allows serial input (one bit after the other through a single data line) and produces a serial output is known as Serial-In Serial-Out shift register. Since there is only one output, the data leaves the shift register one bit at a time in a serial pattern, thus the name Serial-In Serial-Out Shift Register.

The logic circuit given below shows a serial-in serial-out shift register. The circuit consists of four D flip-flops which are connected in a serial manner. All these flip-flops are synchronous with each other since the same clock signal is applied to each flip flop.

##### 2) Serial-In Parallel-Out shift Register (SIPO):

The shift register, which allows serial input (one bit after the other through a single data line) and produces a parallel output is known as Serial-In Parallel-Out shift register.

The logic circuit given below shows a serial-in-parallel-out shift register. The circuit consists of four D flip-flops which are connected. The clear (CLR) signal is connected in addition to the clock signal to all the 4 flip flops in order to RESET them. The output of the first flip flop is connected to the input of the next flip flop and so on. All these flip-flops are synchronous with each other since the same clock signal is applied to each flip flop.

### 2) Parallel-In Serial-Out Shift Register (PISO):

The shift register, which allows parallel input (data is given separately to each flip flop and in a simultaneous manner) and produces a serial output is known as Parallel-In Serial-Out shift register.

The logic circuit given below shows a parallel-in-serial-out shift register. The circuit consists of four D flip-flops which are connected. The clock input is directly connected to all the flip flops but the input data is connected individually to each flip flop through a multiplexer at the input of every flip flop. The output of the previous flip flop and parallel data input are connected to the input of the MUX and the output of MUX is connected to the next flip flop. All these flip-flops are synchronous with each other since the same clock signal is applied to each flip flop.

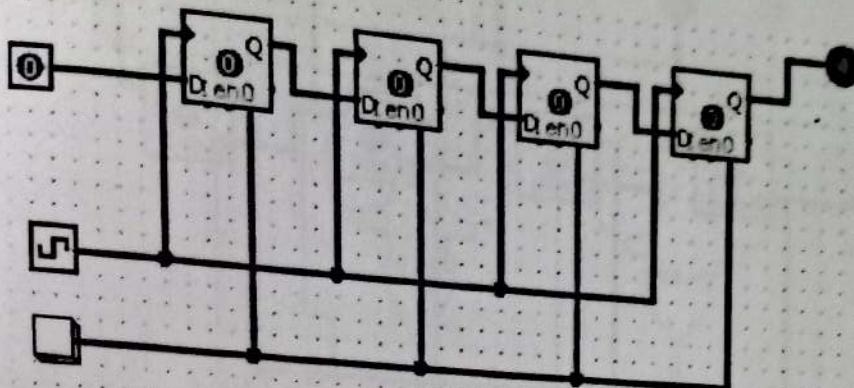
### 3) Parallel-In Parallel-Out Shift Register (PIPO):

The shift register, which allows parallel input (data is given separately to each flip flop and in a simultaneous manner) and also produces a parallel output is known as Parallel-In parallel Out shift register.

The logic circuit given below shows a parallel-in-parallel-out shift register. The circuit consists of four D flip-flops which are connected. The clear (CLR) signal and clock signals are connected to all the 4 flip flops. In this type of register, there are no interconnections between the individual flip-flops since no serial shifting of the data is required. Data is given as input separately for each flip flop and in the same way, output also collected individually from each flip flop.

### Snapshots:

## 1. Serial-In Serial-Out Shift Register (SISO):



enrollment no. 210130107041

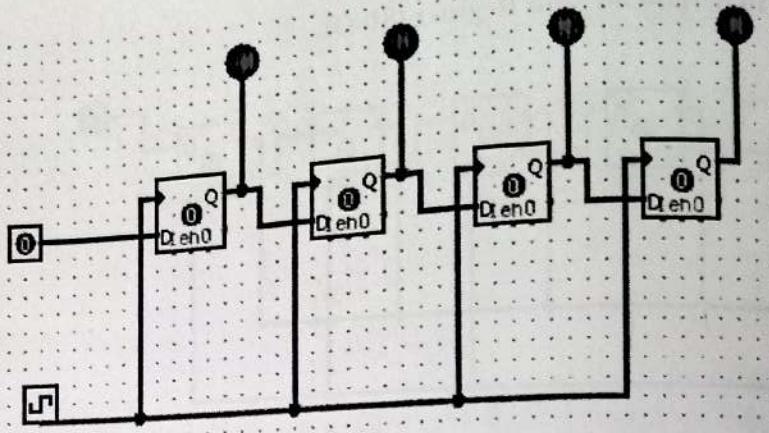
Date: 2/1/2023

Time : 5:28

### Truth Table:

CLK	Serial in	Serial out
1	1	0
2	0	0
3	0	0
4	1	1
5	X	0
6	X	0
7	X	1

## 2. Serial-In Parallel-Out Shift Register (SIPO):



enrollment no. 210130107041

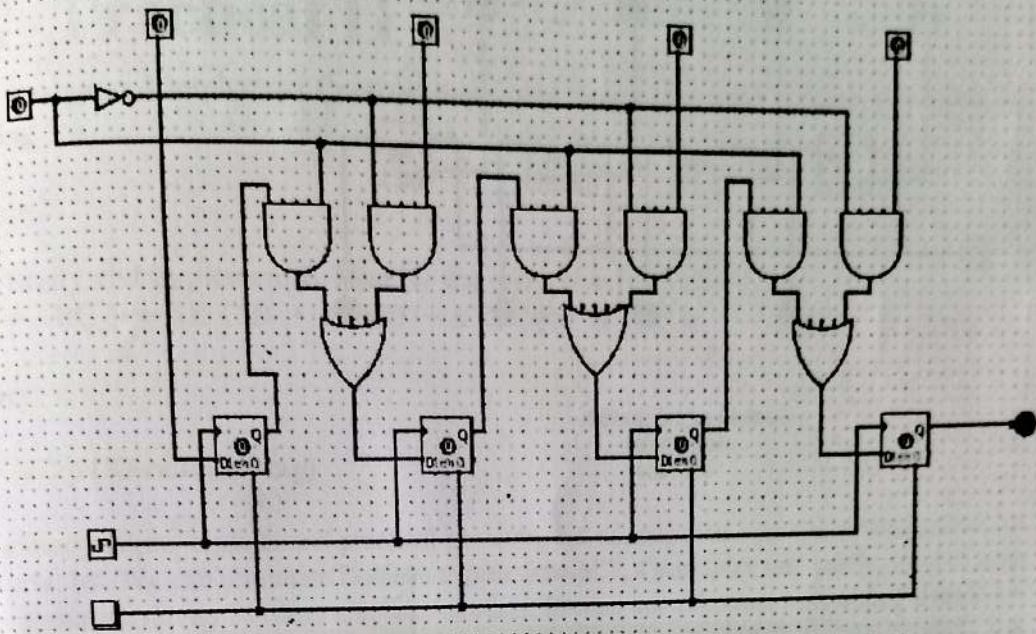
Date: 2/1/2023

Time 4:50

**Truth Table:**

Clock Pulse No	QA	QB	QC	QD
0	0	0	0	0
1	1	0	0	0
2	0	1	0	0
3	0	0	1	0
4	0	0	0	1
5	0	0	0	0

### 3. Parallel-In Serial-Out Shift Register (PISO):



enrollment no. 210130107041

Date: 02/01/23

Time : 2:46

$$\begin{array}{r}
 23 \\
 149 \\
 \hline
 42
 \end{array}
 \begin{array}{r}
 0010\ 0011 \\
 +0001\ 0100 \\
 \hline
 0011\ 1100
 \end{array}$$

~~0001 0001~~

Truth Table :

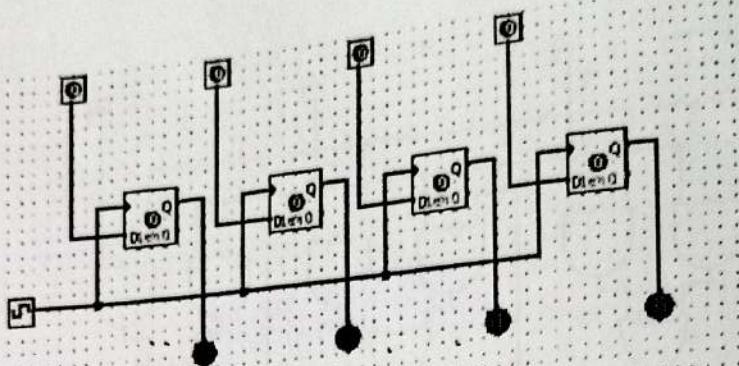
CLK	Q3	Q2	Q1	Q0	O/P
0	1	0	0	1	1
1	0	0	0	0	0
2	0	0	0	0	0
3	0	0	0	0	1

#### 4. Parallel-In Parallel-Out Shift Register (PIPO):

$$\begin{array}{r}
 0011\ 1100 \\
 +1010\ 1010 \\
 \hline
 1001\ 0010
 \end{array}$$

$$\begin{array}{r}
 0011\ 1100 \\
 -0011\ -0011 \\
 \hline
 0011
 \end{array}$$

$$\begin{array}{r}
 0011\ 1100 \\
 0110\ 0110 \\
 \hline
 01000010
 \end{array}$$



enrollment no. 210130107041

Date: 2/1/2023

Time : 2:30

Truth Table :

CLK	DATA INPUT				OUTPUT			
	DA	DB	DC	DD	QA	QB	QC	QD
1	1	0	0	1	1	0	0	1
2	1	0	1	0	1	0	1	0

PS

# 23. Practical 11

CO3: Design and implement Combinational and Sequential logic circuits and verify its working

## Module 2

Aim: Study and implement K-Map for the given function:(SOP)

$$F = ABC'D + AB'C'D' + AB'CD' + A'B'CD' + A'B'C'D'$$

Theory:

### K-Map (Karnaugh Map)

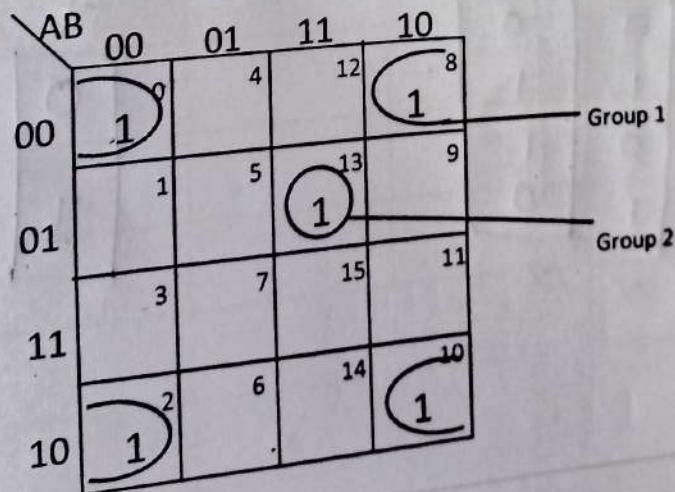
In many digital circuits and practical problem, we need to find expression with minimum variables. We can minimize Boolean expressions of 3, 4 variables very easily using K-map without using any Boolean algebra theorems. K-map can take two forms Sum of Product (SOP) and Product of Sum (POS) according to the need of problem. K-map is table like representation but it gives more information than TRUTH TABLE. We fill grid of K-map with 0's and 1's then solve it by making groups.

Code:

### K-Map:

$$F = ABC'D + AB'C'D' + AB'CD' + A'B'CD' + A'B'C'D'$$

$$F(A, B, C, D) = \Sigma(0, 2, 8, 10, 13)$$



Here, In RED color is First group and BLUE color is second group.

First consider group 1,

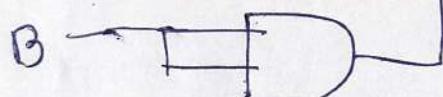
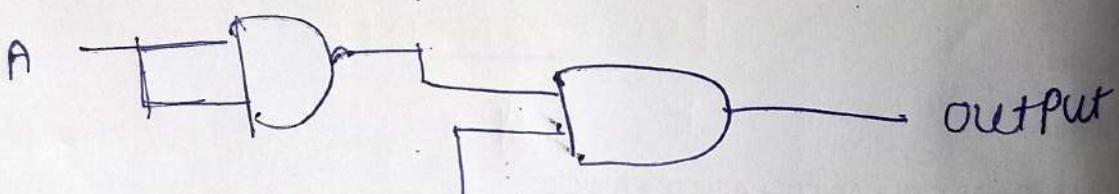
RED group's Product term =  $B'D'$

Now, consider group 2,

BLUE group's Product term =  $ABC'D$

To obtain final solution we add product term of Group 1 and Group 2,

$$F(A, B, C, D) = B'D + ABC'D$$



En	T	Qn	Qntl
0	X	X	X
0	0	0	0
0	0	1	1
1	0	0	1
1	1	1	0

T	Qn	Qntl
0	0	0
1	0	1

Truth Table :

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

ED