



Government Engineering College, Gandhinagar

**Computer Engineering
B.E. Semester III
(AY 2021-22)**

SUBJECT: Digital Fundamental (3130704)



Government Engineering College

Sec-28 Gandhinagar

Certificate

This is to certify that

Mr./Ms. Rajashree Sugal N. Of class

3rd Division At., Enrollment No. D2D.17..... Has

Satisfactorily completed his/her term work

Digital Attachment Subject for the term ending in

Jan 2022-23

Date:-

Contents

1. Institute Vision/Mission	4
Vision:	4
Mission:	4
2. Computer Engineering Department Vision/Mission	7
Vision:	7
Mission:	7
3. Program Educational Outcomes (PEO)	8
4. PSO	8
5. POs	8
6. Submission Instructions:	Email Bookmark not defined.
7. Assignment Index	10
8. Practical Index	11
9. Assignment 1	12
10. Assignment 2	13
11. Assignment 3	14
12. Assignment 4	15
13. Assignment 5	16
14. Practical 1	17
15. Practical 2	18
16. Practical 3	19
17. Practical 4	20
18. Practical 5	21
19. Practical 6	22
20. Practical 7	23
21. Practical 8	24
22. Practical 9	25
23. Practical 10	26
24. Practical 11	27



Institute Vision/Mission

Vision:

- To be a premier engineering institution, imparting quality education for innovative solutions relevant to society and environment.

Mission:

- To develop human potential to its fullest extent so that intellectual and innovative engineers can emerge in a wide range of professions.
- To advance knowledge and educate students in engineering and other areas of scholarship that will best serve the nation and the world in future.
- To produce quality engineers, entrepreneurs and leaders to meet the present and future needs of society as well as environment.

Computer Engineering Department

Vision/Mission

Vision:

Mission:

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	
CO-2	
CO-3	
CO-4	
CO-5	

7. Assignment Index

Sr. No	Assignment	Date	Page No.	Sign
1	Assignment 1			
2	Assignment 2			
3	Assignment 3			
4	Assignment 4			
5	Assignment 5			

Assignment Index

Sr No	Assignment	Date	Page No	Sign
1	Assignment - 1		1	✓
2	Assignment - 2		4	✓
3	Assignment - 3		14	✓
4	Assignment - 4		19	
5	Assignment - 5		24	

8. Practical Index

Sr. No	Assignment	Date	Page No.	Sign
1	Practical 1			
2	Practical 2			
3	Practical 3			
4	Practical 4			
5	Practical 5			
6	Practical 6			
7	Practical 7			
8	Practical 8			
9	Practical 9			
10	Practical 10			
11	Practical 11			

9. Assignment 1

CO1: Solve the given problem using fundamentals of Number systems and Boolean algebra

Module 1

1. State and explain De Morgan's theorems with truth tables.
2. Simplify Boolean Function: $F = A'B'C + A'BC + AB'$.
3. List & explain logic family.
4. Describe error detecting & correcting code.
5. Differentiate TTL, Schottky TTL, CMOS

DE

Assignment - I

module 1

(1)

→ De morgan suggested two theorems that form an important part of Boolean algebra.

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

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

Truth Table:-

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

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

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

Truth Table:-

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

(2)

$$\begin{aligned} F &= \overline{ABC} + \overline{ACB} + A\overline{B} \\ &= \overline{AC} (\overline{B} + B) + A\overline{B} \\ &= \overline{AC} + A\overline{B} \quad [A + \overline{A} = 1] \end{aligned}$$

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

3.

→ logic families are divided in two parts.

(1) BiPolar :-

a) Saturated:

- (i) Resistor Transistor logic (RTL)
- (ii) Diode Transistor logic (DTL)
- (iii) Direct Coupled Transistor logic (DCTL)
- (iv) Integrated Injection logic (I^2L)
- (v) High Threshold logic (HTL)
- (vi) Transistor Transistor logic (TTL)

b) Unsaturated:

- (i) Schottky TTL
- (ii) Emitters Coupled logic (ECL)

(2) UniPolar:

- (i) P-Channel mosFET (PMOS)
- (ii) n-Channel mosFET (NMOS)
- (iii) Complementary mosFET (CMOS)

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

Transistor Transistor logic (TTL) :- It is a named for its dependence on transistors 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 Shottky.

Cmos :- 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 for logic functions.

④ Describe error detecting & 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 & some times correction of errors in data. The data along with the extra /bit bits forms the codes. Codes which allow only error detection are called error detecting codes & codes which allow error detection and correction are called error detecting & correcting code.

→ Parity Code:

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

Binary Code	even Parity bit	Even Parity Code
000	0	0000
001	1	0011
010	1	0101
011	0	0110
100	1	1001
101	0	1010
110	0	1100
111	1	1111

Odd Parity Code:- The values of odd parity bit should be zero, if odd number of ones present in the binary code. otherwise, should be one. so, odd number of ones present in odd parity code.

Binary Code	odd Parity bit	odd Parity code
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 error present in the received data. This code uses multiple Parity bits and we have to place these Parity bits in the position of powers of 2.

Bit designation	D ₇	D ₆	D ₅	P ₄	D ₃	P ₂	P ₁
Bit location	7	6	5	4	3	2	1
Binary Location number	111	110	101	100	011	010	001

→ Parity P₁ checks bit locations 1, 3, 5, 7 and assigns P₁ according to even or odd Parity.

→ Parity P₂ checks bit locations 2, 3, 6, 7 and assigns P₂ according to even or odd Parity.

→ Parity P₄ checks bit locations 4, 5, 6, 7 and assigns P₄ according to even or odd Parity.

5. Differentiate TTL, Schottky TTL and CMOS.

→ Parameters	CMOS	TTL	Schottky TTL
Device used	n-channel & p-channel mosFET	Bipolar junction transistor	Schottky diode
V _{IH} (min)	3.5 V	2V	2V
V _{IL} (max)	1.5 V	0.8 V	0.8 V
V _{OH} (min)	4.95 V	2.7 V	2.7 V
V _{OL} (max)	0.005 V	0.4 V	0.5 V
Power dissipation per gate	0.1 mW	10 mW	1 mW
Fan out	50	10	50
Application	Portable instrument where battery supply is used	Laboratory instruments	Voltage clamping applications to prevent transistors saturation.

10. Assignment 2

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

Module 2

1. Explain K map.
2. Obtain the simplified expressions in sum of products for the following Boolean Functions:
 - 2.1. $F(x,y,z) = \Sigma (2,3,6,7)$
 - 2.2. $F(A,B,C,D) = \Sigma (4,6,7,15)$
3. Describe adder & subtractor.
4. Explain multiplexer & demultiplexer.
5. Describe parity checker & generator

Assignment - 2

module 2.

1.

- 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 containing $2^3 = 8$ cells and forth.

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

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

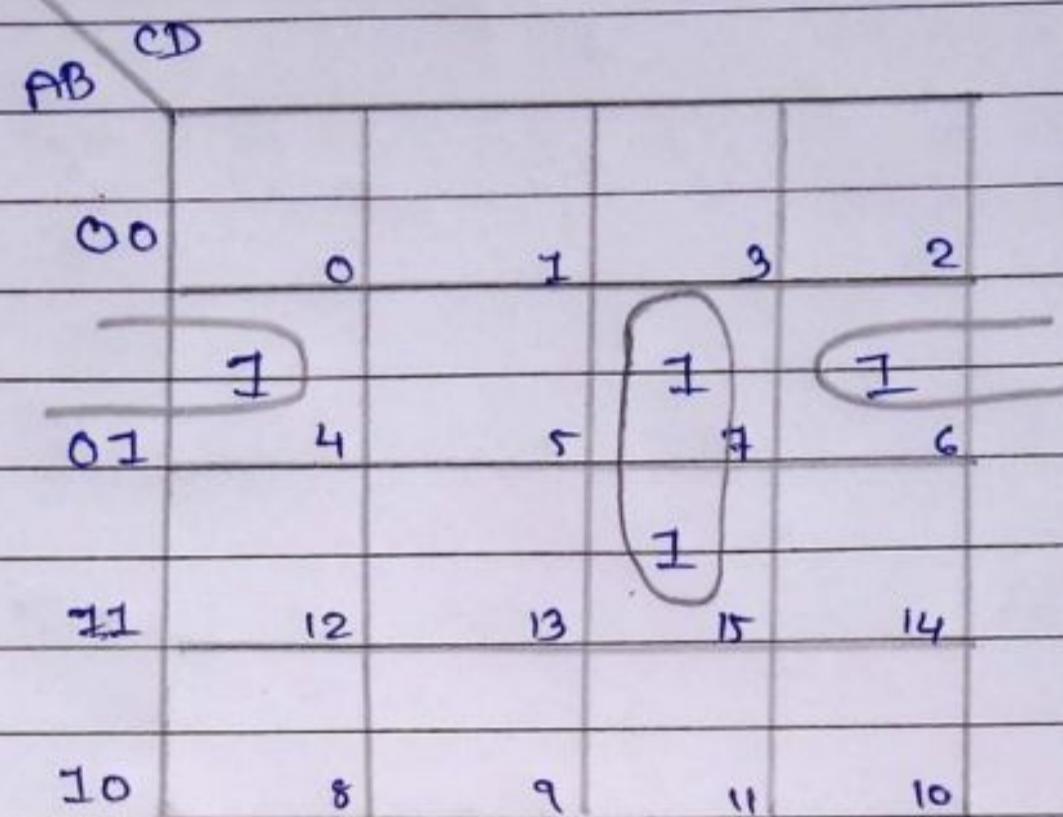
2.

$$\rightarrow 1. F(x, y, z) = \Sigma(2, 3, 6, 7)$$

Σ	xz	00	01	11	10
0			1	1	
1		0	1	3	2
		4	5	7	6

$$F = \Sigma(2, 3, 6, 7)$$

$$2. F(A, B, C, D) = \Sigma(4, 6, 7, 15)$$



$$F = \overline{A}B\overline{D} + BCD$$

3.

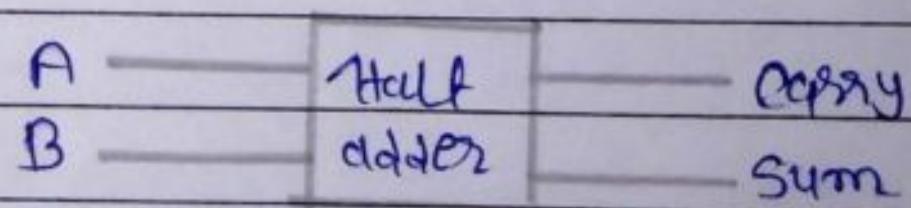
→ Adder:-

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

Half adder:-

- Two binary inputs: a send and added bits,
- Two binary outputs: Sum and Carry.

Block diagram:-



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

Full adder:-

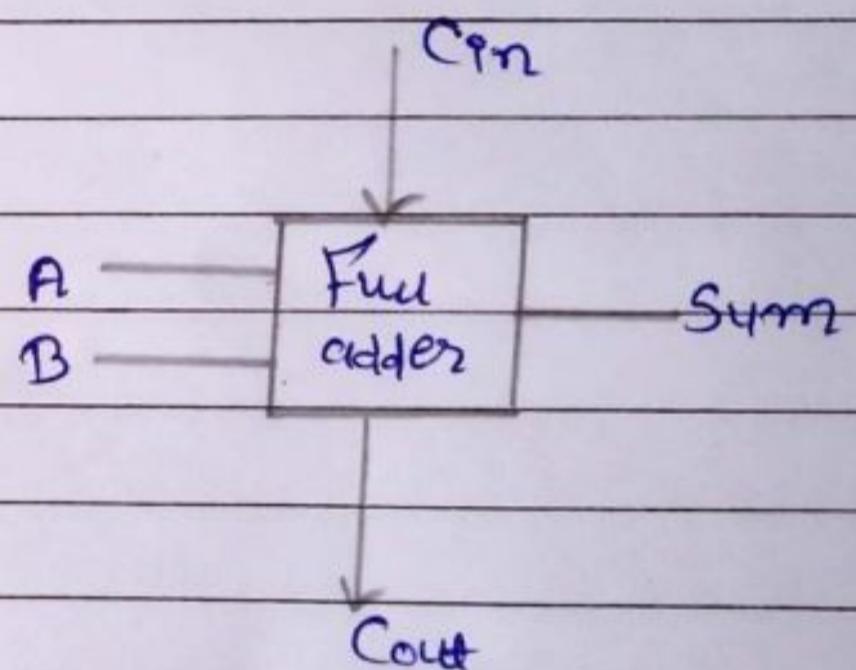
→ Three binary inputs (augend, addend, and, previous carry).

→ Two binary output: 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.



Subtractor:-

→ A half Subtractor is a Combinational circuit that subtracts two-bits and produces their difference.

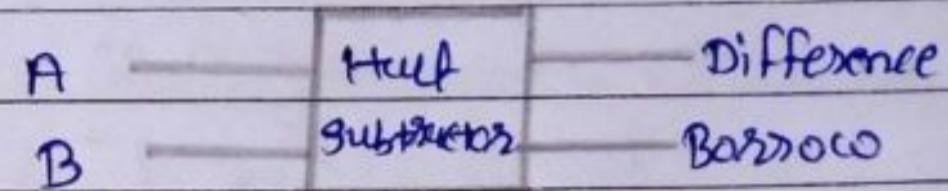
→ A full subtractor is a Combinational circuit that subtracts between two bits, taking into account borrow of lower stage.

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



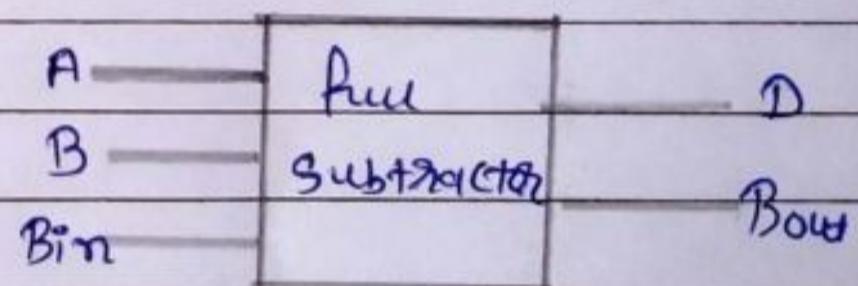
Full Subtractor:

→ Three inputs: minuend, subtrahend and borrowin

→ 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



4.

→ Multiplexer:

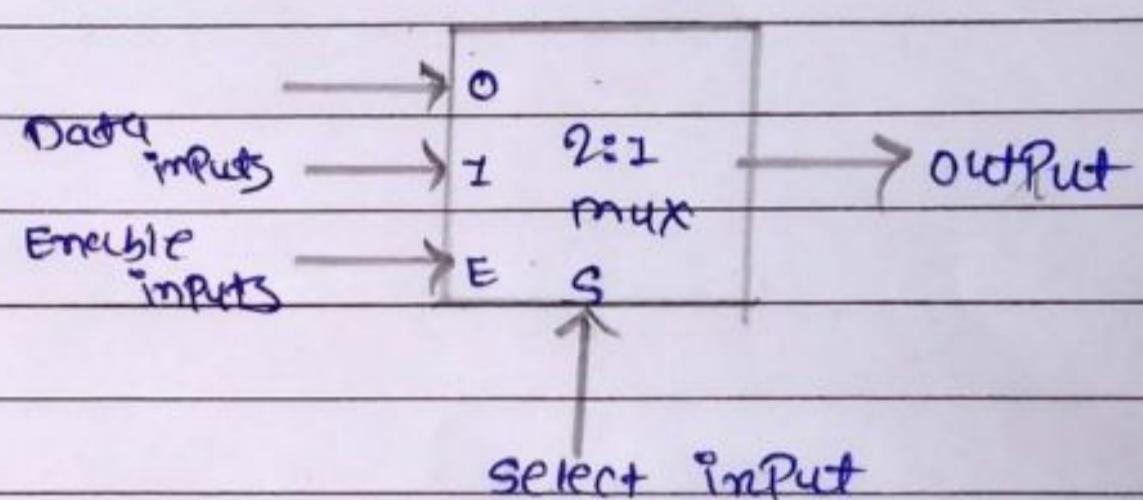
To select single data line from several data input lines, and the data from the selected data line should be available on the output. The digital circuit which does this task is a multiplexer.

2:1 Multiplexer.

- D_0 is applied as an input to one AND gate and D_1 is applied as an input to another AND gate.
- Enable input is applied to both gates as one input.
- Selection line is connected as second input to second AND gate. An inverted S is to second input to first AND gate.
- Outputs of both AND gate applied as to inputs OR gate.

E	S	Y
I	0	D_0
I	1	D_1
0	X	0

Block Diagram.



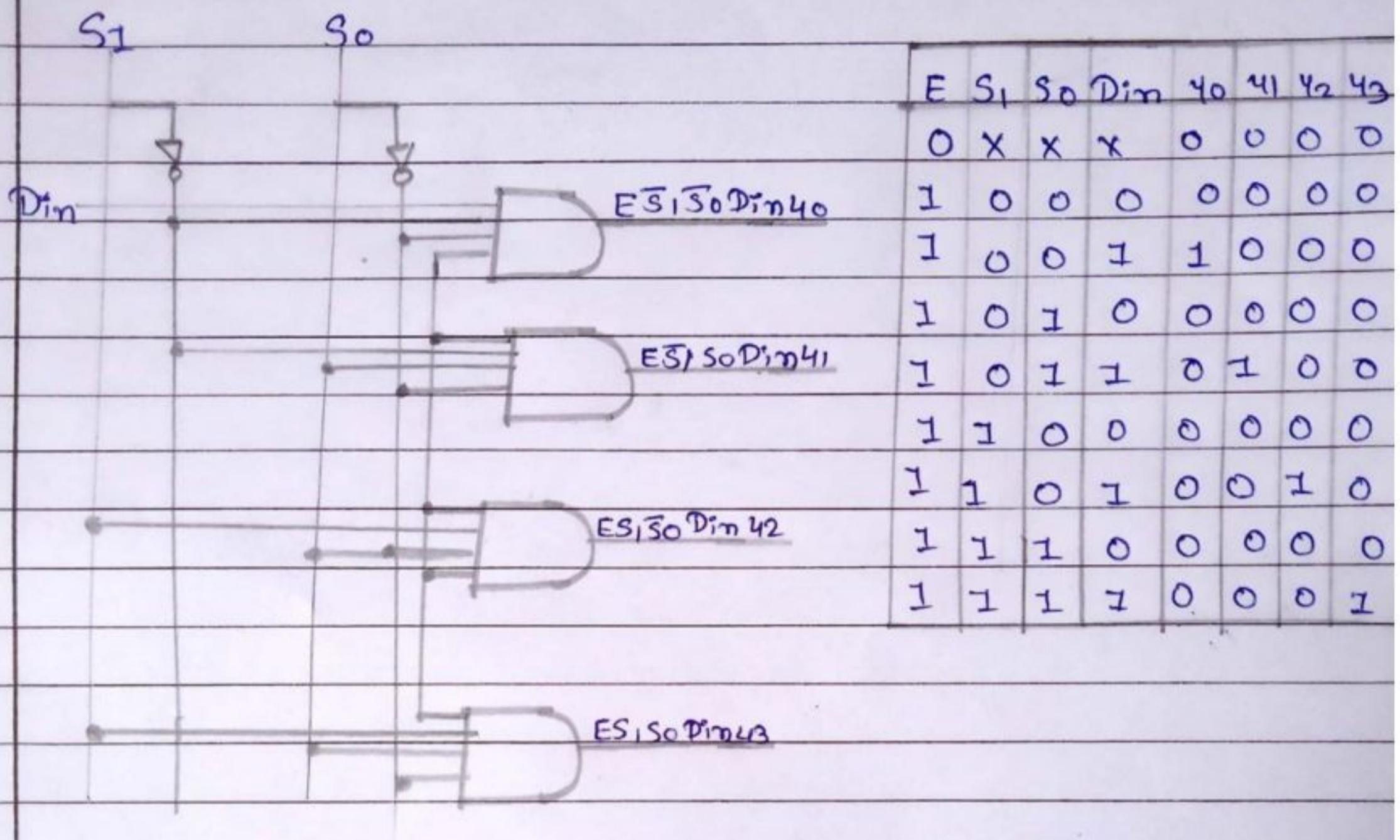
* Demultiplexers:

- 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 Demultiplexers:-

- The single input variable D_{in} 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 demultiplexers.

Logic Diagram



(5)

- A Parity bit is used for the purpose of detecting errors during transmission of binary information.
- The circuit that generates the Parity bit in the transmission is called a Parity generator and the circuit that checks the Parity in the receiver is called Parity checked.
- * Parity generates Truth table for even and odd 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 the Parity bit are transmitted to their destination , where they are applied to the Parity checker circuit

→ The Parity checker circuits checks for possible error in the transmission.

Decimal equivalent	P	A	B	C	P.E.C
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	1	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

11. Assignment 3

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

Module 3

1. Differentiate sequential & combination circuits.
2. List & explain flip flops.
3. List & explain registers.
4. Describe ring counter.
5. Describe how to design counters using flip flops.

Assignment-3

module-3

1.

Combinational circuits

Sequential circuits

- | | |
|---|---|
| 1. In Combinational circuits, the output variables are all dependent only on the present input variables. | 1. In sequential circuits, 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. |

2.

→ Flip flops are

- (1) JK flip flop
- (2) D flip flop
- (3) T flip flop
- (4) SR flip flop.

1) JK flip flop-

→ The data inputs are J and K which are ANDed with \bar{Q} and Q respectively, to obtain S and R input.

J	K	Q_{n+1}
0	0	Q_n
0	1	0
1	0	1
1	1	\bar{Q}_n

2) D flip flop-

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

CP	D	Q_{n+1}
T	0	0
T	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

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

4) SR flip-flop

→ The circuit is similar to SR latch except enable signal is replaced by the Clock Pulse (CP) followed by the positive edge detector circuit.

→ The edge detector circuit is a differentiator.

CP	S	R	Q _{n+1}
0	X	X	Q _n
↑	0	0	Hold
↑	0	1	0
↑	1	0	1
↑	1	1	Invalid

(3)

- 1) Buffer Register
- 2) Controlled Buffer Register
- 3) Shift Register

1) Buffer Register

→ Constructed using four D - flip flops. This register is called buffer register.

→ Each D flip flop is triggered with a common negative edge clock pulse.

→ The input bits set up the flip flop for loading.

2) Controlled Buffer Register

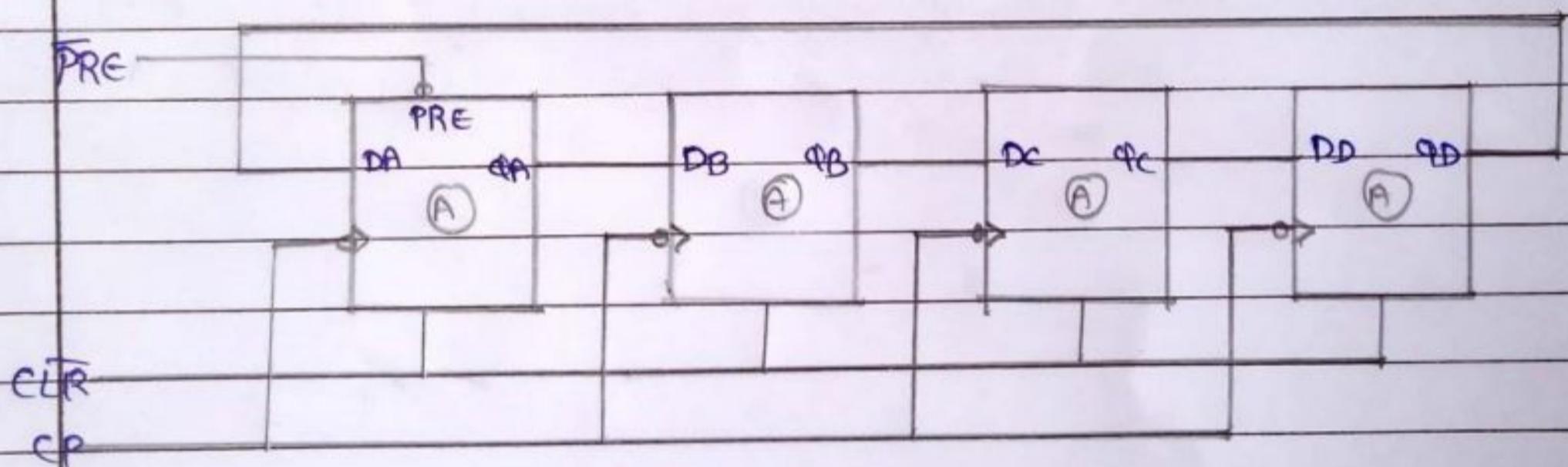
→ We can control input and output of the register by connecting tri-state devices at the input and output sides of register. So this register is called controlled buffer register.

3) Shift Registers

- The binary information in a register can be moved from stage to stage within the register or into or out of the register upon application of clock pulses.
- This type of bit movement or shifting is essential for certain arithmetic and logic operations used in microprocessors. This gives rise to a group of register called shift registers.

(4)

→ Ring Counter



- The Q output of each stage is connected to the D input of the next stage and the output of last stage is fed back to the input of first stage.
- The $\bar{C}LR$ followed by PRE makes the output of first stage to '1' and remaining outputs are zero, 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 outputs are zero.
- The first clock pulse produces $Q_B=1$ & remaining zero.

→ According to the clock pulses applied at the clock input CP, a sequence of four state is produced.

(5)

→ Steps involved in the design of asynchronous counter.

- 1) Determine the number of flip-flops needed.
- 2) Choose the type of flip-flops to be used: T or JK. If T-flip-flops are used connect T input of all flip-flops to logic 1. If JK flip-flops are used connect both J and K inputs of all flip-flops to logic 1. Such connection toggles the flip-flop output on each clock transition.
- 3) Write the truth table for the counter.
- 4) Derive the reset logic by K-map simplification.
- 5) Draw the logic diagram.

12. Assignment 4

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

Module 4

1. Explain weighted resistor/converter.
2. Explain R-2R Ladder D/A converter.
3. Describe specification of A/D & D/A converter.
4. Explain quantization and encoding.
5. Explain parallel comparator A/D converter.

Assignment-4

Module-4

(1)

- Weighted Resistor 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 DEF switch, $I = 0$

- Due to high input impedance of op-amp summing current will flow through R_f . Hence the total current through R_f can be given as

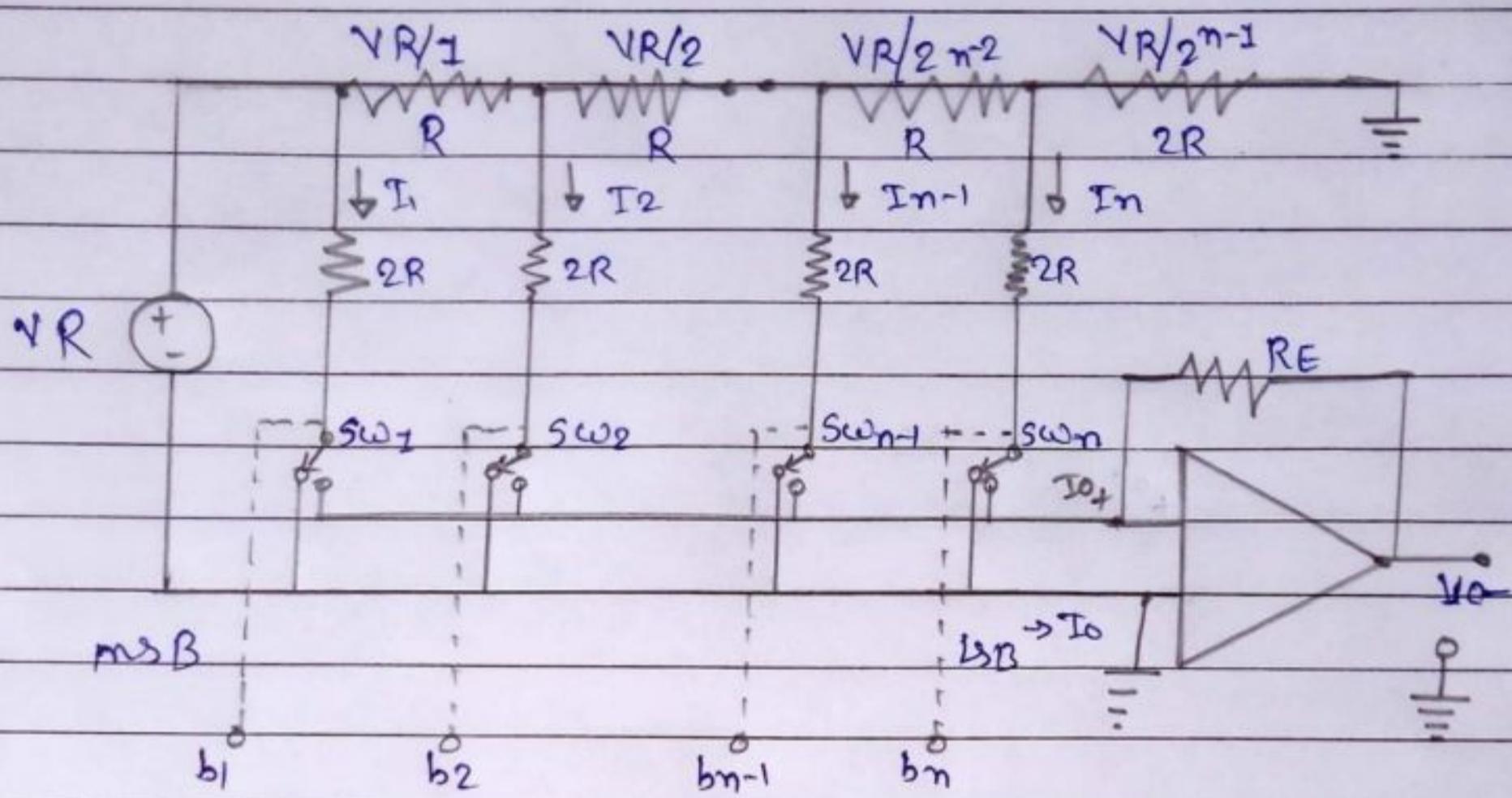
$$I_T = I_1 + I_2 + I_3 + \dots + I_n$$

- When $R_f = R$, V_o is given as

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

(2)

- R/ZR Ladder D/A Converter uses only two resistor values. This avoids resistance spread disadvantages 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/ZR values.



3.

→ Specification of DA converter.

1) Resolution

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

→ % resolution = step size / full scale × 100%.

→ Full Scale = no. of steps × step size

$$\text{Resolution} = 1 / (\text{no. of steps} \times 100\%)$$

2) Accuracy

→ Specified in terms of full-scale errors and linearity errors.

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 successive 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.

Specification of A/D Converter

1) Range of input Voltage

2) Input impedance

3) Accuracy

4) Conversion time

5) Format of digital output.

(1ii)

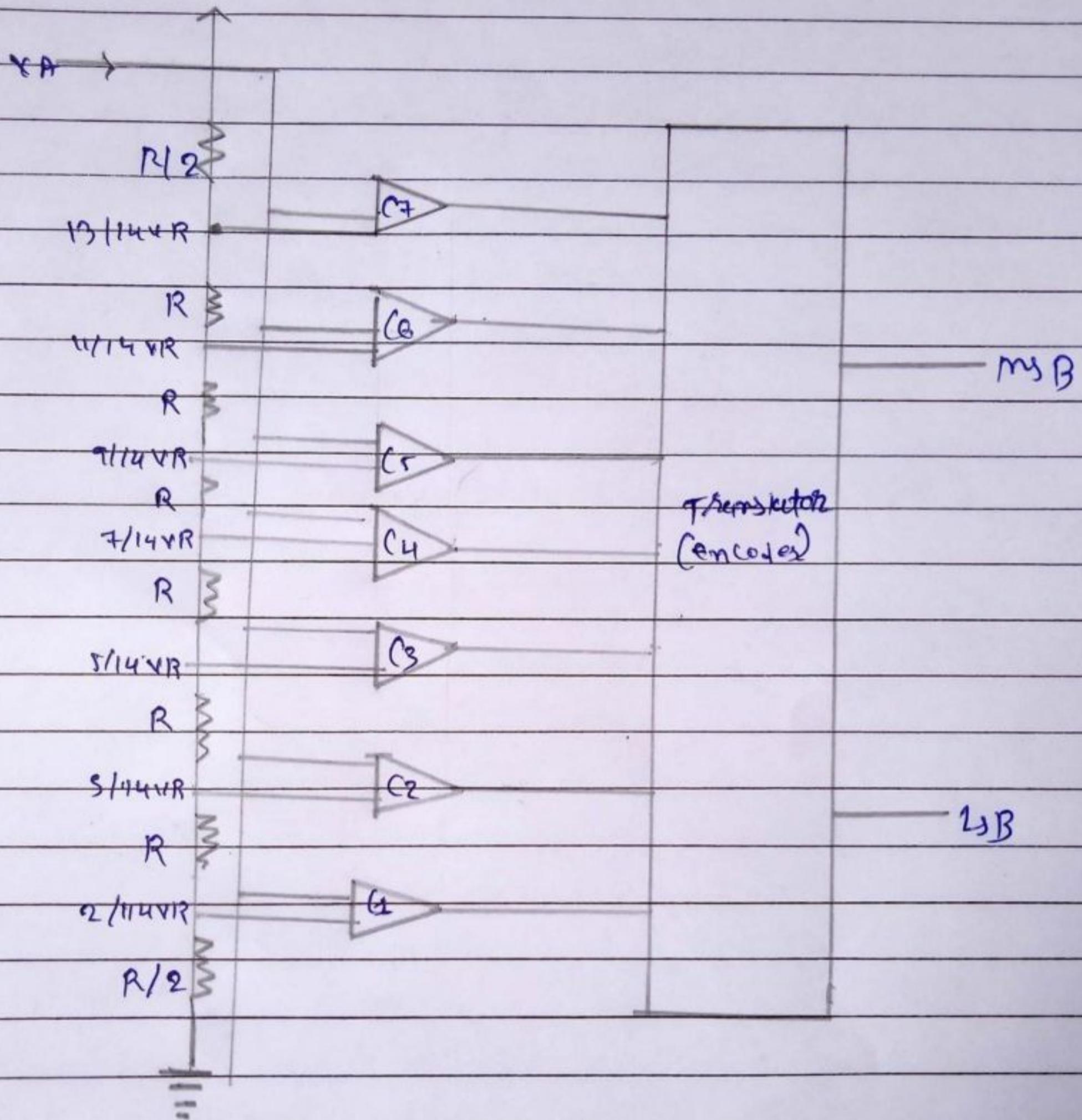
- Quantizing / Encoding: The process of mapping the sampled analog voltage values to discrete voltage levels, which are then represented by binary numbers (bits). This is needed because the analog sample values are real numbers that occur on a continuum.

$$q = \frac{V_{max} - V_{min}}{2^M} = \frac{7 - (-1)}{8} = 0.25V$$

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

5.

- This circuit is formed of a series of Comparators, each one comparing the input signal to a unique reference voltage. The Comparison outputs connects to the input of a Priority encoder circuit, which then produces binary output.
- Based on the principle of comparing analog input voltage with a set of reference voltage.
- To convert the analog input voltage into a 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 of ADC.
- Construction is simple and easier to design.



[3 bit Flash / Parallel Comparators]

13. Assignment 5

CO5: Implement PLDs for the given logical problem.

Module 5

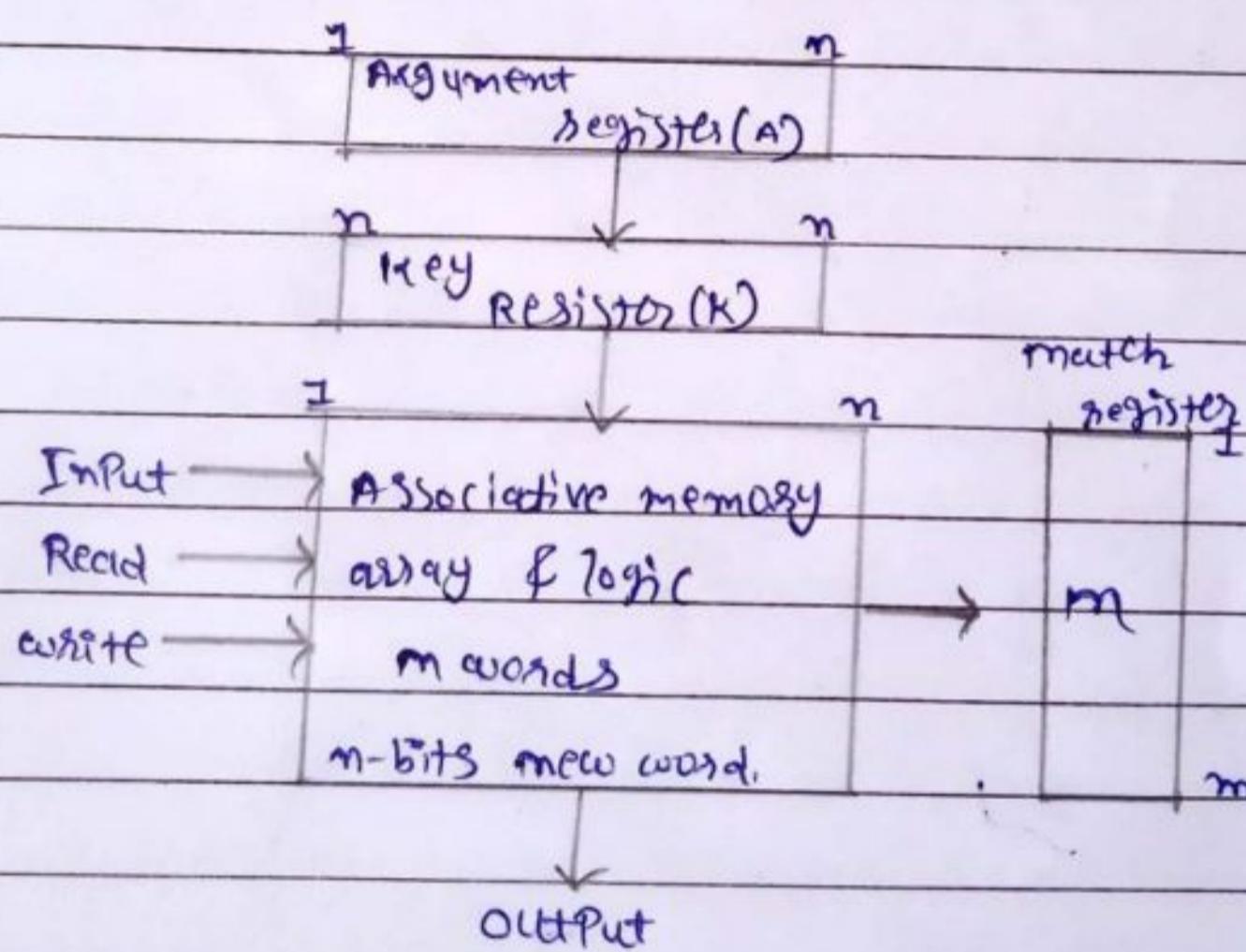
1. Explain content addressable memory (CAM).
2. Explain charge de coupled device memory (CCD).
3. Explain classification of memory.
4. Describe semiconductors.
5. Explain Field Programmable Gate Array (FPGA)

Assignment - 5

Module - 5

Q. 1.

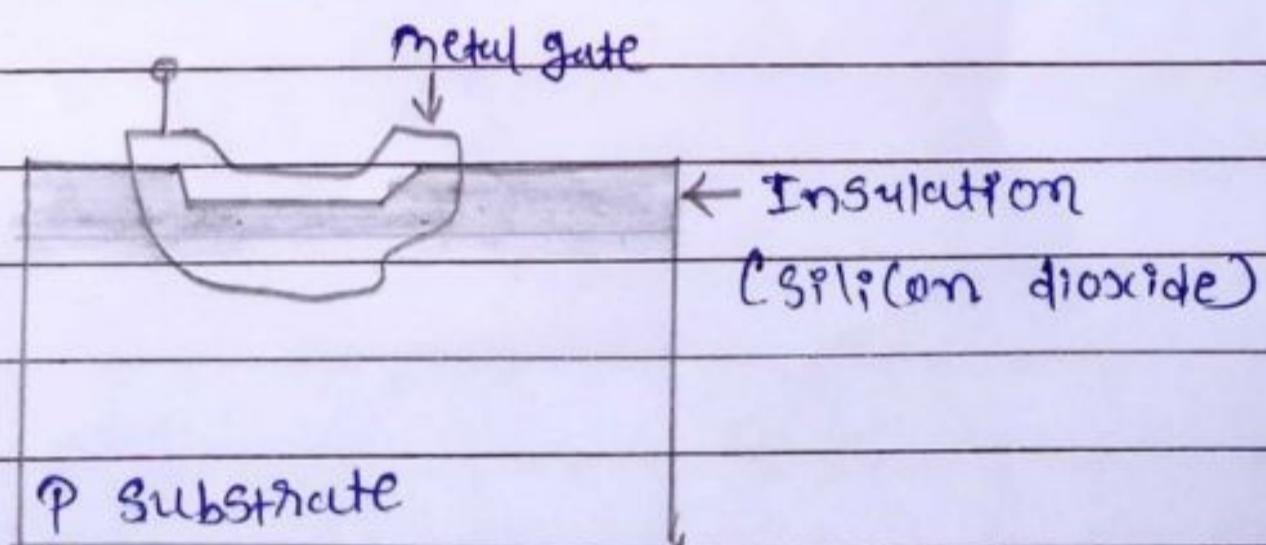
- The time required to find an object stored in memory can be reduced considerably if objects are selected based on their contents, not on their locations. A memory unit accessed by the content is called an associated memory or Content Addressable memory (CAM).
- This type of memory is accessed simultaneously and in parallel on the basis of data content rather than by specific address or location.



(Block diagram of associative memory).

②

- Charge-Coupled device (CCD) memory is a type of dynamic memory in which packets of charge are continuously transferred from one mos device 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 metal gate, holes are p-type substrate. This region, called a potential well, is then capable of accepting a packet of negative charge electrons. Therefore, data is stored in CCD as charge, and it is transferred from one device to an adjacent one by clocking their gates.
- Its simple cell structure. It 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 by rather complex, multi-phase clock signals.



3.

→ Non volatile memory

1) Read only memory (ROM)

i) mask Programmable ROM

ii) Programmable ROM

2) Read/write memory (NVRAM)

i) EEPROM

ii) EEPROM

iii) FLASH

Volatile memory

1) Read/write memory (RAM)

a) Random access

i) SRAM

ii) DRAM

b) Non Randomly 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 (DRAM) stores data as a charge on the capacitors and they need recharging or change on the capacitor after every few millisecond to hold the data even if power is ON.

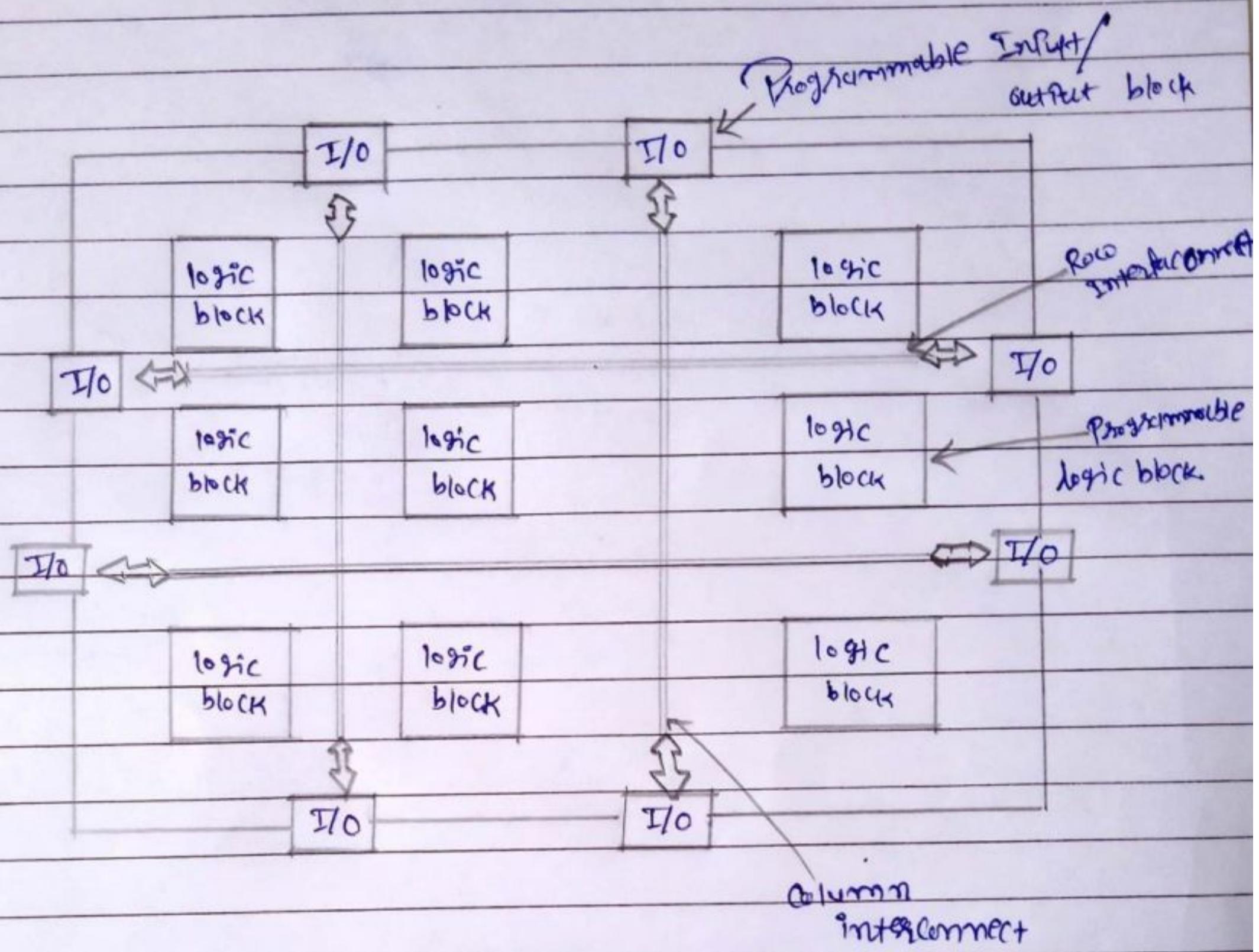
→ EEPROM and EEPROM are erasable memories in which the stored data can be erased and data can be stored.

④.

- A semiconductor is a substance that has specific electrical properties that enable it to serve as a foundation for computers and other electronic devices. It is typically a solid chemical element or compound that conducts electricity under certain conditions but not others.
- Semiconductors are materials which have a 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.

⑤.

- Field Programmable Gate Array (FPGA) provide 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 for a specific functions by the user instead of by the manufacturers of the device.
- The word array is used to indicate a series of column and rows of gates that can be programmed by the end user.
- The programmable logic-blocks of FPGA are called logic blocks or Configurable Logic Blocks (CLBs)



(Basic architecture of FPGA)

9. 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:

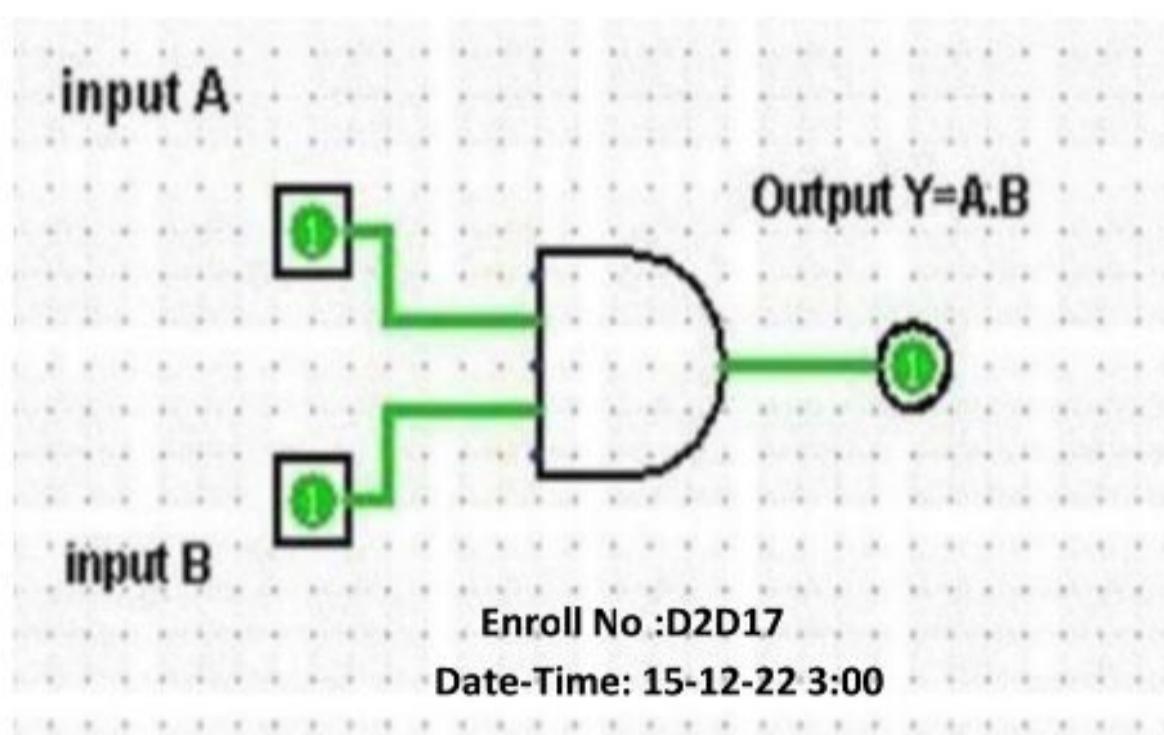
- Logic gates are devices that can combine multiple inputs at independent logic levels and come up with an output accordingly. There are many kinds of logic gates, and the distinction lies in that each kind processes the inputs differently, and may give different outputs for the same inputs.
- There are 3 basic gates (AND, OR, NOT), 2 universal gates(NAND, NOR) and 2 special gates(XOR, XNOR) available for different operations.

I. AND GATE:

- 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. $A \cdot B$.
- The dot is sometimes omitted i.e., AB
- Boolean Expression: $A \cdot B$

Practical No : 1

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



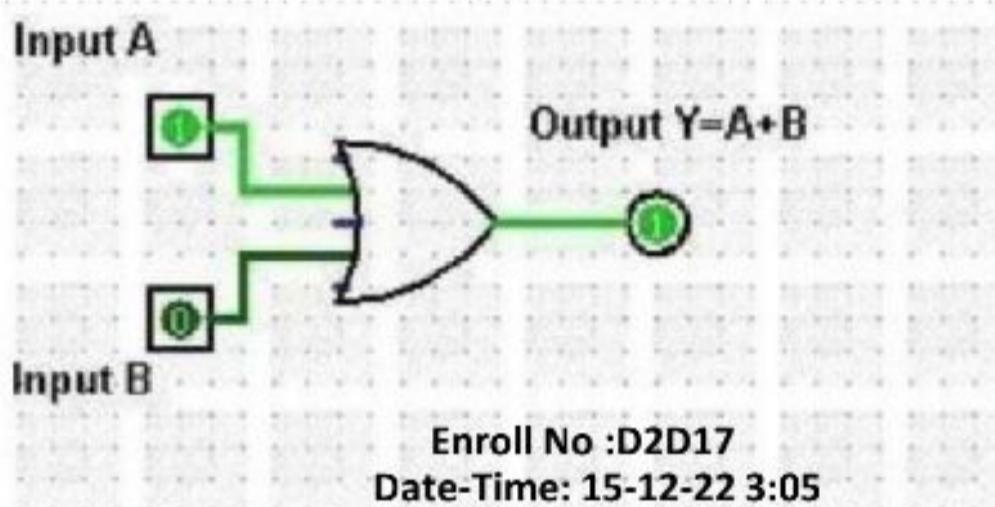
A	B	Y
0	0	0
0	1	0
1	0	0
1	1	1

II. OR Gate:

- 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.
- Boolean Expression: $A+B$

Practical No : 1

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



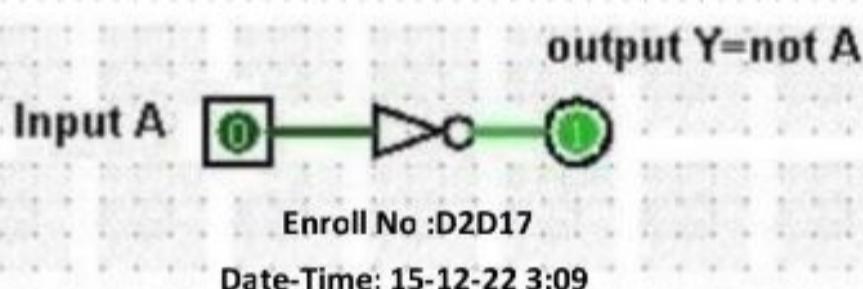
A	B	Y
0	0	0
0	1	1
1	0	1
1	1	1

III. NOT Gate:

- 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
- Boolean Expression: A'

Practical No : 1

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

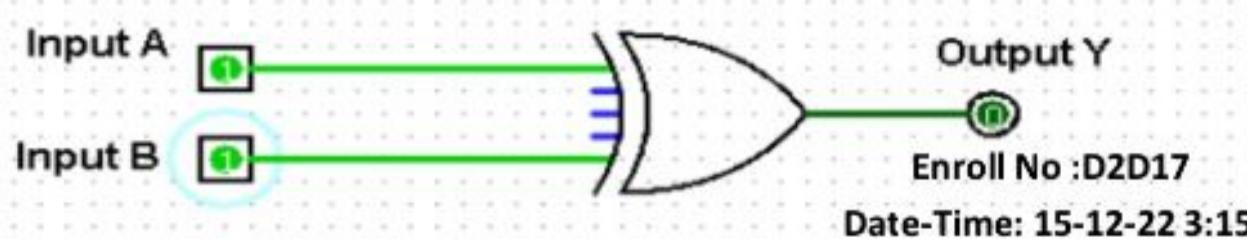


A	Y
0	1
1	0

IV. Exclusive OR Gate :

Practical No : 1

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

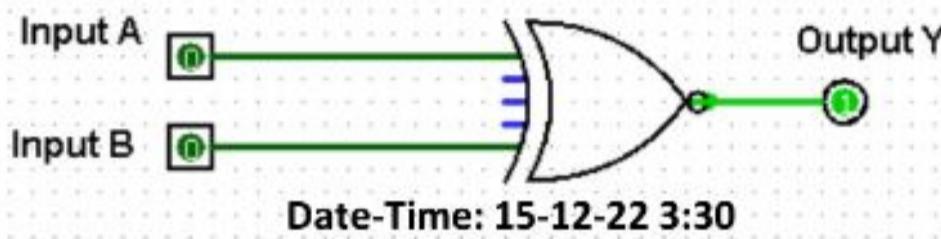


A	B	Y
0	0	0
0	1	1
1	0	1
1	1	0

V. Exclusive NOR Gate:

- 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

Practical No : 1
Enroll No :D2D17
Aim : Analyze working of logic families and logic gates and
design the simple circuits using various gates for a given problem.



Date-Time: 15-12-22 3:30

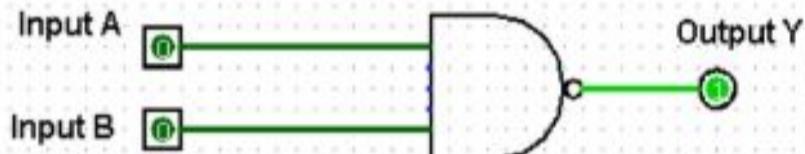
A	B	Y
0	0	1
0	1	0
1	0	0
1	1	1

VI. NAND Gate :

- 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.
- Boolean Expression: $(A \cdot B)'$

Practical No : 1
Aim : Analyze working of logic families and logic gates and
design the simple circuits using various gates for a given problem.

Enroll No :D2D17 Date-Time: 15-12-22

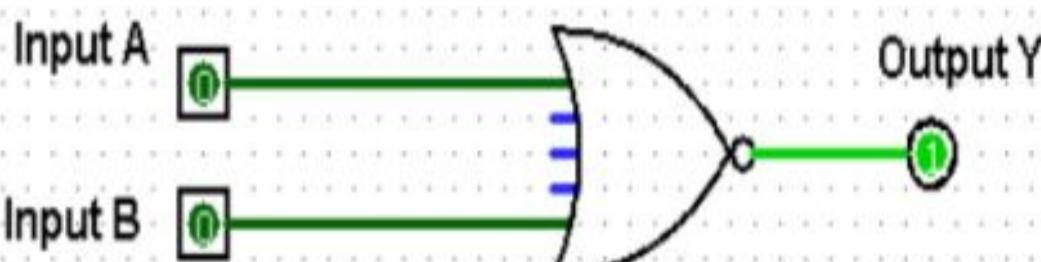


A	B	Y
0	0	1
0	1	1
1	0	1
1	1	0

VII. NOR GATE:

- 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.
- Boolean Expression: $(A + B)'$

Enroll No :D2D17 Date-Time: 15-12-22 4:40

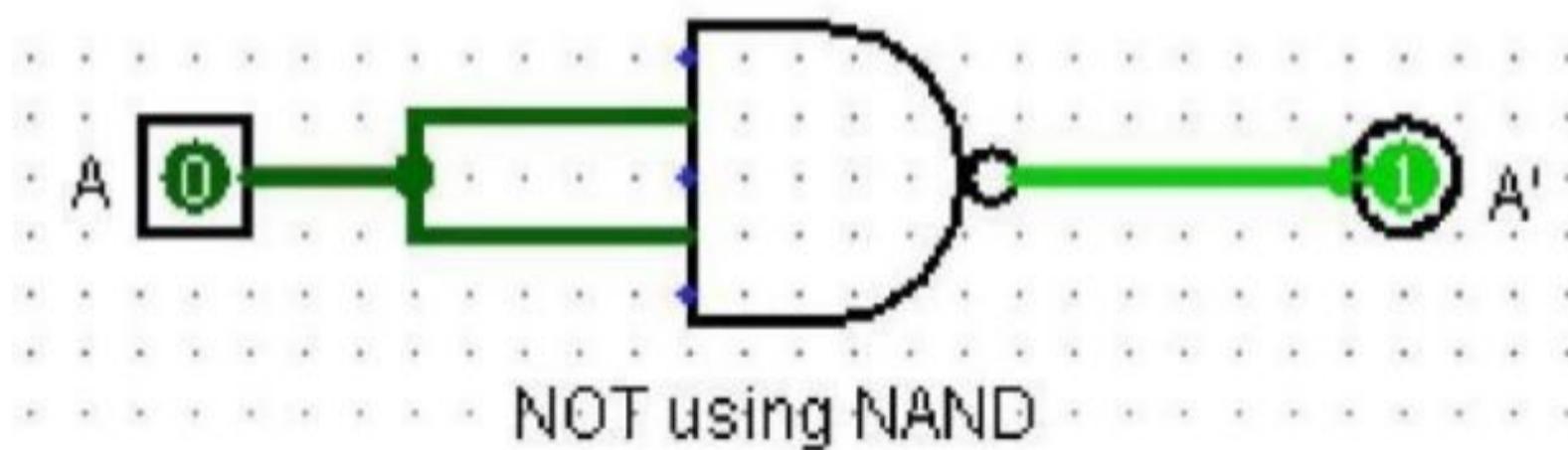


A	B	Y
0	0	1
0	1	1
1	0	1
1	1	0

NAND AS UNIVERSAL

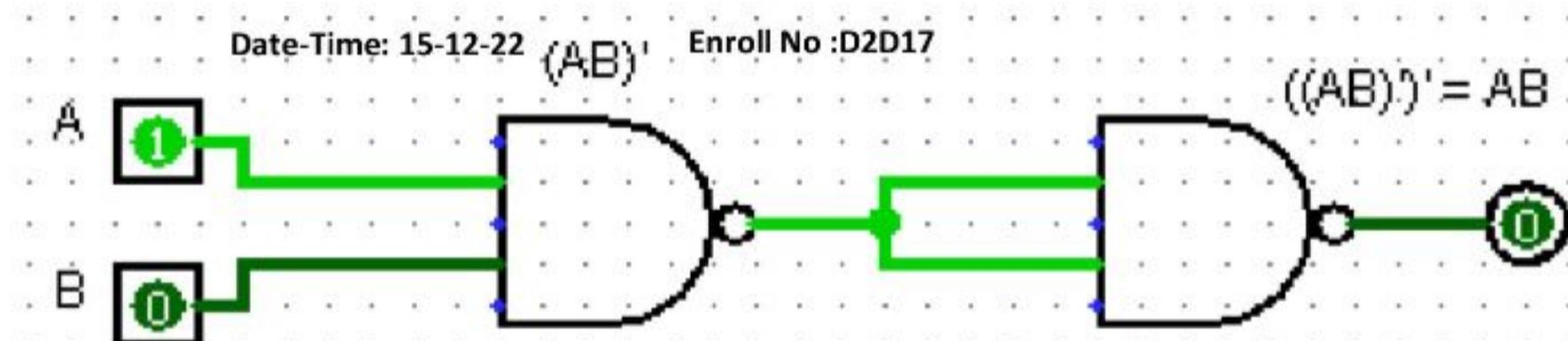
1) NOT using NAND gate :

This is the circuit diagram of a NAND gate used to make work like a NOT gate



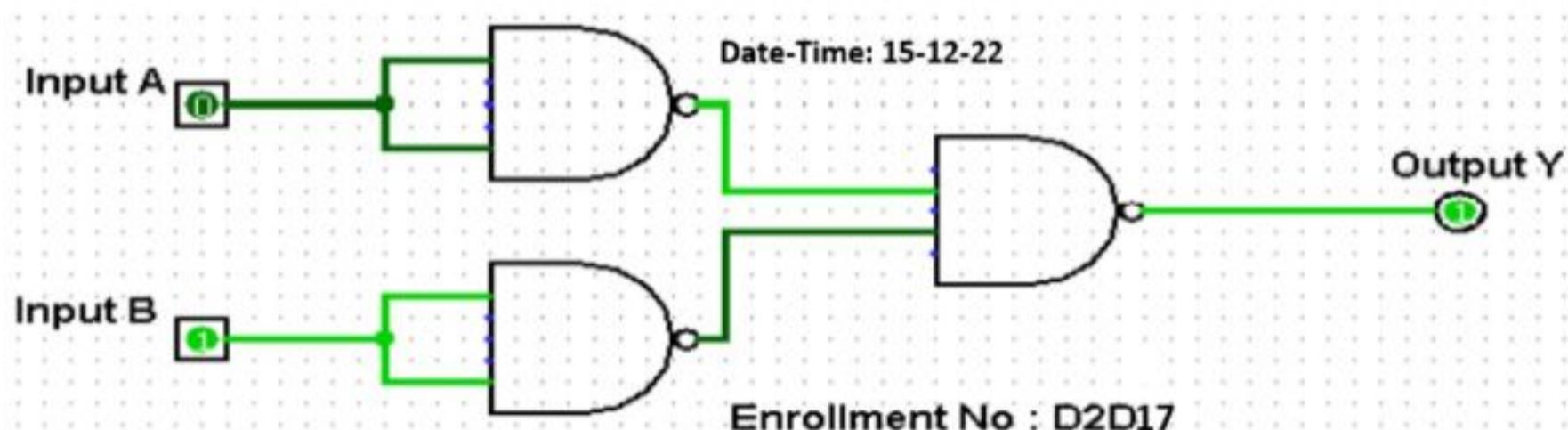
2) AND using NAND gate :

A NAND produces complement of AND gate. So, if the output of a NAND gate is inverted, overall output will be that of an AND gate



3) OR using NAND gate :

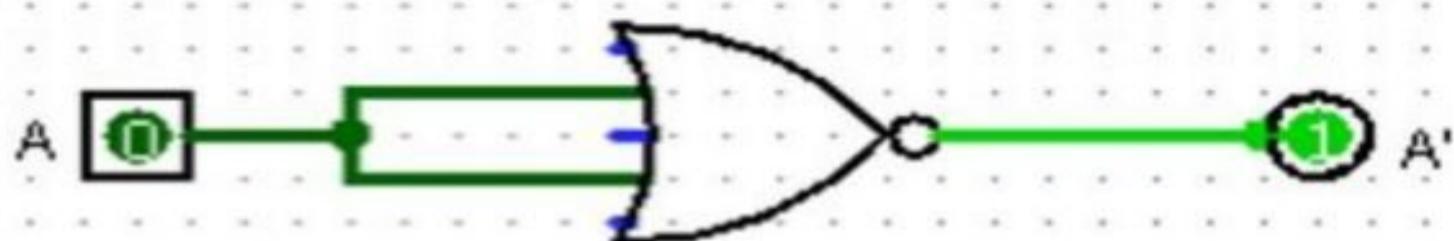
an OR gate made from combinations of NAND gates, arranged in a proper manner.



NOR as Universal

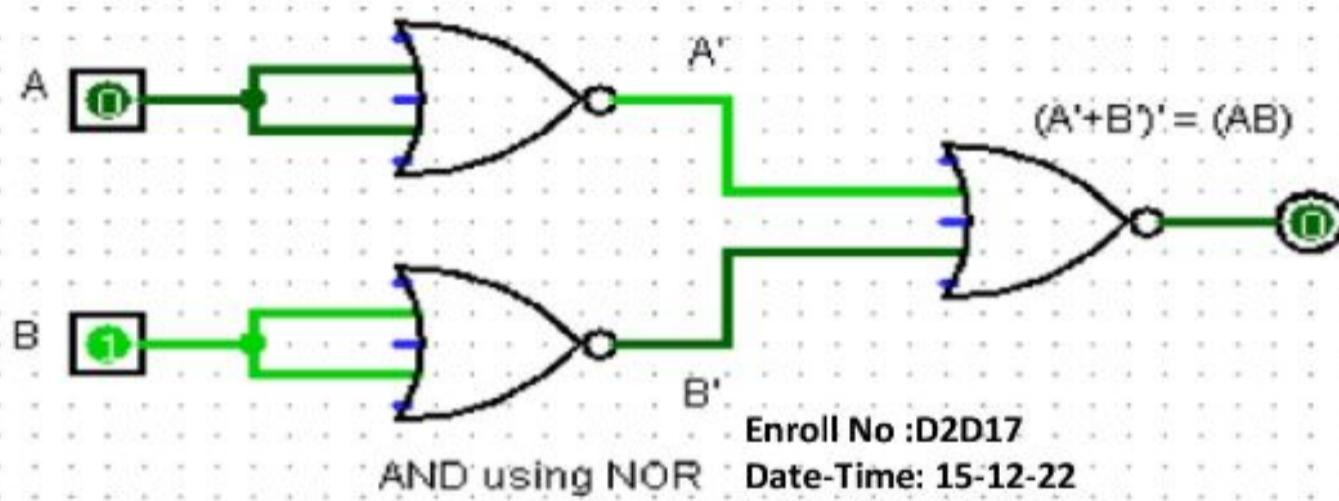
1) NOT using NOR gate :

NOT Logic Gate using NOR Gate. To derive NOT gate using NOR, it has to be considered that both the inputs of NOR gate are connected which is taken as a single input to NOR gate because NOT has a single input. Here, the IC 7402 is used which is a quad 2-input NOR gate.



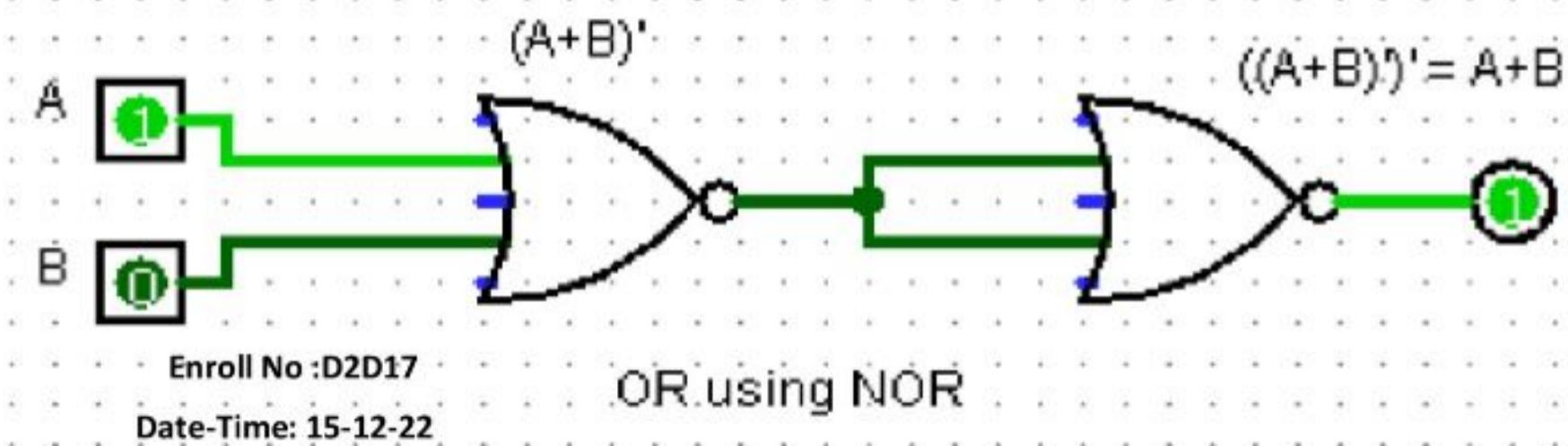
2) AND using NOR gate :

An AND gate gives a 1 output when both inputs are 1; a NOR gate gives a 1 output only when both inputs are 0. Therefore, an AND gate is made by inverting the inputs to a NOR gate.



3) OR using NAND gate :

A NOR produces complement of OR gate. So, if the output of a NOR gate is inverted, overall output will be that of an OR gate.



10. Practical 2

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

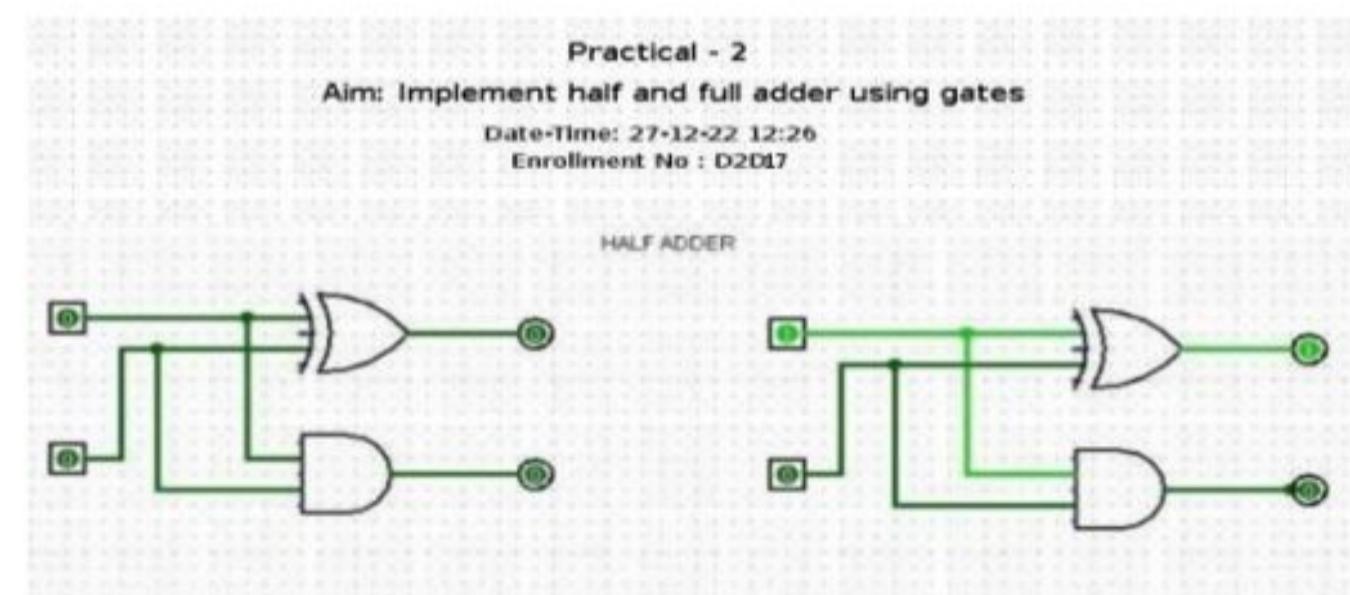
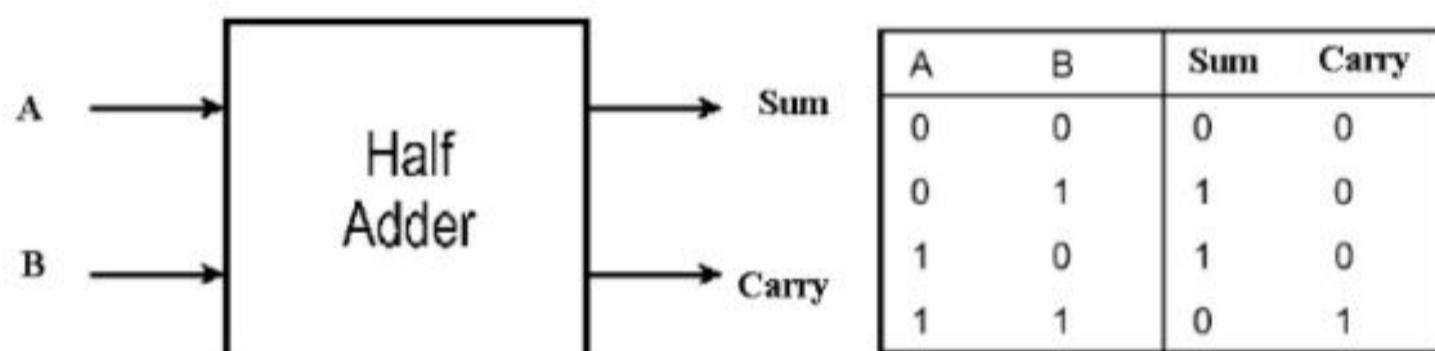
Module 2

Aim: Implement half and full Adders using logic gates.

THEORY:

1) Half Adder :

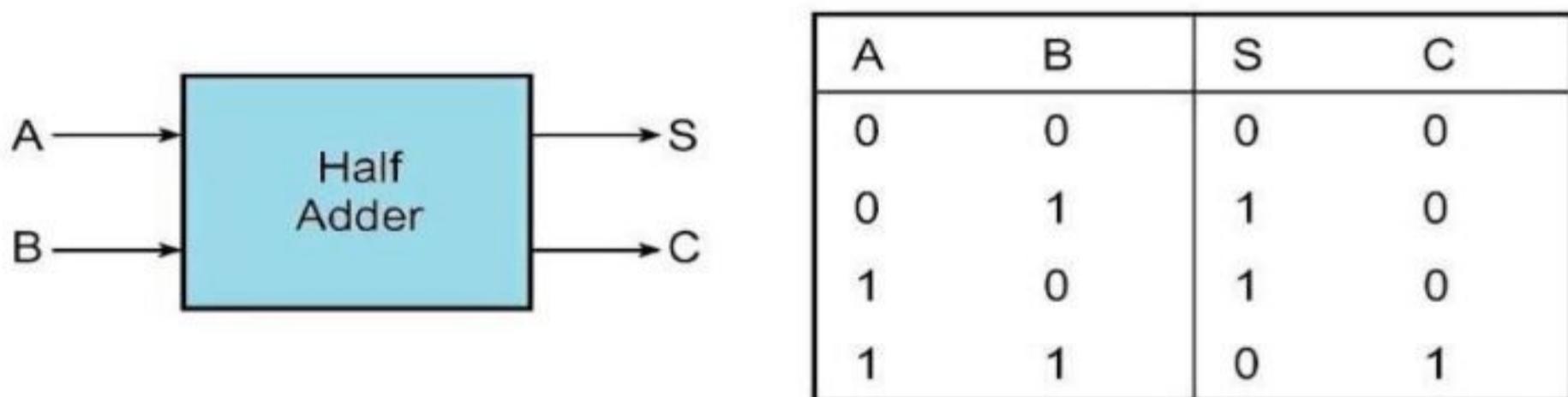
- Half adder is a combinational arithmetic circuit that adds two numbers and produces a sum bit (S) and carry bit (C) as the output.
- If A and B are the input bits, then sum bit (S) is the X-OR of A and B and the carry bit (C) will be the AND of A and B. From this it is clear that half adder circuit can be easily constructed using one X-OR gate and one AND gate.
- 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 has a carry, then it will be neglected 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: SUM S = $X'Y + XY' = X \oplus Y$
CARRY C = $X' \cdot Y$



2) 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, output S is obtained. 24| page 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: $SUM\ S = X \oplus Y \oplus Z$

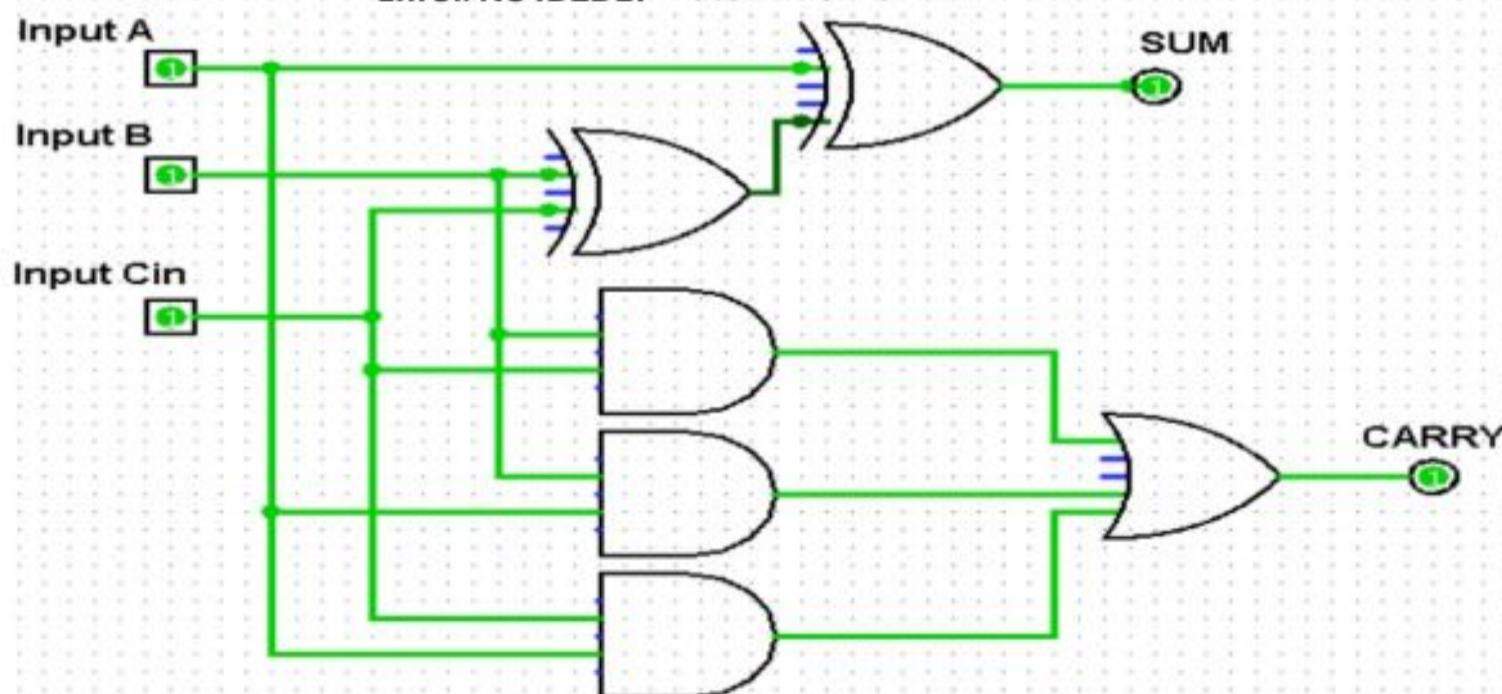
$$CARRY\ C = X'(Y \oplus Z) + YZ$$



Practical No : 2

Aim: Implement half and full Adders using logic gates.

Enroll No :D2D17 Date-Time: 15-12-22



11. Practical 3

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

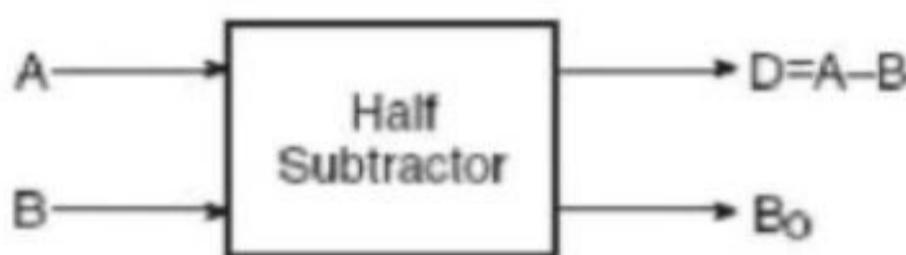
Module 2

Aim: Implement half and full Subtractors using logic gates.

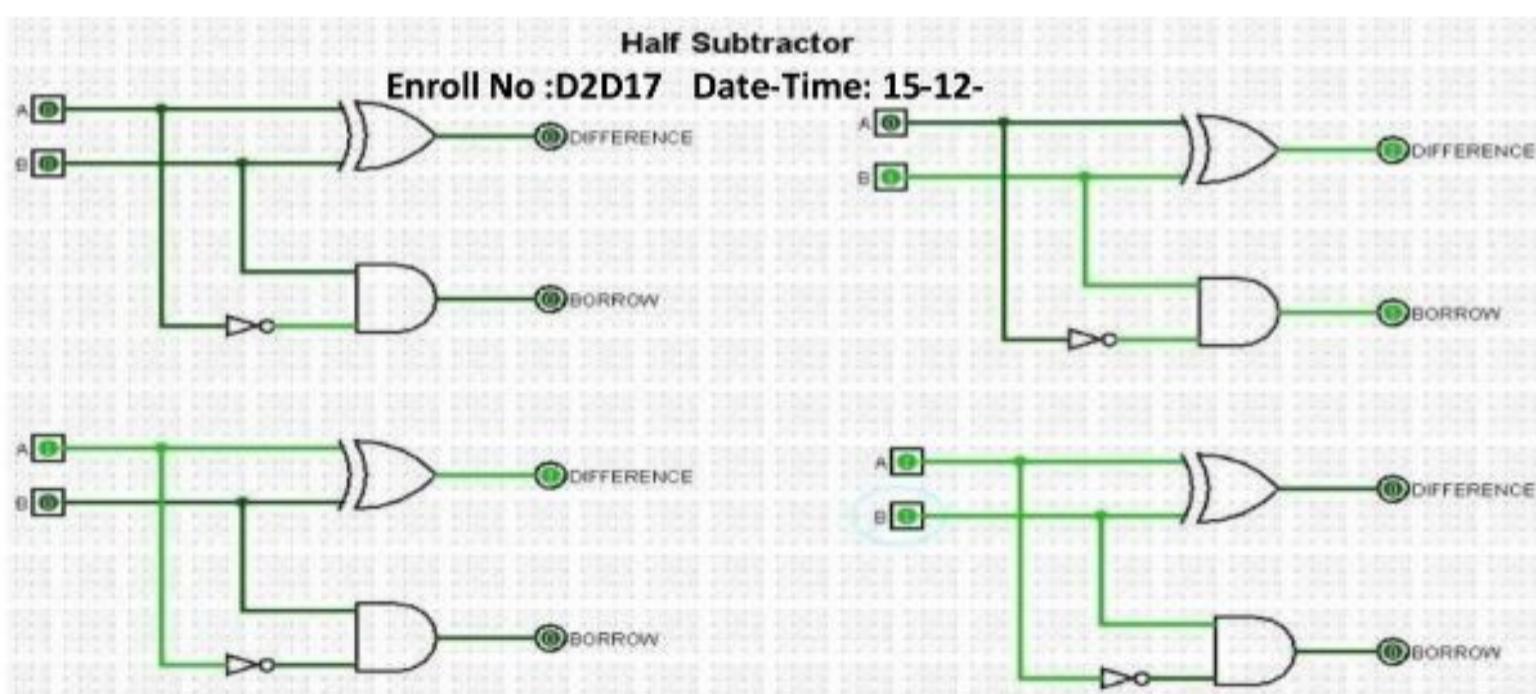
THEORY:

1) Half Subtractor:

- Half Subtractor is a combinational arithmetic circuit that subtracts two numbers and produces a difference bit (D) and borrow bit (B₀) 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 (B₀) will be the AND of X' and Y.
- 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: DIFFERENCE D = X'Y+XY' = X \oplus Y
BORROW B = X' · Y



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



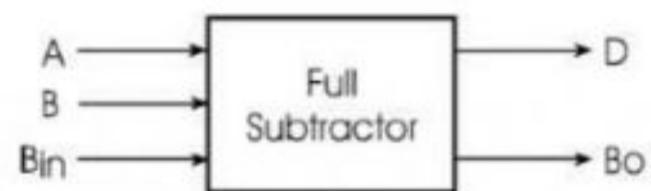
2) Full Subtractor:

- 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 *Bout* 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 $D = X \oplus Y \oplus Z$

$$\text{BORROW } B = X'(Y \oplus Z) + YZ$$

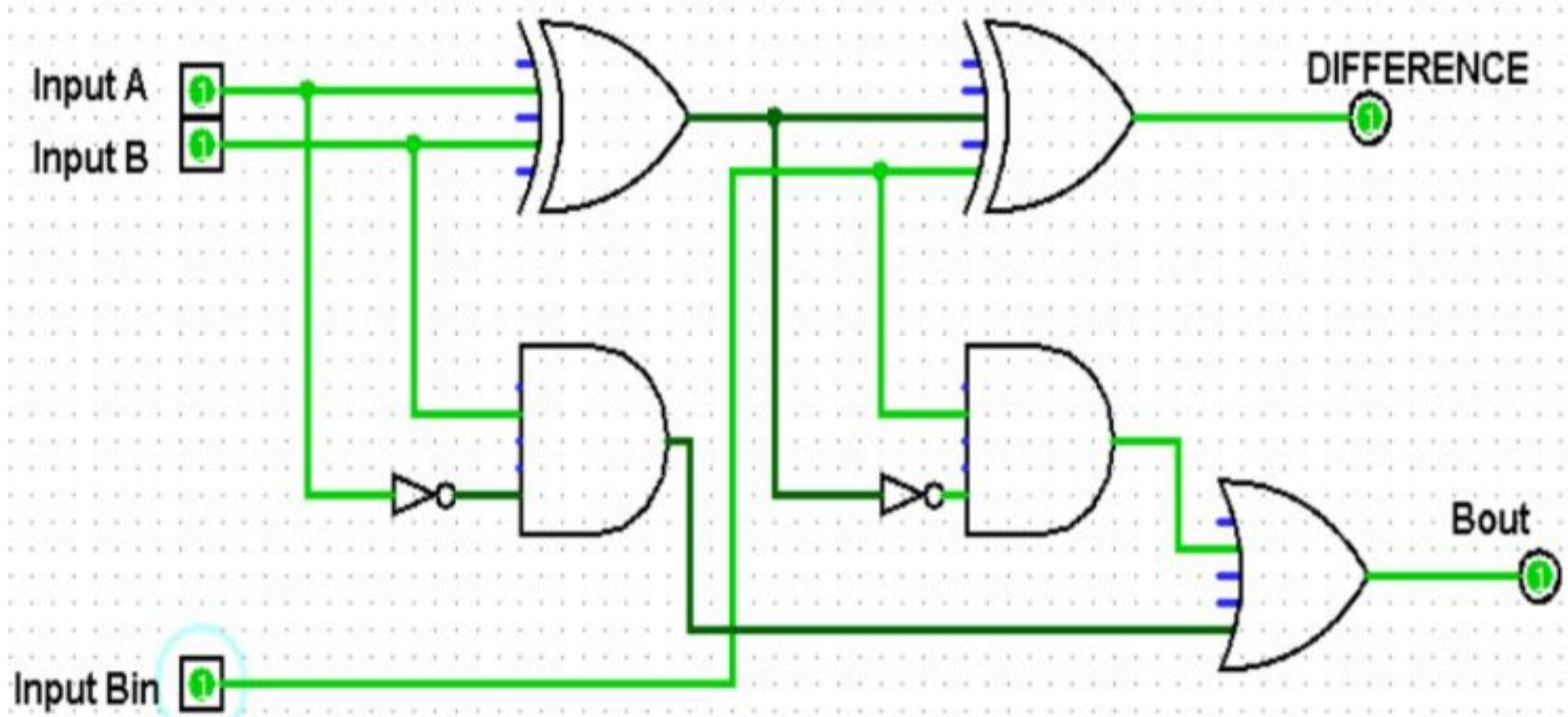
Minuend (A)	Subtrahend (B)	Borrow In (Bin)	Difference (D)	Borrow Out (B_0)
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	0	1
1	0	0	1	0
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1



Practical No : 3

Date-Time: 20-12-22 4:29 Enroll No :D2D17

Aim: Implement half and full Subtractors using logic gates.



12. Practical 4

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

Module 2

Aim: Perform Parity Checker.

Theory:

1.) Parity Generator:

- The parity generator is a combination circuit at the transmitter, it takes an original message as input and generates the parity bit for that message and the transmitter in this generator transmits messages along with its parity bit. There are two types of Parity Generator Even parity generator and Odd parity generator

Even Parity Generator:

- The even parity generator maintains the binary data in even number of 1's, for example, the data taken is in odd number of 1's, this even parity generator is going to maintain the data as even number of 1's by adding the extra 1 to the odd number of 1's. This is also a combinational circuit whose output is dependent upon the given input data, which means the input data is binary data or binary code given for parity generator.

Odd Parity Generator:

- The odd parity generator maintains the binary data in an odd number of 1's, for example, the data taken is in even number of 1's, this odd parity generator is going to maintain the data as an odd number of 1's by adding the extra 1 to the even number of 1's. This is the combinational circuit whose output is always dependent upon the given input data. If there is an even number of 1's then only parity bit is added to make the binary code into an odd number of 1's.

3-bit message			Odd parity bit generator (P)
A	B	C	Y
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	0

3-bit message			Even parity bit generator (P)
A	B	C	Y
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1

2.) 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:

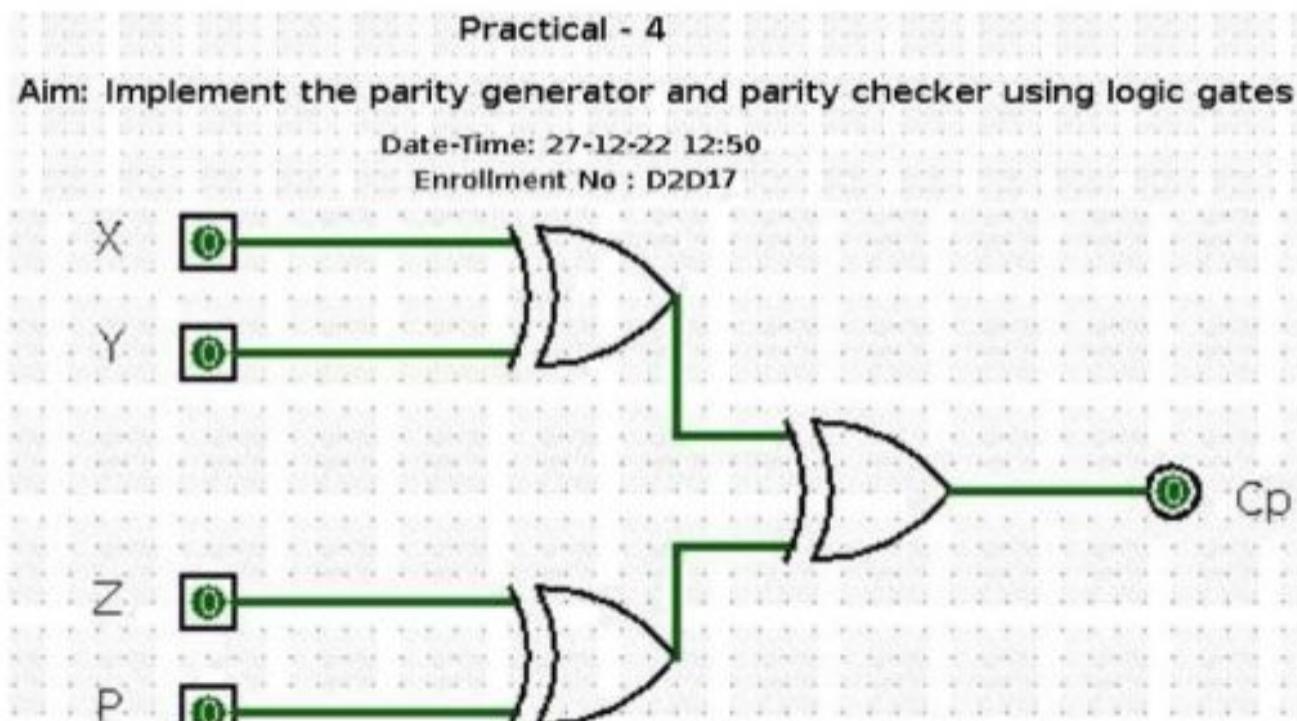
- In even parity checker if the error bit (E) is equal to '1', then we have an error. If error bit E=0 then indicates there is no error. Error Bit (E) =1, error occurs Error Bit (E) =0, no error

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

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

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



13. Practical 5

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

Module 2

Aim: Study and implement Multiplexer and Demultiplexer.

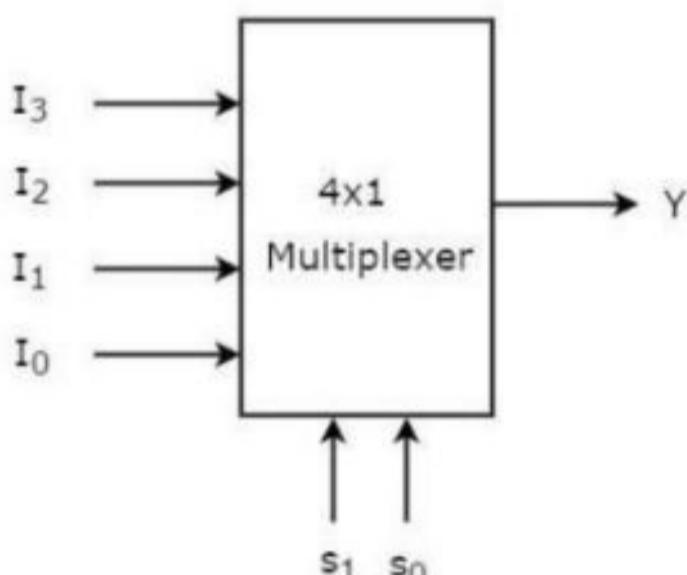
THEORY :

1) Multiplexer:

- A multiplexer is a combinational circuit that has $2n$ input lines and a single output line. Simply, the multiplexer is a multi-input and single-output combinational circuit. The binary information is received from the input lines and directed to the output line. On the basis of the values of the selection lines, one of these data inputs will be connected to the output.
- Unlike encoder and decoder, there are n selection lines and $2n$ input lines. So, there is a total of $2N$ possible combinations of inputs. A multiplexer is also treated as Mux.

4x1 Multiplexer

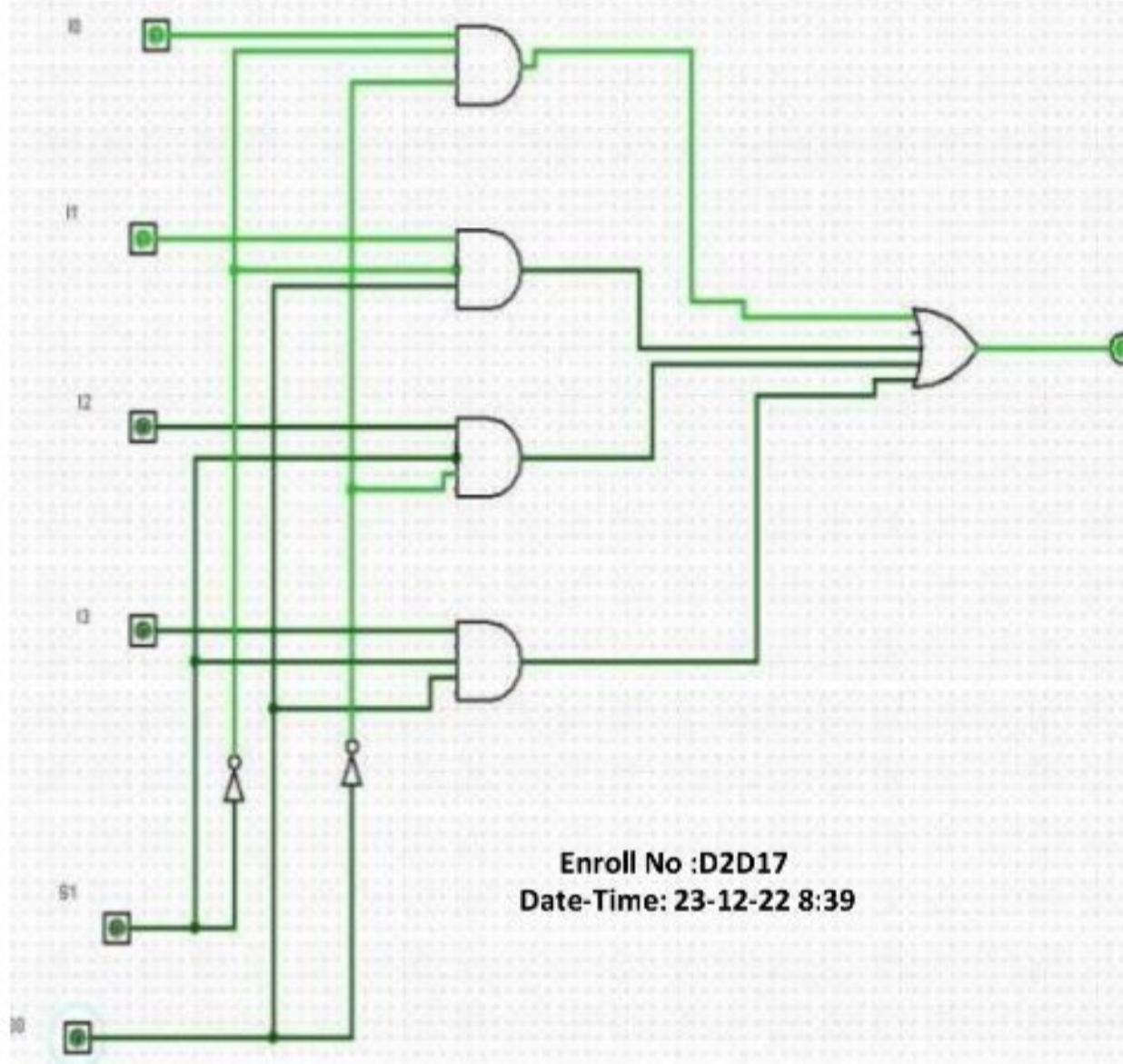
- 4x1 Multiplexer has four data inputs I_3, I_2, I_1 & I_0 , two selection lines s_1 & s_0 and one output Y . One of these 4 inputs will be connected to the output based on the combination of inputs present at these two selection lines.
- Boolean Expression: $Y = S_1'S_0'I_0 + S_1'S_0I_1 + S_1S_0'I_2 + S_1S_0I_3$



Selection Lines		Output
S_1	S_0	Y
0	0	I_0
0	1	I_1
1	0	I_2
1	1	I_3

Practical No : 5

Aim: Study and implement Multiplexer and Demultiplexer.

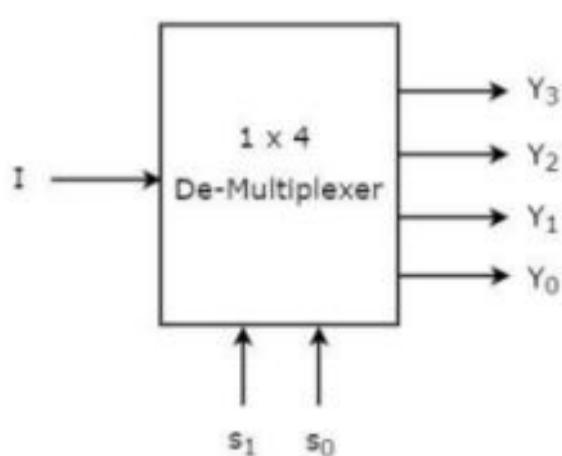


2) De-multiplexer

- A De-multiplexer is a combinational circuit that has only 1 input line and $2N$ output lines. Simply, the multiplexer is a single-input and multi-output combinational circuit. The information is received from the single input lines and directed to the output line. On the basis of the values of the selection lines, the input will be connected to one of these outputs. De-multiplexer is opposite to the multiplexer
- Unlike encoder and decoder, there are n selection lines and $2n$ outputs. So, there is a total of $2n$ possible combinations of inputs. De-multiplexer is also treated as De-mux.

1x4 De-Multiplexer

- 1x4 De-Multiplexer has one input I, two selection lines, s1 & s0 and four outputs Y3, Y2, Y1 & Y0. 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, Y3 to Y0 based on the values of selection lines s1 & s0. The Truth table of 1x4 De-Multiplexer is shown below.



$$Y_3 = s_1 s_0 I \quad Y_3 = s_1 s_0 I$$

$$Y_2 = s_1 s_0' I \quad Y_2 = s_1 s_0' I$$

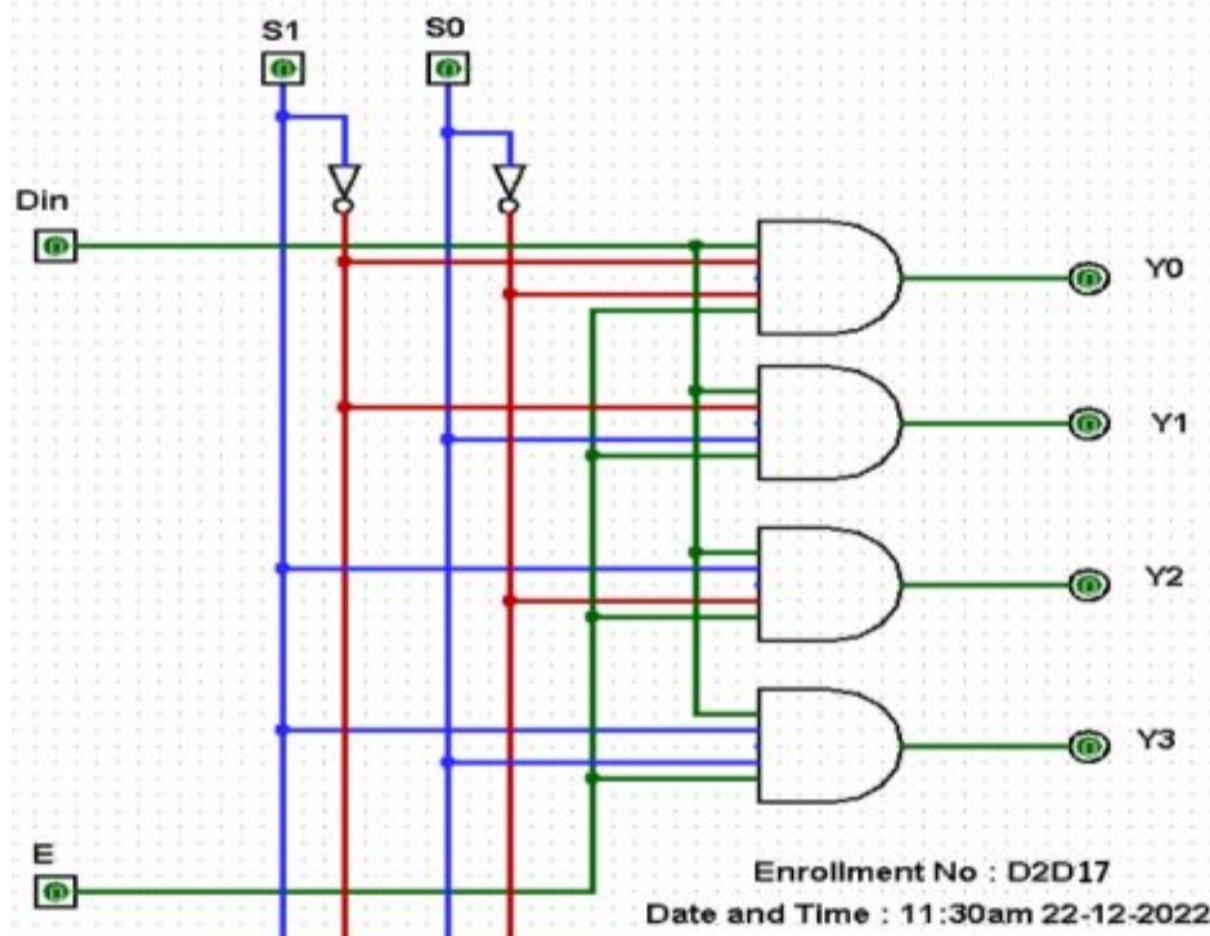
$$Y_1 = s_1's_0 I \quad Y_1 = s_1's_0 I$$

$$Y_0 = s_1's_0' I$$

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

Practical No : 5

Aim: Study and implement Multiplexer and Demultiplexer.



14. 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 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.

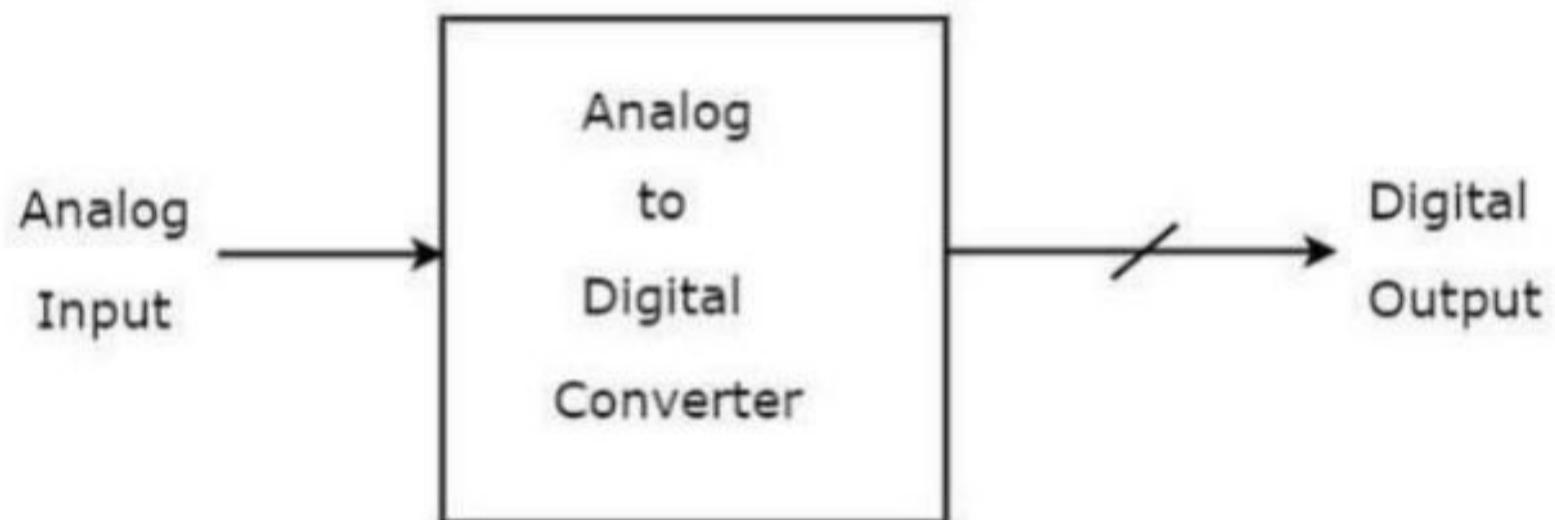
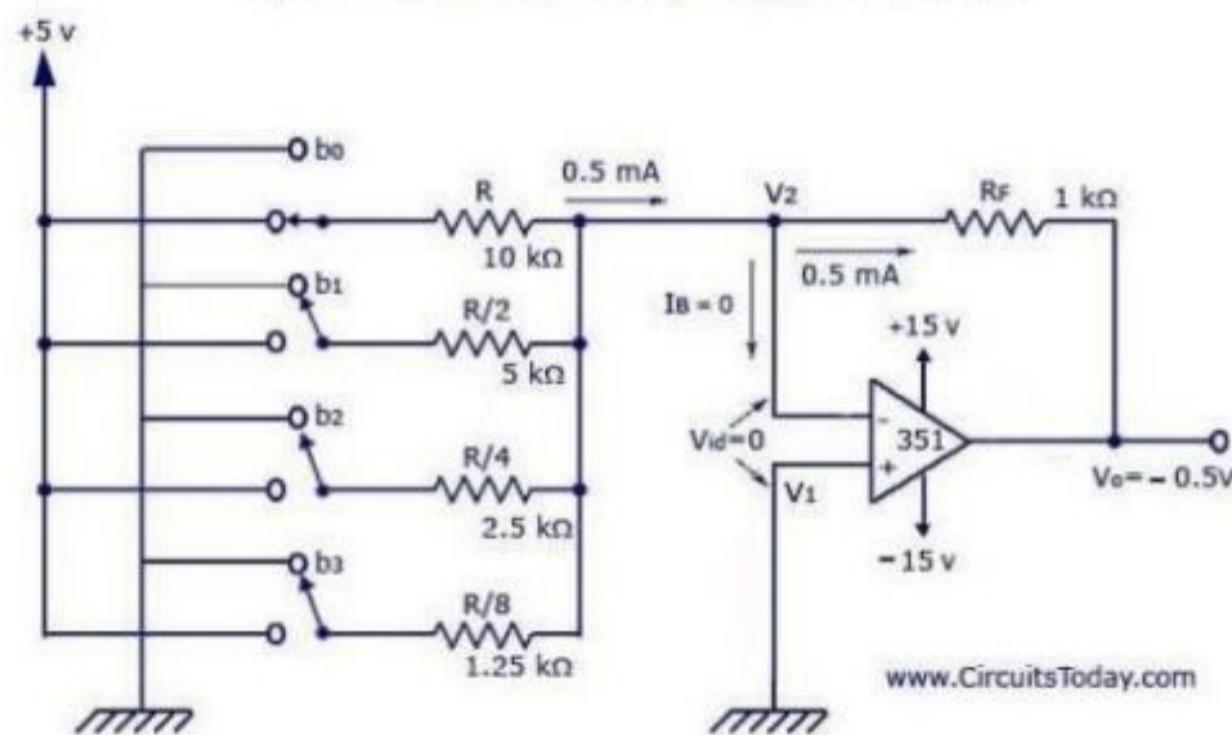
➤ Digital-to-Analog Converter (DAC):

- 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.

D/A Converter With Binary Weighted Resistors



15. Practical 7

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

Module 3

Aim: Study and implement a shifter.

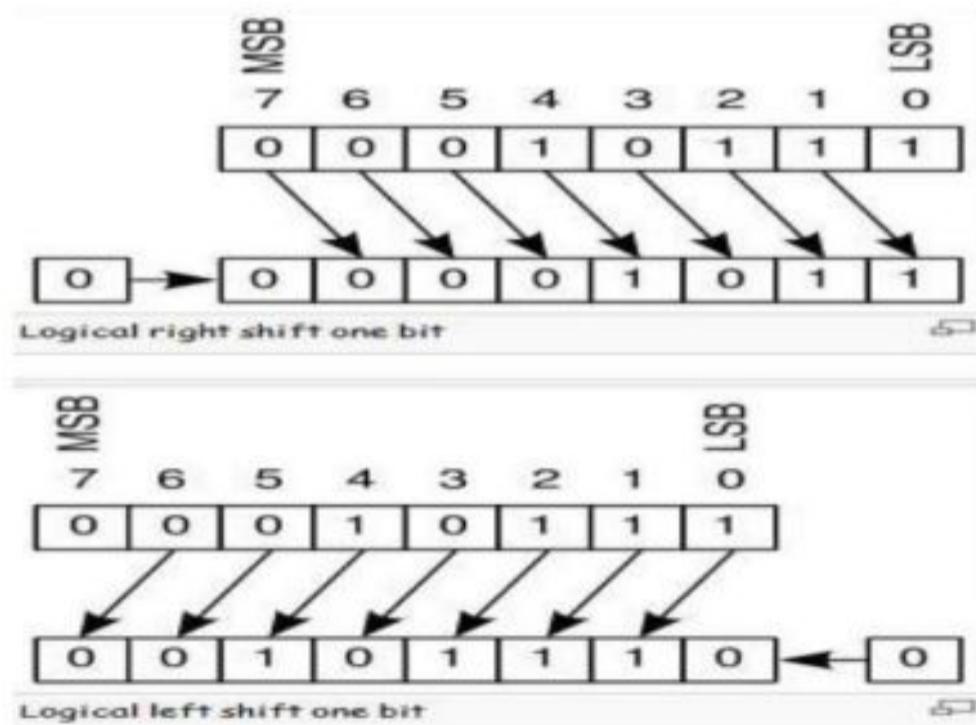
THEORY :

Shifters 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.

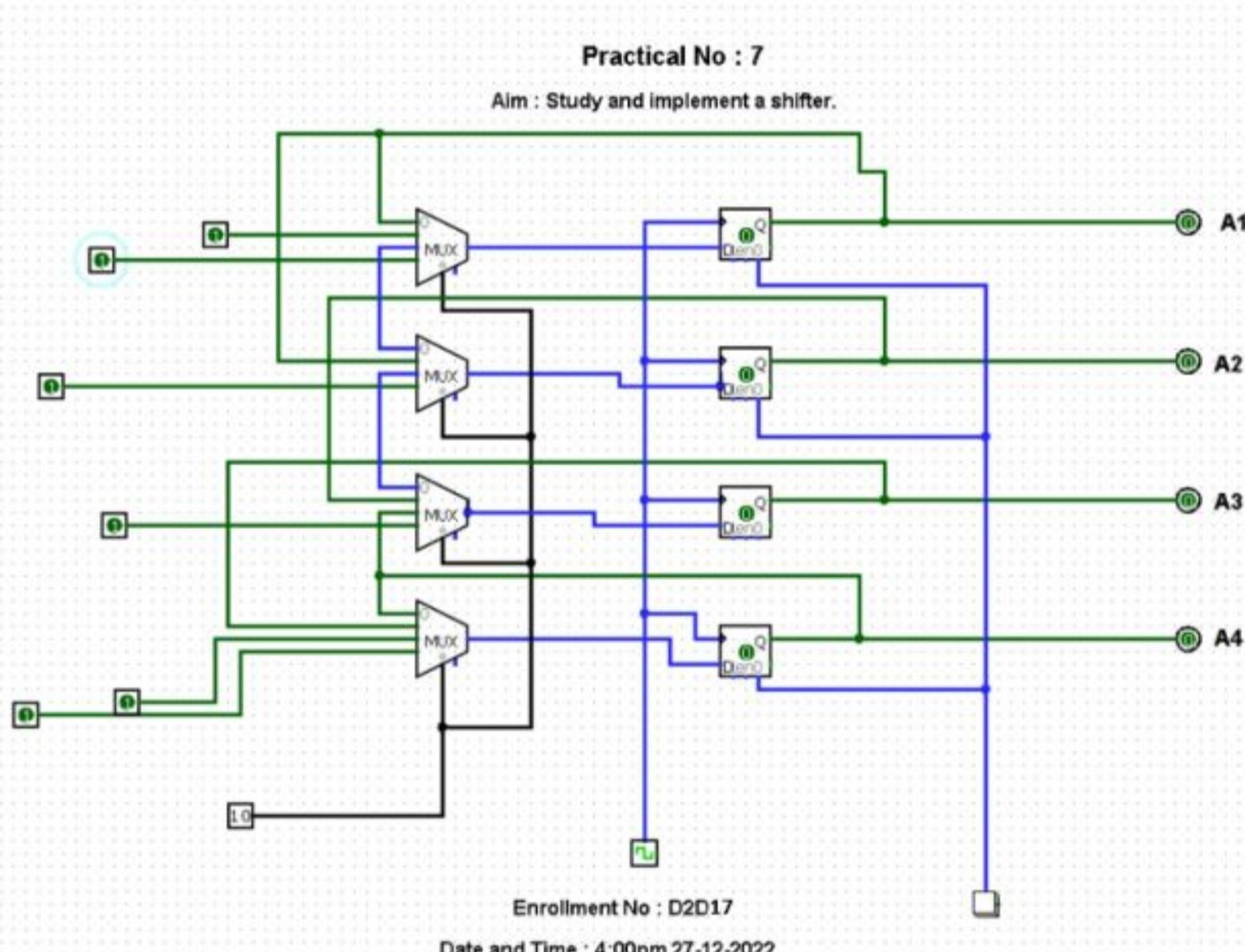
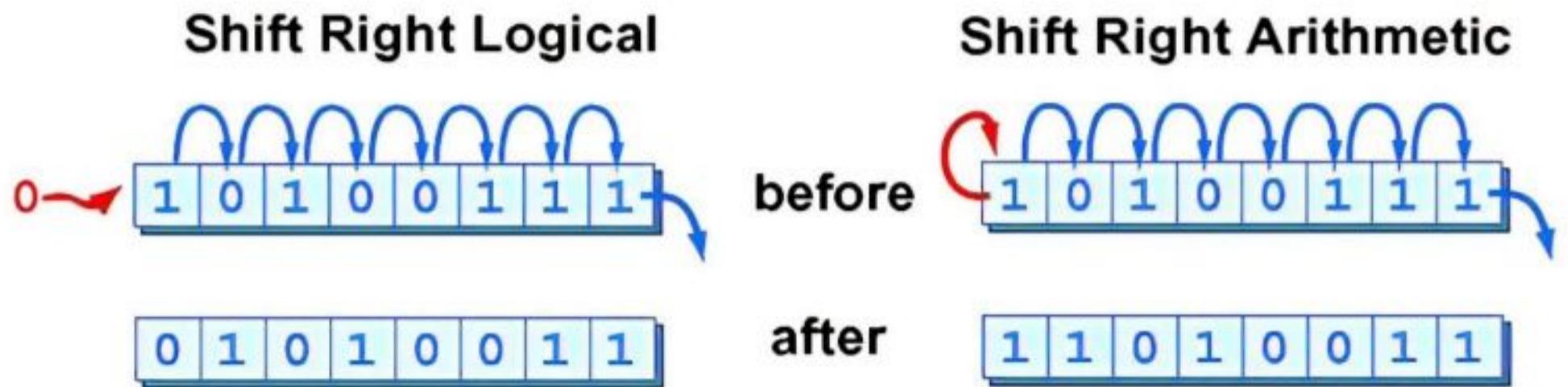
Row	S2	S1	S0	Output
0	0	0	0	?
1	0	0	1	No Shift
2	0	1	0	Shift Left
3	0	1	1	Rotate Right
4	1	0	0	?
5	1	0	1	?
6	1	1	0	?
7	1	1	1	?

There are several kinds of commonly used shifters:-

- 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



16. Practical 8

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

Module 3

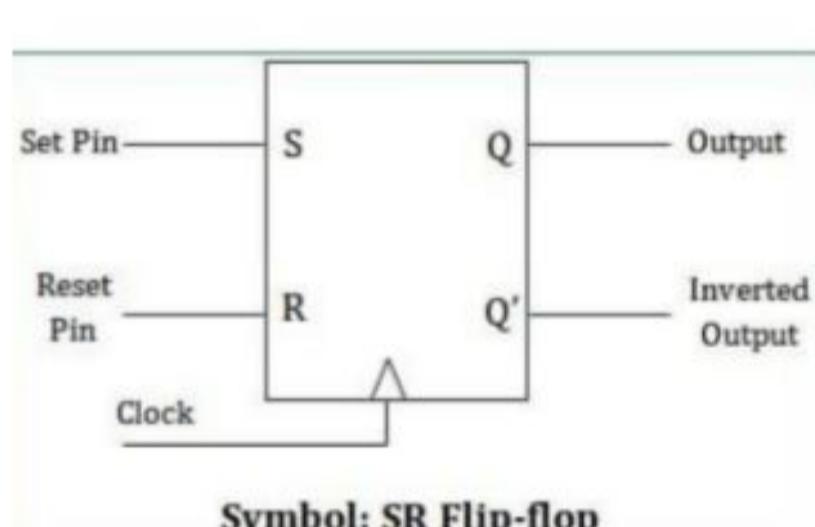
Aim: Study and implement Flip-flops.

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.

1. 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.

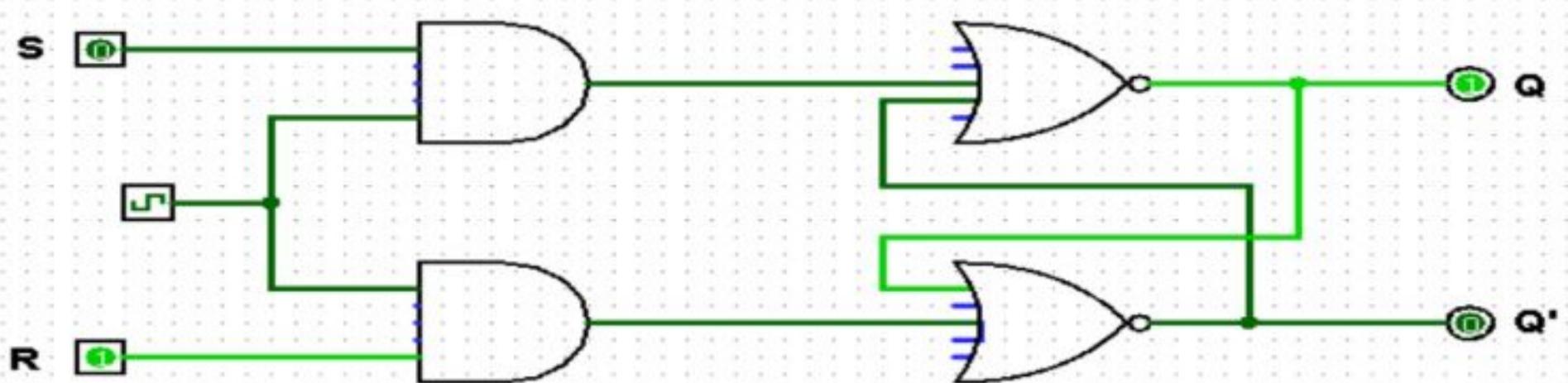


Symbol: SR Flip-flop

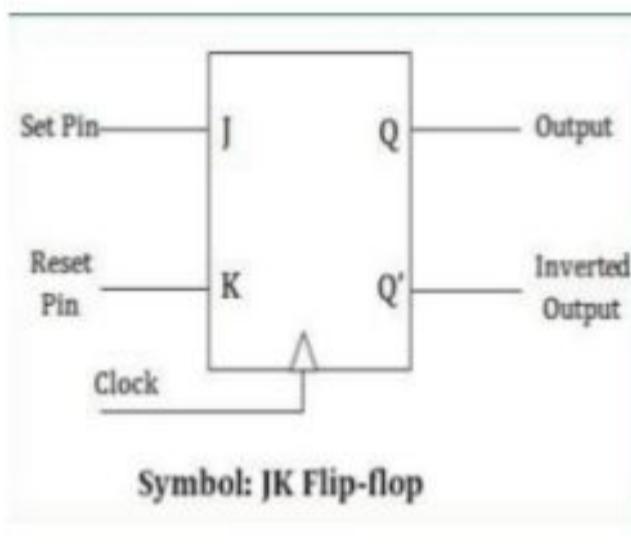
S	R	Q _n	Q _{n+1}	State
0	0	0	0	Q _n
0	0	1	1	
0	1	0	0	
0	1	1	0	
1	0	0	1	1
1	0	1	1	
1	1	0	INVALID	DON'T CARE
1	1	1	INVALID	

Practical No : 8

Aim: Study and implement Flip-flops.



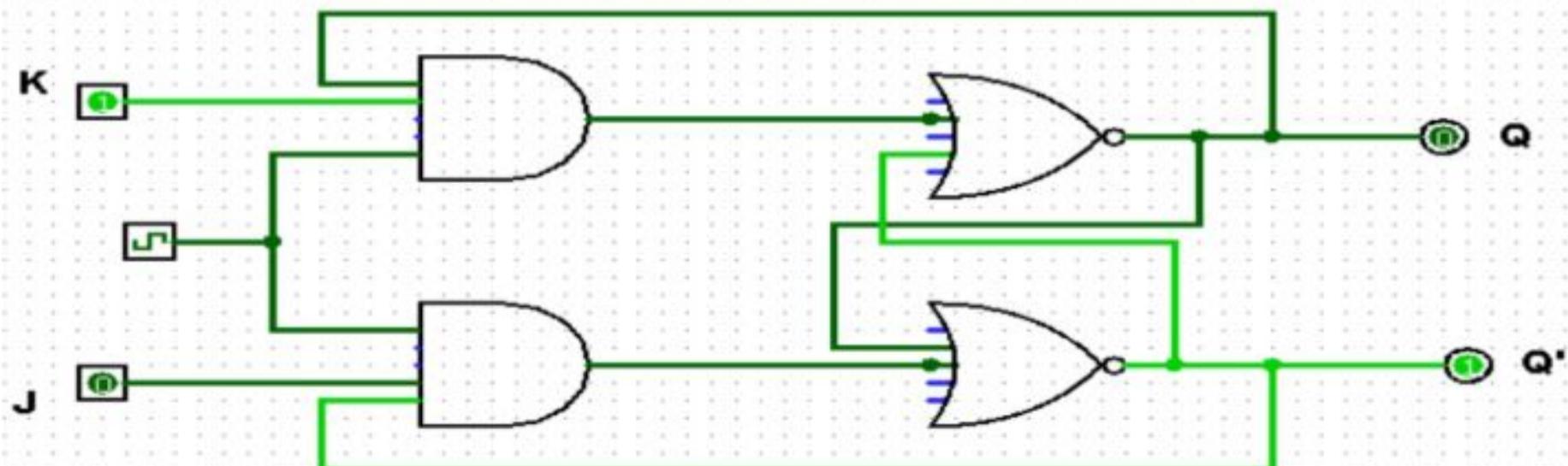
2. JK flip-flop : The J-K flip-flop is the most versatile of the basic flip-flops. It has the input-following character of the clocked D flip-flop but has two inputs, traditionally labeled J and K. If J and K are different then the output Q takes the value of J at the next clock edge.



J	K	Q _n	Q _{n+1}	State
0	0	0	0	Q _n
0	0	1	1	
0	1	0	0	0
0	1	1	0	
1	0	0	1	1
1	0	1	1	
1	1	0	1	Q _{n'}
1	1	1	0	

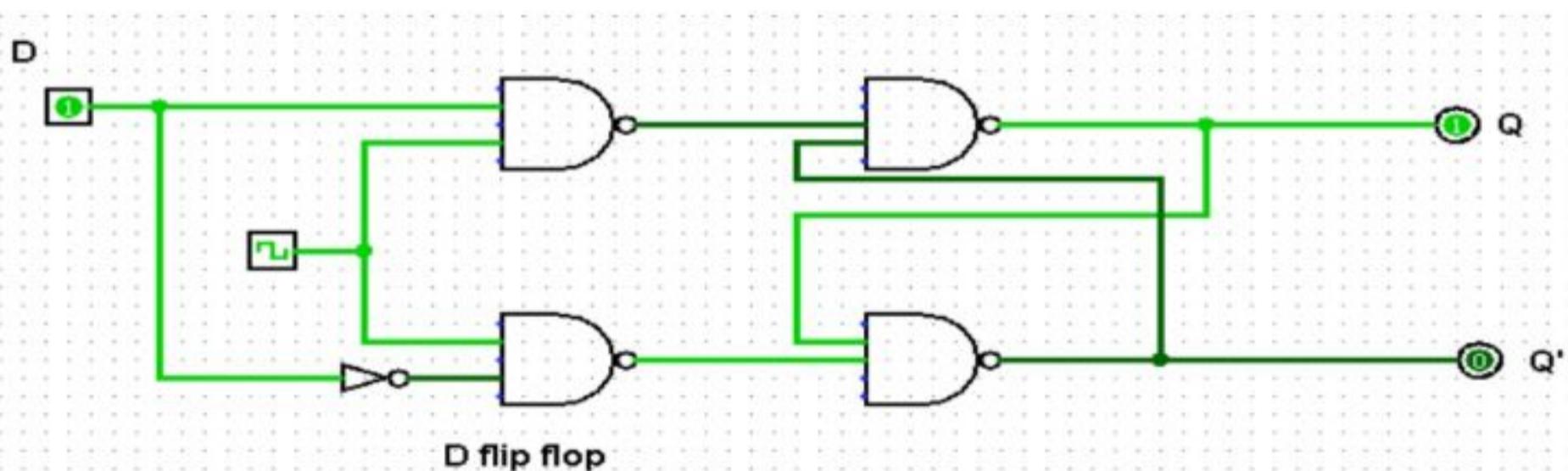
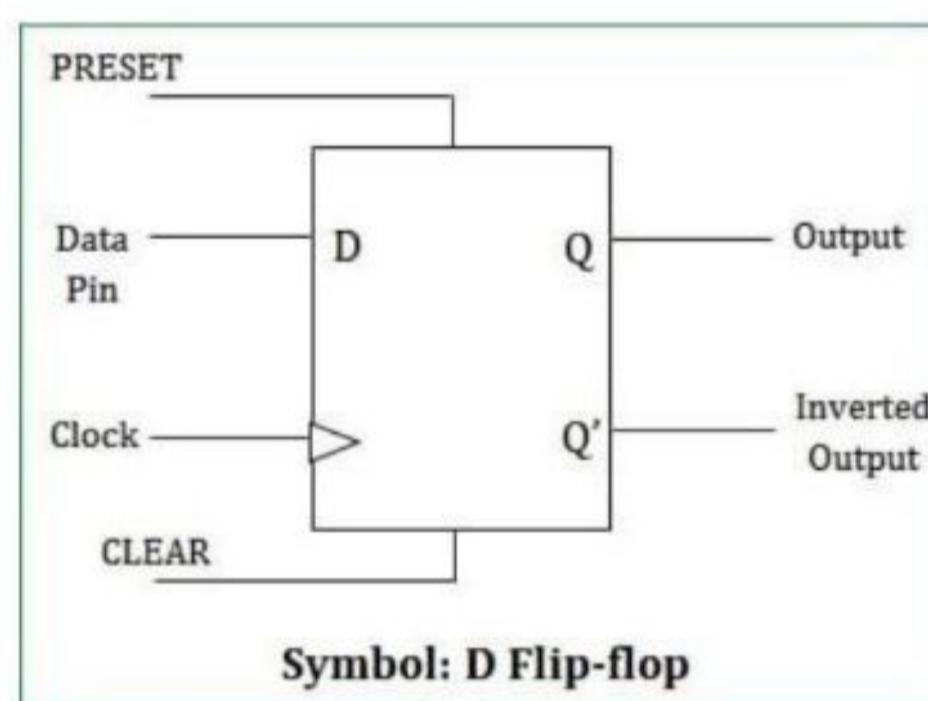
Practical No : 8

Aim: Study and implement Flip-flops.



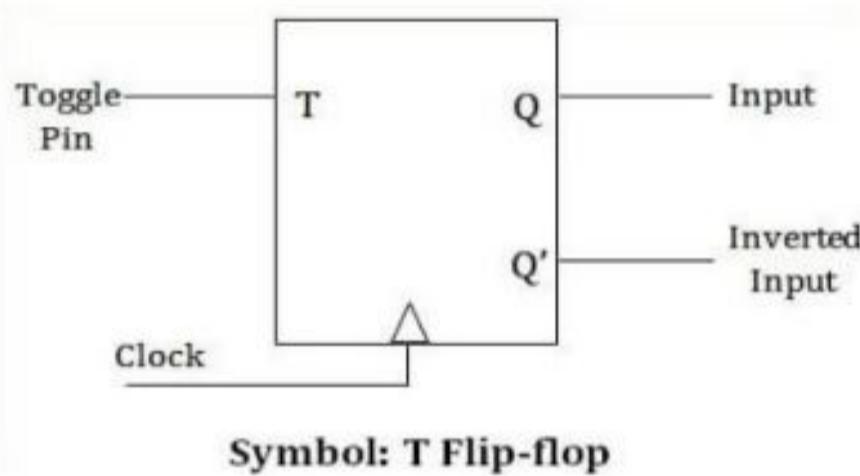
3. D flip-flop :

- 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 are 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 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.

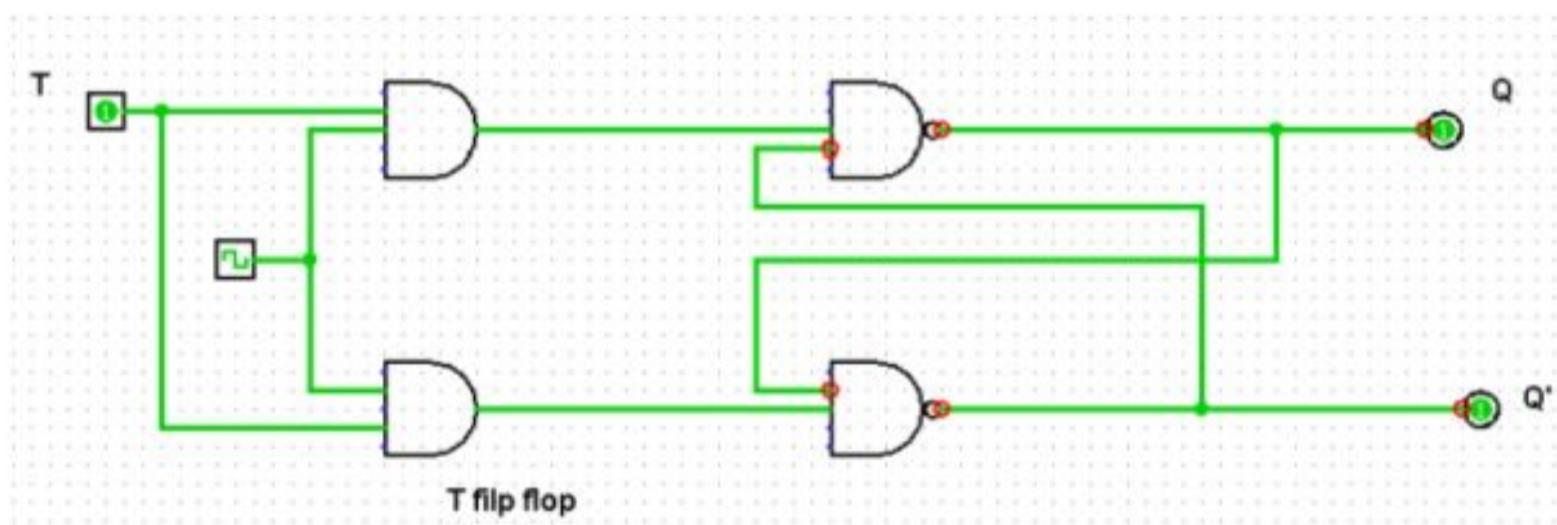


3. T flip-flop :

- 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.



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



Practical 9

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

Module 3

Aim: Study and implement Counter.

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.

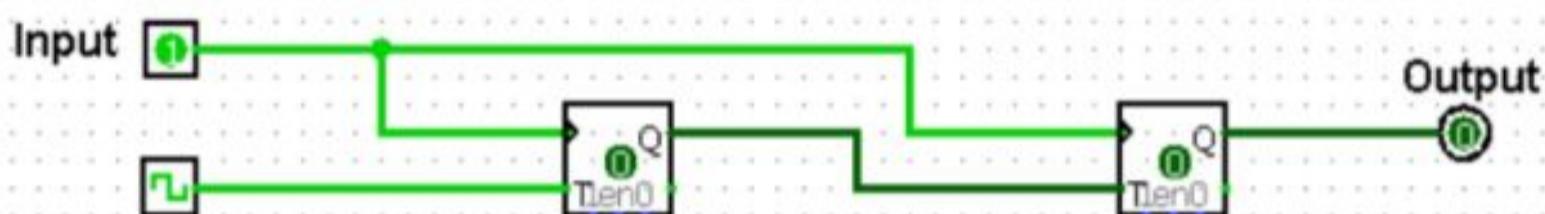
There are two types of counter based on the flip-flops that are connected in synchronous or not

- Asynchronous counters
- Synchronous counters

Synchronous 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.

Practical No : 9

Aim : Study and implement a counter.



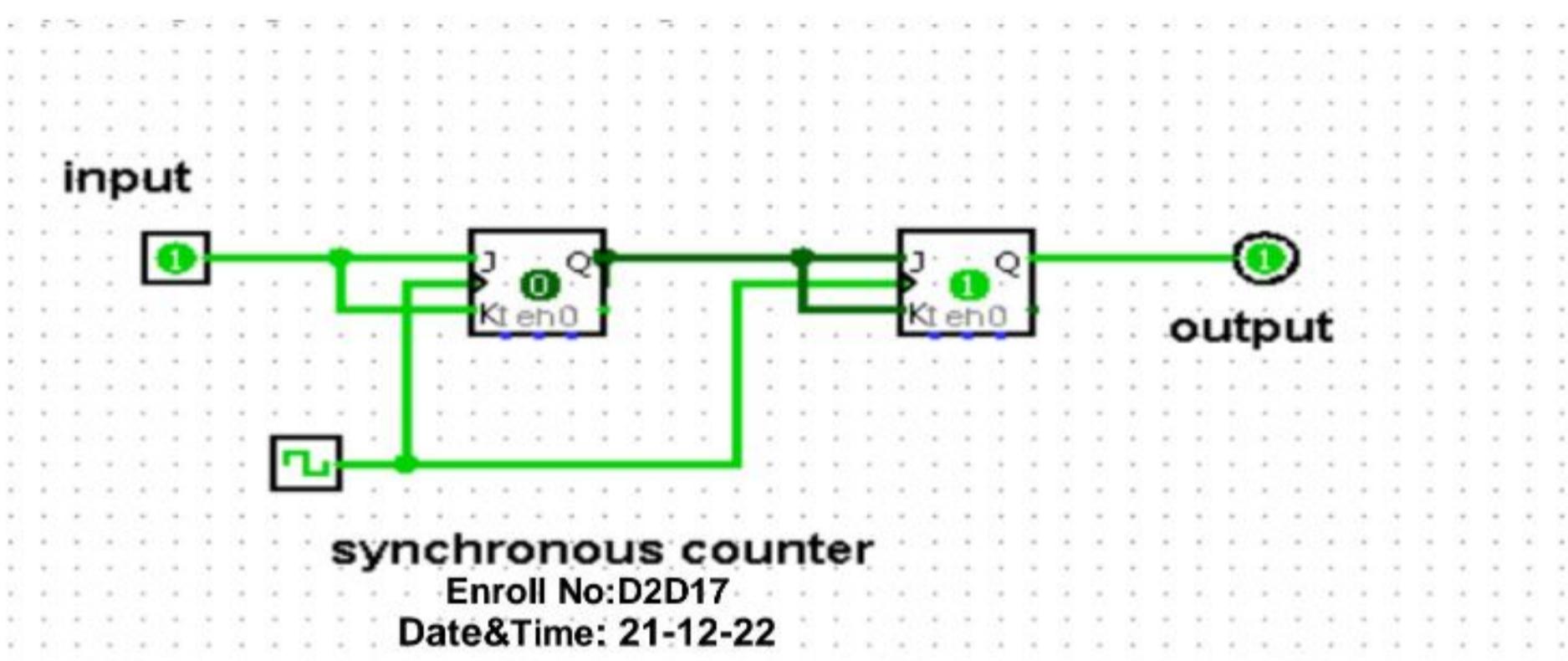
Enrollment No : D2D17

Date and Time : 1:30pm 28-12-2022

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.

Counter state	Q2	Q1	Q0
0	0	0	0
1	0	0	1
2	0	1	0
3	0	1	1
4	1	0	0
5	1	0	1
6	1	1	0
7	1	1	1



17. Practical 10

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

Module 3

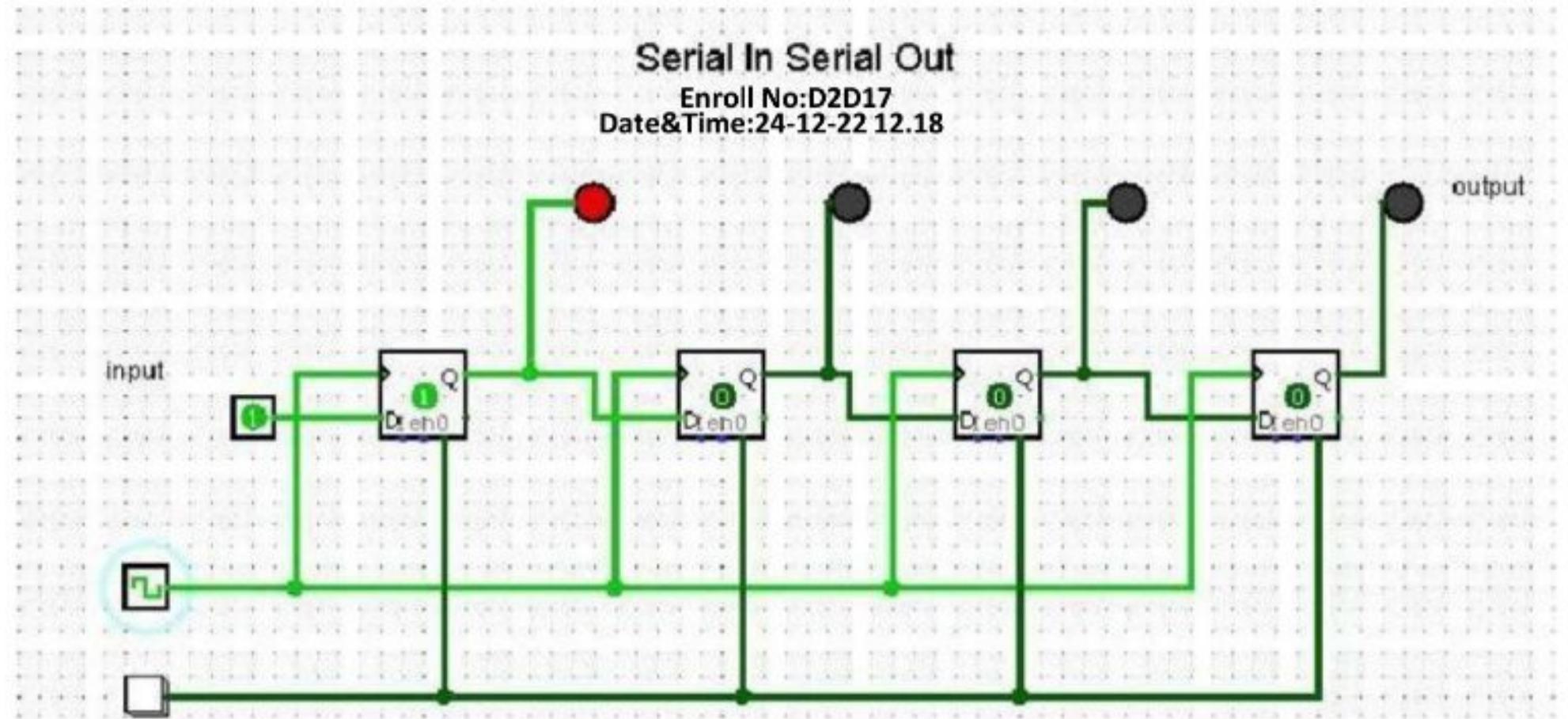
Aim: Study and implement a shift register.

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) :

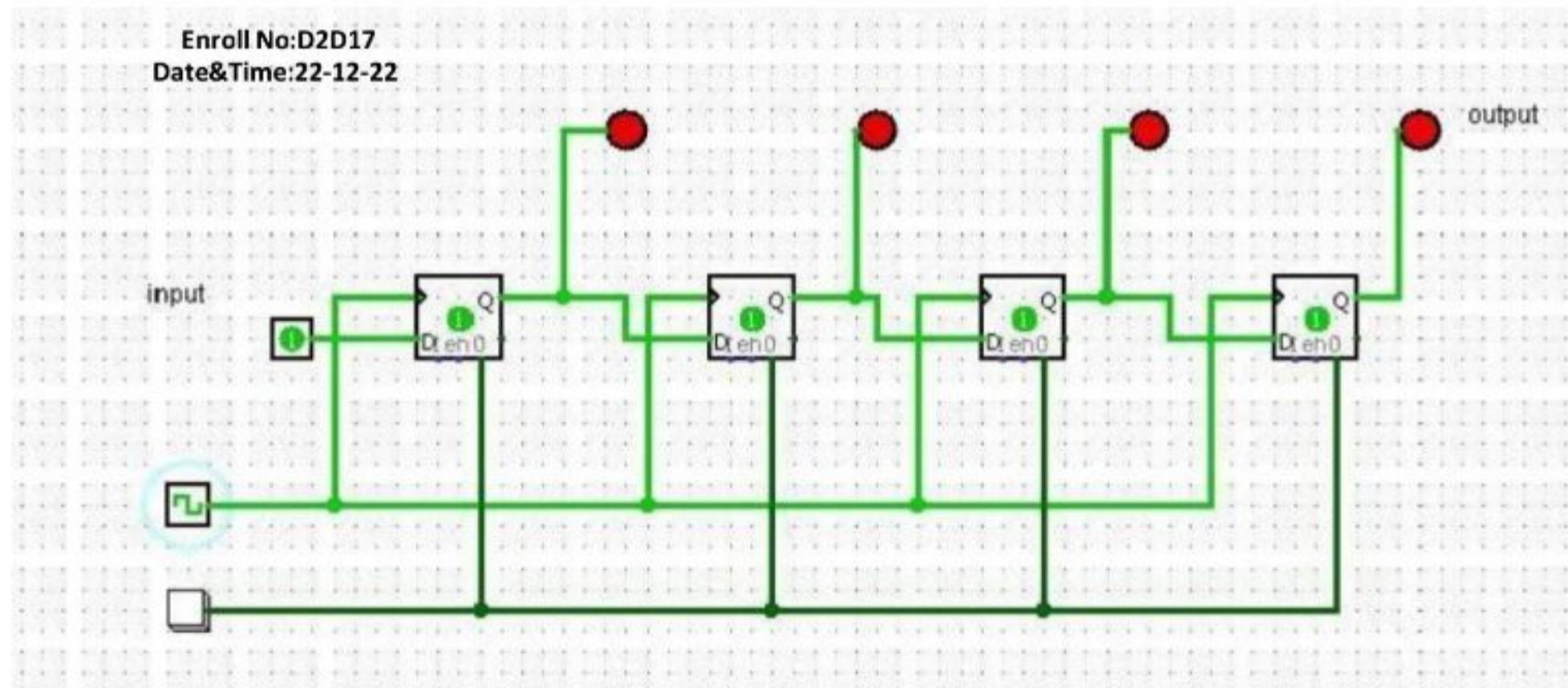
- The shift register, which allows serial input (one bit after the other through a single data line) and produces a serial output is known as Serial-In Serial-Out shift register. Since there is only one output, the data leaves the shift register one bit at a time in a serial pattern, thus the name Serial-In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

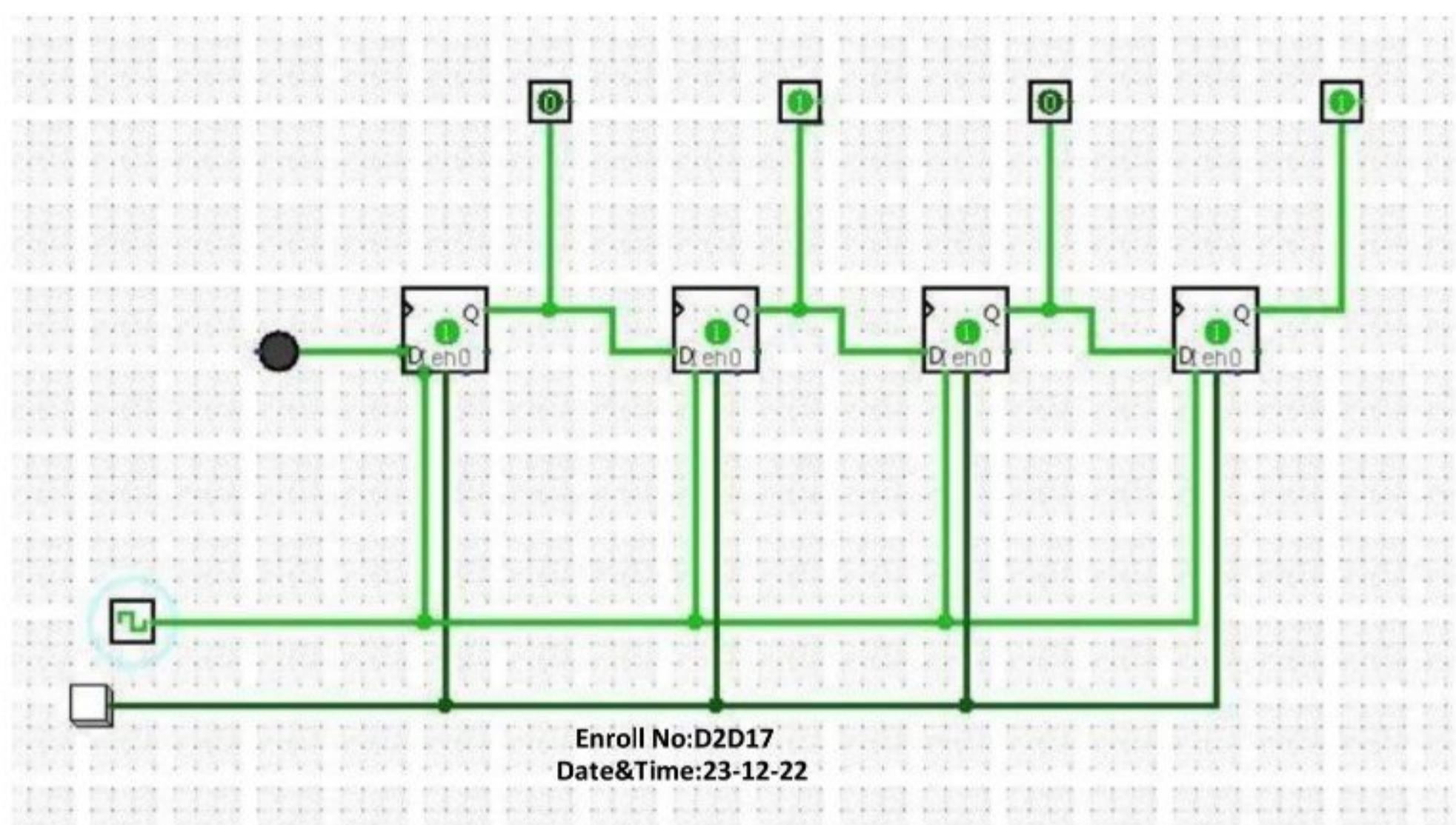


3. Parallel In Parallel Out (PIPO)

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

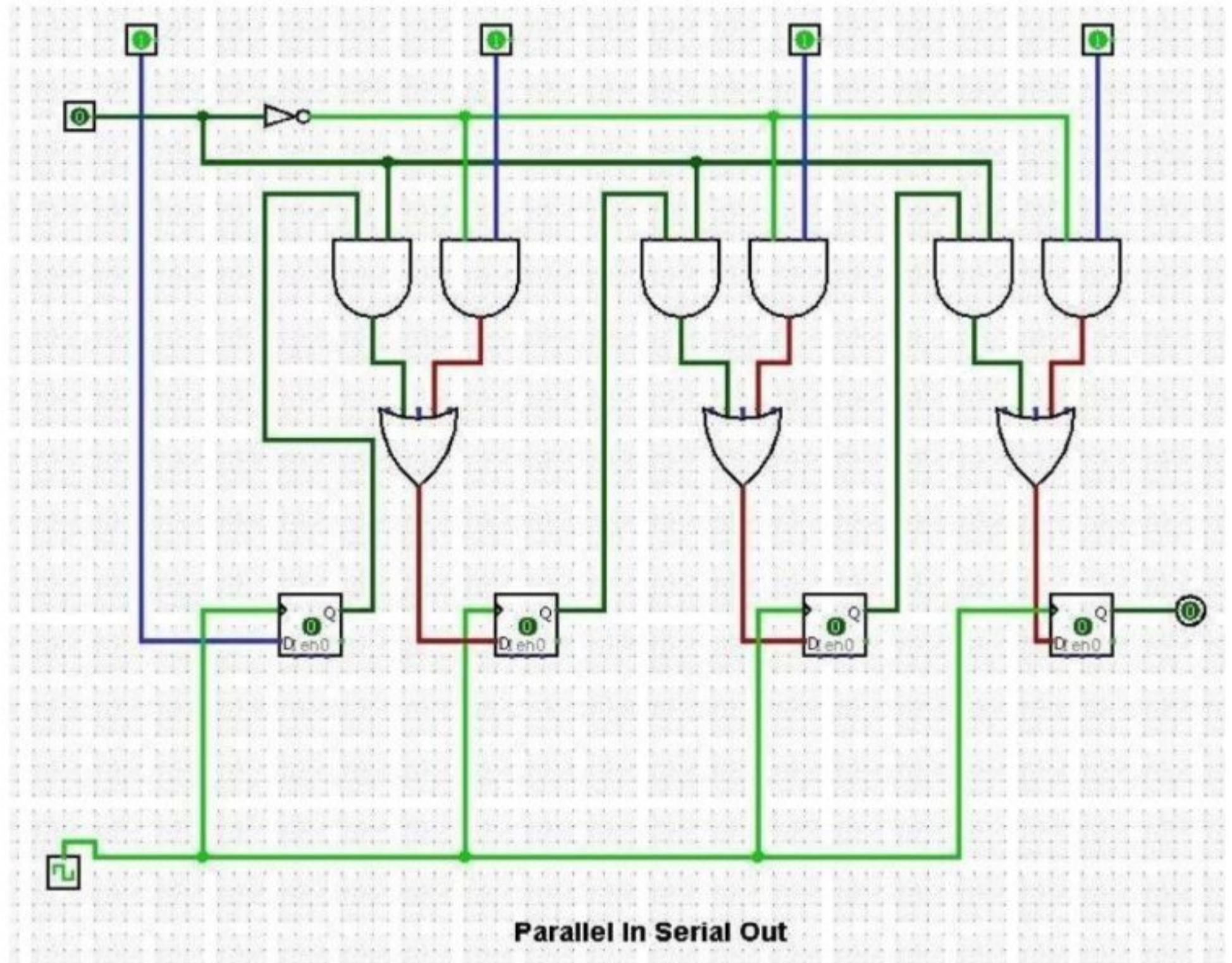
A Parallel in Parallel out (PIPO) shift register is used as a temporary storage device and like SISO Shift register it acts as a delay element.



4. Parallel In Serial Out (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 ParallelIn Serial-Out shift register. The logic circuit given below shows a parallel-in-serialout 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.



1. 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)

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

❖ K-Map (Karnaugh Map)

In many digital circuits and practical problems 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.

Step 1 : The K-map for four variables and it is plotted according to the given expression.

		AB			
		00	01	11	10
CD		00	01	11	10
00	0	4	12	8	
01	1	5	13	9	
11	3	7	15	11	
10	2	6	14	10	

Step 2: Cell 13 is only cell containing a 1 that is not adjacent to any other 1. It is referred to separately as group 1.

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

Step 3 : The top and bottom rows are considered to be adjacent to each other and the leftmost and rightmost columns are also adjacent to each other i.e 1 in the cell 0 , 1 in the cell 2, 1 in the cell 8 and 1 in the cell 10. And referred as group 2.

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

Step 4 : Each group generates a term in the expression for Y. In group 1 variable is not eliminated
In group 2 variable A and C are eliminated and we get

$$Y = AB'C'D + B'D'$$