



Government Engineering College

Sec-28 Gandhinagar

Sem: - 3

Subject: - Digital Fundamental

Subject Code: - 3130704

Contents

1. Institute Vision/Mission	6
Vision:	6
Mission:	6
2. Computer Engineering Department Vision/Mission	7
Vision:	7
Mission:	7
3. Program Educational Outcome (PEO)	8
4. PSO	8
5. POs	8
6. Submission Instructions:	Error! 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



Government Engineering College

Sector-28 Gandhinagar

Certificate

This is to certify that

Mr./Ms. Khemani..... Dinesh Dilipbhai Of class

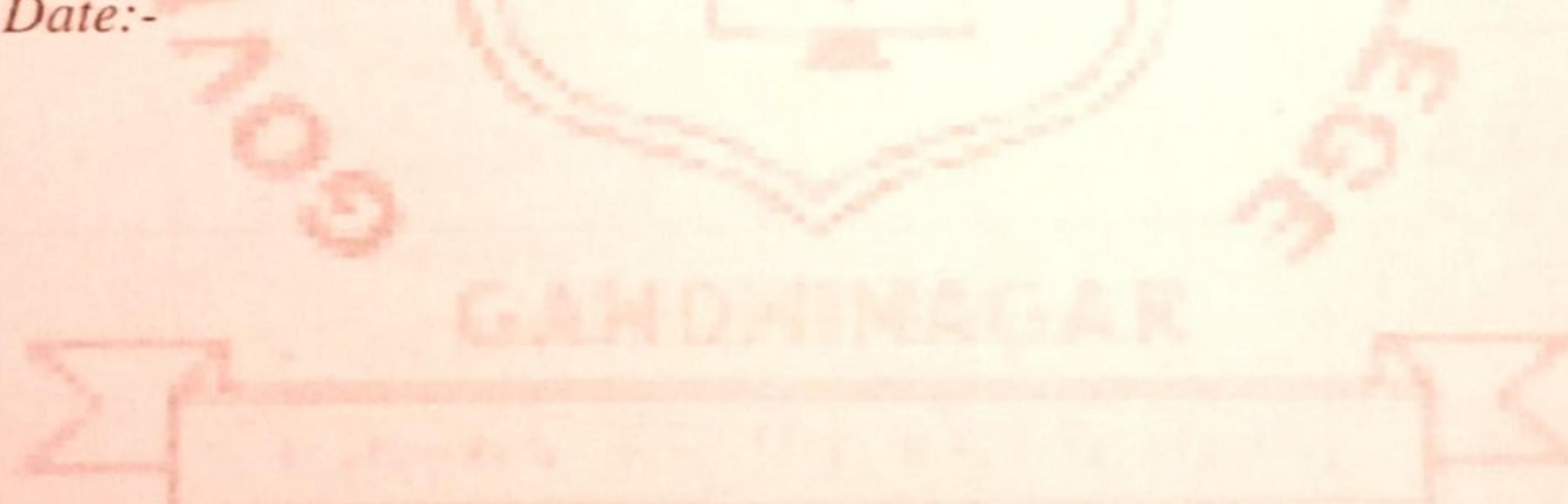
.... 3rd Division A....., Enrollment No. D2D04 Has

satisfactorily completed his/her term work in

..... DF Subject for the term

ending in ... Jan ... 2022-23.

Date:-



Signature of Teacher

Head of Department

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.

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.

Assignment Index

Sr. No	Assignment	Date	Page No.	Sign
1	Assignment 1	30/09/22		
2	Assignment 2	01/10/22		
3	Assignment 3	30/10/22		✓ <i>Mehdi</i>
4	Assignment 4	01/11/22		
5	Assignment 5	30/11/22		

Practical Index

Sr. No	Assignment	Date	Page No.	Sign
1	Practical 1	13/09/22		✓ <i>Mehdi</i>
2	Practical 2	24/09/22		
3	Practical 3	06/10/22		
4	Practical 4	19/10/22		
5	Practical 5	18/10/22		✓ <i>Mehdi</i>
6	Practical 6	15/11/22		
7	Practical 7	22/11/22		
8	Practical 8	29/11/22		
9	Practical 9	06/12/22		
10	Practical 10	23/12/22		
11	Practical 11	20/12/22		

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 one at 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.
- There are 3 basic gates (AND, OR, NOT), 2 universal gates (NAND, NOR) and 2 special gates (XOR, XNOR) available for different operations.

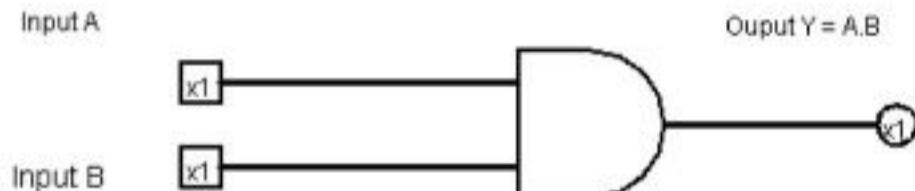
1. AND Gate :

- 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: $A \cdot B$

Enrollment no: D2D04

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

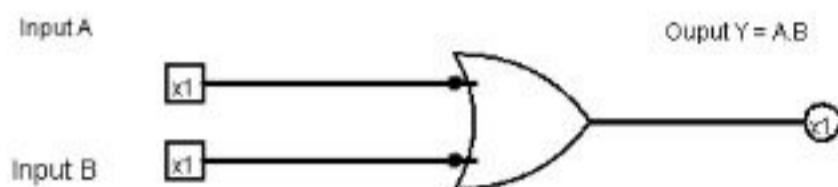
2. OR Gate :

- 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: $A+B$

Enrollment no:D2D04

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

3. 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'

Enrollment no:D2D04

Practical No : 1

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



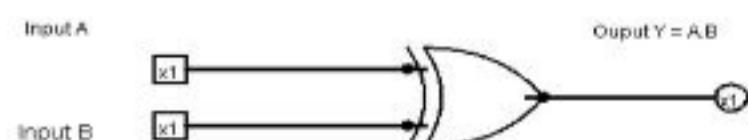
A	Y
0	1
1	0

4. Exclusive OR Gate :

- 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.

Practical No : 1

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



Enrollment No: R2P04

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

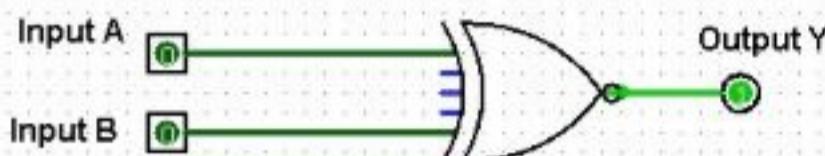
5. Exclusive NOR Gate :

- 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.

Enrollment no:D2D04

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	1
0	1	0
1	0	0
1	1	1

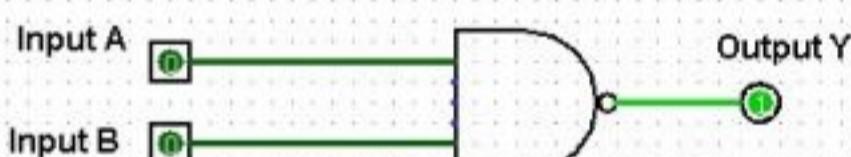
6. NAND Gate :

- 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.

Enrollment no:D2D04

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	1
0	1	1
1	0	1
1	1	0

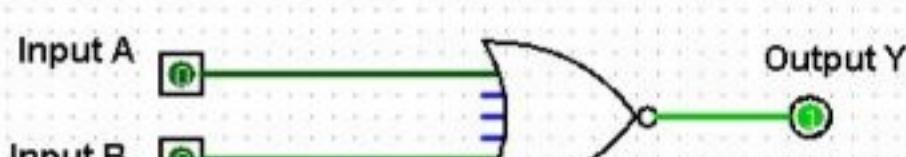
7. NOR GATE

- 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.

Enrollment no:D2D04

Practical No : 1

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

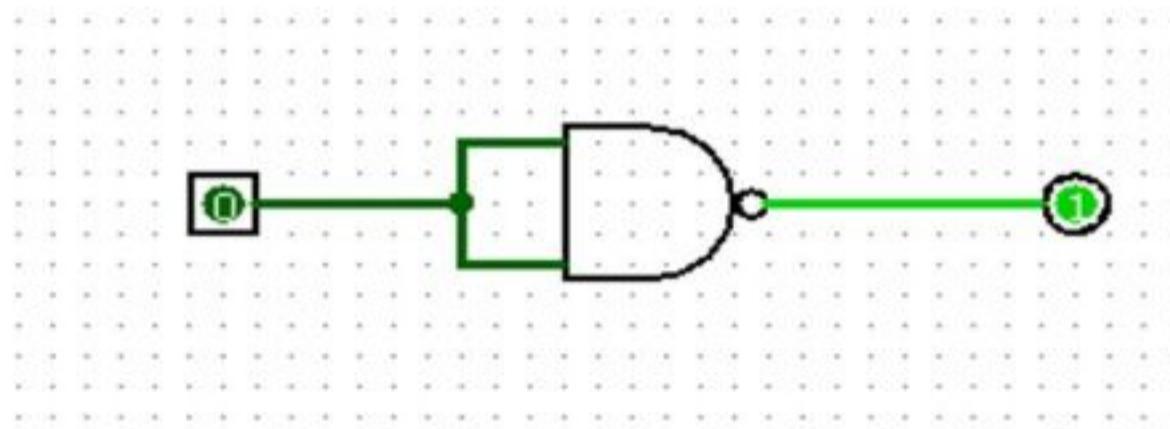


A	B	Y
0	0	1
0	1	0
1	0	0
1	1	0

NAND AS UNIVERSAL

1. NOT using NAND gate :

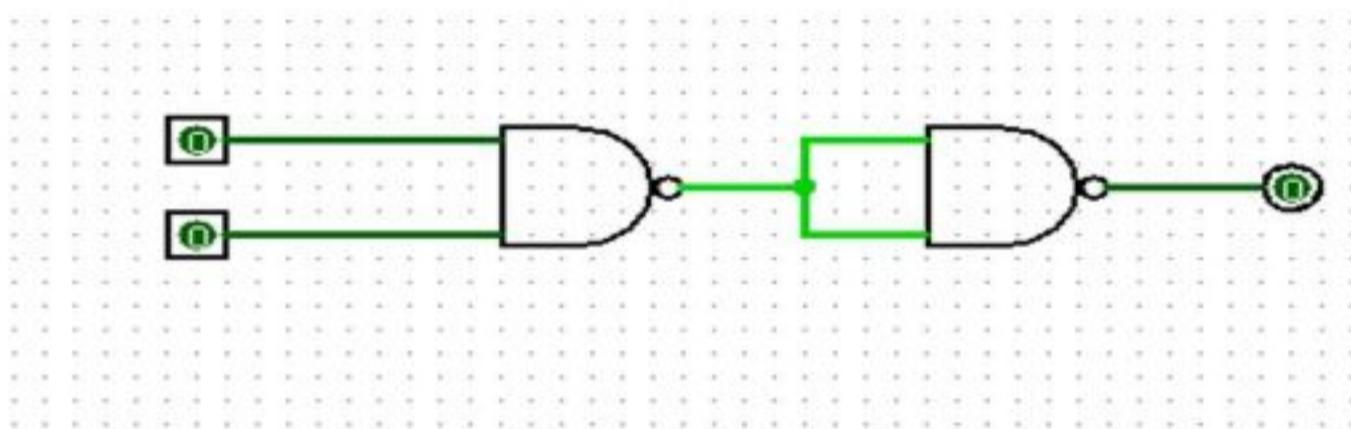
This is the circuit diagram of a NAND gate used to make work like a NOT gate
Enrollment no:D2D04



2. AND using NAND gate :

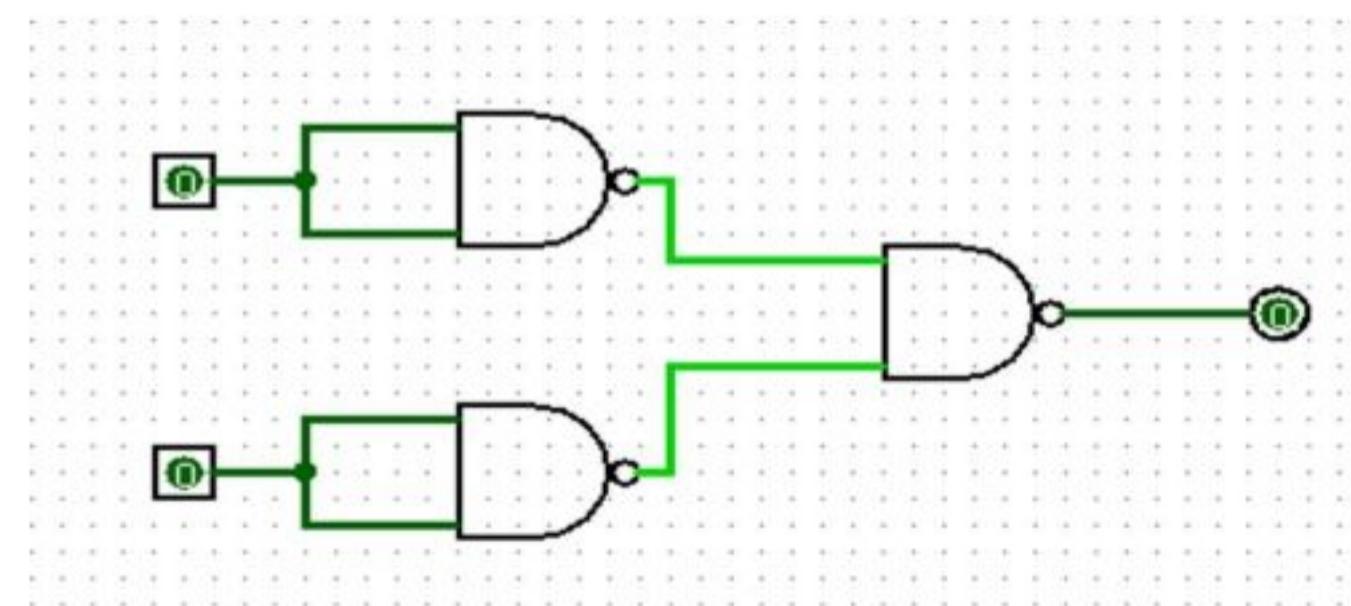
A NAND produces complement of AND gate. So, if the output of a NAND gate is inverted, overall output will be that of an AND gate.

Enrollment no:D2D04



3. OR using NAND gate :

an OR gate made from combinations of NAND gates, arranged in a proper manner.
Enrollment no:D2D04

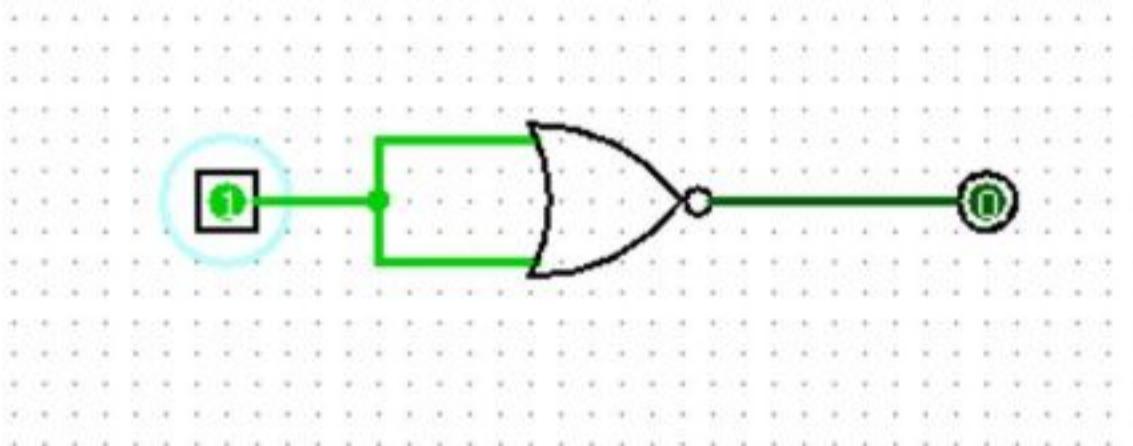


NOR as Universal

1. NOT using NOR gate :

This is made by joining the inputs of a NOR gate. As a NOR gate is equivalent to an OR gate leading to NOT gate, this automatically sees to the "OR" part of the NOR gate, eliminating it from consideration and leaving only the NOT part.

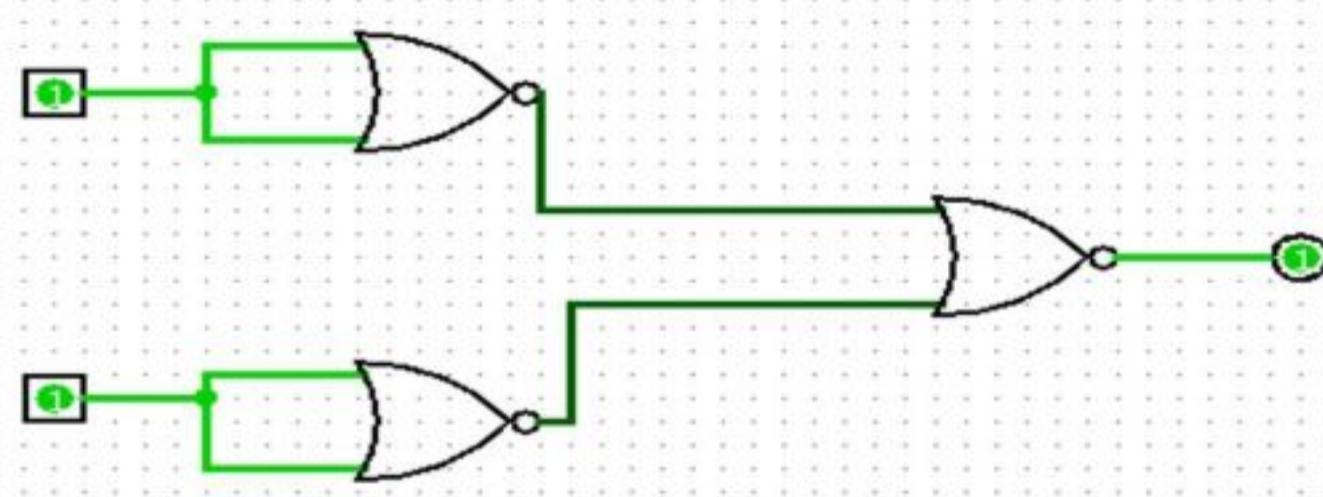
Enrollment no: D2D04



2. AND using NAND 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.

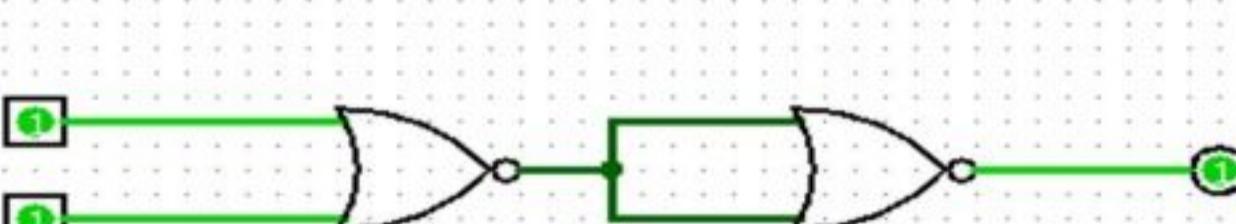
Enrollment no: D2D04



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.

Enrollment no:D2D04



15. Practical 2

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

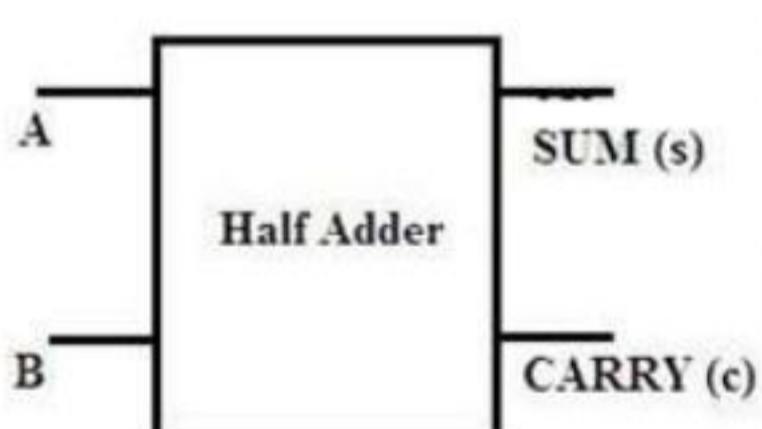
Module 2

Aim: Implement half and full Adders using logic gates.

THEORY :

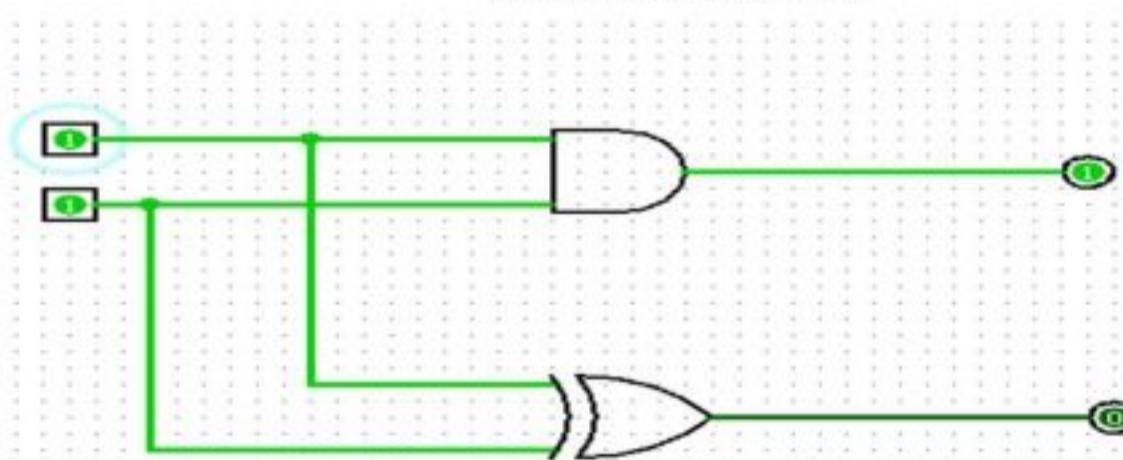
1. Half Adder :

- 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 = $X' \cdot Y$



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

Enrollment no :D2D04



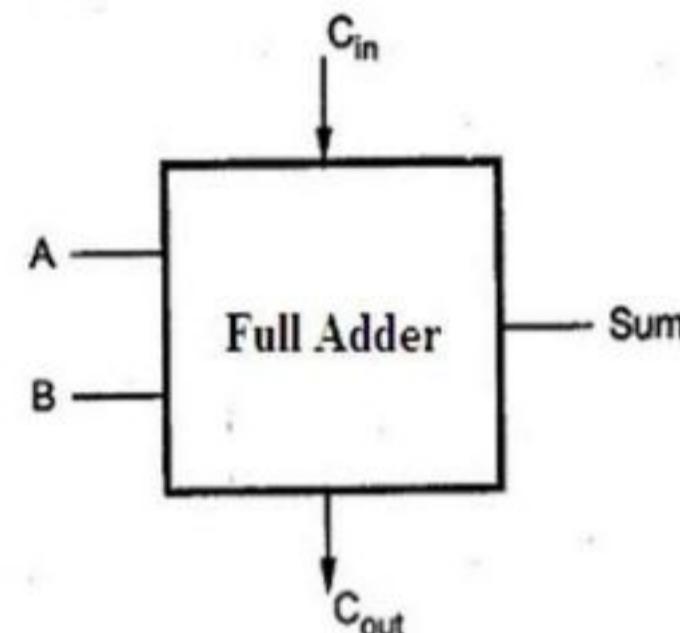
2. Full Adder :

- 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.
- 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. 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.

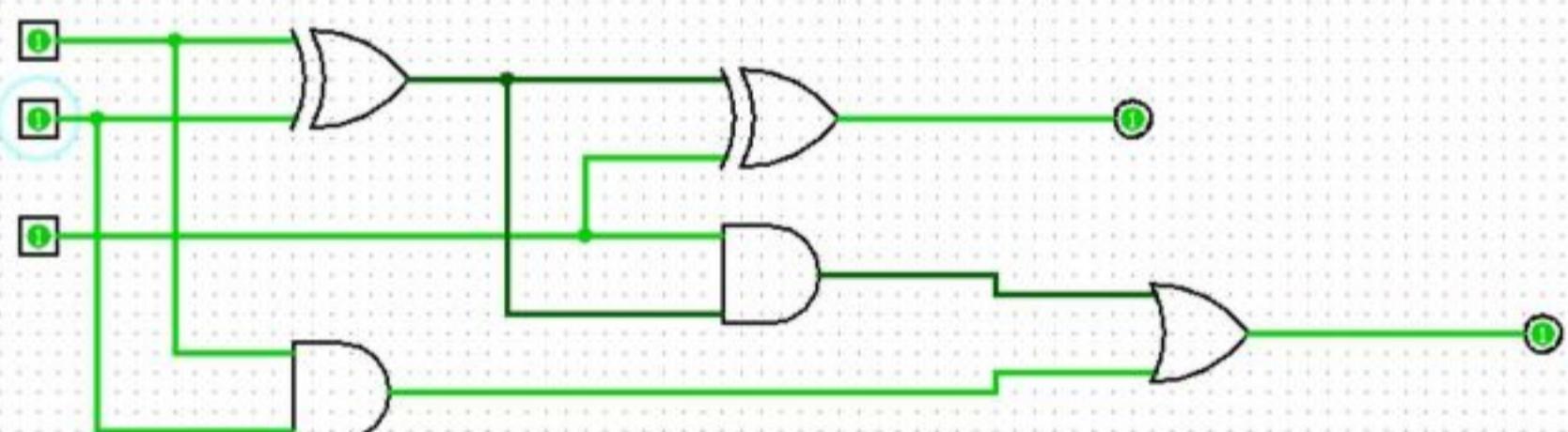
Boolean Expression: $\text{SUM } S = X \oplus Y \oplus Z$

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

Inputs			Outputs	
A	B	C _{in}	Sum	Carry
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1



Enrollment no: D2D04



16. Practical 3

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

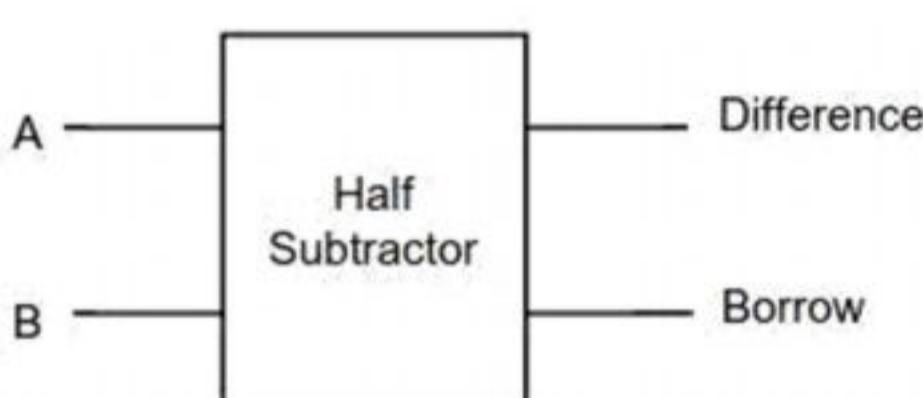
Module 2

Aim: Implement half and full Subtractors using logic gates.

THEORY :

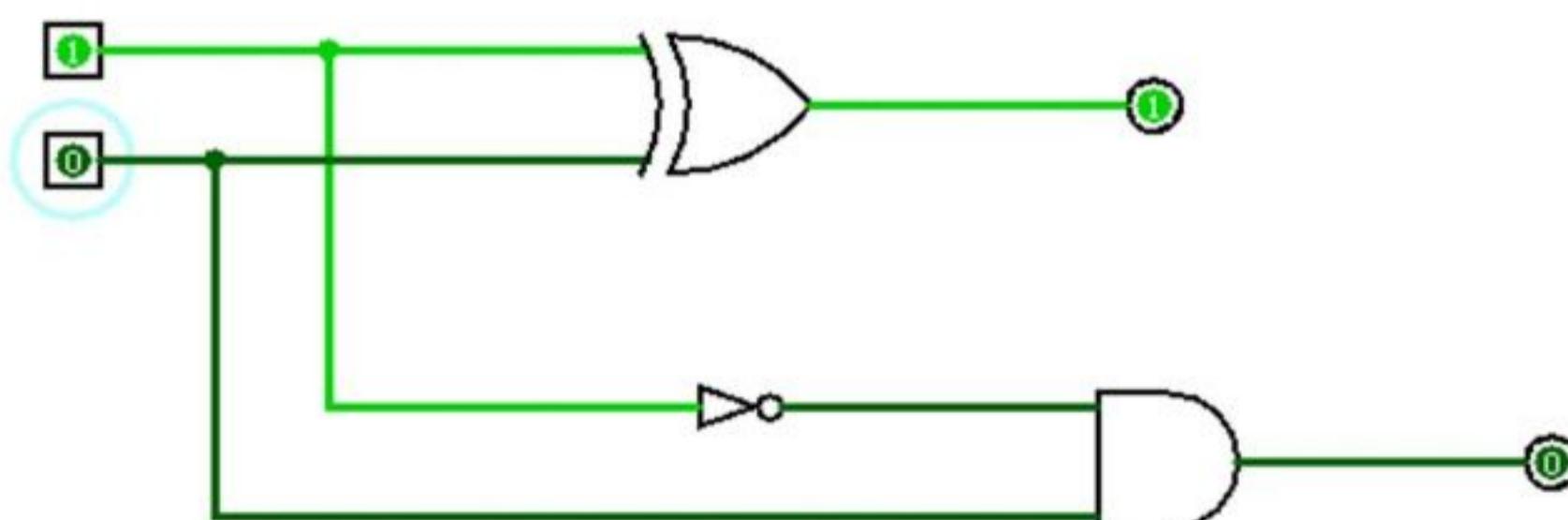
1. Half Subtractor :

- Half Subtractor is a combinational arithmetic circuit that subtracts two numbers and produces a difference bit (D) and borrow bit (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 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 $B = X' \cdot Y$



Inputs		Outputs	
A	B	Diff	Borrow
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0

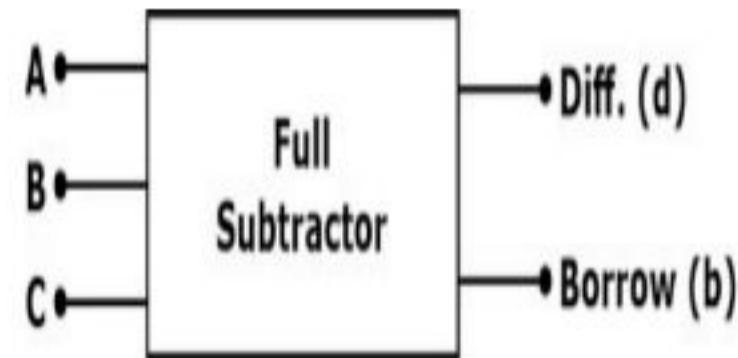
Enrollment No : D2D04



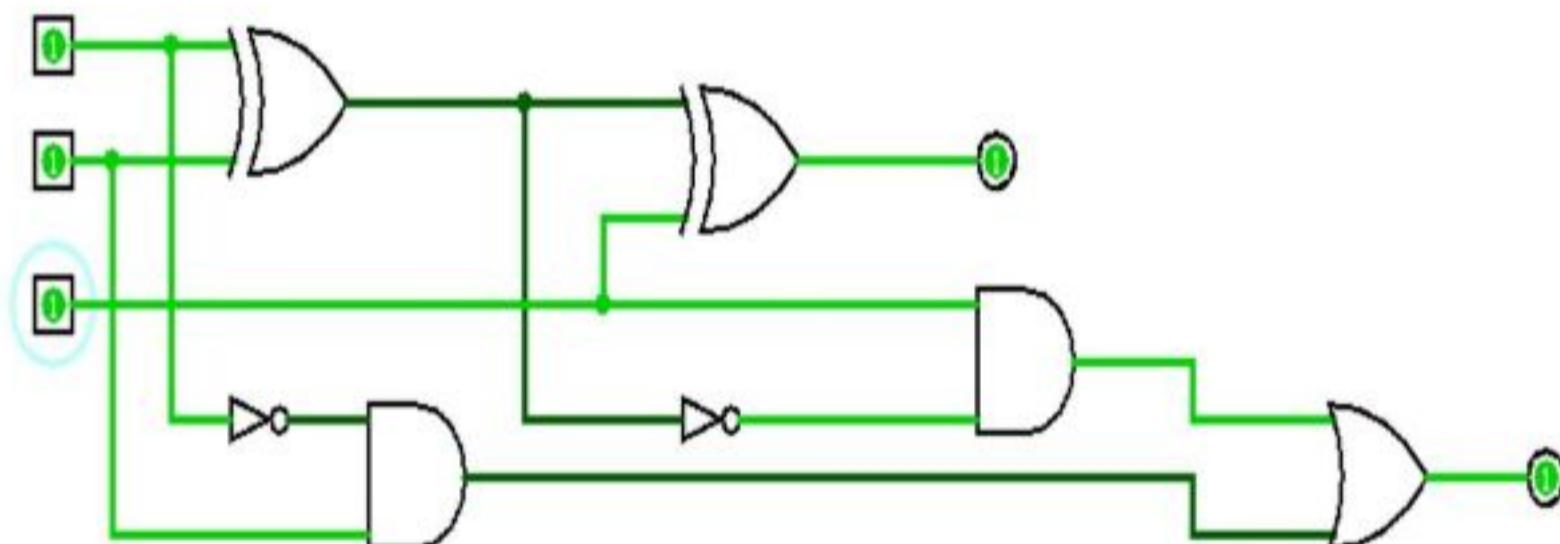
2. Full Subtractor :

- 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 $D = X \oplus Y \oplus Z$
BORROW $B = X'(Y \oplus Z) + YZ$

Inputs			Outputs	
A	B	Borrow _{in}	Diff	Borrow
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	0	1
1	0	0	1	0
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1



Enrollment No : D2D04



17. Practical 4

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

Module 2

Aim: Perform Parity Checker.

THEORY :

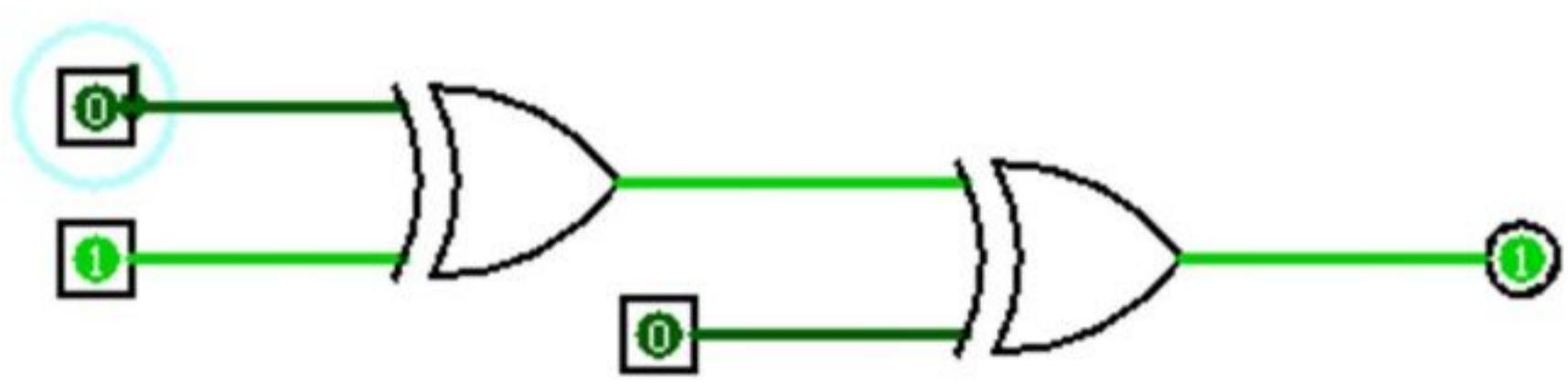
- **Parity Checker :** It is a logic circuit that checks for possible errors in the transmission. This circuit can be an even parity checker or odd parity checker depending on the type of parity generated at the transmission end. When this circuit is used as even parity checker, the number of input bits must always be even.
- **Even Parity Checker :** Consider that three input message along with even parity bit is generated at the transmitting end. These 4 bits are applied as input to the parity checker circuit, which checks the possibility of error on the data. Since the data is transmitted with even parity, four bits received at circuit must have an even number of 1s.
- **Odd Parity Checker :** Consider that a three bit message along with odd parity bit is transmitted at the transmitting end. Odd parity checker circuit receives these 4 bits and checks whether any error are present in the data. If the total number of 1s in the data is odd, then it indicates no error, whereas if the total number of 1s is even then it indicates the error since the data is transmitted with odd parity at transmitting end.

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

Even 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

Odd Parity Checker



18. 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 :

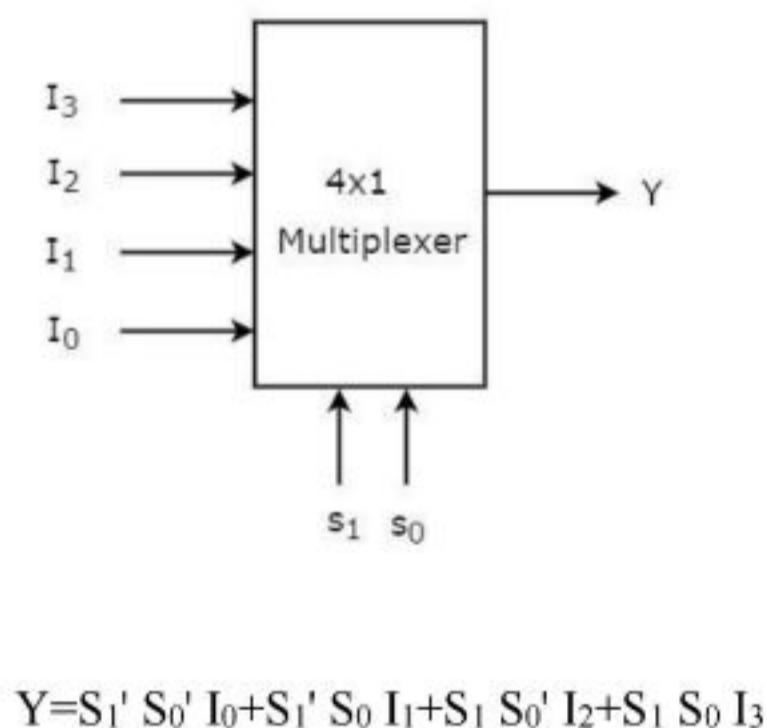
1. Multiplexer :

- A multiplexer is a combinational circuit that has 2^n 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 2^n input lines. So, there is a total of 2^N possible combinations of inputs. A multiplexer is also treated as Mux.

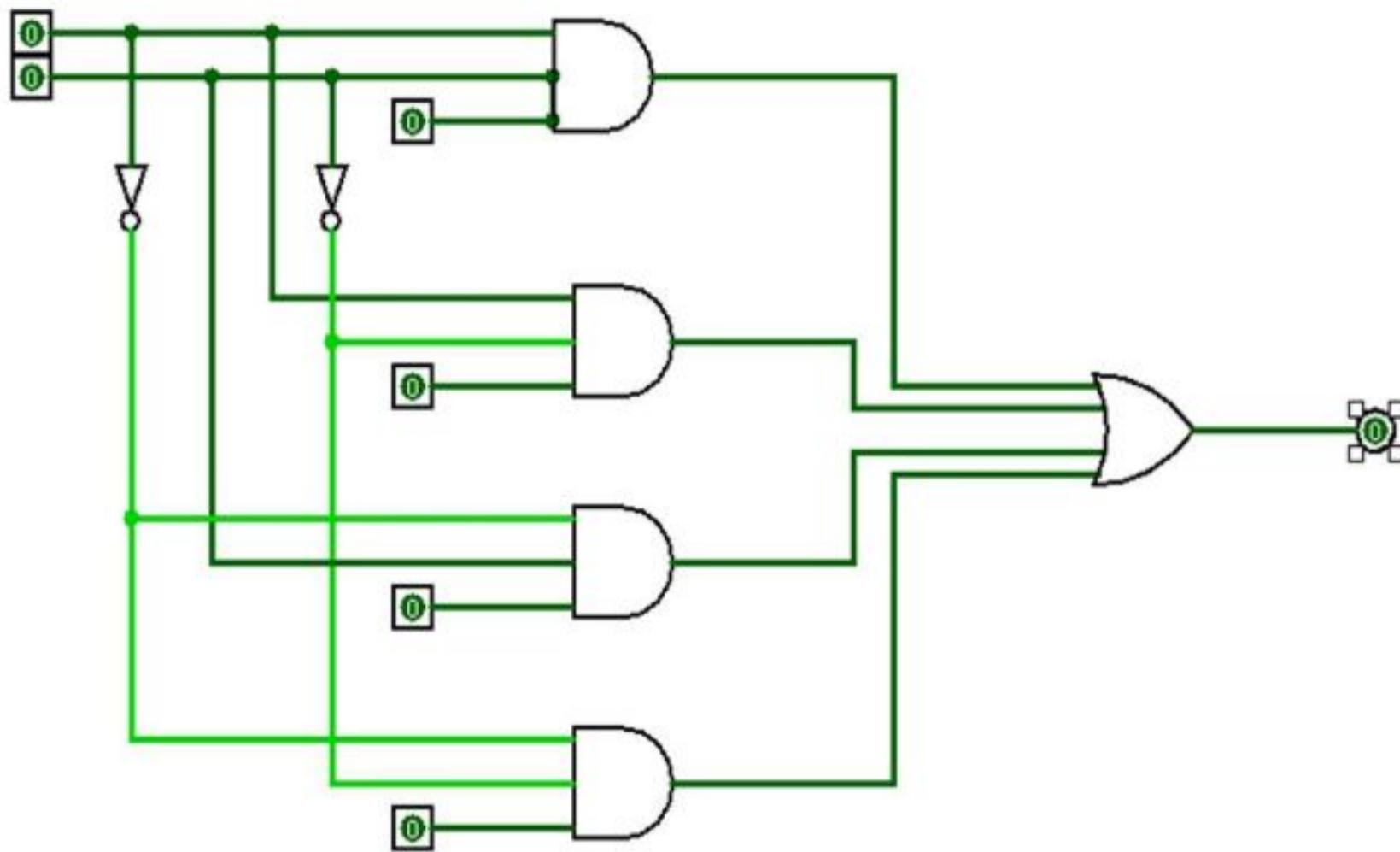
4x1 Multiplexer

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 . 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.



S ₁	S ₀	Y
0	0	I ₀
0	1	I ₁
1	0	I ₂
1	1	I ₃



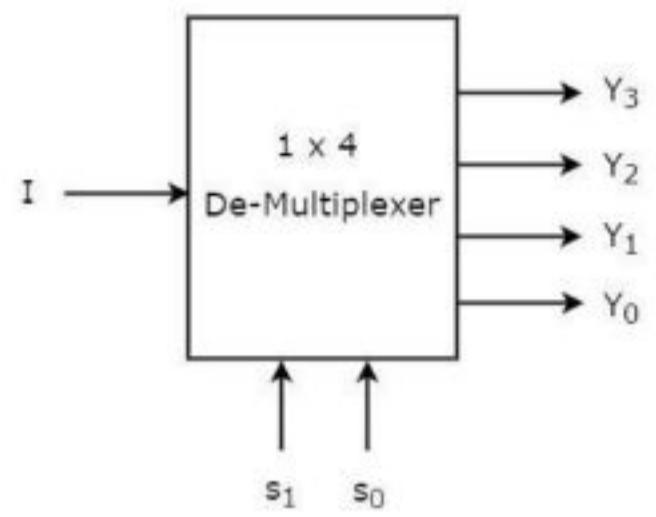
2. De-multiplexer

- A De-multiplexer is a combinational circuit that has only 1 input line and 2^N 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 2^n outputs. So, there is a total of 2^n 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, 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.



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

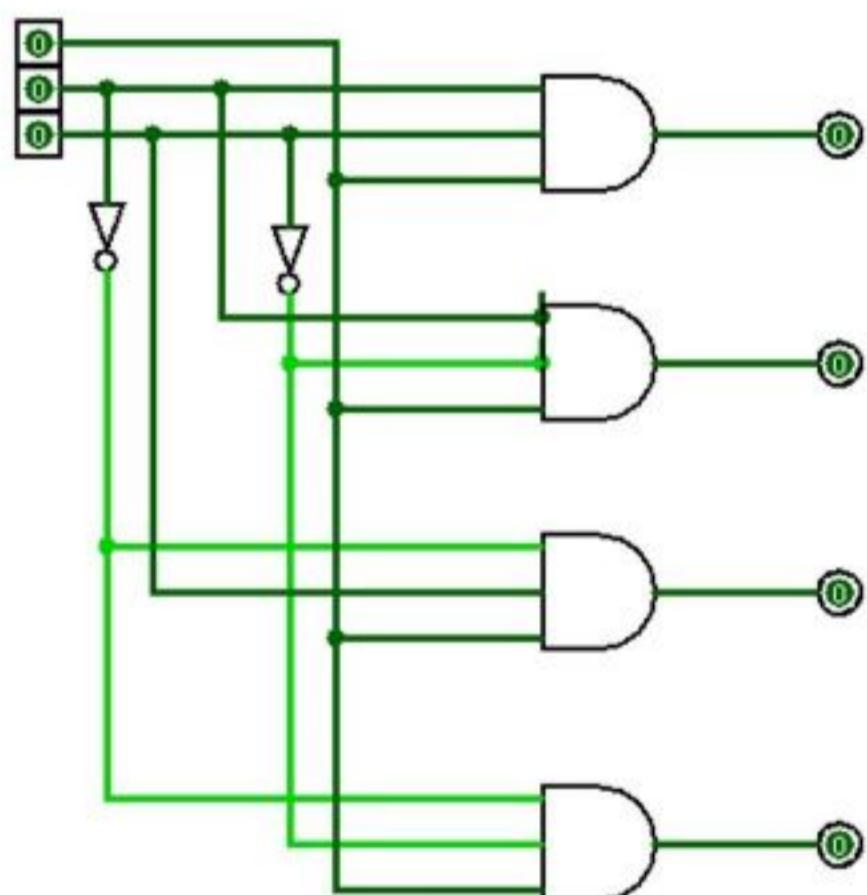
$$Y_2 = s_1 s_0' I \quad Y_2 = s_1 s_0' I$$

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

$$Y_0 = s_1' s_0' I$$

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

Enrollment No : D2D04



19. 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 ADC's 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.

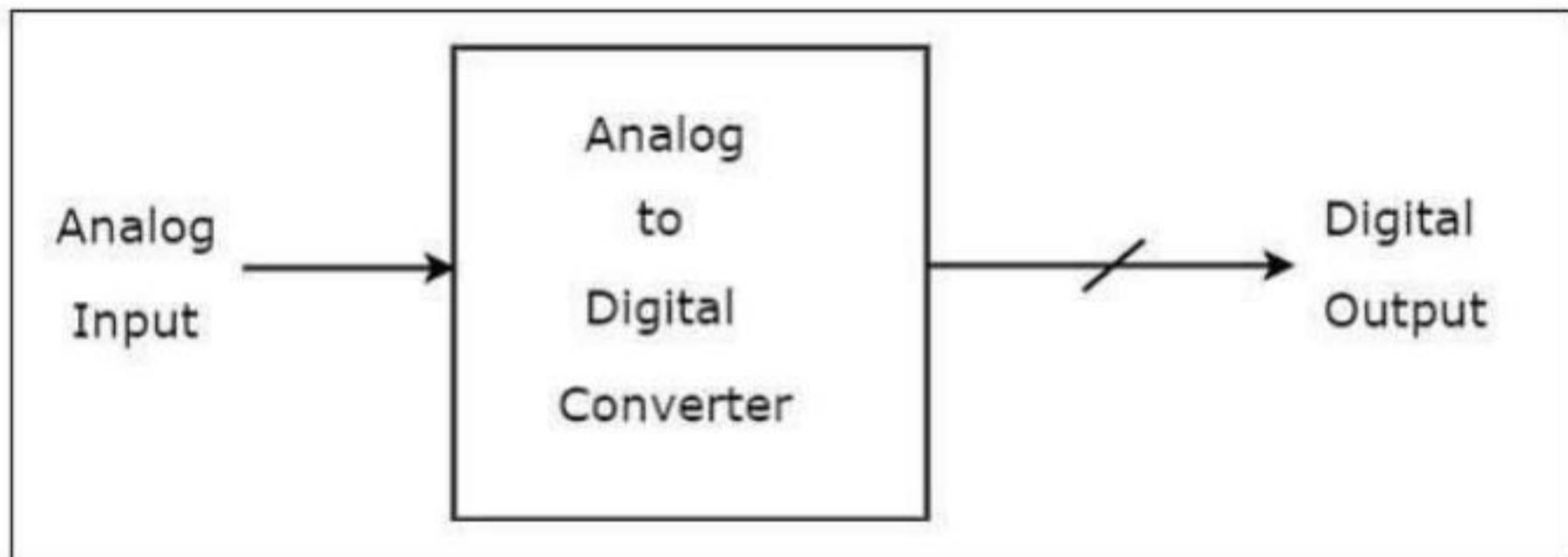
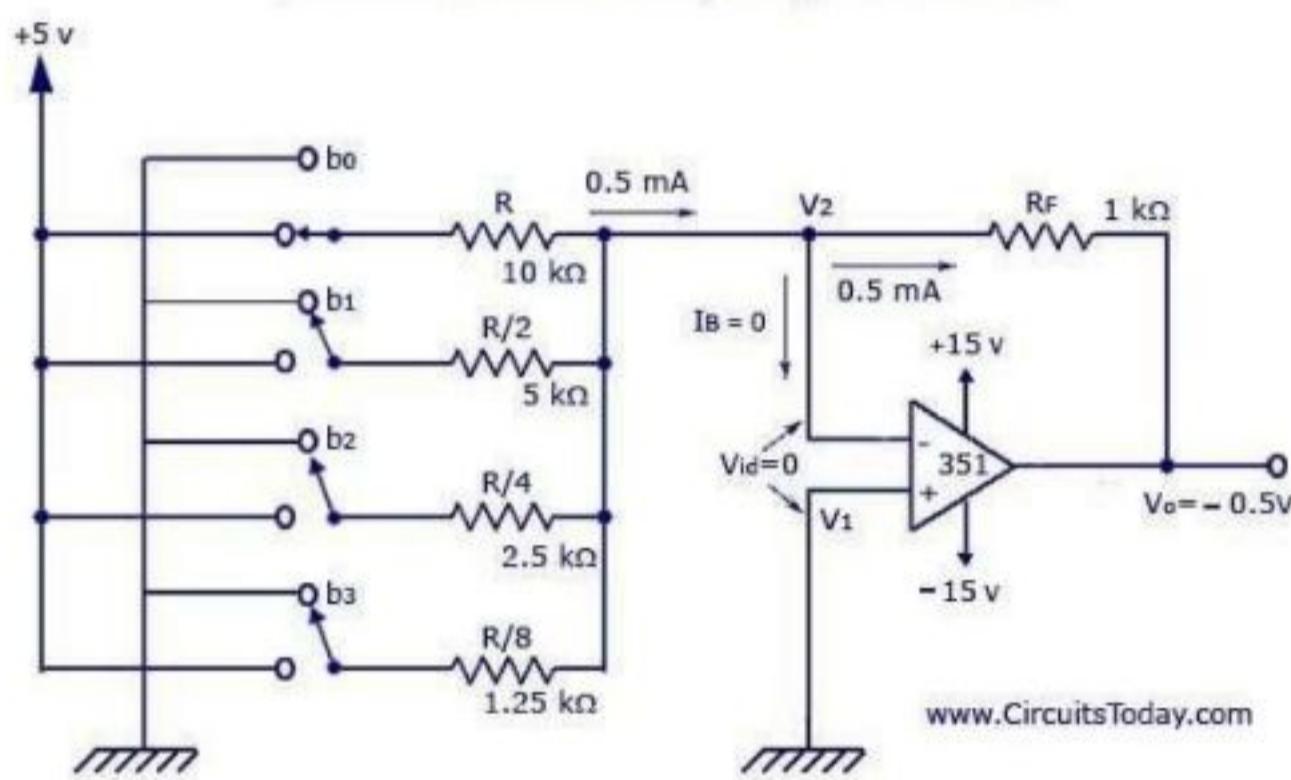
➤ **Digital-to-Analog Converter (DAC):**

- 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



20. Practical 7

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

Module 3

Aim: Study and implement a shifter.

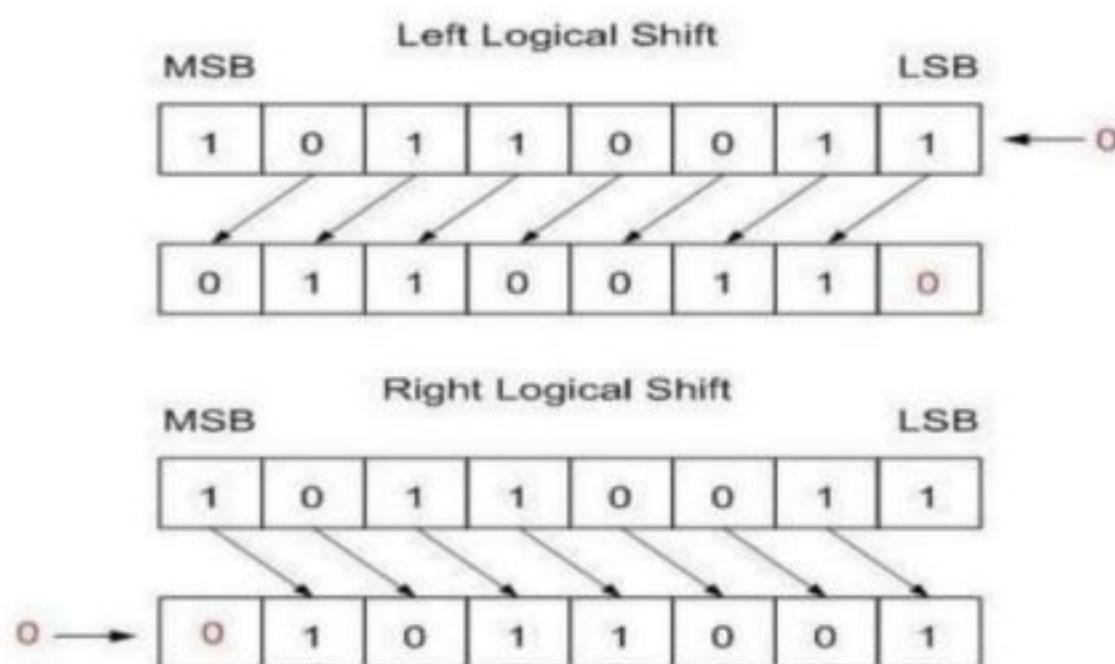
THEORY :

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	?

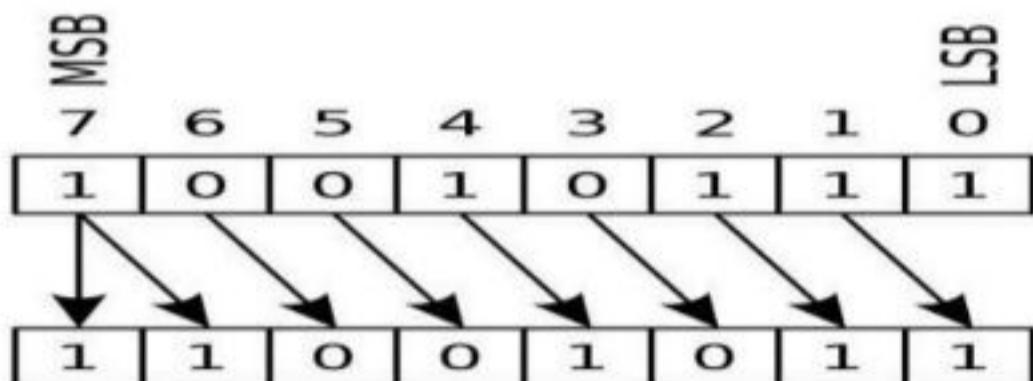
There are several kinds of commonly used shifters:-

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

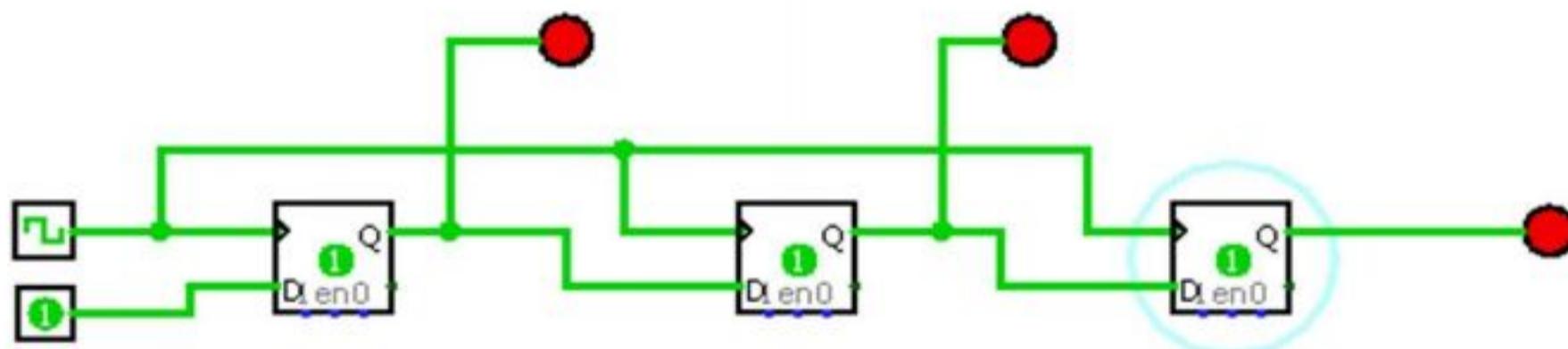


2. **Arithmetic Shifter** : is the same as a logical shifter, but on right shifts fills the most significant bits with a copy of the old most significant bit (MSB). This is useful for multiplying and dividing signed numbers Arithmetic shift left (ASL) is the same as logical shift left (LSL).

Ex: 11001 ASR 2=11110; 11001 ASL 2=00100



Enrollment No : D2D04



21. Practical 8

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

Module 3

Aim: Study and implement Flip-flops.

THEORY :

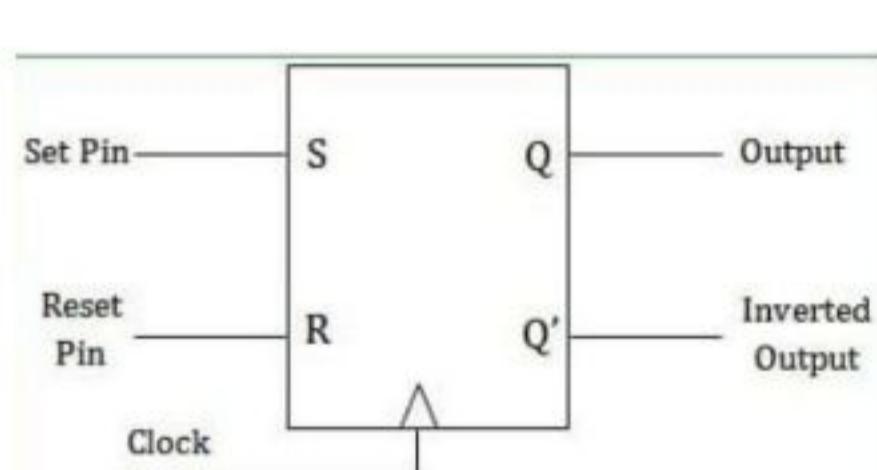
In electronics, a flip-flop or latch is a circuit that has two stable states and can be used to store state information – a bistable multivibrator. The circuit can be made to change state by signals applied to one or more control inputs and will have one or two outputs. It is the basic storage element in sequential logic.

Flip-flops and latches are 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.

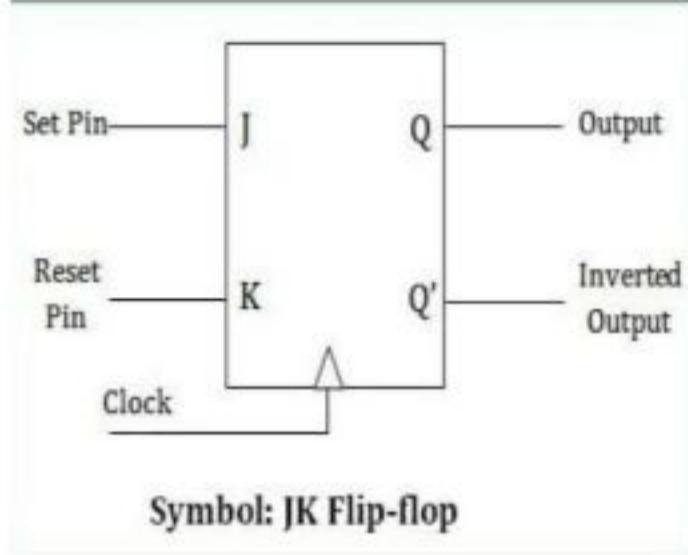
This simple flip-flop is basically a one-bit memory bistable device that has two inputs, one which will "SET" the device (meaning the output = "1"), and is labelled S and one which will "RESET" the device (meaning the output = "0"), labelled R.



Symbol: SR Flip-flop

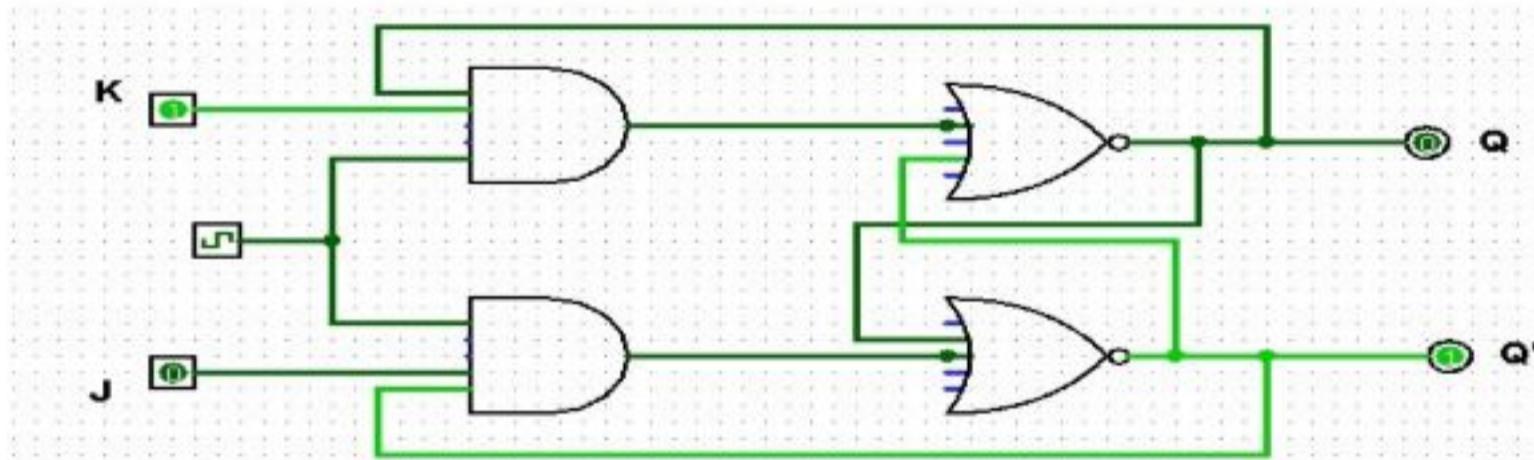
S	R	Q _n	Q _{n+1}	State
0	0	0	0	Q _n
0	0	1	1	
0	1	0	0	
0	1	1	0	
1	0	0	1	1
1	0	1	1	
1	1	0	INVALID	
1	1	1	INVALID	DON'T CARE

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.



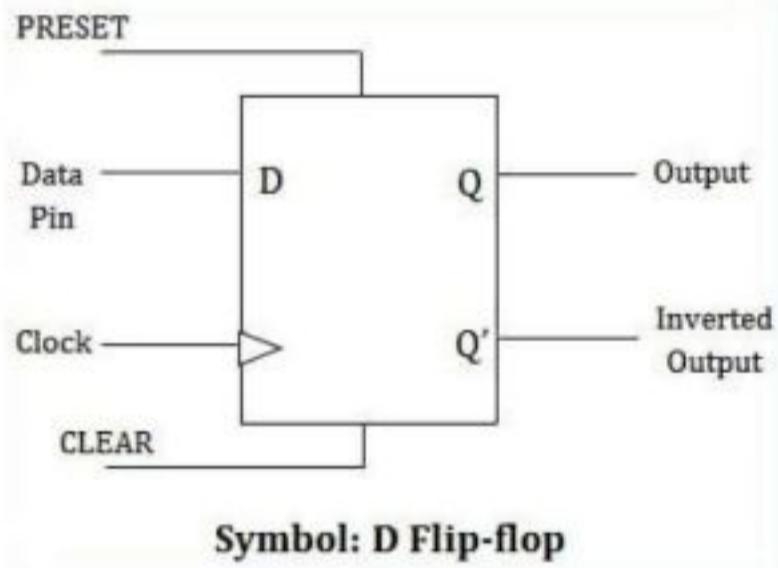
J	K	Q _n	Q _{n+1}	State
0	0	0	0	Q _n
0	0	1	1	
0	1	0	0	0
0	1	1	0	
1	0	0	1	1
1	0	1	1	
1	1	0	1	Q _{n'}
1	1	1	0	

Enrollment No :D2D04



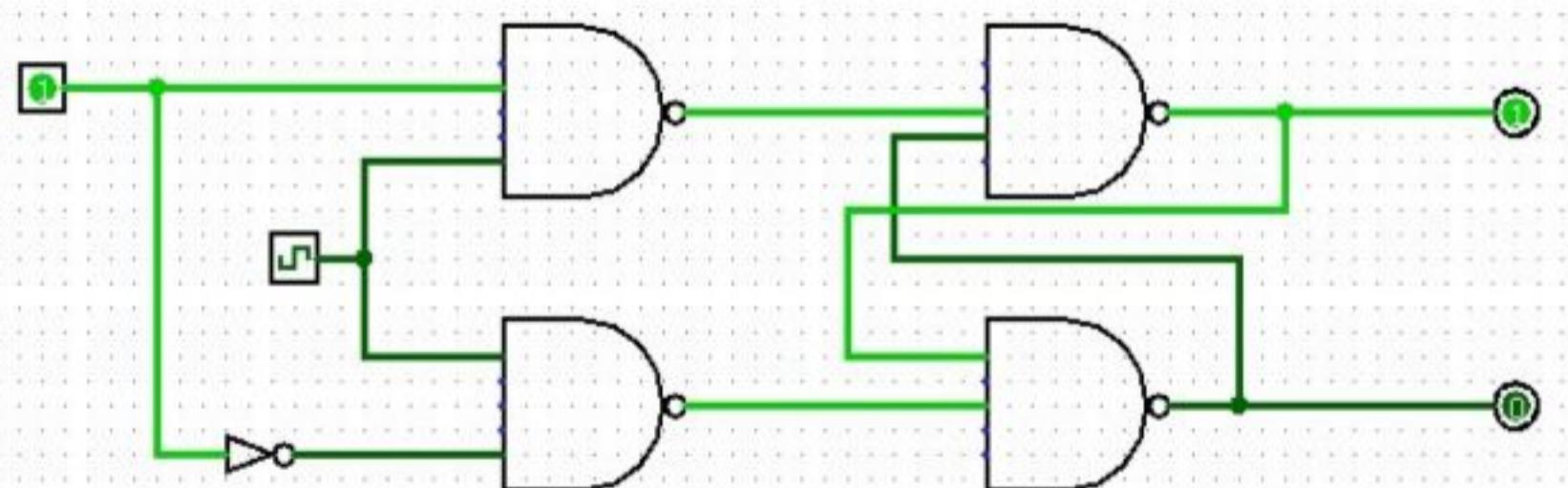
3. D flip-flop :

- 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.



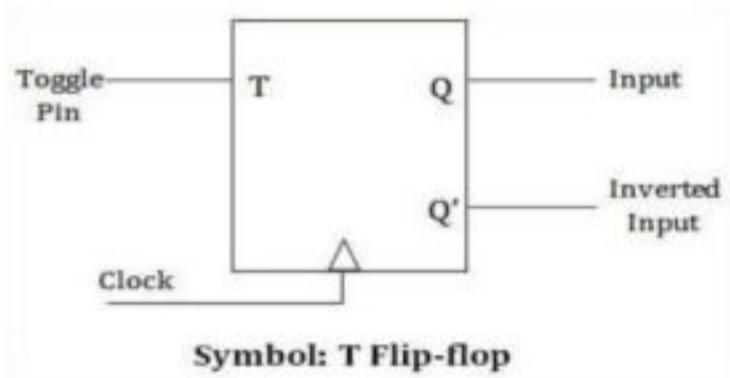
clk	D	Q	Q'
0	0	Q	Q'
0	1	Q	Q'
1	0	0	1
1	1	1	0

Enrollment No : D2D04



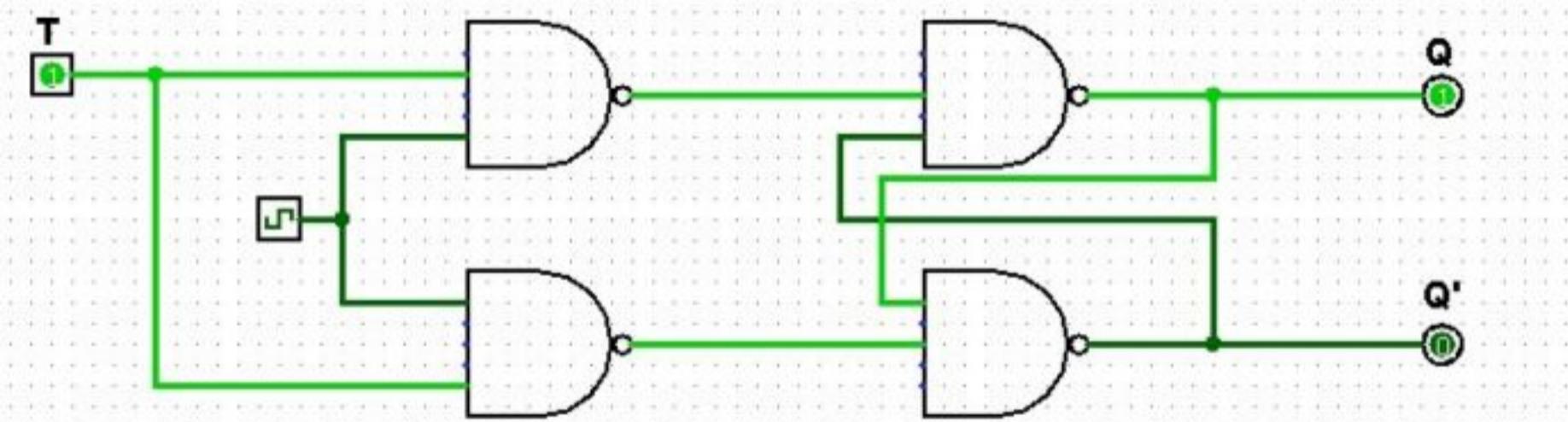
4. T flip-flop :

- 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	

Enrollment No : D2D04



22. 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 counter based on the flip-flops that are connected in synchronous or not.

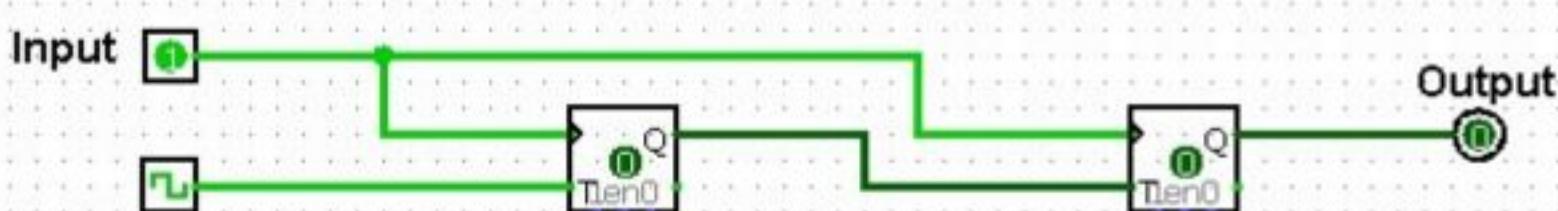
- Asynchronous counters
- Synchronous counters

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.

Clock	Counter Output		State Number	Decimal counter Output
	Q _s	Q _a		
Initially	0	0	-	0
1 st	0	1	1	1
2 nd	1	0	2	2
3 rd	1	1	3	3
4 th	0	0	4	0

Enrollment No : D2D04



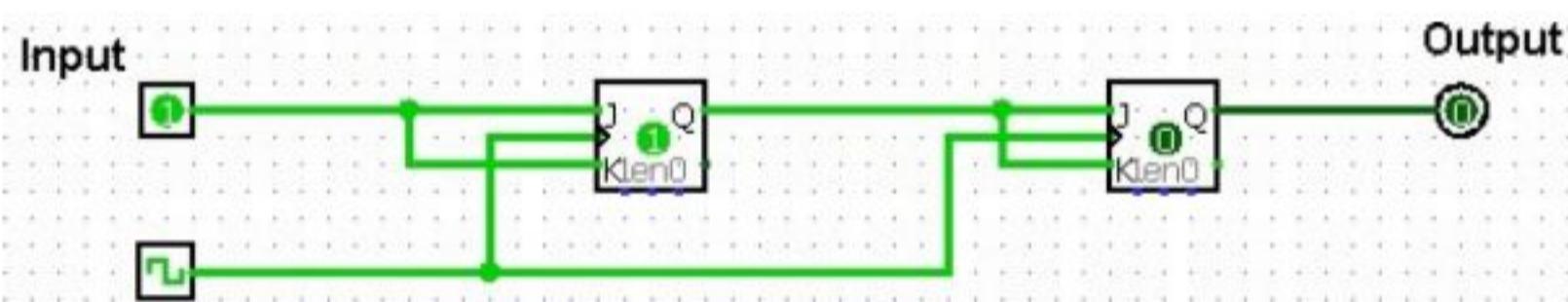
Synchronous Counters :

If the "clock" pulses are applied to all the flip-flops in a counter simultaneously, Then such a counters 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

Enrollment No : D2D04



23. 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”.

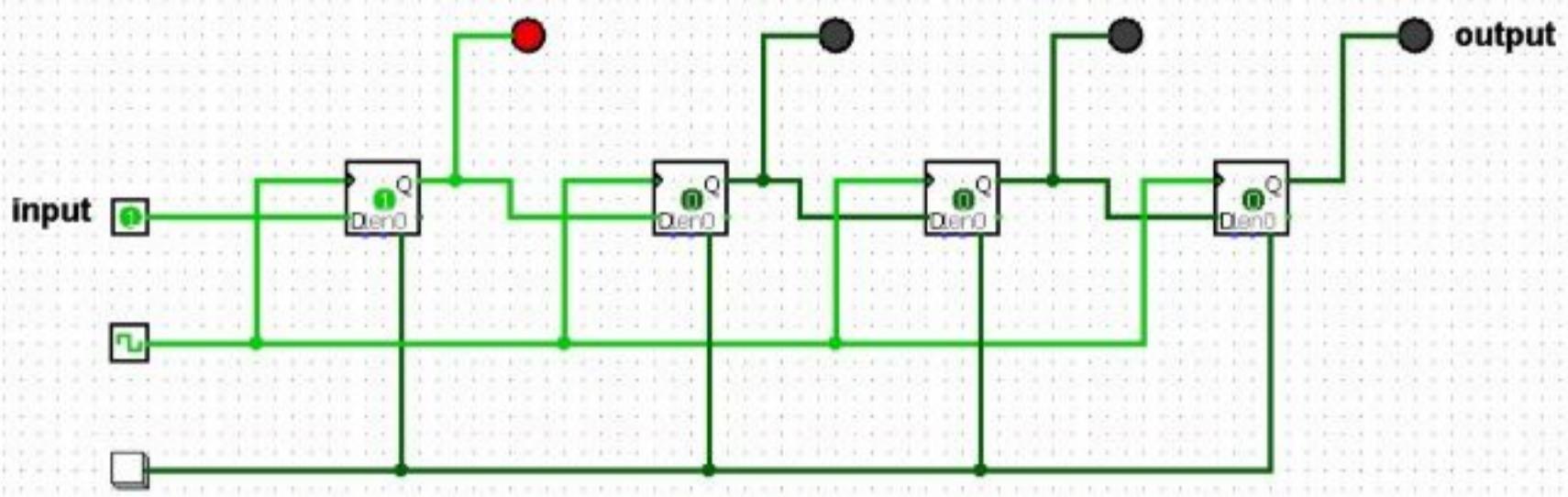
Shift registers are basically of 4 types.

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.



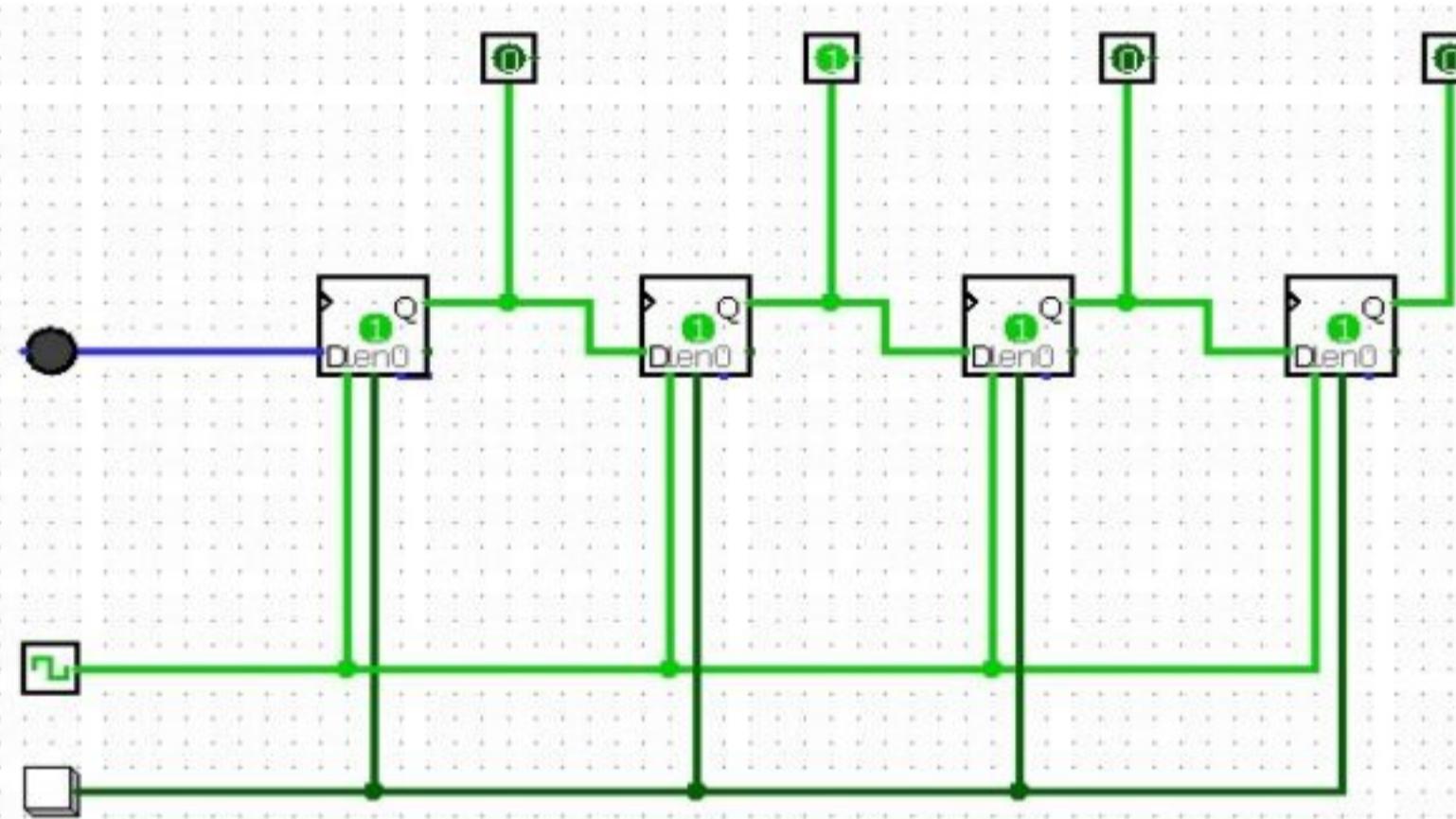
2. Serial-In Parallel-Out shift Register (SIPO) :

- 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.

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.

Enrollment No : D2D04

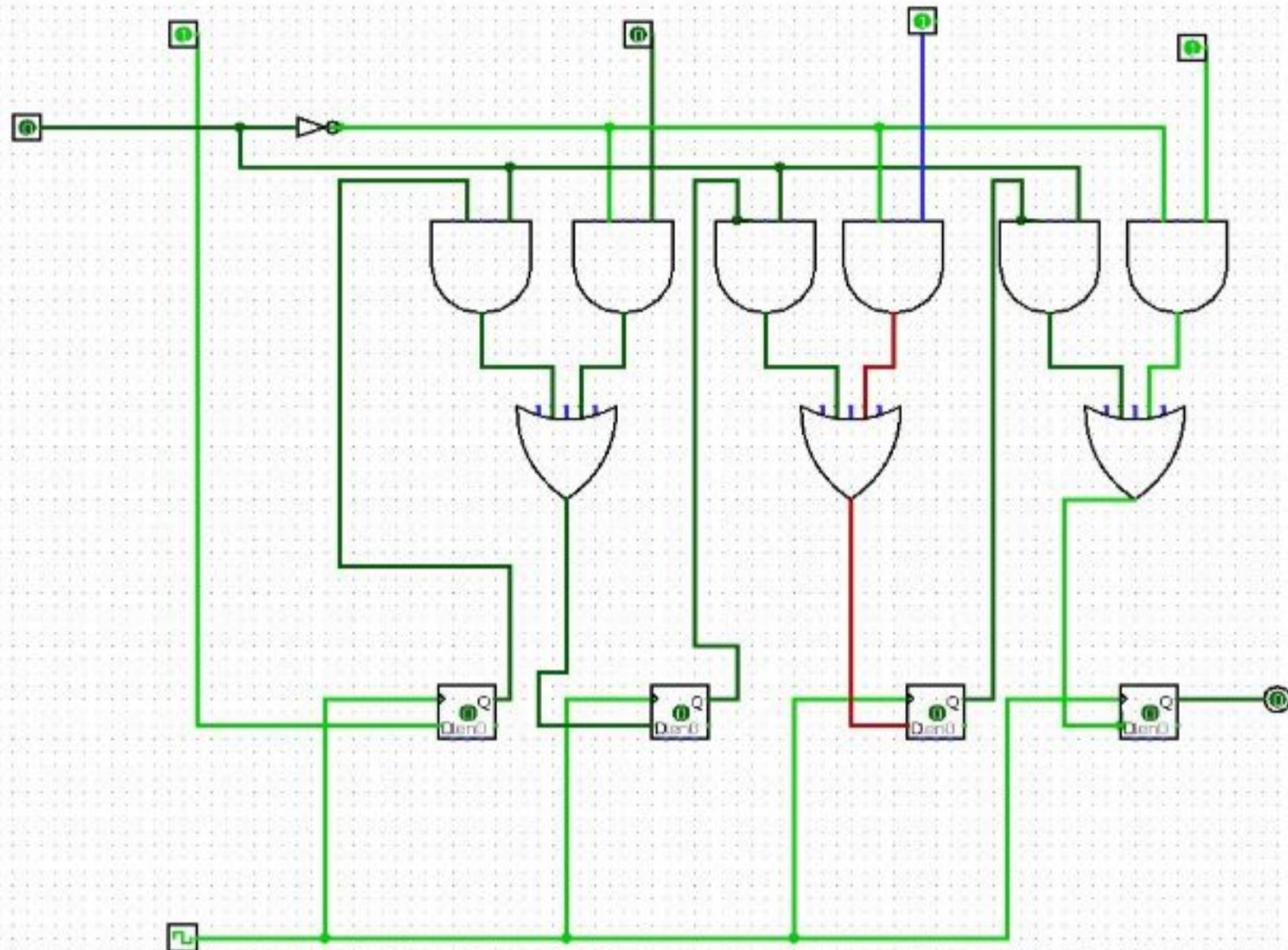


4. Parallel In Serial Out (PISO) :

- 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 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.

Enrollment No : D2D04



24. 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.

		BC	00	01	11	10
		A B	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 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

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

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

Assignment - 7

(Q1) Solve the given problem using fundamental of number systems and boolean algebra.

-1 Module 1

(Q1) Explain de morgan's theorem with truth table.

$$(Q1) \text{ Law 1: } A + B + C = \bar{A} \cdot \bar{B} \cdot \bar{C}$$

+ The complement of sum of variable equal to complement of product of variable

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

$$(Q1) \text{ Law 2: } \bar{A} \cdot \bar{B} \cdot \bar{C} = \bar{A} + \bar{B} + \bar{C}$$

+ The complement of product is equal to complement of sum of variable

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

Q2 Simplify Boolean function $f = \bar{A}B\bar{C} + \bar{A}\bar{B}\bar{C} + A\bar{B}$
 $f = \bar{A}\bar{B} + \bar{A}B\bar{C} + A\bar{B}$

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

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

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

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

$$\{f = \bar{A}C + AB\} \text{ Ans}$$

Q3 List and explain logic family

(i) Logic family circuits are

(ii) TTL (cmos) (iii) ECL

+ TTL - Transistor-transistor Logic

- TTL family is named for its dependence on transistor alone to perform basic logic operations

- The basic TTL Logic circuit is the NAND gate.

good speed, low manufacturing cost wide range of circuits and the availability in SSI and MSI are its merit.

- TTL logic family consist of several sub families

- Standard TTL, high speed TTL, low power TTL, low power Schottky TTL, advanced low power Schottky TTL, Schottky

- Advantages of TTL

(i) high speed operation, fastest among the saturated logic families. The propagation delay time is about 10 ns.

- 2) Moderate Power dissipation
- 3) Available in commercial and military versions
- 4) Available for wide range of functions.
- 5) Low cost
- 6) Moderate packaging density

- 1) Disadvantage of TTL
- (1) higher power dissipation than CMOS.
- 2) Lower noise immunity than CMOS.
- 3) Less fan-out than CMOS.

- Q4 describe error detecting and correcting.
- 1) When binary data is transmitted from one circuit to another circuit or system an error may occur. This means a signal corresponding to may change to 1 or vice-versa due to presence of noise.
 - 2) Codes which allow only error detection are called error detecting codes
 - 3) Codes which allow error detection and correction both are called error detecting and correcting codes

1) Parity Bit

- 1) A parity bit is used for the purpose of detecting errors during transmission of binary information.
- 2) A parity bit is a extra bit included with a binary message to make the number of 1s either odd or even.

- 1 The circuit that generates the parity bit in the transmitter is called a parity generator and the circuit that checks the parity in the receiver is called parity checker.
 - In even parity the added parity bit will make the total number of 1s an even amount.
 - 1 In odd parity the added parity bit will make the total number of 1s an odd amount.
- x. The received code is 1000 0001. check whether code is correctly received or not if odd parity is used
- 1 The received code has even parity hence the code is not received correctly.
- ### 1 Hamming code
- 1 Hamming code is a error detection & correction code that means this code not only provides the detection of a bit error, but also identifies which bit is in error so that it can be corrected.
 - 1 The code uses a number of parity bits to be located at certain position in the group.
 - 1 Hamming code can be constructed for single ~~and~~ error detection.

1) three steps involved in hamming code for error detection and correction

1) Number of parity bits

2) location of the parity bits in the code

3) Assigning values to parity bits.

Ex Determine the single error-correcting code for the information code 10111 for odd parity.

Step 1: Number of parity bits

$$\text{Let } p=3 \text{ then } 2^p = 2^3 = 8 \therefore x+p+1 = 5+3+1 = 9$$

This will not work, Try $p=4$ then

$$2^4 = 2^p = 16, x+p+1 = 5+4+1 = 10$$

equation $2^p \geq x+p+1$ is satisfied and hence four bits are sufficient

\therefore total code bit = 5 + 4 = 9

Step 2: construct location table for bits.

Bit designation	D ₇	P ₈	D ₇	D ₆	D ₅	P ₄	D ₃	P ₂	P ₁
bit location	9	8	7	6	5	4	3	2	1
binary number	1001	1000	0111	0110	0101	0000	0011	0010	01
information bits	1		0	1	1		1		1
parity bits		0				1		1	0

Step 3: determine the parity bits.

for p₁: 3, 5, 7 have three 1s for odd parity P₁ must be 0

for p₂: 3, 6, 7 have two 1s for odd parity P₂ must be 1.

for p₄: 5, 6, 7 have two 1s for odd parity P₄ must be 0.

for p₈: P₈ checks 5 and 9 and must be 0 for odd parity

Step 4: enter the parity bits into the table to form a nine bit hamming code = 100111110

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

(Q2) Analyze working of logic families and logic gates and design the simple circuits using various gates for a given problem.

Q3) Explain K-map.

- During the process of simplification of boolean expression we have to predict each successive step.
- we can never be absolutely certain that an expression simplified by boolean algebra alone is the simplest possible expression.
- On the other hand, the map method gives us a systematic approach for simplifying a boolean expression.
- The map method, first proposed by Veitch and modified by Karnaugh, hence it is known as the Veitch diagram or the Karnaugh map.
- The basis of this method is a graphical chart known as Karnaugh map (K-map).
- It contains boxes called cells.
- Each 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 and so on.
- Most commonly used K-maps are 1, 2, 3 and 4 variable.

A	B	D	I	DC	00	11	12	AB	D	00	01	11	10
0	0			0				00					
1	1			1				01					

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

4-variable
(16 cells)

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

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

$\{F = C\}$ Ans

Q2. Obtain the simplified expressions in sum of products for the following boolean functions.

2.1. $F(x, y, z) = E(2, 3, 6, 7)$

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

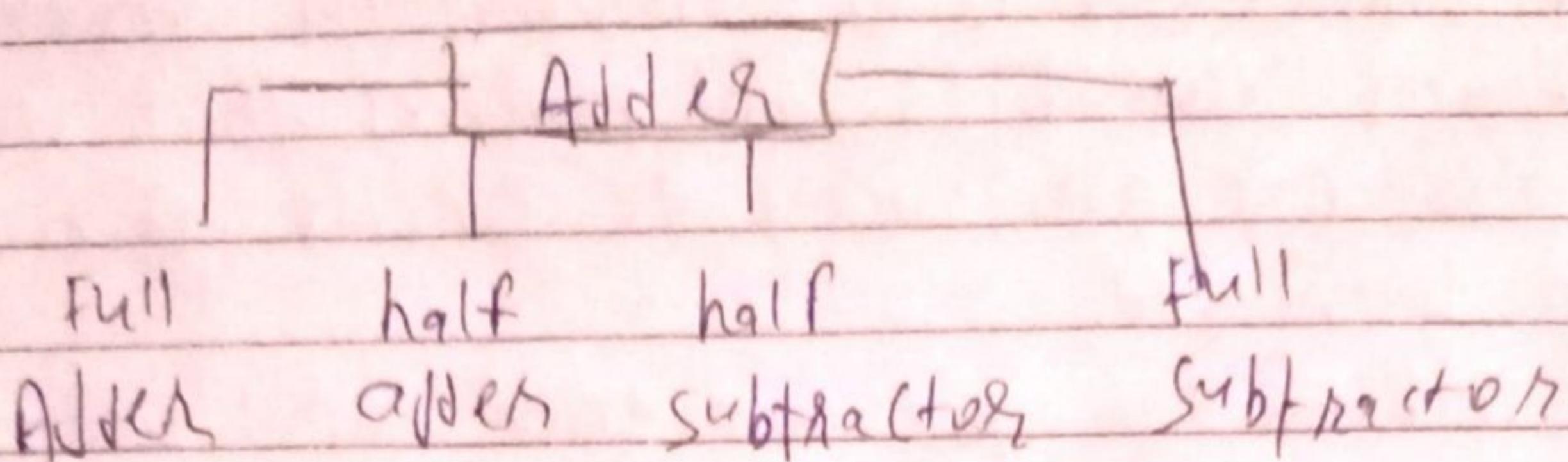
[Answer $E = \bar{B}C$]

2.2. $f(A, B, C, D) = E(4, 6, 7, 15)$

AB	00	01	11	10	11
CD	00	0	1	3	2
01	1	5	7	15	16
11	12	13	14	11	10
10	8	9	11	10	

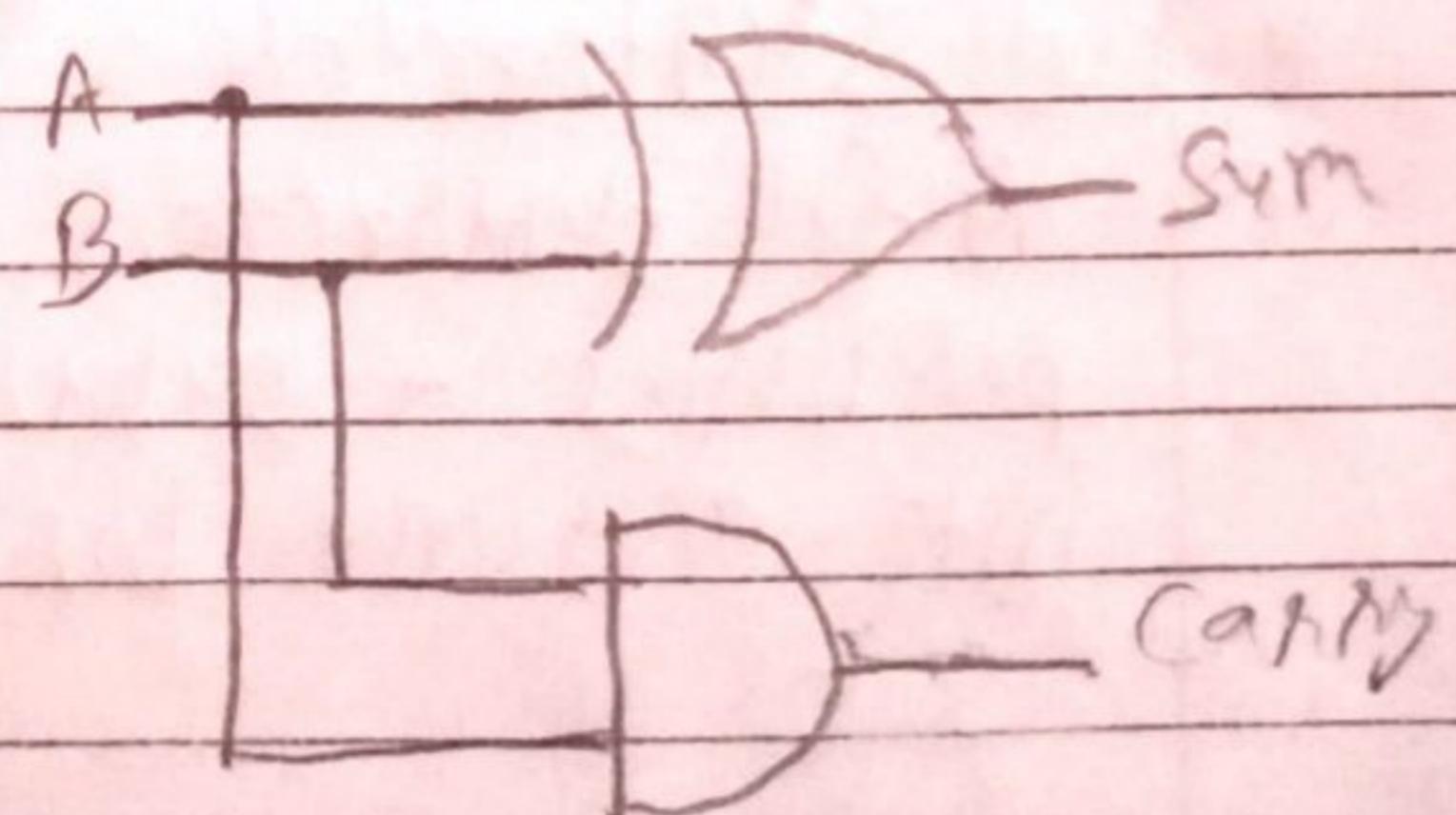
$\rightarrow [\bar{A}BCD + \bar{A}\bar{B}BCD + \bar{ABC}D]$

Q3 Describe adder and subtractor.



- 1 ~~half~~ adder - The logic circuit which performs the sum operation called half adder, and the circuit which performs addition of three bits (two significant bits and a previous carry) is a full adder.
- 1 half adder + half adder is a combinational circuit with two input and two outputs.
- 1 It returns the sum and carry of two inputs.
- 1 Truth table of half adder | Logic diagram

A	B	Sum	Carry	S. $\bar{A}B + A\bar{B}$	$\neg A \oplus B$
0	0	0	0	$\neg A \oplus B$	$\neg A \oplus B$
0	1	1	0	$\neg C = A \oplus B$	$\neg C = A \oplus B$
1	0	1	0	$\neg C = A \oplus B$	$\neg C = A \oplus B$
1	1	0	1	$\neg C = A \oplus B$	$\neg C = A \oplus B$



- 1 full adder
- 1 A full adder is a combinational circuit that forms the arithmetic sum of three inputs bits. and it consists of ~~to~~ three input and two output. Two inputs and third input will be carry from the previous layer significant position. It is same as half adder only variable changed.

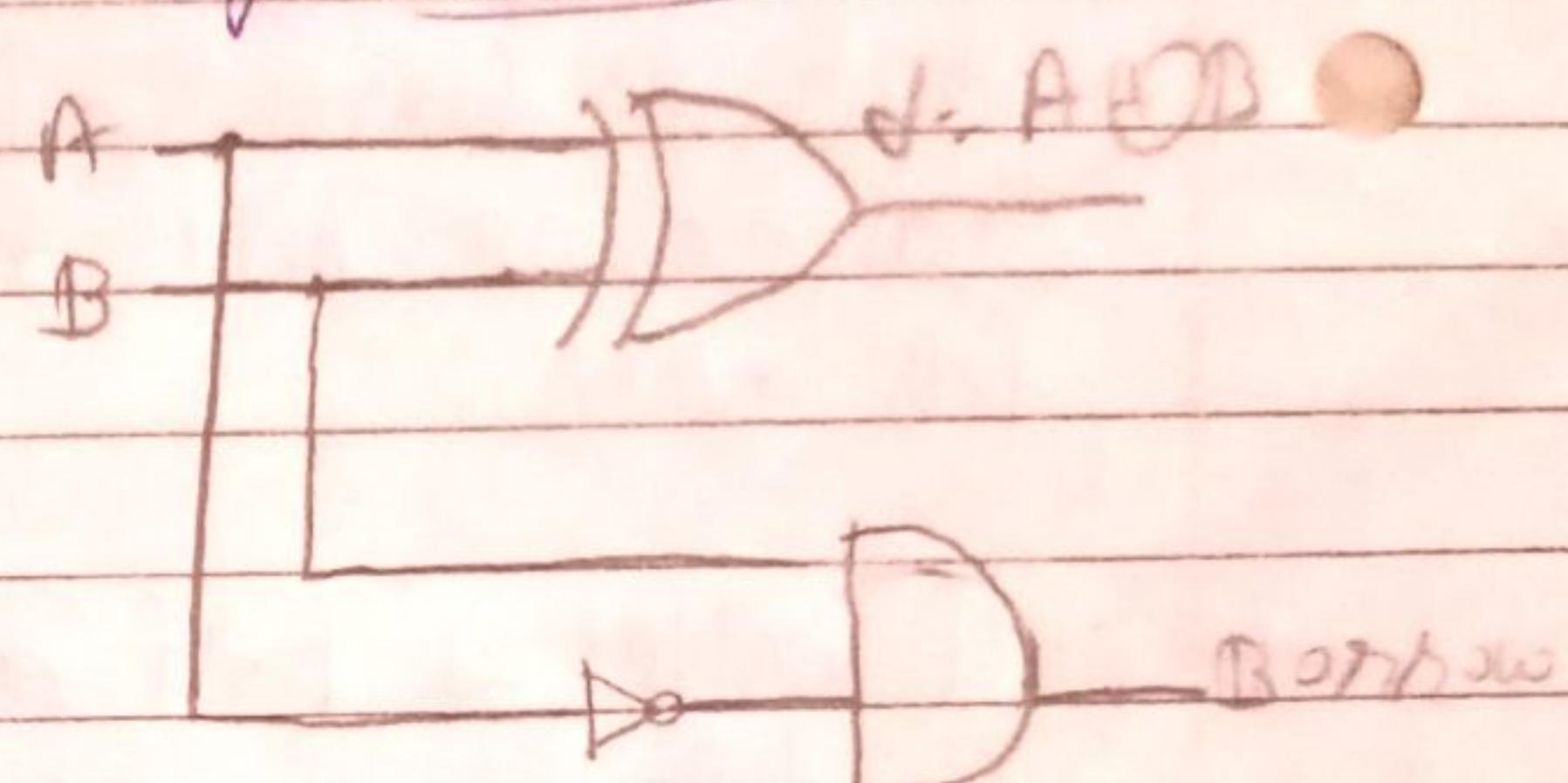
1 half subtractor

- 1 half subtractor is a combinational circuit that subtracts two bits and produces their difference.
- 1 it also has an output to specify if a 1 has been borrowed.
- 1 half subtractor has two input variables and two output variables.

Truth table

Input	Output		
A	D	J	b
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0

Logic circuit



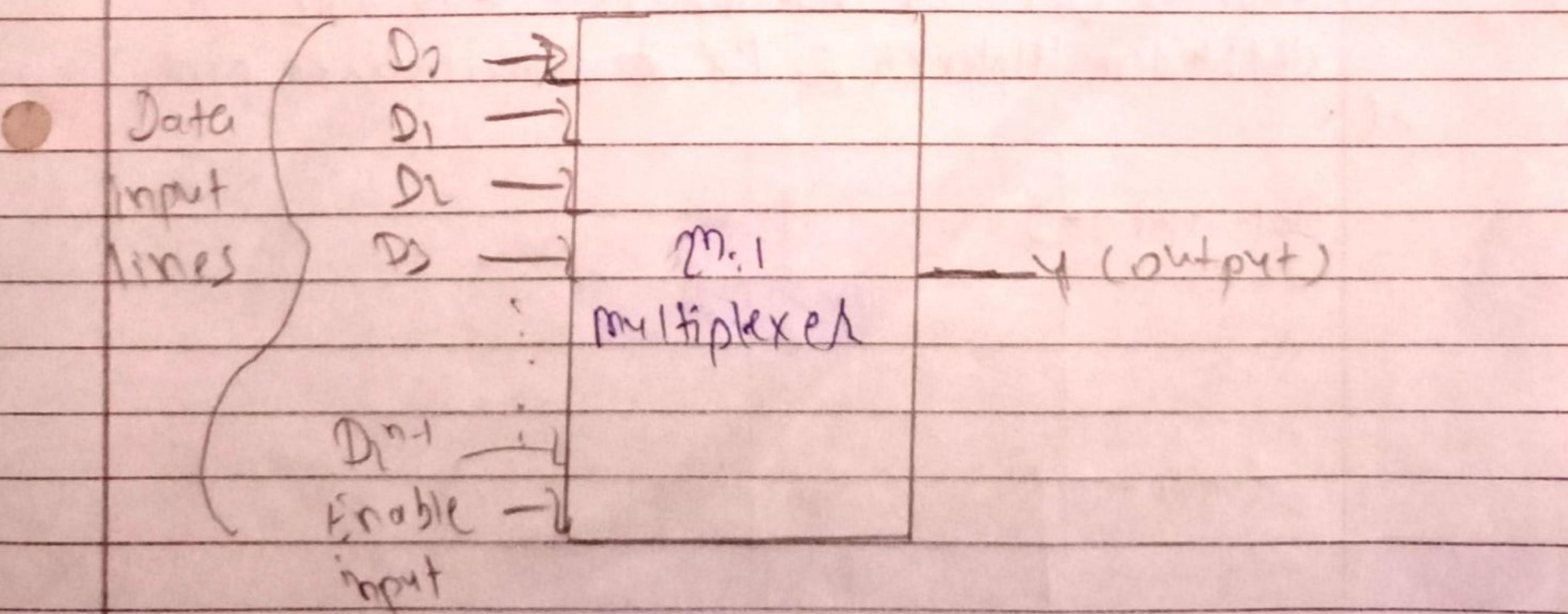
$$J = \bar{A}\bar{B} + B\bar{A} = A(\bar{B}) \oplus B \text{ and } D = \bar{A}B$$

2 full subtractor

- 1 A full subtractor is a combinational circuit that performs a subtraction between two bits, taking into account borrow of the lower significant stage.
- 1 This circuit has three inputs and two outputs.
- 1 Three inputs are minuend, subtrahend, and previous borrow respectively.
- 1 The two outputs, Difference and Bout means output borrow respectively.
- 1 This is same as half subtractor only difference is that it take three inputs.

Q) Explain multiplexers and demultiplexers.

- 1 In digital systems many times we need to select single data line from ~~several~~ several data input lines, and the data from the selected data line should be available on the output.
- 2 The digital circuit which does this task is multiplexer.
- 3 It is a digital switch. It allows digital information from several sources to be routed onto a single output line.
- 4 Multiplexer selects one of the input and routes it to output. It is also known as data selector.
- 5 Normally it contains 2^n input lines and n selection lines whose bit combinations determine which input is selected.
- 6 Multiplexer is also known as "many to one".
- 7 Various types of multiplexers available
 - 1) 2:1 multiplexer, 2 to 1, 3, 8:1, 16:1 multiplexer etc.

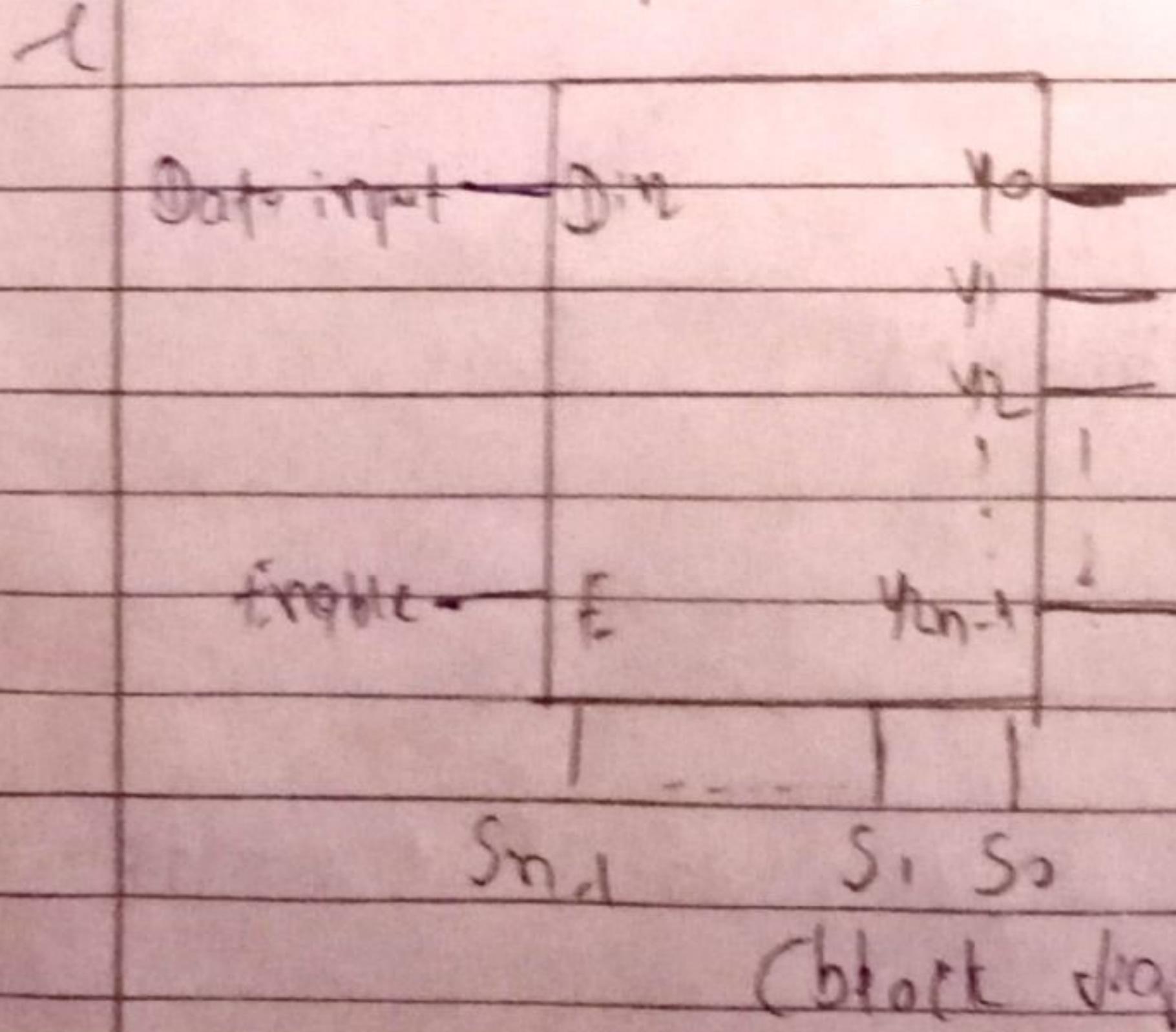


(block diagram of $2^n:1$ multiplexer)

- 1 Applications of multiplexer
- 2) They are used as a data selector to select one out of many data inputs.
- 3) They can be used to implement combinational logic circuit.
- 4) They are used in time multiplexing systems.
- 5) They are used in frequency multiplexing systems.
- 6) They are used in A/D and D/A converters.
- 7) They are used in data acquisition systems.

2 Demultiplexers

- 1 A demultiplexer is a circuit that receives information on a single line and transmits this information on one of 2^n possible output lines. This is the reverse of multiplexer.
- 2 The selection of specific output lines is controlled by the value of n selection lines.
- 3 Various types of demultiplexers available
 - (1) 1:4 demultiplexer
 - (2) 1:8 demultiplexer etc.



c Application of demultiplexer

- (1) it can be used as a decoder
- (2) it can be used as a data distributor.
- (3) it is used in time division multiplexing at the receiving end as a data separator
- (4) it can be used to implement boolean expressions.

Q5 Describe parity checkers and generators

- 1 A parity bit is used for the purpose of detecting errors during transmission of binary information.
- 2 A parity bit is an extra bit included with a binary message to make the number of 1s either odd or even.
- 3 The message including the parity bit is transmitted and then checked at the receiving end for errors. An error is detected if the checked parity does not correspond with the one transmitted.
- 4 The circuit that generates the parity bit in the transmitter is called a parity generator.
- 5 The circuit that checks the parity in the receiver is a called parity checker.
- 6 In even parity, the added parity bit will make total number of 1s an even amount.
- 7 In odd parity, the added parity bit will make the total number of 1s an odd amount.

1 parity checker

- 1 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 circuit checks for possible errors in the transmission.
- If even parity, then received bits must have even number of ones or zero when it contains odd number of 1s.
- Truth table for even parity checker.

Decimal equivalent	Four bits received	Parity check	check + the output of parity checker is denoted by $P_E C$ (parity error checker). It will equal to 1 if error occurs, that is if the four bits received have an odd number of 1s.
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	

- IC 74LS180 - Parity generator / checker
- The 74LS180 is a 9-bit parity generator or checker commonly used to detect errors in high speed data transmission or data retrieval systems.

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

(Q3) Design and Implement Combinational and Sequential circuits and verify its working.

1) Module 3

Q) Differentiate sequential and combinational circuits.

Ans.

Combinational circuits

(1) In combinational circuits, the output variables are at all times dependent on the combination of input variables.

Sequential circuits

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) Memory unit is not required in combinational circuits

2) Memory unit is required to store the past history of input variables in the sequential circuit.

3) Combinational circuit are faster in speed because the delay between input and output is due to propagation delay of gates.

3) Sequential circuit are slower than the combinational circuit.

4) Combinational circuits are easy to design.

4) Sequential circuits are comparatively harder to design.

5) Parallel adder is a combinational circuit.

5) Serial adder is a sequential circuit.

Q2 List and explain flip-flops

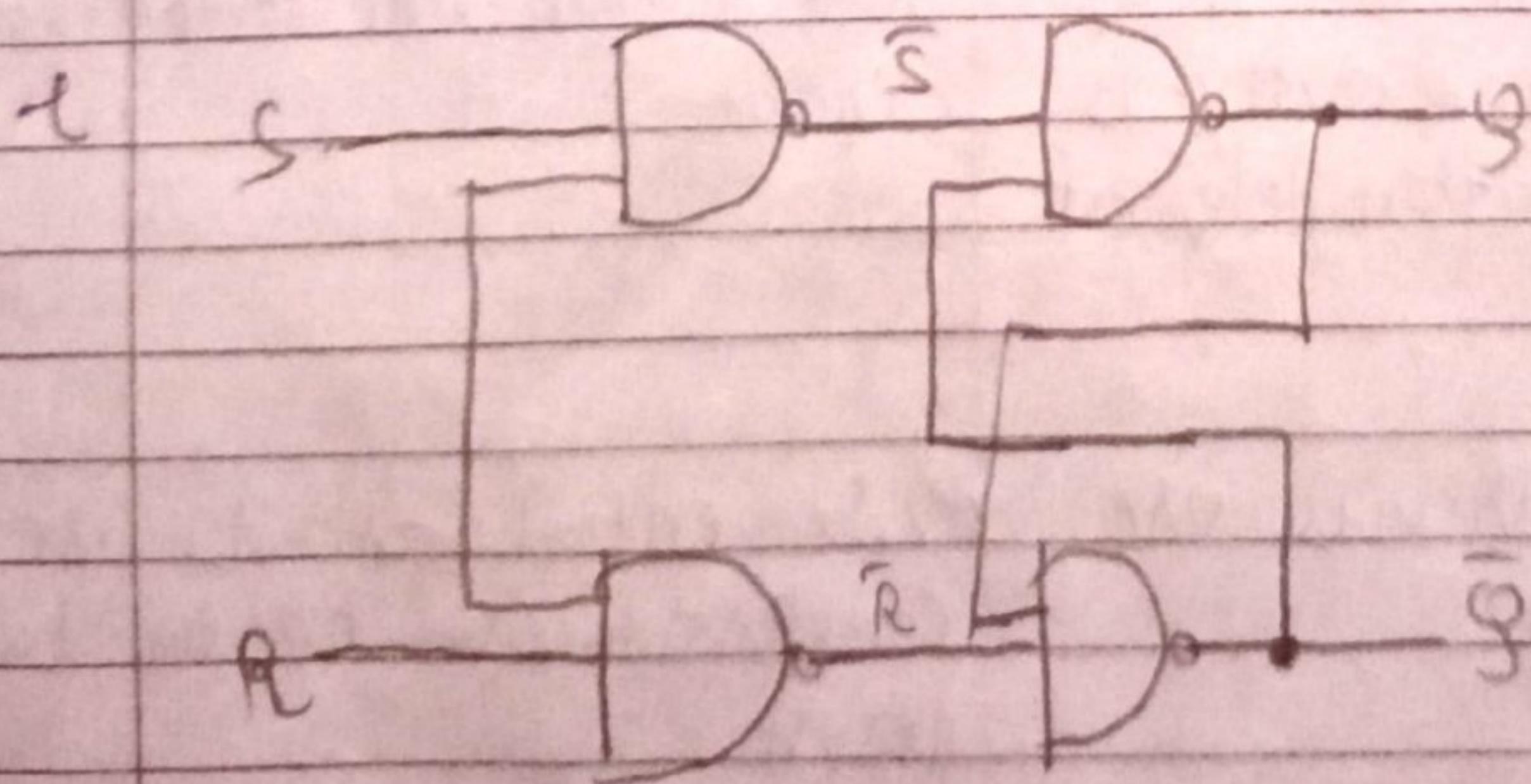
A circuit that has two stable states is treated as a flip-flop. These stable states are used to store binary data that can be changed by applying varying inputs.

1. The flip-flop are the fundamental building blocks of the digital system.

- 1. Flip-flop and latches are example of digital storage element.
- 2. In the sequential logic circuit, the flip-flop are the basic storage element.

1 (1) SR-Flip flop.

The SR flip flop is the most common flip flop is used in the digital system. In SR flip flop, when the Set input "S" is high, the output Y will be high, and it will be low. It is required that the wiring of the circuit is maintained when the outputs are established. We maintain the wiring until set or reset input goes high, or power is shutdown.



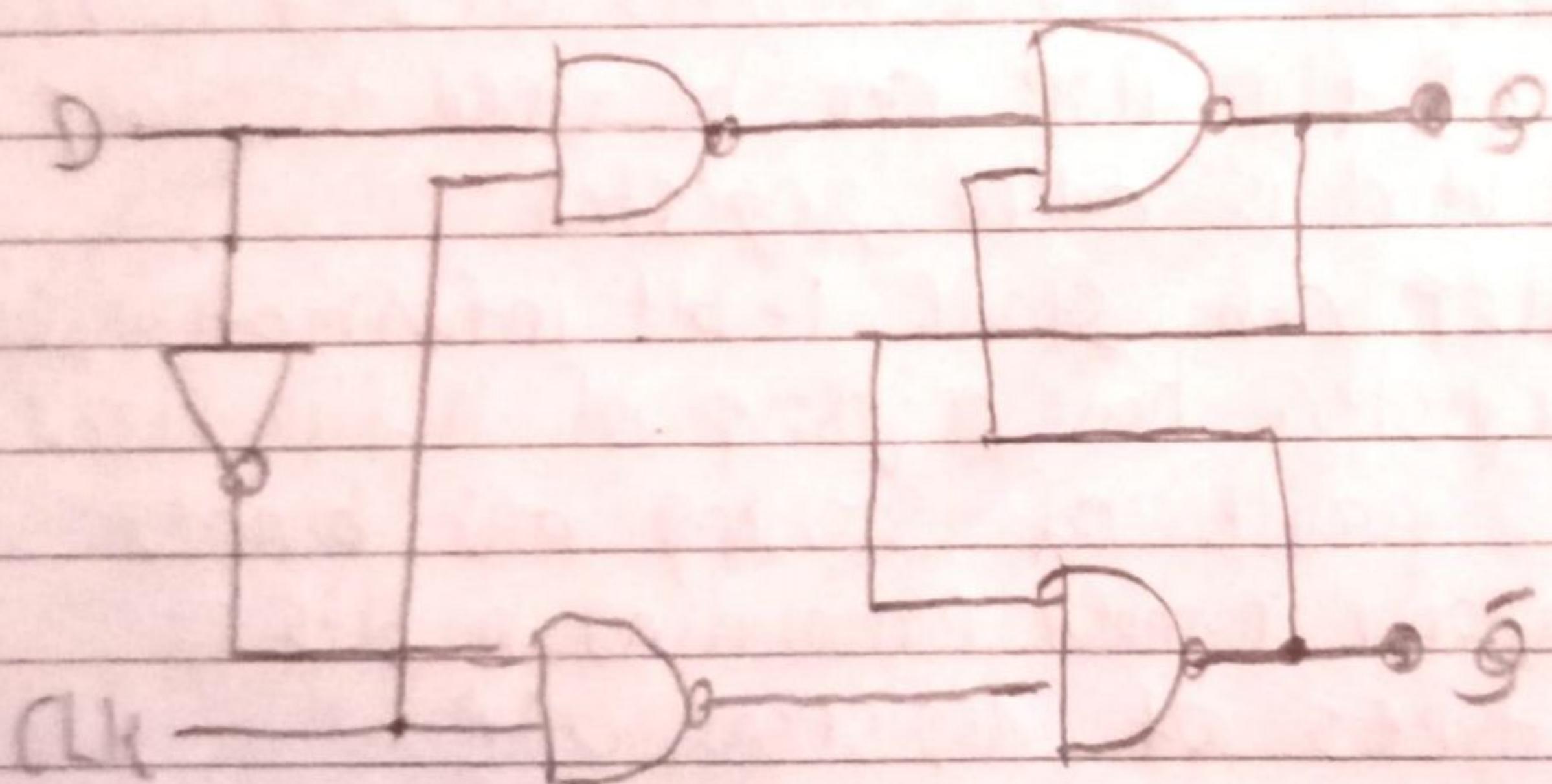
(S-R flip flop using NAND Gates)

1 Truth table.

S	R	Y	Y'
0	0	0	1
0	1	0	1
1	0	1	0
1	1	0	0

2 D flip flop

D flip flop is a widely used flip flop in digital systems. The D flip flop is mostly used in shift-registers, counters and input synchronization.



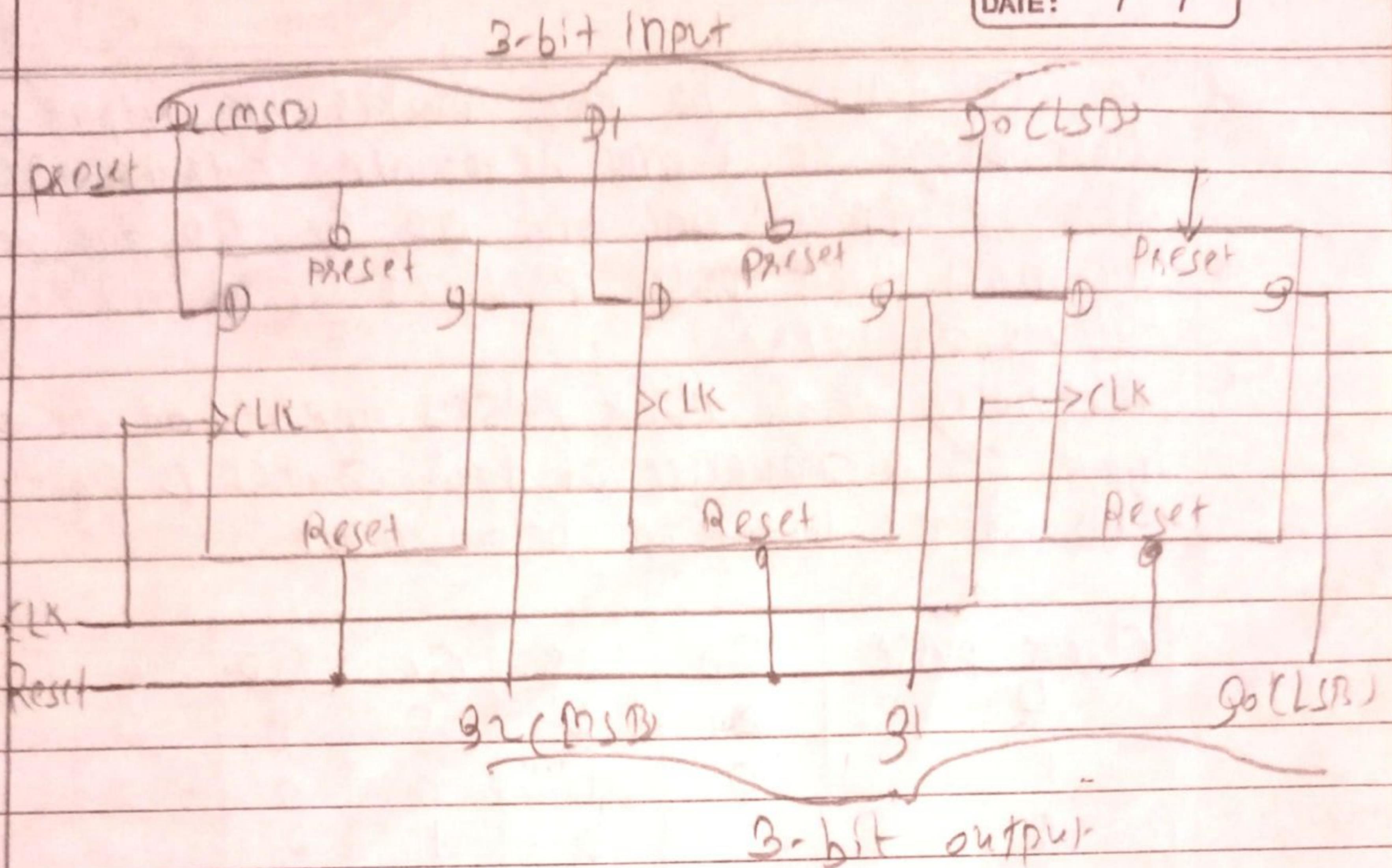
1 Truth table

clock	D	Y	Y'
↓ = 0	0	0	1
↑ = 1	0	0	1
↓ = 0	1	0	1
↑ = 1	1	1	0

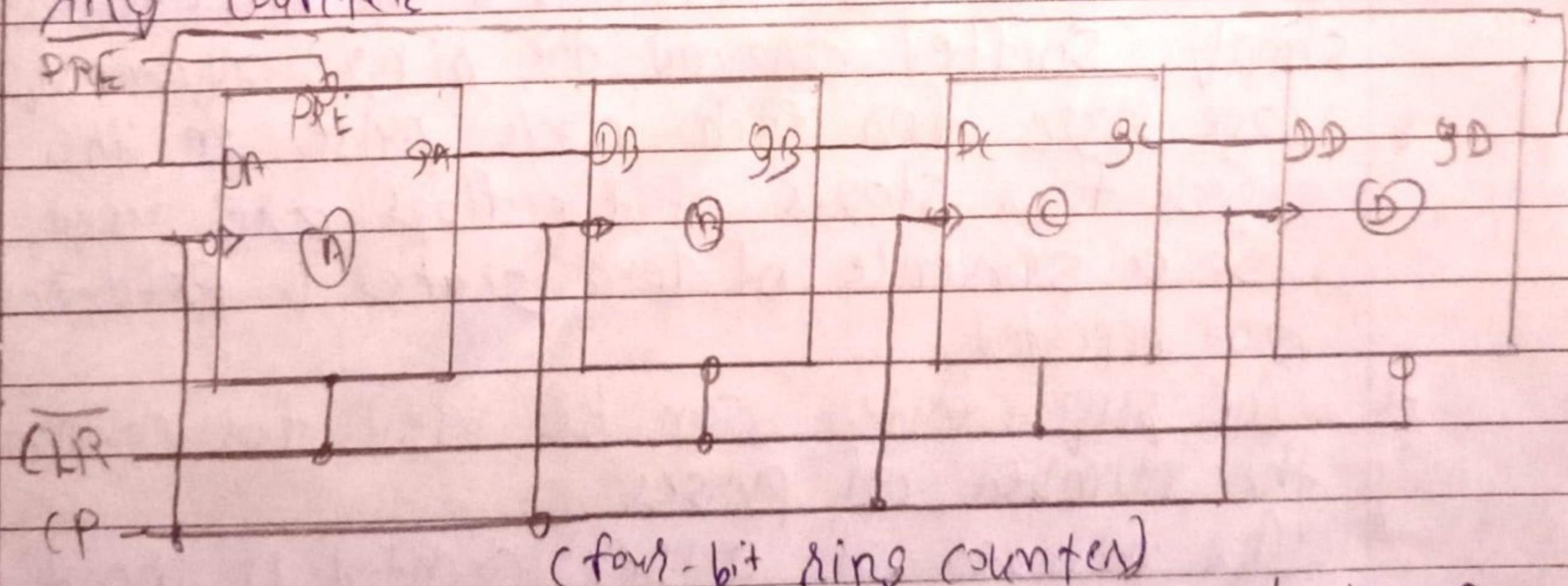
Q3) List and explain registers

A group of flip-flops connected together forms a register.

- c) A register is used solely for storing and shifting data which is in the form of 1s and 0s entered from an external source.
- c) Reset, Reset and clock inputs are connected in parallel and a bit input is applied to D inputs of the flip-flop.
- c) During positive edge of the clock, D inputs of the flip-flop are stored within the flip-flop and are available at the outputs of D-flip-flop.
- c) A group of flip-flop can be used to store a word which is called register.
- c) A flip-flop can store 1-bit information so an n-bit register has a group of n flip-flops and is capable of storing any binary information / number containing n-bits.
- 1) Various types of registers are
 - a) Buffer register
 - b) Controlled buffer register
- L 3-bit register using 3 D-flip-flops shown below.



Qn Explain Ring counter.
Ans The below figure shows the logic diagram of a four-bit 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 inputs of first stage

- The CLR 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.
- According to the clock pulses applied at the clock input CP, a sequence of four states is produced. This states listed in below table.

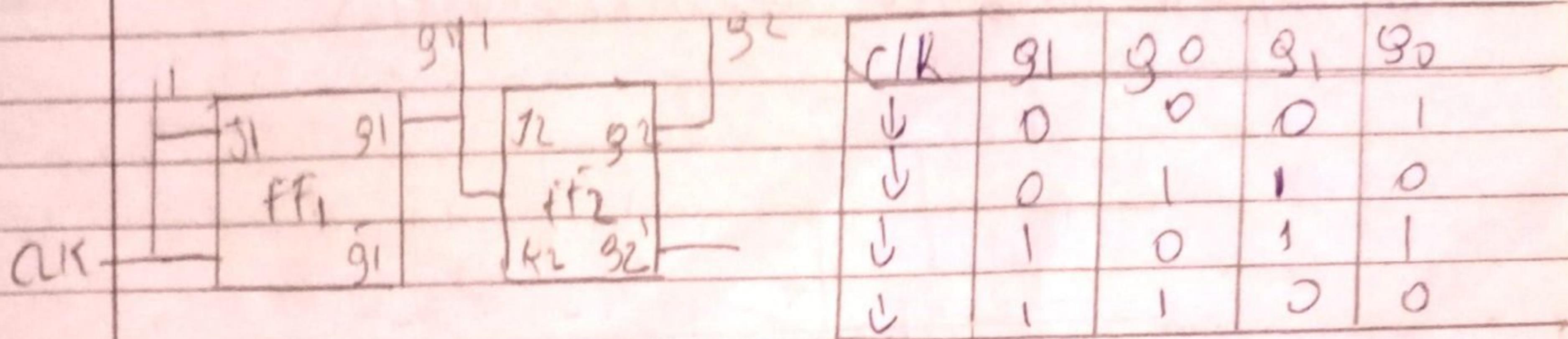
Clock pulse	Q _A	Q _B	Q _C	Q _D
0	1	0	0	0
1	0	1	0	0
2	0	0	1	0
3	0	0	0	1
4	1	0	0	0

1. Seas (Ring counter sequence 4-bits)

- 1 is always ~~after~~ retained in the counter and simply shifted around the ring, advancing one stage for each clock pulse. In the case four stages of flip-flops are used. So a sequence of four states is produced and repeated.

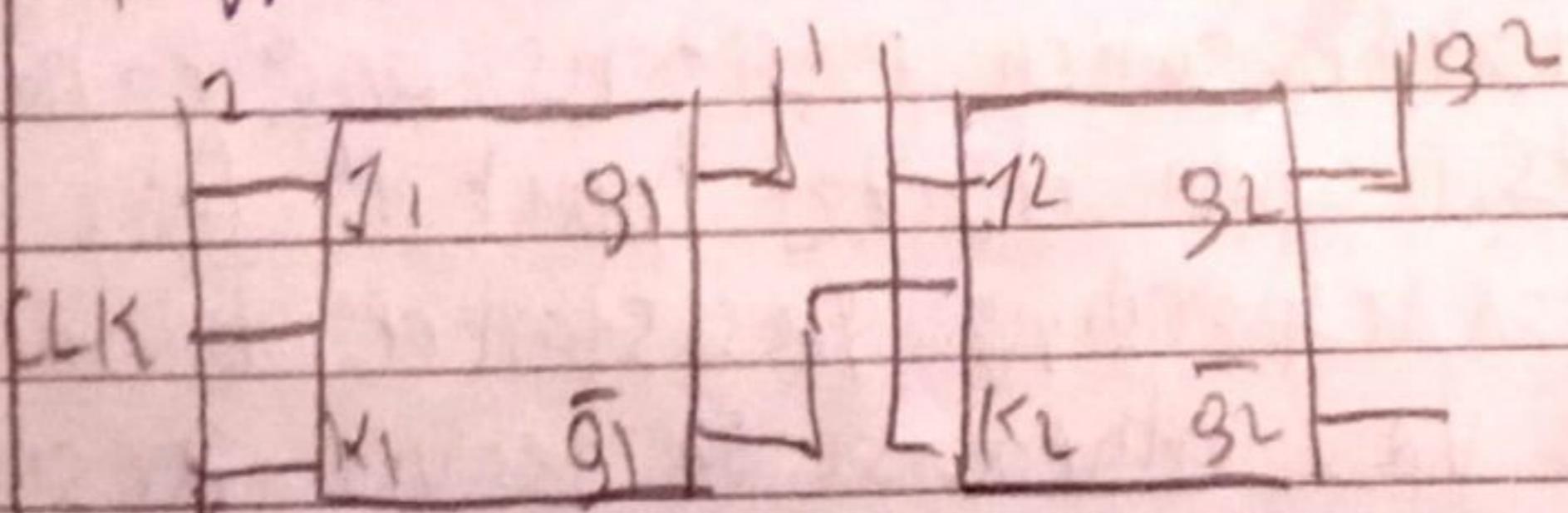
- The ring counter can be used for counting the number of pulses.
- The number of pulses counted is read by noting which flip-flop is in state 1.
- No decoding circuitry is required.
- Ring counters can be instructed for any desired mod number, that is mod-N ring counter requires N flip-flops.

Q5 Describe how to design counters using flip-flops
 (i) 2-bit ripple down counter using negative edge triggered
 FLIP-FLOP.



- 1 The 2 bit up counter in the order 0, 1, 2, 3; 0
- 2 The counter is initially reset to 00.
- 3 When the first clock pulse is initialized FF1 toggles at the negative edge of this pulse therefore Q_1 goes from low to high

Q6 2 bit ripple down counter using negative edge triggered ff.



- 1 At the negative edge of the first clock pulse FF1 toggle so Q_1 goes from 0 to 1 and \bar{Q}_1 goes from 1 to 0.
- 2 This negative edge Q_1 applied to the clock pulse of FF1 toggle FF2 and therefore Q_2 goes from 0 to 1. So one clock pulse $Q_{L=1}$ and $Q_{1,1} = 1$ then the state of the Counter is 11

12. Assignment 4

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

Module 4

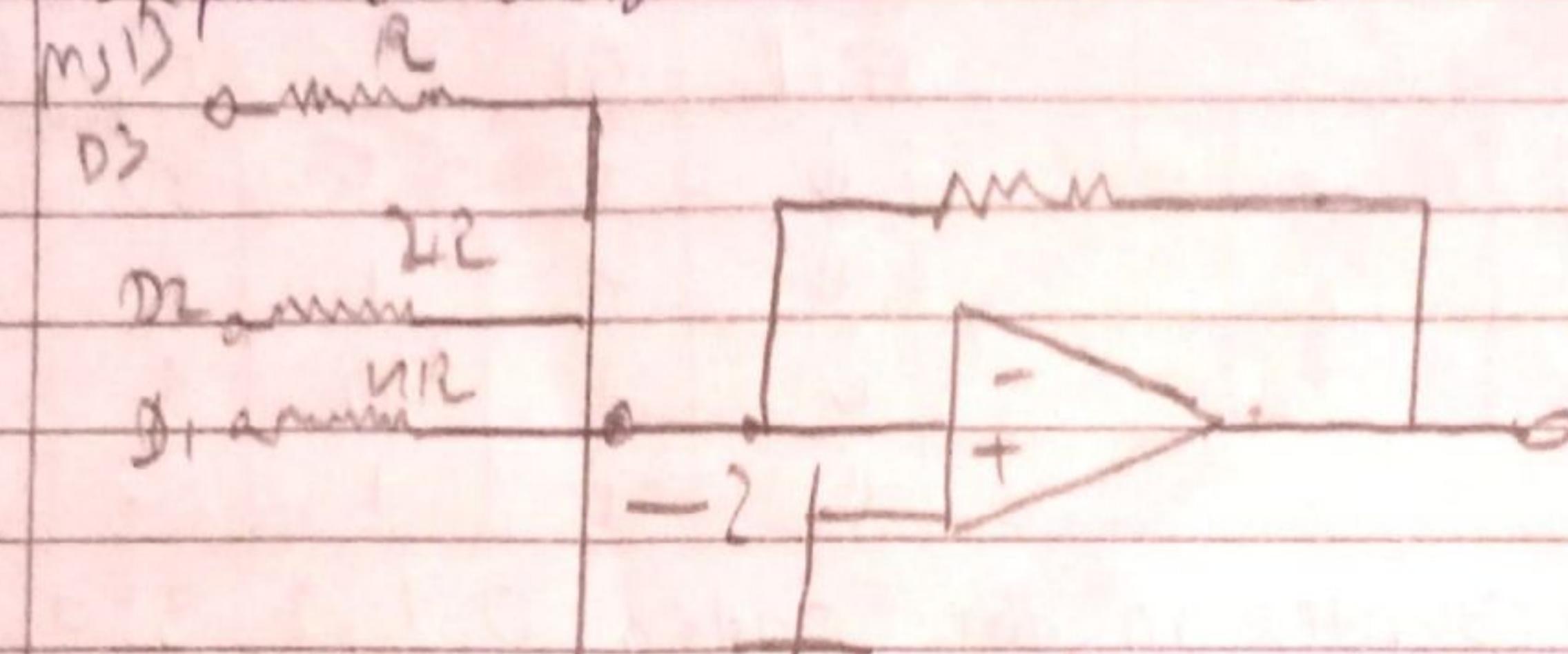
1. Explain weighted resistor/converter.
2. Explain R-2R Ladder D/A converter.
3. Describe specification of A/D & D/A converter.
4. Explain quantization and encoding.
5. Explain parallel comparator A/D converter.

Assignment 4

(e) examine the process of analog to digital conversion and digital to analog conversion

-1 Module b

Q1 Explain weighted resistor converters



$$V_{\text{out}} = (D_3 + \frac{1}{2} D_2 + \frac{1}{4} D_1 + \frac{1}{8} D_0) \frac{R_s}{R}$$

-1 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 resistors $2^0 R, 2^1 R, 2^2 R, \dots 2^n R$ as shown in above figure.

-1 As shown in the figure switch positions are controlled by the digital inputs. When digital input is logic 1, it connects the corresponding resistance to the reference voltage V_R ; otherwise it leaves resistor open. Therefore,

for ON switch if V_R and R

for off - switch $I = 0$

-1 The output voltage is the voltage across R_f and it's given as

$$V_o = I_1 R_f = \frac{V_R}{2^0 R} - (I_1 + I_2 + I_3 + \dots + I_n) R_f$$

When $R_f = R_v$ V_o is given as

$$V_o = -V_R (b_0 2^{-1} + b_1 2^{-2} + b_2 2^{-3} + \dots + b_n 2^{-n})$$

1 Drawbacks

- 1) wide range of resistor values are required. For 8-bit DAC, the resistors required are $2^0 R, 2^1 R, 2^2 R, 2^3 R, \dots$ and $2^8 R$. therefore, the largest resistor is 256 times the smallest one.
- 2) the finite resistance of the switches distributes the binary-weighted relationship among the various currents, particularly in the most significant bit positions, where the current setting resistances are smaller.
- 3) All these drawbacks especially the requirement of wide range of resistors restricts the use of binary weighted resistor DACs below 8-bits.

Q2 Explain R-2R ladder DAC converter.

- 1) R2R ladder consists of two types
 - (i) Inverter R2R ladder (current steering mode) DAC converter
 - (ii) R2R ladder (voltage switching mode) DAC converter

1 (i)

(i) Inverter R2R ladder.

- 1) R2R ladder DAC converter uses only two resistor values. This avoid resistance spread drawbacks of binary weighted DAC converter.
- 2) Like binary weighted resistor DAC, it also uses short resistors to generate n binary weighted currents; however it uses voltage scaling and identical resistors instead of resistors scaling and common voltage

reference used in binary weighted resistor DAC. Voltage scaling requires an additional set of voltage dropping series resistance between adjacent nodes.

- 1 The inverted R12R ladder DAC works on the principle of summing currents and it is also said to operate in the current steering mode.
- 2 An important advantage of the current mode is that all ladder node voltages remain constant with changing input codes, thus avoiding any shutdown effects by stray capacitances.

-e R12R ladder (Voltage switching mode) DA

- + in this type, reference voltage is applied to one of the switch positions and other switch position is connected to ground

1 Advantages of R12R ladder DACs

- (1) Easier to build accurately as only two precision metal film resistors are required.
- 2 Number of bits can be expanded by adding more selection of same R12R values.
- 3 In inverted R12R ladder DAC, node voltage remain constant with changing input binary words. This avoids any shutdown effects by stray capacitances.

Q3. Describe specification of A/D and D/A converter.

1) Specification of ADC

(1) Resolution

Resolution is also defined as the ratio of a change in the value of input voltage, V_i , needed to change the digital output by 1 LSD. If the full scale input voltage requires to cause a digital output of all 1's is V_{IFS} then resolution can be given as

$$\text{Resolution} = \frac{V_{IFS}}{2^n - 1}$$

2) Gain and offset Drifts

The gain drift is defined as a change in the full-scale transition voltage measured over the entire operating temperature range. It is usually expressed in parts per million per degree Celsius (ppm/°C)

The offset limit is defined as a change due to temperature in the analog zero for an A/D converter operating in bipolar mode. It is also expressed in parts per million per degree Celsius (ppm/°C)

B) Differential Non-Linearity (DNL)

DNL is the maximum of the difference in the each conversion current code width ($C(W)$) and the ideal code width ($I(W)$).

(iv) Conversion Time (Settling Time)

It is an important parameter for ADC.

It is defined as the total time required to convert an analog signal into its digital output. It depends on the conversion technique used and the propagation delay of circuit components.

Q4. Explain quantization and encoding

- 1 In a digital to analog converter, the possible number of digital inputs is n.
- 2 for example in a 3-bit D/A converter there are 8 possible outputs.

Quantization & The digitization of analog signals involves the rounding off the values which are approximately equal to analog values.

The method of sampling chooses a few points on the analog signal and then these points are joined to round off the value to a near stabilized value. Such a process is called as quantization.

Two types of quantization

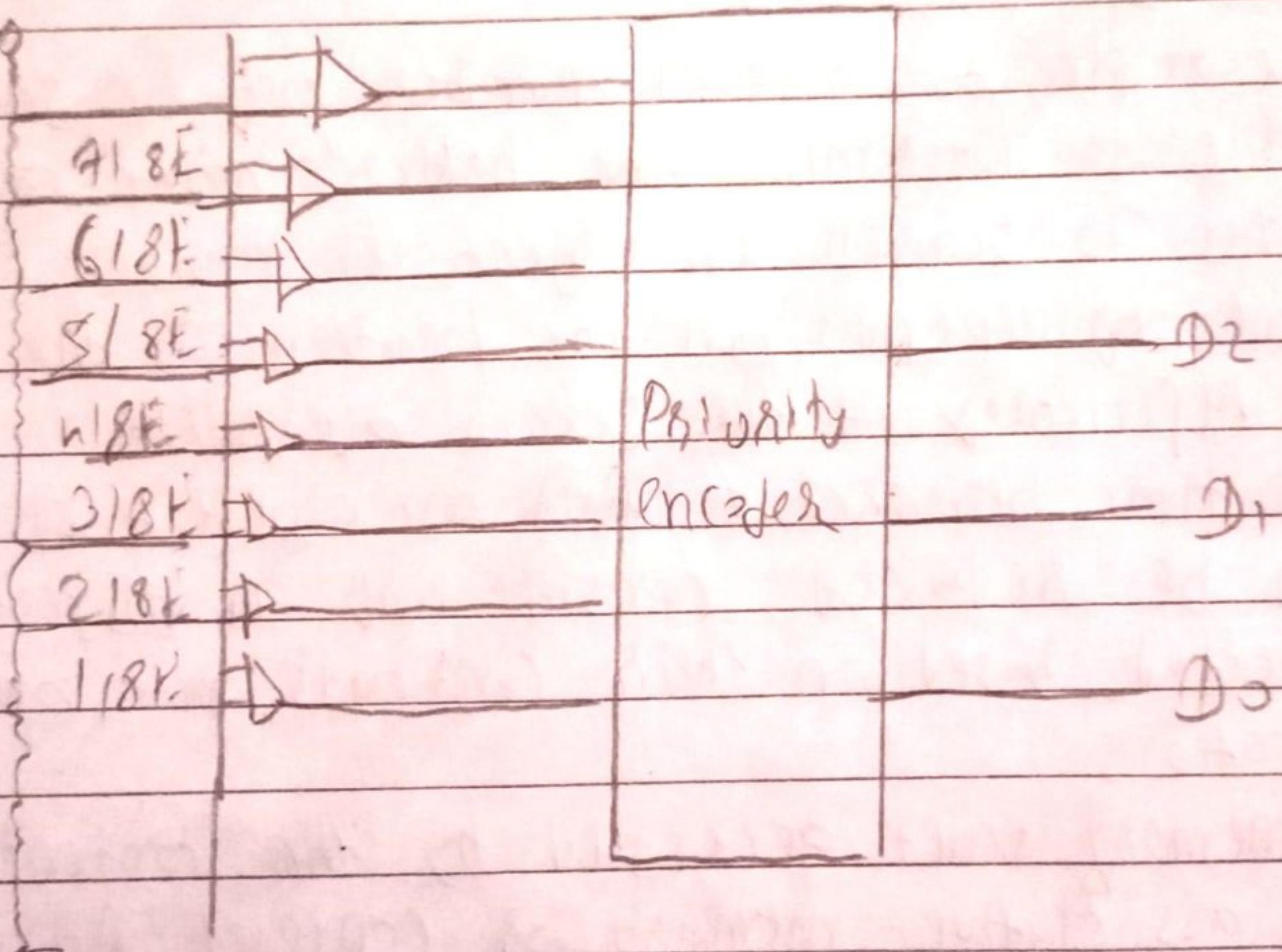
1 Uniform Quantization

2 Non-Uniform Quantization

Encoding - In digital electronics encoding refers to analog to digital conversion.

Transmission of digital data using a binary format is usually limited to short distances such as a computer to printer.

- Q5 Explain parallel comparator A/D converter
- 1 Flash ADC is also known as parallel comparator. This circuit is simplest to understand.
 - 2 It is formed of a series of comparators, each one comparing the input signal to a unique reference voltage. The comparator outputs connect to the inputs of a priority encoder circuit, which then produces a binary output.



1 each level is compared to the analog level by a voltage comparator.

2 This type of converter utilizes the parallel differential comparators reference voltage with the ~~at~~ analog input voltage

~~Metric~~ $\rightarrow \times \leftarrow$

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

Q2) Implement PLDs for the given logical problem.

Ans: Modules

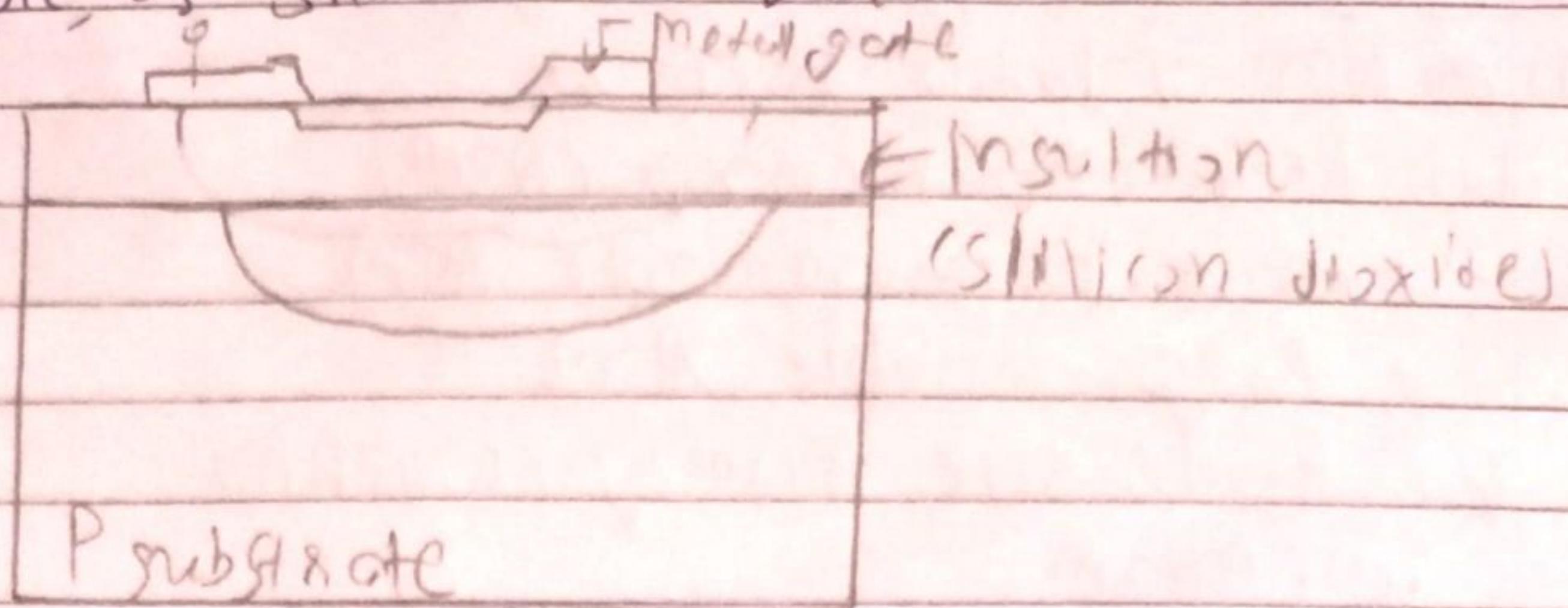
Q3) Explain content addressable memory (CAM).

- 1 Many data-processing applications require the search of items in a table stored in a memory.
- 2 They use object names or number to identify the location of the named or numbered object within a memory space.
- 3 For example, an account number may be searched in a file to determine the holder's name and account status. To search an object, the number of accesses to memory depends on the location of the object and the efficiency of the search algorithm.
- 4 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.
- 5 A memory unit accessed by the content is called an associative memory or content Addressable Memory (CAM).
- 6 This type of memory is accessed simultaneously and in parallel on the basis of data content rather than by specific address or location.

Q4) Explain charge-coupled device memory (CCD).

- Ans: 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 figure.



- 1 It consists simply of a p substrate, an insulating oxide layer, and isolate gates.
- 2 When a high voltage is applied to the metal gate, holes are repelled from a region beneath the gate in the p-type substrate.
- 3 The main advantage of CCD is that its simple cell structure. It makes it possible to construct large-capacity memories at low cost. Since CCDs are dynamic in nature, they must be periodically refreshed, and must be dynamic and driven by rather complex, multi-phase clock signals.
- 4 Another disadvantage of CCDs is due to serial data storage. Serial data storage has a longer access time in comparison to semiconductor RAM memory. It is on the order of 100ns.
- 5 CCDs can be used for digital or analog delay, and as serial data memories.
- 6 Another exciting application of CCDs is as the light sensitive image sensor in television cameras.

Q2

Explain classification of memory

Ans - 1 Classification of memory

(1) Non-volatile memory

(i) Read-only memory (ROM)

- Mask-programmable ROM

- Programmable ROM

(ii) Read/write memory (NVRAM)

- EEPROM

- Flash

- 3FLASH

(2) Volatile memory

- Read/write memory (RwM)

(i) Random Access

- SRAM

- DRAM

(ii) Non-Random Access

- FIFO

- LIFO

- Shift register

Q. Describe semiconductors.

Ans A semiconductor is a material product usually comprise of silicon which conducts electricity more than an insulator, such as glass but less than a pure conductor such as copper or aluminum.

• Their conductivity and other properties can be altered with the introduction of impurities, called doping, to meet the specific needs of the electronic components in which it resides.

- Semiconductor also known as semicon chips
Semiconductor can be found in thousands of products such as computers, smartphones, ~~appliances~~ appliances, gaming hardware and medical equipment.
- Semiconductors can be used in memory chips, microprocessors etc.

Q5 Explain ~~FET~~.ps Field programmable Gate Array (FPGA)

- The field programmable Gate Arrays (FPGAs) 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 function by the user instead of by the manufacturer of the device.

The word array is used to indicate a series of columns and rows of gates that can be programmed by the end user.

As compared to standard gate arrays, the field programmable gate Arrays are larger devices.

The basic cell structure for FPGAs is somewhat complicated than the basic cell structure of Standard Gate Array.

The programmable logic blocks of FPGAs are called logic blocks or configurable logic blocks (CLBs).

The basic architecture of FPGA consists of an array of logic blocks with programmable row and column interconnecting channels surrounded by programmable I/O blocks as shown in below figure.

210

210

210

210

210 block

Logic
block

Logic
block

Logic
block

Logic
block

210

210

Row
inter-
connect

210

Logic
block

Logic
block

Logic
block

Logic
block

210

Logic
block

Logic
block

Logic
block

Logic
block

210

210

210

210

210

column

inter-connect

—x—

Verilog