



Government Engineering College

Sec-28 Gandhinagar

Sem: - 3

Subject: - Digital Fundamental

Subject Code: - 3130704



Government Engineering College

Sec-28 Gandhinagar

Certificate

This is to certify that

Mr./Ms. PANCHAL JIGIAR SANJAYKUMAR Of class

.....CE..... DivisionB...., Enrollment No. 210130107066.....Has

Satisfactorily completed his/her term work in

.....DF..... Subject for the term ending in

....Jan....2022.

Date: -

Contents

1. Institute Vision/Mission	4
Vision	4
Mission.....	4
2. Computer Engineering Department Vision/Mission.....	5
Vision	5
Mission.....	5
3. Program Educational Outcome (PEO).....	6
4. PSO.....	6
5. POs	6
6. Submission Instructions:	Error! Bookmark not defined.
7. Assignment Index.....	8
8. Practical Index	9
9. Assignment 1	10
10. Assignment 2	11
11. Assignment 3	12
12. Assignment 4	13
13. Assignment 5	14
14. Practical 1.....	15
15. Practical 2.....	24
16. Practical 3.....	28
17. Practical 4.....	32
18. Practical 5.....	37
19. Practical 6.....	42
20. Practical 7.....	45
21. Practical 8.....	48
22. Practical 9.....	53
23. Practical 10.....	57
24. Practical 11.....	62

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:

- To achieve excellence for providing value based education in computer engineering through innovation, team work and ethical practice.

Mission:

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

Program Educational Outcome (PEO)

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

PSO

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

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

POs

Engineering Graduates will be able to:

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

mathematics, natural sciences, and engineering sciences.

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

Digital Fundamental (3130704)

Course Outcomes (COs)

CO-1	Solve the given problem using fundamentals of Number systems and Boolean Algebra
CO-2	Analyze working of logic families and logic gates and design simple circuits using various gates
CO-3	Design and implement combinational and sequential logic circuits & verify its working
CO-4	Examine the process of Analog to digital conversion & Digital to Analog converter
CO-5	Implement PLDs for given logical problem

7. Assignment Index

Sr. No	Assignment	Date	Page No.	Sign
1	Assignment 1	30/9/22		
2	Assignment 2	17/10/22		
3	Assignment 3	30/11/22		
4	Assignment 4	14/12/22		
5	Assignment 5	30/12/22		

8. Practical Index

Sr. No	Assignment	Date	Page No.	Sign
1	Practical 1	13/9/22	15 - 24	
2	Practical 2	27/9/22	25 - 31	
3	Practical 3	4/10/22	32 - 36	
4	Practical 4	11/10/22	37 - 42	
5	Practical 5	18/10/22	43 - 48	
6	Practical 6	15/11/22	49 - 51	
7	Practical 7	22/11/22	52 - 55	
8	Practical 8	29/11/22	56 - 61	
9	Practical 9	6/12/22	62 - 65	
10	Practical 10	13/12/22	66 - 71	
11	Practical 11	20/12/22	72 - 73	

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 - 1

D) State and explain De-Morgan's theorems with truth tables.

$$\Rightarrow \text{Law 1 : } A+B = \bar{A} \cdot \bar{B}$$

This law states that the complement of a sum of variables is equal to the product of their individual complements.

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

Hence, Law 1 is proved from the above truth table.

$$\text{w) Law 2: } \bar{AB} = \bar{A} + \bar{B}$$

This law states that the complement of a product of variables is equal to the sum of the individual complements.

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

Hence, Law 2 is proved from the above truth table.

② Simplify Boolean function: $F = A'B'C + A'BC + AB'C$

$$\begin{aligned} F &= A'C(B + B') + AB' \\ &= A'C + AB' \end{aligned}$$

③ List & explain Logic family.



Characteristic	ECL	TTL	CMOS
Power input	Moderate-high	Moderate	Low
Frequency limit	Very high	High	Moderate
Circuit density	Moderate	Moderate-high	High-Very high
Circuit type per family	Moderate	High	High

④ Describe error detecting & correcting code

⇒ Error - detecting codes:

→ Noise can alter or distort the data in transmission

→ The 15 may get changed to 05 or 05 to 15

→ Because digital systems must be accurate to the digit, errors can pose a serious problem

→ Single bit error should be detect & correct by different schemes

→ Parity, check sums and block parity are few examples of error detecting codes.

⇒ Parity:-

→ Parity bit is the simplest technique

→ Two types of parity - odd parity & even parity

→ Odd parity, the parity is set at 0 or 1 at the transmitter such that the total number

of 1 bits in the word including the parity bit is an even odd number.

→ Even parity, the parity set is of 00 or 01 at the transmitter such that the total number of 1 bits in the word including the parity is an even number.

→ Ex.: 0110 binary number has "1" as odd parity and "0" as even parity.

→ Detect a single bit-error but cannot detect two or more errors within the same word.

→ In any practical system, there is always a finite probability of the occurrence of single error.

→ Eg: In an even-parity scheme, code 110111001 is erroneous because number of 1 is odd, while code 11110110 is error free because number of 1 is even.

⇒ Check sums:-

→ Simple parity can not detect two errors within the words.

→ Added to the sum of the previously transmitted words.

→ At the transmission, the checksum upto that time is sent to the receiver.

→ The receiver can check its sum with the transmitted sum.

→ If the two sums are the same, then no errors were detected at the receiver end.

→ If there is an error, the receiving location can ask for retransmission of the entire data.

Block parity

0 1 0 1 1 0 1 1 0	0 1 0 1 1 0 1 1 0	0 1 0 1 1 0 1 1 0
1 0 0 1 0 1 0 1 1	1 0 0 1 0 1 0 1 1	1 0 0 1 0 1 0 1 1
0 1 1 0 1 1 1 0 0	0 1 1 0 0 1 1 0 0	0 1 1 0 1 1 1 0 0
1 1 0 1 0 0 1 1 0	1 1 0 1 0 0 1 1 0	1 0 0 0 0 1 1 0 0
1 0 0 0 1 1 0 1 1	1 0 0 0 1 1 0 1 1	1 0 0 0 1 1 0 1 1
Parity row → 0 1 1 1 0 1 1 1	0 1 1 1 0 1 1 1	0 1 1 1 0 1 1 1
0 1 1 1 0 1 1 0 0	0 1 1 1 0 1 1 0 0	0 1 1 1 0 1 1 0 0

↑ Parity column

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 : 02

Q) Explain K-map

- The Karnaugh map method is a systematic method of simplifying the boolean expression
- The K-map is a chart or a graph, composed of an arrangement of adjacent cells, each representing a particular combination of variables in sum or product form.
- The output values placed in each cell are derived from the minterms of a boolean expression
- A minterm is a product term that contains all of the function's variables exactly once, either complemented or not complemented

w) Two-variable K-map

- The two variable expression can have $2^2 = 4$ possible combinations of the input variables A and B
- Each of these combinations, $A'B'$, $A'B$, AB' and AB are called minterms

A	B	minterm	
0	0	$m_0 = A'B'$	
0	1	$m_1 = A'B$	
1	0	$m_2 = AB'$	
1	1	$m_3 = AB$	

w) Three variable K-map:

- A function in three variable expressed in the standard SOP form can have eight possible combination: $A'B'C'$, $A'B'C$, $A'B'C'$, $A'BC$, $AB'C'$, $AB'C$, ABC' , ABC

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

Minterm number

- ② Obtain the simplified expressions in SOP for the following Boolean functions:

(i) $F(x, y, z) = \Sigma(2, 3, 6, 7)$

z	x\y	00	01	11	10
1	1	3	7	5	
2	1	4	8	6	

Final Boolean expression: $y_2' + y_2 = y_2 \oplus y_2$

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

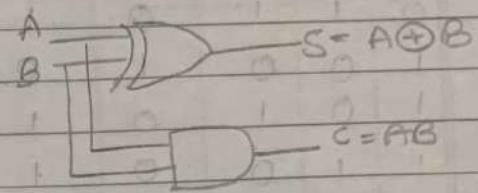
CD\AB				
1	1			
1	1	1	1	

Final Boolean expression: $B'C'D' + A'B'C'D + A'B'C'D'$

③ Describe adder & subtractor
 ➡ (i) Half adder.

→ A combinational ckt. which adds two one-bit binary numbers is called half-adder.

input		Outputs	
A	B	Sum	Carry
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1



- The sum column resembles like an output of the XOR gate
 → The carry column resembles like an output of the AND gate

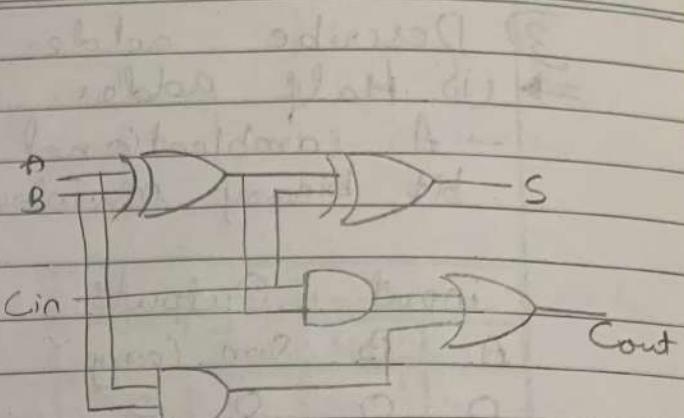
(ii) Full adder

→ The full-adder adds the bits A and B and the carry from the previous column called carry-in Cin and outputs the sum bits and the carry-out Cout

$$\begin{aligned}
 S &= A'B'C_{in} + A'BC_{in} + AB'C_{in} + ABC_{in} \\
 &= (AB' + A'B)C_{in} + (AB + A'B')C_{in} \\
 &= (A \oplus B)C_{in} + (A \oplus B)'C_{in} \\
 &= A \oplus B \oplus C_{in}
 \end{aligned}$$

$$\begin{aligned}
 Cout &= A'B'C_{in} + AB'C_{in} + ABC_{in} + ABC_{in} \\
 &= AB + (A \oplus B)C_{in}
 \end{aligned}$$

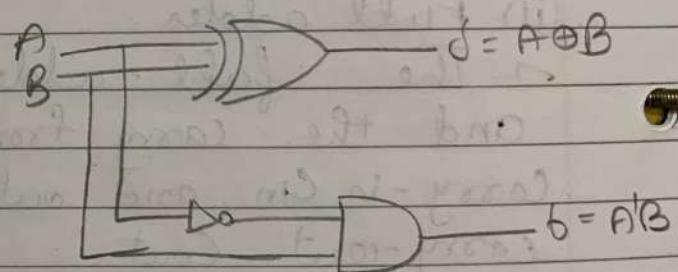
input			Output	
A	B	Cin	S	Cout
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1



(iii) Half subtractor

- Subtracts one bit from the other and produces the difference
- Other output is to specify if 2 is borrowed

inputs		Outputs	
A	B	d	b
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0

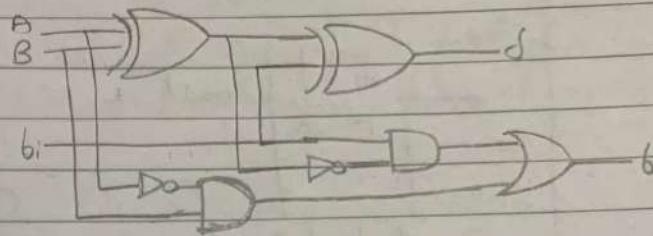


(iv) Full subtractor

- Full subtractor is a combinational ckt with 3 inputs (A, B, b_i)
- Subtraction = $A - B - b_i$

$$\begin{aligned}
 f &= A'B'b_i + A'Bb_i' + AB'b_i + ABb_i \\
 &= (A \oplus B)b_i + (A \oplus B)'b_i \\
 &= A \oplus B \oplus b_i
 \end{aligned}$$

inputs	outputs			
A	B	b _i	S	b
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	0	1
1	0	0	1	0
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1



- 4) Explain multiplexer & demultiplexer.
 ⇒ (i) Multiplexer

A multiplexer (MUX) is a device that allows digital information from several sources to be routed onto a single line for transmission over that line to a common destination.

→ Consider an integer 'm', which is constrained by the following relation:

$$m = 2^n, \text{ where } m \text{ and } n \text{ are both integers}$$

→ A $m \times 1$ multiplexer has

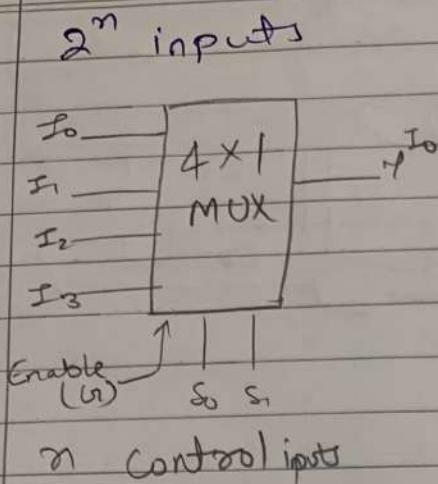
m inputs: $I_0, I_1, I_2, \dots, I_{m-1}$

one output : y

n control inputs = $S_0, S_1, S_2, \dots, S_{n-1}$

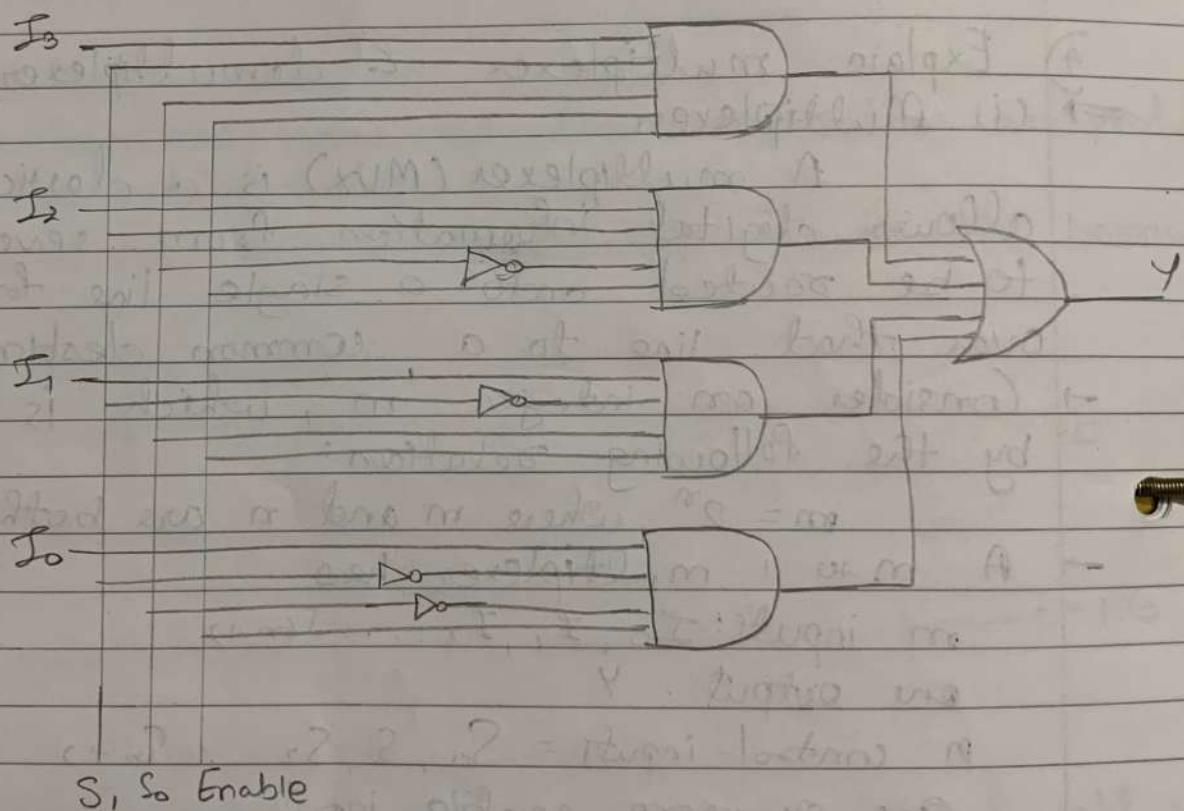
One or more enable input

Such that y may be equal to one of the inputs depending upon the control inputs.



Select	inputs	output
S ₁ 0	I ₀ 0	I ₀
0 1	0 0	I ₀
0 1	1 1	I ₁
1 1	0 1	I ₂
0 1	1 0	I ₃

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



(ii) Demultiplexer:

→ A demultiplexer (DEMUX) is a device that allows digital information from one source to be routed onto a multiple lines for transmission over different destinations.

→ Consider an integer 'm', which is constrained by the following relation:

$$m = 2^n, \text{ where } m \text{ & } n \text{ are integers}$$

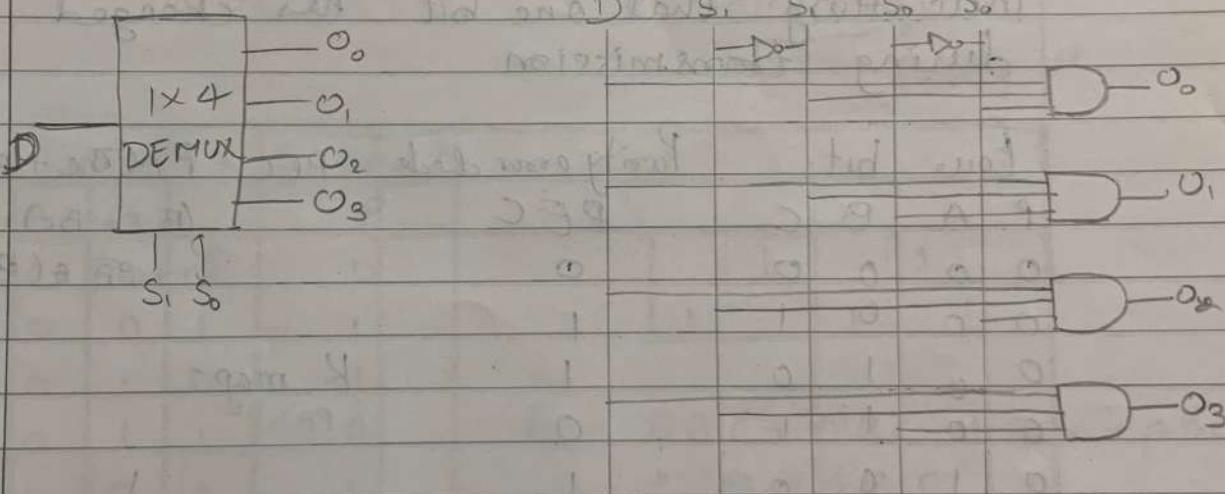
→ A 1-to-m Demultiplexer has one input: D

m Outputs: $O_0, O_1, O_2, \dots, O_{m-1}$

n control inputs: $S_0, S_1, S_2, \dots, S_{n-1}$

one (or more) enable input(s)

→ Such that D may be transfer to one of the outputs, depending upon the control inputs



Select code Output

$S_1 \ S_0 \quad O_3 \ O_2 \ O_1 \ O_0$

0 0 0 0 0 D

0 1 0 0 D 0

1 0 0 D 0 0

1 1 D 0 0 0

⑤ Describe parity checker & generator.

\Rightarrow (i) Parity checker

- (i) 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.
 - The parity checker at. checks for possible error in the transmission

→ Since the info. was transmitted with even parity, the four bits received must have an even no. of 1s. If an error occurs during the transmission, the four bits received have an odd no. of 1s, indicating that one bit has changed in value during transmission.

Four bit				Parity error check	$PEC = \bar{P} \bar{A}(\bar{B}C + \bar{B}\bar{C} + \bar{B}\bar{A}) \oplus (\bar{B}\bar{C} + B\bar{C}) + P\bar{A}(\bar{B}C + BC) + P\bar{A} \oplus (\bar{B}\bar{C} + BC)$
P	A	B	C	PEC	
0	0	0	0	0	
0	0	0	1	1	
0	0	1	0	1	
0	0	1	1	0	K map:
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	

→ Parity generator:

→ Binary data, when transmitted and processed is susceptible to noise that can alter its 1s to 0s and 0s to 1s.

→ To detect such errors, an additional bit called parity bit is added to the data bits and the word containing the data bits and parity bit is transmitted.

→ At the receiving end the no. of 1s in the word received are counted and the errors detected.

even parity
 $\begin{array}{|c|c|c|c|} \hline 0 & 0 & 1 & 1 \\ \hline \end{array}$

odd parity
 $\begin{array}{|c|c|c|c|} \hline 0 & 0 & 1 & 0 \\ \hline \end{array}$

Data Parity bit

input			Output
A	B	C	f
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

$A \oplus B$

c	1	1	1
1	1	1	1

$$\begin{aligned}
 f &= A'B'C + A'BC' + ABC + AB'C' \\
 &= A \oplus B \oplus C
 \end{aligned}$$



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 03

D) Differentiate sequential & combinational Circuits.

Combinational Circuit

- In combinational ckt, the output variables at any instant of time are dependent only on the present input variables.

- Memory unit is not required.

- Combinational ckt are faster because the delay between the input and the output is due to propagation delay of gates only.

- Combination ckt are easy to design

Sequential Circuits

- In sequential ckt, the output variables at any instant of time are dependent not only on the present state.

- Memory unit is required to store the past history of the input variables in this ckt.

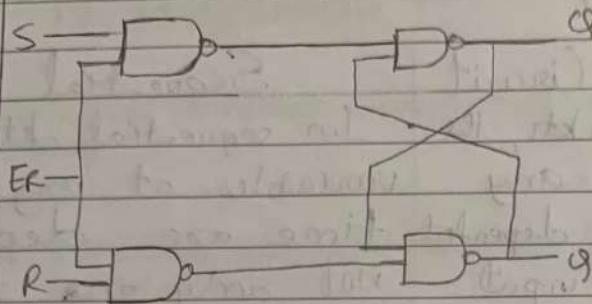
- Sequential ckt are slower than combinational ckt.

- They are comparatively harder to design

② List & explain flip-flop

- A flip-flop, also known formally as bistable multivibrator, has two stable states
- Following are flip-flop

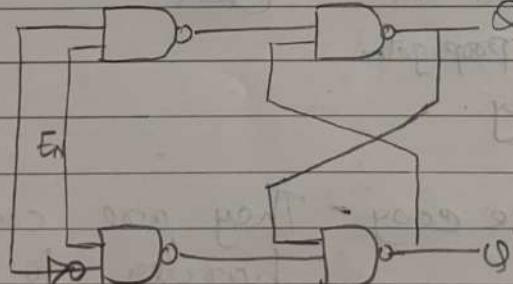
(i) S-R flip flop



En	S	R	Q_n	Q_{n+1}	State
1	0	0	0	0	No change
1	0	0	1	1	Reset
1	0	1	0	0	Set
1	0	1	1	0	Set
1	1	0	0	1	Set
1	1	1	0	X	invalid
0	X	X	0	0	No change

- A gated S-R latch requires an EN input
- This type of flip flop responds to the changes in inputs only as long as the clock is HIGH, these type of flip-flop are also called triggered.

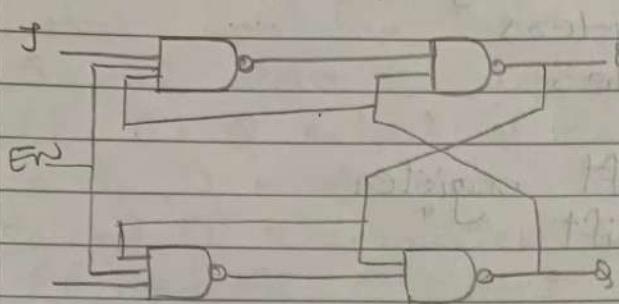
(ii) D flip-flop



En	D	Q_n	Q_{n+1}	State
1	0	0	0	Reset
1	0	1	0	
1	1	0	1	Set
1	1	1	1	
0	X	0	0	No change
0	X	1	1	No change

- It differs from the SR latch in that it has only one input in addition to EN
- We can say that the output Q follows the D input when EN is HIGH

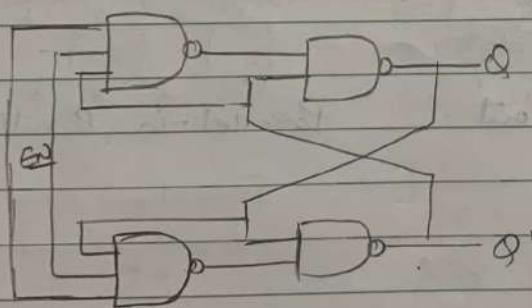
(iii) J-K flip-flop



En	J	K	Q _n	Q _{n+1}	State
1	0	0	0	0	No change
1	0	0	1	1	Change
1	0	1	0	0	Reset
1	0	1	1	0	Set
1	1	0	0	1	Set
1	1	0	1	1	Toggle
0	X	X	0	0	No change
0	X	X	1	1	Change

- The J-K flip-flop is very versatile and also the most widely used.
- The functioning is identical to that of the S-R flip flop except that it has no invalid state like that of S-R flip-flop.

(iv) T flip-flop



En	T	Q _n	Q _{n+1}	State
1	0	0	0	No change
1	0	1	1	Change
1	1	0	1	Toggle
1	1	1	0	Change
0	X	0	0	No change
0	X	1	1	Change

- A T flip-flop has a single control input, labeled T for toggle.
- Although T flip-flop are not widely available commercially it is easy to convert a J-K flip flop to the functioned equivalent of a T flip flop by just connecting J and K together and labeling the common connection as T.

③ List & Explain registers

→ Types of registers

① Buffer register

② Shift register

③ Bidirectional Shift register

④ Universal shift

↪ Shift register:

A number of FFS connected together such that data may be shifted into and shifted out of them is called a shift register.

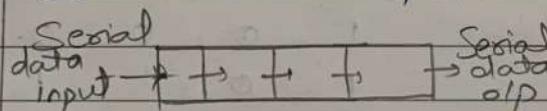
→ There are four basic type of shift registers

① Serial-in, Serial-out

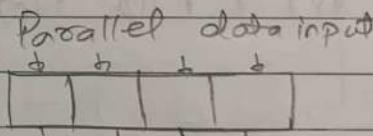
② Serial-in, Parallel-out

③ Parallel-in, Serial-out

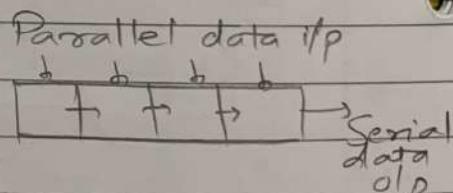
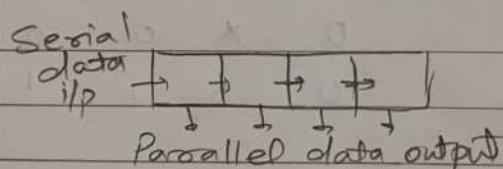
④ Parallel-in, Parallel-out



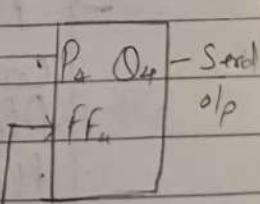
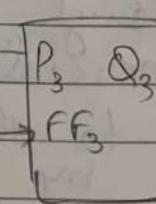
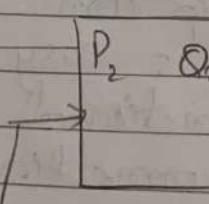
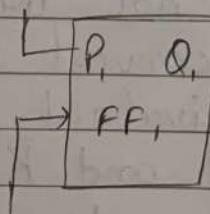
Serial-in, Serial-out



Parallel-in, Parallel-out



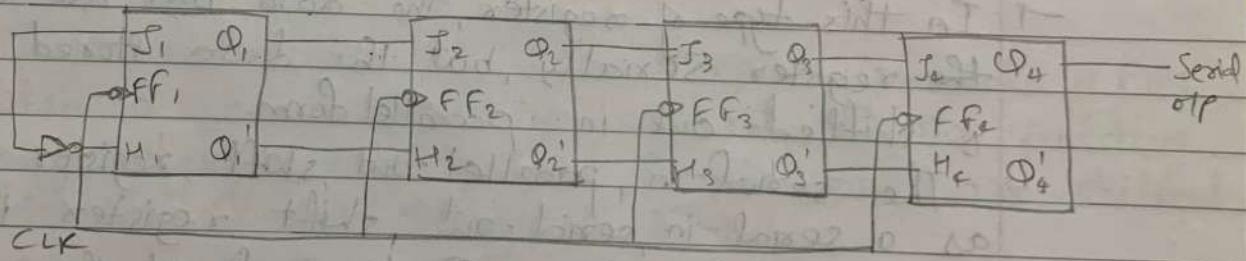
• Serial-in, serial-out



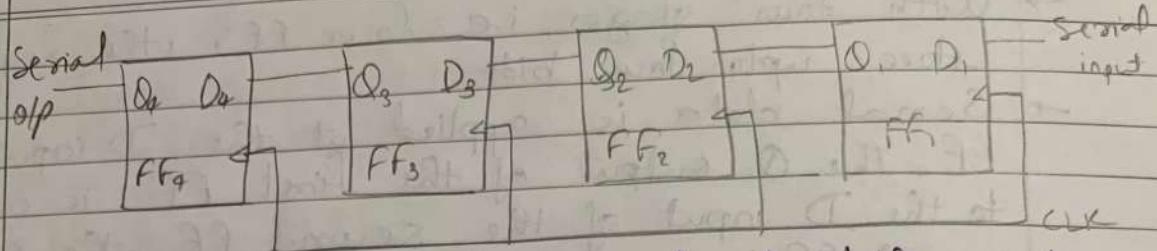
CLK

4-bit serial in, serial out, Shift right

- With four stages, i.e. four FFs, the register can store upto four bits.
- Serial data is applied at the D input of the first FF. The Q output of the first FF is connected to the D input of the second FF, the output of the second FF is connected to the D input of the third FF and the Q output of the third FF is connected to the D input of fourth FF.
- When serial data is transferred into a register, each new bit is clocked into the first FF at the positive edge of each clock pulse.
- The bit that was previously stored by the first FF is transferred to the second FF. The bit that was stored by the second FF is transferred to the third FF and so on.

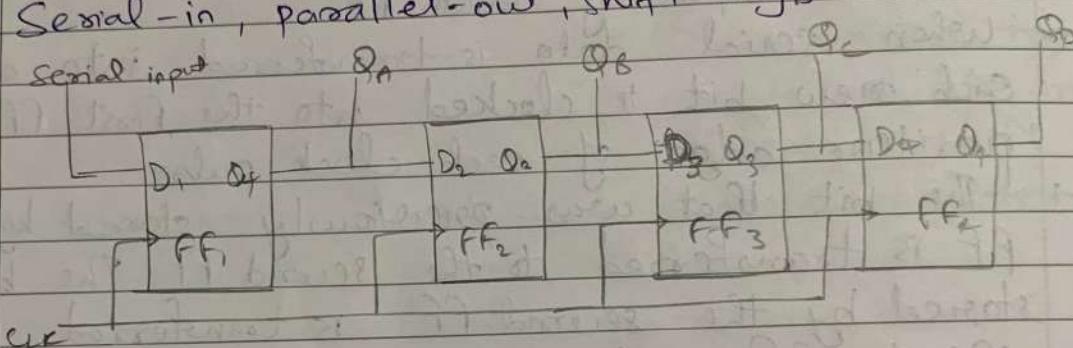


- A shift register can also be constructed using T-K FFs as shown in above fig.
- The data is applied at the T input of the first FF, the complement of this is fed to the K input of FF.
- The Q output of the first FF is connected to T input of the second FF, the Q output of the second FF to T input of the third FF and so on.
- Also, Q'_1 is connected to K_2 , Q'_2 is connected to K_3 , and so on.



4-bit serial in, serial out, shift register

→ Serial-in, parallel-out, shift register



→ In this type of registers, the data bits are entered in the register serially, but the data stored in the register is shifted out in parallel form

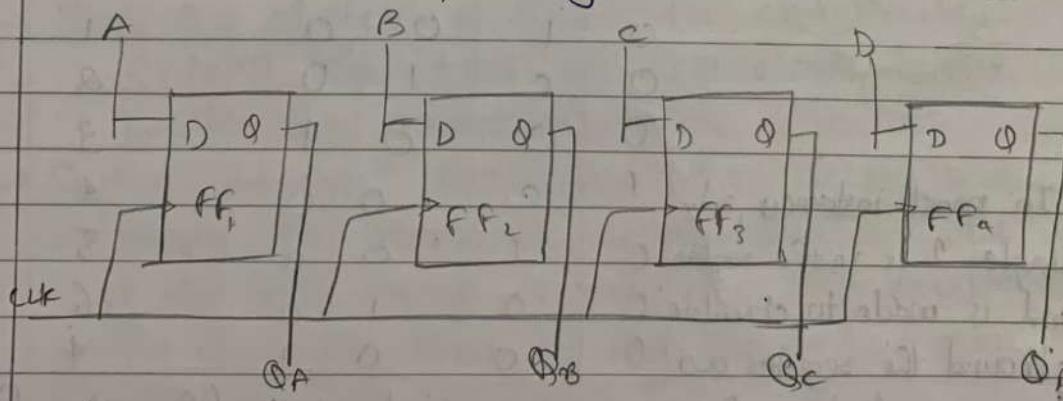
→ The serial-in, parallel-out, shift register can be used as a serial-in, serial-out, shift register if the o/p is taken from Q terminal of the last ff

→ Parallel-in, serial-out, shift register

→ There are four data lines A, B, C and D through which the data is entered into the register in parallel form

→ The signal shift LOAD allows (a) the data to be entered in parallel form to the register and (b) the data to be shifted out serially from terminal Q4

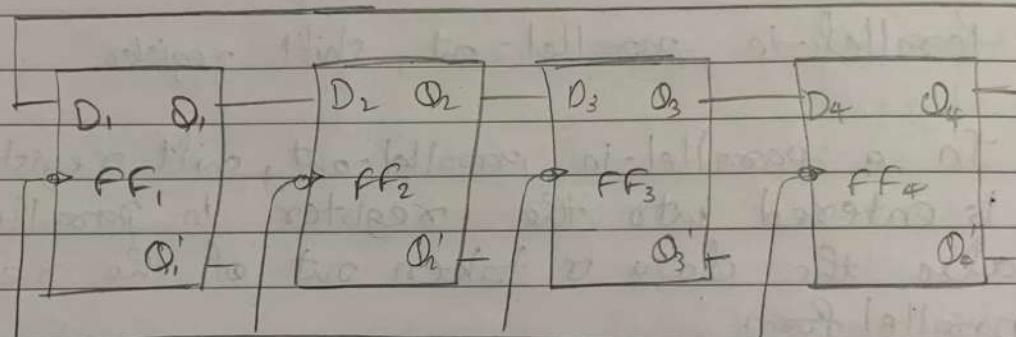
- When shift LOAD line is HIGH, gates G₁, G₂ & G₃ are disabled but gates G₄, G₅ and G₆ are enabled allowing the data bits to shift right from one stage to next.
- When a clock pulse is applied, these data bit are shifted to the Q output terminals of the FFs and therefore, data is inputted in one step.
- The OR gate allows with the normal shifting operation or the parallel data entry depending on which NAND gates are enabled by the level on the Shift LOAD SH/P
- Parallel-in, parallel-out, shift register
- In a parallel-in, parallel-out, shift register the data is entered into the register in parallel form and also the data is taken out of the register in parallel form.
- Data is applied to the D input terminals of FFs
- When a clock pulse is applied, at the positive going edge of that pulse, the D input are shifted into the Q outputs of the FFs



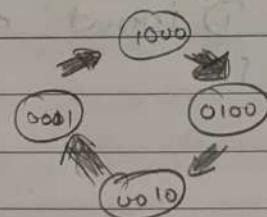
- The register now stores the data, the stored data is available instantaneously for shifting out in parallel form

4) Describe ring counter

- This is the simplest shift register counter. The basic ring counter using D FFs is shown in fig.
- The FFs are arranged as in a normal shift register, i.e. Q output of each stage is connected to the D input of the next stage, but the Q output of the last FF is connected back to the D input of the first FF such that the array of FFs is arranged in a ring, and therefore the name ring counter.



→ State diagram

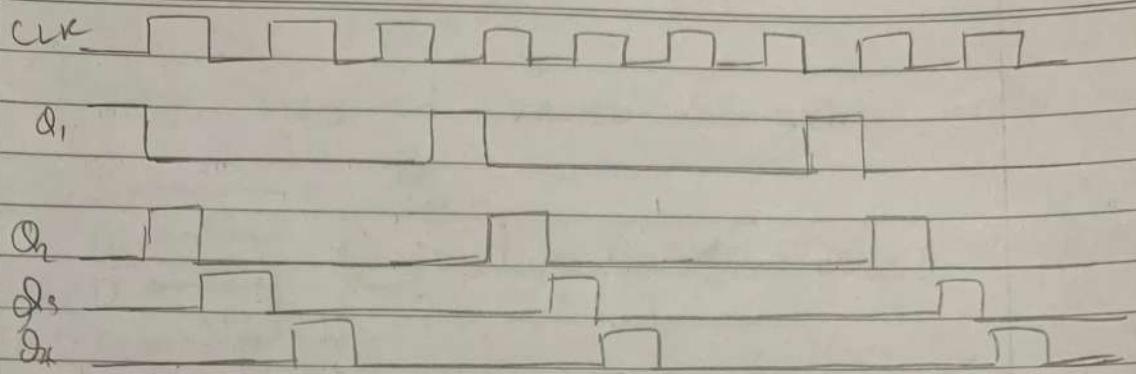


• Sequence table

	Q_1	Q_2	Q_3	Q_4	After clock pulse
	1	0	0	0	0
	0	1	0	0	1
	0	0	1	0	2
	0	0	0	1	3
	1	0	0	0	4
	0	1	0	0	5
	0	0	1	0	6
	0	0	0	1	7

→ In most instances, only single 1 is in the register and is made to circulate around the register as

long as clock pulses are applied initially, the first FF is present to a 1



- So, the initial state is 1000. After each clock pulse the contents of the register are shifted to 0.
- The sequence repeats after four clock pulse

⑤ Describe how to design counters using flip flops

→ Step 1: Number of flip flops
→ Based on the description of the problem, determine the required number n of the FFs. The smallest value of n is such that the number of states $N \leq 2^n$ and desired counting sequence

→ Step 2: State diagram

→ Draw the state diagram showing all the possible states

→ Step 3: choice of flip-flops and excitation table:

- Select the type of flip-flops to be used and write the excitation table.

→ Step 4: Minimal expressions for excitations:

→ Obtain the minimal expressions for the excitations of the FFs using K-maps for the excitations of the flip flops in terms of the present states & ilps

→ Step 5: Logic Diagram

→ Draw the logic diagram based on the minimal expressions.

12. Assignment 4

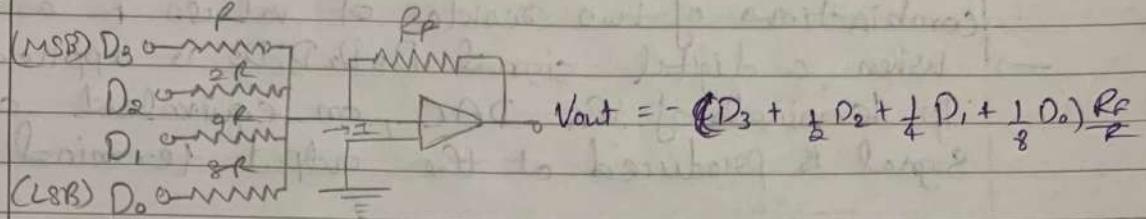
CO4: Examine the process of Analog to Digital conversion and Digital to Analog conversion.

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 - 04

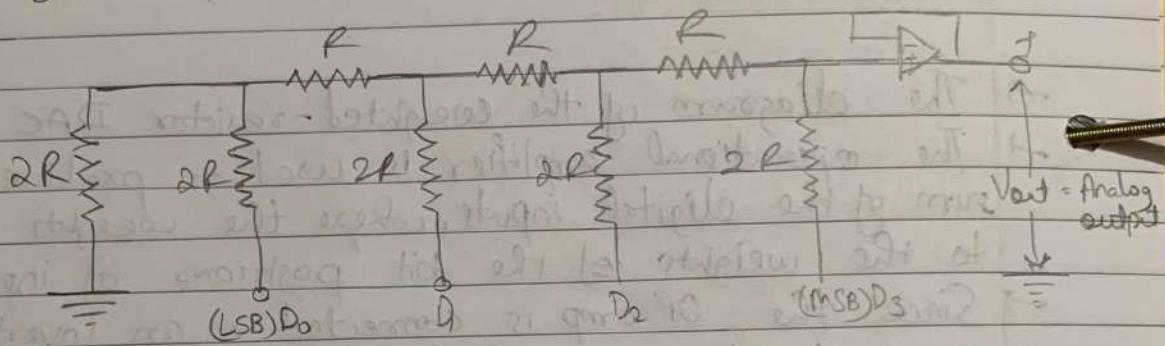
D) Explain weighted resistor converter.



- The diagram of the weighted-resistor DAC is shown in fig
- The operational amplifier is used to produce a weighted sum of the digital inputs, where the weights are proportional to the weights of the bit positions of inputs
- Since the OP-amp is connected as an inverting amplifier each input is amplified by a factor equal to the ratio each input is amplified by a factor equal to the ratio to which it is connected.
- The MSB D_3 is amplified by R_f/R , D_2 is amplified by $R_f/2R$, D_1 is amplified by $R_f/4R$ & D_0 the ~~the~~-LSB is amplified by $R_f/8R$
- The inverting terminal of the OP-amp in figure acts as a virtual ground.
- Since the ~~OP~~ OP-amp adds and inverts

$$V_{out} = -\left(D_3 + D_2/2 + D_1/4 + D_0/8\right) \times \left(\frac{R_f}{R}\right)$$
- The main disadvantage of this type of DAC is that a different, valued precision resistor must be used for each bit position of the digital input

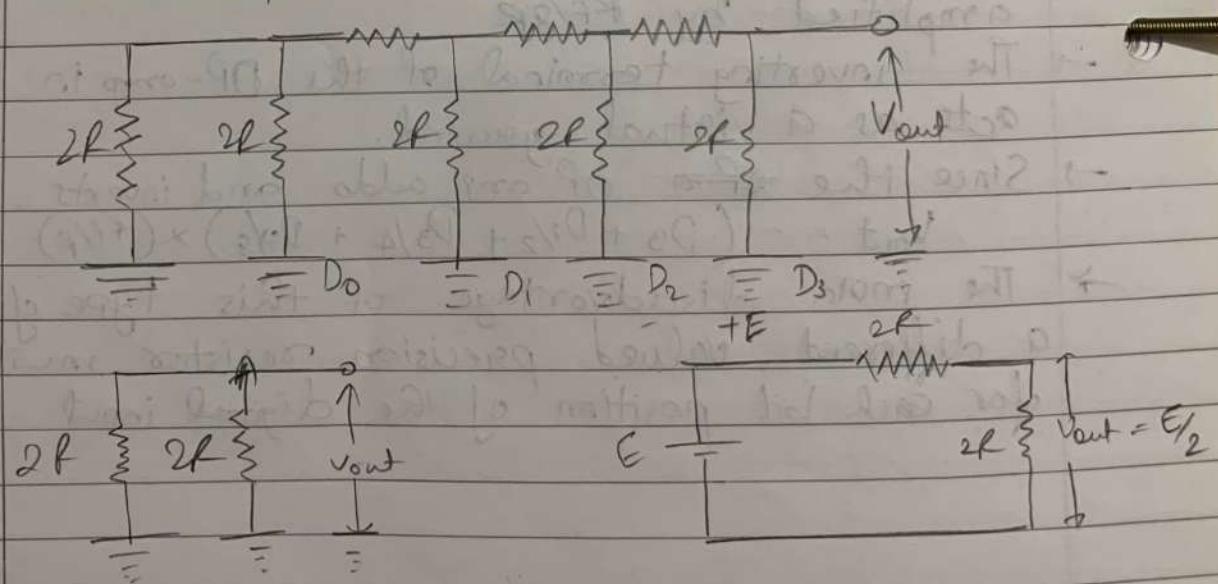
- ② Explain R-2R ladder D/A converter
- It uses a ladder network containing series-parallel combinations of two resistors of values R and $2R$
- When a digital signal $D_3 D_2 D_1 D_0$ is applied at the input terminals of the DAC, an equivalent analog signal is produced at the output terminal



→ Case 1: When input is 1000

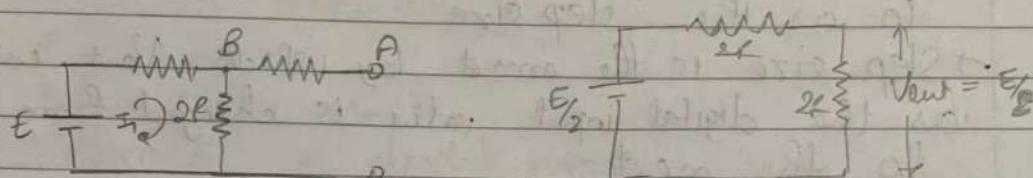
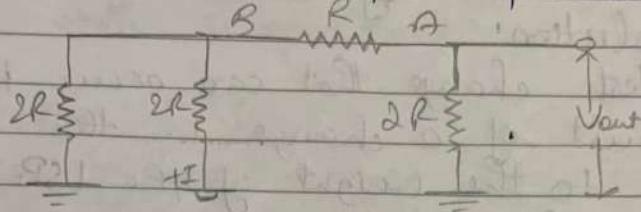
→ Below figure illustrates the procedure to calculate V_{out} when the input is 1000

→ The output voltage, $\Rightarrow V_{out} = E/2$



→ Case 2: When the input is 0100

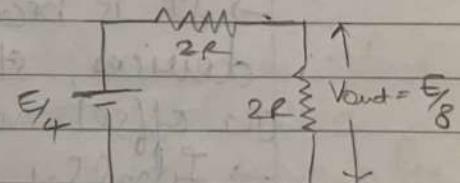
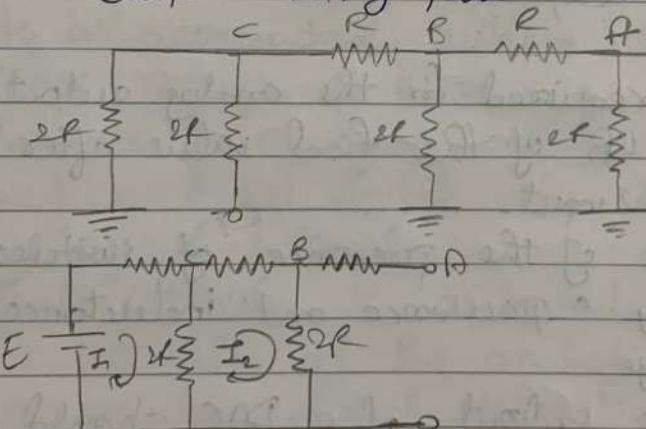
→ Below fig. illustrates, the procedure to calculate V_{out}



→ Case 3: When the input is 0010

→ Below fig. illustrates the procedure to calculate V_{out}

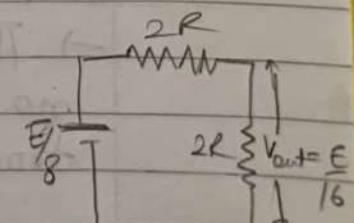
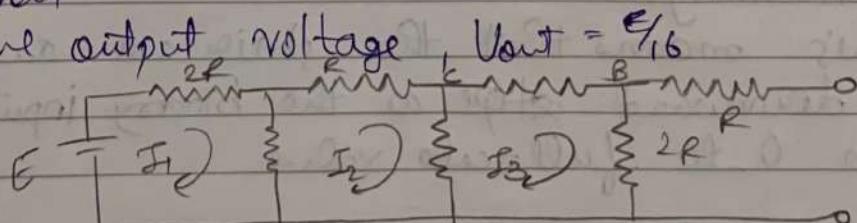
→ Here, we find that to left of terminal C, $R_{eq} = 2R$,
The output voltage, $V_{out} = E/8$



→ Case 4: When the input is 0001

→ Below figure illustrate the procedure to calculate V_{out}

→ The output voltage, $V_{out} = E/16$



③ Describe specification of A/D & D/A converter
 => Specification of DAC

(i) Resolution:

- Smallest change that can occur in an analog o/p as a result of a change in the digital i/p
- Equals to the weight of the LSB & also referred to as the step size
- Step size is the amount by which V_{out} will change as the digital input value is changed from one value to the next

$$\% \text{ resolution} = \text{step size} / \text{full scale} \times 100\%$$

(ii) Accuracy:

- Specified in terms of full-scale error and linearity error
- Full-scale error is the maximum deviation of DAC's o/p from its expected value

(iii) Setting time:

- The time required for the analog output to settle to within $\frac{1}{2}$ LSB of the final value after a change in the digital input
- It is because of the presence of switches, active devices, stray capacitance and inductance.

(iv) offset voltage

- Ideally the output of a D/A should be zero when the binary i/p is zero
- However, in practice, there is a very small o/p voltage under this situation called the offset voltage

(v) Monotonicity:

- This means that the staircase output will have no downward steps as the binary input is incremented from 0 to full-scale value

(vi) Temperature sensitivity:

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

→ Specification of ADC

→ Range of input voltage

→ Input impedance

→ Accuracy

→ Conversion time

→ Format of digital output

④

Explain quantization and encoding.

→ In a digital-to-analog converter, the possible no. of digital inputs is fixed.

→ Therefore, the whole range of analog voltage is required to be represented suitably in 2^n intervals.

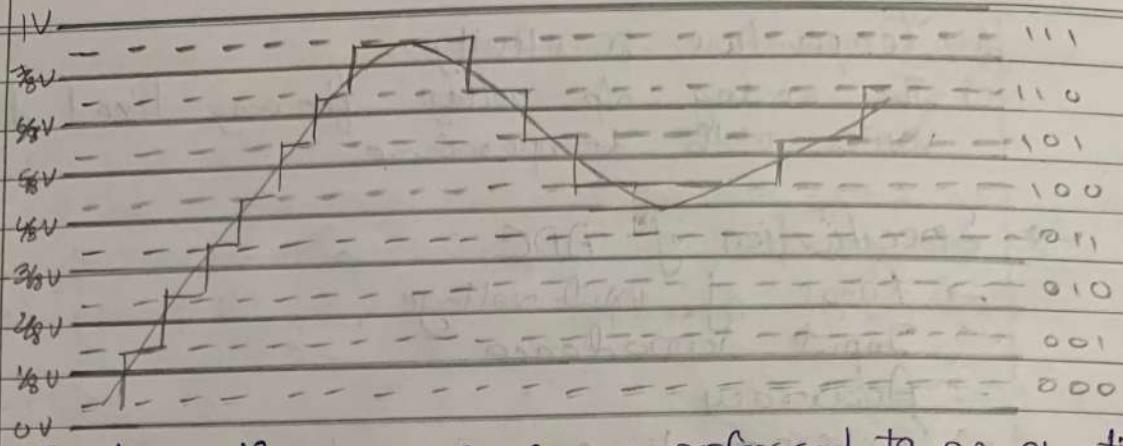
→ This process is known as quantization.

→ Consider an analog voltage in the range of 0 to V and a 8-bit digital the whole range of analog voltage & intervals

→ Each intervals is assigned a 3-bit binary value

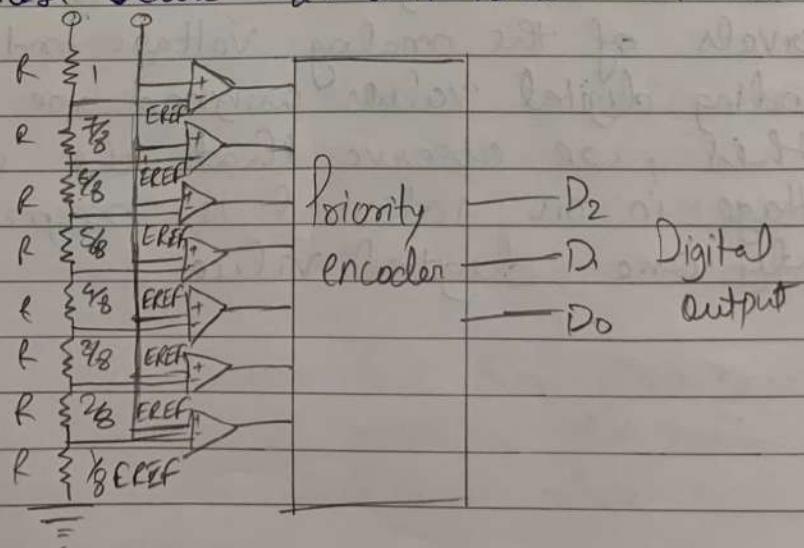
→ The intervals of the analog voltage and their corresponding digital values assigned are shown in fig.

→ From this, we observe that the whole range of voltage in an interval is represented by only one digital value



→ Therefore, there is an error referred to as quantization error, involved in this process of quantization

- ⑤ Explain parallel comparator A/D converter
- The flash type A/D converter is the fastest type of A/D converter
- This type of converter utilizes parallel differential comparators that compare reference voltages with the analog input voltage
- A reference voltage E_{ref} is connected to a voltage divider that divides it into 7 equal increment levels
- For any given analog input, one comparator and all those below it will have a HIGH output



- All comparator outputs are connected to a priority encoder, which produces a digital output corresponding to the input having produces in this case is the one that represents the largest input
- Thus the digital output represents the voltage that is closest in value to the analog input

$$\left(\frac{7R}{7R+R}\right) \times E_{REF} = \frac{7}{8} \times E_{REF}$$

- Similarly, the voltage applied to the inverting terminal of the second comparator is

$$\left(\frac{6R}{6R+R}\right) \times E_{REF} = \frac{6}{8} \times E_{REF}$$

- The flash converter uses no clock signal, because there is no timing or sequencing period.

13. Assignment 5

CO5: Implement PLDs for the given logical problem.

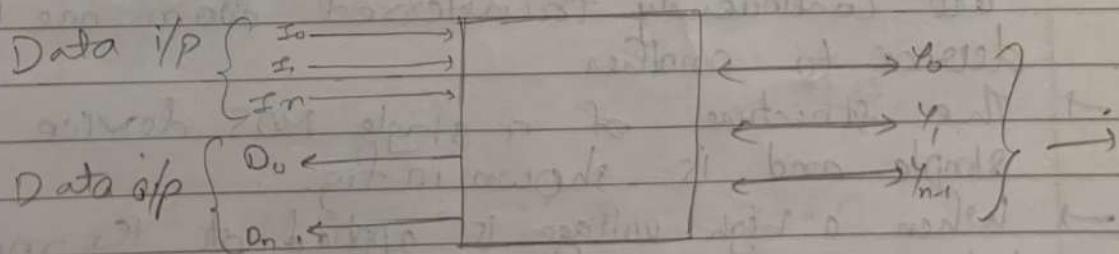
Module 5

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

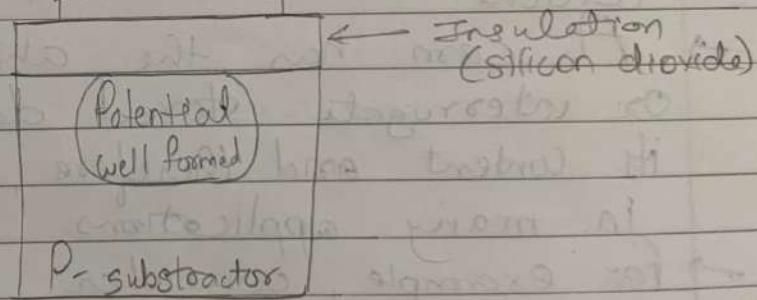
D) Explain content addressable memory (CAM)

- The content addressable memory is a special purpose random access memory device that can be accessed by searching for data content.
- For this purpose, it is addressed by associating the input data, returned to as key, simultaneously
- The operation is returned to as association or interrogation and this type of memory is also known as associative memory.
- The key to be used may either consists of the entire data word or only some specific bits of the data word.
- A CAM differs from the conventional memory organization in that the addressing of a location in the latter has no relation to the memory content.
- A cam has the ability to search out or interrogate stored data on the basis of its content and therefore, can be powerful asset in many applications.
- For example, consider a list containing the names of persons, their ages professions and nationalities, stored on CAM



- On the other hand if it is required to find the engineers of Indian nationality the key will consist of the combination of the codes corresponding to engineer and Indian nationality.
- All the memory locations with engineers of Indian nationality will be identified and remaining data can then be retrieved by using thread operation.
- To do the same search process with a conventional memory each memory word is to be read out and compared with the key.
- This search is a serial process and hence time consuming.

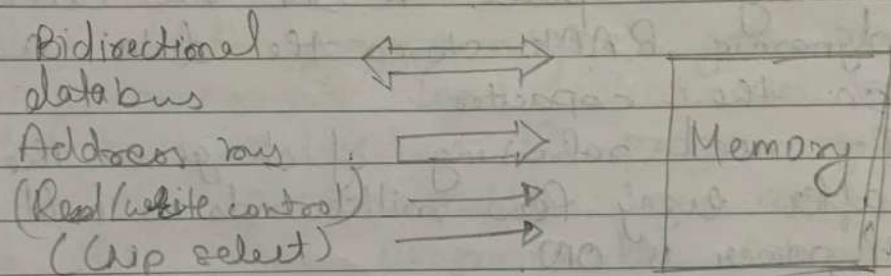
Q) Explain charge de-coupled device memory (CD)



- The charge coupled device memory is a type of dynamic memory in which packets of charges are continuously transferred from one MOS device to another.
- The structure of a single MOS device is quite simple and is shown in fig.
- When a high voltage is applied at the metal gate holes are replaced from a region beneath the gate in the p-type substrator.

- Data in the form of charge is transferred from the device to an adjacent one by one clocking their gates.
- This principle advantages of the CCD memory is that its single cell structure make it possible to construct large capacity memories at low cost.
- Since data are stored serially the average access time is long compared with the semi-conductor RAM memory.

③ Explain classification of memory.



Classification of semiconductor			
Non Volatile memory	Volatile memory		
ROM	Read/Write memory	Read/Write Memory	
Mask Programmable ROM	<ul style="list-style-type: none"> • EPROM • EEPROM 	<ul style="list-style-type: none"> Random access SRAM 	<ul style="list-style-type: none"> Non-Random access FIFO
Programmable Rom	FLASH	DRAM	<ul style="list-style-type: none"> LIFO Shift register

- Above table shows classification of semi-conductor memory devices the semi-conductor memory device can be categorized in several types according to their functional and architecture characteristics
- Broadly SC memories are classified as ~~volatile~~ volatile & non volatile memories.
- Volatile memories can retain their state as long as power is applied. On the other hand non-volatile memories can hold data even if power is turned off.
- The volatile memory which can hold data as long as power is ON are called static RAM, dynamic RAMs stores the data as a charge on the capacitor.
- They need refreshing of charge on the capacitor after every few milliseconds to hold the data even if power is ON.
- EPROM and EEPROM are erasable memories in which the stored data can be erased and new data can be stored.

Q) Describe Semiconductor.

- Semiconductor is a substance that specifies 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 conductor electricity under certain conditions but not others.
- This makes it an ideal medium to control

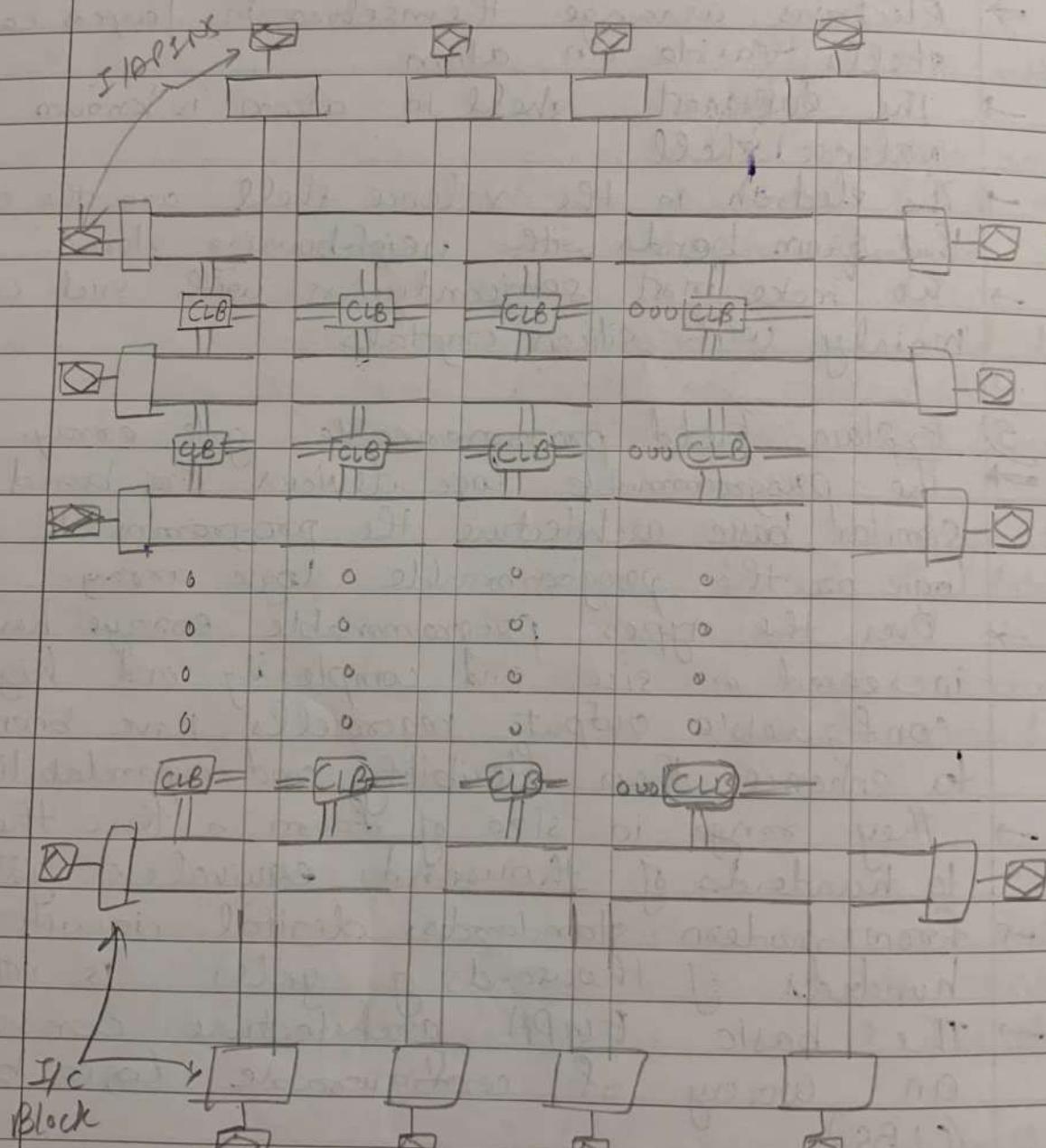
- To better understand how semiconductor work, users must understand atoms and how electrons organize themselves within the atom.
- Electrons arrange themselves in layers called shells inside an atom.
- The outermost shell in atom is known as valence shell.
- The electron in the valence shell are the ones that form bonds with neighbouring atoms.
- We make most semiconductors with such crystals, mainly with silicon crystals.

③ Explain Field programmable gate array.

- The programmable logic devices are based on similar basic architecture the programmable array logic or the programmable logic array.
- Over the types programmable arrays have increased in size and complexity and highly configurable output macrocells have been added to enhance their flexibility and expandability.
- They range in size of from a few thousand to hundreds of thousands equivalent gates
- From modern standards digital circuits with hundreds of thousands of gates is not too large
- The basic FPLA architecture consists of an array of configurable logic blocks (CLBs)
- The logic blocks are surrounded by configurable input/output blocks.
- There are rows and columns of programmable

interconnection paths

→ The I/O blocks can be individually configured as input, output or bidirectional.



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.

THE LOGIC AND FUNCTION:

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

THE LOGIC OR FUNCTION:

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

THE LOGIC NOT FUNCTION:

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

THE LOGIC EXCLUSIVE-OR FUNCTION:

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

THE LOGIC EXCLUSIVE-NOR FUNCTION:

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

THE LOGIC NANDFUNCTION:

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

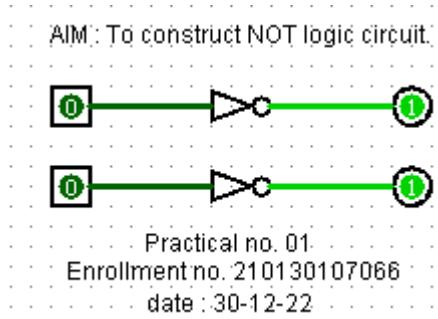
THE LOGIC NOR FUNCTION:

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

Code:

Output snapshot: (In output include practical details and execution date & time with your enrollment number)

NOT gate:

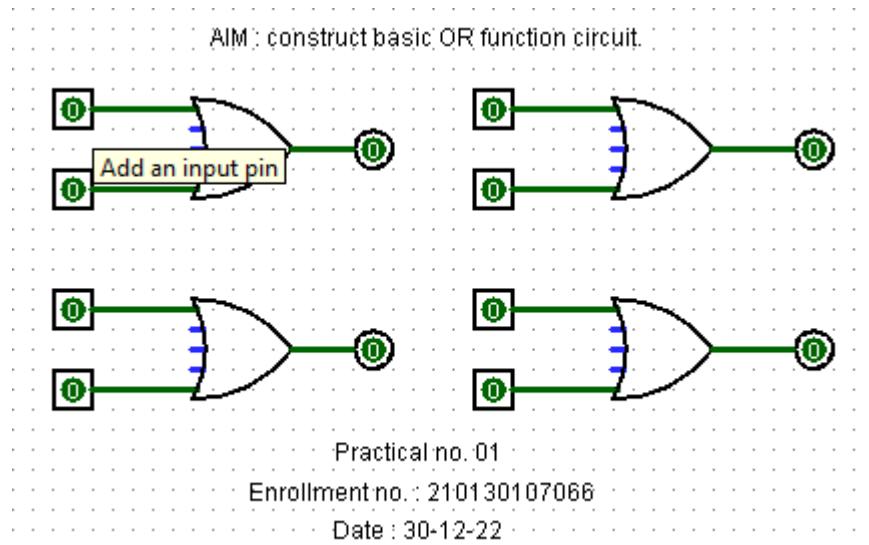


- Truth Table:

A	C
0	1
1	0

- Algorithm:

OR gate:

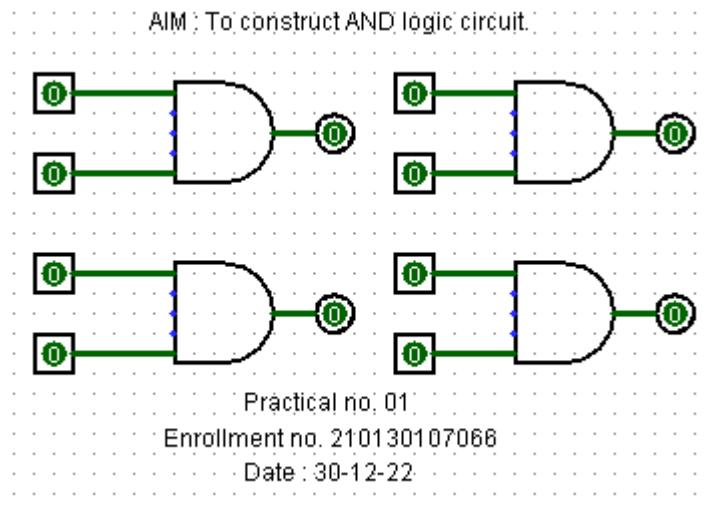


- Truth Table:

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

- Algorithm:

AND gate:

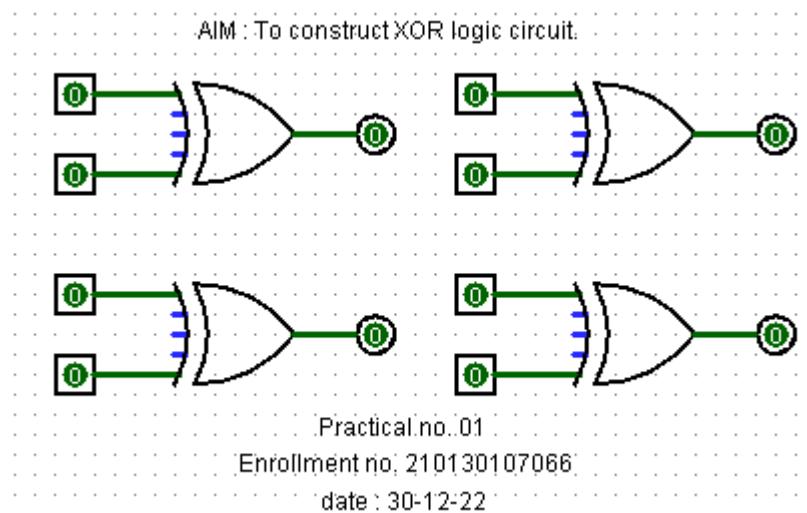


- Truth Table:

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

- Algorithm:

Exclusive-OR gate:



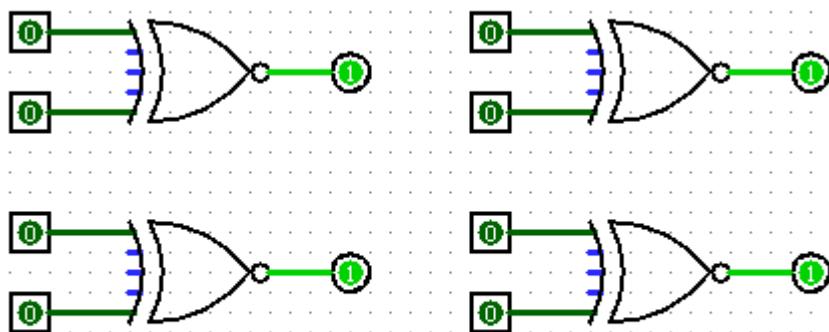
- Truth Table:

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

- Algorithm:

Exclusive-NOR gate:

Aim: To construct XNOR logic circuit.



Practical no.: 01

Enrollment no.: 210130107066

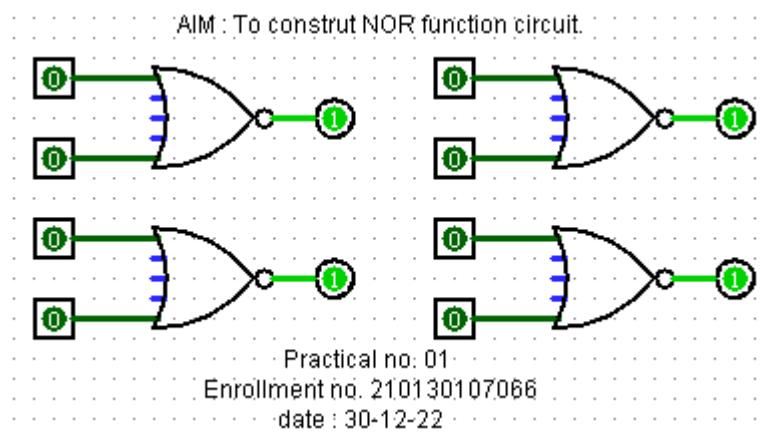
Date: 30-12-22

- Truth Table:

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

- Algorithm:

NOR gate:

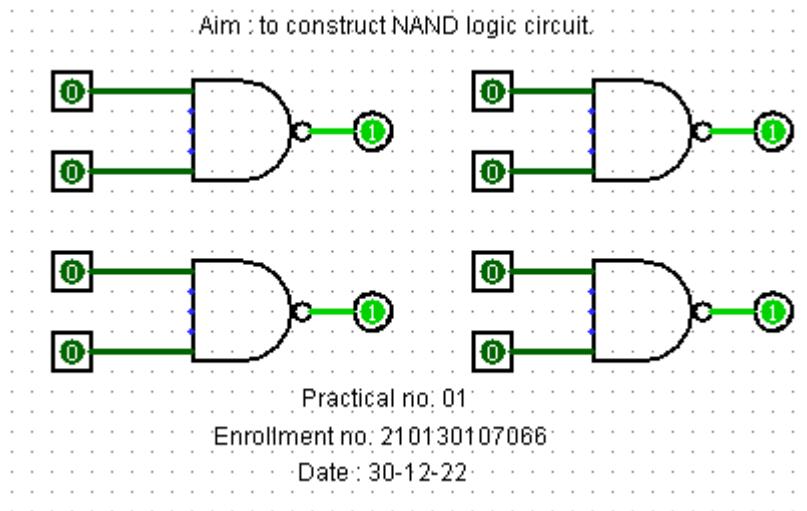


- Truth Table:

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

- Algorithm:

NAND gate:



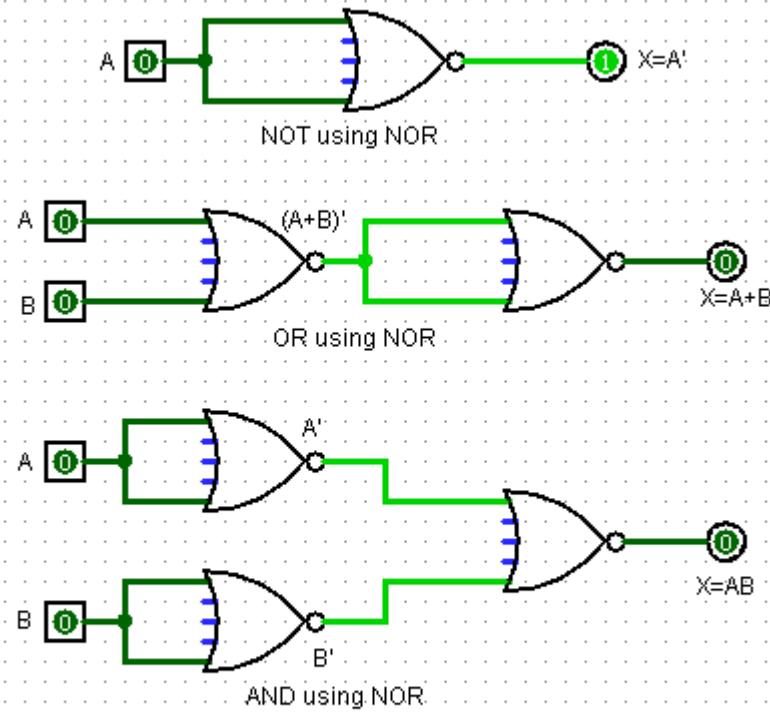
- Truth Table:

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

- Algorithm:

NOR as Universal gate:

AIM : Use NOR gate as universal to construct NOT,OR,AND gate.



Practical No: 01

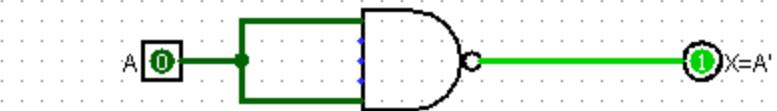
Enrollment no.: 210130107066

Date : 30-12-22

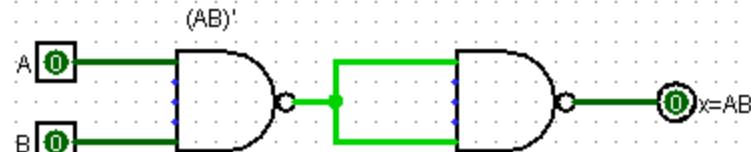
- Algorithm:

NAND as Universal gate:

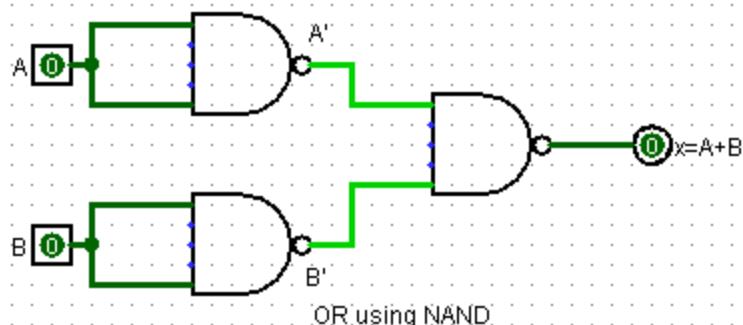
AIM : Use NAND gate as universal gate to construct NOT,AND,OR gate.



NOT using NAND



AND using NAND



OR using NAND

Practical no. 01
Enrollment no. 210130107066
Date : 30-12-22

- Algorithm:

15. Practical 2

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

Module 2

Aim: Implement half and full Adders using logic gates.

Theory:

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

Half Adder:

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

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

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

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

Full Adder:

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

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

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

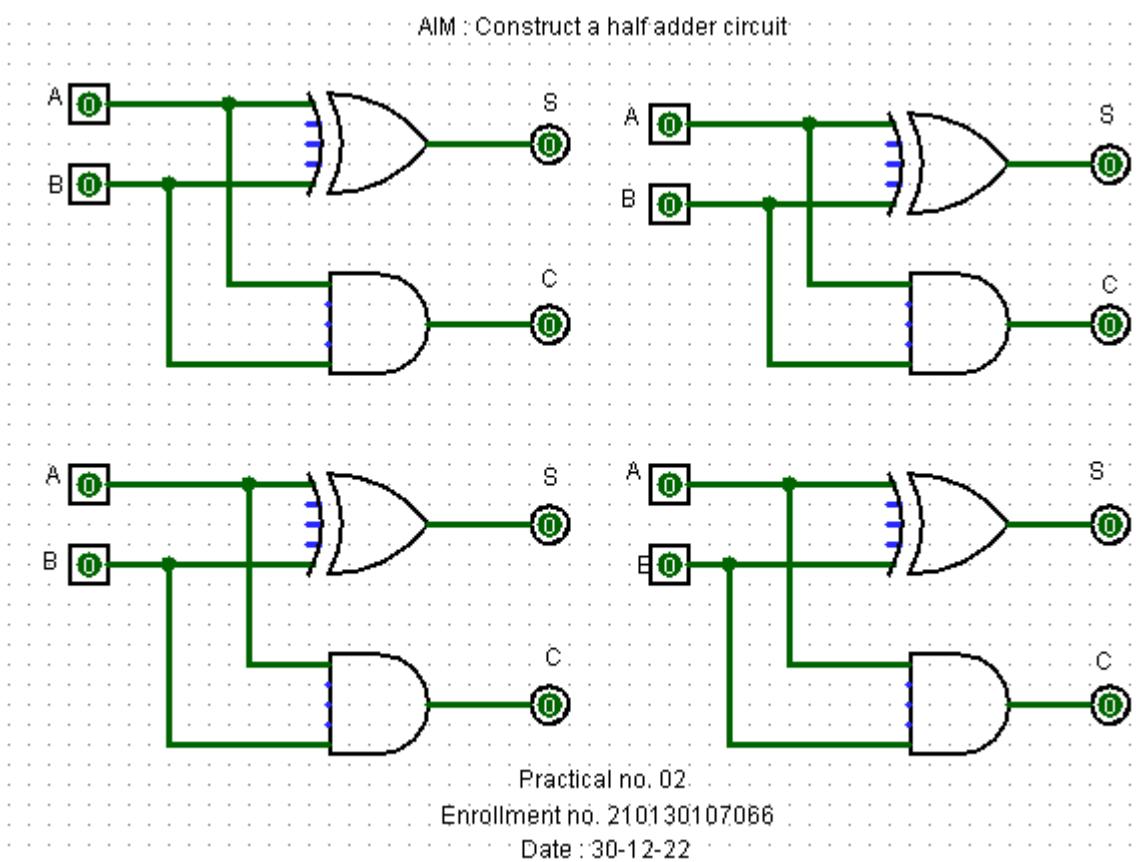
Boolean Expression: Difference S = $X \oplus Y \oplus Z$

$$\text{Borrow } B_{out} = X' (Y \oplus Z) + YZ$$

Code:

Output snapshot: (In output include practical details and execution date & time with your enrollment number)

Half Adder:



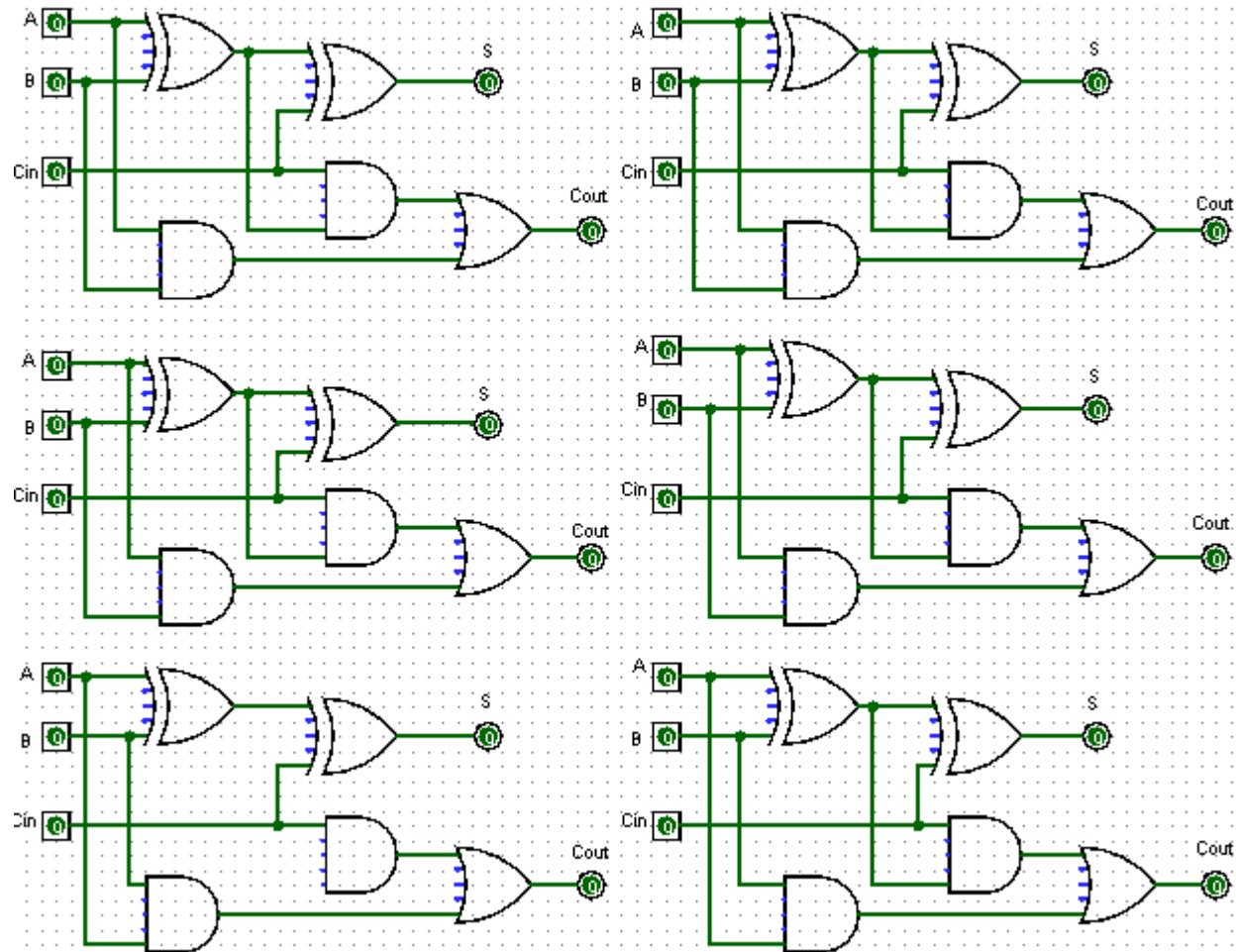
- Truth Table:

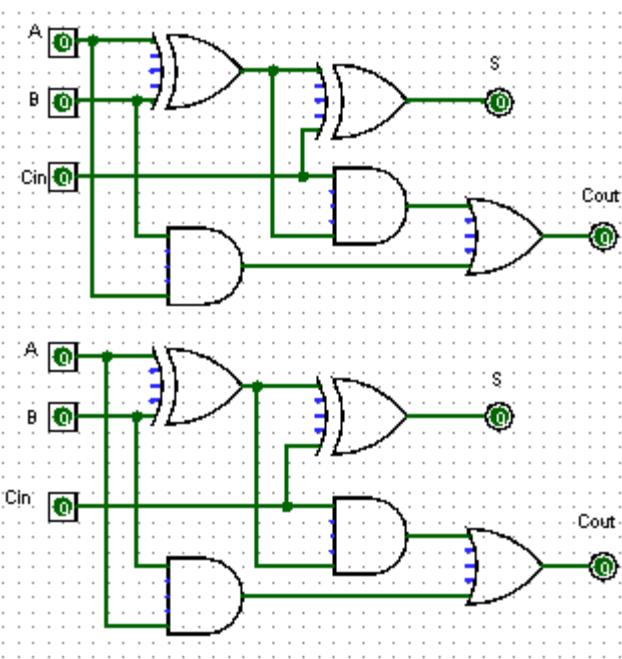
Inputs		Outputs	
A	B	Sum	Carry
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

- Algorithm:

Full Adder:

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





Prac No. 02
Enrollment no. 210130107066
Date : 30-12-22

- Truth Table:

Inputs			Outputs	
A	B	C _{in}	S	C _{out}
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

- Algorithm:

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:

Sub tractor circuit is a combinational digital circuit that issued for subtracts two numbers. A typical sub tractor circuit produces a Difference (D) and a Borrow (B) as the output.

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

$$1. \quad A - B = A + (\text{2's compliment of } B)$$

$$2. \quad A - B = A + (\text{1's compliment of } B) + 1 \quad (\text{here 1 is nothing but EAC})$$

Sub tractor circuits are of two types namely Full Sub tractor and Half Sub tractor.

Half Sub tractor:

- Half Sub tractor 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. From this it is clear that a half Sub tractor circuit can be easily constructed using one X-OR gate, one NOT gate and one AND gate.
- Half Subtractor is the simplest of all Subtractor circuit, but it has a major disadvantage. The half Subtractor can Subtract only two input bits (X and Y) and has nothing to do with the borrow if there is any in the input. So, if the input to a half Subtractor have a borrow, then it will be neglected it and subtracts only the A and B bits.
- That means the binary addition process is not complete and that's why it is called a half Subtractor. Boolean Expression:

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

Full Subtractor:

- Full Subtractor is little more difficult than a half- Subtractor circuit. The main difference in both Subtractor is that the full- Subtractor has three inputs and half- Subtractor has only two inputs.
- We have used three input variables X, Y and Z (Bin) which refers to the term minuend(X), subtrahend(Y) and borrow (Z or Bin) bit respectively. The output borrow is designated as *Bout* and the normal output is designated as D.

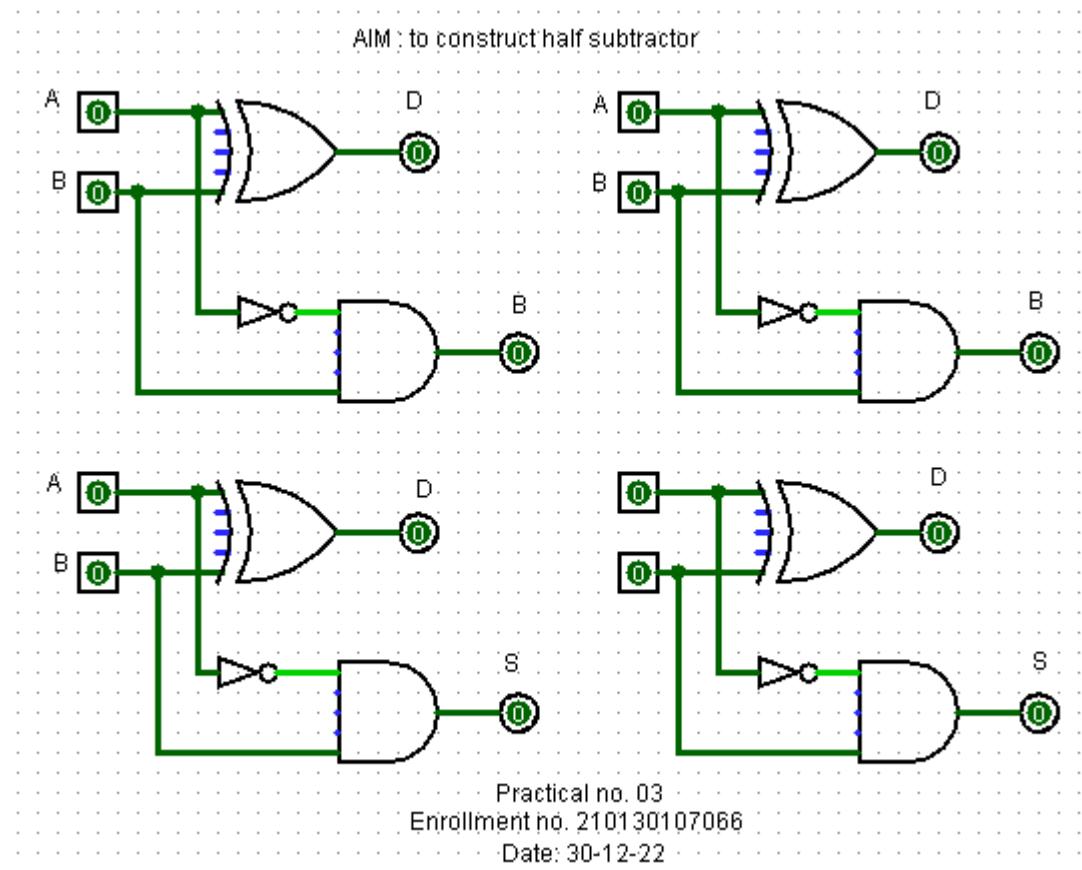
- Though the implementation of larger logic diagrams is possible with the below full adder logic a simpler symbol is mostly used to represent the operation. Given above is a simpler schematic representation of a one-bit full adder.

- Boolean Expression: Difference $S = X \oplus Y \oplus Z$
Borrow $Bout = X' (Y \oplus Z) + YZ$

Code:

Output snapshot: (In output include practical details and execution date & time with your enrollment number)

Half Subtractor:

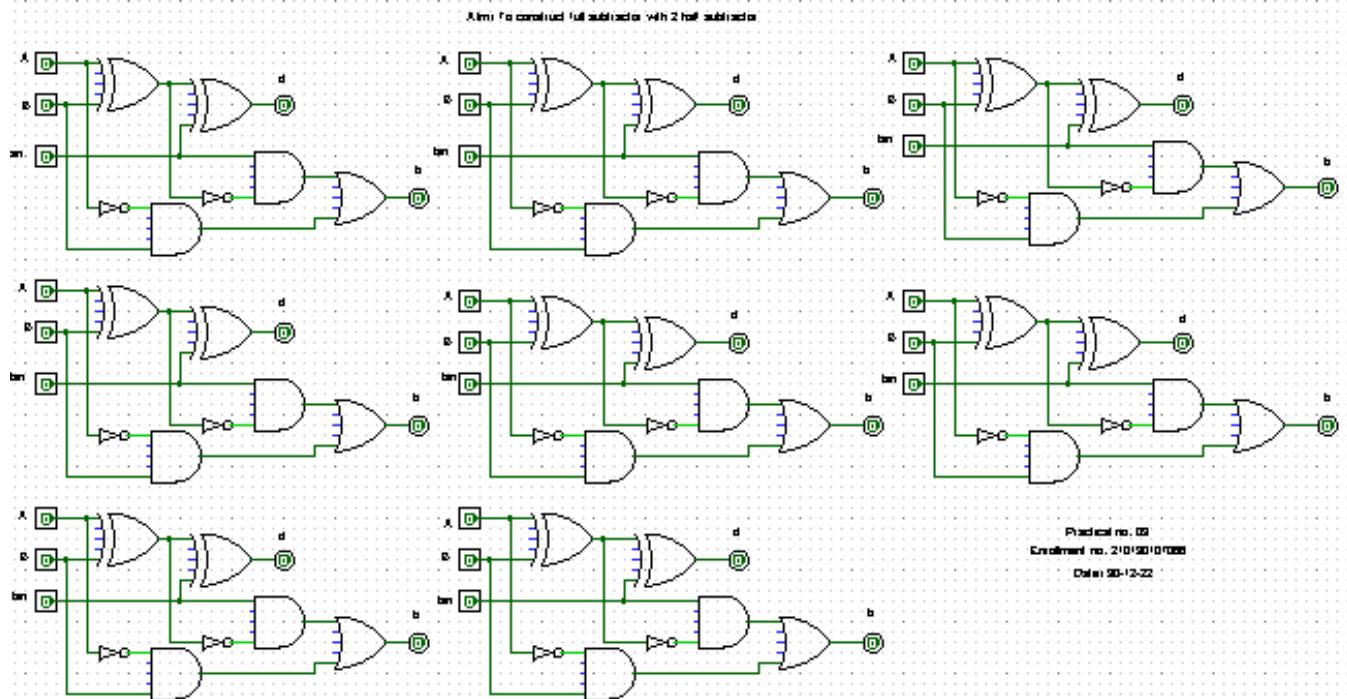


- Truth table:

Inputs		Outputs	
A	B	d	b
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0

- Algorithm:

Full Subtractor:



- Truth table:

Inputs			Outputs	
A	B	b _i	d	b
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	0	1
1	0	0	1	0
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1

- Algorithm:

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:

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

Even Parity Checker:

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

(E) =1, error occurs

Error Bit (E) =0, no error

Odd Parity Checker:

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

Error Bit (E) =1, no error

Error Bit (E) =0, error occurs

For Even Parity Checker:

K-map as shown below

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

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

The above truth table even parity checker:

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

For Odd Parity Checker:

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

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

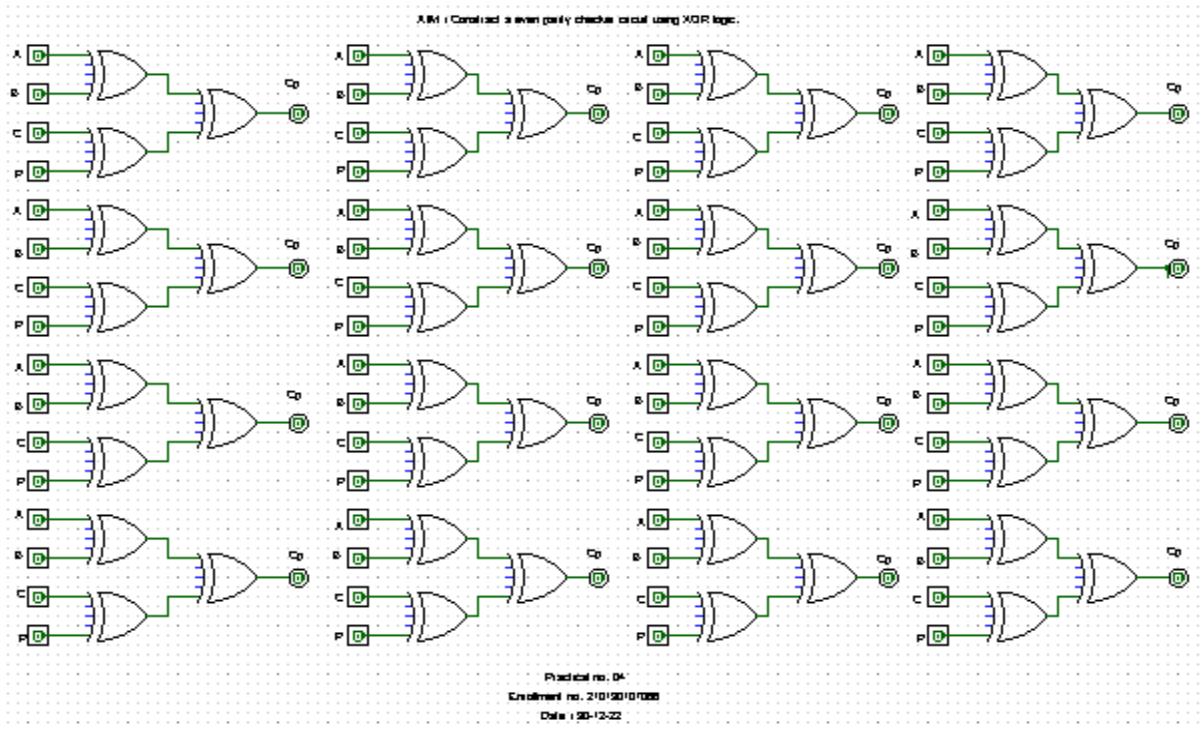
		CP	00	01	11	10
		AB	00	01	11	10
00	01	00	1	0	1	0
		01	0	1	0	1
11	10	00	1	0	1	0
		01	0	1	0	1

After simplification, the final expression for the PEC is obtained as
PEC = (A Ex-NOR B) Ex-NOR (C Ex-NOR D)

Code:

Output snapshot: (In output include practical details and execution date & time with your enrollment number)

Even Parity Checker:

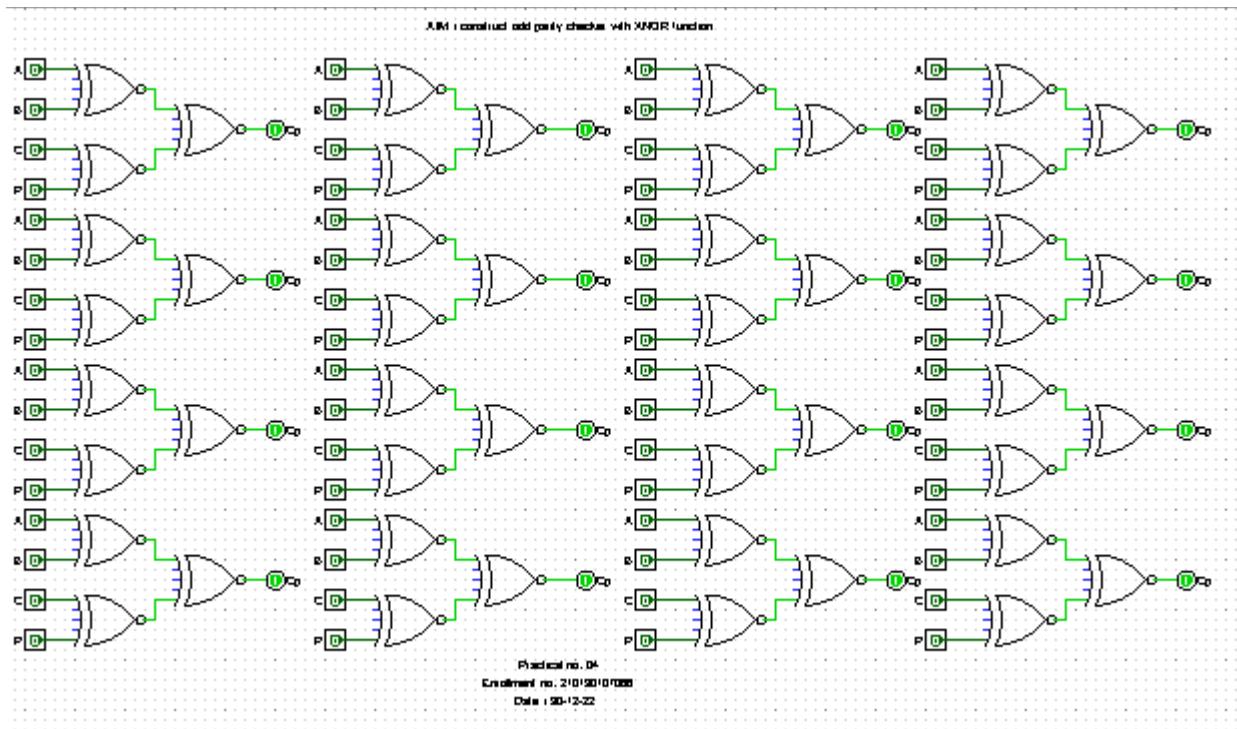


- Truth Table:

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

- Algorithm:

Odd Parity Checker:



Truth Table:

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

Algorithm:

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:

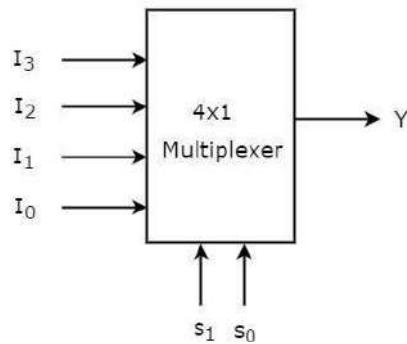
Multiplexer:

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

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

4x1 Multiplexer

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



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

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

From Truth table, we can directly write the **Boolean function** for output, Y as

$$Y = S_1'S_0'I_0 + S_1'S_0I_1 + S_1S_0'I_2 + S_1S_0I_3$$

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

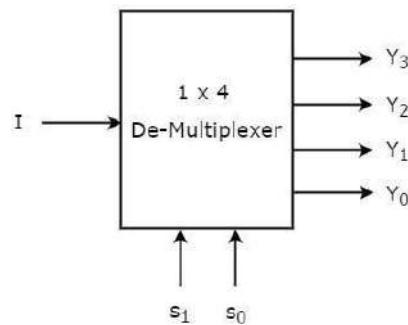
De-Multiplexer:

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

Since there are 'n' selection lines, there will be 2^n possible combinations of zeros and ones. So, each combination can select only one output. De-Multiplexer is also called as De-Mux.

1x4 De-Multiplexer:

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



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

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

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

$$Y_3 = s_1 s_0 I$$

$$Y_2 = s_1 s_0' I$$

$$Y_1 = s_1' s_0 I$$

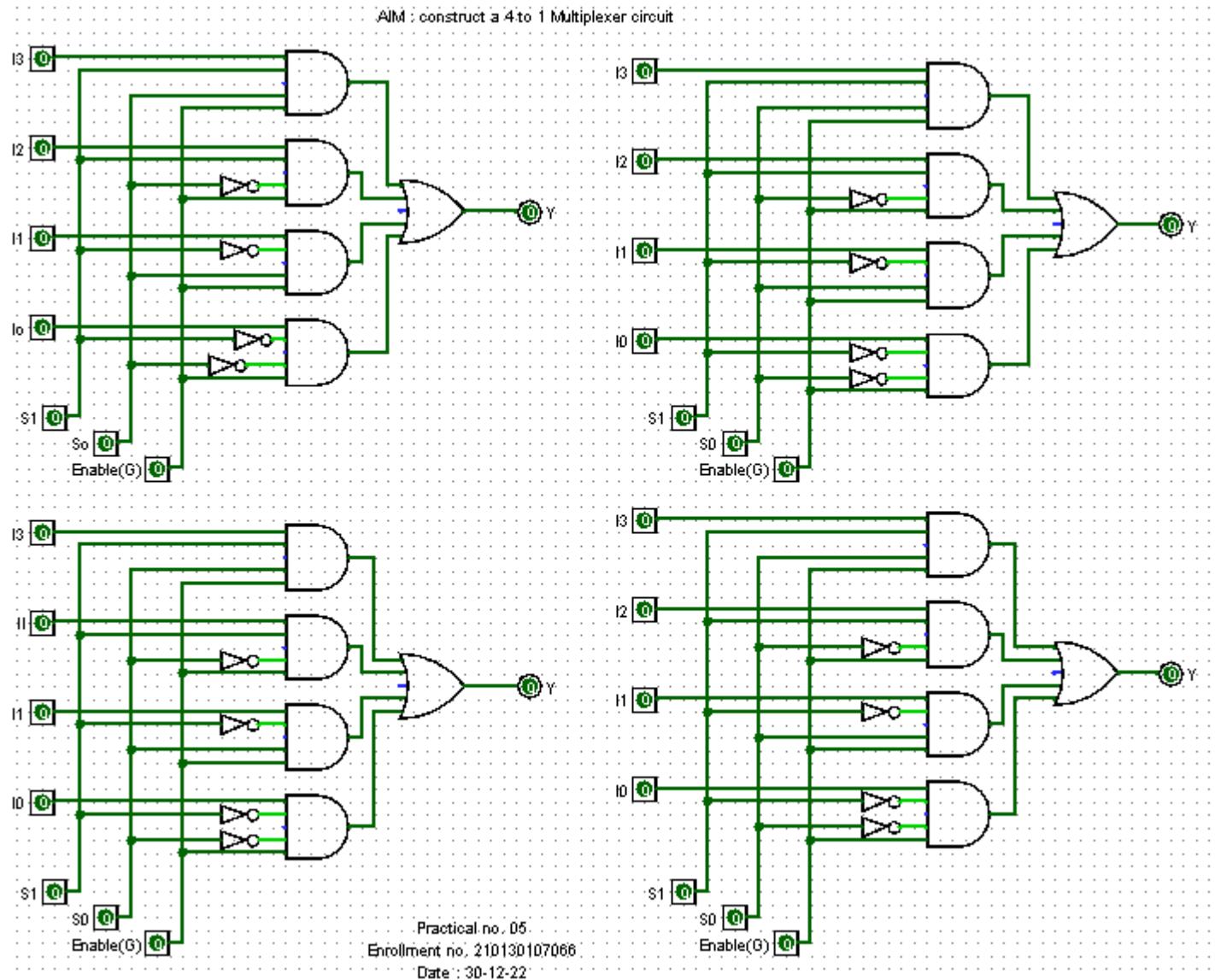
$$Y_0 = s_1' s_0' I$$

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

Code:

Output snapshot: (In output include practical details and execution date & time with your enrollment number)

1 to 4 Multiplexer:

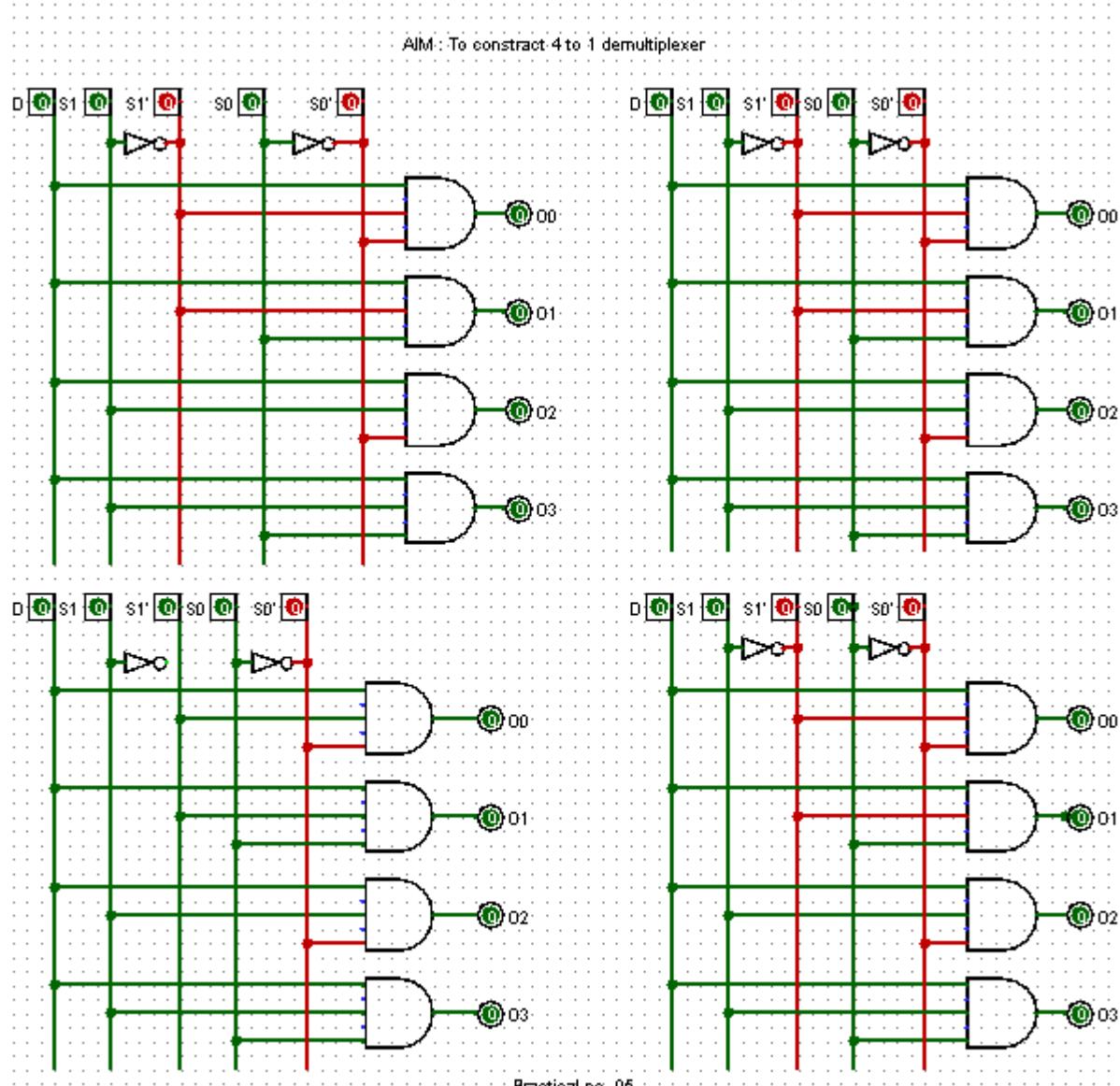


- Truth table:

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

- Algorithm:

4x1 Demultiplexer:



- Truth table:

Selection Inputs		Outputs			
s_1	s_0	O_3	O_2	O_1	\bar{O}_0
0	0	0	0	0	1
0	1	0	0	1	0
1	0	0	1	0	0
1	1	1	0	0	0

- Algorithm:

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. The START COMMAND, initiates the conversion process. The op-amp compactor has two analog inputs and a digital output that switches states, depending on which analog input is greater.

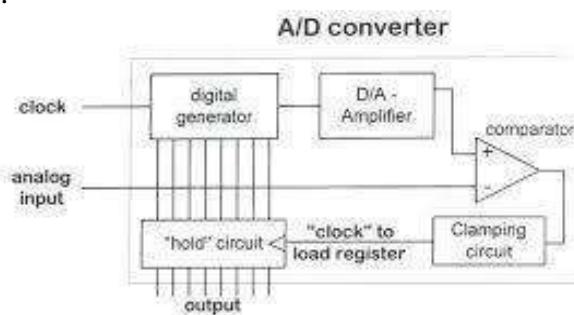


Figure 18.3 The A/D circuit layout.

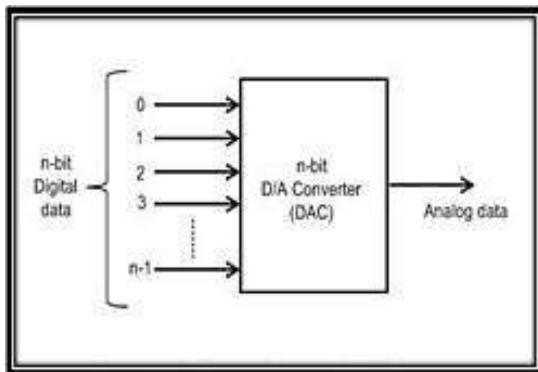
Digital-to-Analog Converter (DAC)

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

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

Digital to Analog (D to A) Conversion:

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



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

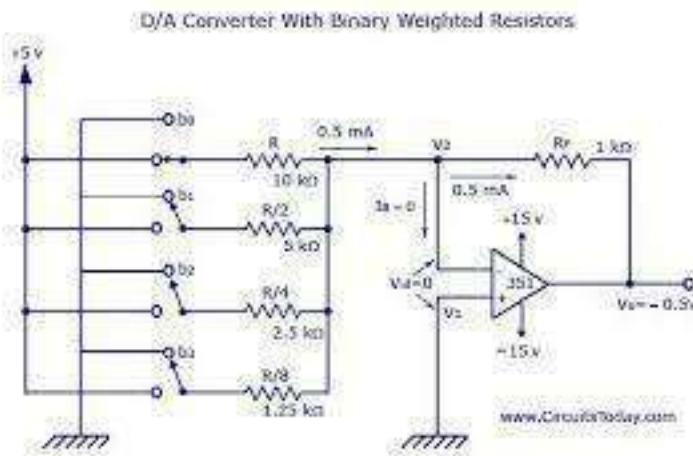


Figure shows the basic circuit of 4-bit DAC. The inputs A,B,C, and D are binary inputs which are assumed to have values of either 0 V or 5 V. The operational amplifier is employed as a summing amplifier, which produces

The purpose of a digital- to-analog converter is to convert a binary word to a proportional current or voltage.

The weighted sum of these input voltages. The summing amplifier multiplies each input voltage by the ratio of the feedback resistor R_F to the corresponding input resistor R_{IN} . In this circuit $R_F = 1k\Omega$ and the input resistors range from 1 to $8k\Omega$. The D input has R

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

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

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

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

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

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

Code:

Output snapshot: (In output include practical details and execution date & time with your enrollment number)

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:

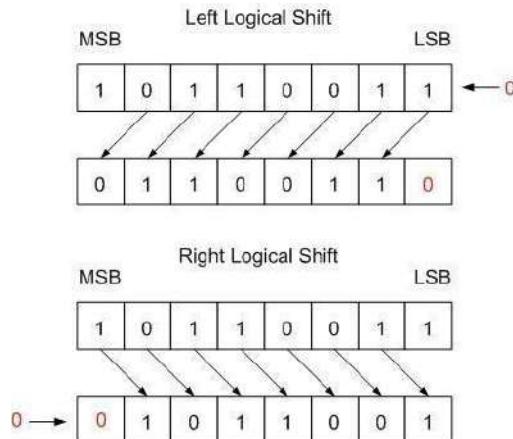
Shifters: -

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

There are several kinds of commonly used shifters: -

1) Logical Shifter:

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



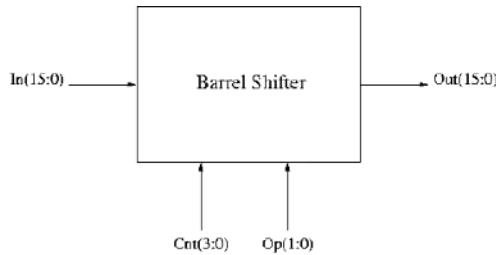
2) Arithmetic shifter:

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

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

3) Barrel Shifters

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



4) Frequency Shifter

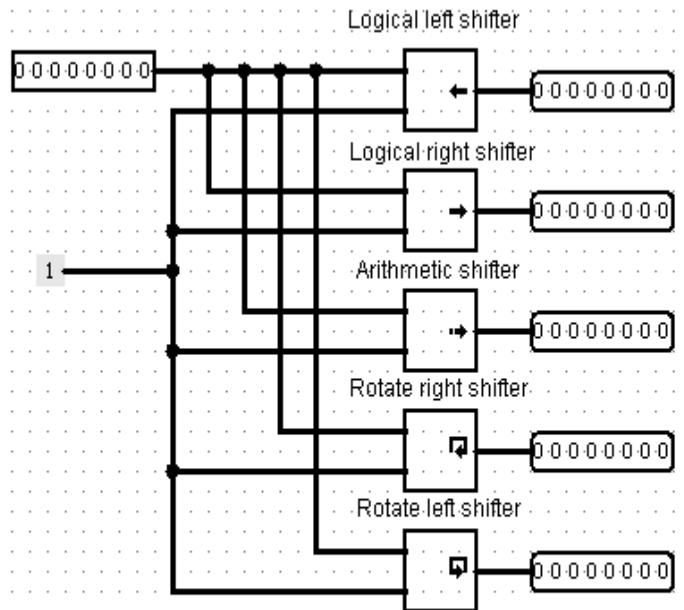
As demonstrated in Figure the conversion from LP01 to LP11 occurs with a frequency shift equal to the acoustic frequency. Thus, the setup in Figure acts as an optical frequency shifter. The fact that the frequency-shifted light is in the LP11 mode is inconvenient for coupling into a single-mode fiber.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

Code:

Output snapshot: (In output include practical details and execution date & time with your enrollment number)

Shifter:

AIM : to construct different type of shifter

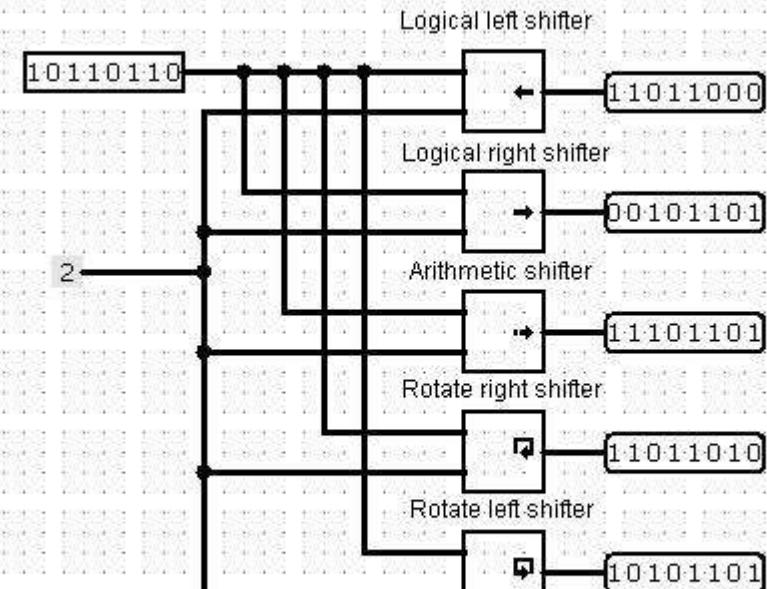


Practical no. 07

Enrollment no. 210130107066

Date : 30-10-22

AIM : to construct different type of shifter



Prac no. 07

Enrollment no. 210130107066

Date : 30-10-22

- Algorithm:

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 fundamental building blocks of digital electronics systems used in computers, communications, and many other types of systems. Flip-flops and latches are used as data storage elements. A flip-flop is a device which stores a single bit (binary digit) of data; one of its two states represents a "one" and the other represents a "zero". Such data storage can be used for storage of state, and such a circuit is described as sequential logic in electronics.

SR flip flop:

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

JK flip flop:

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

D flip flop:

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

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

But in order to prevent this from happening an inverter can be connected between the "SET" and the "RESET" inputs to produce another type of flip flop circuit known as a Data Latch, Delay flip flop, D-type Bistable, D type Flip Flop or just simply a D Flip Flop as it is more generally called.

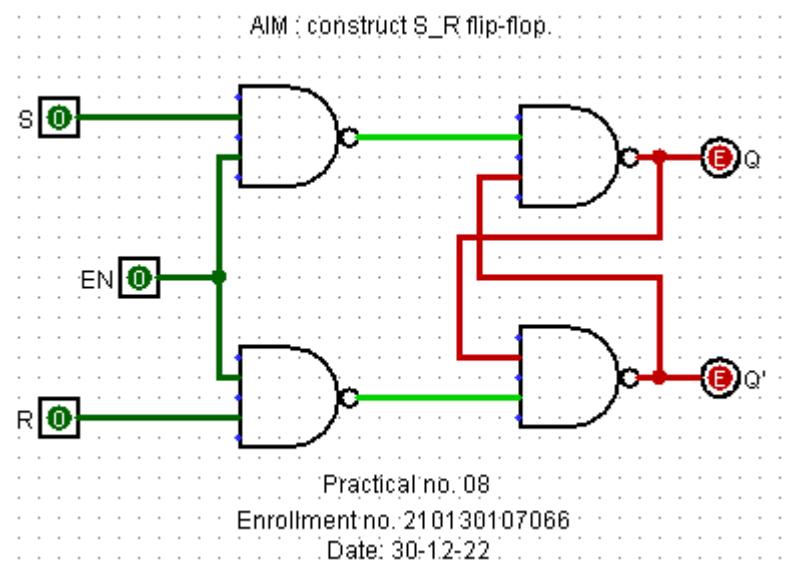
T flip flop:

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

Code:

Output snapshot: (In output include practical details and execution date & time with your enrollment number)

S-R Flip-Flop:

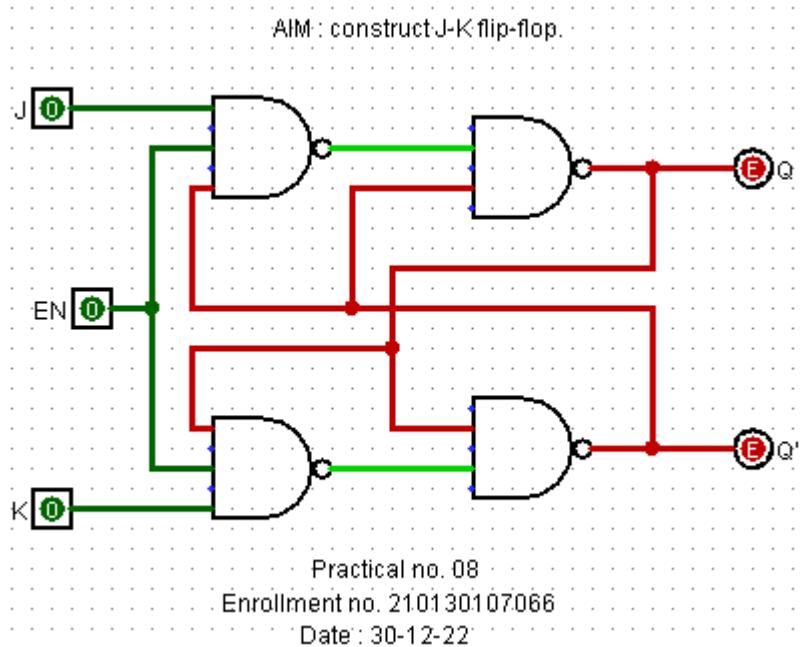


- Truth Table:

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

- Algorithm:

J-K Flip-Flop:

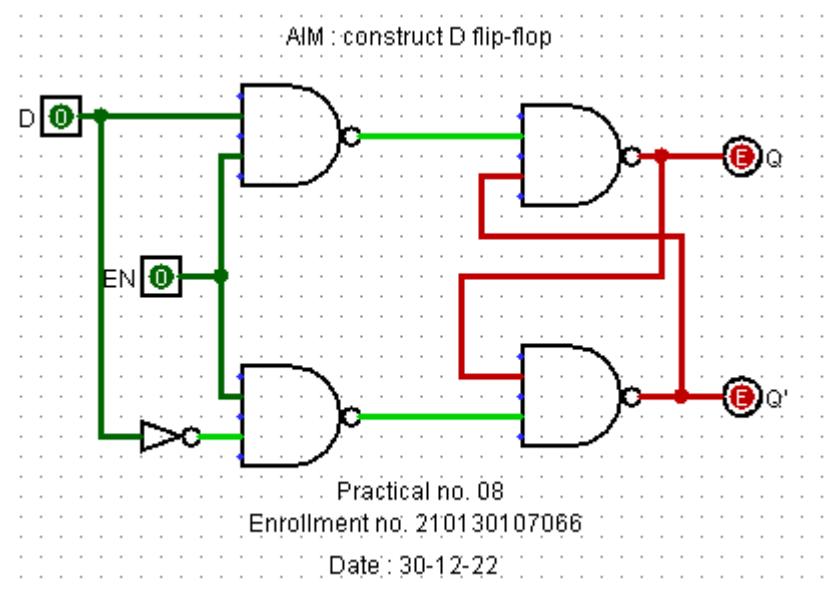


- Truth Table:

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

- Algorithm:

D Flip-Flop:



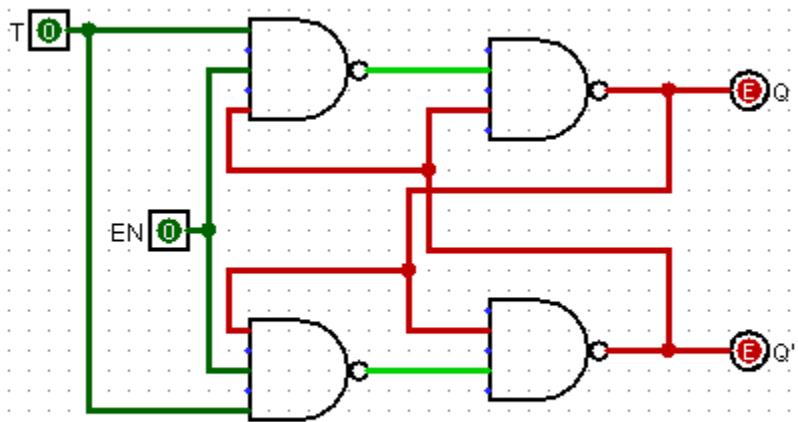
- Truth table:

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

- Algorithm:

T Flip-Flop:

AIM : construct T flip-flop



Practical no. 08

Enrollment no. 210130107066

Date : 30-12-22

- Truth table:

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

- Algorithm:

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

- Asynchronous counters
- Synchronous counters

Asynchronous counters:

An Asynchronous counter can count using **Asynchronous clock input**. Counters can be easily made using flip-flop. As the count depends on the clock signal, in case of an Asynchronous counter, changing state bits are provided as the clock signal to the subsequent flip-flops. Those Flip-flops are serially connected together, and the clock pulse ripples through the counter. Due to the ripple clock pulse, it's often called a ripple counter. An Asynchronous counter can count $2^n - 1$ possible counting states.

There are two types of Asynchronous counter:

- (i) Asynchronous up counter
- (ii) Asynchronous down counter

Synchronous counters

The synchronous counter also referred to as a parallel counter is the one in which each establishing flip flops are clocked with the similar clock input at the same time. In the synchronous counter, all the flip-flops in the cascade network are independently linked to an external clock.

This supports the clocking of all the flip-flops constituting the counter simultaneously instant with a similar clock input. This represents the output of each flip-flop change in synchronization with the clock input. In the synchronous counter, the similar clock pulse is moved to the clock input of all the flip-flops. The clock signals made by all the flip-flops are equal to each other.

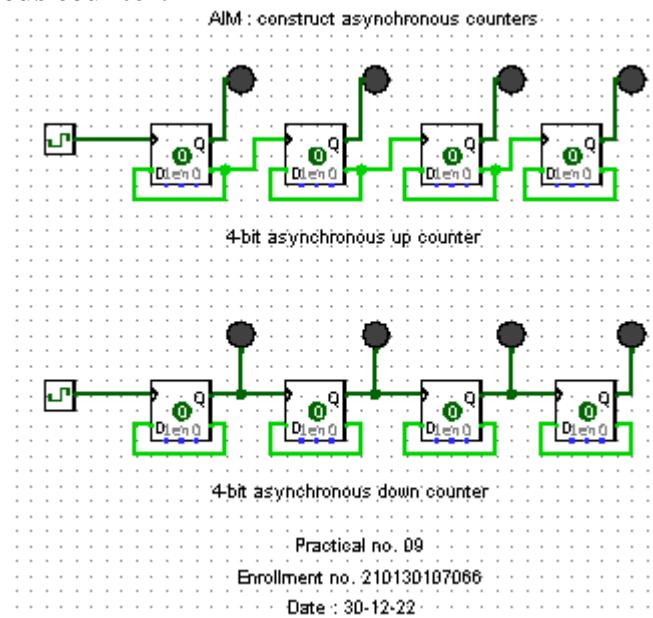
There are two types of synchronous counter:

- (i)synchronous up counter
- (ii)synchronous down counter

Code:

Output snapshot: (In output include practical details and execution date & time with your enrollment number)

Asynchronous counter:

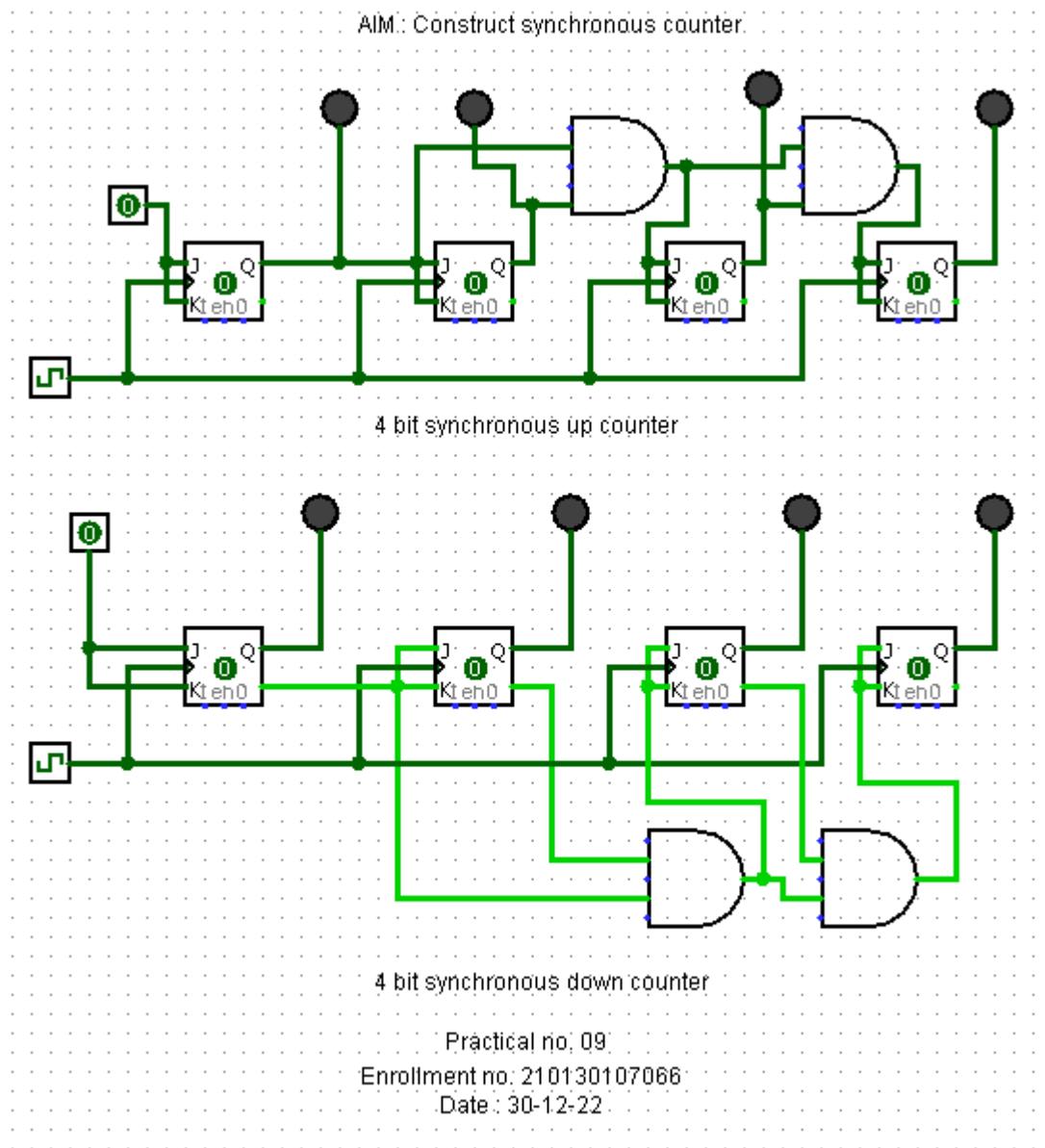


- Truth table:

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

- Algorithm:

synchronous counter:



- Truth table:

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

For up counter

For down counter

- Algorithm:

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 Serial-Out Shift Register (PISO):

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

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

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

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

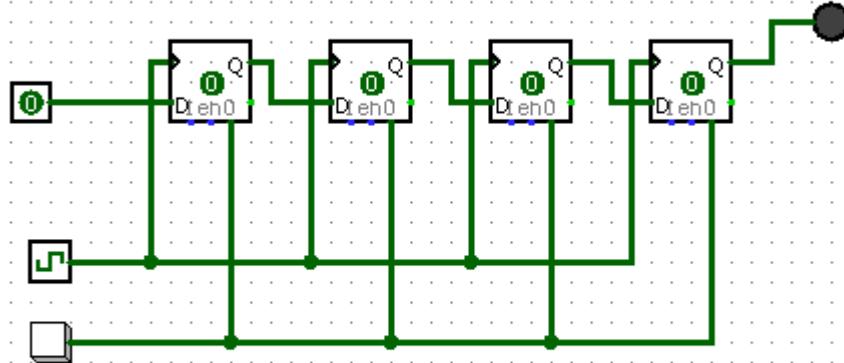
The logic circuit given below shows a parallel-in-parallel-out shift register. The circuit consists of four D flip-flops which are connected. The clear (CLR) signal and clock signals are connected to all the 4 flip flops. In this type of register, there are no interconnections between the individual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

Code:

Output snapshot: (In output include practical details and execution date & time with your enrollment number)

Serial-In Serial-Out Shift

SERIAL IN SERIAL OUT SHIFT REGISTER



Practical no. 10
Enrollment no.- 210130107066
Date- 30-12-22

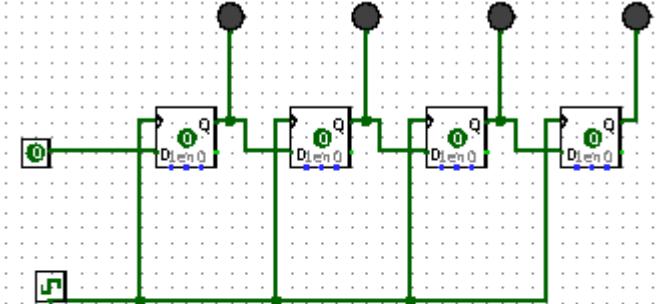
Register (SISO):

- Truth Table:

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

- Algorithm:

Serial-In Parallel-Out Shift Register (SIPO):



4 BIT SERIAL IN PARALLEL OUT SHIFT REGISTER

Practical no.: 10

Enrollment no.- 210130107066

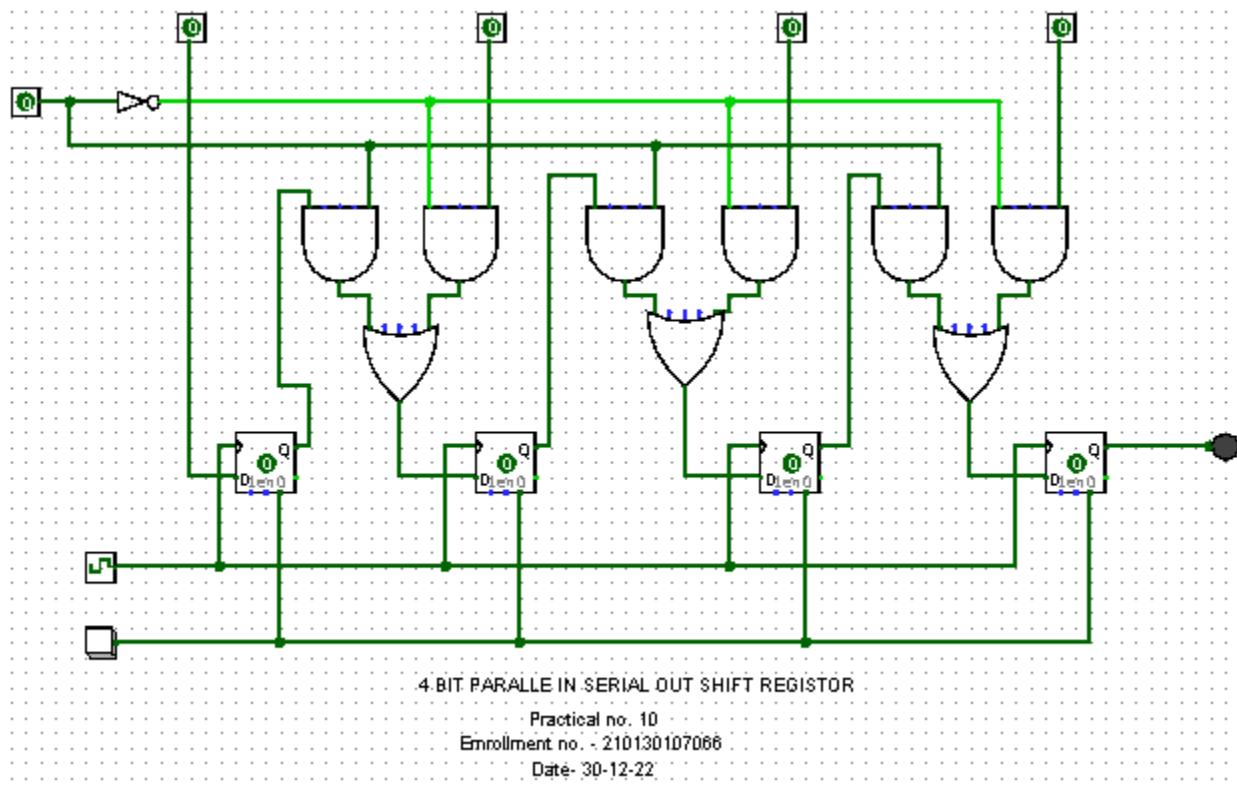
Date- 30-12-22

- Truth Table:

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

- Algorithm:

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



- Truth Table:

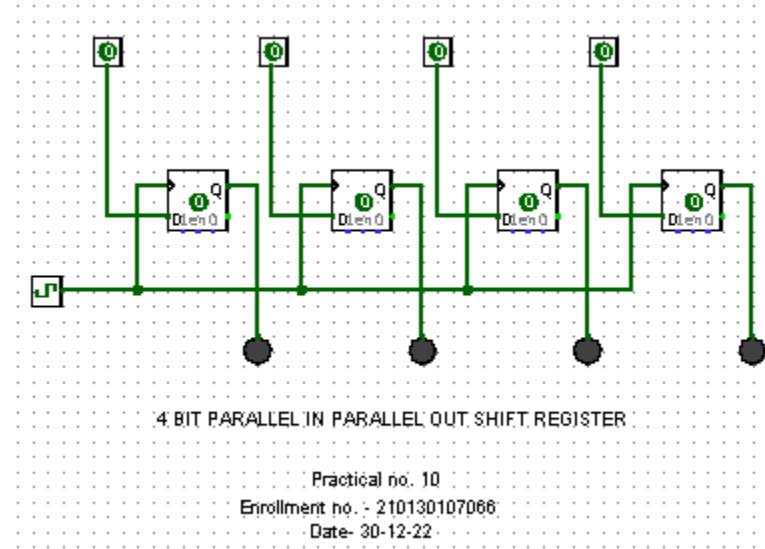
CLK	Q3	Q2	Q1	Q0	O/P
0	1	0	0	1	1
1	0	0	0	0	0
2	0	0	0	0	
3	0	0	0	0	

0

1

- Algorithm:

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



Truth Table:

CLK	DATA INPUT				OUTPUT			
	D _A	D _B	D _C	D _D	Q _A	Q _B	Q _C	Q _D
1	1	0	0	1	1	0	0	1
2	1	0	1	0	1	0	1	1

- Algorithm:

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)

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

Theory:

K-Map (Karnaugh Map)

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

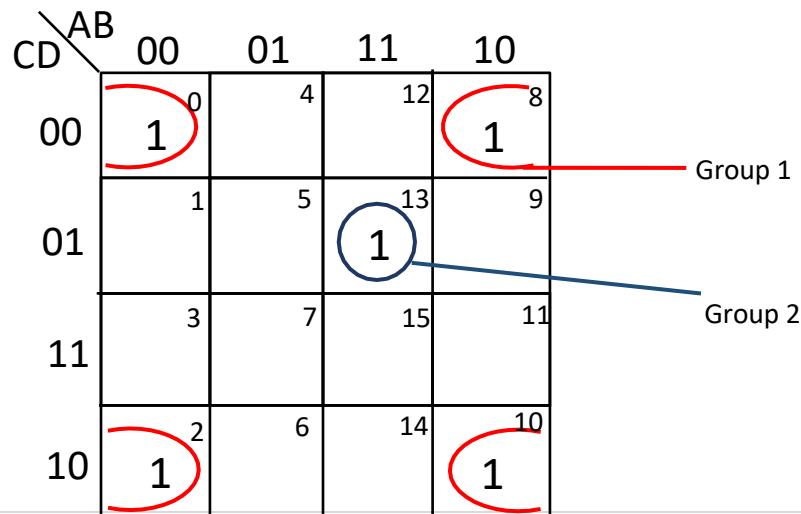
Code:

Output snapshot: (In output include practical details and execution date & time with your enrollment number)

K-Map:

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

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



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

First consider group 1,

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

Now, consider group 2,

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

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

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