

S.NO.	NAME OF EXPERIMENT	DATE OF EXPERIMENT	SIGN	MARKS	REMARK
1.)	logical Gates	10 Dec., 2020	AleD	9+	
2.)	NAND and NOR as Universal Gates	10 Dec., 2020	AleD	9+	
3.)	HALF ADDER	10 Dec., 2020	AleD	2	
4.)	HALF SUBTRACTOR	10 Dec., 2020	AleD	9+	
5.)	FULL ADDER	10 Dec., 2020	AleD	9+	
6.)	FULL SUBTRACTOR	10 Dec., 2020	AleD	2	
7.)	3x8 DECODER	7 Jan., 2021	AleD	9+	
8.)	2x4 DECODER	7 Jan., 2021	AleD	2	
9.)	4x1 MULTIPLEXER	14 Jan., 2021	AleD	B	
10.)	S-R LATCH	21 Jan., 2021	AleD	2	
11.)	D FLIP-FLOP	21 Jan., 2021	AleD	2	
12.)	5-bit binary code into Gray code	28 Jan., 2021	AleD	9+	
13.)	5-bit Gray code into Binary code	28 Jan., 2021	AleD	9+	

EXPERIMENT-1

Objective :- Implementing different logical gates to perform different operations.

Theory :- Logic Gates are the building blocks of any digital system. It is an electronic circuit having one or more input and only one output. The relationship between the input and the output is based on a certain logic.

Based on this, logic gates are of different types :-

1) AND Gate : The AND gate works in the same way as the logical operator "AND". The AND gate is a circuit that performs the AND operation of the inputs. It has a minimum of 2 input values and an output value.

$$Y = A \text{ AND } B \text{ AND } C \text{ AND } D \dots N$$

$$Y = A \cdot B \cdot C \cdot D \dots N$$

for 2 input variables, say A and B, Y as an output variable.

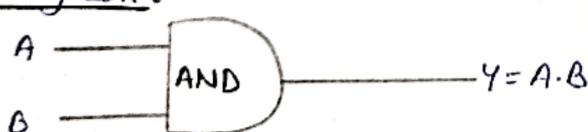
Truth table :-

Inputs		Output
A	B	Y
0	0	0
0	1	0
1	0	0
1	1	1

If both are 1, the output is 1; otherwise, the output is 0.

Boolean Expression :- $Y = A \cdot B = AB$

Circuit diagram :-



2) OR Gate : This gate works in the same way as the logical operator "OR". The OR gate is a circuit which performs the OR operation of the inputs. This gate also has a minimum of 2 input values and an output value.

If both are 0, the output is 0; otherwise, the output is 1.

Truth Table :-

Inputs		Output
A	B	Y
0	0	0
0	1	1
1	0	1
1	1	1

Boolean Expression :-

$$Y = A + B$$

Circuit diagram :-



3) NOT Gate :- The NOT gate is called as an inverter. This gate gives the inverse value of the input value as a result. This gate has only one input and one output value.

Truth table :-

Input	Output
0	1
1	0

Boolean expression :-

$$Y = \overline{A} = A'$$

Circuit diagram :-



4) NAND Gate :- The NAND gate is the combination of AND gate and NOT gate. This gate gives the same result as a NOT-AND operation. This gate can have two or more than two input values and only one output value.

If both are 1, the output is 0; otherwise, the output is 1.

Truth table :-

Inputs		Output
A	B	Y
0	0	1
0	1	1
1	0	1
1	1	0

Boolean Expression :-

$$Y = \overline{A \cdot B} = (A \cdot B)'$$

Circuit diagram :-



5.) NOR Gate :- The NOR gate is the combination of an OR gate and NOT gate. This gate gives the same result as the NOT-OR operation. This gate can have two or more than two input values and only one output value.

If both are 0, the output is 1; otherwise, the output is 0.

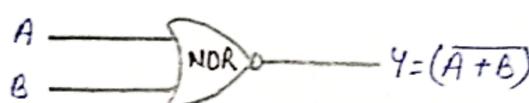
Truth table :-

Inputs		Output
A	B	Y
0	0	1
0	1	0
1	0	0
1	1	0

Boolean Expression :-

$$Y = (\overline{A+B})$$

Circuit diagram :-



6.) XOR Gate :- The XOR gate is also known as the Exclusive-OR / Ex-OR gate. The XOR gate is used in half and full adder and subtractor. The XOR gate is sometimes called Ex-OR or X-OR gate. This gate can have two or more than two input values and only one output value.

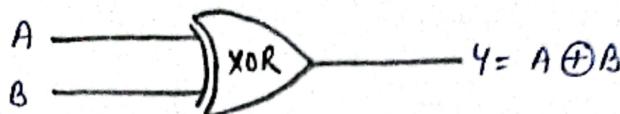
If both inputs are the same, the output is 0; otherwise, the output is 1.

Truth table :-

Inputs		Output
A	B	Y
0	0	0
0	1	1
1	0	1
1	1	0

Boolean Expression :-

$$Y = A \oplus B$$

Circuit diagram :-

7) X-NOR Gate:- The X-NOR gate is also known as the EX-NOR gate. The X-NOR gate is used in half and full adder and subtractor. The exclusive-NOR gate is sometimes called as EX-NOR and X-NOR gate. This gate can have two or more than two input values and only one output value.

If both inputs are the same, the output is 0; otherwise, the output is 1.

Truth table :-

Inputs		Output
A	B	Y
0	0	1
0	1	0
1	0	0
1	1	1

Boolean Expression :-

$$Y = (\overline{A \oplus B}) = A \otimes B$$

Circuit diagram :-

8) TRI-STATE BUFFER Gate:- It is a type of digital buffer circuit. It requires 1 input to amplify its power and produce same result.

Truth table :-

Input	Output
A	Y
0	0
1	1

Boolean Expression :-

$$Y = A$$

Circuit diagram :-

Procedure :- Procedure to implement different logic gates to perform different operations :-

- Step 1. Open logic.ly website in your browser and click on the link 'logi.ly'.
- Step 2. A logic simulator will be opened.
- Step 3. For different logic gates, there are different symbols and Boolean expression used to design circuits for gates.
- Step 4. In the simulator, as we know each basic logic gate has different number of inputs but all gives an only one output :-
 - NOT gate
 - Tru-State Buffer gate
 - AND gate
 - OR gate
 - NAND gate
 - NOR gate
 - XOR gate
 - X-NOR gate

} 1 input and 1 output

} 2, 3, ... inputs and 1 output.
- Step 5. Now, we will select desired gate from left hand-side.
- Step 6. Takes desired no. of Input bits from the input control section and choose desired binary input for verification.
- Step 7. For the output, we will go under the output control section and drag the bulb on white screen.
- Step 8. The Input will be connected to the input block and stretch it to the gate input node and the output will be connected to the output block of gate node and arrange the circuit as above.

EXPERIMENT-2

Objective :- Realization of NAND and NOR as universal gates.

Theory :- A universal gate is a gate which can implement any Boolean function without need to use any other gate type.

NAND as Universal Gate :-

1) NAND as NOT Gate :-

Circuit diagram :-

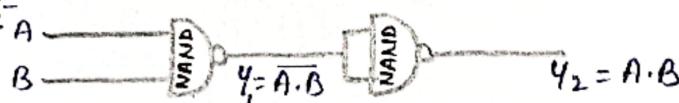


Truth table :-

Input	Output
A	Y
0	1
1	0

2) NAND as AND Gate :-

Circuit diagram :-

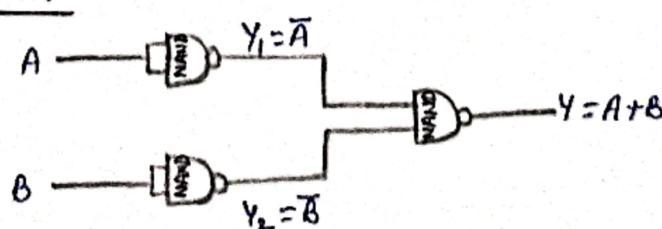


Truth table :-

Inputs		Y_1	Output
A	B		Y
0	0	1	0
0	1	1	0
1	0	1	0
1	1	0	1

3) NAND as OR Gate :-

Circuit diagram :-



Truth table :-

Inputs		$y_1 = \bar{A}$	$y_2 = \bar{B}$	Output
A	B			y
0	0	1	1	0
0	1	1	0	1
1	0	0	1	1
1	1	0	0	1

NOR as Universal Gate :-

1) NOR as NOT Gate :-

Circuit Diagram :-

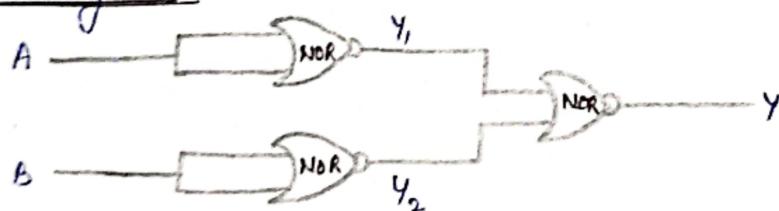


Truth table :-

Input	Output
A	y
0	1
1	0

2) NOR as AND Gate :-

Circuit Diagram :-

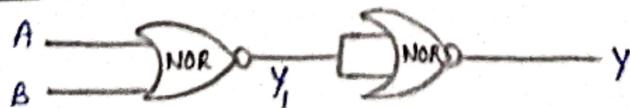


Truth table :-

Inputs		y_1	y_2	Output
A	B			y
0	0	1	1	0
0	1	1	0	0
1	0	0	1	0
1	1	0	0	1

3) NOR as OR Gate :-

Circuit Diagram :-



Truth-table :-

Inputs		y_1	Output
A	B		y
0	0	1	0
0	1	0	1
1	0	0	1
1	1	0	1

Procedure :- Procedure to implement of NAND and NOR gate as universal gates :-

Step 1. Open logic.ly website in your browser and click on the link 'try logic.ly'.

Step 2. A logic simulator will be opened.

Step 3. → For NAND as universal gate :-

NANDs are only used for designing circuit of NOT gate, AND gate and OR gate.

→ For NOR as universal gate :-

NORs are only used for designing circuit of NOT gate, AND gate and OR gate.

Step 4. In the simulator, as we know each basic logic gates has different number of inputs but only one output:-

→ NOT gate :- 1 input

→ AND gate :- 2 inputs, 3 inputs

→ OR gate :- 2 inputs, 3 inputs..... But, we will use 2 inputs here.

Step 5. Now, we will select desired gate from left-hand side (as in circuit diagram).

Step 6. Take two inputs from the input control section and choose desired binary input for verification (say, A and B).

Step 7. For the output, we will go under the output control section and drag the bulb on white screen.

Step 8. The input will be connected to the input block and stretch it to the gate input node and the ~~output~~ will be connected to the output block of gate node and arrange the circuit as above.

~~Step 8~~

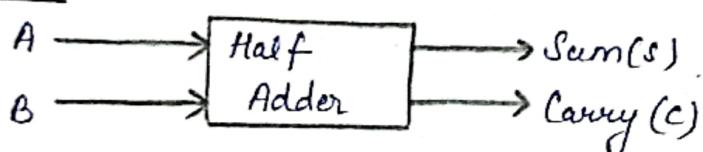
EXPERIMENT-3

Objective :- Implementing HALF ADDER using basic logic gates.

Theory :- An adder is a digital logic circuit in electronics that implements addition of numbers (bits).

The Half-adder is a basic building block of adding two numbers as two inputs and produce out two outputs. The adder is used to perform OR operation of two single bit binary numbers. The 'augend' and 'addend' bits are two input states, and 'carry' and 'sum' are two output states of the half adder.

Block Diagram :-



Truth table :-

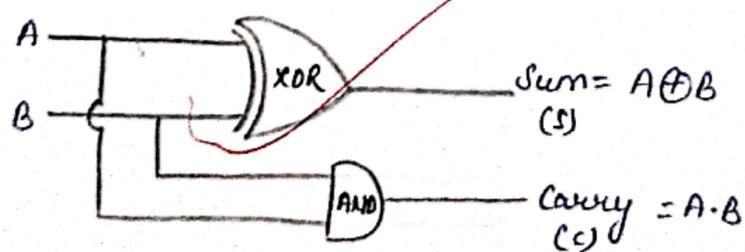
Inputs		Outputs	
A	B	Sum	Carry
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

Boolean Expression :-

$$\text{Sum}(s) = A'B + AB' = A \oplus B$$

$$\text{Carry}(c) = AB$$

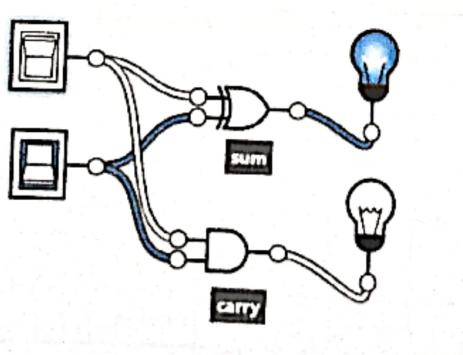
Circuit diagram :- The half adder is designed with the help of 2-input AND gate and 2-input XOR gate.



Procedure :- Procedure to implement HALF ADDER using basic logic gates:-

- Step 1. Open logic.ly website in your browser and click on the link "try logic.ly".
- Step 2. A logic simulator will be opened.
- Step 3. For HALF ADDER, two types of gates are used is XOR gate and AND gate.
- Step 4. In the simulator, as you all know HALF ADDER has two inputs and two outputs (sum and carry).
- Step 5. Now we will select desired gate from left hand side.
- Step 6. Take two input from the input control section and choose desired binary input for verification.
- Step 7. For the output we will go under the output control section and drag the bulb on white screen.
- Step 8. The input will be connected to the input block and stretch it to the gate input node and the output will be connected to the output block of gate node and arrange the circuit as above.

~~Delete~~

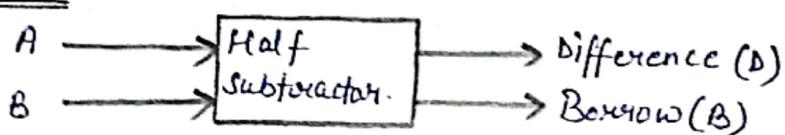


EXPERIMENT-4

Objective :- Implementing HALF SUBTRACTOR using basic logic gates.

Theory :- The half-subtractor is also a building block for subtracting two binary numbers. It has two inputs and two outputs. This circuit is used to subtract two single bit binary numbers A and B. The 'difference' and 'borrow' are two output states of the half subtractor.

Block Diagram :-



Truth table :-

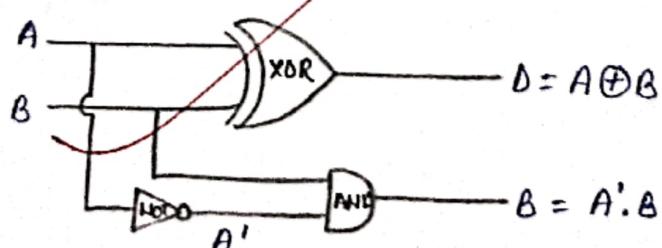
Inputs		Outputs	
A	B	Difference	Borrow
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0

Boolean Expression :-

$$\begin{aligned} \text{Difference (D)} &= A'B + AB' \\ &= A \oplus B \end{aligned}$$

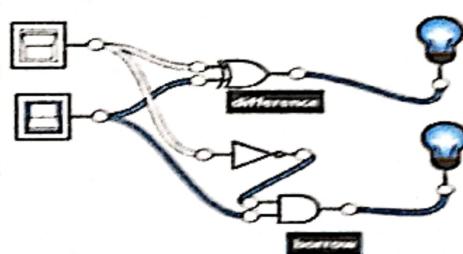
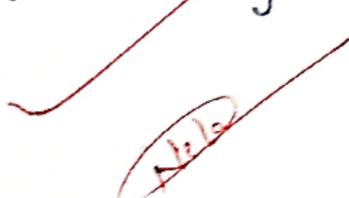
$$\text{Borrow (B)} = A'B$$

Circuit Diagram :- The Half-Subtractor is designed with the help of 2-input AND gate, 2-input XOR gate and NOT gate.



Procedure :- Procedure to implement HALF SUBTRACTOR using basic logic gates:-

- Step 1. Open logic.ly website in your browser and click on the link 'try logic.ly'.
- Step 2. A logic simulator will be opened.
- Step 3. For HALF SUBTRACTOR, three types of gates are used, that are :- XOR gate, AND gate and NOT gate, according to Boolean expression.
- Step 4. In the simulator, as we know HALF SUBTRACTOR has two inputs and two outputs (difference and borrow).
- Step 5. Now, we will select desired gate from left hand side.
- Step 6. Take two inputs from the input control section and choose desired binary input for verification (say, A and B (0,0), (0,1), (1,0) (1,1)).
- Step 7. For the output, we will go under the output control section and drag the bulb on white screen.
- Step 8. The input will be connected to the input block and stretch it to the gate input node and the output will be connected to the output block of gate node and arrange the circuit as above.

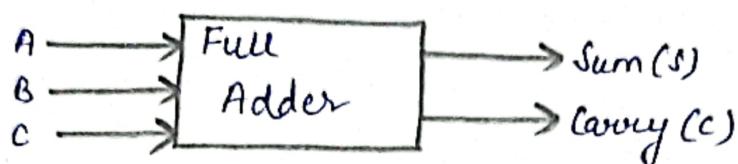


EXPERIMENT-5

Objectives :- Implementing FULL ADDER using basic logic gates.

Theory :- The half-adder is used to add only two numbers. To overcome this problem, the full adder was developed. The full-adder is used to add three 1-bit binary numbers A, B and carry c. The full-adder has three input states and two output states, i.e., sum and carry.

Block Diagram :-



Truth Table :-

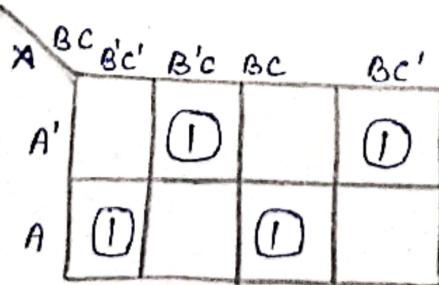
Inputs			Output	
A	B	C	Sum(S)	Carry(C)
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

Boolean Expression :-

$$\begin{aligned} \text{Sum}(S) &= A'B'C + A'B'C' + AB'C' + ABC \\ &= \Sigma(1, 2, 4, 7) \end{aligned}$$

$$\begin{aligned} \text{Carry}(C) &= A'BC + AB'C + ABC' + ABC \\ &= \Sigma(3, 5, 6, 7) \end{aligned}$$

K-Map :-



$$\text{Sum}(S) = A'B'C + A'B'C' + AB'C' + ABC$$

$A \cdot B \cdot C$	$B' \cdot C'$	$B' \cdot C$	$B \cdot C$	$B \cdot C'$
A'			1	
A	1	1	1	1

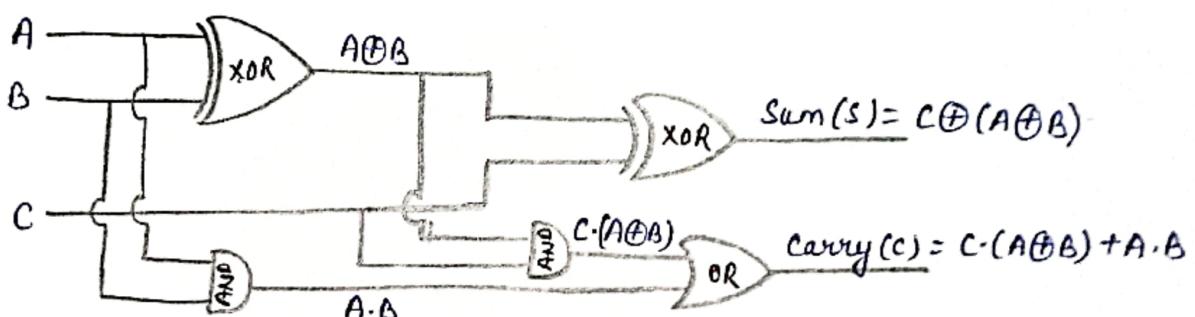
$$\text{Carry } (C) = BC + AC + AB$$

Circuit Diagram :-

$$\begin{aligned} \text{Sum}(S) &= A'B'C + A'BC' + AB'C' + ABC \\ &= C(A'B' + AB) + C'(A'B + AB') \\ &= C(\overline{A \oplus B}) + C'(A \oplus B) = C \oplus (A \oplus B) \end{aligned}$$

$$\begin{aligned} \text{Carry } (C) &= A'Bc + AB'C + ABC' + ABC \\ &= c(A'B + AB') + AB(C' + C) \\ &= c(A \oplus B) + AB \quad \{ \because (C' + C) = 1 \} \end{aligned}$$

Full-Adder can be designed with the help of two 2-inputs XOR gates, two 2-inputs AND gates and one 2-inputs OR gate.



Procedure :- Procedure to implement FULL ADDER using basic logic gates :-

- Step 1. Open logic.ly website in your browser and click on the link 'try logic.ly'.
- Step 2. A logic simulator will be opened.
- Step 3. For FULL ADDER, 3 types of gates and total number of 5 gates are used, as follows :-
 - two 2-inputs XOR gates
 - two 2-inputs AND gates
 - one 2-inputs OR gate, according to Boolean expression.

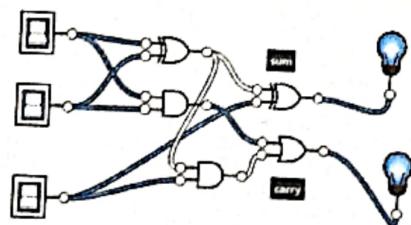
Step4. In the simulator, as we know FULL ADDER has atleast three inputs and two outputs (sum and carry).

Step5. Now, we will select desired gate from left hand-side.

Step6. Take three inputs ($A, B \Rightarrow$ input and $C \Rightarrow$ carry) from the Input control section and choose desired binary input for verification (as shown in the Truth table).

Step7. For the output, we will go under the output control section and drag the bulb on white screen.

Step8. The input will be connected to the input block and stretch it to the gate input node and the output will be connected to the output block of gate node and arrange the circuit as above.



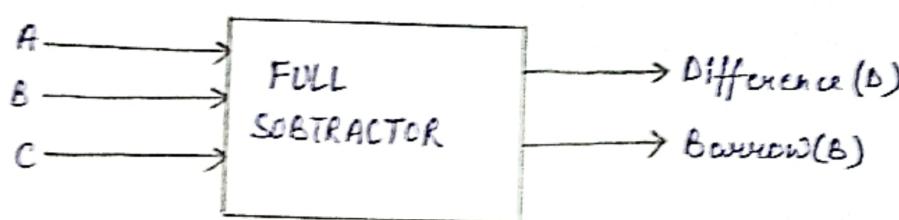
EXPERIMENT-6

(16)

Objective:- Implementing FULL SUBTRACTOR using basic logic gates.

Theory :- A full-subtractor is a combinational circuit that performs subtraction of two bits, one is minuend and other is subtrahend, taking into account borrow of the previous adjacent lower minuend bit. This circuit has three inputs and two outputs. The three inputs are minuend, subtrahend and previous borrow. The two outputs represent the difference and borrow.

Block Diagram :-



Truth Table :-

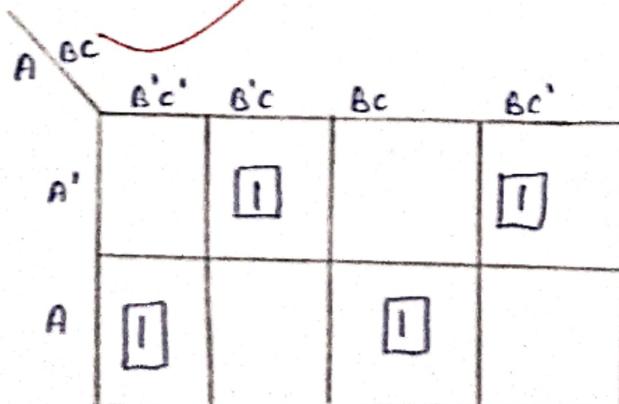
Inputs			Outputs	
A	B	C	Difference (D)	Borrow (B)
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	0	1
1	0	0	1	0
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1

Boolean Expression :-

$$\text{Difference (D)} = A'B'C + A'BC' + AB'C' + ABC$$

$$\text{Borrow (B)} = A'B'C + A'BC' + A'BC + ABC$$

K-Map :-



$$D = A'B'C + A'BC' + AB'C' + ABC$$

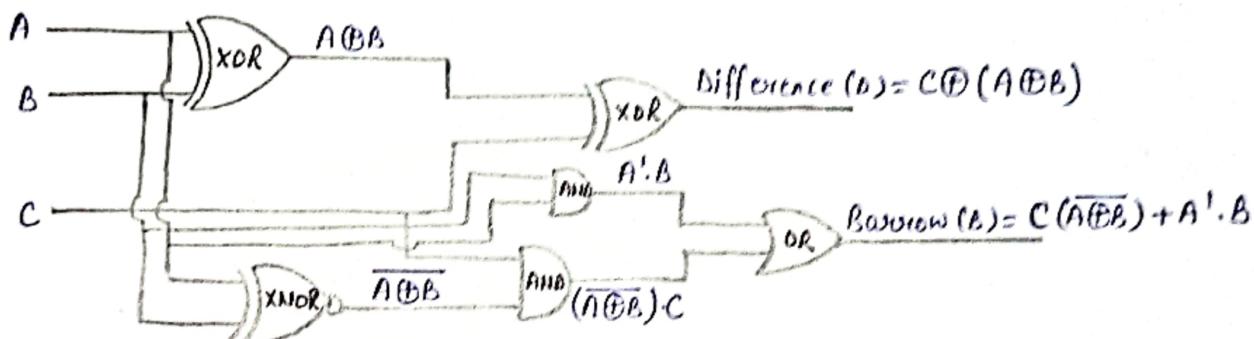
$A \cdot BC$	$B'c'$	$B'c$	Bc	Bc'
A'	1	1	1	1
A			1	

$$B = A'c + A'B + BC$$

Circuit Diagram :-

$$\begin{aligned} \text{Difference } (D) &= A'B'C + A'BC' + AB'C' + ABC \\ &= C(A'B' + AB) + C'(A'B + AB') \\ &= C(\overline{A \oplus B}) + C'(A \oplus B) \\ &= C \oplus (A \oplus B) \end{aligned}$$

$$\begin{aligned} \text{Borrow } (B) &= A'B'C + A'BC' + A'BC + ABC \\ &= C(AB + A'B') + A'B(C' + C) \\ &= C(\overline{A \oplus B}) + A'B \cdot 1 \quad \{ \because (C' + C) = 1 \} \\ &= C(\overline{A \oplus B}) + A'B \end{aligned}$$



Procedure :- Procedure to implement FULL SUBTRACTOR using basic logic gates :-

Step 1. Open logic.ly website in your browser and click on the link "try logic.ly".

Step 2. A logic simulator will be opened.

Step 3. For FULL SUBTRACTOR, 4 types of gates and total number of 6 gates are required, as follows:-

- two 2-inputs XOR gates
- one 2-inputs XNOR gate
- two 2 inputs AND gates
- one 2-inputs OR gate.

Step 4. In the simulator, as we know FULL SUBTRACTOR has atleast three inputs and two outputs (difference and borrow).

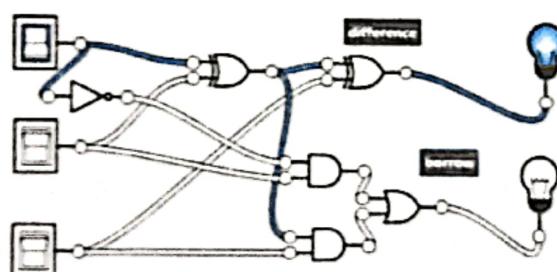
Step 5. Now, we will select desired gate from left hand-side.

Step 6. Take three inputs ($A, B \Rightarrow$ inputs, $C \Rightarrow$ borrow (previous)) from the input control section and choose desired binary input for verification (as shown in the truth table).

Step 7. For the output, we will go under the output control section and drag the bulb on white screen.

Step 8. The input will be connected to the input block and stretch it to the gate ^{input} node and the output will be connected to the output block of gate node and arrange the circuit as above.

Nice

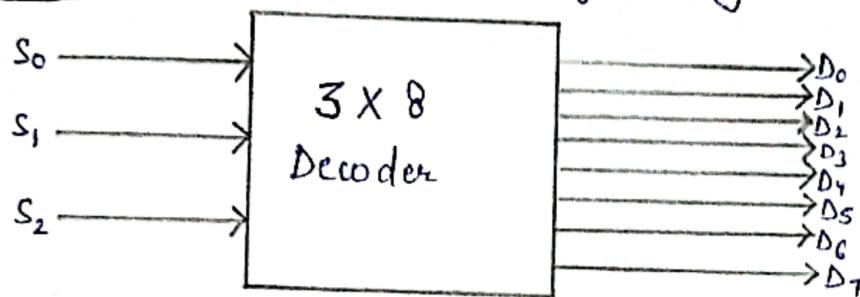


EXPERIMENT-7

Objective :- Implementing 3x8 decoder using AND Gates.

Theory :- A decoder is a combinational circuit which is used to change the code into a set of signals. It is the source of encoder. A decoder takes multiple inputs and gives multiple output. $2^n = m$, where $n \Rightarrow$ inputs $m \Rightarrow$ outputs

Block Diagram :- 3x8 Decoder using AND gates.



Truth Table :-

Enable	INPUTS				OUTPUTS							
	E	S ₂	S ₁	S ₀	D ₀	D ₁	D ₂	D ₃	D ₄	D ₅	D ₆	D ₇
0	X	X	X	X	0	0	0	0	0	0	0	0
1	0	0	0	0	1	0	0	0	0	0	0	0
1	0	0	1	0	0	1	0	0	0	0	0	0
1	0	1	0	0	0	0	1	0	0	0	0	0
1	0	1	1	0	0	0	0	1	0	0	0	0
1	1	0	0	0	0	0	0	0	1	0	0	0
1	1	0	1	0	0	0	0	0	0	0	1	0
1	1	1	0	0	0	0	0	0	0	0	0	1
1	1	1	1	0	0	0	0	0	0	0	0	1

Boolean Expression :-

$$D_0 = S_0' \cdot S_1' \cdot S_2'$$

~~$$D_1 = S_0 \cdot S_1' \cdot S_2'$$~~

~~$$D_2 = S_0' \cdot S_1 \cdot S_2'$$~~

~~$$D_3 = S_0 \cdot S_1 \cdot S_2'$$~~

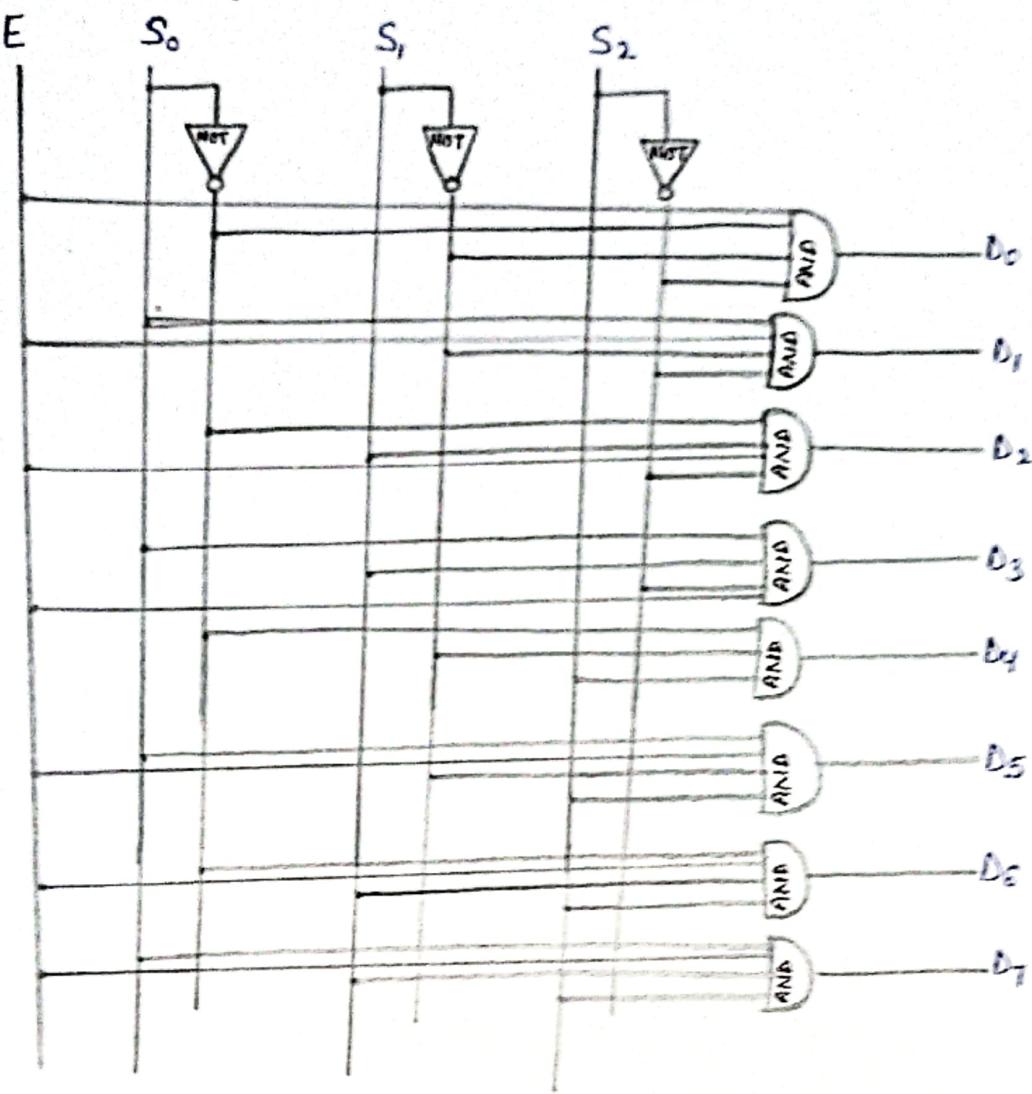
~~$$D_4 = S_0' \cdot S_1' \cdot S_2$$~~

~~$$D_5 = S_0 \cdot S_1' \cdot S_2$$~~

~~$$D_6 = S_0' \cdot S_1 \cdot S_2$$~~

~~$$D_7 = S_0 \cdot S_1 \cdot S_2$$~~

Circuit Diagram :-



Procedure :- Procedure to implement 3x8 Decoder using AND Gates :-

Step 1. Open logic.ly website in your browser and click on the link "try logic.ly".

Step 2. A logic simulator will be opened.

Step 3. For DECODER, 2 types of gates are used:-

→ three 1-input NOT gates

→ eight 3-input AND gates, according to Boolean expression.

Step 4. In the simulator, as we know DECODER (3x8) has three inputs and eight outputs.

Step 5. Now, we will select desired gate from left hand-side.

Step 6. Take Enable pin, three inputs from the input control section and choose desired binary input for verification (as shown in Truth Table).

Step 7. for the output, we will go under the output control section and drag the bulb on white screen.

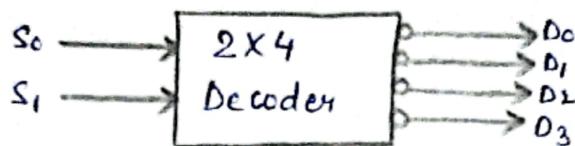
Step 8. The input will be connected to input block and stretch it to gate input node & output will be connected to output block of gate node and arrange it with

EXPERIMENT- 8

Objective :- Implementing 2x4 decoder using NAND gates.

Theory :- A decoder is a combinational circuit which is used to change the code into a set of signals. It is the source of encoder. A decoder takes multiple inputs and gives multiple outputs.

Block Diagram :- 2x4 Decoder using NAND gates.



Truth table :-

Enable Pin	INPUTS		OUTPUTS				
	E	S ₁	S ₀	D ₀	D ₁	D ₂	
1	X	X		1	1	1	1
0	0	0		0	1	1	1
0	0	1		1	0	1	1
0	1	0		1	1	0	1
0	1	1		1	1	1	0

Boolean expression :-

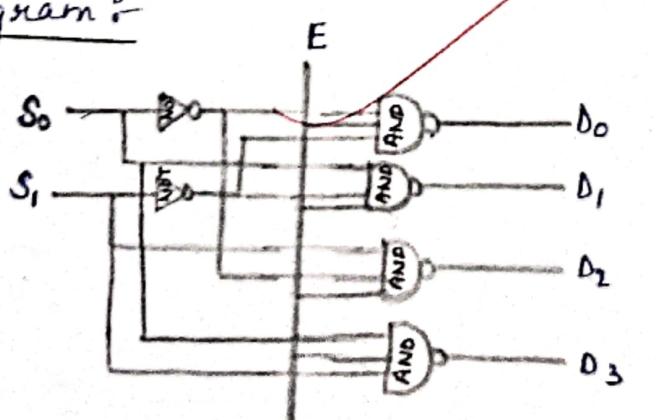
$$D_0 = S_0' \cdot S_1'$$

$$D_1 = S_1' \cdot S_0$$

$$D_2 = S_1 \cdot S_0'$$

$$D_3 = S_0 \cdot S_1$$

Circuit Diagram :-



Procedure :- Procedure to Implement 2x4 decoder using NAND Gates :-

Step 1. Open logic.ly website in your browser and click on the link "try logic.ly".

Step 2. A logic simulator will be opened.

Step 3. For DECODER, 2 types of gates are used :-

- 2 one-input NOT gate

- 4 two-input and enable pin NAND gates, according to Boolean Expression.

Step 4. In the simulator, as we know DECODER(2x4) has two inputs and four outputs.

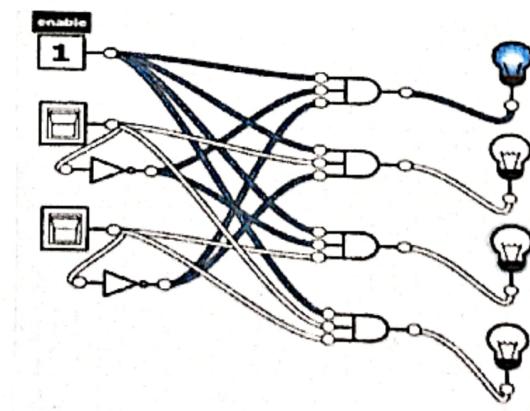
Step 5. Now, we will select desired gate from left hand-side.

Step 6. Take Enable pin, two inputs (toggle switch can be used) from the input control section and choose desired binary input for verification (as shown in truth-table).

Step 7. For the output, we will go under the output control section and drag the bulb on the screen.

Step 8. The input will be connected to input block and stretch it to gate input node and output will be connected to output block of gate node and arrange circuit.

Step 8



EXPERIMENT-9

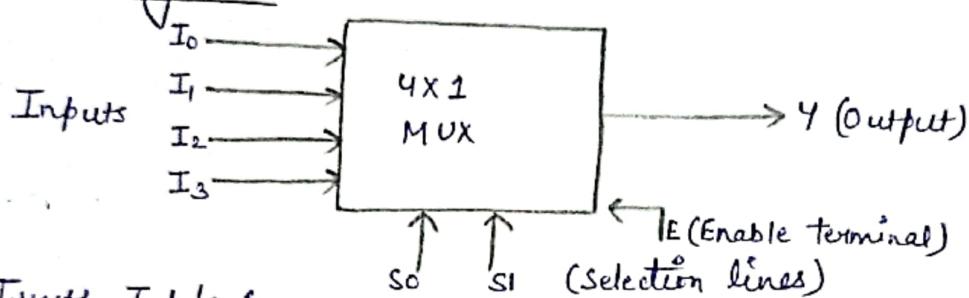
Objective :- Design 4x1 Multiplexer with AND and OR Gate.

Theory :- It is a combinational circuit which have many inputs or data inputs and single output depending on control or select inputs.

Multiplexers are also known as data selector.

For N input lines, 2^n input lines and n selection lines are required and there is only one output line.

Block Diagram :-



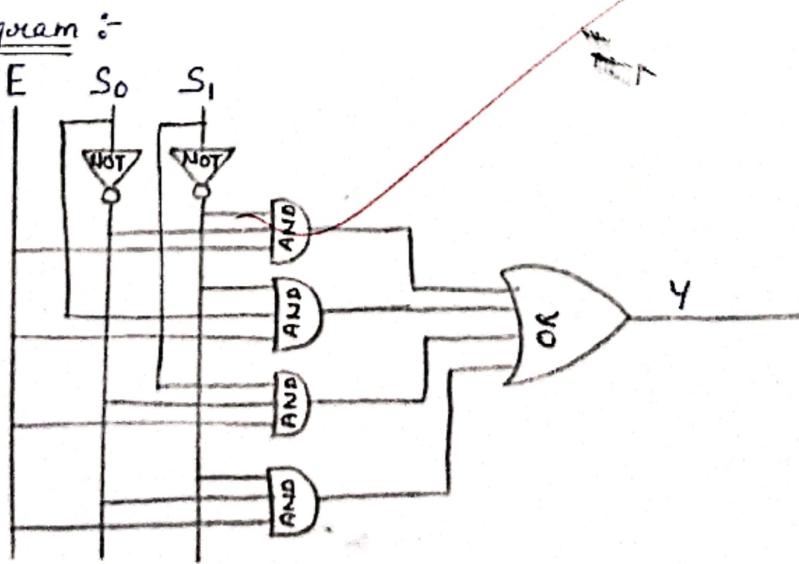
Truth Table :-

Inputs			Output
E	S_0	S_1	Y
0	X	X	0
1	0	0	I_0
1	0	1	I_1
1	1	0	I_2
1	1	1	I_3

Boolean Expression :-

$$Y = E \cdot [S_1' S_0' \cdot I_0 + S_1' \cdot S_0 \cdot I_1 + S_1 \cdot S_0' \cdot I_2 + S_1 \cdot S_0 \cdot I_3]$$

Circuit Diagram :-



Procedure :- Procedure to implement 4x1 multiplexer using AND and OR gates.

Step 1. Open logic.ly website in your browser and click on the link "try logic.ly".

Step 2. A logic simulator will be opened.

Step 3. For MULTIPLEXER, 3 types of gates are used :-
 → two one-input NOT gate.
 → four three-input AND gate
 → one four-input OR gate.

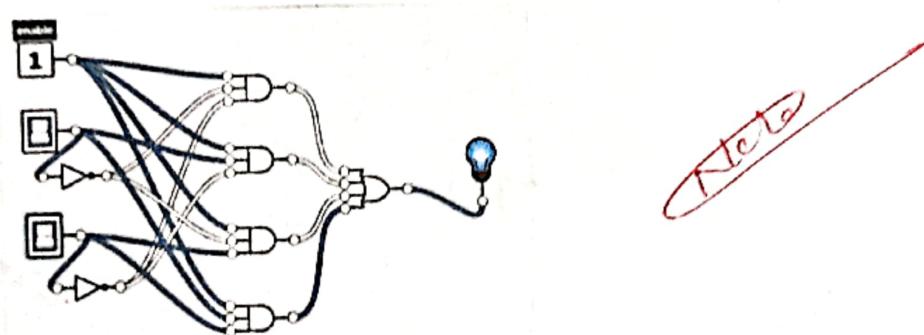
Step 4. In the simulator, as we know Multiplexer has four inputs and one output.

Step 5. Now, we will select desired gate from left-hand side.

Step 6. Take enable pin, Inputs from the input control section and choose desired binary input for verification (as shown in truth-table).

Step 7. For the output, we will go under the output control section and drag the bulb on the screen.

Step 8. The input will be connected to input block and stretch it to gate input node and output will be connected to output block of gate node and arrange circuit.



EXPERIMENT-10

Objective :- Design a S-R Latch using NAND gates / NOR gates.

Theory :- S-R flip-flop stands for SET-RESET flip-flop, also called S-R latch. The circuit has two inputs, S(SET) and R(RESET), and two outputs Q and \bar{Q} , and consists of two NOR and two NAND gates.

Control input determines when the state of the circuit is to be changed.

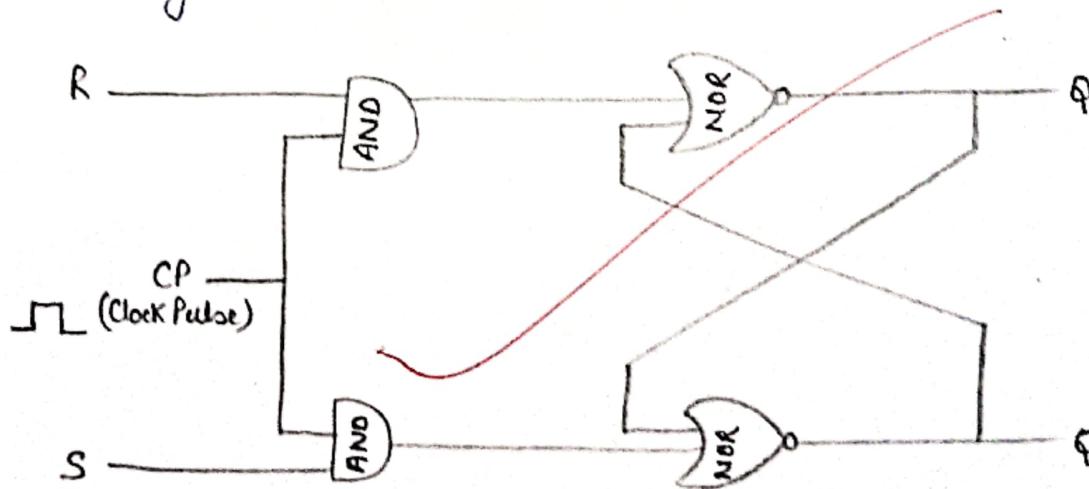
The limitation with a S-R flip-flop using NOR and NAND gates is the invalid state. A clock pulse is given to the inputs of the AND gate. If the value of the clock pulse is '0', the outputs of both the AND gates remain '0'.

Clocked S-R Latch :-

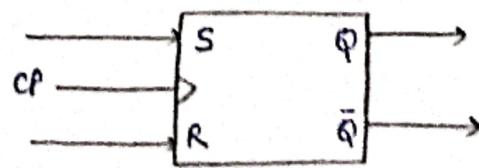
→ Characteristic table :-

Current Inputs		Current State (Q_n)	Next State (Q_{n+1})
S	R	0	0 } no change condition
0	0	1	1
0	0	0	0
0	1	1	0
0	1	0	0
1	0	0	1
1	0	1	1
1	1	0	- } Intermediate state
1	1	1	-

→ Circuit diagram :-



→ Block diagram :-



→ Truth Table :-

S	R	Q_{n+1}
0	0	Q_n
0	1	0
1	0	1
1	1	-

Procedure :- Procedure to implement S-R flip-flop using NOR gates :-

Step 1: open logic.ly website in your browser and click on the link "try logic.ly".

Step 2: A logic simulator will be opened.

Step 3: For S-R flip-flop, following devices are used :-
 → 2 AND gates (two-inputs)
 → 2 NOR gates (two-inputs)
 → Clock-Pulse

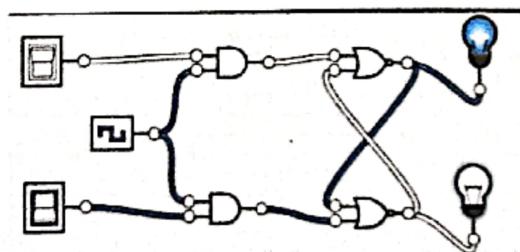
Step 4: In the simulator, we know S-R flip-flop has two inputs and two outputs.

Step 5: Now, we will select desired gate from left hand side.

Take clock-pulse, inputs from the input control section and choose desired binary input for verification (as shown in characteristic table).

Step 6: For the output, we will go under the output control section and drag the bulb on the screen.

Step 7: The input will be connected to input block and stretch it to gate input node and output of it will be connected as feedback to another input gate with another input node and vice-versa with another input.



EXPERIMENT-11

Objective :- Design a D flip-flop.

Theory :- One problem with S-R flip-flop is that the condition $R=1, S=1$ and vice-versa must be avoided. One way to do this is to allow just a single input. The D flip-flop accomplishes this.

By using an inverter, the nonclock inputs to the two AND gates are guaranteed to be the opposite of each other. The output of the D flip-flop is always equal to the most recent value applied to the input.

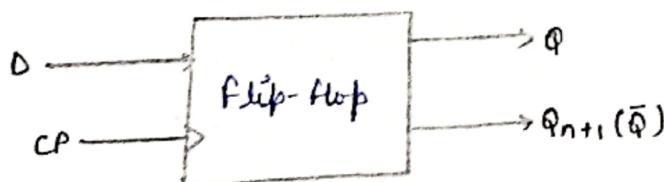
Truth Table :-

D	Q_{n+1}
0	0
1	1

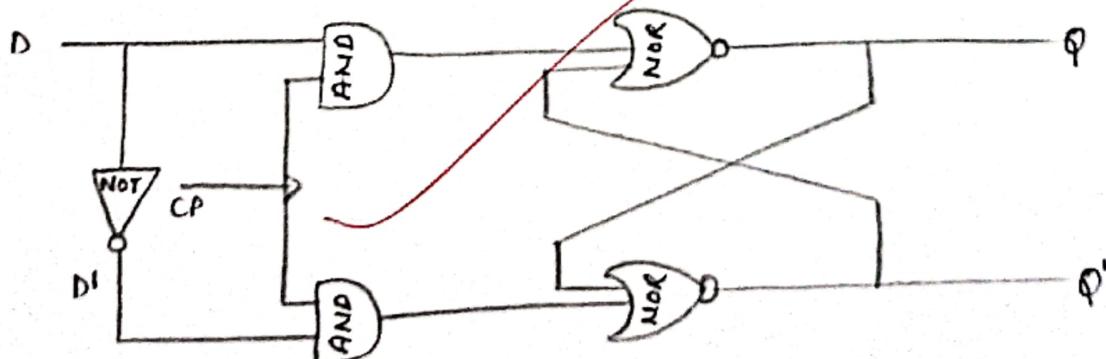
Characteristic Table :-

D	Q_n	Q_{n+1}
0	0	0
1	0	1
0	1	0
1	1	1

Block diagram :-

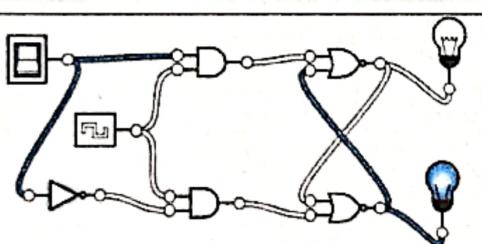


Circuit Diagram :-



Procedure:- Procedure to implement D flip-flop :-

- Step 1. Open logic.ly website in your browser and click on the link "try logic.ly".
- Step 2. A logic simulator will be opened.
- Step 3. For D flip-flop, following devices are used :-
 - 2 two-inputs AND gates
 - 2 two-inputs NOR gates
 - 1 one-input NOT gate
 - clock-pulse.
- Step 4. In the simulator, we know D flip-flop has one input and two outputs.
- Step 5. Now, we will select desired gate from left hand side. Take clock-pulse, inputs from the input control section and choose desired binary input for verification (as shown in characteristic table).
- Step 6. For the output, we will go under the output control section and drag the bulb on the screen.
- Step 7. The input will be connected to input block and stretch it to gate input node and output of it will be connected as feedback to another input gate (inverter of D) with another input node and vice-versa with another input, i.e., inverter of D (as shown in circuit diagram).



Note

EXPERIMENT-12

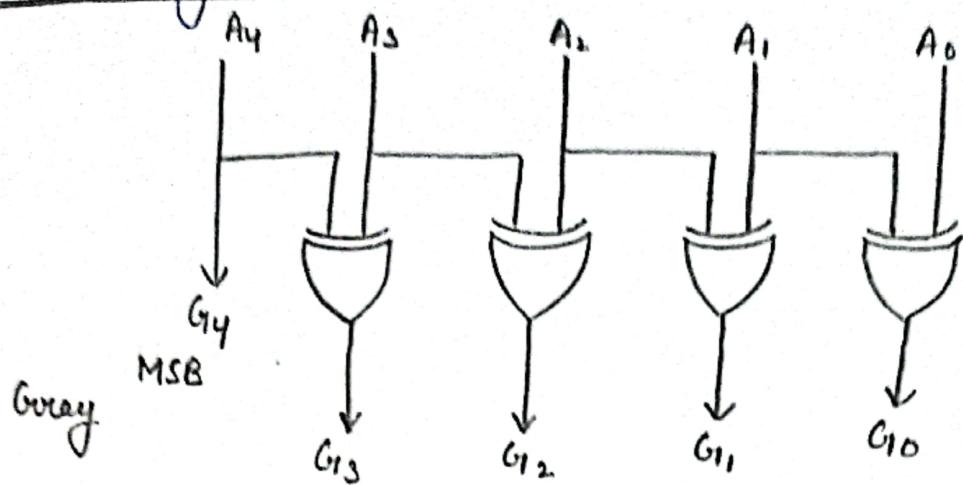
Objectiu:- Design a circuit diagram to convert a 5-bit binary code into Gray code.

Theory:- The Binary to Gray Code converter is a logical circuit that is used to convert the binary code into its equivalent Gray code. By putting the MSB of below the axis and the NIB of 1 above the axis and reflecting the $(n-1)$ bit code about an axis after 2ⁿ⁻¹ steps, we can obtain the n-bit gray code.

Truth Table:-

Decimal Number	5-bit Binary code $b_4 b_3 b_2 b_1 b_0$	5-bit Gray code $g_4 g_3 g_2 g_1 g_0$
0	00000	00000
1	00001	00001
2	00010	00011
3	00011	00010
4	00100	00110
5	00101	00111
6	00110	00101
7	00111	00100
8	01000	01101
9	01001	01111
10	01010	01110
11	01011	01010
12	01100	01011
13	01101	01001
14	01110	01000
15	01111	10000
16	10000	11000
17	10001	11001
18	10010	11011
19	10011	11010
20	10100	11110
21	10101	11111
22	10110	11101
23	10111	11100
24	11000	10100
25	11001	10101
26	11010	10111
27	11011	10110
28	11100	10010
29	11101	10011
30	11110	10001
31	11111	10000

Circuit Diagram :-



Procedure :- Steps to make logic gate of Binary to Gray Scale :-

- Step 1. Open logic.ly website in your browser and click on the link "gray logic.ly".
- Step 2. A logic simulator will be opened.
- Step 3. For making a logic diagram of binary to gray code 5-bit switch and 5-bit display and 4 xor-gate.
- Step 4. From the input section we take 5-binary input and four xor gate and then drag it into white screen.
- Step 5. Now, from the output section take 5 bulb and place it into the screen.
- Step 6. The input connected to input block and output to output block in the manner shown in figure.
- Step 7. for the desired input connected we will get the desired output.

Note

EXPERIMENT-13

Objective :- Design a circuit diagram to convert a 5-bit Gray code into Binary code.

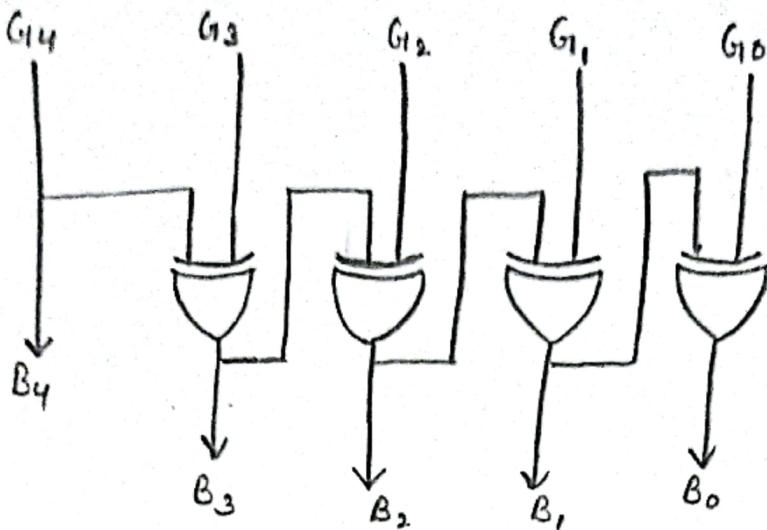
Theory :- Gray code system is a binary system number in which the normal sequence of binary numbers generated by the hardware differs in only one bit. It is used in applications in which the normal sequence of binary nos. generated by the hardware may produce an error or ambiguity during the transition from one number to the next.

Let, b_0, b_1, b_2, b_3 and b_4 be the bits representing the binary numbers, where b_0 is the LSB and b_4 is the MSB, and let, g_0, g_1, g_2, g_3 and g_4 be the bits representing the gray code of the binary numbers, where g_0 is the LSB and g_4 is the MSB.

Truth Table :-

Decimal Number	5-bit Gray code	6-bit Binary code
0	0 0 0 0 0	0 0 0 0 0 0
1	0 0 0 0 1	0 0 0 0 0 1
2	0 0 0 1 1	0 0 0 0 1 0
3	0 0 0 1 0	0 0 0 1 0 0
4	0 0 1 1 0	0 0 1 1 0 0
5	0 0 1 1 1	0 0 1 1 0 1
6	0 0 1 0 1	0 0 1 1 1 0
7	0 0 1 0 0	0 0 1 1 1 1
8	0 1 1 0 0	0 0 0 0 1 1
9	0 1 1 0 1	0 0 0 0 1 0
10	0 1 1 1 0	0 0 0 1 1 1
11	0 1 1 1 1	0 0 0 1 1 0
12	0 1 0 1 1	0 0 1 0 1 1
13	0 1 0 1 0	0 0 1 0 1 0
14	0 1 0 0 0	0 0 1 0 0 1
15	1 1 0 0 0	0 0 1 0 0 0
16	1 1 0 0 1	0 1 0 0 1 1
17	1 1 0 1 1	0 1 0 0 1 0
18	1 1 0 1 0	0 1 0 1 1 1
19	1 1 1 1 0	0 1 0 1 1 0
20	1 1 1 1 1	0 1 0 1 0 1
21	1 1 1 0 1	0 1 0 1 0 0
22	1 1 1 0 0	0 1 1 0 1 1
23	1 1 0 0 0	0 1 1 0 1 0
24	0 1 0 0 0	0 1 1 0 0 1
25	0 1 0 0 1	0 1 1 0 0 0
26	0 1 1 1 1	0 1 1 1 1 0
27	0 1 1 1 0	0 1 1 1 1 1
28	0 1 0 1 0	0 1 1 1 0 1
29	0 1 0 1 0	0 1 1 1 0 0
30	0 0 0 0 1	0 1 1 1 1 1
31	0 0 0 0 0	0 1 1 1 1 0

Circuit Diagram :-



Procedure :- Steps to make circuit diagram of Gray code to Binary code :-

- Step 1. Open logic.ly website in your browser and click on the link "try logic.ly".
- Step 2. A logic simulator will be opened.
- Step 3. for making a logic diagram of Gray code to Binary 5-bit switch and 5-bit display and 4-XOR-gate.
- Step 4. From the input section, we take 5-binary input and four XOR gate and then drag it into white screen.
- Step 5. Now, from the output section, take 5-bulb and place it into the screen.
- Step 6. The input connection and output connection in the manner shown in figure.
- Step 7. for the desired input connected we will get the desired output.

Notes