

Experiment no: 12

Experiment title: Universal Shift Register

Theory: A register capable of shifting both right and left is called a bidirectional shift register or universal shift register.

Also it has the capabilities of parallel-load.

An universal shift register may contain following characteristics:

- ① A clear control to clear register
- ② CP input for clock pulses to synchronize all operations
- ③ A shift-right control to enable the shift-right operation and the serial input and output lines associated with the shift-right.
- ④ A shift-left control to enable the shift-left operation and the serial input and output lines associated with the shift-left.

- ⑤ A parallel-load control to enable a parallel transfer and the n input lines associated with the parallel transfer
- ⑥ n parallel output lines
- ⑦ A control state that leaves the info. in the register unchanged even though clock pulses are continuously applied

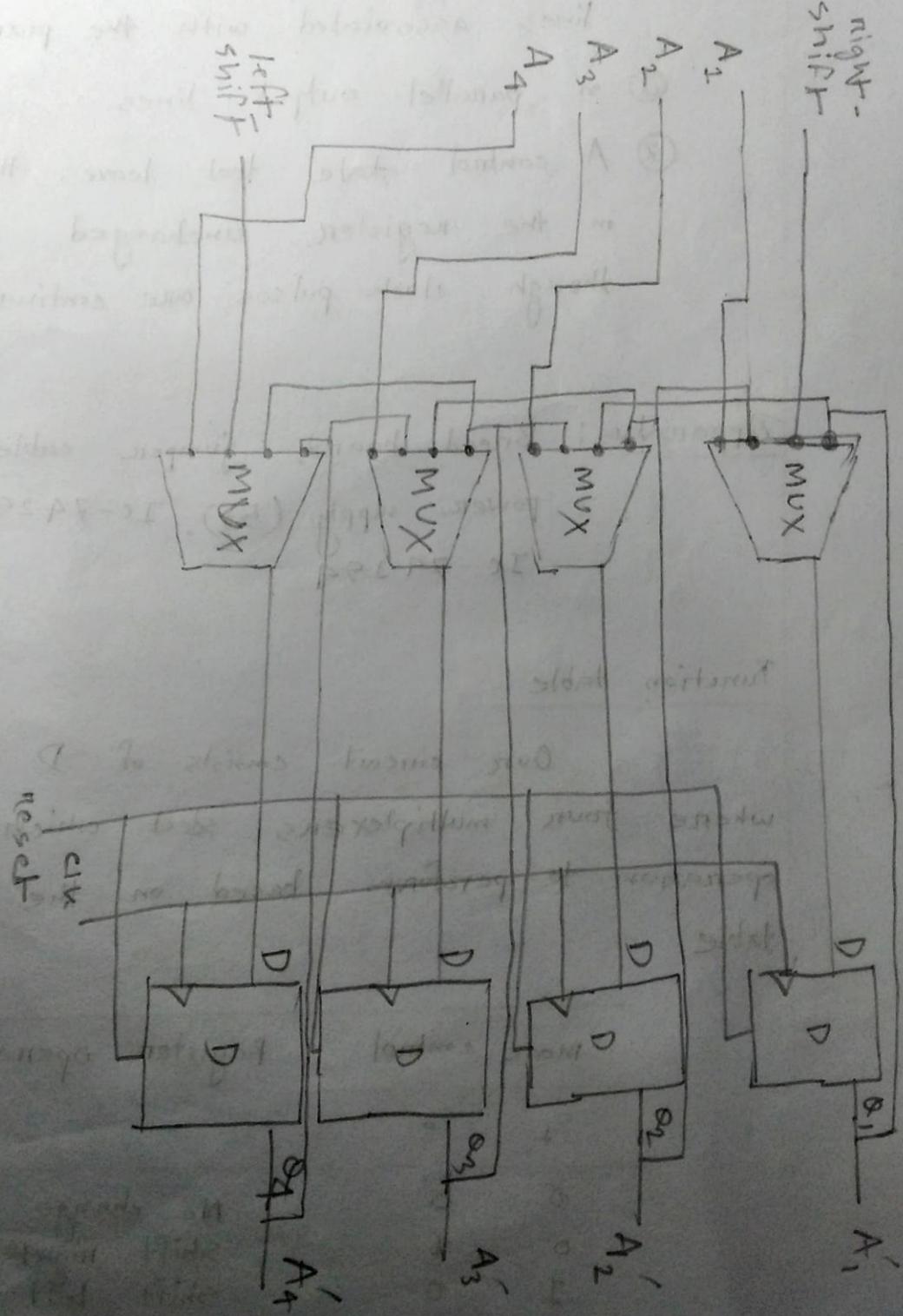
Apparatus: Bread board, jumper cables, power supply (5V), IC-74299

Function table:

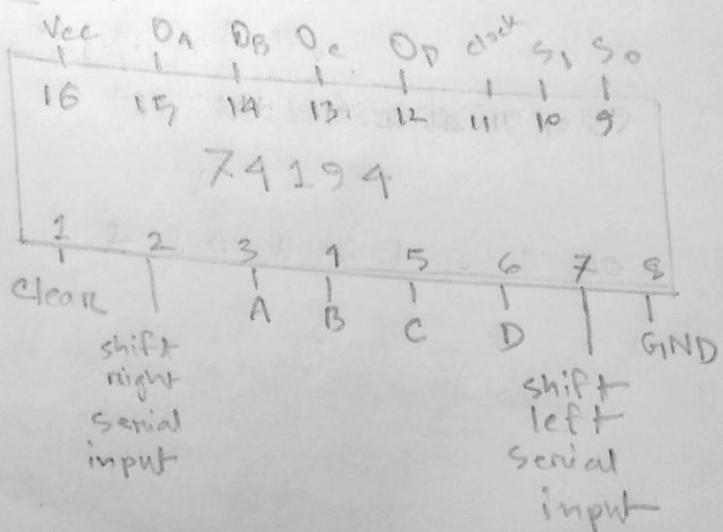
Our circuit consists of D flip-flops where four multiplexers select which operation to perform based on the following table

Mode control		Register operation
s_1	s_0	
0	0	No change
0	1	Shift right
1	0	Shift left
1	1	Parallel load

Logic diagram:

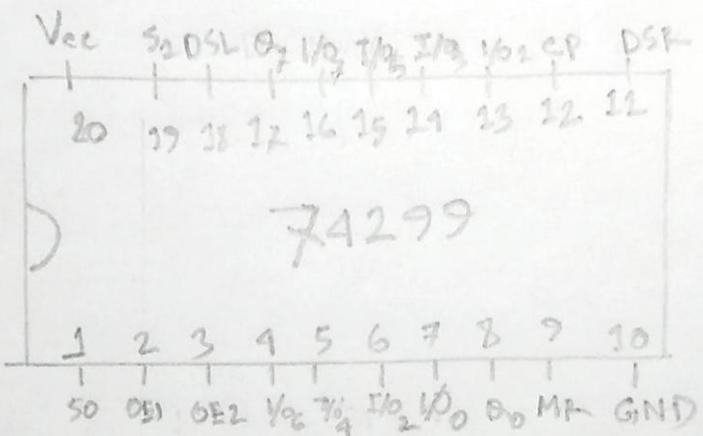


Pin Diagram(74194)



Hence this circuit works like our logic diagram. Hence we have the ability to load input and shift bi-direction (both left or right). Also we can achieve output, which can be controlled by the selection line,

Pin diagram:



Hence,

S_{0,1} = mode select input

OE 1/2 = output enable (active low)

I/O_{1,2,3,4,5,6,7} = parallel data input/output

Q_{1,2} = serial output

MR = asynchronous master reset (active low)

GND = Ground

CP = clock input (low-to-high, edge triggered)

DSR = serial data shift-right input

DSL = serial data shift-left input

V_{cc} = supply voltage

Output: Hence after constructing and implementing the circuit we can conclude that it works like the function table as we specified earlier.

Md. Sharafat Karzim

ID: 2102024

Reg: 10151

Ex: 15
Sequence
Detector

Experiment no: 15

Experiment Title: Sequence Detector

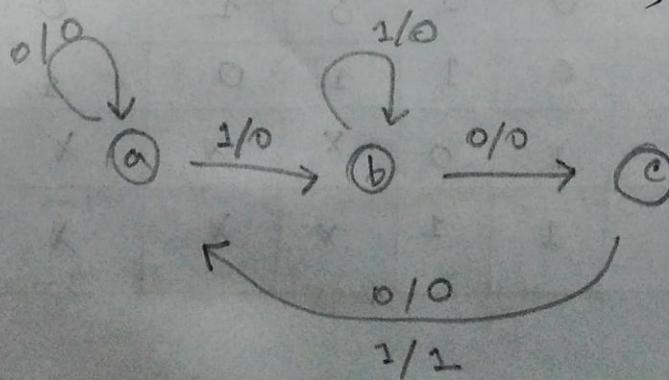
Theory: A sequence detector is a sequential state machine that takes an input string of bits and generates an output 1 whenever the target sequence has been detected. In a mealy machine, output depends on the present state and the external input.

Let's consider a non-overlapping sequence 101. Here's a sample test case,

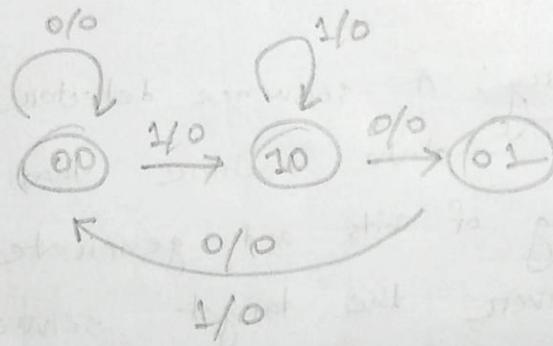
Input : 0110101011001

Output : 0000100010000

Let's draw a state table first,



Now let's assign some binary values for a , b and c . Now after assignment our state diagram will be,



state table: With consideration of D flip flop we can draw a state table like,

Present state		i/p	Next states		Flip flop excitation		O/P
X	Y		X'	Y'	D _x	D _y	
0	0	0	0	0	0	0	0
0	0	1	1	0	1	0	0
0	1	0	0	0	0	0	0
0	1	1	0	0	0	0	1
1	0	0	0	1	0	1	0
1	0	1	1	0	1	0	0
1	1	0	X	X	X	X	X
1	1	1	X	X	X	X	X

Now for D_x , D_y and output (Z) by taking K-map approach to simplify our equations we get,

xy	00	01	11	10
0	0	0	X	0
1	1	0	X	1

$$D_x = \bar{Y} \cdot I$$

xy	00	01	11	10
0	0	0	X	1
1	0	0	X	0

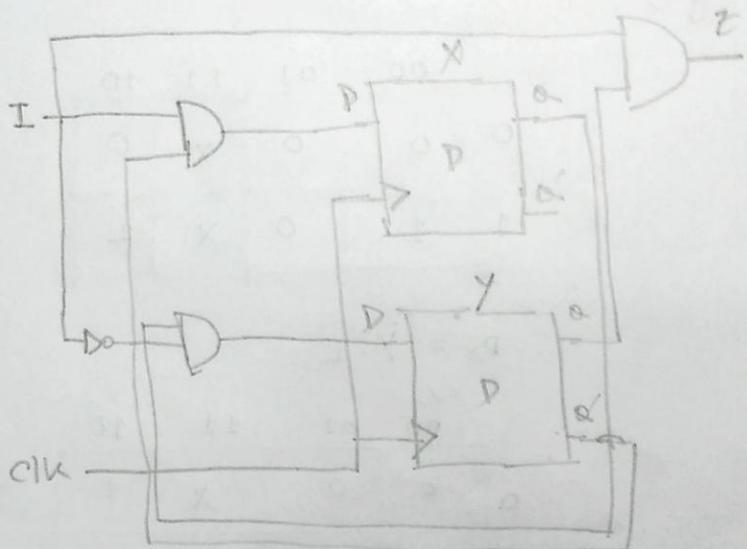
$$D_y = X \cdot \bar{I}$$

xy	00	01	11	10
0	0	0	X	0
1	0	1	X	0

$$Z = Y \cdot I$$

Now with these equations we can implement the logic diagram.

Circuit Design:



Output: In this sequential circuit, in the simulator we change output I and apply clock pulse. Whenever we found 101 in the input consecutively, then our output will be 1. For other cases our output will be 0.

Experiment 6:

Ex: 6 - Full Subtractor

Experiment title: Design and implementation of a full subtractor using logic gate.

Theory: A full subtractor is a combinational circuit that performs subtraction of three binary bits.

The three input denoted by minuend (A), subtrahend(B) previous borrow (C) and output different (D) borrow (B).

Apparatus:

- (i) IC 7486 (XOR gate)
- (ii) IC 7408 (AND gate)
- (iii) IC 7432 (OR gate)
- (iv) IC 7404 (NOT gate)
- (v) Breadboard
- (vi) connecting wires
- (vii) power supply
- (viii) Led

Truth table:

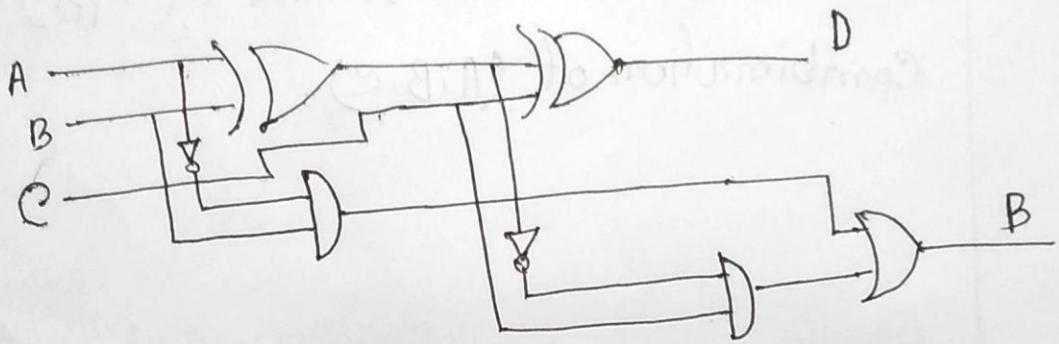
inputs			outputs	
A	B	C	D	B_1
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	0	1
1	0	0	1	0
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1

The boolean function:

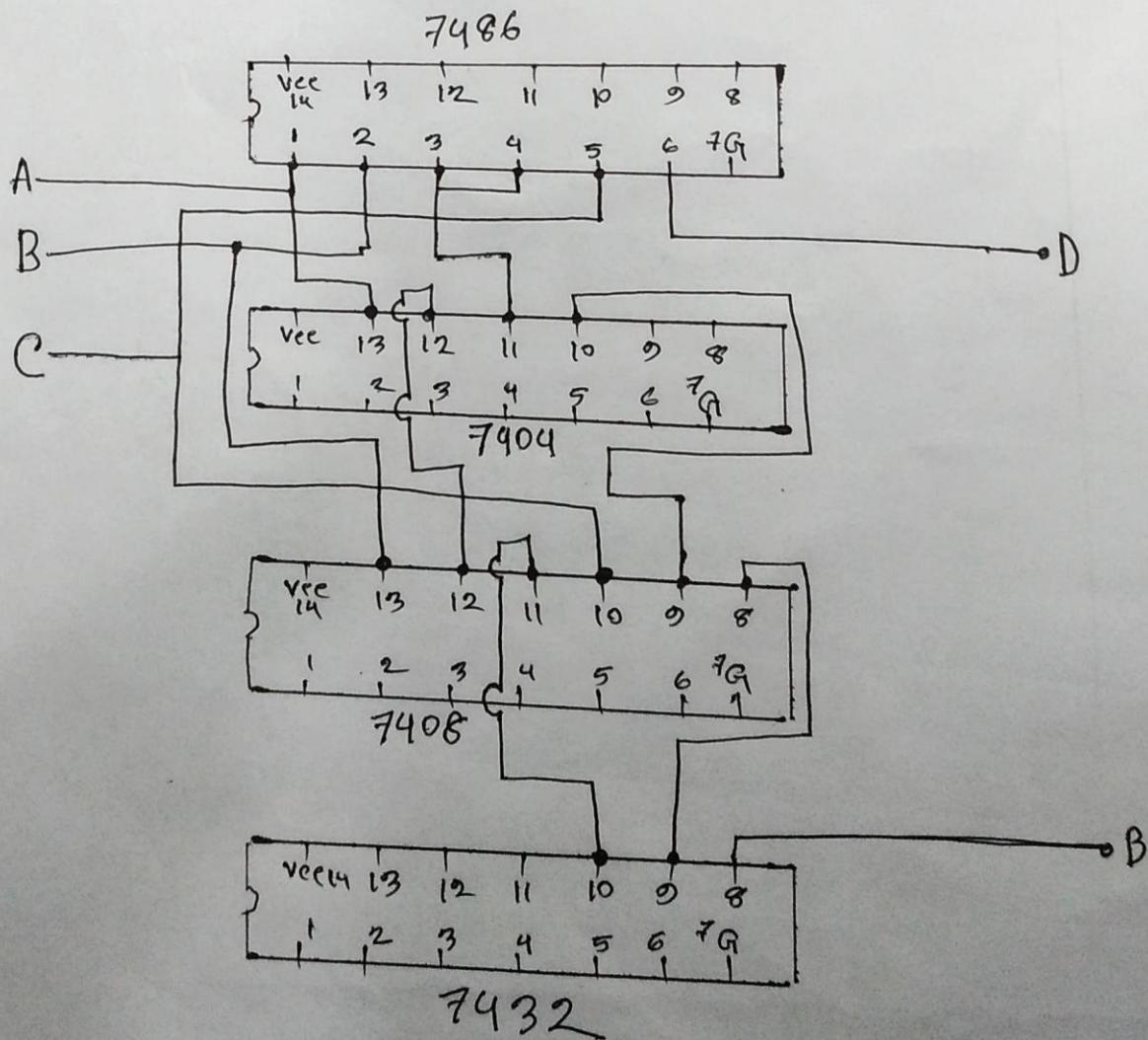
$$\begin{aligned}
 \therefore D &= A'B'C + A'BC' + AB'C' + ABC \\
 &= C(AB + A'B') + C'(A'B + AB') \\
 &= C(A \oplus B)' + C'(A \oplus B) \\
 &= A \oplus B \oplus C
 \end{aligned}$$

$$\begin{aligned}
 \therefore B_1 &= A'B'C + A'BC' + A'BC + ABC \\
 &= C(A'B' + AB) + A'B(C' + C) \\
 &= C(A \oplus B)' + A'B
 \end{aligned}$$

logic circuit:

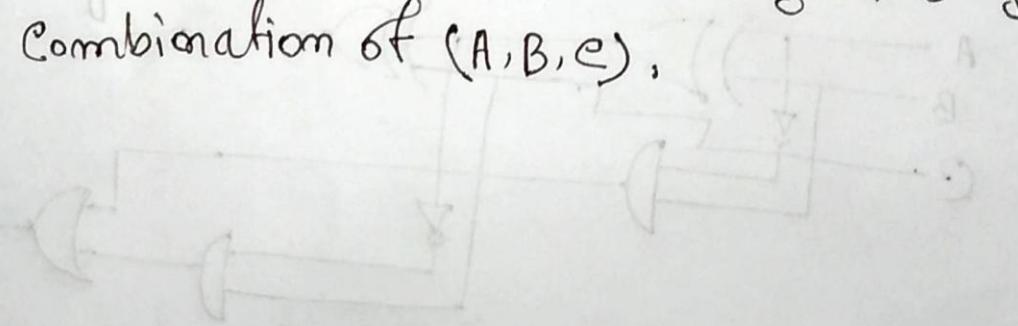


Pin diagram:



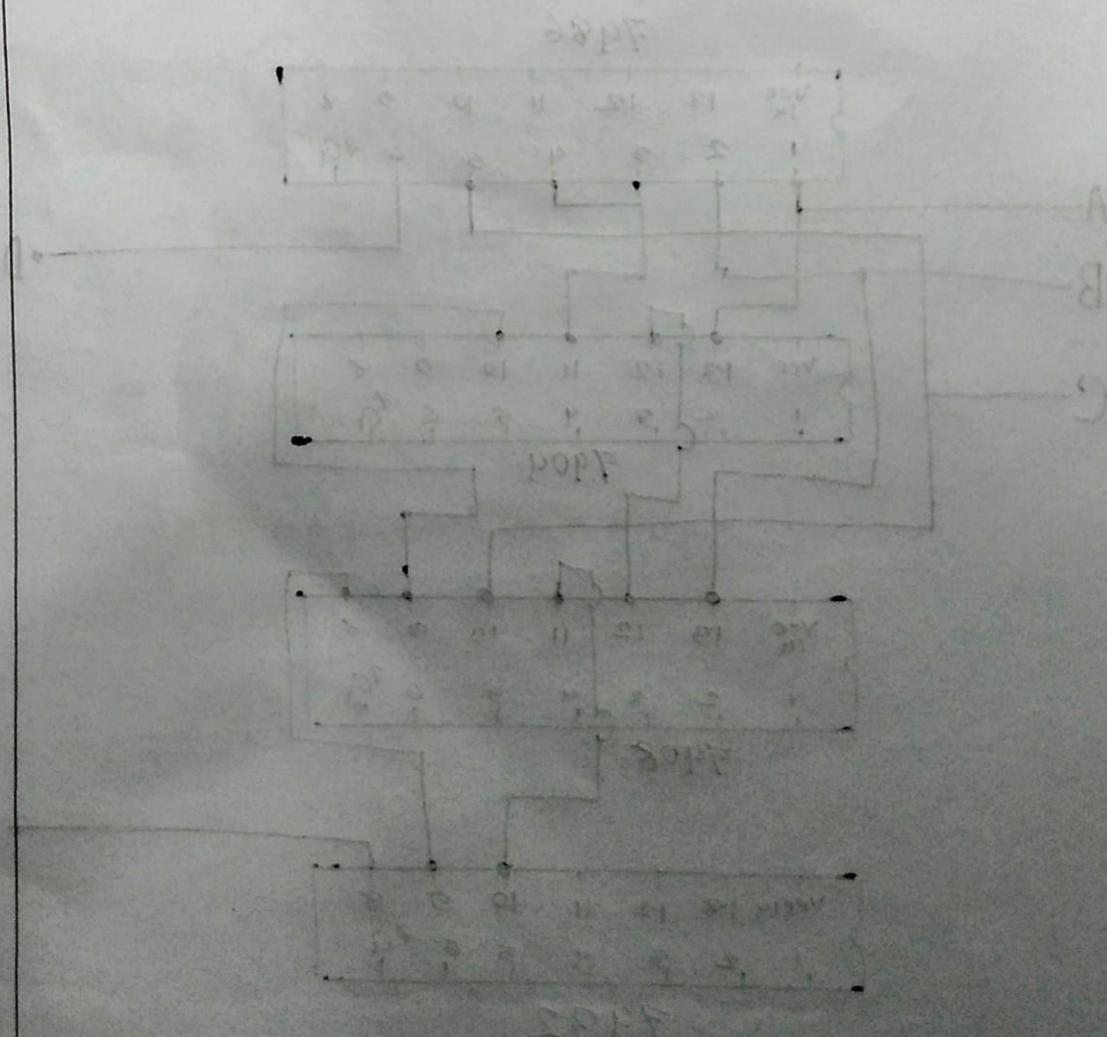
output:

The output can be verified by applying different combination of (A,B,C).



Timing signal

corresponding output



Ex: 10 - Asynchronous Counter

Experiment- 10

Experiment title:

Counters.

Theory:

A counter is a sequential logic circuit that progresses through a predefined sequence of states upon the application of input pulses. In an asynchronous counter only the first flip-flop receives the external clock signal. Each subsequent flip-flop is triggered by the output of the previous flip-flop.

Apparatus:

Breadboard, Connecting wires, Power supply,
LED's, IC-7493.

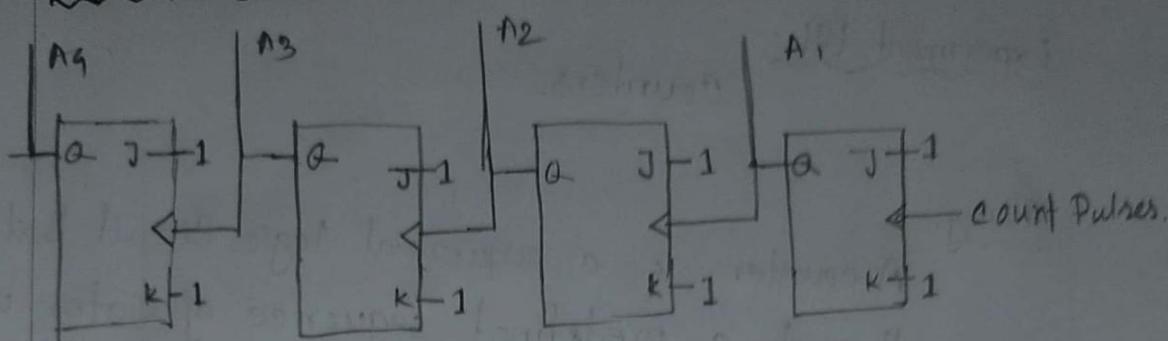
Truth Table: Count Sequence table

Current Sequence

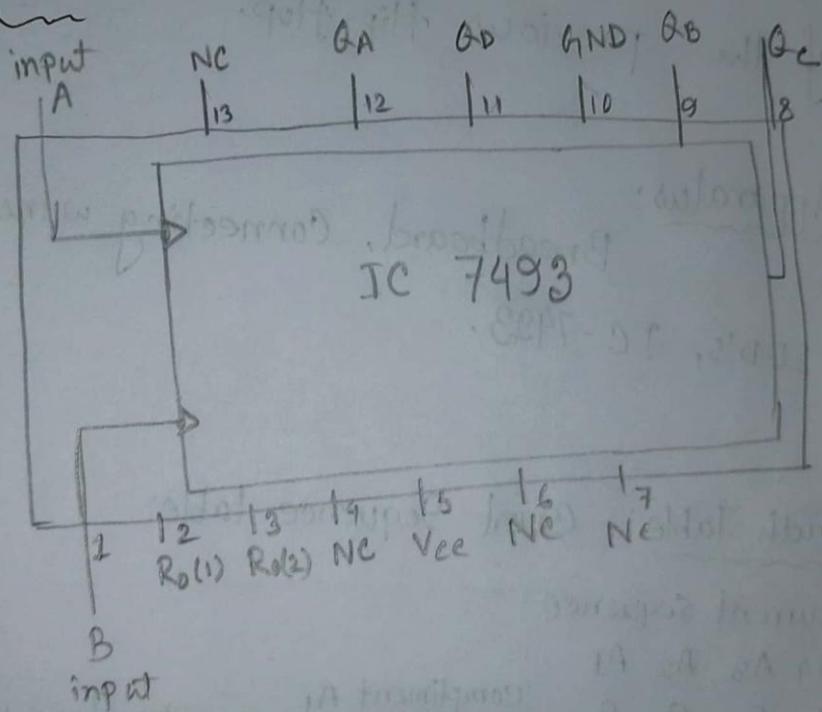
A₄ A₃ A₂ A₁

0 0 0 0	Compliment A ₁ , compliment A ₁ , A ₁ will go from 1-0 and compliment A ₂
0 0 0 1	Compliment A ₁ ,
0 0 1 0	compliment A ₁ , A ₁ will go from 1-0 and compliment A ₂
0 0 1 1	compliment A ₁ , A ₁ will go from 1-0 and compliment A ₂
0 1 0 0	compliment A ₁ , compliment A ₁ , A ₁ will go from 1-0 and compliment A ₂
0 1 0 1	and so.. on..

Logic Diagram:



Pin diagram:



Pin Number	Pin Name	Description
4, 6, 7, 13	NC	No connection
8, 9, 11, 12	Q _a , Q _d , Q _b , Q _c	Output pins
10	G _{ND}	Ground of the system
14	C _{KA}	Clock input
1	C _{LKB}	Clock input
2, 3	R	Reset

Output:

In this experiment, a 4-bit asynchronous ripple counter was constructed using J-K flip flops configured in toggle mode. The counter was tested with a clock signal. At the first pulse, the output became 0001 then 0010, 0011 and so...on.

Ex: 10 - Synchronous Counter

Experiment:-10

Experiment title: Counters.

Theory: A counter is a sequential logic circuit that progresses through a predefined sequence of states upon the application of input pulses. A synchronous counter is a digital circuit that counts binary numbers in a predefined sequence, where all the flipflops are triggered simultaneously by a common clock signal.

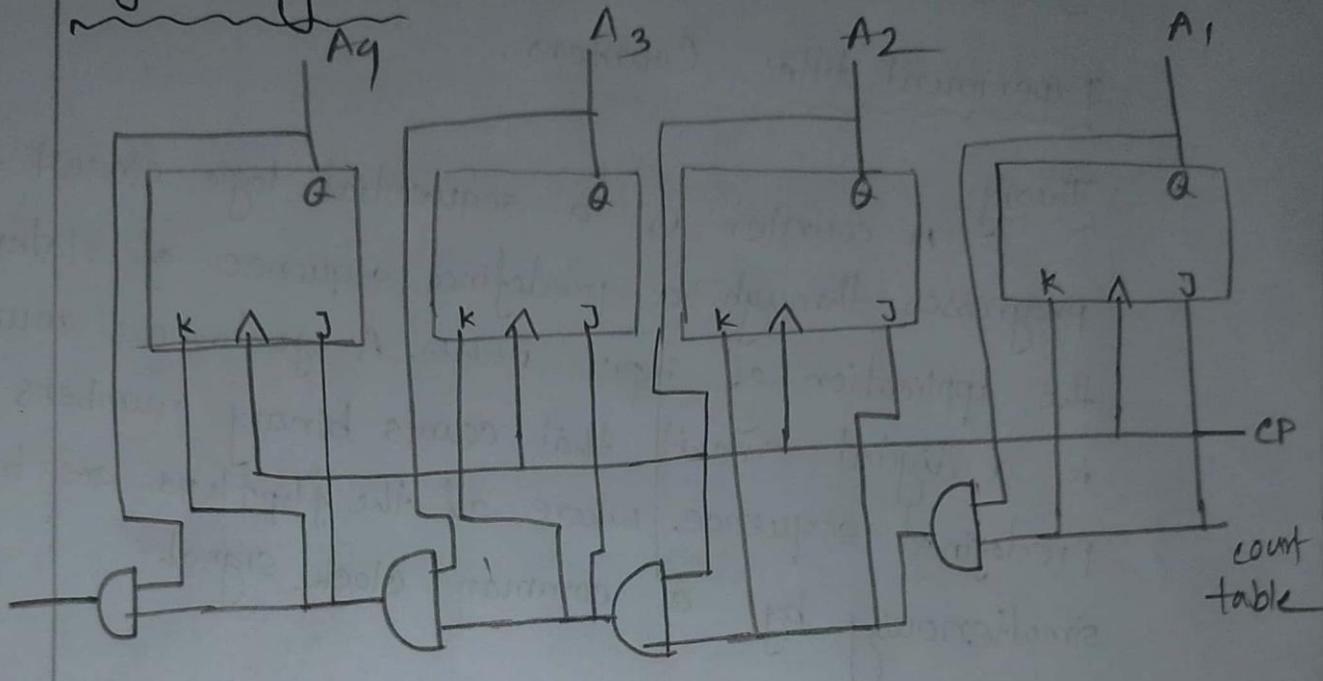
Apparatus:

Breadboard, connecting wires, power supply,
LED's, IC - 74163.

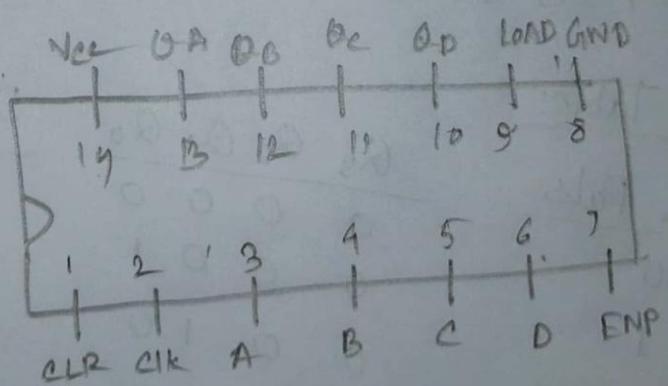
Truth Table:

<u>Clock Pulse</u>	<u>Q_3</u>	<u>Q_2</u>	<u>Q_1</u>	<u>Q_0</u>	<u>Decimal</u>
0	0	0	0	0	0
	0	0	0	1	1
1	0	0	1	0	2
2	0	0	1	1	3
3	---	---	---	---	---
...					
15	1	1	1	1	15
16	0	0	0	0	0 (Reset)

Logic Diagram:



Pin Diagram:



Output:

The output of the synchronous counter experiment demonstrate a precise and simultaneous transition of all flip-flops with each clock pulse, confirming the theoretical operation of a 4-bit synchronous up counter. The circuit successfully counted from 0000 to 1111 in binary, with each flipflop (Q_0 to Q_3) toggling exactly when expected base on the AND gate logic.

Ex: 8 - Master-Slave JK

Experiment title → Study of Master-Slave JK Flip-flop (212)

↳ Theory: The Master-slave JK flip-flop is an improved version of the RS flip-flop designed to eliminate the forbidden state problem. It consists of two RS flip-flops connected in series forming a Master stage and a slave stage, controlled by opposite clock edges. The JK flip-flop has three main inputs: J(SET), K(PRESET) and CLK(Clock). When $J=1$ and $K=0$ the output Q becomes 1 on the clock edge. When $J=0$ and $K=1$ the output Q becomes 0. If $J=K=1$ the output toggles with each clock pulse. The master-slave configuration ensures stable operation by separating input sampling and output updating preventing race conditions.

Apparatus →

- i) IC 74LS73.
- ii) Breadboard and connecting wires.
- iii) Power supply (5V DC).
- iv) LEDs (for output indication).
- v) Push-button switches (for JK inputs).
- vi) Clock pulse generator (for CLK input).

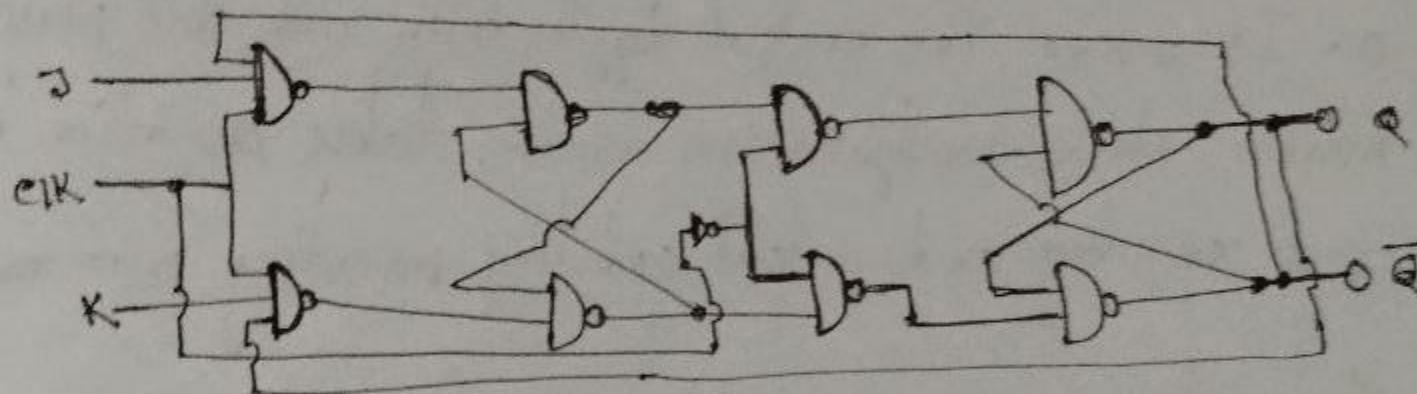
Boolean Function →

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

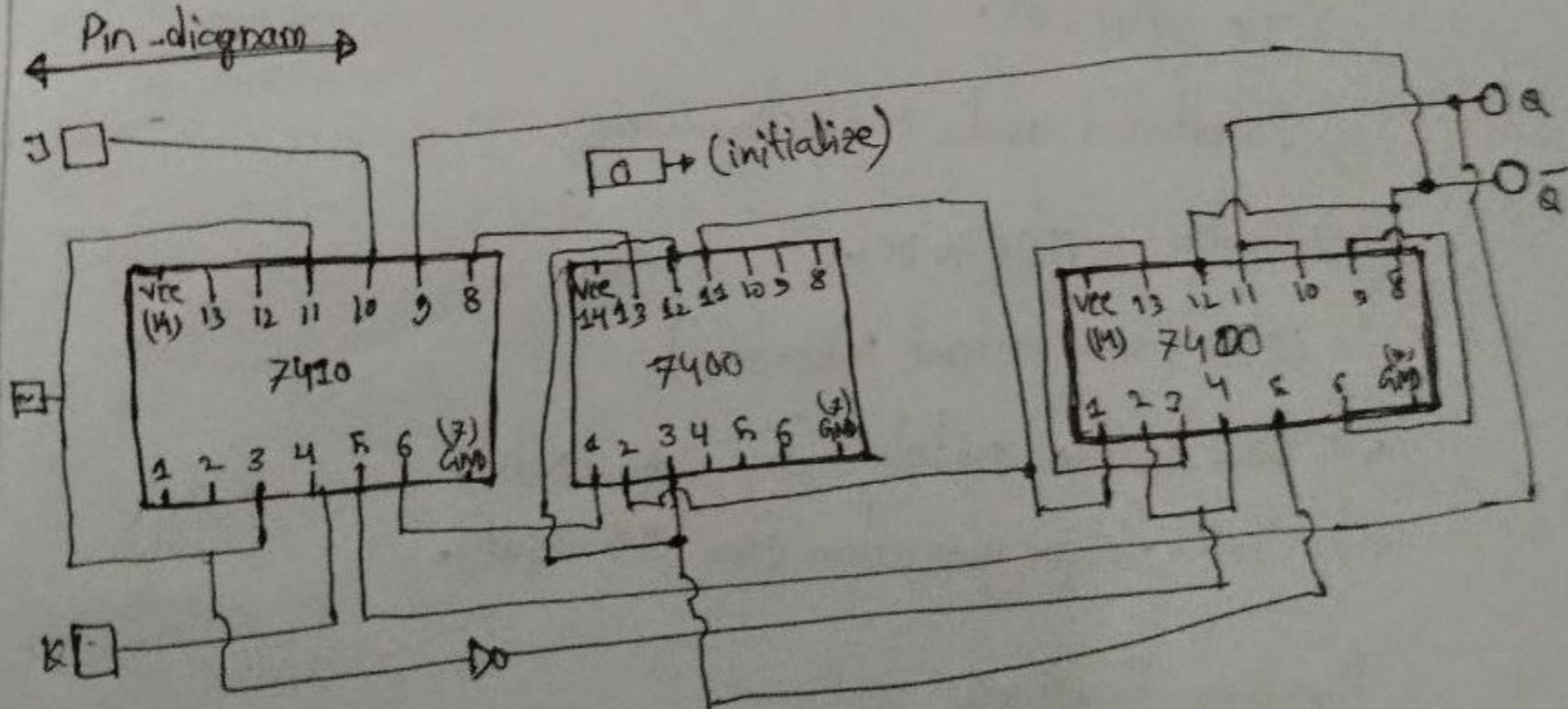
↔ Truth table →

J	K	CK	Q_{n+1}
0		0	Q_n
0		1	Q_n
1		1	0
1	0	1	1
1	1	1	\overline{Q} (flip)

↔ Logic diagram →



↔ Pin-diagram →



Observed Output:

Inputs (J, K, Clk)	Output (Q, Q̄)
0 0 1	No change (Q)
0 1 1	0 1
1 0 1	1 0
1 1 1	Toggle (Q)

Discussion: The Master-slave JK Flip-Flop effectively

eliminates the forbidden state problem of the RS Flip-Flop and introduces toggle functionality. The two-stage operation (Master-slave) ensures glitch-free transitions making it suitable for counters, registers and memory units. However, propagation delay between Master and Slave stages must be considered in high-speed circuits.

Ex: 8 - RS Flip flop

Experiment title: Study of RS Flip-flop

Theory: The RS Flip-Flop is a fundamental sequential logic circuit that serves as a basic memory element. It has two inputs: SET(s) and RESET(R), and two outputs: Q and Q'. The RS flip-flop can be constructed using either NOR gates or NAND gates with slight difference in their truth table. When s=1 and r=0, the output Q is set to 1, regardless of its previous state. Conversely, when s=0 and r=1, the output Q is reset to 0. If both inputs are s=0 and R=0, the flip-flop retains its previous state. However, the condition s=1 and R=1 is forbidden because it leads to an indeterminate state (Q and Q' become 0, violating the complementary rule). The RS flip-flop is widely used in basic storage applications but is limited but is limited due to its restricted no. of its input.

Apparatus:

- i) IC 74LS00 (NAND) gates or 74LS02 (NOR) gates.
- ii) Breadboard and connecting wires.
- iii) Power supply (5V DC).
- iv) LEDs for output indication.
- v) Push-button switches (for inputs S and R).

Boolean function:

For nand based RS flip-flop,

$$Q_{n+1} = S + R' Q_n$$

For nor based RS flip-flop,

$$Q_{n+1} \neq S + \bar{R} Q_n$$

Truth table

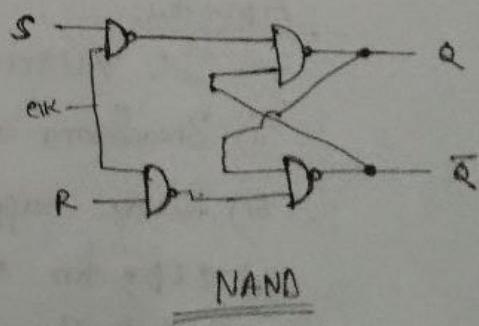
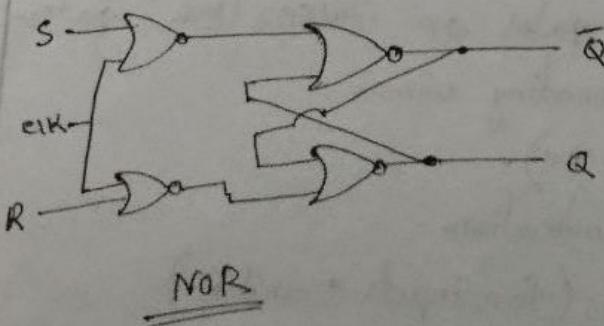
NOR implement

CLK	S	R	Q_{n+1}
0	X	X	Q_n
1	0	0	Q_n
1	0	1	0
1	1	0	1
1	1	1	invalid

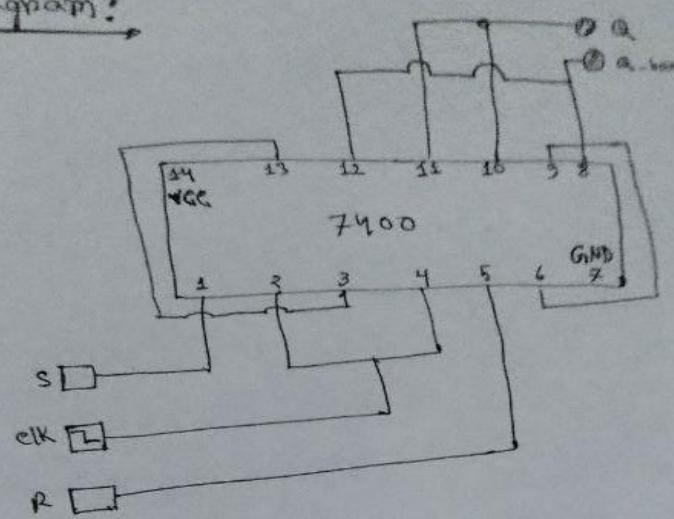
NAND implementation

CLK	S	R	Q_{n+1}
0	X	X	Q_n
1	0	0	Q_n
1	0	1	0
1	1	0	1
1	1	1	invalid

Logic Diagram



Pm Diagram:



with NAND Gate

Observed output:

CK	S	R	Q _{n+1}
0	x	x	Q _n
1	0	0	Q _n
1	0	1	0
1	1	0	1
1	1	1	invalid

Discussion → The RS Flip-Flop successfully demonstrates basic memory storage with set, reset, and hold functionalities. However, the forbidden state ($S=1, R=1$) leads to unpredictable behaviour, making it unsuitable for complex sequential circuits. The limitation can be overcome by more advanced flip-flops like JK flip-flops.

Experiment : 7

Ex: 7 - IC-7483 (Addition Operation)

Experiment Name : 4-bit adder circuit design using IC 7483. (addition operation)

Theory : A binary parallel adder is a digital function that produces the arithmetic sum of two binary numbers in parallel. It consists of full-adders connected in cascade with the output carry from one full-adder connected to the input carry of the next full-adder.

Apparatus : IC - 74283, connecting wires, power, bread board, simulation software, etc logic input switches (x8), leds for outputs (x5) etc.

Truth Table :

x	y	z	c	s
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	1	0
1	0	1	1	0
1	1	0	1	1
1	1	1	1	1

Boolean Expression

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

$$S = xyz + xy\bar{z} + \bar{x}yz' + \bar{x}y\bar{z}'$$

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

$$C = xy + xz + yz$$

Logic Diagrams An n-bit parallel adder requires n full-adders.

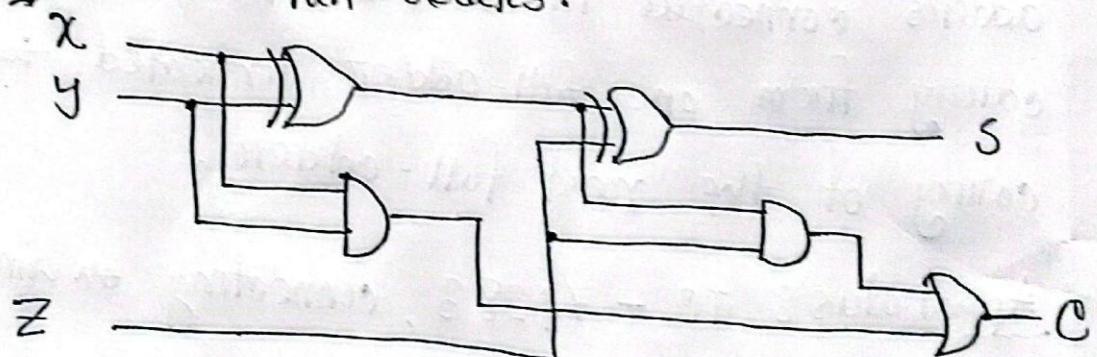


fig : Implementation of full-adder

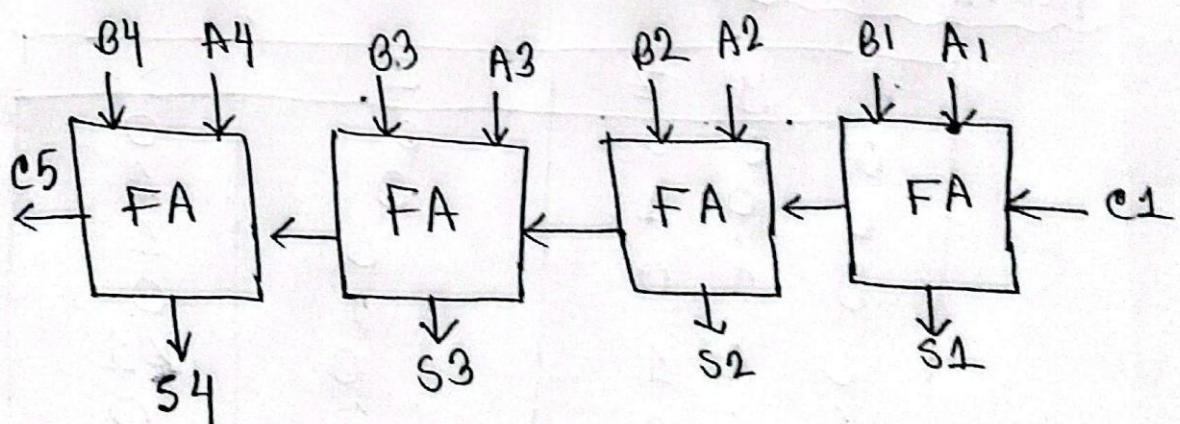


Fig: 4-bit full-adders

Pin Diagram :

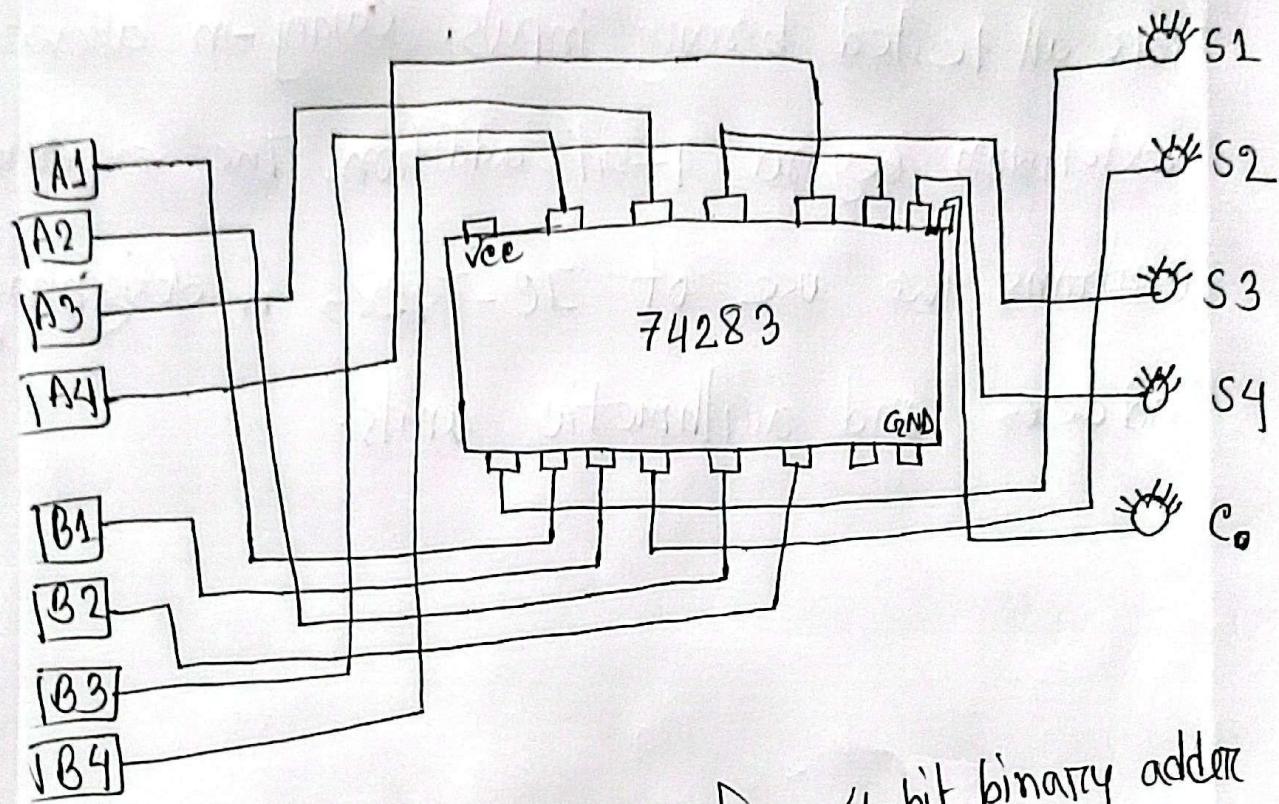


Fig: Pin diagram for 4-bit binary adder

Observed Output : Here, $A = 0110$, $B = 1100$

$$A = A_4 A_3 A_2 A_1$$

$$B = B_4 B_3 B_2 B_1$$

$$\begin{array}{r}
 & A_4 A_3 A_2 A_1 & B_4 B_3 B_2 B_1 \\
 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \\
 \hline
 & C_0 & S_4 & S_3 & S_2 & S_1 & C_4 \\
 & 0 & 0 & 0 & 0 & 0 & 1
 \end{array}$$

$$\begin{array}{r}
 \text{Now,} \quad 0110 \\
 \quad \quad \quad 1100 \\
 \hline
 \boxed{1} \quad 0010
 \end{array}$$

Carry

Hi we got the same output from the simulation also from the implementation in bread board.

Discussion 8 The IC -74283 functioned correctly for all tested binary inputs. Carry-in allowed extension beyond 4-bit addition. The experiment confirms the use of IC-74283 is designing adders and arithmetic units.

Ex: 7 - IC-7483 (BCD Sum)

Experiment : 7

Experiment Name : constructing BCD adder using IC type 7483.

Theory : A BCD adder adds two BCD digits and produces output as a BCD digit. A BCD or Binary Coded Decimal digit can't be greater than 9. If sum is greater than 9 or carry = 1, the result is wrong and correction must be done. The wrong result can be corrected adding six (0110) to it.

Apparatus : IC - 7483, Logic gates (AND, OR), Breadboard, connecting wires, Power supply etc.

Block Diagram :

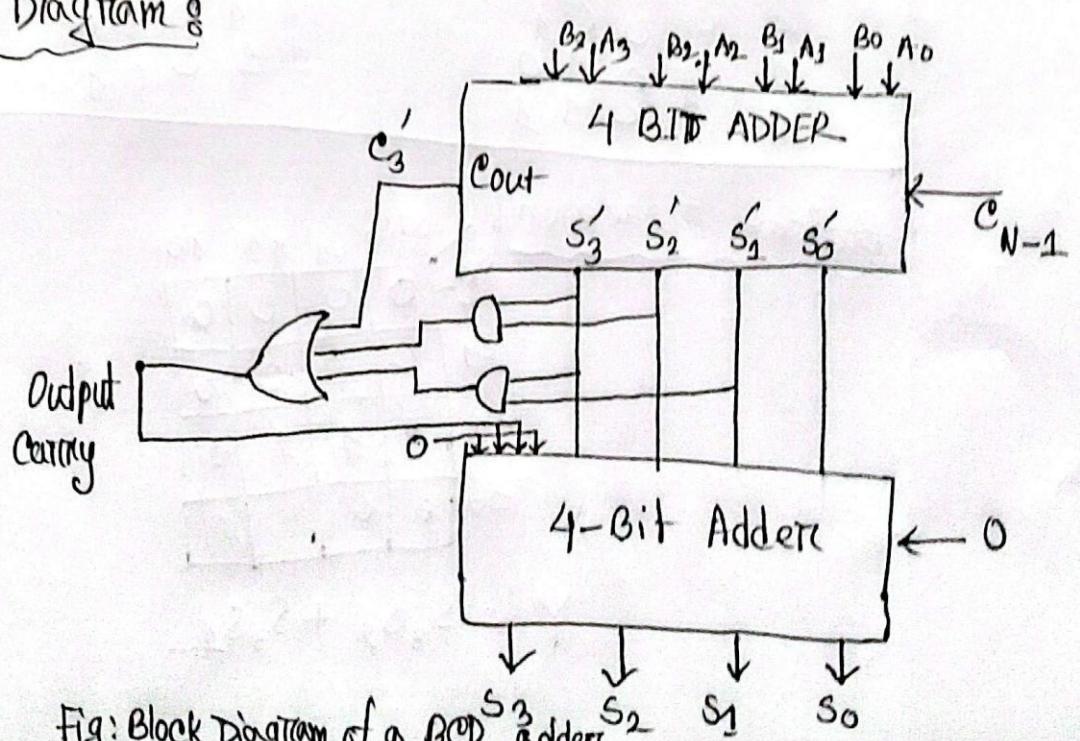


Fig: Block Diagram of a BCD adder

Truth Table :

<u>Binary Sum</u>					<u>BCD Sum</u>					<u>Decimal</u>	
C'	S_3'	S_2'	S_1'	S_0'	C	S_3	S_2	S_1	S_0	Y	
0	0	0	0	0	0	0	0	0	0	0	
0	0	0	0	1	0	0	0	0	1	1	
0	0	0	1	0	0	0	0	1	0	2	
0	0	0	1	1	0	0	0	1	1	3	
0	0	1	0	0	0	0	1	0	0	4	
0	0	1	0	1	0	0	1	0	1	5	
0	0	1	1	0	0	0	1	0	0	6	
0	0	1	1	1	0	0	1	1	1	7	
0	1	0	0	0	0	1	0	0	0	8	
0	1	0	0	1	0	1	0	0	1	9	
<hr/>					<hr/>					<hr/>	
0	1	0	1	0	1	0	0	0	0	10	
0	1	0	1	1	1	0	0	0	1	11	
0	1	1	0	0	1	0	0	1	0	12	
0	1	1	0	1	1	0	0	1	1	13	
0	1	1	1	0	1	0	1	0	0	14	
0	1	1	1	0	1	0	1	0	1	15	
0	1	1	1	1	1	0	1	1	0	16	
1	0	0	0	0	1	0	1	1	0	17	
1	0	0	0	1	1	0	1	1	1	18	
1	0	0	1	0	1	1	0	0	0	19	
1	0	0	1	1	1	1	0	0	1		

Boolean Expression

$S_3 S_2$	00	01	11	10
00	0	0	0	0
01	0	0	0	0
11	1	1	1	1
10	0	0	1	1

$$Y = S_3 S_2 + S_3 S_1$$

Pin Diagram:

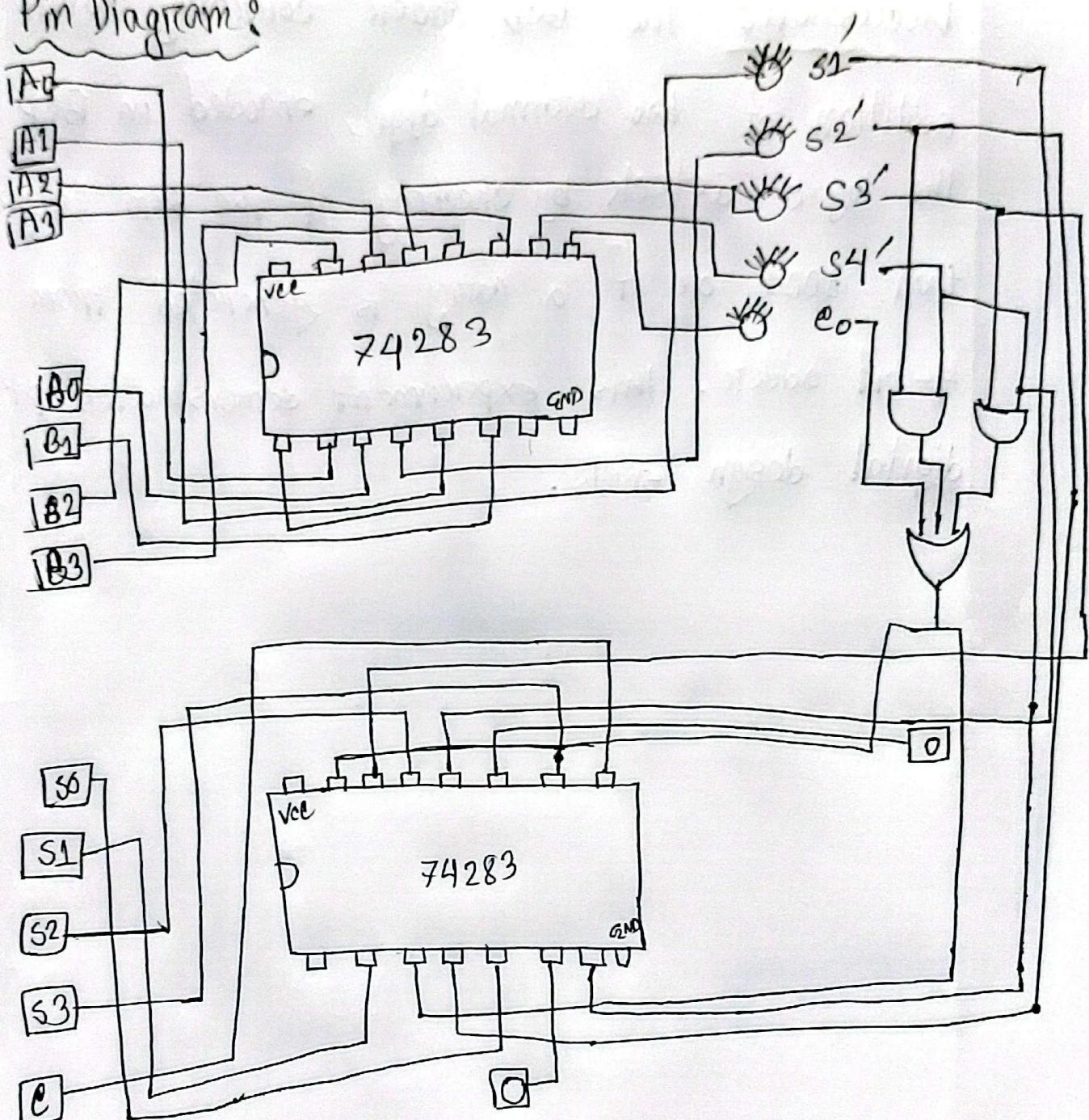
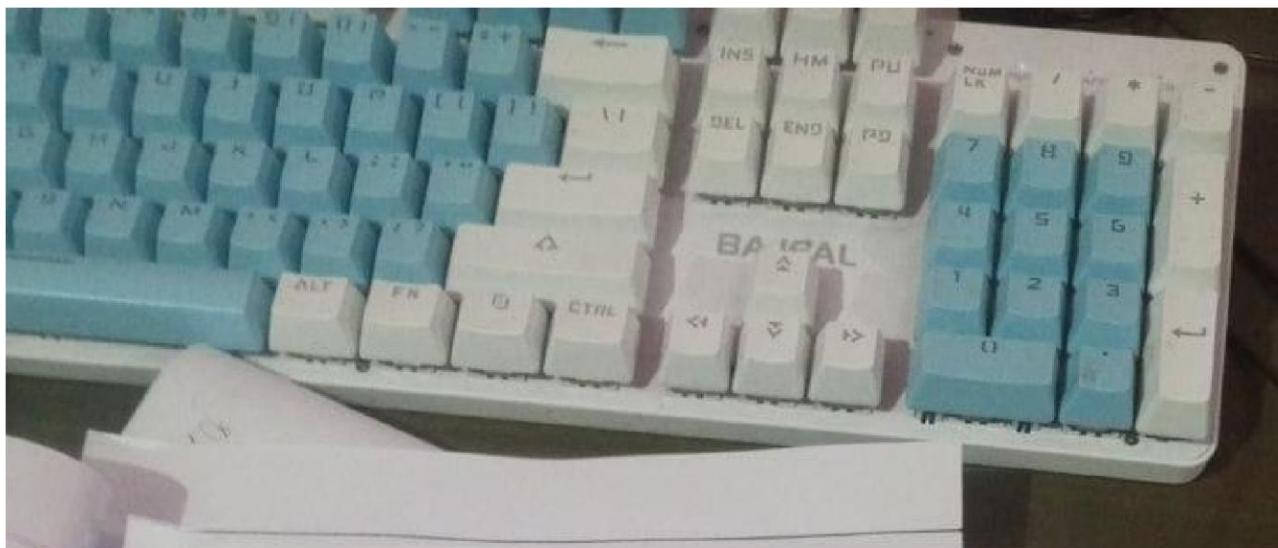


Fig: Pin diagram for BCD sum

Observation 8

A (BCD I/P)	B (BCD I/P)	Binary Sum	Correction Needed	Final BCD O/P	Carry
0100 (4)	0011 (3)	0111 (7)	No	0111 (7)	0
0110 (6)	0110 (6)	1100 (12)	Yes (add 0110)	0010 + Carry 1	1

Discussion: The BCD adder correctly handles the addition of two decimal digits encoded in BCD format. The logic detects by checking if the sum is greater than 1001 or if a carry is generated from the 4-bit adder. This experiment demonstrates practical digital design skills.



Ex: 5 - Design with Multiplexer

Experiment Name: Design with multiplexer

Theory: Multiplexer is a circuit which has many input lines and one output lines.

The function of the multiplexer is ~~set~~ to select one of the input lines and connect it to the output.

4x1 multiplexer: A four input multiplexer which contains 4 input I_0, I_1, I_2, I_3 which are not separately but selectively transmitted to the output Y depending on the select input combinations. Here two select line inputs are required as $2^N = 4$ where N is the number of select input i.e., $N=2$. Each data input is routed to the output depending on the select input combination.

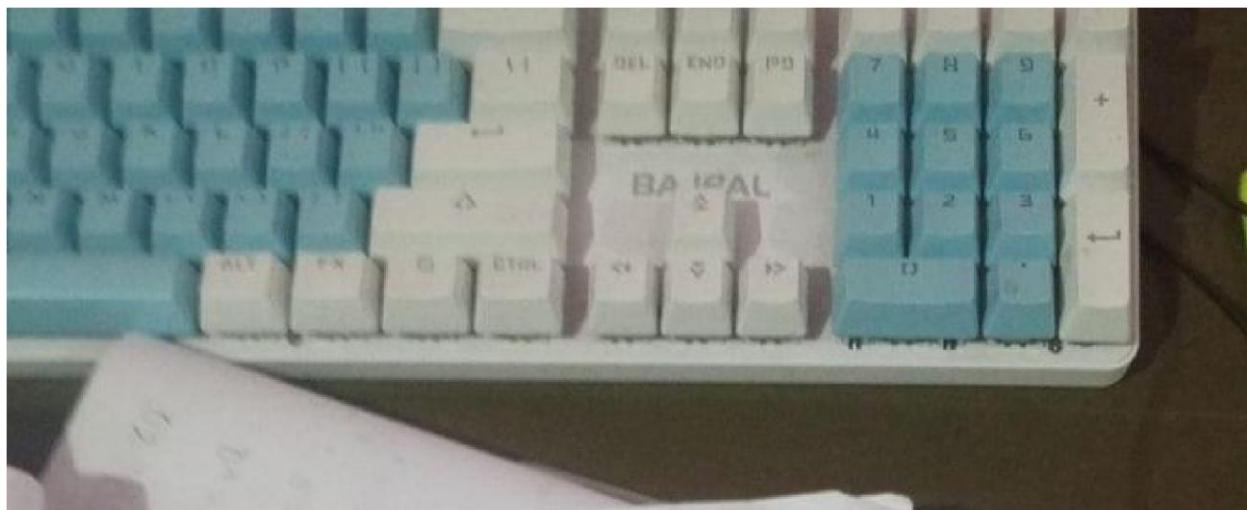


Apparatus: IC 7408, 7404, 7432, Connecting wires, breadboard, Power Supply,

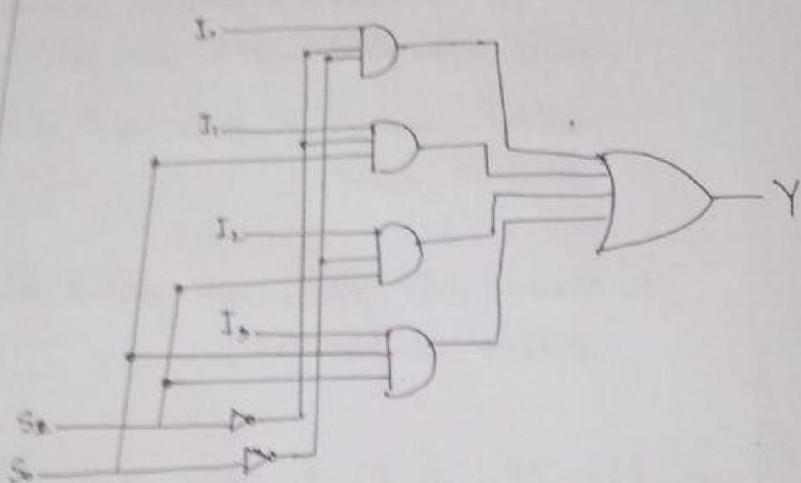
Truth table of 4x1 multiplexer:

Input	S_1	S_0	Y
I_0	0	0	I_0
I_1	0	1	I_1
I_2	1	0	I_2
I_3	1	1	I_3

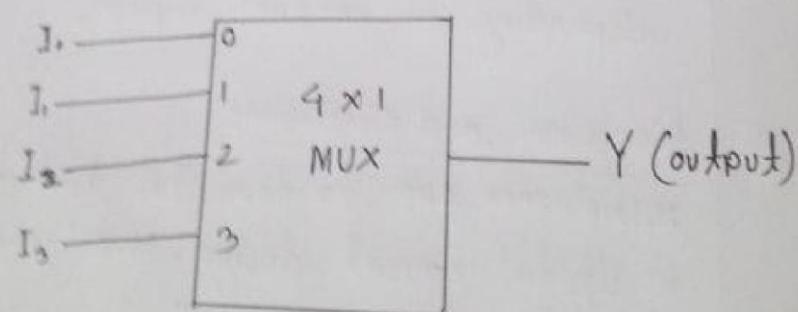
$$Y = \bar{S}_1 \bar{S}_0 I_0 + \bar{S}_1 S_0 I_1 + S_1 \bar{S}_0 I_2 + S_1 S_0 I_3$$



Logic Diagram:



Block Diagram:





Testing and Procedure:

1. The 4×1 mux is designed and implemented using appropriate IC's and ability to select one of 4 data input based on original signal.
2. Various data inputs are applied to ensure proper functionality.

Result: The 4 to 1 MUX accurately selected one of the 4 data input as the output depending on control input.

Discussion and Conclusion:

Multiplexers are fundamental building block in digital circuit design. MUXs are essential in data multiplexing where multiple data source are combined into one transmission line.

(20 वाले, 20 वाले) Minimum (20 वाले) (20 वाले)

Ex: 12 - BCD-to-7 Segment Display

Experiment title:

BCD to 7 segment display Decoder

Theory:-

A BCD to 7 segment display decoder converts a 4 bit binary coded decimal (BCD) input into signals that can drive a 7-segment display to show digits 0-9. Each digit on the display is formed by turning on the appropriate combination of segments (labeled a to g).

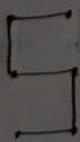
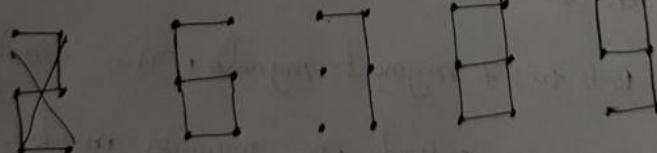
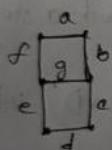
The BCD input consists of 4 bits (A, B, C, D) where:

- A is the most significant bit (MSB)
- D is the least significant bit (LSB)

A decoder like 7447 or 4511 is used to convert BCD to 7 segment signals. The display used can be common cathode or common anode, and the decoder output logic differs for each type.

Apparatus :-

- 1) 8-bit to 7-Segment Decoder (e.g. 7447 or 4511)
- 2) 7-segment display
- 3) 4-bit Binary input switches.
- 4) Resistors
- 5) Breadboard
- 6) connecting wires.
- 7) +5v power supply



	A	B	c	D	a	b	c	d	e	f	g	
0	0	0	0	0	1	1	1	1	1	1	0	
1	0	0	0	1	0	1	1	0	0	0	0	
2	0	0	1	0	1	1	0	1	1	0	1	
3	0	0	1	1	1	1	1	1	0	0	1	
4	0	1	0	0	0	1	1	0	0	1	1	
5	0	1	0	1	1	0	1	1	0	1	1	
6	0	1	1	0	1	0	1	1	1	1	1	
7	0	1	1	1	1	1	1	1	0	0	0	
8	1	0	0	0	1	1	1	1	1	1	1	
9	1	0	0	1	1	1	1	1	1	0	1	

Equation: —

$$a = A + c'D + b'D'$$

$$b = b' + c'D' + cD$$

$$c = c' + D + b$$

$$d = A + b'D' + b'c + cD' + b'c'D$$

$$e = b'D + cD$$

$$f = A + c'D' + b'c + bD'$$

$$g = A + b'c' + b'a + cD$$

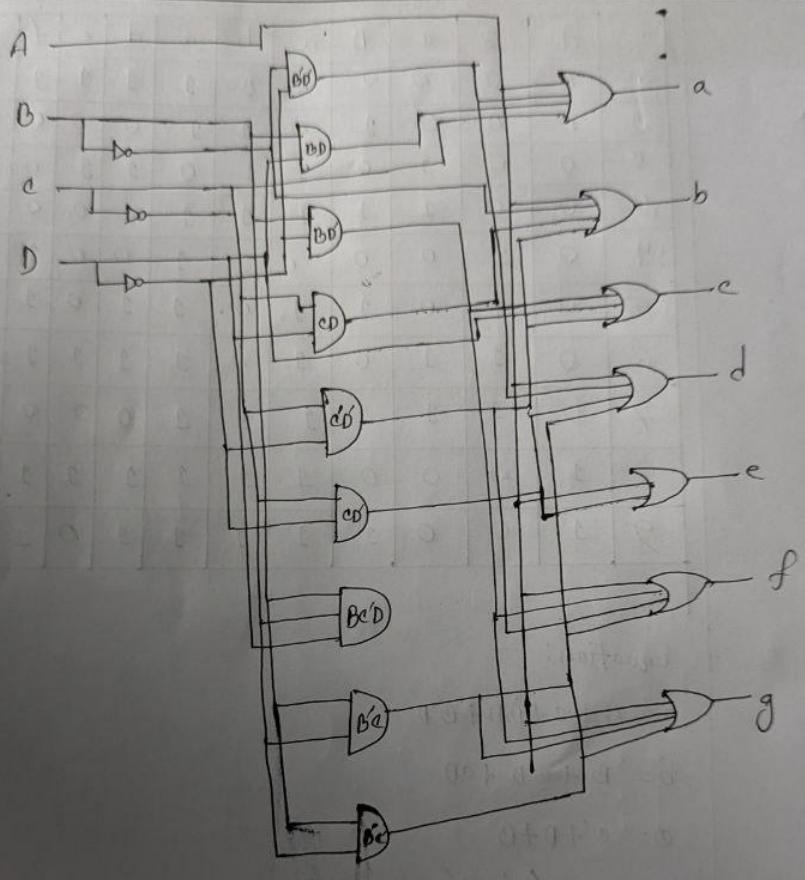


Fig: Schematic of BCD to 7-segment Decoder

Output

when a 4-bit BCD number is applied to the input of the decoder.

- The decoder outputs logic HIGH on low to each of the 7 segments
- The segments light up to form the decimal digit corresponding to the BCD input
- For example, BCD 0100 will display number "4" on the 7 segment display.

Ex: 6 - Full Adder

Experiment 5: Arithmetic Circuit design (full adder)

Components Used:

1. Jumper Wire
2. EX-OR Gate IC 74LS86
3. AND Gate IC 74LS08
4. OR Gate IC 74LS32

Theory:

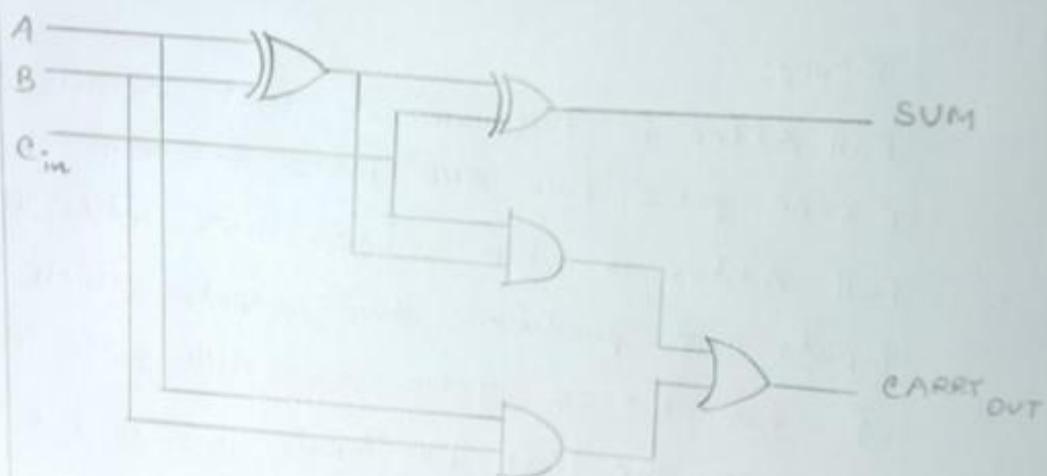
Full Adder is the circuit that consists of two EX-OR gates, two AND gates and one OR gate. Full Adder is the adder that adds three inputs and produces two outputs which consist of two EX-OR gates, two AND gates, and one OR gate. The first two inputs are A and B and the third input is an input carry as C_{in} . The output carry is designated as C_{out} and the normal output is designated as S which is SUM. The equation obtained by the EX-OR gate is the sum of binary digits. While the output obtained by the AND gate is the carry obtained by addition.

Logical Expression :

$$SUM = (A \oplus B) \oplus C_{in}$$

$$CARRY_{out} = AB + C_{in}(A \oplus B)$$

Full Adder Logic Gate Diagram :



Truth Table :

A	B	C _{in}	C _{out}	S
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

Discussion: After doing this laboratory experiment I had learnt practically the carry concept in Binary addition. I implemented my learnings in making an addition circuit for three inputs with a binary 3-bit full adder.

Ex: 4 - Encoder Circuit

Experiment No: 04

Experiment Title: Encoder circuit design.

Theory:

An encoder is a combinational circuit that converts active input signals into coded output signals. In simple terms, it has multiple input lines, but only one is active at a time and it encodes that into a binary output.

An encoder has 2^n input lines and n output lines.

Apparatus:

Breadboard, ICs - 7408, 7403, 7404, 4072, connecting wires, power supply, LED, resistors.

Boolean Functions:

For 8-to-3 encoder, there will be 8 input lines and 3 output lines.

$$x = D_4 + D_5 + D_6 + D_7$$

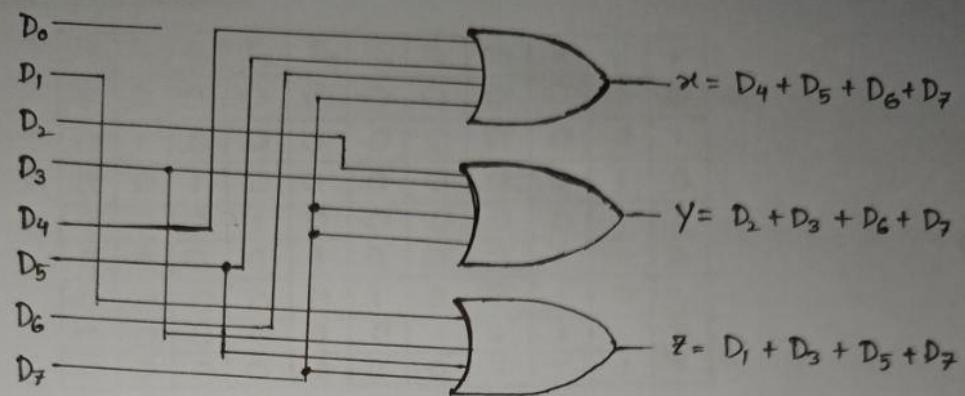
$$y = D_2 + D_3 + D_6 + D_7$$

$$z = D_1 + D_3 + D_5 + D_7$$

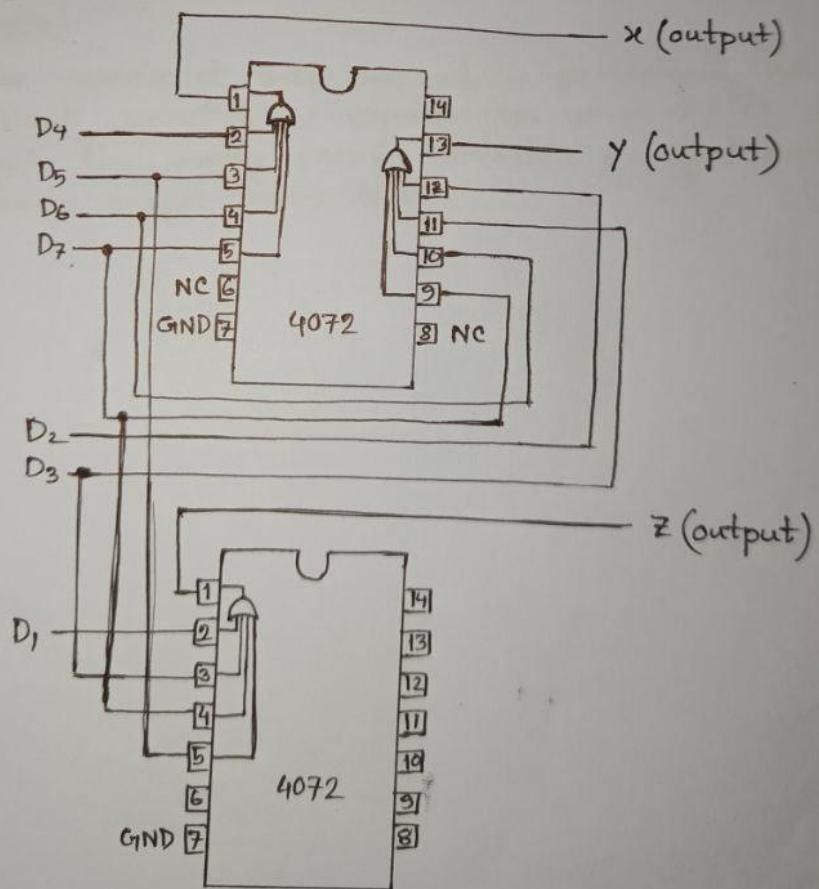
Truth Table:

D_0	D_1	D_2	D_3	D_4	D_5	D_6	D_7	x	y	z
1	0	0	0	0	0	0	0	0	0	0
0	1	0	0	0	0	0	0	0	0	1
0	0	1	0	0	0	0	0	0	1	0
0	0	0	1	0	0	0	0	0	1	1
0	0	0	0	1	0	0	0	1	0	0
0	0	0	0	0	1	0	0	1	0	1
0	0	0	0	0	0	0	1	1	0	0
0	0	0	0	0	0	0	1	1	1	1

Logic Diagram:



PIN Diagram:



Observed Output:

D ₀	D ₁	D ₂	D ₃	D ₄	D ₅	D ₆	D ₇	x	y	z
1	0	0	0	0	0	0	0	0	0	0
0	1	0	0	0	0	0	0	0	0	1
0	0	1	0	0	0	0	0	0	1	0
0	0	0	1	0	0	0	0	0	1	1
0	0	0	0	1	0	0	0	1	0	0
0	0	0	0	0	1	0	0	1	0	1
0	0	0	0	0	0	1	0	1	1	0
0	0	0	0	0	0	0	1	1	1	1

Conclusion:

The experiment was successfully performed, the encoder circuit operated as expected. We were able to verify that the verify encoder converts a signal into a binary unique binary code.

Ex: 4 - Decoder Circuit

Experiment No: 04

Experiment Title: Decoder circuit design

Theory:

A decoder is a combinational circuit that converts n input lines to a maximum of 2^n lines.

For example, 3-to-8 decoder has 3 input (x, y, z) and 8 output lines ($D_0 - D_7$), where only one output is high at a time.

Apparatus:

Breadboard, IC-4073, LED, resistor, connecting wire, power supply (5v), Logiaprobe.

Boolean Function:

For 3-to-8 decoder, inputs x, y, z , outputs ($D_0 - D_7$)

$$D_0 = x'y'z'$$

$$D_1 = x'y'z$$

$$D_2 = x'yz'$$

$$D_3 = x'yz$$

$$D_4 = xy'z'$$

$$D_5 = xy'z$$

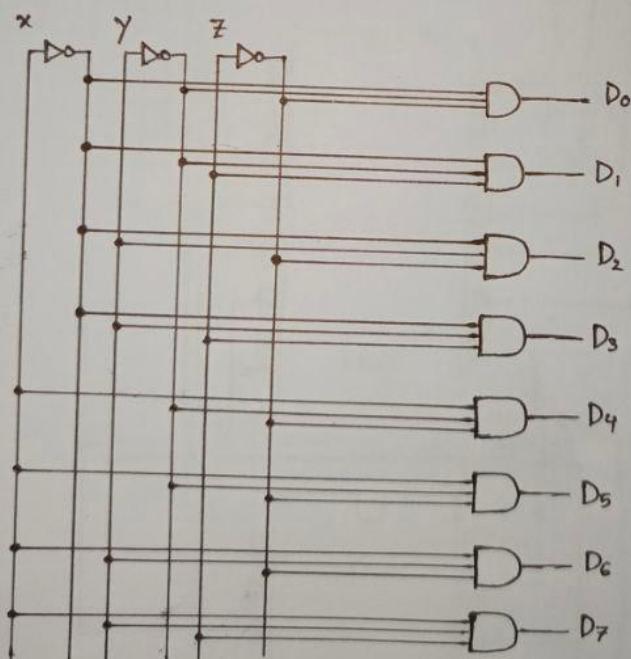
$$D_6 = xyz'$$

$$D_7 = xyz$$

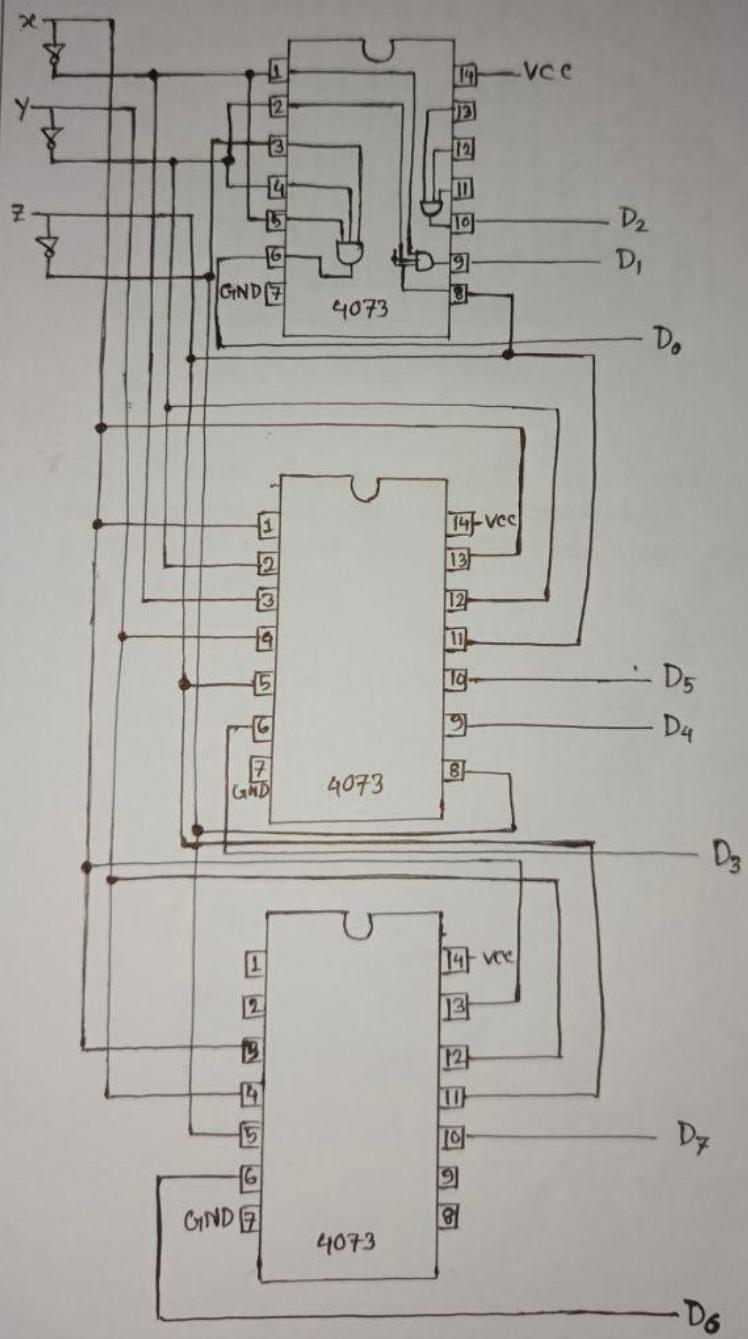
Truth Table:

x	y	z	D ₀	D ₁	D ₂	D ₃	D ₄	D ₅	D ₆	D ₇
0	0	0	1	0	0	0	0	0	0	0
0	0	1	0	1	0	0	0	0	0	0
0	1	0	0	0	1	0	0	0	0	0
0	1	1	0	0	0	1	0	0	0	0
1	0	0	0	0	0	0	1	0	0	0
1	0	1	0	0	0	0	0	1	0	0
1	1	0	0	0	0	0	0	0	1	0
1	1	1	0	0	0	0	0	0	0	1

Logic Diagram:



PIN Diagram:



Observed Output:

x	y	z	D ₀	D ₁	D ₂	D ₃	D ₄	D ₅	D ₆	D ₇
0	0	0	1	0	0	0	0	0	0	0
0	0	1	0	1	0	0	0	0	0	0
0	1	0	0	0	1	0	0	0	0	0
0	1	1	0	0	0	1	0	0	0	0
1	0	0	0	0	0	0	1	0	0	0
1	0	1	0	0	0	0	0	1	0	0
1	1	0	0	0	0	0	0	0	1	0
1	1	1	0	0	0	0	0	0	0	1

Conclusion:

The 3-to-8 decoder circuit was successfully designed and implemented. The output confirmed that for every 3-bit input combination, only one corresponding output line is activated.