Bangladesh University of Business and Technology (BUBT)
Faculty of Engineering& Applied Sciences (FEAS)
Department of Computer Science and Engineering (CSE)

LAB MANUAL

1	**Program**	B.Sc. Eng. in CSE
2	**Course Code**	CSE 206
3	**Course Title**	Digital Logic Design Lab
4	**Course Type**	Core Course
5	**Credit Hour**	1.50
6	**Course Objectives**	The main objectives of this sessional course are to reinforce the theory presented in the CSE 2 Digital Logic Design course by allowing students to experiment with digital logic in a hands- manner. And also understand both logical and electrical characteristics of digital design.
7	**Text Book**	1. Digital Fundamentals, 9th Edition, by Thomas L. Floyd
8	**Reference Book**	1. Digital Design -With an introduction to the Verilog HDL, 5th Edition, by M. Morris Man Michael D. Ciletti
9	**Course Outcomes (COs)**	Upon completing this course students will be able to: **CO1**: **Demonstrate** the working principles of different ICs & basic logic gates. **CO2:** **Apply** different IC's & logic gates for different digital logic circuits. **CO3:** Design and **implement** combinational circuits and sequential circuits.

Mapping of COs to POs

Course Outcome	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	√											
CO2									√			
CO3			√									

CO No.	PO No.	Bloom's Domain / Level	Delivery Methods / Activities	Assessment Tools
CO1	PO1	Cognitive / Understanding	Lectures, Lab Task, Assignment.	Lab Performance
CO2	PO9	Affective/Responding	Lectures, Lab Task, Assignment.	Lab Performance and Final Exam
CO3	PO3	Psychomotor / Guided Response	Lectures, Lab Task, Assignment.	Lab Performance and Final Exam

10 Weekly Schedule

Week	Lab	Task	Outcome

Week1	Lab1	To verify the Behavior of Logic Gates using Truth Table and familiarization with Digital Integrated Circuits	CO1
Week2	Lab2	Implementation of Boolean Function using Logic Gates **Lab Performance Evaluation**	CO1
Week3	Lab3	Simplification of Boolean function by using Boolean Algebra and K- Map. **Lab Performance Evaluation**	CO2
Week4	Lab4	Construct and test various adder and sub-tractor circuits **Lab Performance Evaluation**	CO2
Week5	Lab5	Design, construct and test combination logic circuits like Majority Logic circuit, Parity Generator etc.	CO2
Week6	Lab6	Design of Code converters like Gray Code to Binary, Seven-Segment Display etc. **Lab Performance Evaluation**	CO2
Week7	Lab7	To check the operation of 2-to-4-line Decoder/3-to-8-line Decoder and design 4 to 16 line decoder.	CO2
Week8	**Mid Term Week**		
Week9	Lab8	To Design BCD (Binary Code Decimal) to 7-SEGMENT Decoder.	CO2
Week10	Lab9	To check the operation of active low decoder. Binary Coded Decimal encoder. Designing of octal to binary encoder and decimal to binary encoder. **Lab Performance Evaluation**	CO2
Week11	Lab10	Designing of 2-to-1, 4-to-1-line multiplexer and a quadruple 2-to-1-line MUX. Also implement 1-to-4 line de-multiplexer.	CO3
Week12	Lab11	Designing of 4-BIT Magnitude comparator and magnitude comparison using 74LS85 IC Implementing a Full Adder using (a) Decoder (b) Multiplexer	CO3
Week13	Lab12	Construct, test and investigate the operation of various flip-flop circuits, SR Latch, RS flip-flops, D flip-flop etc. **Lab Performance Evaluation**	CO3
Week 14	Lab13	Construction of Asynchronous Counters, Synchronous Counters and Registers. **Lab Performance Evaluation**	CO3
Week15	**Lab Final Exam**		

11	**S/W and H/W Requirements**	The following software's are needed to conduct this lab course: (i) Proteus (ii) Circuitverse The following equipment's are needed to conduct this lab course: (i) Breadboard (ii) IC's (iii) Connecting wires (iv) LED's
12	**Experiment No.: 01**	

Experiment No. 01:

Name of the Experiment: To verify the Behavior of Logic Gates using Truth Table and familiarizati with Digital Integrated Circuits.

Objective:

Verification and interpretation of truth tables for AND, OR, NOT, NAND, NOR Exclusive OR (EX-OR), Exclusive NOR (EX-NOR) Gates.

Course Outcomes (COs), Program Outcomes (POs) and Assessment:

CO Statement	Corresponding PO	Domain / level of learning taxonomy	Delivery methods and activities	Assessment tools
CO1**:** Demonstrate the working principles of different ICs & basic logic gates.	**Engineering Knowledge (PO1)**	**Cognitive / Understanding**	☑ Simulation ☑ Experiment ☑ Practice lab ☑ Group discussion ☑ Tutorial	☑ Lab tests ☑ Lab reports ☑ Final lab test ☐ Open ended lab ☐ Project show & project presentation

Theory:

Logic gates are electronic circuits which perform logical functions on one or more inputs to produ one output. There are seven logic gates. When all the input combinations of a logic gate are written a series and their corresponding outputs written along them, then this input/ output combination is calle Truth Table. OR, AND, NOT are basic gates. NAND, NOR are known as universal gates. Various gat and their working is explained here.

Required equipment and devices:

SL No.	COMPONENT	SPECIFICATION	QTY
1.	AND GATE	IC 7408	1
2.	OR GATE	IC 7432	1
3.	NOT GATE	IC 7404	1
4.	NAND GATE 2 I/P	IC 7400	1
5.	NOR GATE	IC 7402	1
6.	X-OR GATE	IC 7486	1

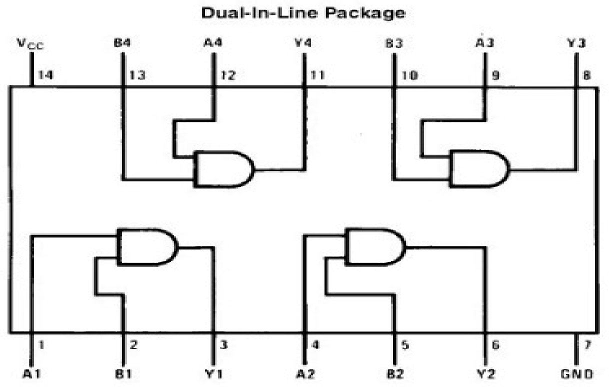
7.	IC TRAINER KIT	-	1
8.	PATCH CORD	-	As per Required

BASIC GATES:

1. AND GATE

The AND gate performs a logical multiplication commonly known as AND function. The output high when both the inputs are high. The output is low level when any one of the inputs is low.

InputA	InputB	Output
1	1	1
1	0	0
0	1	0
0	0	0

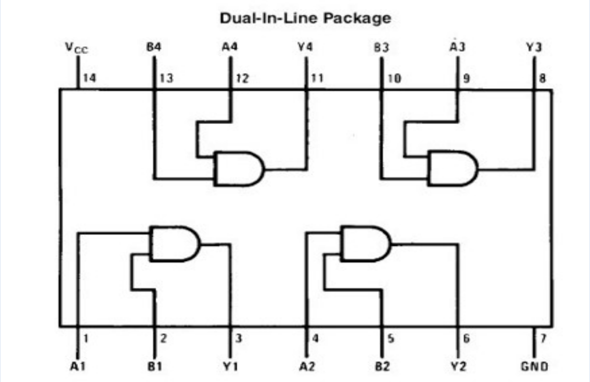


IC 7408

2. OR GATE

The OR gate performs a logical addition commonly known as OR function. The output is high whe any one of the inputs is high. The output is low level when both the inputs are low.

InputA	InputB	Output
1	1	1
1	0	1
0	1	1
0	0	0

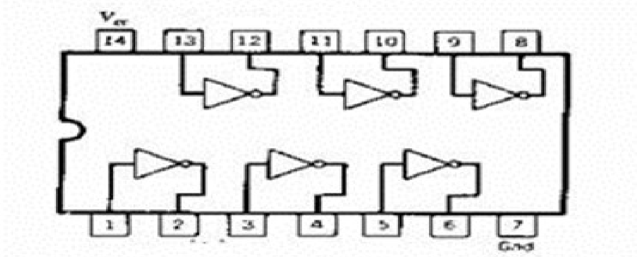


IC 7432

3. NOT GATE

The NOT gate is called an inverter. The output is high when the input is low. The output is low whe the input is high.

Input	Output
0	1
1	0



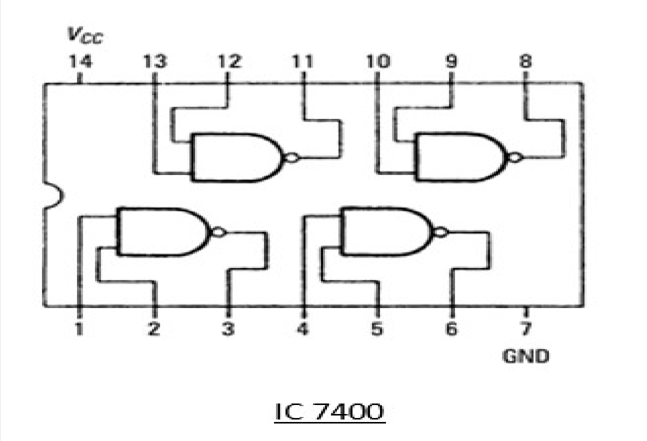
IC 7404

UNIVERSAL GATES:

1. NAND GATE

The NAND gate is a contraction of AND-NOT. The output is high when both inputs are low and ar one of the input is low .The output is low level when both inputs are high.

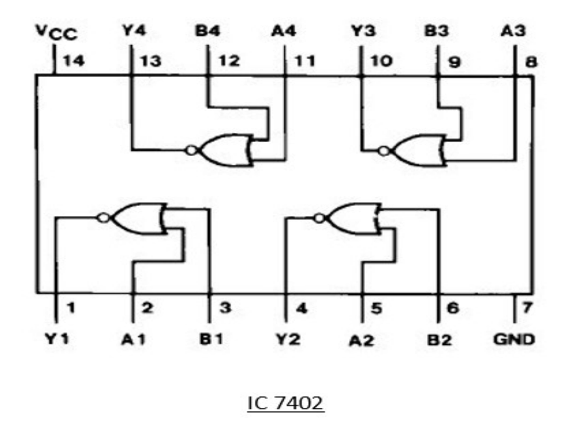
Input A	Input B	Output
1	1	0
1	0	1
0	1	1
0	0	1



2. NOR GATE

The NOR gate is a contraction of OR-NOT. The output is high when both inputs are low. The output low when one or both inputs are high.

Input A	Input B	Output
1	1	0
1	0	0
0	1	0
0	0	1



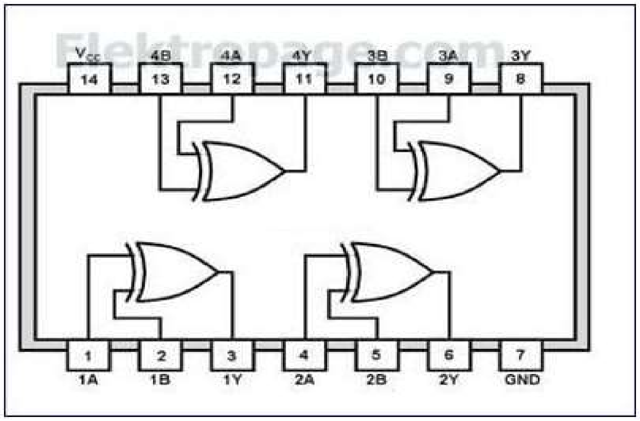
ADVACED GATES:

1. X-OR GATE

The output is high when any one of the inputs is high. The output is low when both the inputs are lo and both the inputs are high.

Input A	Input B	Output
1	1	0

1	0	1
0	1	1
0	0	0



IC 74136

PROCEDURE:

1. Connect the trainer kit to ac power supply.
2. Connect the inputs of any one logic gate to the logic sources and its output to the logic indicator.
3. Apply various input combinations and observe output for each one.
4. Verify the truth table for each input/ output combination.
5. Repeat the process for all other logic gates.
6. Switch off the ac power supply.

DISCUSSIONS:

References:

1. Reference book.
2. Online sources.

21 **Experiment No.: 02**

Experiment No.02

Name of the Experiment: Realization of Basic gates using Universal Gates.

- **Objective:** To construct NOT, AND, OR, Exclusive OR (EX-OR), Exclusive NOR (EX-NOR) logic gates using only NAND gates.
- To construct NOT, AND, OR, Exclusive OR (EX-OR), Exclusive NOR (EX-NOR) logic gates using only NOR gates.

Course Outcomes (COs), Program Outcomes (POs) and Assessment:

CO Statement	Corresponding PO	Domain / level of learning taxonomy	Delivery methods and activities	Assessment tools
CO1**:** Demonstrate the working principles of different ICs & basic logic gates.	**Engineering Knowledge (PO1)**	**Cognitive / Understanding**	☑ Simulation ☑ Experiment ☑ Practice lab ☑ Group discussion ☑ Tutorial	☑ Lab tests ☑ Lab reports ☑ Final lab test □ Open ended lab □ Project show & project presentation

Components required:

SL No.	COMPONENT	SPECIFICATION	QTY
1.	NAND GATE 2 I/P	IC 7400	1
2.	NOR GATE 2 I/P	IC 7402	1
3.	IC TRAINER KIT	-	1
4.	PATCH CORD	-	As per required

(A) NAND AS A UNIVERSAL GATE:

THEORY:

NAND gate is actually a combination of two logic gates: AND gate followed by NOT gate. So its outp is complement of the output of an AND gate. This gate can have minimum two inputs, output is alwa one. By using only NAND gates, we can realize all logic functions: AND, OR, NOT, X-OR, X-NO NOR. So this gate is also called universal gate.

1. **NAND gate as NOT gate:**

A NOT produces complement of the input. It can have only one input, tie the inputs of a NAND gate together. Now it will work as a NOT gate. Its output is

$$\mathbf{Y = (A.A)' = (A)'}$$

2. **NAND gates as AND gate:**

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

$$\mathbf{Y = ((A.B)')'= (A.B)}$$

3. **NAND gates as OR gate:**

From De Morgan’s theorems: $(A.B)' = A' + B'$. Similarly, $(A'.B')' = A'' + B'' = A+ B$. So, give the inverted inputs to a NAND gate, obtain OR operation at output.

4. **NAND gates as EX-OR gate:**

The output of a two input EX-OR gate is given by: $Y = A'B + AB'$. EX-OR gate can be implemented using four NAND gates as follows.

Gate No.	Inputs	Output
1	A, B	(AB)’
2	A, (AB)’	(A (AB)’)’
3	(AB)’, B	(B (AB)’)’
4	(A (AB)’)’, (B (AB)’)’	A’B + AB’

Now the output from gate no. 4 is the overall output of the configuration.

$$\begin{aligned}
\mathbf{Y} \; &\mathbf{=((A\,(AB)')'\,(B\,(AB)')')'} \\
&= \mathbf{(A(AB)')'' + (B(AB)')''} \\
&= \mathbf{(A(AB)') + (B(AB)')} \\
&= \mathbf{(A(A' + B)') + (B(A' + B'))} \\
&= \mathbf{(AA' + AB') + (BA' + BB')} \\
&= \mathbf{(0+AB'+BA'+0)} \\
&= \mathbf{AB' + BA'}
\end{aligned}$$

$$\mathbf{So\ Y \quad = \quad AB' + A'B}$$

5. **NAND gates as EX-NOR gate**

EX-NOR gate is actually EX-OR gate followed by NOT gate. So give the output of EX-OR gate to a NOT gate, overall ouput is that of an EX-NOR gate.

Y = AB+ A'B'

PROCEDURE:

(i) Verify the gates and connect the NAND gates as per logic diagrams (A) for any of the logic functions to be realized.

(ii) Connect Pin-14 of all ICs to +5V and Pin-7 to ground.

(iii) Feed the logic 0 (0V) or 1(5V) in different combinations at the inputs A & B according to truth table.

(iv) Observe and note down the output readings for Y for different combinations of inputs and verify the truth table for input/output combination

(v) Repeat the process for all logic functions/gates.

Logic diagrams (A) observations table (A)

1. **NAND gate as NOT gate:**

A	Y
0	1
1	0

2. **NAND gates as AND gate:**

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

3. **NAND gates as OR gate:**

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

4. **NAND gates as EX-OR gate:**

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

5. **NAND gates as EX-NOR gate**

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

DISCUSSIONS (A):

(B) NOR AS A UNIVERSAL GATE:

THEORY:

NOR gate is actually a combination of two logic gates: OR gate followed by NOT gate. So its outpu is complement of the output of an OR gate. This gate can have minimum two inputs, output is alway one. By using only NOR gates, we can realize all logic functions: AND, OR, NOT, X-OR, X-NOR, NAND. So this gate is also called universal gate.

1. **NOR gate as NOT gate:**

A NOT produces complement of the input. It can have only one input, tie the inputs of a NOR ga together. Now it will work as a NOT gate. Its output is

$$\mathbf{Y = (A+A)' = (A)'}$$

2. **NOR gates as OR gate:**

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

$$\mathbf{Y = ((A+B)')' = (A+B)}$$

3. NOR gates as AND gate:

From De Morgan’s theorems: $(A+B)' = A'.\ B'$. Similarly, $(A'+B')' = A''.\ B'' = A\ .B$ So, give the inverted inputs to a NOR gate, obtain AND operation at output.

4. NOR gates as EX-NOR gate:

The output of a two input EX-NOR gate is given by: $Y = AB + A'B'$. EX-NOR gate can be implemented using four NOR gates as follows.

Gate No.	Inputs	Output
1	A, B	$(A + B)'$
2	$A, (A + B)'$	$(A + (A+B)')'$
3	$(A + B)', B$	$(B + (A+B)')'$
4	$(A + (A + B)')', (B + (A+B)')'$	$AB + A'B'$

Now the output from gate no. 4 is the overall output of the configuration.

$$\begin{aligned}
\mathbf{Y} &\mathbf{=((A + (A+B)')' (B +(A+B)')')'} \\
&= \mathbf{(A+(A+B)')''.(B+(A+B)')''} \\
&= \mathbf{(A+(A+B)').(B+(A+B)')} \\
&= \mathbf{(A+A'B').(B+A'B')} \\
&= \mathbf{(A + A').(A + B').(B+A')(B+B')} \\
&= \mathbf{1.(A+B').(B+A').1} \\
&= \mathbf{(A+B').(B+A')} \\
&= \mathbf{A.(B + A') +B'.(B+A')} \\
&= \mathbf{AB + AA' +B'B+B'A'} \\
&= \mathbf{AB+0+0+B'A'} \\
&= \mathbf{AB + B'A'}
\end{aligned}$$

$$\mathbf{So\quad Y \quad = \quad AB + A'B'}$$

5. NOR gates as EX-OR gate

EX-OR gate is actually EX-NOR gate followed by NOT gate. So give the output of EX-NOR gate to a NOT gate, overall output is that of an EX-OR gate.

Y = A'B+ AB'

PROCEDURE:

(i) Verify the gates and connect the NOR gates as per logic diagrams (B) for any of the logic functions to be realized.

(ii) Connect Pin-14 of all ICs to +5V and Pin-7 to ground.

(iii) Feed the logic 0 (0V) or 1(5V) in different combinations at the inputs A & B according to truth table.

(iv) Observe and note down the output readings for Y for different combinations of inputs and verify the truth table for input/output combination

(v) Repeat the process for all logic functions/gates

LOGIC DIAGRAMS (B):

OBSERVATION TABLE (B):

1. NOR gate as NOT gate:

A	Y
0	1
1	0

2. NOR gates as OR gate:

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

3. NOR gates as AND gate:

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

4. NOR gates as EX-NOR gate:

A	B	Y
0	0	0
0	1	1

1	0	1
1	1	0

5. **NOR gates as EX-OR gate**

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

DISCUSSIONS (B):

References:

1. Reference book.
2. Online sources.

22 **Experiment No.: 03**

Experiment No. 03

Name of the experiment: Simplification of Boolean function.

Objectives: To simplify the Boolean expression and to build the logic circuit. Given a Truth table derive the Boolean expressions and build the logic circuit to realize it.

Course Outcomes (COs), Program Outcomes (POs) and Assessment:

CO Statement	Corresponding PO	Domain / level of learning taxonomy	Delivery methods and activities	Assessment tools
CO1: Demonstrate the working principles of different ICs & basic logic gates.	**Engineering Knowledge (PO1)**	**Cognitive / Understanding**	☑ Simulation ☑ Experiment ☑ Practice lab ☑ Group discussion ☑ Tutorial	☑ Lab tests ☑ Lab reports ☑ Final lab test ☐ Open ended lab ☐ Project show & project presentation

Components Required:

IC 7400, IC 7408, IC 7432, IC 7406, IC 7402, IC 7404, IC 7486

Theory:

Canonical Forms (Normal Forms): Any Boolean function can be written in disjunctive norm form (sum of min-terms) or conjunctive normal form (product of max-terms).A Boolean function c be represented by a Karnaugh map in which each cell corresponds to a min-term. The cells a arranged in such a way that any two immediately adjacent cells correspond to two min-terms distance 1. There is more than one way to construct a map with this property.

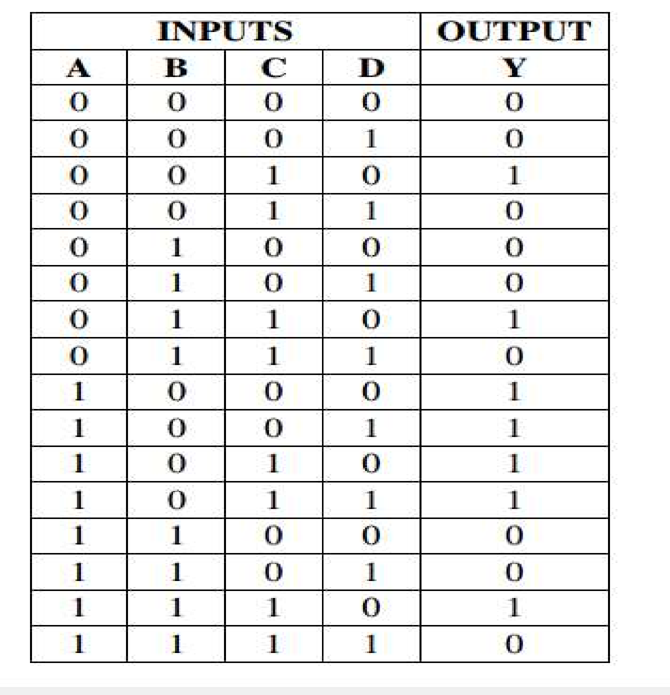
Truth Table:

1) $Y = \bar{A}\bar{B}C\bar{D} + \bar{A}BC\bar{D} + ABC\bar{D} + A\bar{B}C\bar{D} + A\bar{B}\bar{C}\bar{D} + A\bar{B}\bar{C}D + A\bar{B}CD$

AB

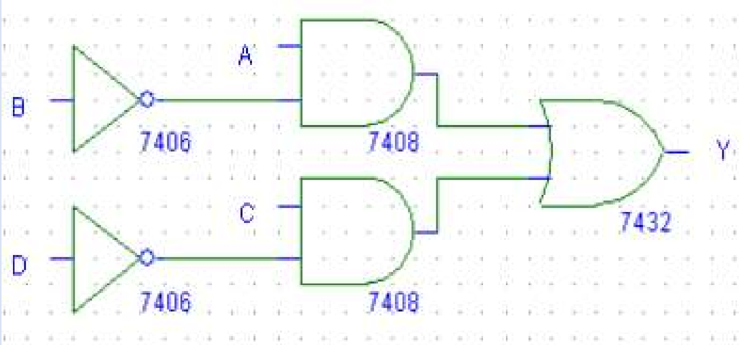
			1
			1
			1
1	1	1	1

After simplifying using K-Map method we get $Y = A\bar{B} + C\bar{D}$



INPUTS				OUTPUT
A	B	C	D	Y
0	0	0	0	0
0	0	0	1	0
0	0	1	0	1
0	0	1	1	0
0	1	0	0	0
0	1	0	1	0
0	1	1	0	1
0	1	1	1	0
1	0	0	0	1
1	0	0	1	1
1	0	1	0	1
1	0	1	1	1
1	1	0	0	0
1	1	0	1	0
1	1	1	0	1
1	1	1	1	0

Logic Diagram:



Procedure:

1. Check the components for their working.
2. Insert the appropriate IC into the IC base.
3. Make connections as shown in the circuit diagram.
4. Provide the input data via the input switches and observe the output on output LEDs

Discussion:

References:

1. Reference book.
2. Online sources.

23 **Experiment No.: 04**

Experiment No. 04

Name of the experiment: Construct and test various adder and subtractor circuits.

- **Objective:** To realize the adder and subtractor circuits using basic gates and universal gat
- To realize full adder using two half adders.
- To realize a full subtractor using two half subtractors.

Course Outcomes (COs), Program Outcomes (POs) and Assessment:

CO Statement	Corresponding PO	Domain / level of learning taxonomy	Delivery methods and activities	Assessment tools
CO2**:** Apply different IC's and logic gates for different digital logic circuits	**Individual Work and Teamwork (PO9)**	**Affective/Responding**	☑ Simulation ☑ Experiment ☑ Practice lab ☑ Group discussion ☑ Tutorial	☑ Lab tests ☑ Lab reports ☑ Final lab test □ Open ended lab □ Project show &

				project presentation

Components Required:

IC 7400, IC 7408, IC 7486, IC 7432, Patch Cords & IC Trainer Kit.

Theory:

Half-Adder: A combinational logic circuit that performs the addition of two data bits, A and B, is called a half-adder. Addition will result in two output bits; one of which is the sum bit, S, and the other is the carry bit, C. The Boolean functions describing the half-adder are:

$$\mathbf{S = A \oplus B \quad C = A\ B}$$

Full-Adder: The half-adder does not take the carry bit from its previous stage into account. This carry bit from its previous stage is called carry-in bit. A combinational logic circuit that adds two data bits, A and B, and a carry-in bit, Cin , is called a full-adder. The Boolean functions describing the full-adder are:

$$\mathbf{S = (x \oplus y) \oplus Cin\ C = xy + Cin\ (x \oplus y)}$$

Half Subtractor: Subtracting a single-bit binary value B from another A (i.e. A -B) produces a difference bit D and a borrow out bit B-out. This operation is called half subtraction and the circuit t realize it is called a half subtractor. The Boolean functions describing the half Subtractor are:

$$\mathbf{S = A \oplus B \quad C = A'\ B}$$

Full Subtractor: Subtracting two single-bit binary values, B, Cin from a single-bit value A produce a difference bit D and a borrow out Br bit. This is called full subtraction. The Boolean functions describing the full-subtractor are:

$$\mathbf{D = (x \oplus y) \oplus Cin \quad Br = A'B + A'(Cin) + B(Cin)}$$

Truth Table& Logic Diagram : To realize Half Adder

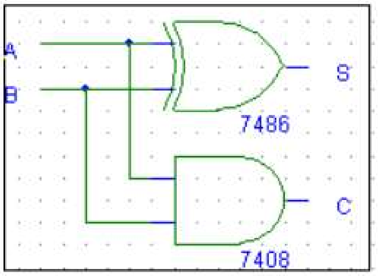
TRUTH TABLE

INPUTS		OUTPUTS	
A	**B**	**S**	**C**
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

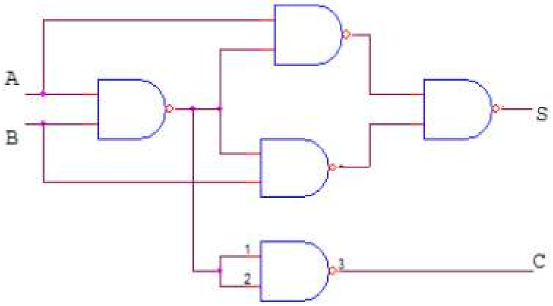
BOOLEAN EXPRESSIONS:

$S=A \oplus B$
$C=A\ B$

i) Basic Gates



ii) NAND Gates



Full Adder

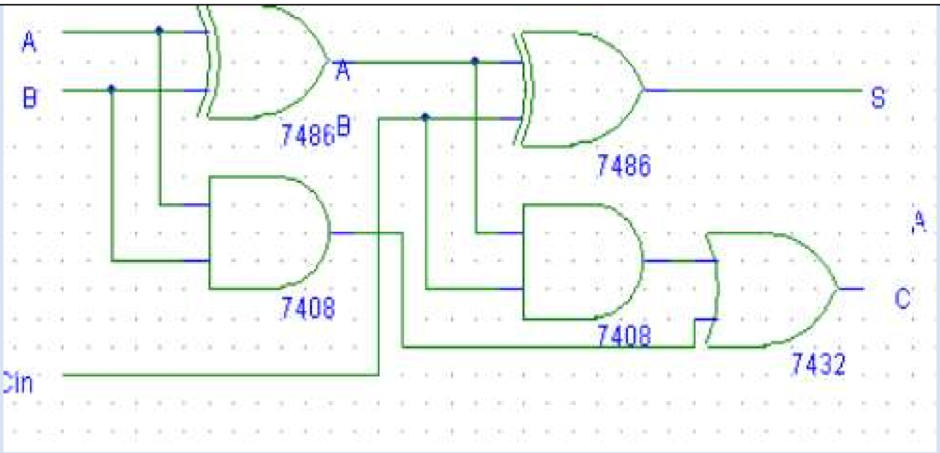
TRUTH TABLE

INPUTS			OUTPUTS	
A	**B**	**Cin**	**S**	**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

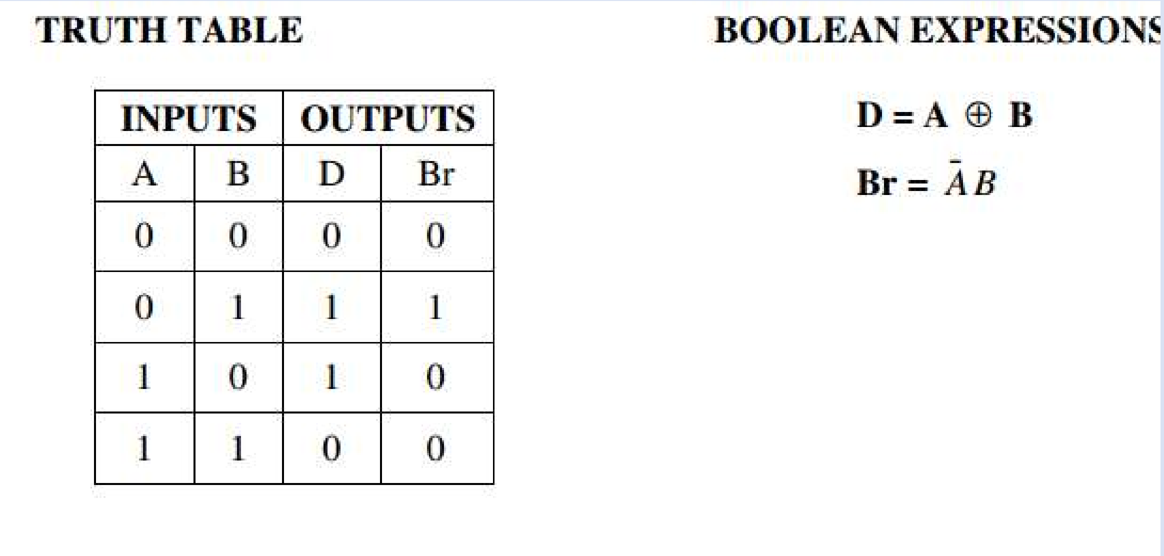
BOOLEAN EXPRESSIONS:

$S= A \oplus B \oplus C$

$C=A\ B + B\ Cin + A\ Cin$



HALF SUBTRACTOR



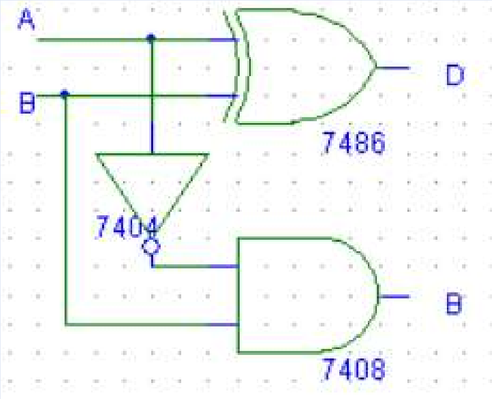
TRUTH TABLE

INPUTS		OUTPUTS	
A	B	D	Br
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0

BOOLEAN EXPRESSIONS

$$D = A \oplus B$$

$$Br = \bar{A}B$$



FULL SUBTRACTOR

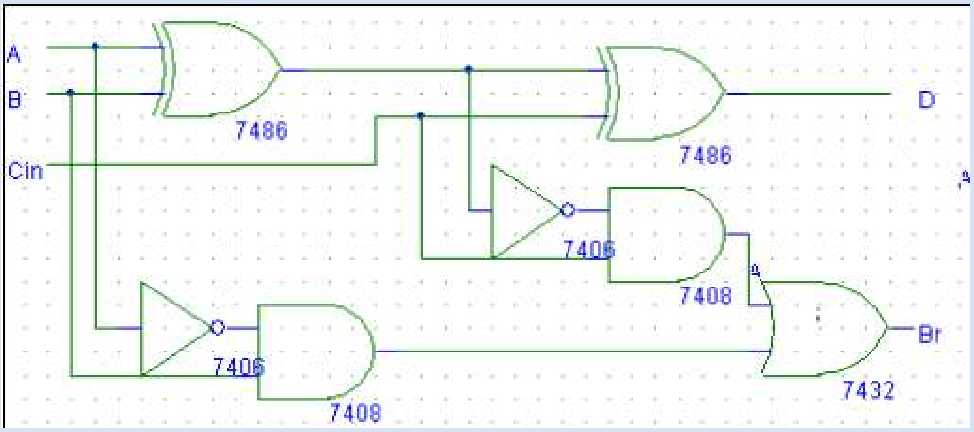
TRUTH TABLE

INPUTS			OUTPUTS	
A	B	Cin	D	Br
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 EXPRESSIONS:

$$D= A \oplus B \oplus C$$

$$Br= \bar{A}\ B + B\ Cin + \bar{A}\ Cin$$



Procedure:

1. Check the components for their working.
2. Insert the appropriate IC into the IC base.
3. Make connections as shown in the circuit diagram.
4. Provide the input data via the input switches and observe the output on output LEDs

Discussion:

References:

1. Reference book.
2. Online sources.

24 **Experiment No.: 05**

Experiment No. 05

Name of the experiment: Parity generator and Checker.

Objective: To design and verify the truth table of a three-bit odd parity generator and checker.

Course Outcomes (COs), Program Outcomes (POs) and Assessment:

CO Statement	Corresponding PO	Domain / level of learning taxonomy	Delivery methods and activities	Assessment tools
CO2: Apply different IC's and logic gates for different digital logic circuits	**Individual Work and Teamwork (PO9)**	**Affective/Responding**	☑ Simulation ☑ Experiment ☑ Practice lab ☑ Group discussion ☑ Tutorial	☑ Lab tests ☑ Lab reports ☑ Final lab test □ Open ended lab □ Project show & project presentation

COMPONENTS REQUIRED:

IC Trainer kit, ICS 7400, IC 7486

THEORY:

A parity bit is used for the purpose of detecting errors during transmission of binary information.
parity bit is an extra bit included with a binary message to make the number of 1's either odd or eve
The message including the parity bit is transmitted and then checked at the receiving end for errors. A
error is detected if the checked parity does not correspond with the one transmitted. The circuit th
generates the parity bit in the transmitter is called a parity generator and the circuit that checks the pari
in the receiver is called a parity checker. In even parity the added parity bit will make the total numb
of 1's an even amount and in odd parity the added parity bit will make the total number of 1's an o
amount. In a three-bit odd parity generator the three bits in the message together with the parity bit a
transmitted to their destination, where they are applied to the parity checker circuit. The parity check
circuit checks for possible errors in the transmission. Since the information was transmitted with o
parity the four bits received must have an odd number of 1's. An error occurs during the transmissio
if the four bits received have an even number of 1's, indicating that one bit has changed durin
transmission. The output of the parity checker is denoted by PEC (parity error check) and it will
equal to 1 if an error occurs, i.e., if the four bits received has an even number of 1's.

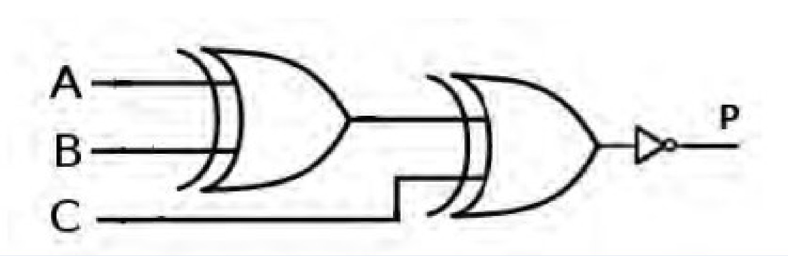
TRUTH TABLE:

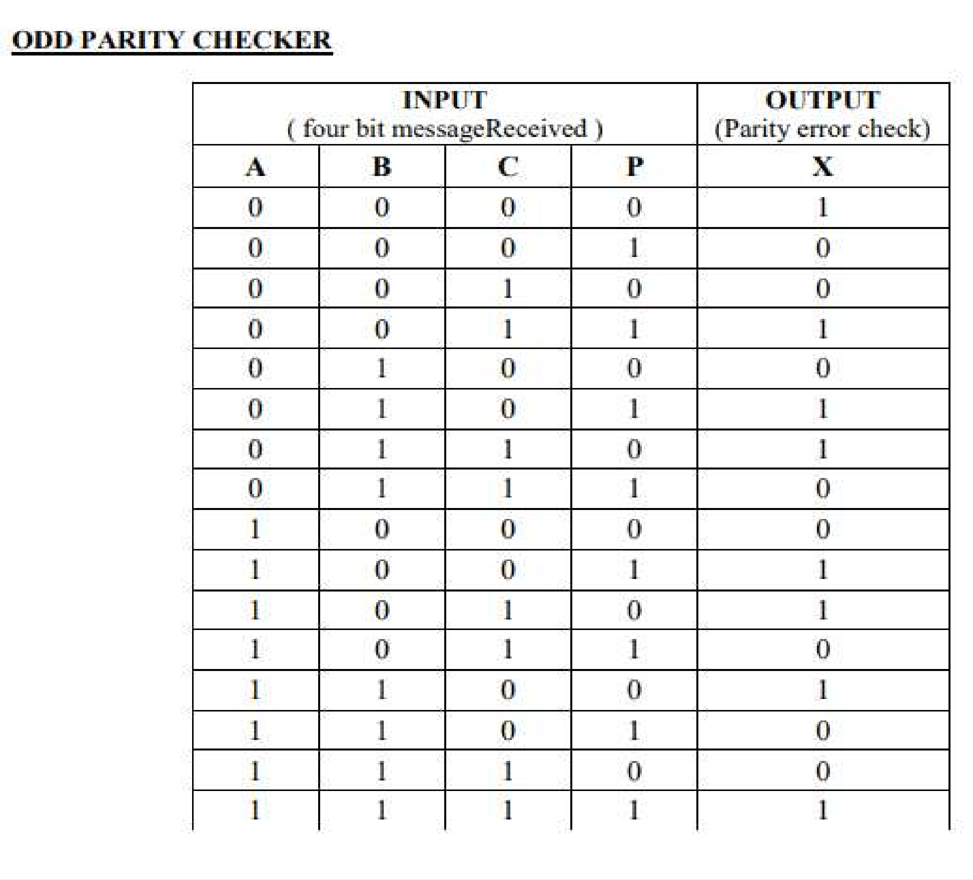
INPUTs (Three bit messages)			OUTPUTs (Odd Parity bit)
A	**B**	**C**	**P**
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	0

From the truth table the expression for the output parity bit is, P (A, B, C) = Σ m (0, 3, 5, 6) Also written as,

$$P = A'B'C' + A'BC + AB'C + ABC'$$
$$P = (A \oplus B \oplus C)'$$

CIRCUIT DIAGRAM: Odd Parity Generator





ODD PARITY CHECKER

INPUT (four bit messageReceived)				OUTPUT (Parity error check)
A	**B**	**C**	**P**	**X**
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

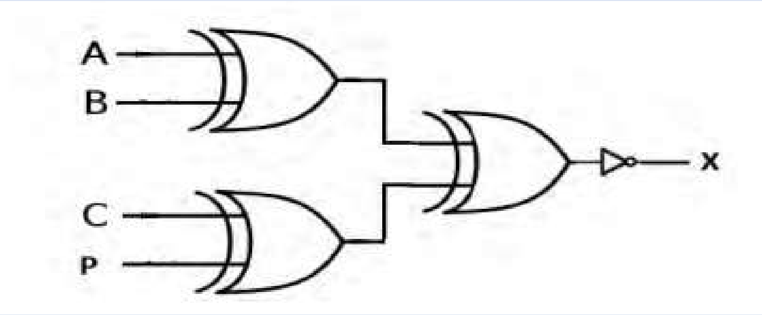
From the truth table the expression for the output parity checker bit is,

$$X\ (A, B, C, P)\ =\ \Sigma\ (0, 3, 5, 6, 9, 10, 12, 15)$$

The above expression is reduced as,

$$X = (A \oplus B \oplus C \oplus P)'$$

CIRCUIT DIAGRAM: Odd Parity Checker



PROCEDURE:

- Connections are given as per the circuit diagrams.
- For all the ICs 7th pin is grounded and 14th pin is given +5 V supply.
- Apply the inputs and verify the truth table for the Parity generator and checker.

DISCUSSION:

The design of the three-bit odd Parity generator and checker circuits was done and their truth tabl were verified.

.

References:

1. Reference book.
2. Online sources.

25 **Experiment No.: 06**

Experiment No. 06

Name of the Experiment: BCD TO EXCESS- 3 CODE CONVERTERS.

Objective: To learn to realize BCD to Excess-3 code using adder IC 7483. To learn to realize Excess-3 to BC Code using adder IC 7483.

Course Outcomes (COs), Program Outcomes (POs) and Assessment:

CO Statement	Corresponding PO	Domain / level of learning taxonomy	Delivery methods and activities	Assessment tools
CO2**:** Apply different IC's and logic gates for different digital logic circuits	**Individual Work and Teamwork (PO9)**	**Affective/Responding**	☑ Simulation ☑ Experiment ☑ Practice lab ☑ Group discussion ☑ Tutorial	☑ Lab tests ☑ Lab reports ☑ Final lab test □ Open ended lab □ Project show & project presentation

COMPONENTS REQUIRED:

IC 7483, IC 7486, Patch Cords & IC Trainer Kit.

THEORY:

Code converter is a combinational circuit that translates the input code word into a new correspondi word. The excess-3 code digit is obtained by adding three to the corresponding BCD digit. To Constru a BCD-to-excess-3-code converter with a 4-bit adder feed BCD code to the 4-bit adder as the fi operand and then feed constant 3 as the second operand. The output is the corresponding excess-3 coc To make it work as an excess-3 to BCD converter, we feed excess-3 code as the first operand and th feed 2's complement of 3 as the second operand. The output is the BCD code.

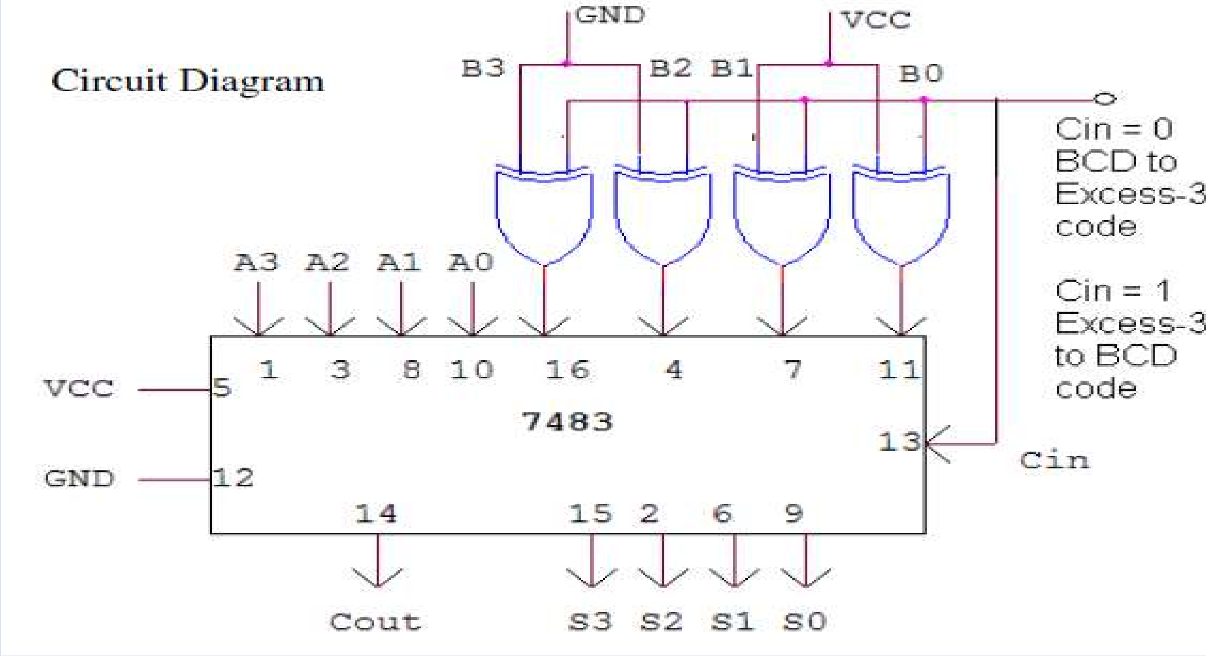
TRUTH TABLE:

i) BCD - EXCESS-3 CODE

BCD				Excess-3			
0	0	0	0	0	0	1	1
0	0	0	1	0	1	0	0
0	0	1	0	0	1	0	1
0	0	1	1	0	1	1	0
0	1	0	0	0	1	1	1
0	1	0	1	1	0	0	0
0	1	1	0	1	0	0	1
0	1	1	1	1	0	1	0
1	0	0	0	1	0	1	1
1	0	0	1	1	1	0	0

ii) EXCESS-3 – BCD CODE

Excess-3				BCD			
0	0	1	1	0	0	0	0
0	1	0	0	0	0	0	1
0	1	0	1	0	0	1	0
0	1	1	0	0	0	1	1
0	1	1	1	0	1	0	0
1	0	0	0	0	1	0	1
1	0	0	1	0	1	1	0
1	0	1	0	0	1	1	1
1	0	1	1	1	0	0	0
1	1	0	0	1	0	0	1



PROCEDURE:

- Check all the components for their working.
- Insert the appropriate IC into the IC base.
- Make connections as shown in the circuit diagram.
- Apply BCD code as first operand(A) and binary 3 as second operand(B) and cin=0 fo *Realizing BCD-to-Excess-3-code:*
- Apply Excess-3-code code as first operand (A) and binary 3 as second operand (B) ar Cin=1 for realizing Excess-3-code to BCD.
- Verify the Truth Table and observe the outputs.

DISCUSSION:

Realized BCD code to Excess-3 code conversion and vice versa using 7483 IC

References:

1. Reference book.
2. Online sources.

26 **Experiment No.: 07**

Experiment No. 07

Name of the Experiment: To check the operation of 2-to-4 line Decoder.

Objective: To learn about working principle of decoder. To learn and understand the working of IC 74LS13

To realize using basic gates as well as universal gates.

Course Outcomes (COs), Program Outcomes (POs) and Assessment:

CO Statement	Corresponding PO	Domain / level of learning taxonomy	Delivery methods and activities	Assessment tools

CO2:Apply different IC's and logic gates for different digital logic circuits	**Individual Work and Teamwork (PO9)**	**Affective/Responding**	☑ Simulation ☑ Experiment ☑ Practice lab ☑ Group discussion ☑ Tutorial	☑ Lab tests ☑ Lab reports ☑ Final lab test ☐ Open ended lab ☐ Project show & project presentation

Components Required:

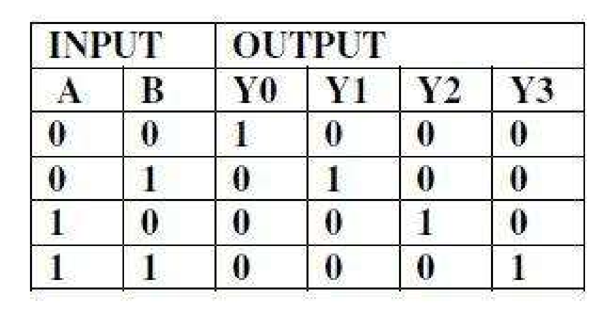
- IC74LS139, IC 7400, IC 7408, IC 7432, IC 7404, IC 7410.
- Patch chords.
- IC Trainer Kit.

Theory:

A decoder is a combinational circuit that connects the binary information from 'n' input lines to maximum of 2^n unique output lines. Decoder is also called a min-term generator/max-term generato A min-term generator is constructed using AND and NOT gates. The appropriate output is indicat by logic 1 (positive logic). Max-term generator is constructed using NAND gates. The appropria output is indicated by logic 0 (Negative logic).
The IC 74139 accepts two binary inputs and when enable provides 4 individual active low outputs. T device has 2 enable inputs (Two active low).

- **2:4 DECODER (MIN TERM GENERATOR):**

Truth Table:



INPUT		OUTPUT			
A	B	Y0	Y1	Y2	Y3
0	0	1	0	0	0
0	1	0	1	0	0
1	0	0	0	1	0
1	1	0	0	0	1

Boolean Expression:

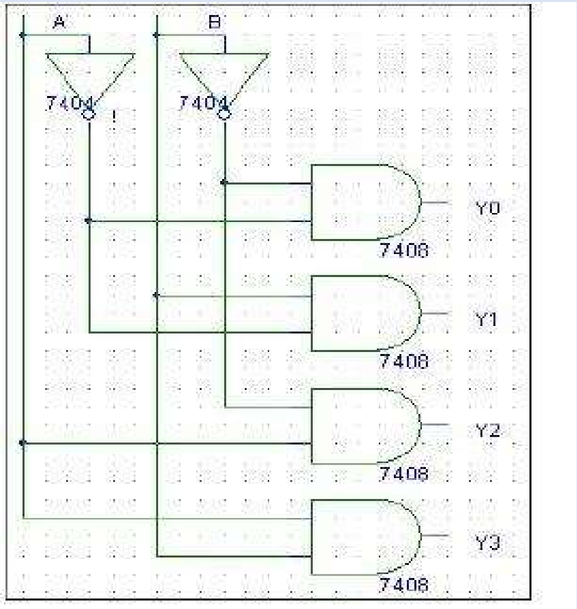
$$Y0 = \overline{A}\overline{B}$$

$$Y1 = \overline{A}B$$

$$Y2 = A\overline{B}$$

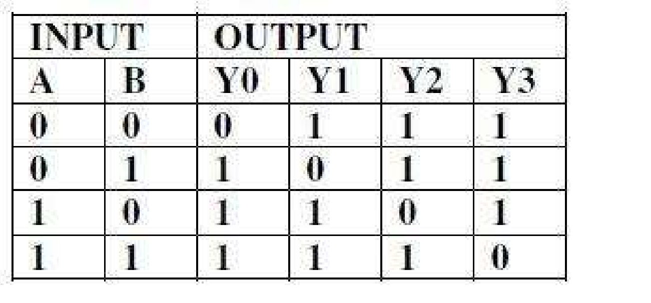
$$Y3 = AB$$

Circuit Diagram:



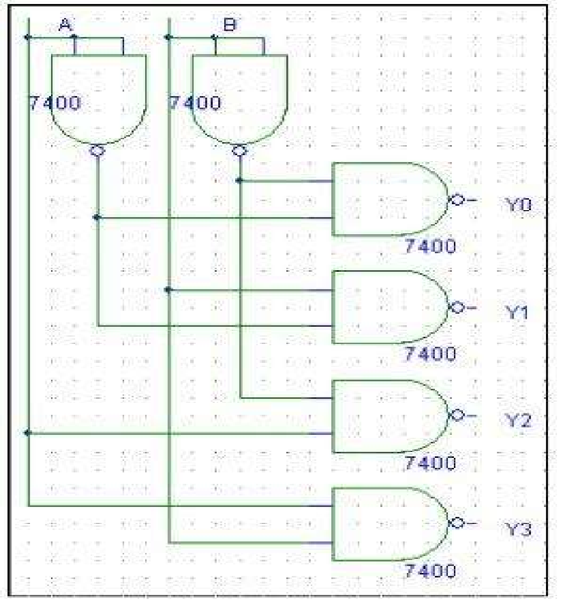
- **2:4 DECODER (MAX TERM GENERATOR):**

Truth Table:



INPUT		OUTPUT			
A	B	Y0	Y1	Y2	Y3
0	0	0	1	1	1
0	1	1	0	1	1
1	0	1	1	0	1
1	1	1	1	1	0

Circuit Diagram:



Procedure:

- Make the connections as per the circuit diagram.
- Change the values of G1, G2A, G2B, A, B, and C, using switches.
- Observe status of Y0, to Y7 on LED's.
- Verify the truth table.

Discussions:

Verified the Operation of 2 to 4 Decoder.

References:

1. Reference book.
2. Online sources.
1. .

27 **Experiment No.: 08**

Experiment No. 08

Name of the Experiment: Using the 7447 IC, design a BCD (Binary Code Decimal) to 7-SEGMENT Decoder.

Objective: To learn about various applications of decoder. To learn about types of seven-segment displa

Course Outcomes (COs), Program Outcomes (POs) and Assessment:

CO Statement	Corresponding PO	Domain / level of learning taxonomy	Delivery methods and activities	Assessment tools
CO2:Apply different IC's and logic gates for different digital logic circuits	**Individual Work and Teamwork (PO9)**	**Affective/Responding**	☑ Simulation ☑ Experiment	☑ Lab tests ☑ Lab reports

			☑ Practice lab ☑ Group discussion ☑ Tutorial	☑ Final lab test □ Open ended lab ☑ Project show & project presentation

Components Required:

- IC7447.
- 7-Segment display (common anode).
- Patch chords, resistor (1K).
- IC Trainer Kit.

Theory:

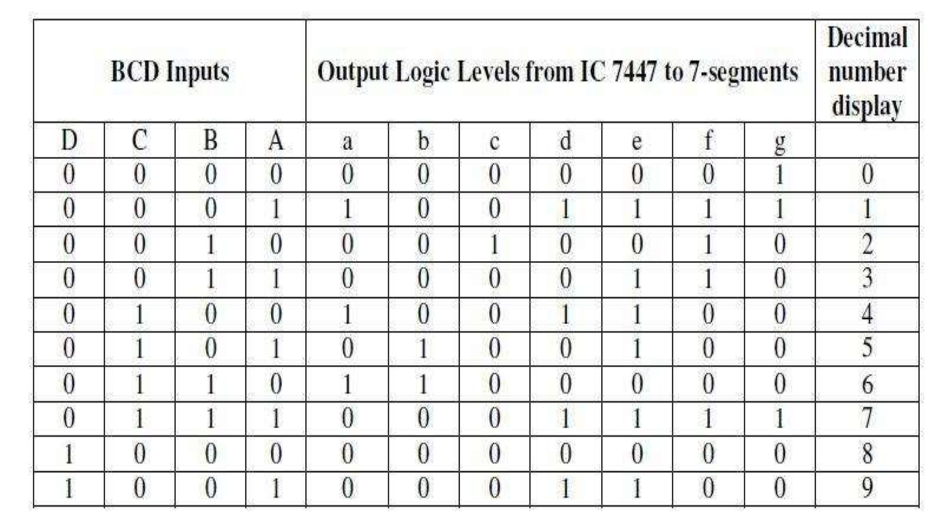
The Light Emitting Diode (LED) finds its place in many applications in these modern electronic fielc One of them is the Seven Segment Display. Seven-segment displays contains the arrangement of tl LEDs in “Eight” (8) passion, and a Dot (.) with a common electrode, lead (Anode or Cathode). Tl purpose of arranging it in that passion is that we can make any number out of that by switching ON ar OFF the particular LED's. Here is the block diagram of the Seven Segment LED arrangement.

LED’s are basically of two types:

- Common Cathode (CC) -All the 8 anode legs uses only one cathode, which is common.
- Common Anode (CA)-The common leg for all the cathode is of Anode type.

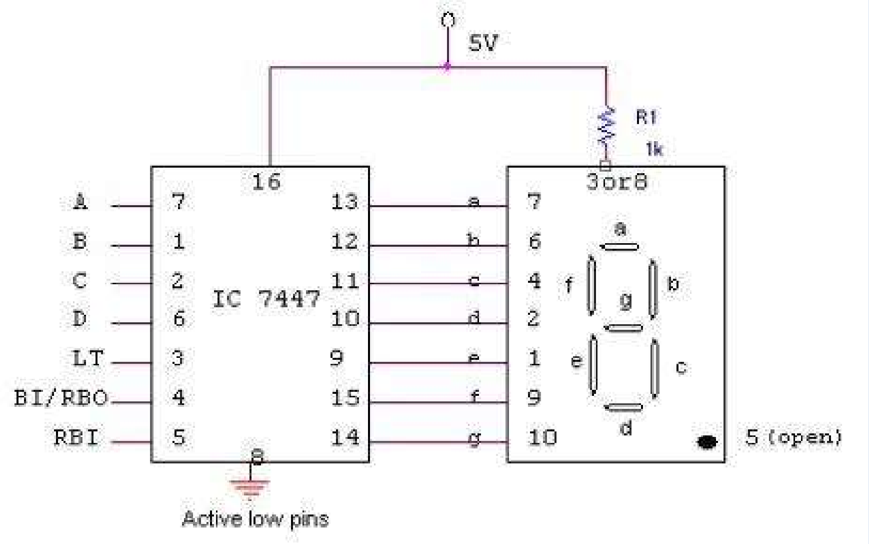
A decoder is a combinational circuit that connects the binary information from ‘n’ input lines to maximum of 2n unique output lines. The IC7447 is a BCD to 7-segment pattern converter. The IC744 takes the Binary Coded Decimal (BCD) as the input and outputs the relevant 7 segment code.

Truth Table:



BCD Inputs				Output Logic Levels from IC 7447 to 7-segments							Decimal number display
D	C	B	A	a	b	c	d	e	f	g	
0	0	0	0	0	0	0	0	0	0	1	0
0	0	0	1	1	0	0	1	1	1	1	1
0	0	1	0	0	0	1	0	0	1	0	2
0	0	1	1	0	0	0	0	1	1	0	3
0	1	0	0	1	0	0	1	1	0	0	4
0	1	0	1	0	1	0	0	1	0	0	5
0	1	1	0	1	1	0	0	0	0	0	6
0	1	1	1	0	0	0	1	1	1	1	7
1	0	0	0	0	0	0	0	0	0	0	8
1	0	0	1	0	0	0	1	1	0	0	9

Circuit Diagram:



Procedure:

- Check all the components for their working.
- Insert the appropriate IC into the IC base.
- Make connections as shown in the circuit diagram.
- Verify the Truth Table and observe the outputs.

Discussions:

It is possible to display the decimal value of a binary number on a 7-segment display using a BC decoder. However, this method will allow displaying only digits from 0 to 9 and letters A to F. T only way to display number more than 9 is to use a display that has more than 7 segments or just usi multiple 7-segment displays at once with the corresponding BCD decoder.

References:

3. Reference book.
4. Online sources.

28 **Experiment No.: 09**

Experiment No. 09

Name of the experiment: Designing of Decimal to BCD Encoder and Octal to Binary Encoder.

Objective: To learn about various applications of Encoders. To learn and understand the working IC 74147 and IC 74148. To learn to do code conversion using encoders.

Course Outcomes (COs), Program Outcomes (POs) and Assessment:

CO Statement	Corresponding PO	Domain / level of learning taxonomy	Delivery methods and activities	Assessment tools
CO2:Apply different IC's and logic gates for different	**Individual Work and Teamwork (PO9)**	**Affective/Responding**	☑ Simulation ☑ Experiment	☑ Lab tests ☑ Lab reports

digital logic circuits			☑ Practice lab ☑ Group discussion ☑ Tutorial	☑ Final lab test □ Open ended lab □ Project show & project presentation

Components Required:

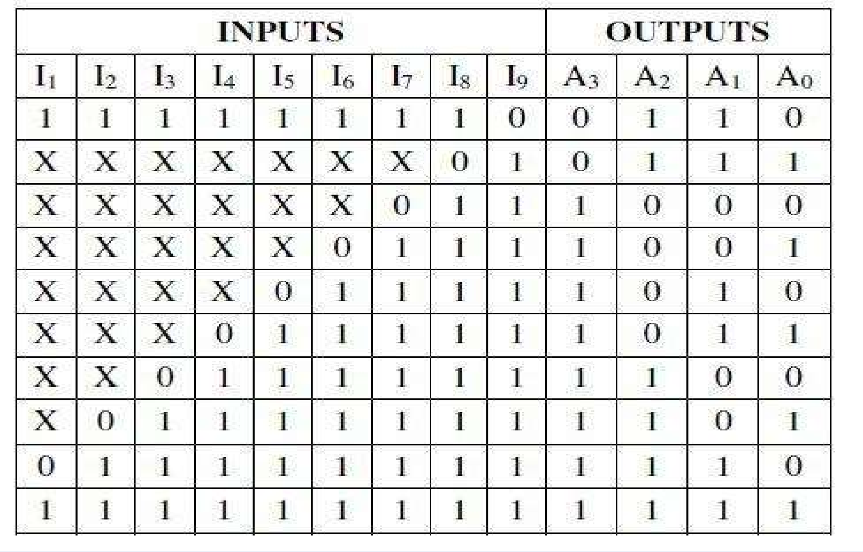
- IC 74147.
- IC 74148.
- Patch chords.
- IC Trainer Kit.

Theory:

An encoder performs a function that is the opposite of decoder. It receives one or more signals in encoded format and output a code that can be processed by another logic circuit. One of the advantag of encoding data, or more often data addresses in computers, is that it reduces the number of requir bits to represent data or addresses. For example, if a memory has 16 different locations, in order access these 16 different locations, 16 lines (bits) are required if the addressing signals are in 1 out *n* format. However, if we code the 16 different addresses into a binary format, then only 4 lines (bit are required. Such a reduction improves the speed of information processing in digital systems.

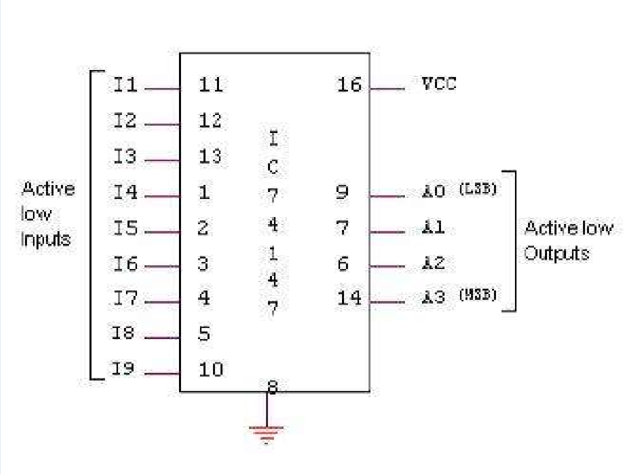
- **Decimal to BCD Encoder**

Truth Table:



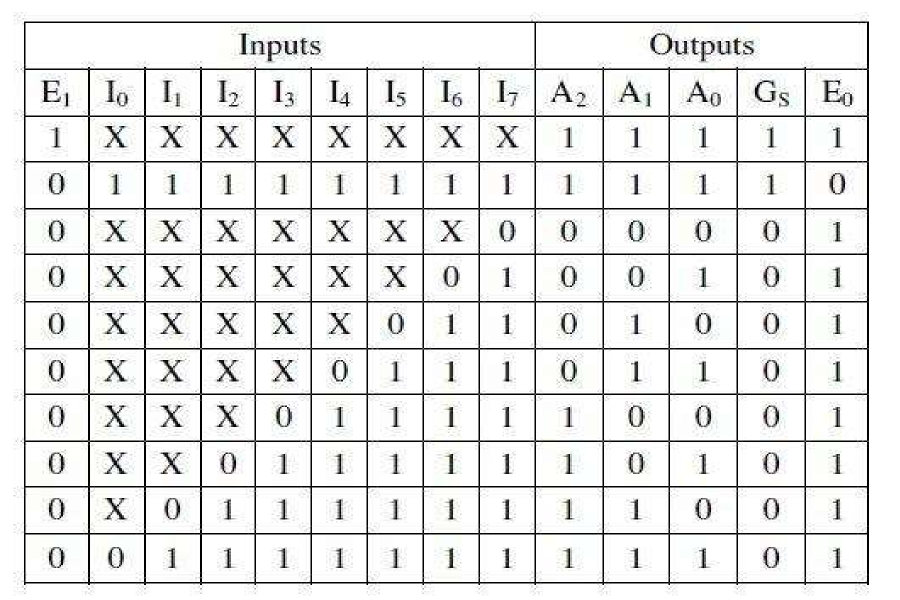
INPUTS									OUTPUTS			
I_1	I_2	I_3	I_4	I_5	I_6	I_7	I_8	I_9	A_3	A_2	A_1	A_0
1	1	1	1	1	1	1	1	0	0	1	1	0
X	X	X	X	X	X	X	0	1	0	1	1	1
X	X	X	X	X	X	0	1	1	1	0	0	0
X	X	X	X	X	0	1	1	1	1	0	0	1
X	X	X	X	0	1	1	1	1	1	0	1	0
X	X	X	0	1	1	1	1	1	1	0	1	1
X	X	0	1	1	1	1	1	1	1	1	0	0
X	0	1	1	1	1	1	1	1	1	1	0	1
0	1	1	1	1	1	1	1	1	1	1	1	0
1	1	1	1	1	1	1	1	1	1	1	1	1

Circuit Diagram:



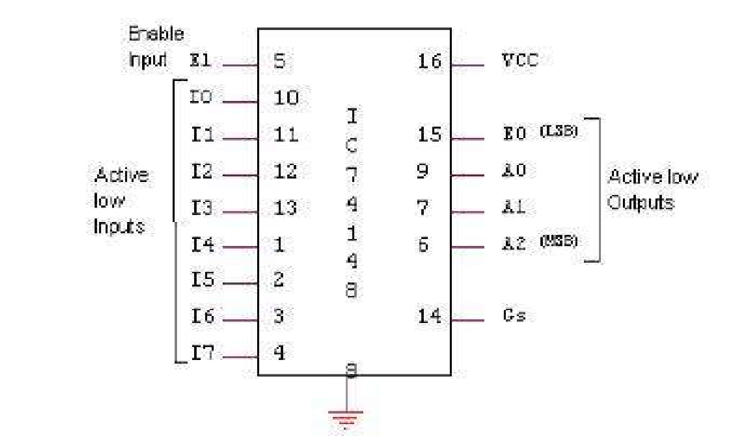
- **Octal to Binary Encoder**

Truth Table:



Inputs									Outputs				
E_1	I_0	I_1	I_2	I_3	I_4	I_5	I_6	I_7	A_2	A_1	A_0	G_S	E_0
1	X	X	X	X	X	X	X	X	1	1	1	1	1
0	1	1	1	1	1	1	1	1	1	1	1	1	0
0	X	X	X	X	X	X	X	0	0	0	0	0	1
0	X	X	X	X	X	X	0	1	0	0	1	0	1
0	X	X	X	X	X	0	1	1	0	1	0	0	1
0	X	X	X	X	0	1	1	1	0	1	1	0	1
0	X	X	X	0	1	1	1	1	1	0	0	0	1
0	X	X	0	1	1	1	1	1	1	0	1	0	1
0	X	0	1	1	1	1	1	1	1	1	0	0	1
0	0	1	1	1	1	1	1	1	1	1	1	0	1

Circuit Diagram:



Procedure:

- Check all the components for their working.
- Insert the appropriate IC into the IC base.

- Make connections as shown in the circuit diagram.
- Verify the Truth Table and observe the outputs.

Discussions:

References:

1. Reference book.
2. Online sources.

29 **Experiment No.: 10**

Experiment No. 10

Name of the Experiment: To design and set up the following circuit

1. Designing of 2 to 1 , 4 to 1 line multiplexer
2. Designing a quadruple 2 to 1 line MUX
3. Designing 1 to 4 line de-multiplexer.
4.

Objective: To learn about various applications of multiplexer and de-multiplexer. To learn a understand the working of IC 74153 and IC 74139. To learn to realize any function using Multiplexe

Course Outcomes (COs), Program Outcomes (POs) and Assessment:

CO Statement	Corresponding PO	Domain / level of learning taxonomy	Delivery methods and activities	Assessment tools
CO3: Design and implement combinational circuit.	**Design of Solutions (PO3)**	**Psychomotor/Guided response**	☑ Simulation ☑ Experiment ☑ Practice lab ☑ Group discussion ☑ Tutorial	☑ Lab tests ☑ Lab reports ☑ Final lab test □ Open ended lab □ Project show & project presentation

Components Requirements:

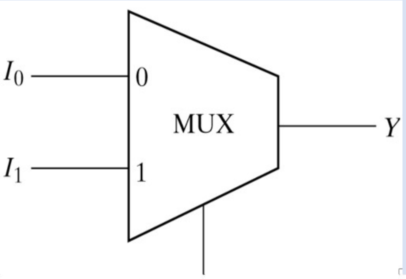
- AND gate (7408), OR gate (7432), NOT gate (7404), IC trainer kit, Wires.

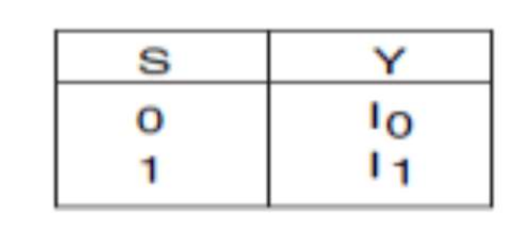
Theory:

Multiplexers are very useful components in digital systems. They transfer a large numb of information units over a smaller number of channels, (usually one channel) under the control selection signals. Multiplexer means many to one. A multiplexer is a circuit with many inputs but on one output. By using control signals (select lines) we can select any input to the output. Multiplexer also called as data selector because the output bit depends on the input data bit that is selected. T general multiplexer circuit has 2^n input signals, n control/select signals and 1 output signal.

De-multiplexers perform the opposite function of multiplexers. They transfer a small number information units (usually one unit) over a larger number of channels under the control of selecti signals. The general de-multiplexer circuit has 1 input signal, n control/select signals and 2^n outp signals. De-multiplexer circuit can also be realized using a decoder circuit with enable.

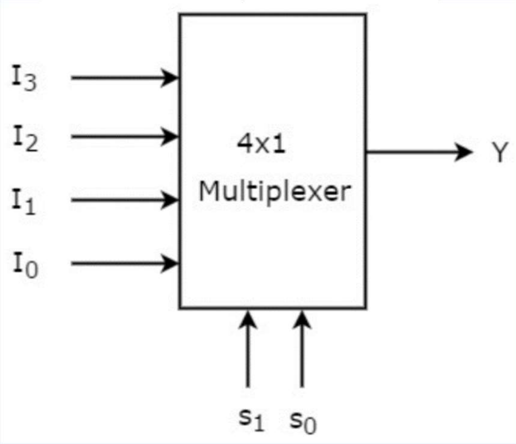
Block Diagram & Truth table:

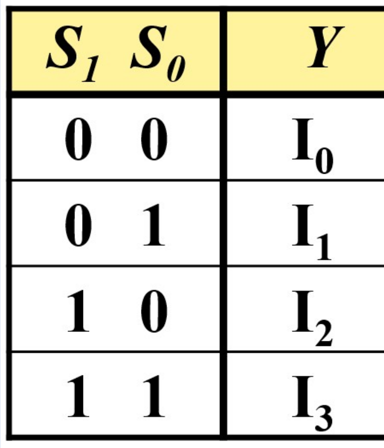




S	Y
0	I_0
1	I_1

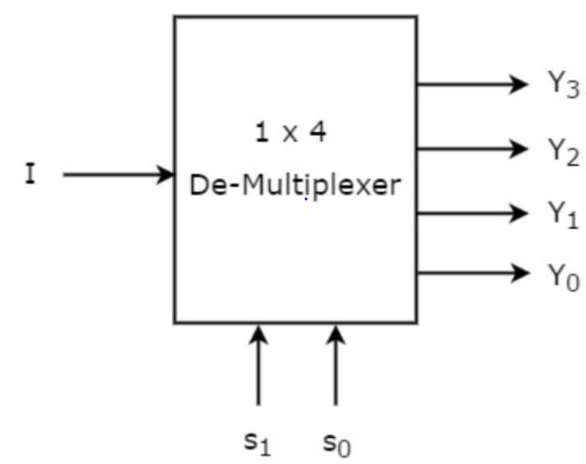
Figure: Block diagram and truth table of 2 to 1 MUX

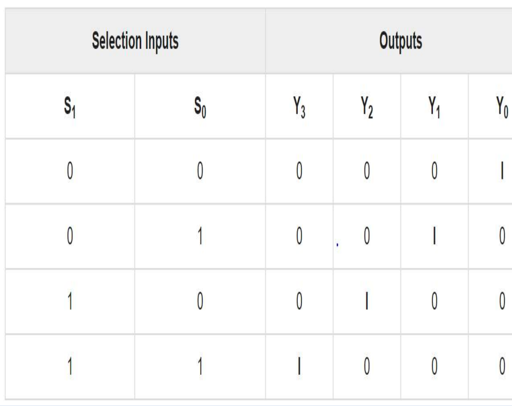




S_1 S_0	Y
0 0	I_0
0 1	I_1
1 0	I_2
1 1	I_3

Figure: Block diagram and truth table of 4 to 1 MUX





Selection Inputs		Outputs			
S_1	S_0	Y_3	Y_2	Y_1	Y_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

Figure: Block diagram and truth table of 1 to 4 Demultiplexer

Logic Diagram:

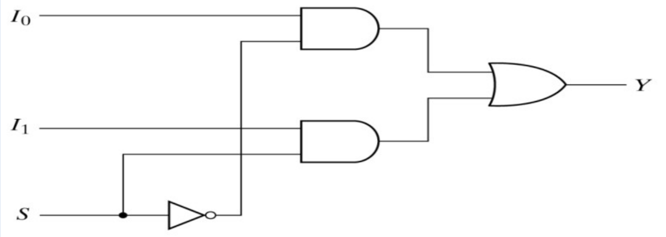


Figure: Logic diagram of 2 to 1 MUX

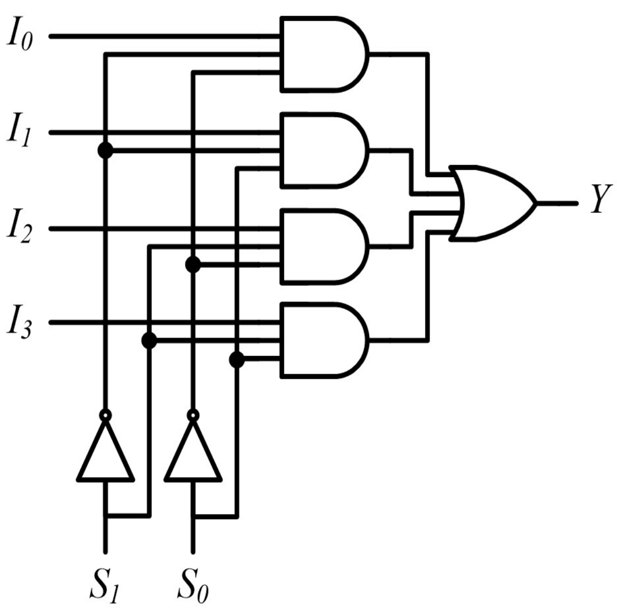


Figure: Logic diagram of 4 to 1 MUX

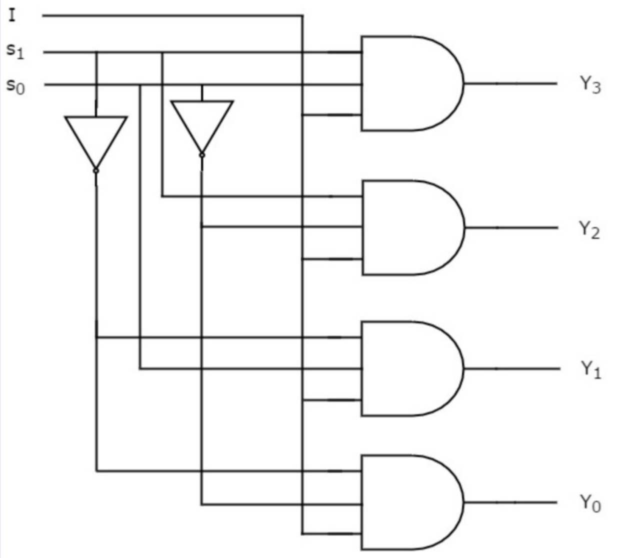


Figure: Logic diagram of 1 to 4 Demux

Procedure:

1. Check all the components for their working.
2. Insert the appropriate IC into the IC base.
3. Make connections as shown in the circuit diagram.
4. Verify the Truth Table and observe the outputs.

Discussion:

1. Supply should not exceed 5v.
2. Connections should be tight and easy to inspect.
3. Use L.E.D. with proper sign convention and check it before connecting in circuit.

References:

1. Reference book.
2. Online sources.

30 **Experiment No.: 11**

Experiment No. 11

Name of the Experiment: Designing of 4-BIT Magnitude comparator and magnitude comparis

using 74LS85 IC.

Objective: To learn about various applications of comparator. To learn and understand the workin

of IC 7485 magnitude comparator.

Course Outcomes (COs), Program Outcomes (POs) and Assessment:

CO Statement	Corresponding PO	Domain / level of learning taxonomy	Delivery methods and activities	Assessment tools
CO3:Design and implement combinational circuit.	**Design of Solutions (PO3)**	**Psychomotor/Guided response**	☑ Simulation ☑ Experiment ☑ Practice lab ☑ Group discussion ☑ Tutorial	☑ Lab tests ☑ Lab reports ☑ Final lab test □ Open ended lab □ Project show & project presentation

Components Requirements:

- AND gate (7408), OR gate (7432), NOT gate (7404), NOR gate (7402), Magnitude comparat (7485) IC trainer kit, Wires.

Theory:

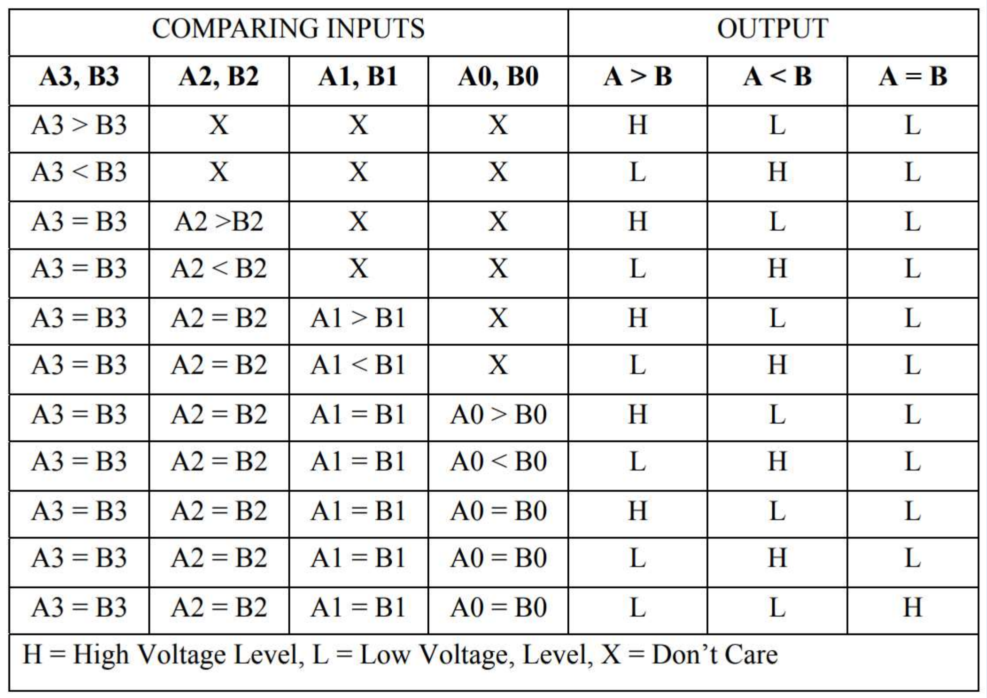
Magnitude Comparator is a logical circuit, which compares two signals A and B and generates three logical outpu whether A > B, A = B, or A < B. IC 7485 is a high speed 4-bit Magnitude comparator, which compares two 4- words. The A = B Input must be held high for proper compare operation.

Truth table:

Input: A=A3A2A1A0
B=B3B2B1B0

(1)A= B :
A3=B3, A2=B2, A1=B1, A0=B0
xi = AiBi + Ai’Bi’
XOR-Invert = (AiBi’+Ai’Bi)’
= (Ai’+Bi)(Ai+Bi’)
= Ai’Ai + Ai’Bi’ + AiBi + BiBi’
= AiBi + Ai’Bi’
Output: x3x2x1x0 6
(2)A> B
Output: A3B’3 + x3A2B’2 + x3x2A1B’1+ x3x2x1A0B’0

(3)(A< B)
Output: A’3B3 + x3A’2B2 + x3x2A’1B1+ x3x2x1A’0B0



COMPARING INPUTS				OUTPUT		
A3, B3	**A2, B2**	**A1, B1**	**A0, B0**	**A > B**	**A < B**	**A = B**
A3 > B3	X	X	X	H	L	L
A3 < B3	X	X	X	L	H	L
A3 = B3	A2 >B2	X	X	H	L	L
A3 = B3	A2 < B2	X	X	L	H	L
A3 = B3	A2 = B2	A1 > B1	X	H	L	L
A3 = B3	A2 = B2	A1 < B1	X	L	H	L
A3 = B3	A2 = B2	A1 = B1	A0 > B0	H	L	L
A3 = B3	A2 = B2	A1 = B1	A0 < B0	L	H	L
A3 = B3	A2 = B2	A1 = B1	A0 = B0	H	L	L
A3 = B3	A2 = B2	A1 = B1	A0 = B0	L	H	L
A3 = B3	A2 = B2	A1 = B1	A0 = B0	L	L	H
H = High Voltage Level, L = Low Voltage, Level, X = Don’t Care						

Figure: Block diagram and truth table of 2 to 1 MUX

Logic Diagram:

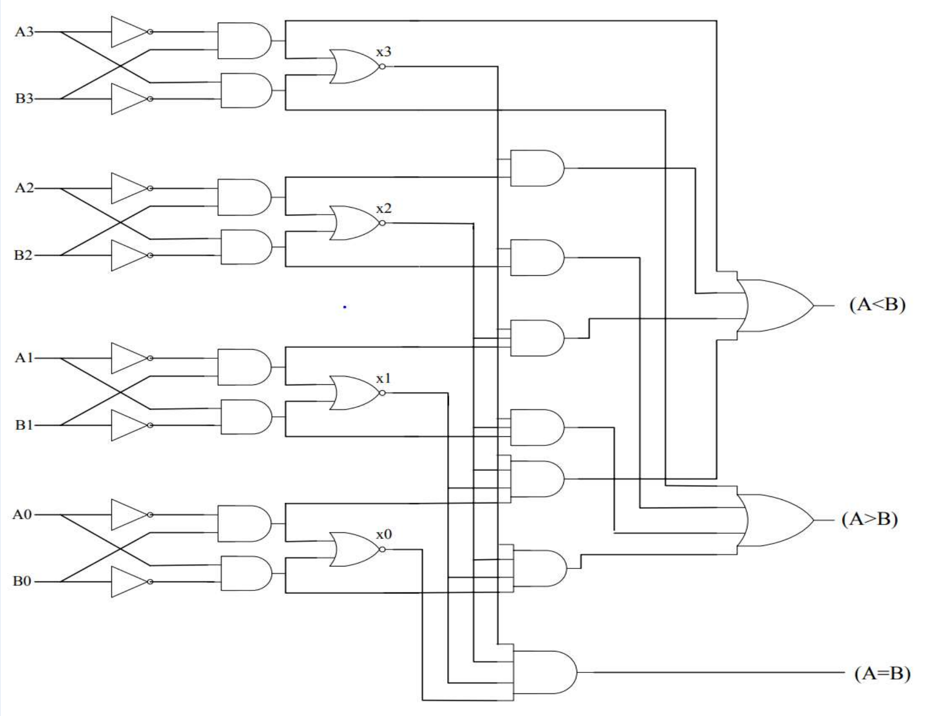


Figure: Logic diagram of 4 bit magnitude comparator

Procedure:

5. Check all the components for their working.
6. Insert the appropriate IC into the IC base.
7. Make connections as shown in the circuit diagram.
8. Verify the Truth Table and observe the outputs.

Discussion:

4. Supply should not exceed 5v.
5. Connections should be tight and easy to inspect.
6. Use L.E.D. with proper sign convention and check it before connecting in circuit.
7. Four bit comparators are verified using basic gates and magnitude comparator IC7485.

References:

1. Reference book.
2. Online sources.

31 **Experiment No.: 12**

Experiment No. 12

Name of the Experiment: Construct, test and investigate the operation of various types of Latches.

Objective: To study and verify the truth table of logic gates of different types of Latches and identi various ICs and their specification

Course Outcomes (COs), Program Outcomes (POs) and Assessment:

CO Statement	Corresponding PO	Domain / level of learning taxonomy	Delivery methods and activities	Assessment tools
CO3: Design and implement combinational and sequential circuit.	**Design of Solutions (PO3)**	**Psychomotor/Guided response**	☑ Simulation ☑ Experiment ☑ Practice lab ☑ Group discussion ☑ Tutorial	☑ Lab tests ☑ Lab reports ☑ Final lab test ☐ Open ended lab ☐ Project show & project presentation

Components Required:

Logic gates (IC) trainer kit. Digital Lab Kit, Breadboard, Power Supplier, Connecting patch chords, I 7400, IC 7408, IC 7432, IC 7406, IC 7402, IC 7404, IC 7486

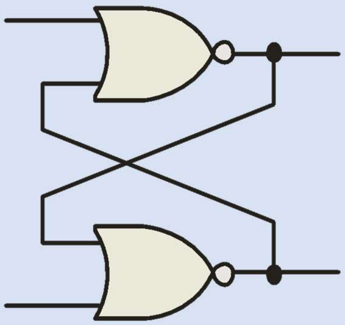
Theory:

A latch is a temporary storage device that has two stable states (bi-stable). It is a basic form of memor The S-R (Set-Reset) latch is the most basic type. It cannot be constructed from NOR gates or NAN gates. With NOR gates, the latch responds to active-HIGH inputs; with NAND gates, it responds active-LOW inputs.

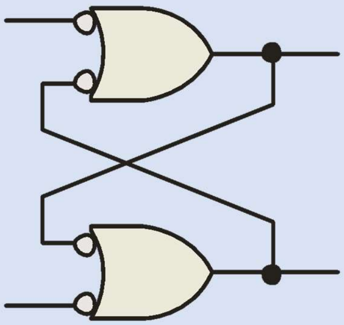
SR Latch:

An S-R latch neither consists of two cross-coupled NOR gates. An S-R flip-flop can also be design Using cross-coupled NAND gates as shown. The truth tables of the circuits are shown below. A clock S-R flip-flop has an additional clock input so that the S and R inputs are active only when the clock high. When the clock goes low, the state of flip-flop is latched and cannot change until the clock go high again. Therefore, the clocked S-R flip-flop is also called “enabled” S-R flip-flop.

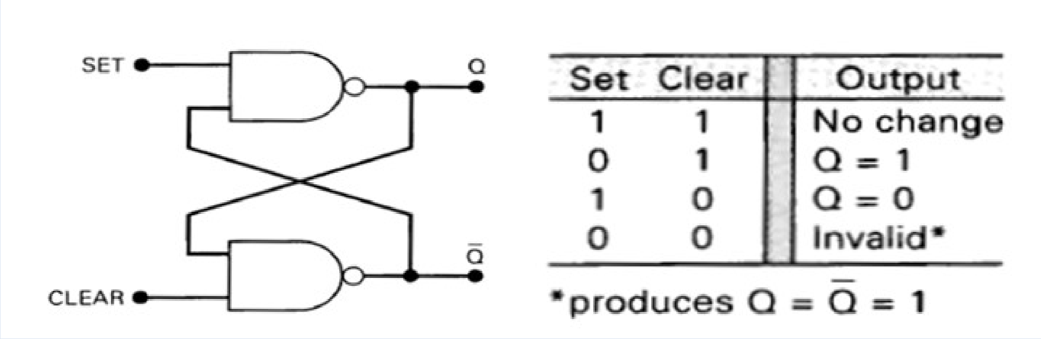
A D latch combines the S and R inputs of an S-R latch into one input by adding an inverter. When t clock is high, the output follows the D input, and when the clock goes low, the state is latched.



NOR Active-HIGH Latch



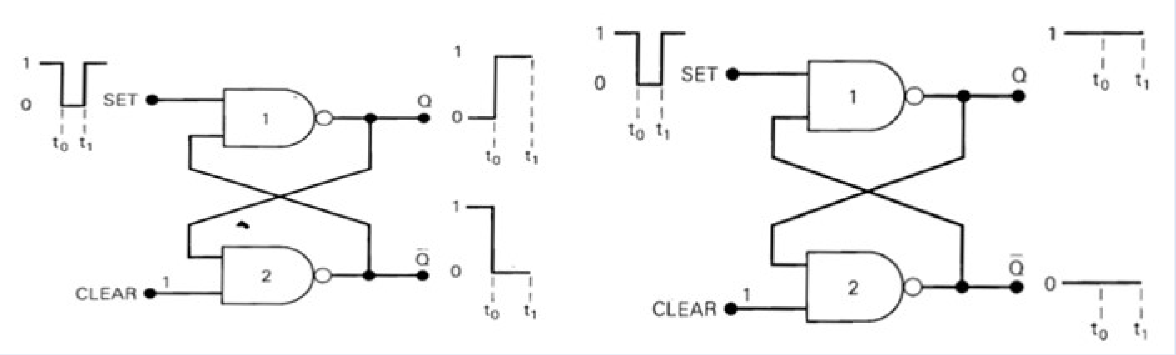
NAND Active-LOW Latch



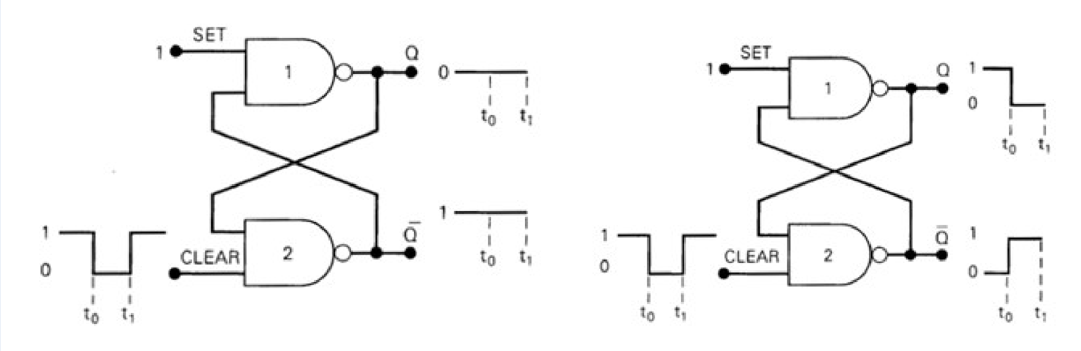
Set	Clear	Output
1	1	No change
0	1	Q = 1
1	0	Q = 0
0	0	Invalid*

*produces $Q = \overline{Q} = 1$

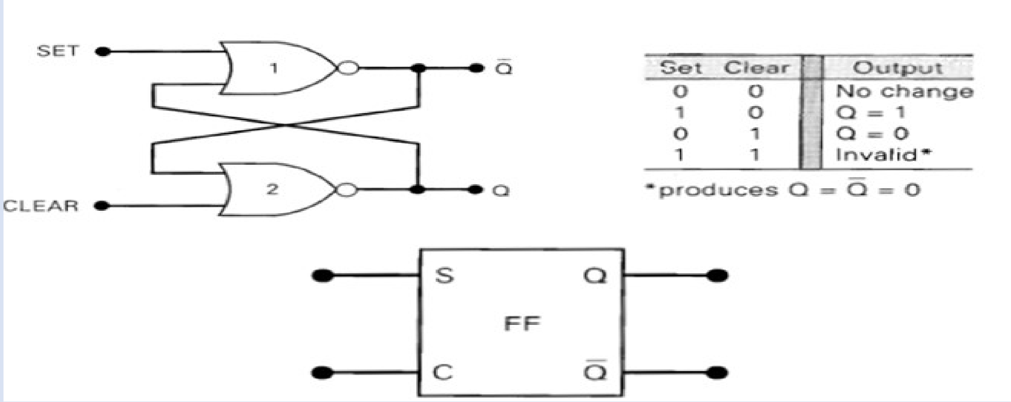
NAND Gate Latch with Truth Table



Setting NAND Gate Latches



Resetting NAND Gate Latches



Set	Clear	Output
0	0	No change
1	0	Q = 1
0	1	Q = 0
1	1	Invalid*

*produces $Q = \overline{Q} = 0$

NOR Gate Latch with Truth Table

PROCEDURE:

- Check all the components for their working.
- Insert the appropriate IC into the IC base.
- Make connections as shown in the circuit diagram.
- Verify the Truth Table and observe the outputs.

Discussion:

Here we learned to Construct SR Latch using NOR and NAND gates, Setting and Resetting SR Latch and observed their Characteristics.

References:

1. Reference book.
2. Online sources.

32 **Experiment No.: 13**

Experiment No. 13

Name of the experiment: Construct, test and investigate the operation of various types of Counters.

Objective: To study and verify the truth table of logic gates of different types of Counters ar identify various ICs and their specification.

Course Outcomes (COs), Program Outcomes (POs) and Assessment:

CO Statement	Corresponding PO	Domain / level of learning taxonomy	Delivery methods and activities	Assessment tools

CO3: Design and implement combinational and sequential circuit.	**Design of Solutions (PO3)**	**Psychomotor/Guided response**	☑ Simulation ☑ Experiment ☑ Practice lab ☑ Group discussion ☑ Tutorial	☑ Lab tests ☑ Lab reports ☑ Final lab test ☐ Open ended lab ☐ Project show & project presentation

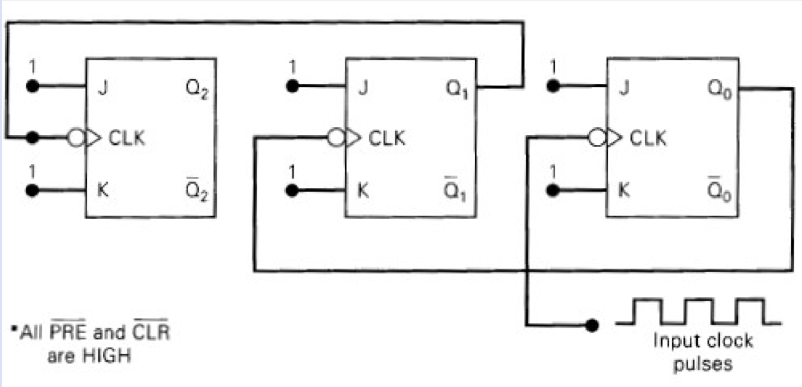
Components Required:

Logic gates (IC) trainer kit. Digital Lab Kit, Breadboard, Power Supplier, Connecting patch chords, I 7400, IC 7408, IC 7432, IC 7406, IC 7402, IC 7404, IC 7486

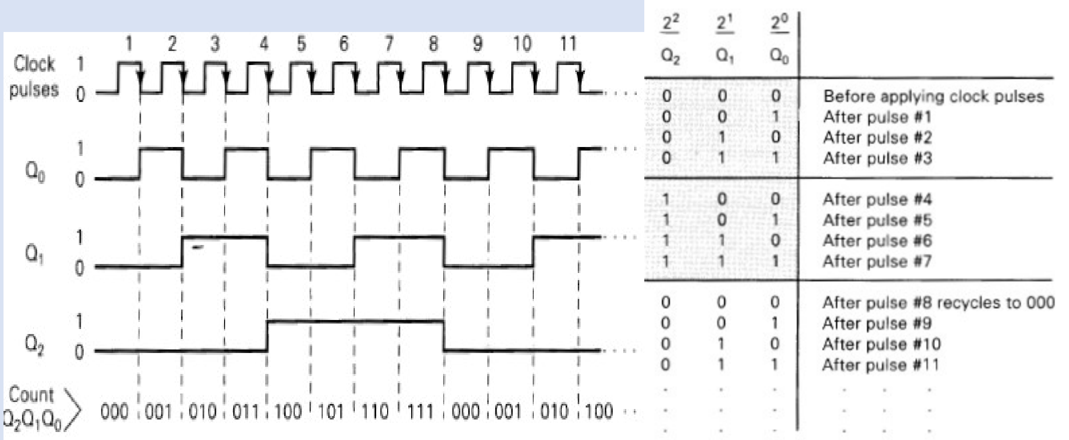
Theory:

Ring counter is a basic register with direct feedback such that the contents of the register simp circulate around the register when the clock is running. Here the last output that is QD in a shift regist is connected back to the serial input.

A basic ring counter can be slightly modified to produce another type of shift register counter call Johnson counter. Here complement of last output is connected back to the not gate input and not ga output is connected back to serial input. A four bit Johnson counter gives 8 state output.

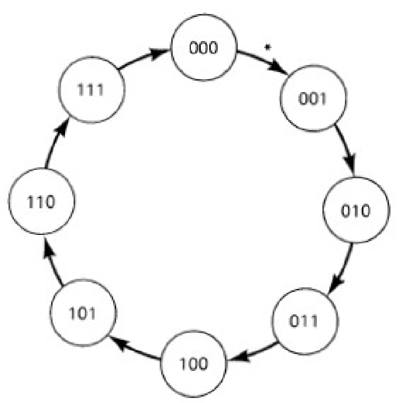


Frequency Division



2^2 Q_2	2^1 Q_1	2^0 Q_0	
0	0	0	Before applying clock pulses
0	0	1	After pulse #1
0	1	0	After pulse #2
0	1	1	After pulse #3
1	0	0	After pulse #4
1	0	1	After pulse #5
1	1	0	After pulse #6
1	1	1	After pulse #7
0	0	0	After pulse #8 recycles to 000
0	0	1	After pulse #9
0	1	0	After pulse #10
0	1	1	After pulse #11
.	.	.	. . .

Frequency Division



State Transition Diagram

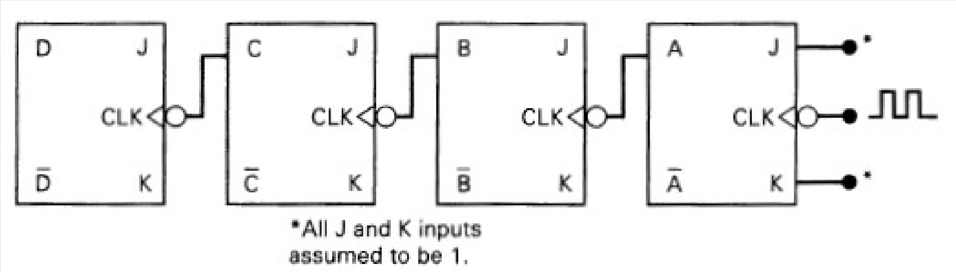
MOD Number

- MOD Number indicates number of states in counting sequence
- Last diagram: MOD number= 2^3= 8
- Consider a counter circuit that has six flip flops wired together as in last diagram.
  - a. What is the MOD number?
  - b. What is the frequency of last FF output if input clock has a frequency of 1 MHz?
  - c. What is the range of counting states of this counter?
  - d. Assume a starting state of (000000). What will be the counter's sequence after 129 pulse?

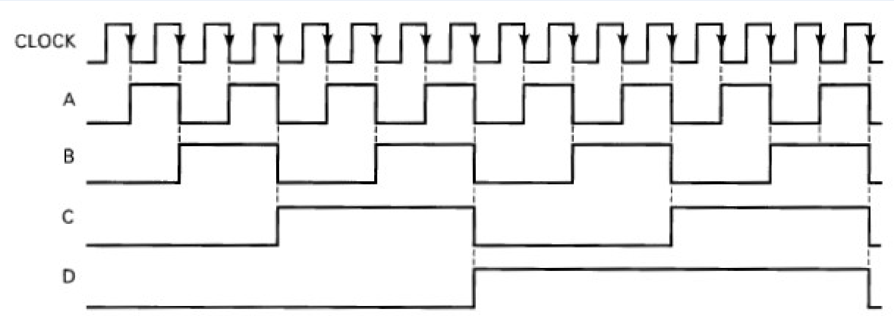
Counters

- Asynchronous (Ripple Counter)
  - – Up Counter
  - – Down Counter

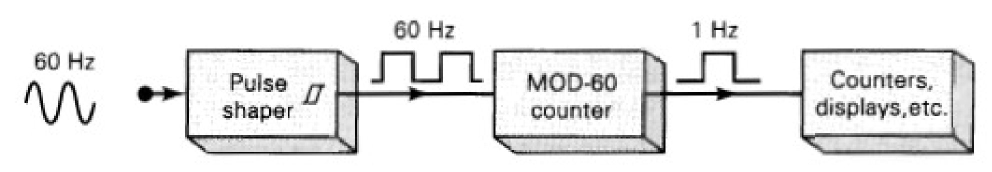
- Synchronous (Parallel Counter)

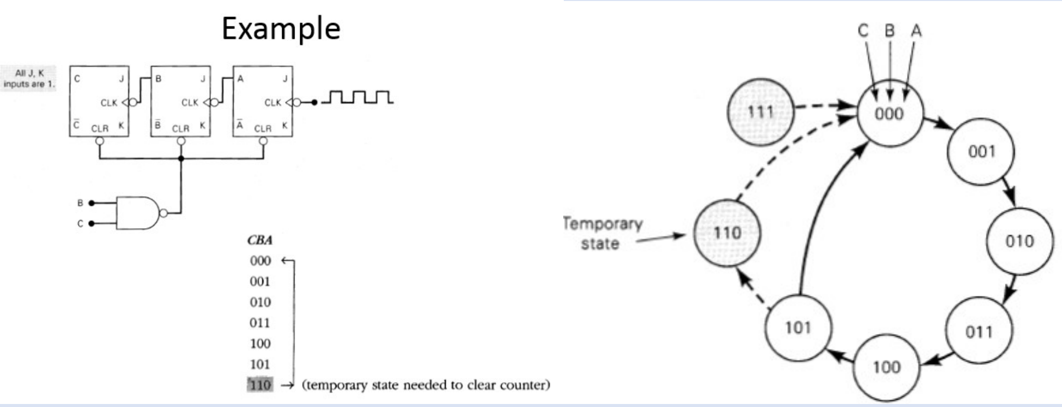


*All J and K inputs assumed to be 1.



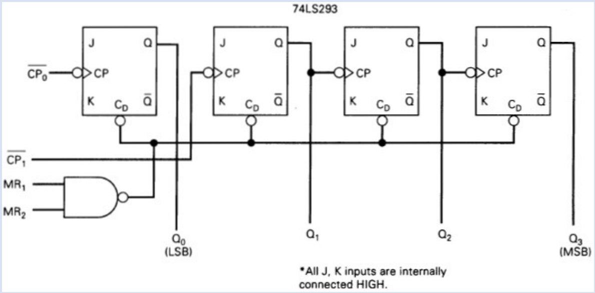
Ripple Counter

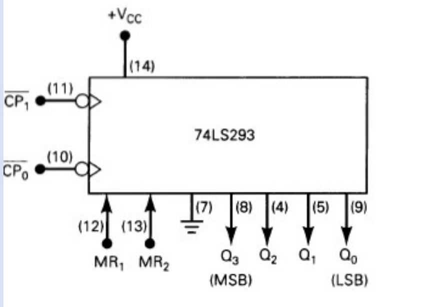



Example

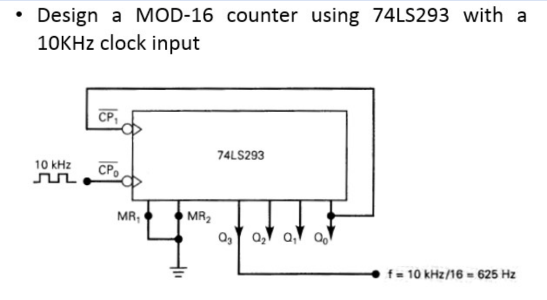
All J, K inputs are 1.

CBA
000 ←
001
010
011
100
101
110 → (temporary state needed to clear counter)

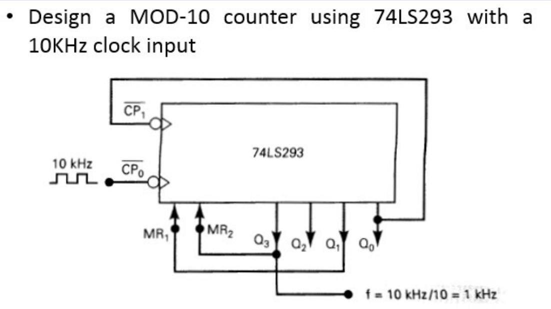




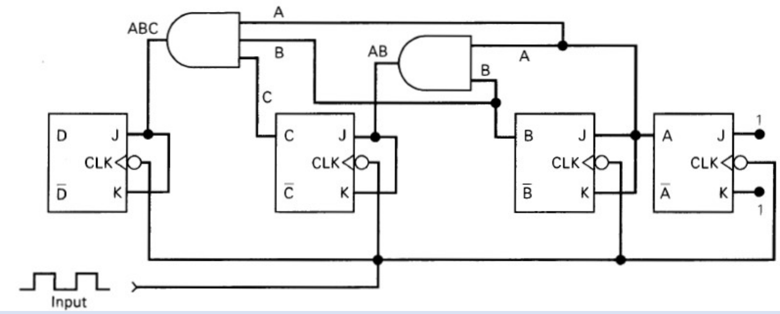
IC Ripple Counter



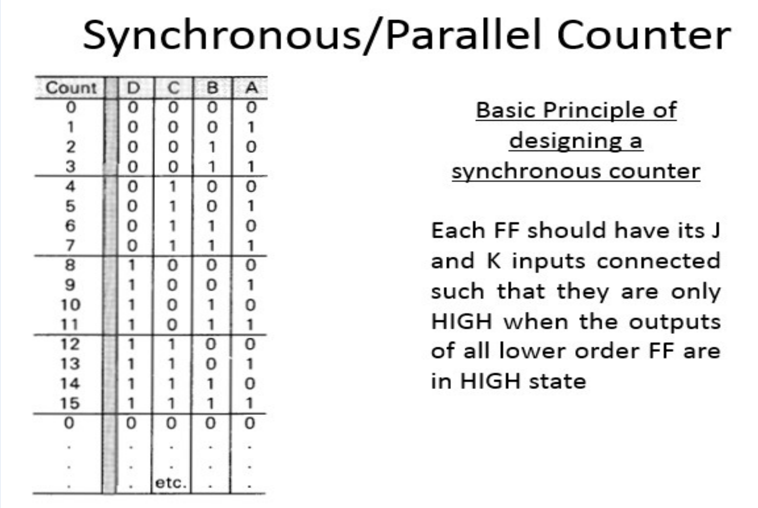
- Design a MOD-16 counter using 74LS293 with a 10KHz clock input



- Design a MOD-10 counter using 74LS293 with a 10KHz clock input



Synchronous/Parallel Counter



Synchronous/Parallel Counter

Count	D	C	B	A
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0
9	1	0	0	1
10	1	0	1	0
11	1	0	1	1
12	1	1	0	0
13	1	1	0	1
14	1	1	1	0
15	1	1	1	1
0	0	0	0	0
.	.	.	.	.
.	.	.	.	.
.	.	etc.	.	.

Basic Principle of designing a synchronous counter

Each FF should have its J and K inputs connected such that they are only HIGH when the outputs of all lower order FF are in HIGH state

Truth Table

Basic Principle of designing a synchronous counter:

Each FF should have its J and K inputs connected such that they are only HIGH when the outputs all lower order FF are in HIGH state

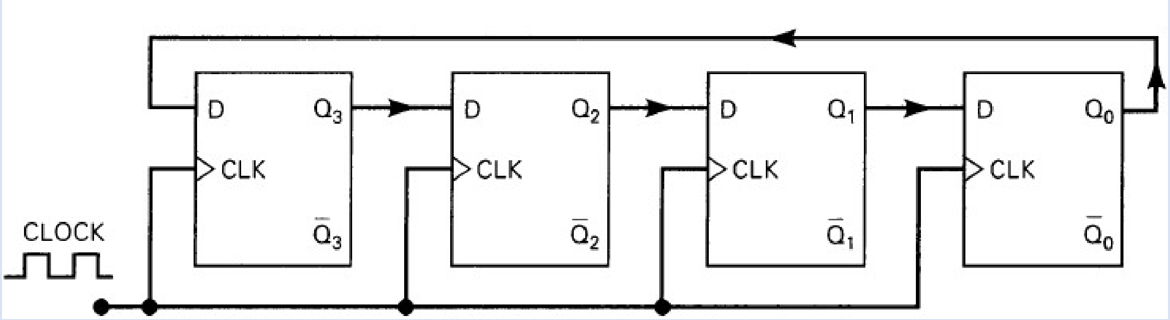
Shift Register Counter

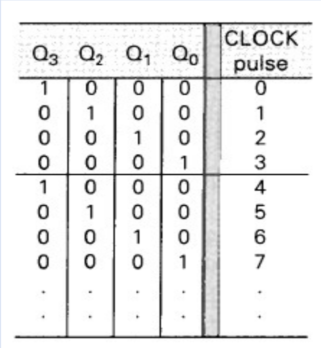
- **RING Counter**
- **Johnson Counter**

4 Bit Ring Counter

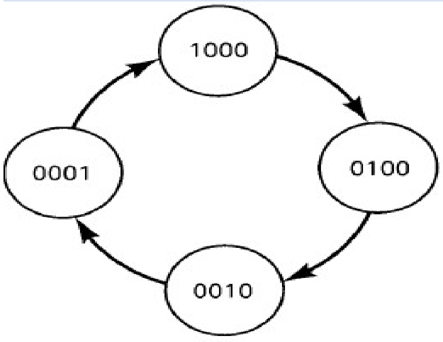
The ring counter can also be implemented with either D flip-flops or J-K flip-flops.

Here is a 4-bit ring counter constructed from a series of D flip-flops. Notice the feedback.

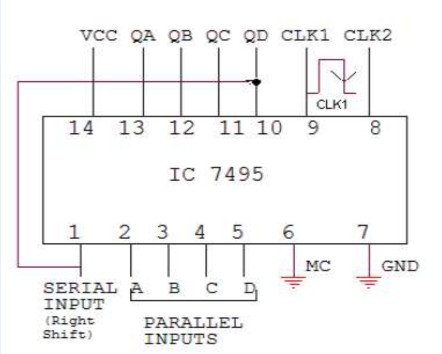




Q_3	Q_2	Q_1	Q_0	CLOCK pulse
1	0	0	0	0
0	1	0	0	1
0	0	1	0	2
0	0	0	1	3
1	0	0	0	4
0	1	0	0	5
0	0	1	0	6
0	0	0	1	7
.	.	.	.	.
.	.	.	.	.

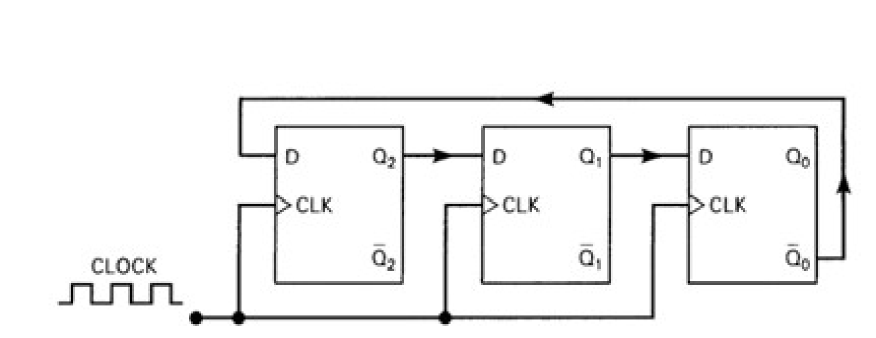


Truth Table **State Transition Diagram**



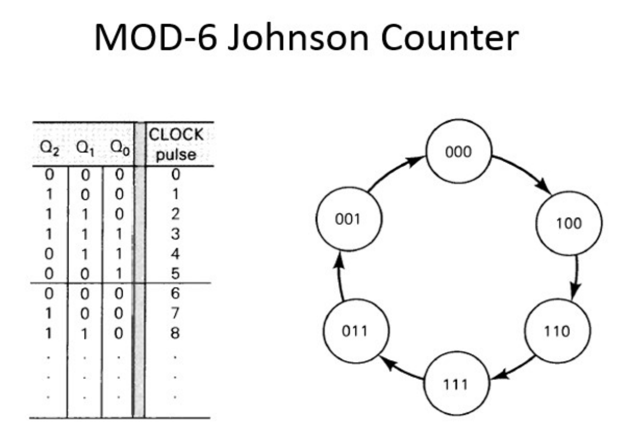
Clock pulses	Q_A	Q_B	Q_C	Q_D
0	1	0	0	0
1	0	1	0	0
2	0	0	1	0
3	0	0	0	1
4	1	0	0	0
5	0	1	0	0
6	0	0	1	0
7	0	0	0	1
8	1	0	0	0

Ring Counter Using IC 7495 **Truth Table**

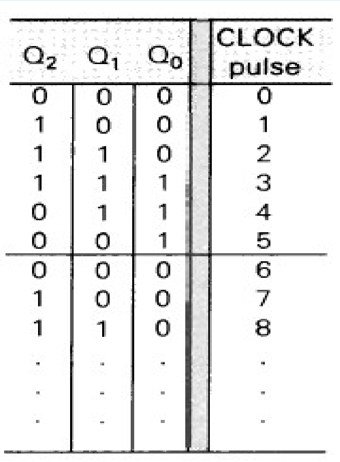


MOD-6 Johnson Counter

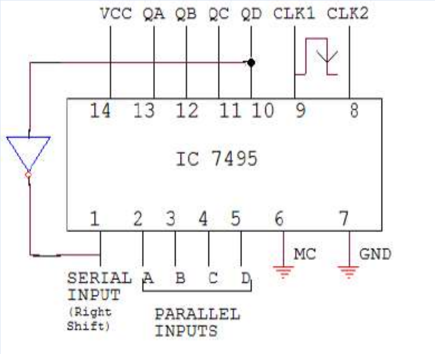
The Johnson counter is useful when you need a sequence that changes by only one bit at a time but has a limited number of states (2*n*, where *n* = number of stages).


MOD-6 Johnson Counter

Q_2	Q_1	Q_0	CLOCK pulse
0	0	0	0
1	0	0	1
1	1	0	2
1	1	1	3
0	1	1	4
0	0	1	5
0	0	0	6
1	0	0	7
1	1	0	8
.	.	.	.
.	.	.	.
.	.	.	.



Q_2	Q_1	Q_0	CLOCK pulse
0	0	0	0
1	0	0	1
1	1	0	2
1	1	1	3
0	1	1	4
0	0	1	5
0	0	0	6
1	0	0	7
1	1	0	8
.	.	.	.
.	.	.	.
.	.	.	.

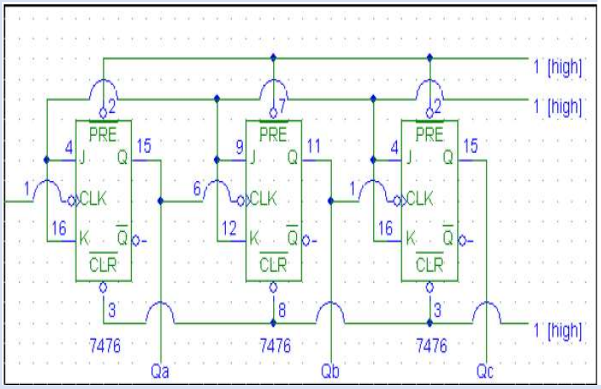


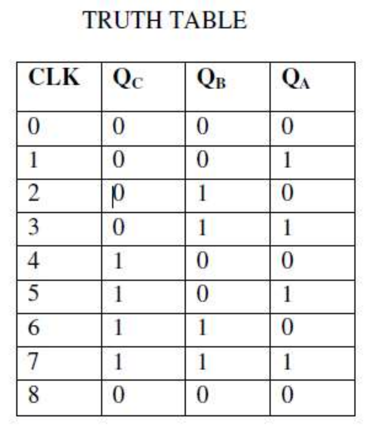
Clock pulses	Q_A	Q_B	Q_C	Q_D
0	0	0	0	0
1	1	0	0	0
2	1	1	0	0
3	1	1	1	0
4	1	1	1	1
5	0	1	1	1
6	0	0	1	1
7	0	0	0	1
8	0	0	0	0

Johnson Counter Using IC 7495 **Truth Table**

MOD-8 UP COUNTER

CIRCUIT DIAGRAM:




TRUTH TABLE

CLK	Q_C	Q_B	Q_A
0	0	0	0
1	0	0	1
2	0	1	0
3	0	1	1
4	1	0	0
5	1	0	1
6	1	1	0
7	1	1	1
8	0	0	0

Procedure:

- Check all the components for their working.
- Insert the appropriate IC into the IC base.
- Make connections as shown in the circuit diagram.
- Apply clock to pin number 9 and observe the output

Discussion: Here we learned to Construct Ripple UP/Down Counter, Ring Counter, Johnson Counte MOD-6 based Counter, MOD-8 Based Counter using frequency division techniques and MOD Techniques with different ICs and Basic gates and observed their Characteristics.

References:

1. Reference book.
2. Online sources.

Prepared by: **Checked by:** **Approved by:**