LAB MANUAL

Digital Electronics Lab

B.Tech. IV Semester EE

Debasish Mohanta, Lecturer, EE

Government College of Engineering, Keonjhar

At-Jamunalia Po. -Oldtown, Dist.-Keonjhar

Pin-758002, Odisha

Web-gcekjr.ac.in

Department of Electrical Engineering

CONTENTS

Sl. No.	Particulars	Page No.
1	Vision & Mission of Institute	1
2	Vision & Mission of Department	2
3	PO's	3
4	PSO'S & PEO's	4
5	CO's	5
6	Syllabus	6
7	Do's & Don'ts	7
8	List of Experiments	8
9	Experiments	9
10	Viva Questions	111
11	Data Sheet of ICs	113

The Vision and Mission of Government College of Engineering, Keonjhar are as follows:

Vision of the Institute:

To make our college a citadel of higher learning which can produce demand-oriented Engineering Graduates with good acumen and perception along with the latest and sophisticated technologies in order to place this college at a commanding height of academic excellence in the field of engineering.

Mission of the Institution:

i. To create globally competent engineers capable of working in an interdisciplinary environment with a strong theoretical and practical knowledge

ii. Enabling them to contribute society through innovation, entrepreneurship and leadership

iii. To inculcate critical thinking abilities among students and the spirit of Sustainable Development of natural resources with high moral and ethical values.

Vision of the Department:

To provide holistic education to build competent and productive researchers and graduates.

Mission of the Department:

M1: To provide quality education facilities for preparing professionals who match global standards.

M2: To create good atmosphere for research and innovation by providing state of the art laboratories.

M3: To prepare a cadre of engineers and scientists who will cater to the industrial development and economic growth of the society and country in future.

M4: To strengthen industry- institute interactions and interactions with alumni for mutual benefits by the exchange of knowledge, ideas and visions to promote lifelong learning.

Program Outcomes (PO's): -

PO1: **Engineering knowledge:** Apply the knowledge of basic sciences and fundamental engineering concepts in solving engineering problems.

PO2: **Problem analysis:** Identify and define engineering problems, conduct experiments and investigate to analyze and interpret data to arrive at substantial conclusions.

PO3: **Design/development of solutions:** Propose an appropriate solution for engineering problems complying with functional constraints such as economic, environmental, societal, ethical, safety and sustainability.

PO4: **Conduct investigations of complex problems:** Perform investigations, design and conduct experiments, analyze and interpret the results to provide valid conclusions.

PO5: **Modern tool usage:** Select/ develop and apply appropriate techniques and IT tools for the design and analysis of the systems.

PO6: **The engineer and society:** Give reasoning and assess societal, health, legal and cultural issues with competency in professional engineering practice.

PO7: **Environment and sustainability:** Demonstrate professional skills and contextual reasoning to assess environmental/ societal issues for sustainable development.

PO8: **Ethics:** An ability to apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.

PO9: **Individual and team work:** Function effectively as an individual and as a member or leader in diverse teams and in multi-disciplinary situations.

PO10: **Communication:** An ability to communicate effectively.

PO11: **Project management and finance:** Demonstrate apply engineering and management principles in their own / team projects in multi-disciplinary environment.

PO12: **Life-long learning:** An ability to do the needs of current technological trends at electrical industry by bridging the gap between academic and industry.

Program Specific Outcomes (PSO's): -

PSO1: Apply the knowledge of electrical engineering to analyze and solve the complex problems in electrical power and engineering with social utility.

PSO2: The application of recent techniques along with modern software tools for design, simulation and analyzing electrical systems.

PSO3: Adapting to technological changes and professional and societal needs by engaging in lifelong learning, thereby contributing to career development.

Program Educational Objectives (PEO's): -

PEO1: To apply fundamental knowledge in mathematics, science and engineering concepts in electrical engineering for the development of engineering field.

PEO2: To analyze, plan and design electrical system including modern methodologies to address the issues in a technically sound and economically viable manner.

PEO3: To develop a skillful workforce who can practice as a team professionally and ethically in a wide range of electrical engineering related fields.

PEO4: To prepare them for lifelong learning for successful carrier development by giving them the state- of the-art technology in the learning process.

Course Objectives: -

The laboratory enables students to get practical experience in design, realization and verification of

- Demorgan's theorem, SOP and POS forms
- Full Adders, Subtractors, Gray to binary converters and vice versa
- Multiplexers and Demultiplexers
- Decoders
- Flip-flops, Counters and Shift registers

Course Outcomes: -

After successful completion of this course, the students will be able to demonstrate the ability to –

- Construct the truth table of various logic gates and combination circuits using logic gates.
- Design, test and evaluate various combinational circuits such as adders, subtractors, multiplexers, demultiplexers, decoders etc.
- Construct flip-flops, counters and shift registers.
- Simulate various combinational circuits using VHDL
- Simulate sequential circuits flip-flops, counters etc. using VHDL.

Syllabus

4th Semester	REC4C201	Digital Electronics Lab	L-T-P 0-0-3	2 CREDITS

1. Digital Logic Gates: Investigate logic behavior of AND, OR, NAND, NOR, EX-OR, EX-NOR, Invert and Buffer gates, use of Universal NAND Gate.

2. Gate-level minimization: Two level and multi-level implementations of Boolean functions.

3. Combinational circuits: design, assemble and test: adders and subtractors, comparators.

4. Design, implement and test a given design example with (i) NAND Gates only (ii) NOR Gates only.

5. Design with multiplexers and demultiplexers.

6. Flip-flop: assemble, test and investigate operation of SR, D and JK Flip-flops.

7. Shift Register: Design and investigate the operation of all type of shift registers with parallel load.

8. Counters: Design, assemble and test various ripple and synchronous counters-decimal counter, Binary counter with parallel load.

9. Memory Unit: Investigate the behaviour of RAM unit and its storage capacity -16 X 4 RAM: testing, simulating and memory expansion.

10. Clock-pulse generator: design, implement and test.

11. Parallel adder and accumulator: design, implement and test.

12. Binary Multiplier: design and implement a circuit that multiplies 4-bit unsigned numbers to produce a 8-bit product.

(Verilog/VHDL simulation and implementation of Experiments Listed at Sl.No.3 to 12)

Do's & Don'ts

Do's

- Conduct yourself in a responsible manner at all times in the laboratory.
- Dress properly during a laboratory activity.
- Long hair, dangling jewellery and loose or baggy clothing are a hazard in the laboratory.
- Observe good housekeeping practices.
- Replace the materials in proper place after work to keep the lab area tidy.

Don'ts

- Do not wander around the room, distract other students, startle other students or interfere with the laboratory experiments of others.
- Do not eat food, drink beverages or chew gum in the laboratory and do not use laboratory glassware as containers for food or beverages.
- Do not open any irrelevant internet sites on lab computer
- Do not use a flash drive on lab computers.
- Do not upload, delete or alter any software on the lab PC.
- Do not switch on the trainer kit without verifying connection.

List of Experiments

EXPT. NO.	Name of the Experiment	PAGE NO.
1	Investigate logic behavior of AND, OR, NAND, NOR, EX-OR, EX-NOR, Invert and Buffer gates, use of Universal NAND Gate.	9
2	a. To simplify the Boolean function and construct the circuit. b. To verify Demorgan's theorem for 2 variables.	22
3	To construct and test a. Adder Circuit b. Subtractor Circuit	28
4	Design, implement and test a given design example with (i) NAND Gates only (ii) NOR Gates only	38
5	Implementation of 4x1 multiplexer and 1x4 demultiplexer.	42
6	Verification of truth tables of SR, J-K, and D Flip-Flops.	48
7	Design, and verify the all types of Shift Registers.	53
8	Design and verify the 4-Bit Synchronous/Asynchronous Counter.	58
9	Design 4-bit binary parallel adder using IC 7483.	62
10	VHDL Simulation of Experiments listed above.	66

EXPERIMENT NO: -1

I. Investigate the logic behavior of AND, OR, NOT, NAND, NOR, EX-OR gates.

AIM

To investigate the logic behavior of AND, OR, NOT, NAND, NOR, EX-OR gates.

APPARATUS REQUIRED

SL NO.	COMPONENT	SPECIFICATION	QUANTITY
1	OR GATE	IC 7432	1
2	AND GATE	IC 7408	1
3	NOT GATE	IC 7404	1
4	NAND GATE	IC 7400	1
5	NOR GATE	IC 7402	1
6	XOR GATE	IC 7486	1
7	BREAD BOARD	-	1
8	RESISTOR	220Ω	1
9	BATTERY	9V	1
10	LED	-	1
11	CONNECTING WIRE	-	AS PER REQUIREMENT

THEORY

Logic Gate

A logic gate is defined as a logic circuit with one output and two (or) more logic inputs. The output signal occurs only for combination of input variables.

There are basically three logic gates

a. AND gate
b. OR gate
c. NOT gate

These three gates are called as fundamental gates, because these gates form building block for most probably all digital circuits. From these fundamental gates other three gates are derived. They are called derived gates.

a. NAND gate

b. NOR gate
c. XOR gate

These logic gates are generally terms as decision making elements.

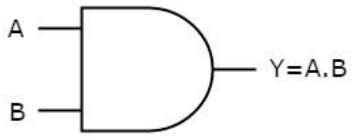
Basic Gates

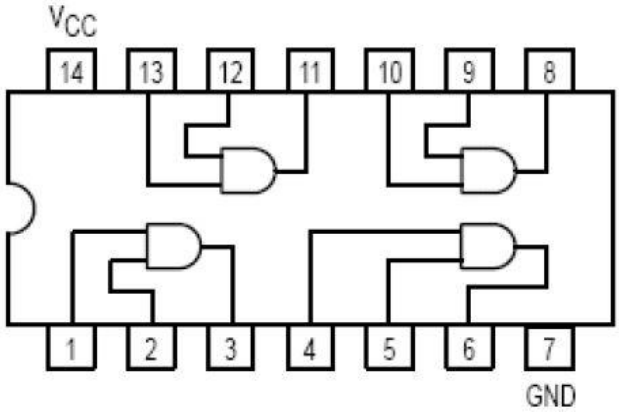
AND Gate

An AND gate has two (or) more logic inputs and single output. The AND operation is defined as the output as (1) one if and only if all the inputs are (1) one. **7408** is the two inputs AND gate IC. A and B are the Input terminals & Y is the Output terminal.

Its logical equation is

Y = A.B





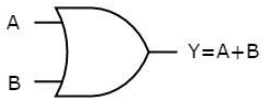
The truth table of two input AND gate is given as

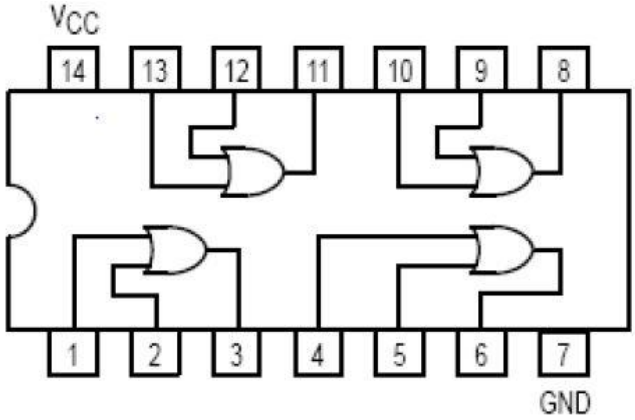
INPUT A	INPUT B	OUTPUT Y
0	0	0
0	1	0
1	0	0
1	1	1

OR Gate

An OR gate has two (or) more logic inputs and single output. The OR operation is defined as the output as (1) one if one or more than inputs are (1) one. **7432** is the two Input OR gate IC. A & B are the input terminals & Y is the Output terminal. Its logical equation is

Y = A + B





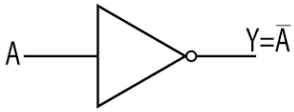
The truth table of two input OR gate is given as

INPUT A	INPUT B	OUTPUT Y
0	0	0
0	1	1
1	0	1
1	1	1

NOT Gate

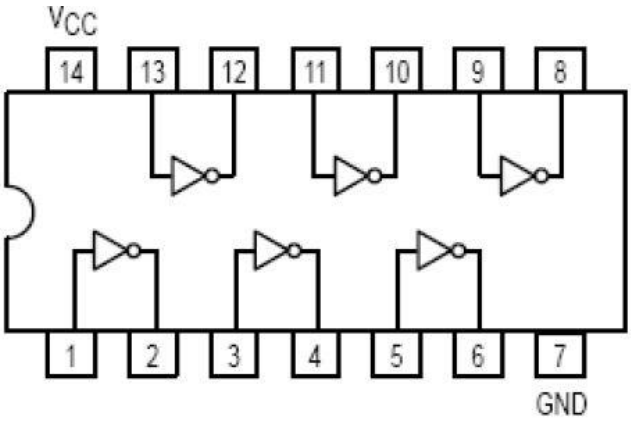
NOT gate has single input and single output. The NOT gate is also known as Inverter. It has one input (A) & one output (Y). IC No. is **7404**. Its logical equation is,

Y = A NOT B, Y = A'



The truth table of NOT gate is given as

INPUT A	OUTPUT Y
0	1
1	0



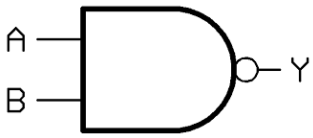
Derived Gates

NAND Gate

An NAND gate has two (or) more logic inputs and single output. The combination of NOT gate and an AND gate is called as NAND gate. The IC no. for NAND gate is **7400**. The NOT-AND operation is known as NAND operation. If all inputs are 1 then output produced is 0. NAND gate is inverted AND gate.

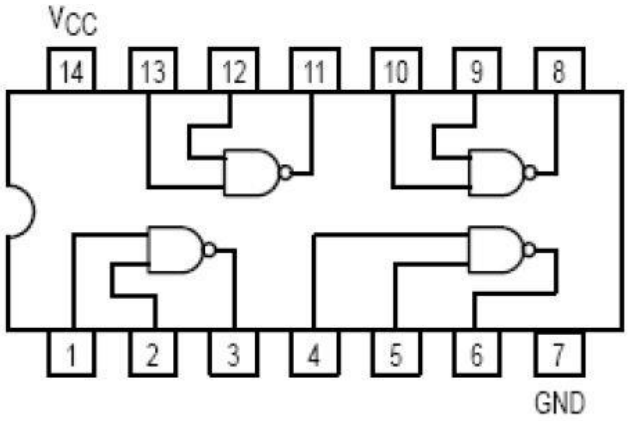
Its logical equation is

Y = (A. B)'



The truth table of two input NAND gate is given as

INPUT A	INPUT B	OUTPUT Y
0	0	1
0	1	1
1	0	1
1	1	0

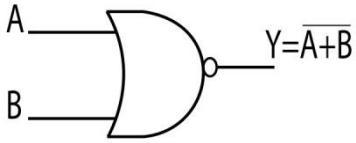


NOR Gate

An NAND gate has two (or) more logic inputs and single output. The combination of NOT gate and an OR gate is called as NOR gate. IC **7402** is two I/P NOR gate IC. The NOT-OR operation is known as NOR operation. If all the inputs are 0 then the O/P is 1. NOR gate is inverted OR gate.

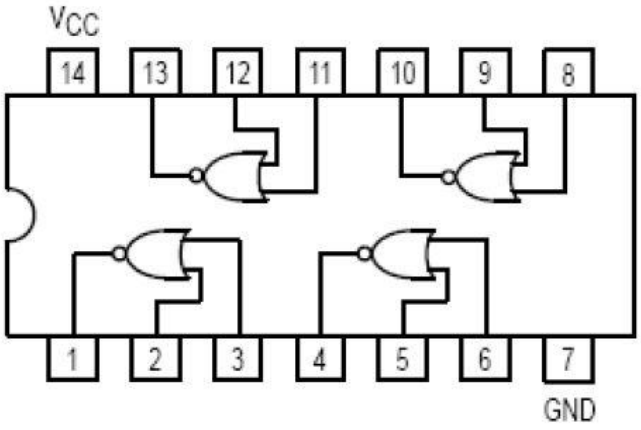
Its logical equation is

Y = (A+B)'



The truth table of two input NAND gate is given as

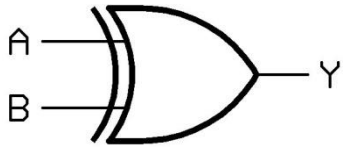
INPUT A	INPUT B	OUTPUT Y
0	0	0
0	1	1
1	0	1
1	1	1



XOR Gate

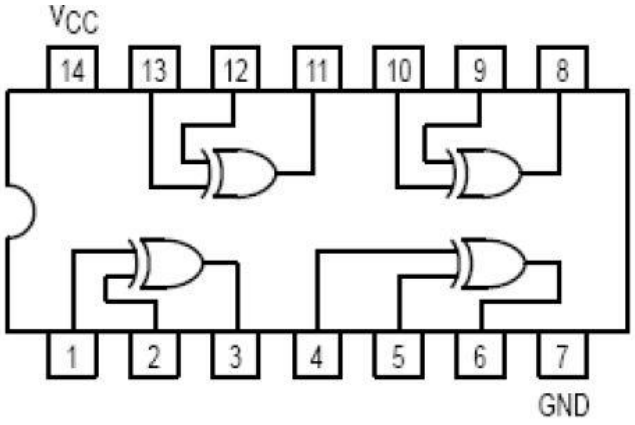
The XOR gate has two (or) more logic inputs and single output. **7486** is two inputs XOR gate IC. EX-OR gate is not a basic operation & can be performed using basic gates. Its logical equation is

Y=A'B+AB'



The truth table of two input XOR gate is given as

INPUT A	INPUT B	OUTPUT Y
0	0	0
0	1	1
1	0	1
1	1	0



Universal gate

The NAND and NOR gates are called Universal gates, because all the logic gates and logic functions can be implemented using NAND and NOR gates.

PROCEDURE

2 Input AND Gate

1. Position the IC [7408] properly on the solder less bread board.
2. Connect the pin 7 to ground using patch chord provided.
3. Connect the pin 14 to +5V supply using patch chord provided.
4. Connect the pin 1 and pin 2 to logic input of low (or) high.
5. Connect the pin 3to logic output of LED indicator.
6. Give the logic input low (or) high and note the output through LED indicator.
7. Verify the truth table.

2 Input OR Gate

1. Position the IC [7432] properly on the solder less bread board.
2. Connect the pin 7 to ground using patch chord provided.
3. Connect the pin 14 to +5V supply using patch chord provided.
4. Connect the pin 1 and pin 2 to logic input of low (or) high.
5. Connect the pin 3to logic output of LED indicator.
6. Give the logic input low (or) high and note the output through LED indicator.
7. Verify the truth table.

NOT Gate

1. Position the IC [7404] properly on the solder less bread board.
2. Connect the pin 7 to ground using patch chord provided.
3. Connect the pin 14 to +5V supply using patch chord provided.
4. Connect the pin 1 to logic input of low (or) high.
5. Connect the pin 2 to logic output of LED indicator.
6. Give the logic input low (or) high and note the output through LED indicator.
7. Verify the truth table.

2 Input NAND Gate

1. Position the IC [7400] properly on the solder less bread board.
2. Connect the pin 7 to ground using patch chord provided.
3. Connect the pin 14 to +5V supply using patch chord provided.
4. Connect the pin 1 and pin 2 to logic input of low (or) high.
5. Connect the pin 3 to logic output of LED indicator.

6. Give the logic input low (or) high and note the output through LED indicator.
7. Verify the truth table.

2 Input NOR Gate

1. Position the IC [7402] properly on the solder less bread board.
2. Connect the pin 7 to ground using patch chord provided.
3. Connect the pin 14 to +5V supply using patch chord provided.
4. Connect the pin 2 and pin 3 to logic input of low (or) high.
5. Connect the pin 1 to logic output of LED indicator.
6. Give the logic input low (or) high and note the output through LED indicator.
7. Verify the truth table.

2 Input XOR Gate

1. Position the IC [7486] properly on the solder less bread board.
2. Connect the pin 7 to ground using patch chord provided.
3. Connect the pin 14 to +5V supply using patch chord provided.
4. Connect the pin 2 and pin 3 to logic input of low (or) high.
5. Connect the pin 1 to logic output of LED indicator.
6. Give the logic input low (or) high and note the output through LED indicator.
7. Verify the truth table.

PRECAUTIONS:

1. Make the connections according to the IC pin diagram.
2. The connections should be tight.
3. The V_{CC} and ground should be applied carefully at the specified pin only.

OBSERVATION

AND Gate [L=logic 0, H=logic 1]

INPUT A	INPUT B	OUTPUT Y
L	L	
L	H	
H	H	
H	L	

OR Gate [L=logic 0, H=logic 1]

INPUT A	INPUT B	OUTPUT Y
L	L	
L	H	
H	L	
H	H	

NOT Gate [L=logic 0, H=logic 1]

INPUT A	OUTPUT Y
L	
H	

NAND Gate [L=logic 0, H=logic 1]

INPUT A	INPUT B	OUTPUT Y
L	L	
L	H	
H	L	
H	H	

NOR Gate [L=logic 0, H=logic 1]

INPUT A	INPUT B	OUTPUT Y
L	L	
L	H	
H	L	
H	H	

XOR Gate [L=logic 0, H=logic 1]

INPUT A	INPUT B	OUTPUT Y
L	L	
L	H	
H	L	
H	H	

RESULT: The logic behavior of AND, OR, NOT, NAND, NOR, EX-OR gates is verified.

QUESTIONS

1. Define Logic gates?

2. Define IC?

3. Define Universal gates?

4. How many no. of input variables can a NOT Gate have?

5. When will the output of a NAND Gate be 0?

II. Study and Verify NAND gate as a Universal Gate

AIM

To Study and Verify NAND gate as a Universal Gate.

APPARATUS REQUIRED

SL NO.	COMPONENT	SPECIFICATION	QUANTITY
1	NAND GATE	IC 7400	1
2	BREAD BOARD	-	1
3	RESISTOR	220Ω	1
4	BATTERY	9V	1
5	LED	-	1
6	CONNECTING WIRE	-	AS PER REQUIREMENT

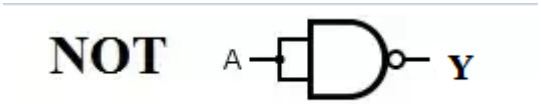
THEORY

Universal gate

The NAND and NOR gates are called Universal gates, because all the logic gates and logic functions can be implemented using NAND and NOR gates.

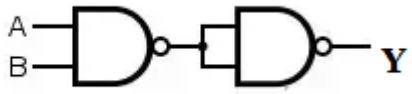
NAND Gate as Universal gate

a. **NAND GATE AS INVERTER:** The circuit diagram of implementation of NAND gate as inverter is shown below:



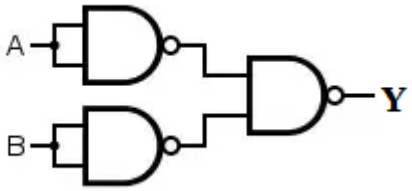
INPUT A	OUTPUT Y
0	1
1	0

b. **NAND GATE AS AND GATE:** The circuit diagram of implementation of NAND gate as AND gate is shown below:

AND 

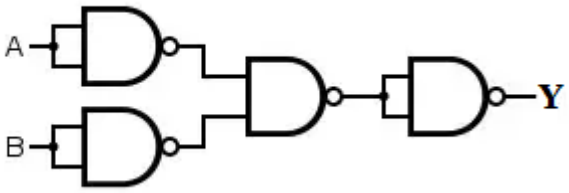
INPUT A	INPUT B	OUTPUT Y
0	0	0
0	1	0
1	0	0
1	1	1

c. NAND GATE AS OR GATE: The circuit diagram of implementation of NAND gate as OR gate is shown below:

OR 

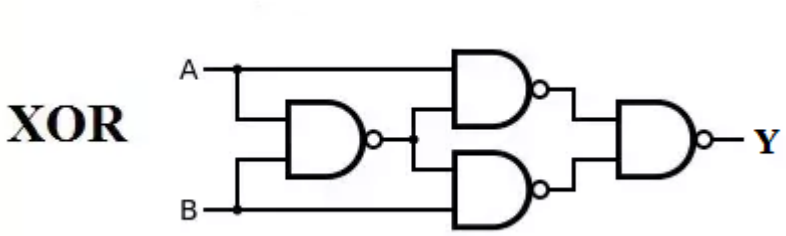
INPUT A	INPUT B	OUTPUT Y
0	0	1
0	1	1
1	0	1
1	1	0

d. NAND GATE AS NOR GATE: The circuit diagram of implementation of NAND gate as is shown below:

NOR 

INPUT A	INPUT B	OUTPUT Y
0	0	1
0	1	0
1	0	0
1	1	0

e. NAND GATE AS EX-OR GATE The circuit diagram of implementation of NAND gate as is shown below:



INPUT A	INPUT B	OUTPUT Y
0	0	0
0	1	1
1	0	1
1	1	0

PROCEDURE:

1. Make the connections as per the logic diagram.
2. Connect +5v to pin 14 & ground to pin 7.
3. Apply diff combinations of inputs to the i/p terminals
4. Note o/p for NAND as universal gate.
5. Verify the truth table.

RESULT: - The NAND gate as a Universal Gate is verified.

EXPERIMENT NO: -2

I. Implementation of the Given Boolean Function using Logic Gates.

AIM

To implement the given Boolean Function using Logic Gates.

$$Y = AB + A'B$$

APPARATUS REQUIRED

SL NO.	COMPONENT	SPECIFICATION	QUANTITY
1	NAND GATE	IC 7400	1
2	OR GATE	IC 7432	1
3	AND GATE	IC 7408	1
2	BREAD BOARD	-	1
3	RESISTOR	220Ω	1
4	BATTERY	9V	1
5	LED	-	1
6	CONNECTING WIRE	-	AS PER REQUIREMENT

THEORY

It is desirable to connect gates together in as few a number as possible to create desired output result given a fixed set of input conditions. Alternatively, it may be necessary to utilize one type of gate to produce several other logic functions. Purchasing one IC type in bulk quantity has the advantage of reducing the cost of these chips. Two theories are used to facilitate these objectives.

i) Boolean Algebra and
ii) Demorgan's Theorem

Boolean algebra utilizes rules based on logic gate operations.

Karnaugh's Map Reduction Technique

The implication of logic circuits is based on Karnaugh's map. There are two fundamental approaches in logic design.

i. Product of Sum method
ii. Sum of product method

The Sum of product solution results in NAND circuit, while the Product of Sum solution results in NOR circuit corresponding to a given truth table.

The designers can select either method, usually selects the simpler circuit because it costs less and more reliable.

The complexity of the digital logic gates can be implemented in a Boolean function. The map method provides a simple straightforward procedure for minimizing the Boolean functions. This map method is known as “Karnaugh’s map”.

The map is the diagram made up squares. Each square represents one minterm. Since any Boolean functions can be expressed as a sum of minterm. The simplest algebraic expression is any one from a sum of products or product of sums that has a minimum number of literals.

Two variable map has 4 minterm. It can be represented as

m0	m1
m2	m3

$\bar{x}\bar{y}$	$\bar{x}y$
$x\bar{y}$	xy

Two Variable Map

A three variable map has 8 minterms for the three variables. This map has 8 squares. Three variable map is represents as shown below.

m0	m1	m3	m2
m4	m5	m7	m6

$\bar{x}\bar{y}\bar{z}$	$\bar{x}\bar{y}\bar{z}$	$\bar{x}yz$	$\bar{x}y\bar{z}$
$x\bar{y}\bar{z}$	$x\bar{y}z$	xyz	$xy\bar{z}$

Three Variable Map

PROCEDURE

1. Position the IC’s [7408, 7432, and 7404] properly on the solder less bread board.
2. Connect the pin 7 of all the chips to ground using patch chord provided.
3. Connect the pin 14 of all the chips to +5V supply using patch chord provided.
4. Input A signal and B signals are connected at pin no 1 and 2 of the IC 7408.
5. Input A signal is complemented by connecting pin 1 of the IC 7404.
6. The inverted input A and B signals are connected to the pin 4 and 5 of IC 7432.
7. The output (7408) pin 3 and 6 are concerted to input pin 1 and 2 of OR gate.
8. The output pin 3 of the OR gate give the Boolean function result.

OBSERVATION [L=logic 0, H=logic 1]]

INPUT A	INPUT B	$Y = AB + A'B$
L	L	
L	H	
H	L	
H	H	

RESULT: The given Boolean Function using Logic Gates is verified.

II. Verification of Demorgan's theorem for 2 variables

AIM

To verify Demorgan's theorem for 2 variables.

APPARATUS REQUIRED

SL NO.	COMPONENT	SPECIFICATION	QUANTITY
1	NAND GATE	IC 7400	1
2	OR GATE	IC 7432	1
3	AND GATE	IC 7408	1
2	BREAD BOARD	-	1
3	RESISTOR	220Ω	1
4	BATTERY	9V	1
5	LED	-	1
6	CONNECTING WIRE	-	AS PER REQUIREMENT

THEORY

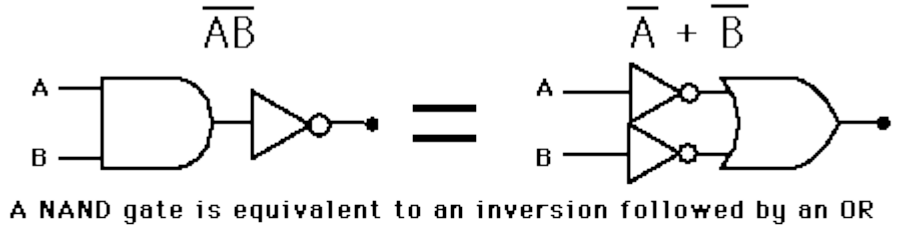
Theorem 1: The compliment of the product of two variables is equal to the sum of the compliment of each variable. Thus according to De-Morgan's laws or De-Morgan's theorem if A and B are the two variables or Boolean numbers. Then accordingly,

$$\overline{A.B} = \bar{A} + \bar{B}$$

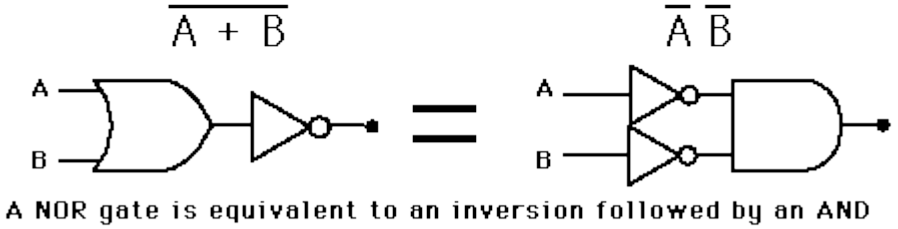
Theorem 2:-

The compliment of the sum of two variables is equal to the product of the compliment of each variable. Thus according to De Morgan's theorem if A and B are the two variables then,

$$\overline{A + B} = \bar{A}.\bar{B}$$



A NAND gate is equivalent to an inversion followed by an OR



A NOR gate is equivalent to an inversion followed by an AND

De-Morgan's laws can also be implemented in Boolean algebra in the following steps:

1. While doing Boolean algebra at first replace the given operator. That is (+) is replaced with (.) and (.) is replaced with (+).
2. Compliment of each of the term is to be found.

PROCEDURE

1. Realize the Demorgan's theorem using logic gates.
2. Connect V_{CC} and ground as shown in the pin diagram.
3. Make connections as per the logic gate diagram.
4. Apply the different combinations of input according to the truth tables and verify that the results are correct.

OBSERVATION [L=logic 0, H=logic 1]

Theorem 1

INPUT A	INPUT B	$\overline{A.B}$	$\bar{A}+\bar{B}$
L	L		
L	H		
H	L		
H	H		

Theorem 2

INPUT A	INPUT B	$\overline{A+B}$	$\bar{A}.\bar{B}$
L	L		
L	H		
H	L		
H	H		

RESULT: The truth table of Demorgan's theorem for 2 variables is verified.

QUESTIONS

1. Define K-map?
2. Define SOP?
3. Define POS?
4. What are combinational circuits?
5. If there are four variables how many cells the K-map will have?
6. Which code is used for the identification of cells?

EXPERIMENT NO: -3

I. Realize binary adder circuits using logic gates.

AIM

To realize half adder using Logic gates.

APPARATUS REQUIRED

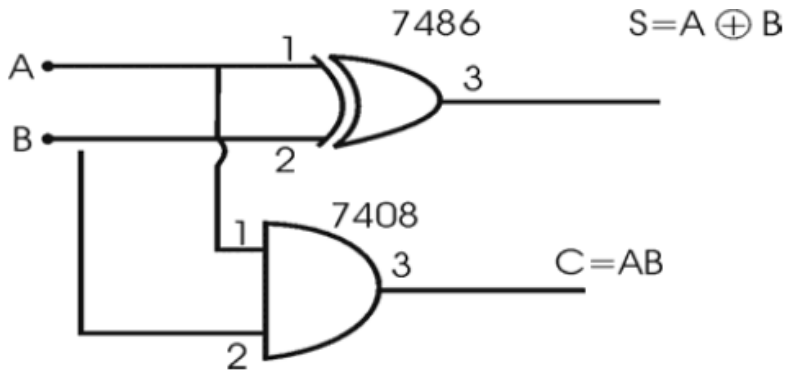
SL NO.	COMPONENT	SPECIFICATION	QUANTITY
1	XOR GATE	IC 7486	1
2	AND GATE	IC 7408	1
3	BREAD BOARD	-	1
4	RESISTOR	220Ω	1
5	BATTERY	9V	1
6	LED	-	1
7	CONNECTING WIRE	-	AS PER REQUIREMENT

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:

$$S = A \oplus B$$

$$C = A.B$$



Truth Table

INPUT A	INPUT B	$Y = A \oplus B$	C=A.B
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

PROCEDURE: -

1. Position the IC's [7486, 7408] properly on the solder less bread board.
2. Connect the pin 7 of all the chips to ground using patch chord provided.
3. Connect the pin 14 of all the chips to +5V supply using patch chord provided.
4. Connect the circuit as shown and get the output of Sum and Carry separately.
5. Take input from pin no. 1&2 of IC no.7486 and take output at pin no.3.
6. Pin no.3 is connected with LED.
7. Short pin no.1 of IC no.7486 to pin no.1 of IC no.7408.
8. Similarly, short pin no.2 of IC no.7486 to pin no.2 of IC no.7408.
9. Take output at Pin no.3 of IC no.7408 and connect to LED.

PRECAUTIONS: -

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.

OBSERVATION [L=logic 0, H=logic 1]

INPUT A	INPUT B	$Y = A \oplus B$	C=A.B
L	L		
L	H		
H	L		
H	H		

RESULT: -The truth table of half adder is verified.

AIM

To realize full adder using Logic gates.

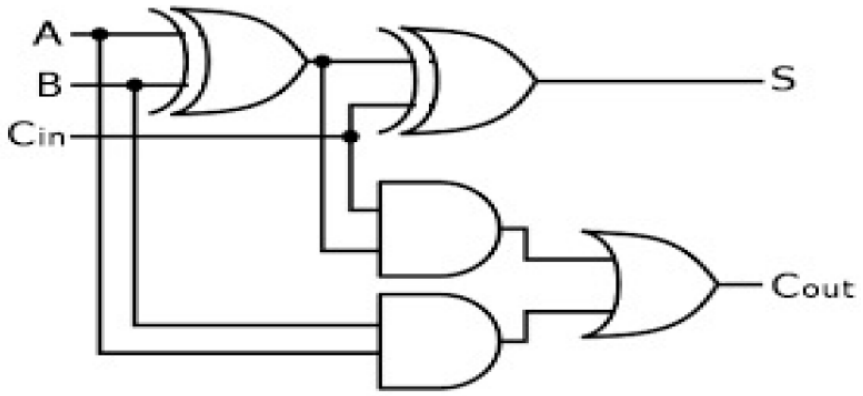
APPARATUS REQUIRED

SL NO.	COMPONENT	SPECIFICATION	QUANTITY
1	XOR GATE	IC 7486	1
2	AND GATE	IC 7408	1
3	OR GATE	IC 7432	
3	BREAD BOARD	-	1
4	RESISTOR	220Ω	1
5	BATTERY	9V	1
6	LED	-	1
7	CONNECTING WIRE	-	AS PER REQUIREMENT

THEORY

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,C_{in} is called a full-adder. The Boolean functions describing the full-adder are:

$$S = A \oplus B \oplus C_{in}$$
$$C = A.B + C_{in}(A \oplus B)$$



Truth Table

INPUT A	INPUT B	INPUT C_{in}	$S = A \oplus B \oplus C_{in}$	$C = A.B + C_{in}(A \oplus B)$
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

PROCEDURE: -

1. Position the IC's [7486, 7408, and 7432] properly on the solder less bread board.
2. Connect the pin 7 of all the chips to ground using patch chord provided.
3. Connect the pin 14 of all the chips to +5V supply using patch chord provided.
4. Connect the circuit as shown and get the output of Sum and Carry separately.
5. Make the connections as per the circuit diagram for the full adder circuit, on the trainer kit.
6. Switch on the V_{CC} power supply and apply the various combinations of the inputs according to the respective truth tables.
7. Verify that the outputs are according to the expected results.

OBSERVATION [L=logic 0, H=logic 1]

INPUT A	INPUT B	INPUT C_{in}	$S = A \oplus B \oplus C_{in}$	$C = A.B + C_{in}(A \oplus B)$
L	L	L		
L	L	H		
L	H	L		
L	H	H		
H	L	L		
H	L	H		
H	H	L		
H	H	H		

RESULT: -The truth table of full adder is verified.

QUESTIONS

1. Give the basic rules for binary addition?
2. Specify the no. of I/P and O/P of Half adder?
3. What is the drawback of half adder?
4. Write the equation for sum & carry of half adder?
5. Write the equation for sum & carry of full adder?
6. How many half adders will be required for implementing full adder?
7. Define Bit?

II. To realize subtractor circuits using Logic gates.

AIM

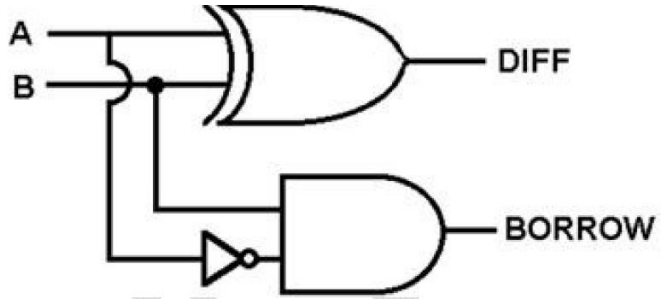
To realize half subtractor using Logic gates.

APPARATUS REQUIRED

SL NO.	COMPONENT	SPECIFICATION	QUANTITY
1	XOR GATE	IC 7486	1
2	AND GATE	IC 7408	1
3	NOT GATE	IC 7404	
3	BREAD BOARD	-	1
4	RESISTOR	220Ω	1
5	BATTERY	9V	1
6	LED	-	1
7	CONNECTING WIRE	-	AS PER REQUIREMENT

THEORY

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_0. This operation is called half subtraction and the circuit to realize it is called a half subtractor. The Boolean functions describing the half-subtractor are:



$$D = A \oplus B$$
$$B_0 = A'B$$

.

Truth Table

INPUT A	INPUT B	$D = A \oplus B$	$B_0 = A'B$
0	0	0	0
0	1	0	1
1	0	1	0
1	1	0	0

PROCEDURE: -

1. Position the IC's [7408, 7404, and 7432] properly on the solder less bread board.
2. Connect the pin 7 of all the chips to ground using patch chord provided.
3. Connect the pin 14 of all the chips to +5V supply using patch chord provided.
4. Connect the circuit as shown and get the output of difference and borrow separately.
5. Take input from pin no. 1&2 of IC no.7486 and take output at pin no.3.
6. Pin no.3 is connected with LED.
7. Short pin no.1 of IC no.7486 to pin no.1 of IC no.7404 and take output at pin no.2 of IC no 7404.
8. Short pin no.2 of IC no.7404 to pin no.1 of IC no.7408.
9. Similarly, short pin no.2 of IC no.7486 to pin no.2 of IC no.7408.
10. Take output at Pin no.3 of IC no.7408 and connect to LED.

OBSERVATION [L=logic 0, H=logic 1]

INPUT A	INPUT B	$D = A \oplus B$	$B_0 = A'B$
L	L		
L	H		
H	L		
H	H		

RESULT: -The truth table of half subtractor is verified.

AIM

To realize full subtractor using Logic gates.

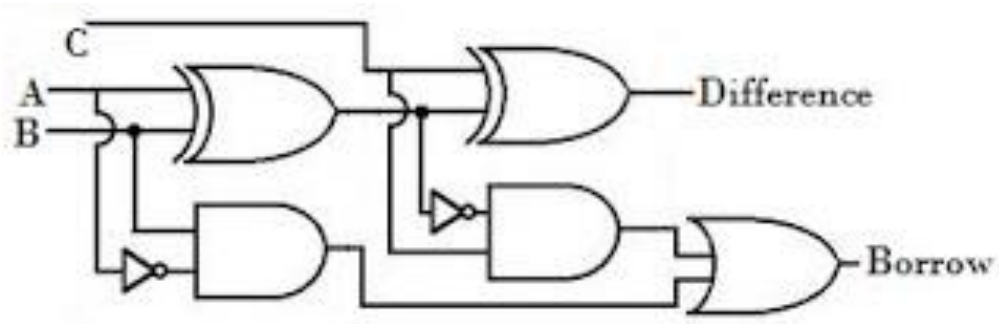
APPARATUS REQUIRED

SL NO.	COMPONENT	SPECIFICATION	QUANTITY
1	XOR GATE	IC 7486	1
2	AND GATE	IC 7408	1
3	OR GATE	IC 7432	
4	NOT GATE	IC 7404	
5	BREAD BOARD	-	1
6	RESISTOR	220Ω	1
7	BATTERY	9V	1
8	LED	-	1
9	CONNECTING WIRE	-	AS PER REQUIREMENT

THEORY

Full-subtractor: Subtracting two single-bit binary values, B, C from a single-bit value A produces a difference bit D and a borrow out B_0 bit. This is called full subtraction. The Boolean functions describing the full subtractor.

$$D = A \oplus B \oplus C$$
$$B_0 = A'B + A'C + BC$$



Truth Table

INPUT A	INPUT B	INPUT C_{in}	$D = A \oplus B \oplus C$	$B_0 = A'B + A'C + BC$
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	0	1

PROCEDURE: -

1. Position the IC's [7486, 7408, 7402 and 7432] properly on the solder less bread board.
2. Connect the pin 7 of all the chips to ground using patch chord provided.
3. Connect the pin 14 of all the chips to +5V supply using patch chord provided.
4. Connect the circuit as shown and get the output of Difference and Borrow separately.
5. Make the connections as per the circuit diagram for the full substractor circuit, on the trainer kit.
6. Switch on the V_{CC} power supply and apply the various combinations of the inputs according to the respective truth tables.
7. Verify that the outputs are according to the expected results.

OBSERVATION [L=logic 0, H=logic 1]

INPUT A	INPUT B	INPUT C_{in}	$D = A \oplus B \oplus C$	$B_0 = A'B + A'C + BC$
L	L	L		
L	L	H		
L	H	L		
L	H	H		
H	L	L		
H	L	H		
H	H	L		
H	H	H		

RESULT: -The truth table of full subtractor is verified.

QUESTIONS

1. What is half subtractor?
2. For implementing half subtractor how many EX-OR, AND gates and Not gates are required?
3. What are the logical equations for difference & borrow?
4. How full subtractor is different from half subtractor.
5. How many bits we use in half subtractor for subtraction?

EXPERIMENT NO: -4

Design, implement and test a given design example with (i) NAND Gates only (ii) NOR Gates only

AIM

To design, implement and test a given design example with (i) NAND Gates only (ii) NOR Gates only.

APPARATUS REQUIRED

SL NO.	COMPONENT	SPECIFICATION	QUANTITY
1	NAND GATE	IC 7400	1
2	BREAD BOARD	-	1
3	RESISTOR	220Ω	1
4	BATTERY	9V	1
5	LED	-	1
6	CONNECTING WIRE	-	AS PER REQUIREMENT

THEORY

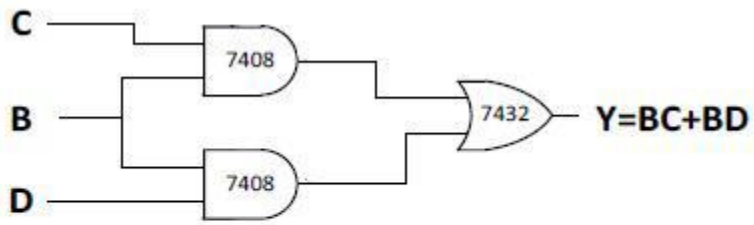
Canonical Forms (Normal Forms): Any Boolean function can be written in disjunctive normal form (sum of min-terms) or conjunctive normal form (product of max-terms). A Boolean function can be represented by a Karnaugh map in which each cell corresponds to one minterm/ maxterm.

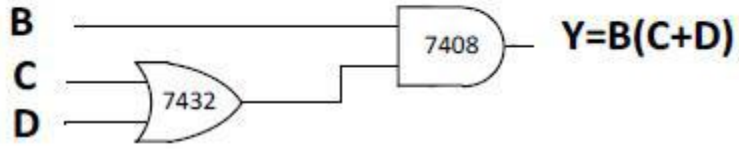
1. Sum of minterms: Sum of Product (SOP)
2. Product of maxterms: Product of Sum (POS)

Simplified Boolean expression:

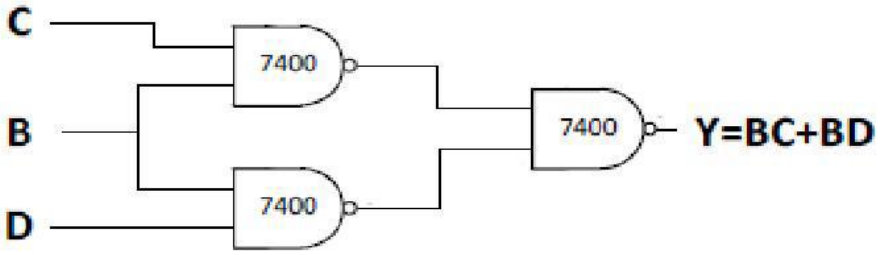
SOP form $Y = f(A, B, C, D) = BC + BD$
POS form $Y = f(A, B, C, D)$

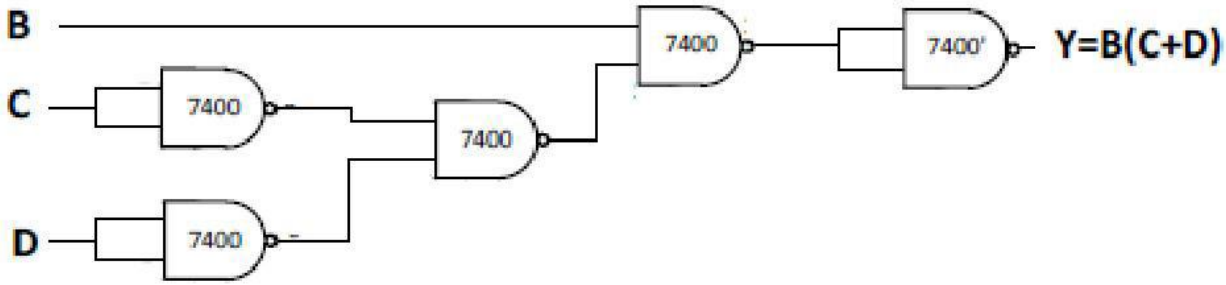
(i) Simplification using Basic Gates:



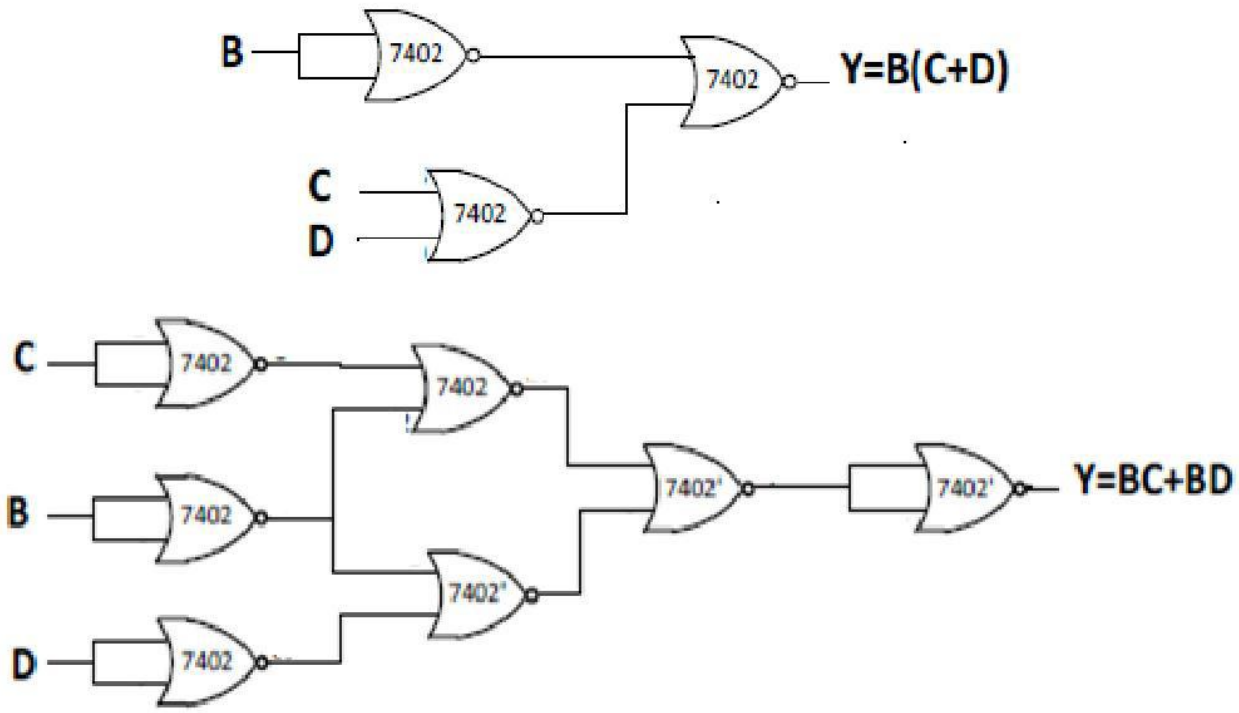


(ii) Simplification using NAND Gate:





(iii) Simplification using NOR Gate:



PROCEDURE

1. Verify that the gates are working.
2. Construct a truth table for the given problem.
3. Draw a Karnaugh's Map corresponding to the given truth table.
4. Simplify the given Boolean expression manually using the Karnaugh's Map.

a. Implementation Using Logic Gates:

1. Realize the simplified expression using logic gates.
2. Connect V_{CC} and ground as shown in the pin diagram.
3. Make connections as per the logic gate diagram.
4. Apply the different combinations of input according to the truth tables.
5. Verify that the results are correct.

b. **Implementation Using Universal Gates:**

1. Convert the AND-OR logic into NAND-NAND and NOR-NOR logic.
2. Realize the simplified Boolean expressions using only NAND gates, and Then by using only NOR gates.
3. Connect the circuits according to the circuit diagrams, apply inputs according to the truth table and verify the results.

RESULTS: - The given design example is implemented using NAND gates only.

QUESTIONS

1. Define universal gates?
2. Which IC is used for NAND and NOR gates?
3. Why NAND-NOR gates are called as universal gate?
4. Design XOR gate using NAND gate.
5. Design NAND gate using NOR gate.

EXPERIMENT NO: -5

Implementation and verification of truth table of 4x1 Multiplexer and 1x4 Demultiplexer.

AIM: -

To implementation and verify of truth table of 4x1 Multiplexer and 1x4 Demultiplexer.

APPARATUS REQUIRED: -

SL NO.	COMPONENT	SPECIFICATION	QUANTITY
1	4x1 MULTIPLEXER	IC 74153	1
2	1x4 DEMULTIPLEXER	IC 74156	1
3	BREAD BOARD	-	1
4	RESISTOR	220Ω	1
5	BATTERY	9V	1
6	LED	-	1
7	CONNECTING WIRE	-	AS PER REQUIREMENT

THEORY: -

MULTIPLEXER:

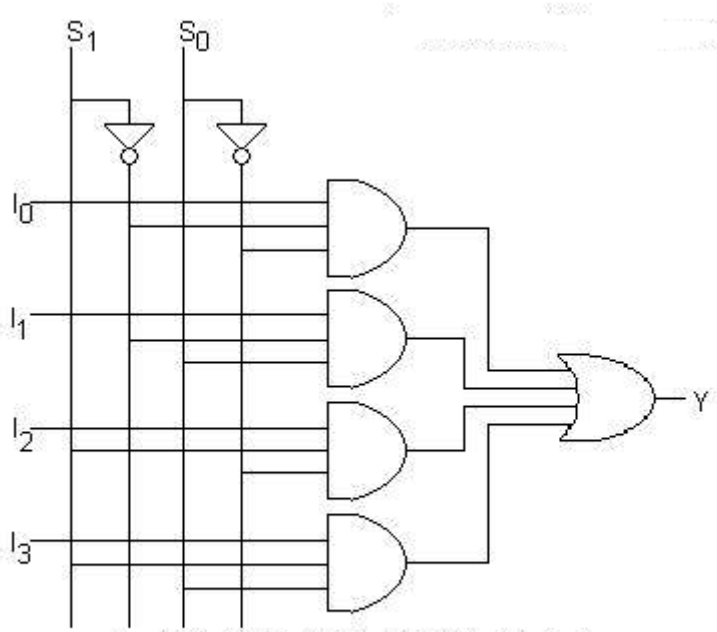
Multiplexer generally means many into one. A multiplexer is a circuit with many inputs but only one output. By applying control signals we can steer any input to the output. The circuit has n-input signal, control signal (m) & one output signal, where $2^n = m$. One of the popular multiplexers is the 16 to 1 multiplexer, which has 16 input bits, 4 control bits & 1 output bit.

4x1 Multiplexer

The 4x1 Multiplexer has four input lines I_0, I_1, I_2 and I_3 and one output line Y. The selection of a particular input is controlled by set of selection lines, S_1 and S_0.

TRUTH TABLE

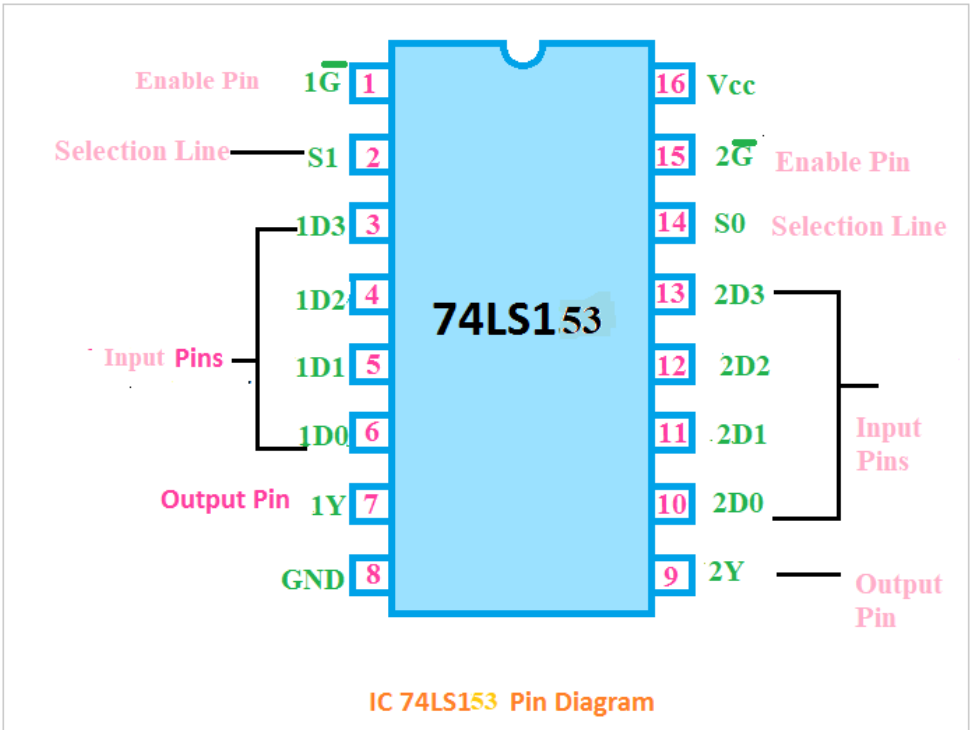
Selection line		Output
S_1	S_0	Y
0	0	I_0
0	1	I_1
1	0	I_2
1	1	I_3



$Y = I_0 S'_1 S'_0 + I_1 S'_1 S_0 + I_2 S_1 S'_0 + I_3 S_1 S_0$

CIRCUIT DIAGRAM OF 4X1 MUX USING LOGIC GATES

PIN DIAGRAM OF IC 74153



IC 74LS153 Pin Diagram

IC 74153 Truth Table

The truth table of IC 74153 is given below

Select Inputs		Inputs (a or b)					Output
S_0	S_1	$\overline{E}$	I_0	I_1	I_2	I_3	Z
X	X	H	X	X	X	X	L
L	L	L	L	X	X	X	L
L	L	L	H	X	X	X	H
H	L	L	X	L	X	X	L
H	L	L	X	H	X	X	H
L	H	L	X	X	L	X	L
L	H	L	X	X	H	X	H
H	H	L	X	X	X	L	L
H	H	L	X	X	X	H	H

H=High voltage level
L=Low voltage level
X=Don't care

DEMULTIPLEXER:

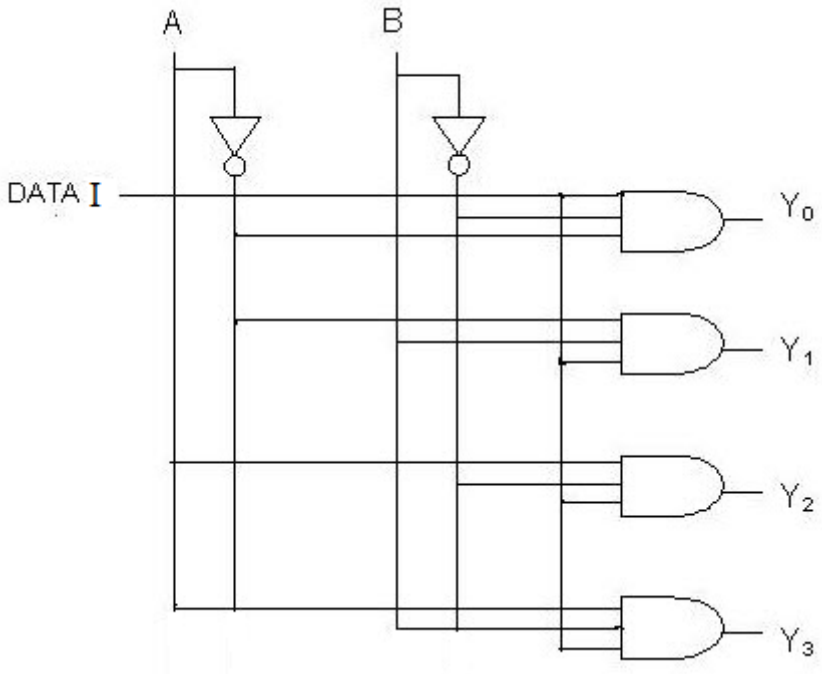
Demultiplexer means generally one into many. A demultiplexer is a logic circuit with one input and many outputs. By applying control signals, we can steer the input signal to one of the output lines. The circuit has one input signal, m control signal and n output signals, where$2^n = m$. It functions as an electronic switch to route an incoming data signal to one of several outputs.

1x4 Demultiplexer

The 1x4 Demultiplexer has one input I and four outputs Y_0, Y_1, Y_2 and Y_3.

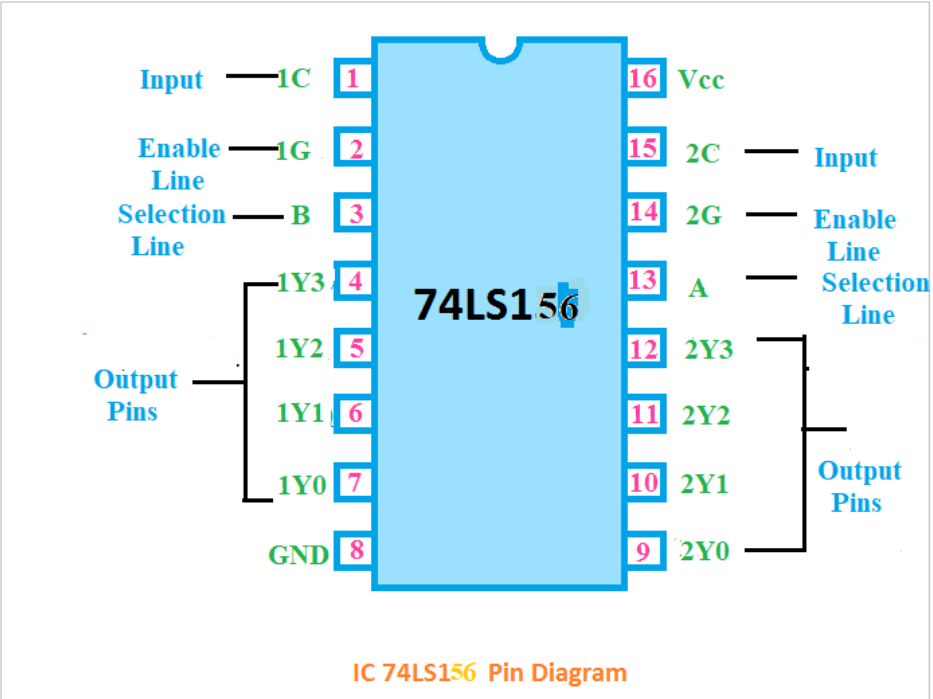
TRUTH TABLE

Input	Selection line		Output			
I	S_1	S_0	Y_3	Y_2	Y_1	Y_0
I	0	0	0	0	0	I
I	0	1	0	0	I	0
I	1	0	0	I	0	0
I	1	1	I	0	0	0



CIRCUIT DIAGRAM OF 1 × 4 DEMUX

PIN DIAGRAM OF IC 74156



IC 74LS156 Pin Diagram

PROCEDURE: -

1. Connect the circuit as shown in figure.
2. Apply V_{CC} & ground signal to the IC.
3. Observe the input & output according to the truth table.

PRECAUTIONS: -

1. Make the connections according to the IC pin diagram.
2. The connections should be tight.
3. The V_{CC} and ground should be applied carefully at the specified pin only.

OBSERVATION: - [L=logic 0, H=logic 1]

4x1 Multiplexer

Input line				Selection line		Output
I_0	I_1	I_2	I_3	S_1	S_0	Y
H	L	L	H	L	L	
L	L	H	H	L	H	
H	H	L	L	H	L	
L	H	L	H	H	H	

1x4 Demultiplexer

Input	Selection line		Output			
I	S_1	S_0	Y_3	Y_2	Y_1	Y_0
H	L	L				
H	L	H				
H	H	L				
H	H	H				

RESULT: - Hence, the truth table of 4x1 Multiplexer and 1x4 Demultiplexer are verified and found ok.

EXPERIMENT NO: -6

Verification of State Tables of RS, J-K and D Flip-Flops.

AIM

To Verify of State Tables of RS, J-K, and D Flip-Flops.

APPARATUS REQUIRED: -

SL NO.	COMPONENT	SPECIFICATION	QUANTITY
1	D Flip Flop	IC 7474	1
2	JK Flip Flop	IC 7476	1
3	BREAD BOARD	-	1
4	RESISTOR	220Ω	1
5	BATTERY	9V	1
6	LED	-	1
7	CONNECTING WIRE	-	AS PER REQUIREMENT

THEORY: -

RS FLIP-FLOP: There are two inputs to the flip-flop defined as R and S. When I/Ps R = 0 and S = 0 then O/P remains unchanged. When I/Ps R = 0 and S = 1 the flip-flop is switches to the stable state where O/P is 1 i.e. SET. The I/P condition is R = 1 and S = 0 the flip-flop is switched to the stable state where O/P is 0 i.e. RESET. The I/P condition is R = 1 and S = 1 the flip-flop is switched to the stable state where O/P is forbidden.

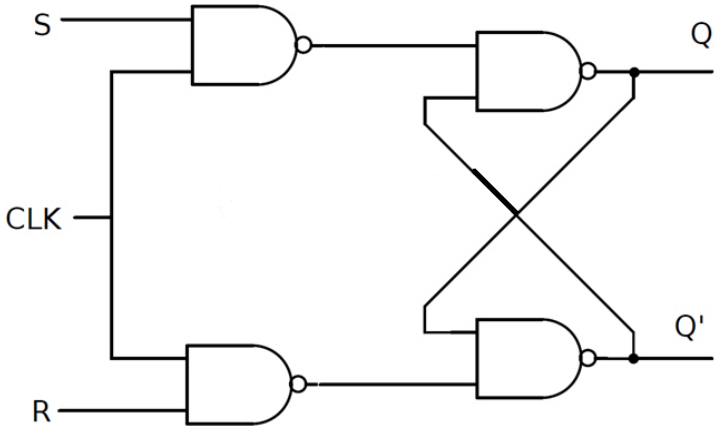
JK FLIP-FLOP: For purpose of counting, the JK flip-flop is the ideal element to use. The variable J and K are called control I/Ps because they determine what the flip- flop does when a positive edge arrives. When J and K are both 0s, both AND gates are disabled and Q retains its last value.

D FLIP –FLOP: This kind of flip flop prevents the value of D from reaching the Q output until clock pulses occur. When the clock is low, both AND gates are disabled D can change value without affecting the value of Q. On the other hand, when the clock is high, both AND gates are enabled. In this case, Q is forced to equal the value of D. When the clock again goes low, Q retains or stores the last value of D. a D flip flop is a bistable circuit whose D input is transferred to the output after a clock pulse is received.

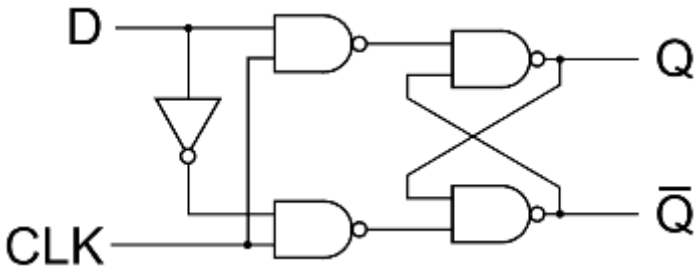
T FLIP-FLOP: The T or "toggle" flip-flop changes its output on each clock edge, giving an output which is half the frequency of the signal to the T input. It is useful for constructing binary counters, frequency dividers, and general binary addition devices. It can be made from a J-K flip-flop by tying both of its inputs high.

CIRCUIT DIAGRAM:

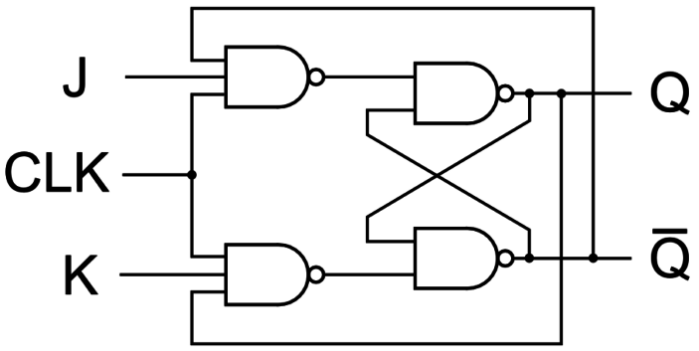
RS FLIP-FLOP



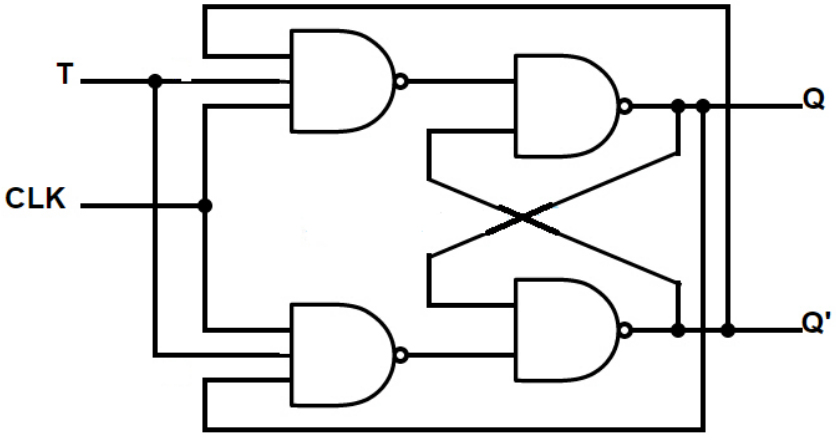
D FLIP –FLOP



JK FLIP-FLOP

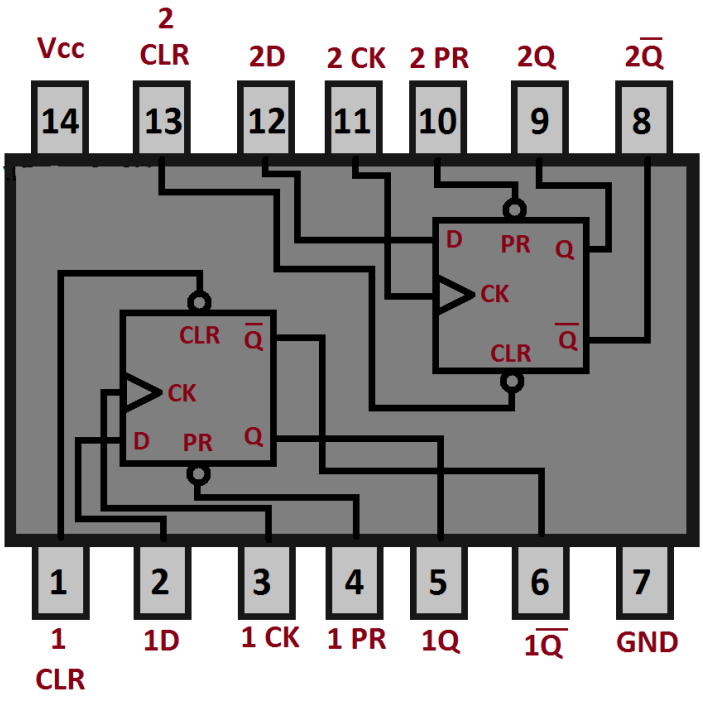


T FLIP-FLOP



IC 7474 Pinout Diagram

The pin diagram of IC 7474 is shown below



IC 7474 Pinout Diagram

IC 7474 or mostly known as IC 74LS74 is a dual D Flip Flop positive edge-triggered IC. It has two independent D Flip Flops with complementary outputs. It has D input and Q output. The data in the D input may be changed during the high or low clock but it does not affect the output and the delay times also do not affect. The IC 7474 can be operated up to 7V voltage and 0 to +70 degrees centigrade temperature. The main features of the IC 74LS74 are, it provides very fast switching, low propagation delay, large operating mode, etc.

IC 7474 truth table

The truth table of IC 7474 is given below

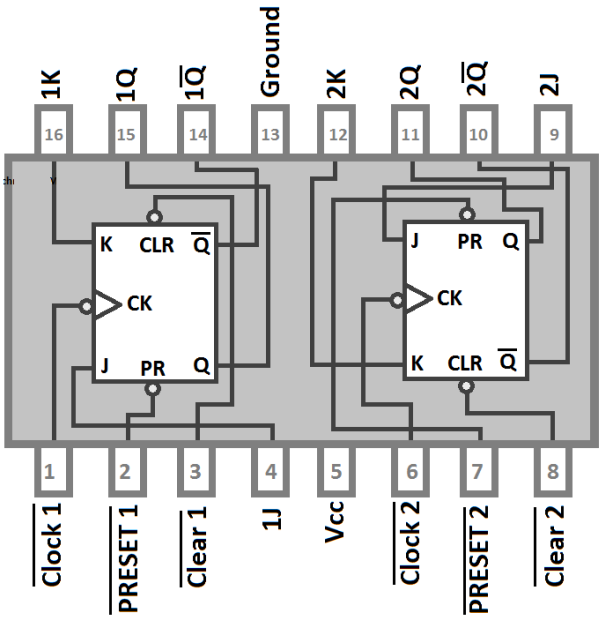
INPUTS				OUTPUTS	
PR	CLR	CLK	D	Q	$\overline{Q}$
L	H	X	X	H	L
H	L	X	X	L	H
L	L	X	X	H	H
H	H	↑	H	H	L
H	H	↑	L	L	H
H	H	L	X	Q_0	$\overline{Q_0}$

H-High logic level, X-Either LOW or HIGH logic level, L-LOW logic level, ↑-Positive going transition of the clock

IC 7474 Truth Table

IC 7476 Pinout Diagram

The pin diagram of IC 7476 is shown below



IC 7476 Pinout Diagram

The IC 7476 commonly known as IC 74LS76 is a dual JK flip IC with Set and Clear Input. The Clock, Clear inputs are active low inputs. During the High-Low Clock Transition, input data is transferred to the output. It is also a 16 pin IC. The IC 7476 can be operated from 4.75 to 5.25V and 0 to +70 degrees centigrade temperature. It works with standard TTL voltage and provides a very fast switching speed.

IC 7476 truth table
The truth table of IC 7476 is given below

INPUTS					OUTPUTS	
PR	CLR	CLK	J	K	Q	$\overline{Q}$
L	H	X	X	X	H	L
H	L	X	X	X	L	H
L	L	X	X	X	H	H
H	H	↑	L	L	Q_0	$\overline{Q_0}$
H	H	↑	H	L	H	L
H	H	↑	L	H	L	H
H	H	↑	H	H	Toggle	

H-High logic level, X-Either LOW or HIGH logic level, L-LOW logic level, ↑-Positive going transition of the clock

IC 7476 Truth Table

PROCEDURE: -

1. Connect the circuit as shown in figure.
2. Apply V_{CC} & ground signal to the IC.
3. Observe the input & output according to the truth table.

RESULT: -

The observation table of RS, J-K, and D Flip-Flops is verified.

PRECAUTIONS: -

1. Make the connections according to the IC pin diagram.
2. The connections should be tight.
3. The V_{CC} and ground should be applied carefully at the specified pin only.
4. Swich off supply after completing the experiment.

EXPERIMENT NO: -7

Realization and study of all types of Shift Registers.

AIM

To realize and study of all type Shift Registers. 1) SISO (Serial in Serial out) 2) SIPO (Serial in Parallel out) 3) PIPO (Parallel in Parallel out) 4) PISO (Parallel in Serial out)

APPARATUS REQUIRED: -

SL NO.	COMPONENT	SPECIFICATION	QUANTITY
1	SHIFT REGISTER	IC 7495	1
2	BREAD BOARD	-	1
3	RESISTOR	220Ω	1
4	BATTERY	9V	1
5	LED	-	1
6	CONNECTING WIRE	-	AS PER REQUIREMENT

THEORY: -

Shift registers are a type of sequential logic circuit, mainly for storage of digital data. They are a group of flip-flops connected in a chain so that the output from one flip-flop becomes the input of the next flip-flop. All the flip-flops are driven by a common clock, and all are set or reset simultaneously.

The serial in/serial out shift register accepts data serially – that is, one bit at a time on a single line. It produces the stored information on its output also in serial form.

The serial in/parallel out shift register accepts data serially – that is, one bit at a time on a single line. It produces the stored information on its output in parallel form.

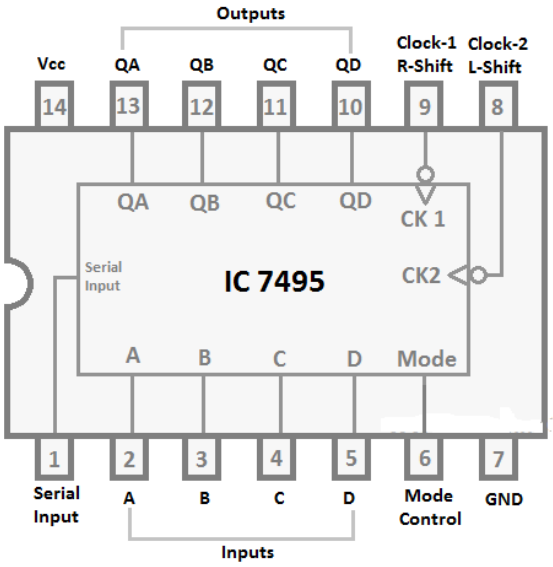
The parallel in/serial out shift register accepts data in parallel. It produces the stored information on its output also in serial form.

The parallel in/parallel out shift register accepts data in parallel. It produces the stored information on its output in parallel form.

Shift Registers using IC 7495

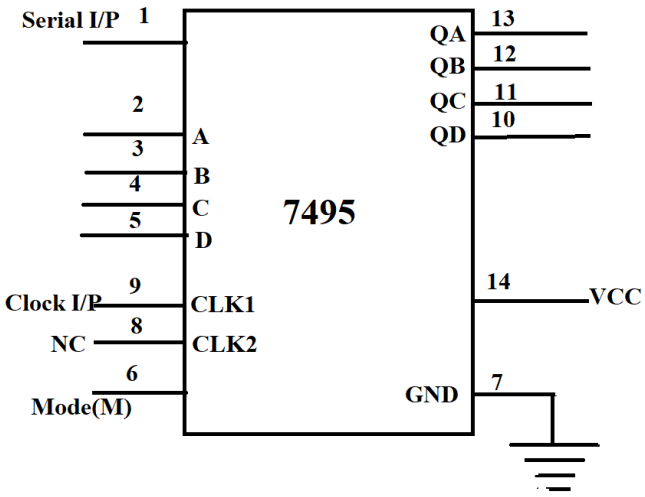
IC 7495 is a shift register IC. It is also known as IC74LS95. It is a 4-bit device and having both serial and parallel synchronous operating modes. The main application of this IC 7495 is encryption and decryption in digital circuits.

IC 7495 pin diagram



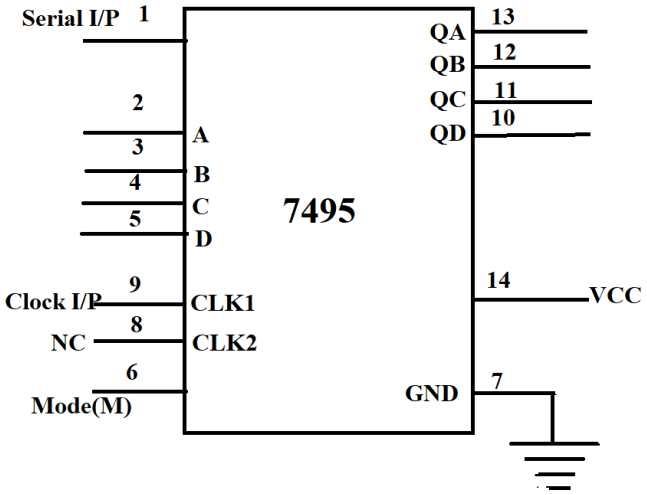
IC 7495 Pin Diagram

SIPO



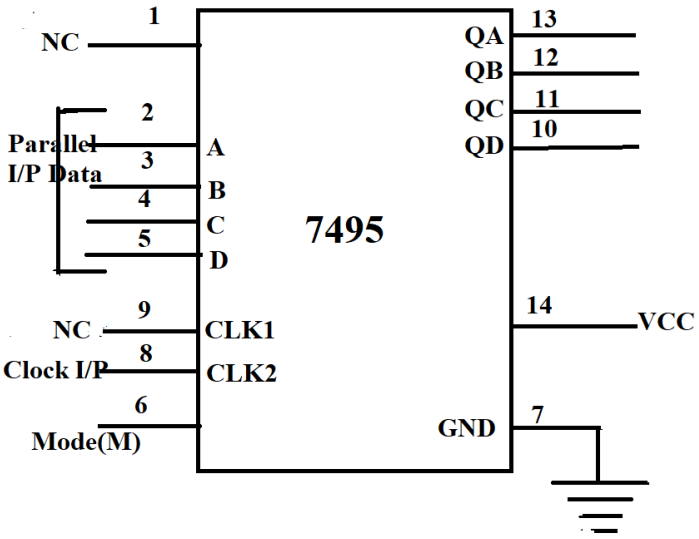
Clock	Serial I/P	Q_A	Q_B	Q_C	Q_D
1	0	0	X	X	X
2	1	1	0	X	X
3	1	1	1	0	X
4	1	1	1	1	0

SISO



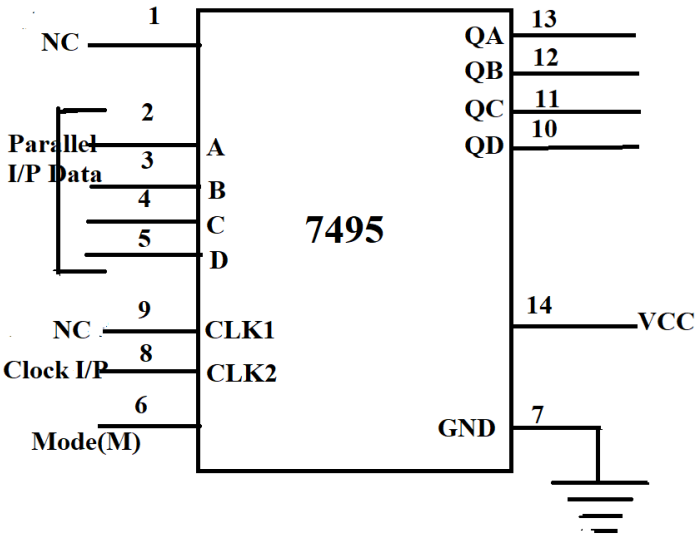
Clock	Serial I/P	Q_A	Q_B	Q_C	Q_D
1	$d_0 = 0$	0	X	X	X
2	$d_1 = 1$	1	0	X	X
3	$d_2 = 1$	1	1	0	X
4	$d_3 = 1$	1	1	1	$d_0 = 0$
5	X	X	1	1	$d_1 = 1$
6	X	X	X	1	$d_2 = 1$
7	X	X	X	X	$d_3 = 1$

PISO



Mode	Clock	Parallel I/P				Parallel O/P			
		A	B	C	D	Q_A	Q_B	Q_C	Q_D
1	1	1	0	1	1	1	0	1	1
0	2	X	X	X	X	X	1	0	1
0	3	X	X	X	X	X	X	1	0
0	4	X	X	X	X	X	X	X	1

PIPO



Clock	Parallel I/P				Parallel O/P			
	A	B	C	D	Q_A	Q_B	Q_C	Q_D
1	1	0	1	1	1	0	1	1

PROCEDURE:

Serial In Parallel Out (SIPO): -

1. Connections are made as per circuit diagram.
2. Keep the mode control in logic 0.
3. Apply the data at serial input.
4. Apply one clock pulse at clock 1 observe this data at Q_A.
5. Apply the next data at serial input.
6. Apply one clock pulse at clock 2, observe that the data on Q_A will shift to Q_B and the new data applied will appear at Q_A.

7. Repeat steps 2 and 3 till all the 4 bits data appear at the output of shift register.

Serial In Serial Out (SISO): -

1. Connections are made as per circuit diagram.
2. Keep the mode control in logic 0.
3. Load the shift register with 4 bits of data one by one serially.
4. At the end of 4th clock pulse the first data 'd0' appears at Q_D.
5. Apply another clock pulse; the second data 'd1' appears at Q_D and so on.
6. Thus, the data applied serially at the input comes out serially at Q_D.

Parallel In Serial Out (PISO): -

1. Connections are made as per circuit diagram.
2. Apply the desired 4-bit data at A, B, C and D.
3. Keeping the mode control M=1 apply one clock pulse. The data applied at A, B, C and D will appear at Q_A, Q_B, $Q_C$and Q_D respectively.
4. Now mode control M=0. Apply clock pulses one by one and observe the Data coming out serially at Q_D.

Parallel In Parallel Out (PIPO): -

1. Connections are made as per circuit diagram.
2. Apply the 4-bit data at A, B, C and D.
3. Apply one clock pulse at Clock 2 (Note: Mode control M=1).
4. The 4-bit data at A, B, C and D appears at Q_A, Q_B, $Q_C$and Q_D respectively.

RESULT: -

The operation of all types of shift registers are verified.

PRECAUTIONS: -

1. Make the connections according to the IC pin diagram.
2. The connections should be tight.
3. The V_{CC} and ground should be applied carefully at the specified pin only.
4. Swich off supply after completing the experiment.

EXPERIMENT: 8

STUDY OF COUNTERS

I. STUDY OF ASYNCHRONOUS COUNTER

AIM: -

To design and test 3-bit binary asynchronous counter using flip-flop IC 7476.

APPARATUS REQUIRED: -

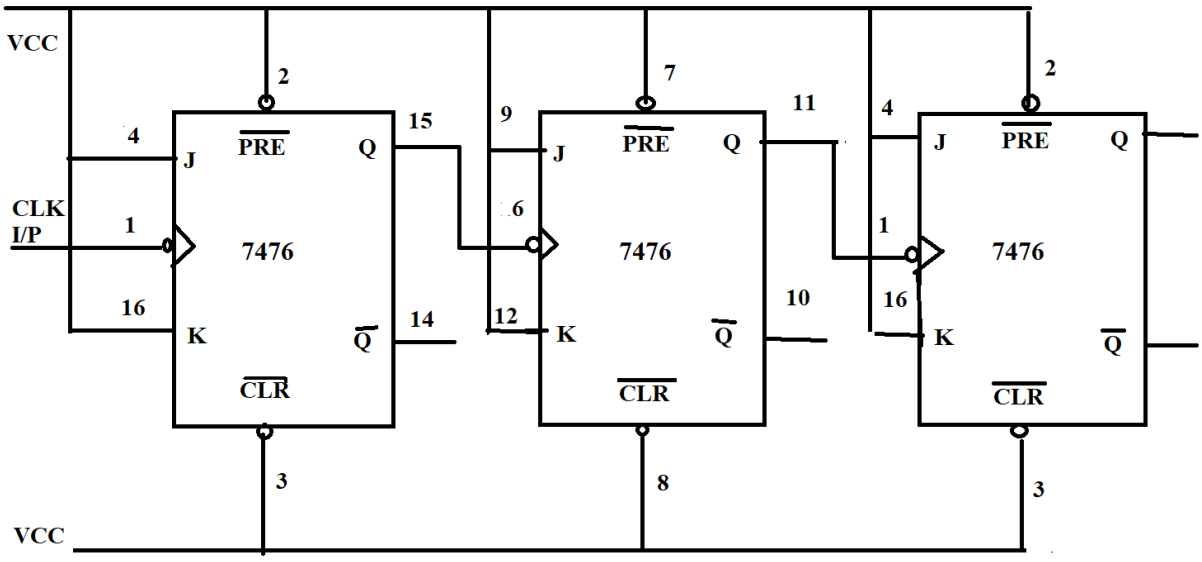
SL NO.	COMPONENT	SPECIFICATION	QUANTITY
1	JK Flip Flop	IC 7476	1
2	BREAD BOARD	-	1
3	RESISTOR	220Ω	1
4	BATTERY	9V	1
5	LED	-	1
6	CONNECTING WIRE	-	AS PER REQUIREMENT

THEORY: -

A counter in which each flip-flop is triggered by the output goes to previous flip-flop. As all the flip-flops do not change state simultaneously spike occur at the output. To avoid this, strobe pulse is required. Because of the propagation delay the operating speed of asynchronous counter is low. Asynchronous counters are easy and simple to construct.

Design:

MOD-8 UP COUNTER



3-bit asynchronous up counter

3-bit Asynchronous up counter			
Clock	**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: -

1. Connect the circuit as shown in figure.
2. Apply V_{CC} & ground signal to the IC.
3. Apply various input data to the logic circuit.
4. Observe the input & output according to the truth table.

RESULT: -

The 3-bit binary asynchronous counter using flip-flop IC 7476 is verified.

PRECAUTIONS: -

1. Make the connections according to the IC pin diagram.
2. The connections should be tight.
3. The V_{CC} and ground should be applied carefully at the specified pin only.
4. Switch off supply after completing the experiment.

II. STUDY OF SYNCHRONOUS COUNTER

AIM: -

To design and test 3-bit binary synchronous counter using flip-flop IC 7476.

APPARATUS REQUIRED: -

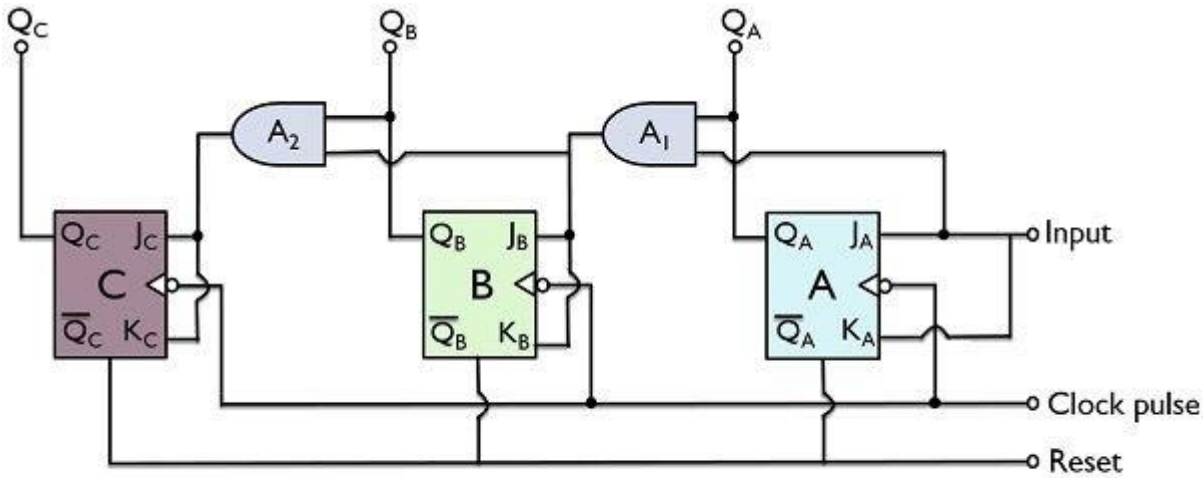
SL NO.	COMPONENT	SPECIFICATION	QUANTITY
1	JK Flip Flop	IC 7476	1
2	BREAD BOARD	-	1
3	RESISTOR	220Ω	1
4	BATTERY	9V	1
5	LED	-	1
6	CONNECTING WIRE	-	AS PER REQUIREMENT

THEORY: -

A counter in which each flip-flop is triggered by the output goes to previous flipflop. As all the flip-flops do not change states simultaneously in asynchronous counter, spike occur at the output. To avoid this, strobe pulse is required. Because of the propagation delay the operating speed of asynchronous counter is low. This problem can be solved by triggering all the flip-flops in synchronous with the clock signal and such counters are called synchronous counters.

Design:

MOD-8 UP COUNTER



Circuit Diagram of 3-bit Synchronous Counter

3-bit Asynchronous up counter			
Clock	**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: -

1. Connect the circuit as shown in figure.
2. Apply V_{CC} & ground signal to the IC.
3. Apply various input data to the logic circuit.
4. Observe the input & output according to the truth table.

RESULT: -

The 3-bit binary synchronous counter using flip-flop IC 7476 is verified.

PRECAUTIONS: -

1. Make the connections according to the IC pin diagram.
2. The connections should be tight.
3. The V_{CC} and ground should be applied carefully at the specified pin only.
4. Switch off supply after completing the experiment.

EXPERIMENT NO: -9

Design 4-bit binary parallel adder using IC 7483.

AIM

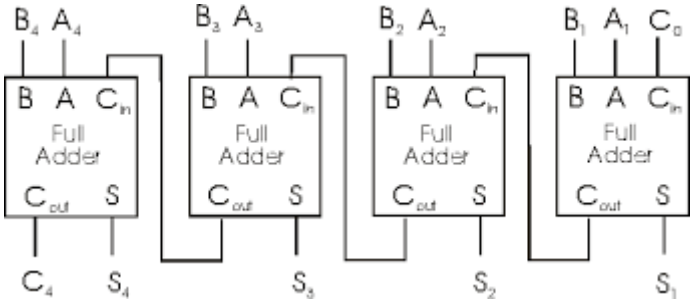
To design 4-bit binary parallel adder using IC 7483.

APPARATUS REQUIRED

SL NO.	COMPONENT	SPECIFICATION	QUANTITY
1	4-BIT ADDER	IC 7483	1
2	BREAD BOARD	-	1
3	RESISTOR	220Ω	1
4	BATTERY	9V	1
5	LED	-	1
6	CONNECTING WIRE	-	AS PER REQUIREMENT

THEORY

A 4-bit adder is a circuit which adds two 4-bits numbers, say, A and B. In addition, a 4-bit adder will have another single-bit input which is added to the two numbers called the carry-in (C_{in}). The output of the 4-bit adder is a 4-bit sum (S) and a carry-out (C_{out}) bit.

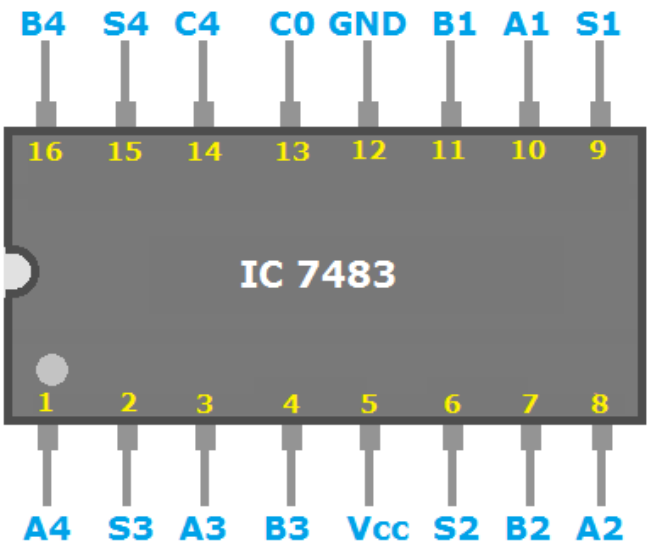


4-bit adder using IC 7483

IC 7483 is a digital adder IC that can add two 4-bit numbers. IC 7483 consists of four individual full adder circuits which are internally connected. It has also input and output carry circuit. IC 7483 can be connected directly to the CMOS, NMOS, and TTL circuits.

IC 7483 Pin Diagram

The pin diagram of IC 7483. It has a total of 16 pins.



IC 7483 Pin Diagram

IC 7483 Truth Table

This truth table shows the two 4-bit numbers as input and 4-bit output numbers with output carry.

Input Data A				Input Data B				Addition				
A4	A3	A2	A1	B4	B3	B2	B1	C_{out}	S4	S3	S2	S1
1	0	0	0	0	0	1	0	0	1	0	1	0
1	0	0	0	1	0	0	0	1	0	0	0	0
0	0	1	0	1	0	0	0	0	1	0	1	0
0	0	0	1	0	1	1	1	0	1	0	0	0
1	0	1	0	1	0	1	1	1	0	1	0	1
0	1	1	0	0	0	1	1	0	1	0	0	1
1	1	1	0	1	1	1	1	1	1	1	0	1
1	0	1	0	1	1	0	1	1	0	1	1	1

PROCEDURE –

a) Make the connections as per the logic diagram.

b) Connect +5v and ground according to pin configuration.

c) Apply diff combinations of inputs to the i/p terminals.

d) Note o/p for summation.

e) Verify the truth table.

RESULT- Binary 4-bit full adder is studied and verified.

PRECAUTIONS:

1. Make the connections according to the IC pin diagram.
2. The connections should be tight.
3. V_{CC} and ground should be applied carefully at the specified pin only.

QUESTIONS

1. What do you understand by parallel adder?
2. What is IC no. of parallel adder?
3. How many bits can be added using IC7483 parallel adder?
4. Can we obtain 1's complement using parallel adder?
5. Can you obtain subtractor using parallel adder?

VHDL SIMULATION OF EXPERIMENTS USING MODELSIM SOFTWARE

PROGRAM 1(a)

AIM: - To write a program in VHDL for implementing the digital logic gates-AND, OR, NOT, NAND, NOR, XOR, XNOR and to verify the functionality.

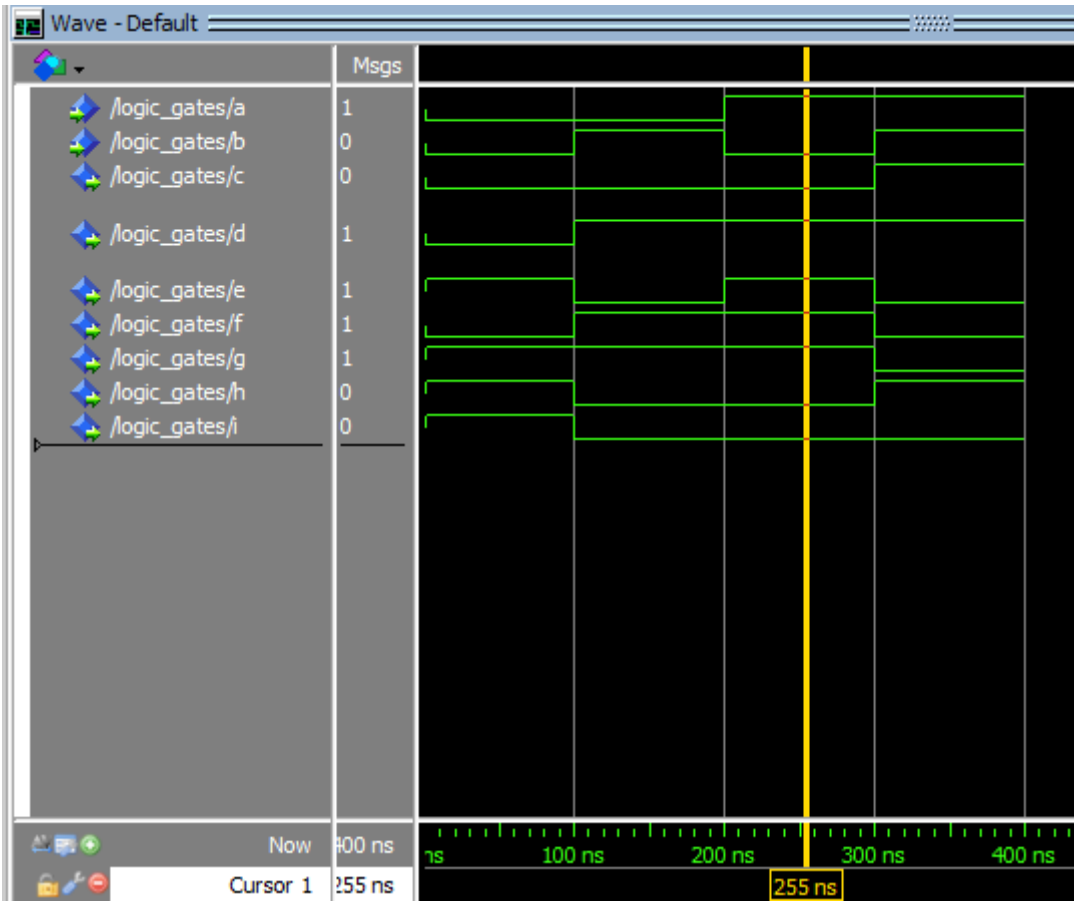
PROGRAM: -

```
library ieee;
use ieee.std_logic_1164.all;
entity logic_gates is
port(a,b:in std_logic;
c,d,e,f,g,h,i:out std_logic);
end logic_gates;
architecture dataflow of logic_gates is
begin
c<=a and b;
d<=a or b;
e<=not b;
f<=a xor b;
g<=a nand b;
h<=a xnor b;
i<=a nor b;
end dataflow;
```

TRUTH TABLE

Inputs		Outputs						
a	b	c	D	e	F	g	h	i
0	0	0	0	1	0	1	1	1
0	1	0	1	0	1	1	0	0
1	0	0	1	1	1	1	0	0
1	1	1	1	0	0	0	1	0

SIMULATION WAVEFORM



OBSERVATION

	Inputs		Outputs						
Timing (ns)	a	b	c	d	E	f	g	h	i
0-100	0	0	0	0	1	0	1	1	1
100-200	0	1	0	1	0	1	1	0	0
200-300	1	0	0	1	1	1	1	0	0
300-400	1	1	1	1	0	0	0	1	0

CONCLUSION: - Hence all the logic gates are implemented in VHDL and their functionality is verified.

PROGRAM 1(b)

AIM: - To write a program in VHDL for implementing NAND gate as universal gate and to verify the functionality.

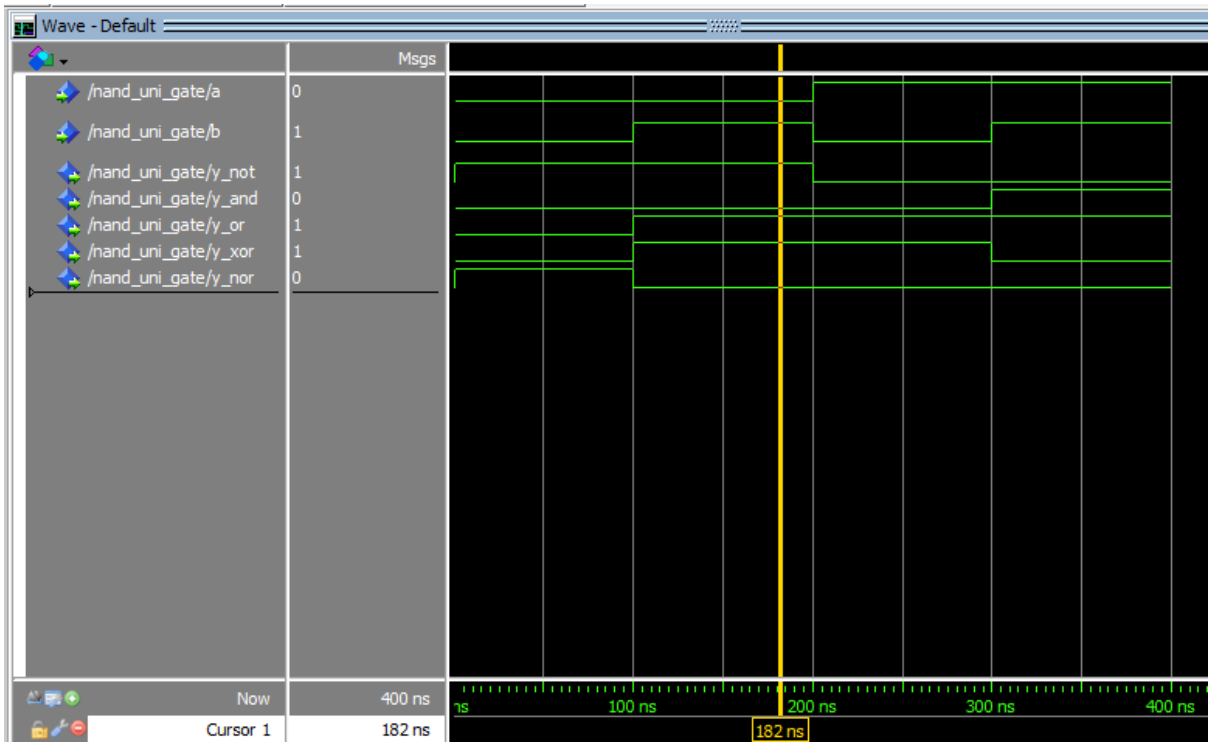
PROGRAM: -

```
library ieee;
use ieee.std_logic_1164.all;
entity nand_uni_gate is
port(a,b:in std_logic;
y_not,y_and,y_or,y_xor,y_nor:out std_logic);
end nand_uni_gate;
architecture nand_uni_gate_arch of nand_uni_gate is
begin
y_not<=a nand a;
y_and<=(a nand b) nand (a nand b);
y_or<=(a nand a) nand (b nand b);
y_nor<=((a nand a) nand (b nand b)) nand ((a nand a) nand (b nand b));
y_xor<=(a nand (a nand b)) nand (b nand (a nand b));
end nand_uni_gate_arch;
```

TRUTH TABLE

Inputs		Outputs				
A	b	y_not	y_and	y_or	y_nor	y_xor
0	0	1	0	0	1	0
0	1	1	0	1	0	1
1	0	0	0	1	0	1
1	1	0	1	1	0	0

SIMULATION WAVEFORM



OBSERVATION

	Inputs		Outputs				
Timing (ns)	a	B	y_not	y_and	y_or	y_nor	y_xor
0-100	0	0	1	0	0	1	0
100-200	0	1	1	0	1	0	1
200-300	1	0	0	0	1	0	1
300-400	1	1	0	1	1	0	0

CONCLUSION: - Hence NAND gate as universal gate is implemented in VHDL and the functionality is verified.

PROGRAM 2(a)

AIM: - To write a program in VHDL for implementing Demorgan's theorem and to verify the functionality.

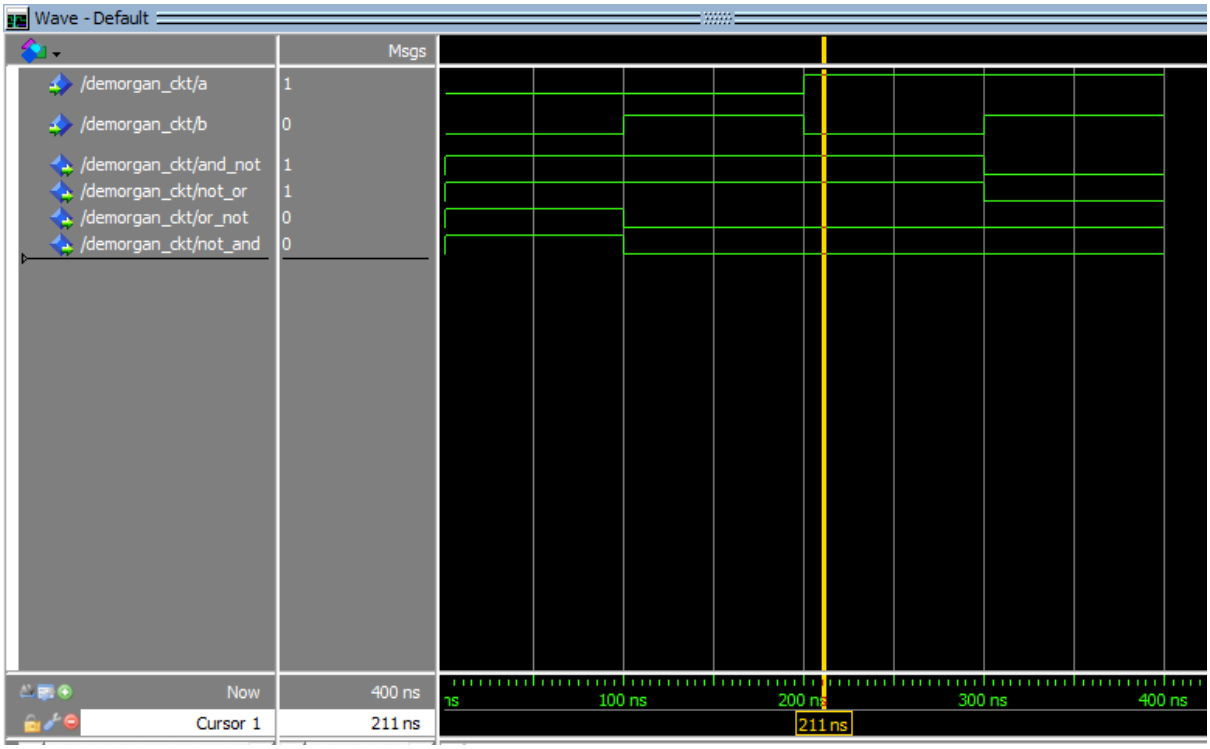
PROGRAM: -

```
library ieee;
use ieee.std_logic_1164.all;
entity Demorgan_ckt is
port(a,b:in std_logic;
and_not,not_or,or_not,not_and:out std_logic);
end Demorgan_ckt;
architecture Demorgan_arch of Demorgan_ckt is
begin
and_not<=not(a and b);
not_or<=(not a) or (not b);
or_not<=not(a or b);
not_and<=(not a) and (not b);
end  Demorgan_arch;
```

TRUTH TABLE

Inputs		Outputs			
a	b	and_not	not_or	or_not	not_and
0	0	1	1	1	1
0	1	1	1	0	0
1	0	1	1	0	0
1	1	0	0	0	0

SIMULATION WAVEFORM



OBSERVATION

	Inputs		Outputs			
Timing (ns)	A	B	and_not	not_or	or_not	not_and
0-100	0	0	1	1	1	1
100-200	0	1	1	1	0	0
200-300	1	0	1	1	0	0
300-400	1	1	0	0	0	0

CONCLUSION: - Hence Demorgan's theorem is implemented in VHDL and the functionality is verified.

PROGRAM 2(b)

AIM: - To write a program in VHDL for implementing the Boolean function $y = ab + a'b$ and to verify the functionality.

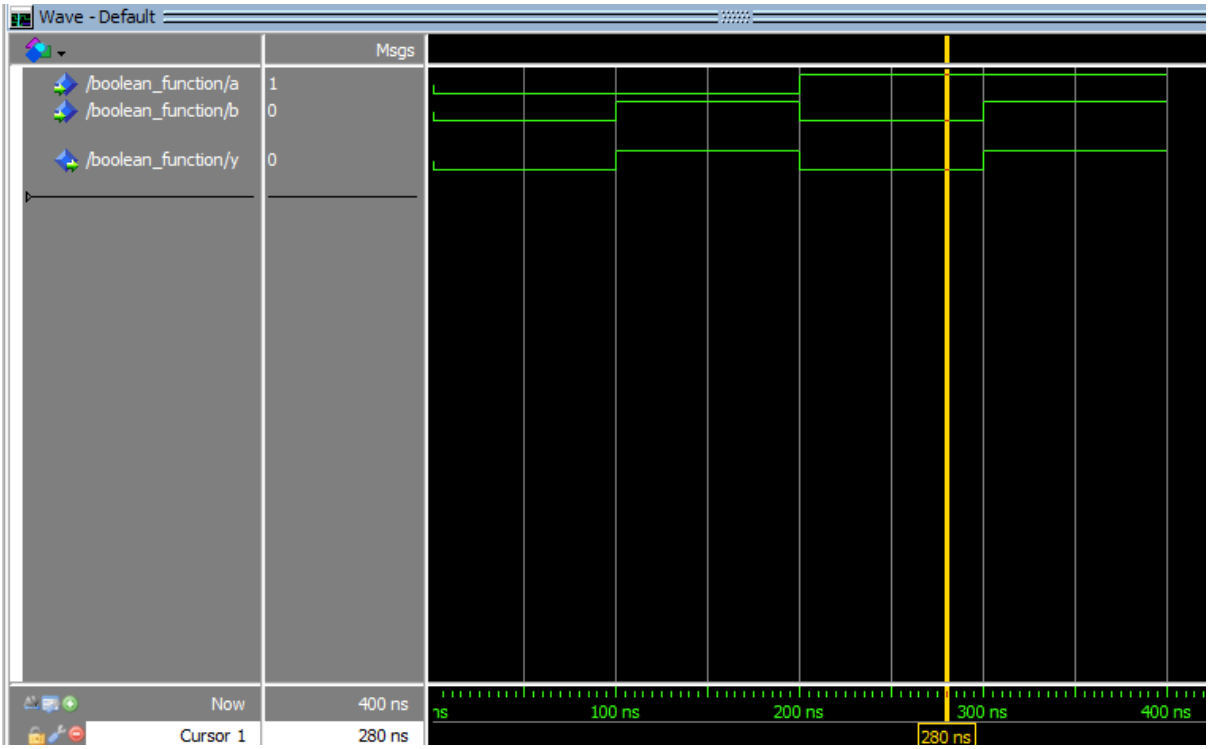
PROGRAM: -

```
library ieee;
use ieee.std_logic_1164.all;
entity boolean_function is
port(a,b:in std_logic;
y:out std_logic);
end boolean_function;
architecture boolean_function _arch of boolean_function is
begin
y<=(a and b) or ((not a) and b);
end boolean_function _arch;
```

TRUTH TABLE

Inputs		Output
a	b	y
0	0	0
0	1	1
1	0	0
1	1	1

SIMULATION WAVEFORM



OBSERVATION

	Inputs		Output
Timing (ns)	a	b	y
0-100	0	0	0
100-200	0	1	1
200-300	1	0	0
300-400	1	1	1

CONCLUSION: - Hence the given Boolean function is implemented in VHDL and the functionality is verified.

PROGRAM 3(a)

AIM: - To write a program in VHDL for implementing the half adder and to verify the functionality.

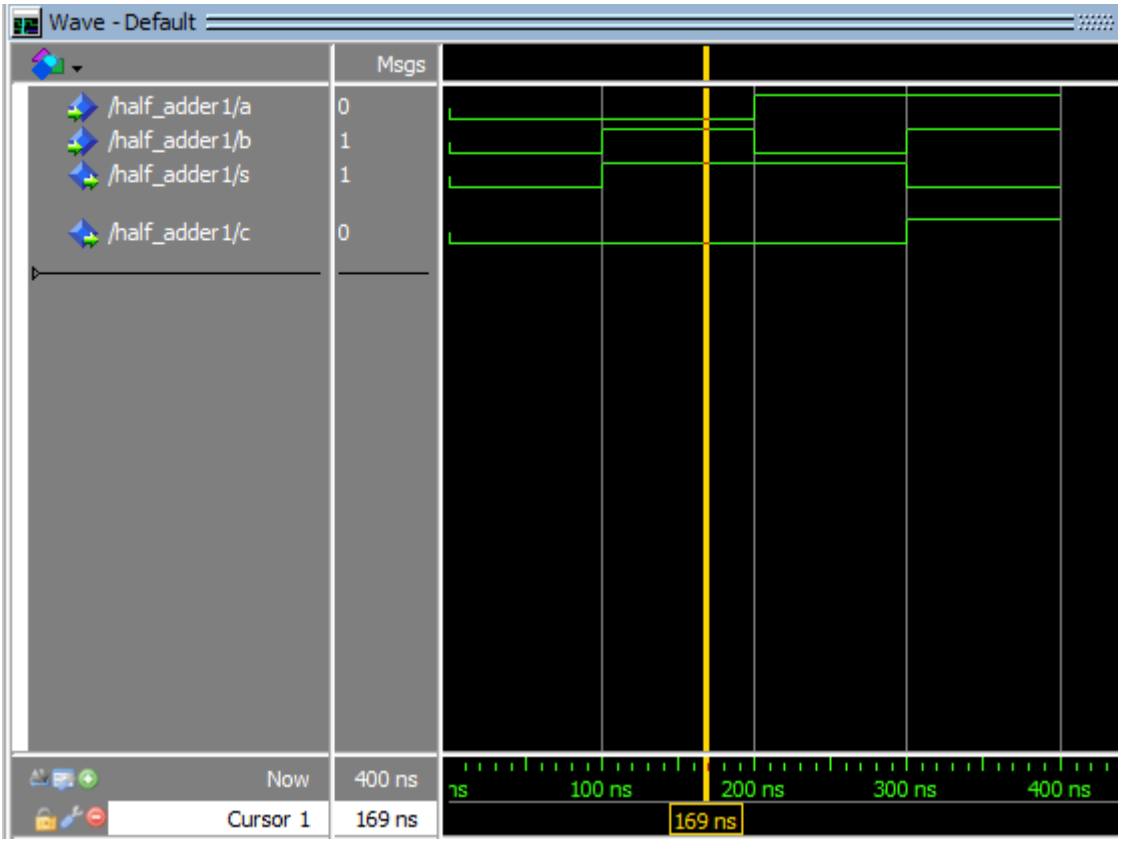
PROGRAM: -

```
library ieee;
use ieee.std_logic_1164.all;
entity half_adder is
port(a,b: in std_logic;
s,c: out std_logic);
end half_adder;
architecture dataflow of half_adder is
begin
s<= a xor b;
c<= a and b;
end dataflow;
```

TRUTH TABLE

Inputs		Outputs	
a	b	S	c
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

SIMULATION WAVEFORM



OBSERVATION

	Inputs		Outputs	
Timing (ns)	a	B	s	c
0-100	0	0	0	0
100-200	0	1	1	0
200-300	1	0	1	0
300-400	1	1	0	1

CONCLUSION: - Hence the half adder is implemented in VHDL and the functionality is verified.

PROGRAM 3(b)

AIM: - To write a program in VHDL for implementing the full adder and to verify the functionality.

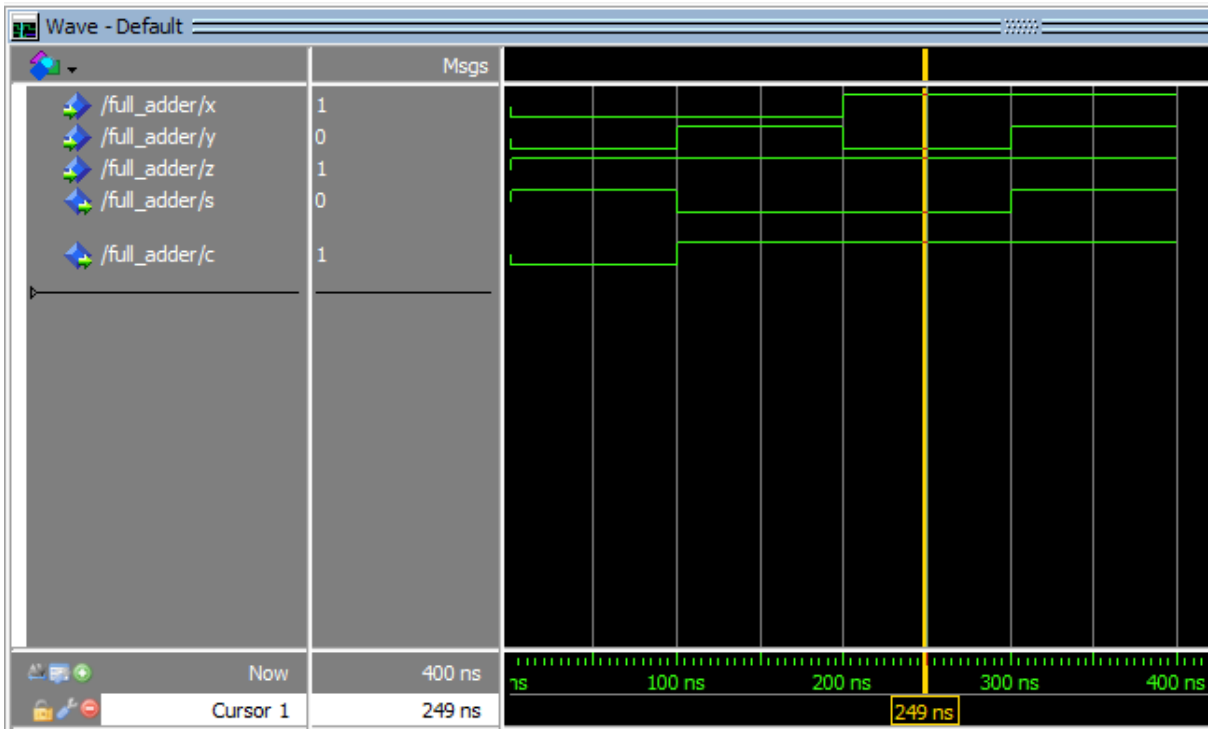
PROGRAM: -

```
library ieee;
use ieee.std_logic_1164.all;
entity full_adder is
port(x,y,z: in std_logic;
s,c: out std_logic);
end full_adder;
architecture dataflow of full_adder is
begin
s<= (x xor y)xor z;
c<= (x and y) or (y and z) or (z and x);
end dataflow;
```

TRUTH TABLE

Inputs			Outputs	
x	Y	z	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

SIMULATION WAVEFORM



OBSERVATION

	Inputs			Outputs	
Timing (ns)	x	y	Z	s	c
0-100	0	0	1	1	0
100-200	0	1	1	0	1
200-300	1	0	1	0	1
300-400	1	1	1	1	1

CONCLUSION: - Hence the full adder is implemented in VHDL and the functionality is verified.

PROGRAM 3(c)

AIM: - To write a program in VHDL for implementing the half subtractor and to verify the functionality.

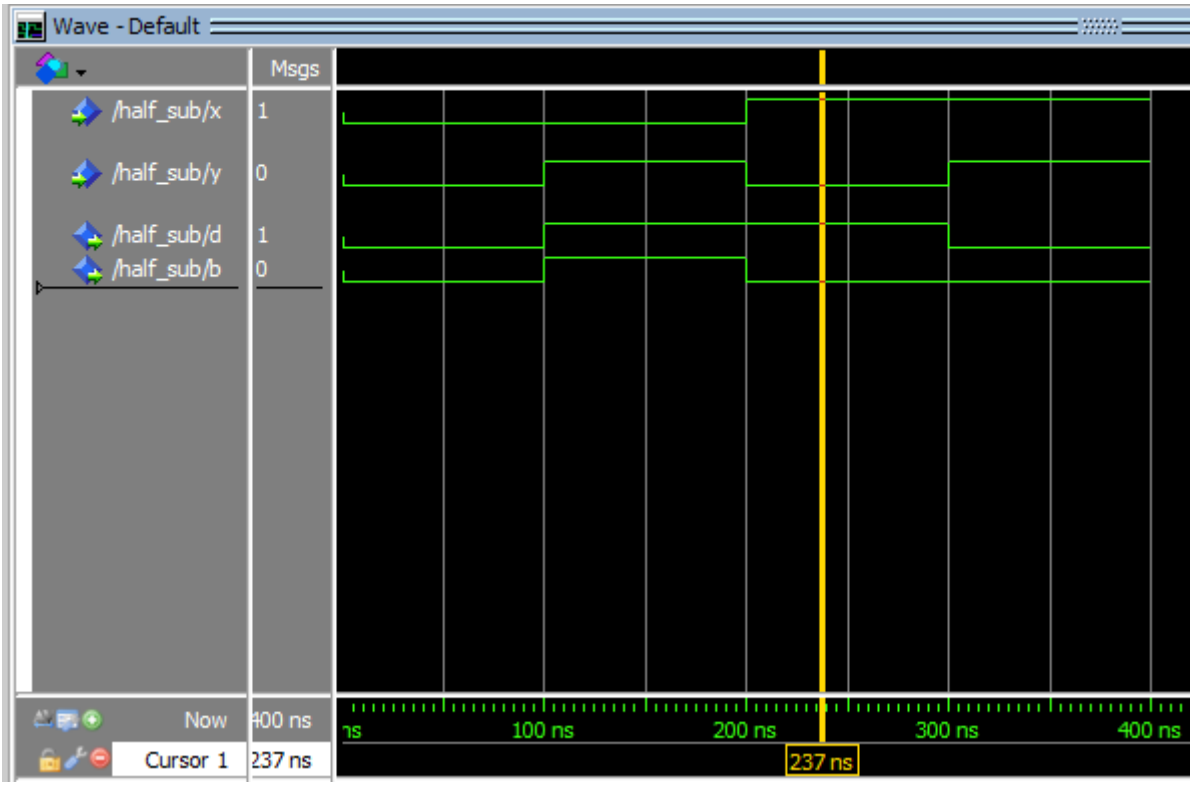
PROGRAM: -

```
library ieee;
use ieee.std_logic_1164.all;
entity half_subtractor is
port(x,y: in std_logic;
d,b: out std_logic);
end half_subtractor;
architecture dataflow of half_subtractor is
begin
d<= x xor y;
b<= (not x) and y;
end dataflow;
```

TRUTH TABLE

Inputs		Outputs	
x	y	D	B
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0

SIMULATION WAVEFORM



OBSERVATION

	Inputs		Outputs	
Timing (ns)	x	Y	d	b
0-100	0	0	0	0
100-200	0	1	1	1
200-300	1	0	1	0
300-400	1	1	0	0

CONCLUSION: - Hence the half subtractor is implemented in VHDL and the functionality is verified.

PROGRAM 3(d)

AIM: - To write a program in VHDL for implementing the full subtractor and to verify the functionality.

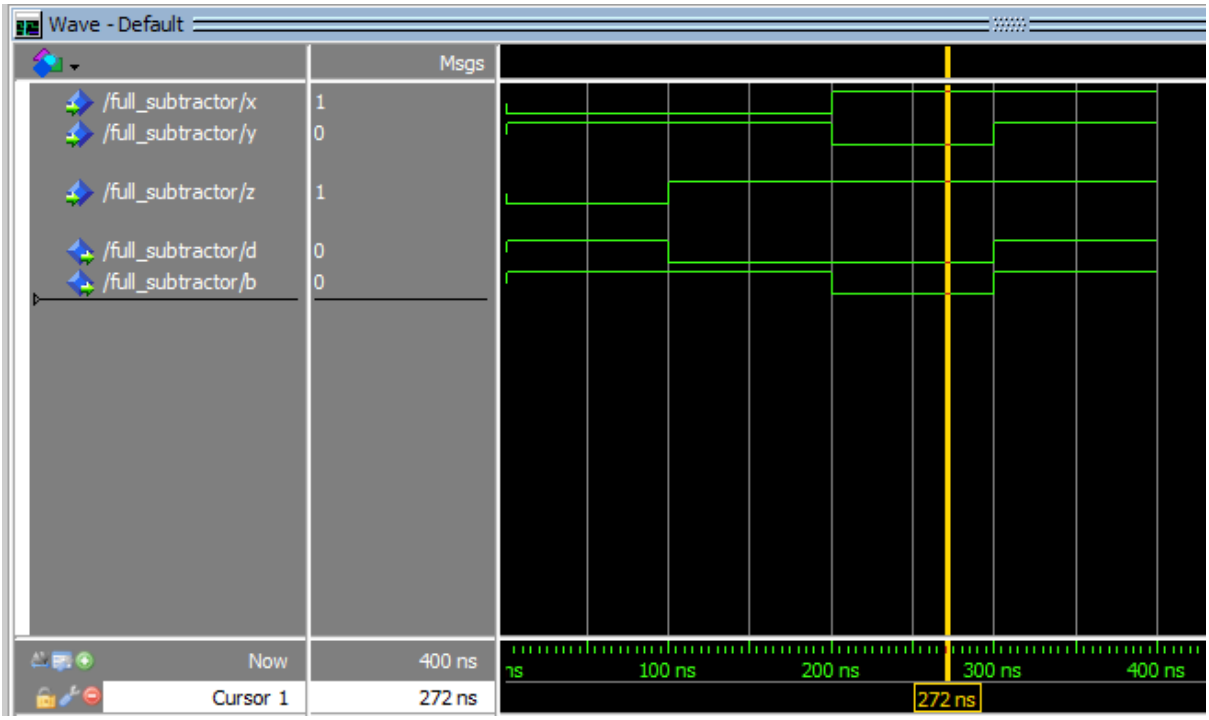
PROGRAM: -

```
library ieee;
use ieee.std_logic_1164.all;
entity full_subtractor is
port(x,y,z: in std_logic;
d,b: out std_logic);
end full_subtractor;
architecture dataflow of full_subtractor is
begin
d<= (x xor y) xor z;
b<= ((not x) and y) or ((not x) and z) or (y and z);
end dataflow;
```

TRUTH TABLE

Inputs			Outputs	
x	Y	z	d	b
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	0	1
1	0	0	1	0
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1

SIMULATION WAVEFORM



OBSERVATION

	Inputs			Outputs	
Timing (ns)	x	Y	Z	d	b
0-100	0	1	0	1	1
100-200	0	1	1	0	1
200-300	1	0	1	0	0
300-400	1	1	1	1	1

CONCLUSION: - Hence the full subtractor is implemented in VHDL and the functionality is verified.

PROGRAM 3(e)

AIM: - To write a program in VHDL for implementing the two bit comparator and to verify the functionality.

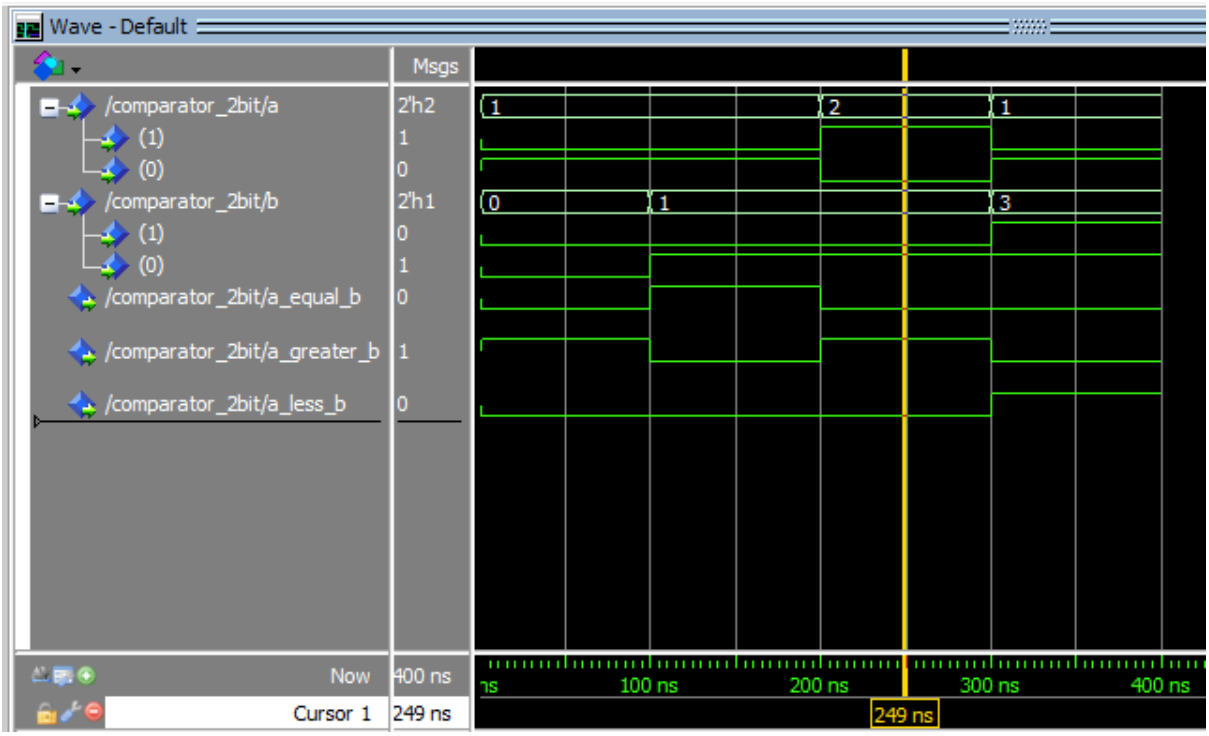
PROGRAM: -

```
library ieee;
use ieee.std_logic_1164.all;
entity comparator_2bit is
port(a:in std_logic_vector(1 downto 0);
b:in std_logic_vector(1 downto 0);
a_equal_b:out std_logic;
a_greater_b:out std_logic;
a_less_b:out std_logic);
end comparator_2bit;
architecture comparator_2bit_arch of comparator_2bit is
begin
a_equal_b<='1' when (a=b) else
'0';
a_greater_b<='1' when (a>b) else
'0';
a_less_b<='1' when (a<b) else
'0';
end comparator_2bit_arch;
```

TRUTH TABLE

Inputs				Outputs		
a(1)	a(0)	b(1)	b(0)	a_greater_b	a_equal_b	a_less_b
0	0	0	0	0	1	0
0	0	0	1	0	0	1
0	0	1	0	0	0	1
0	0	1	1	0	0	1
0	1	0	0	1	0	0
0	1	0	1	0	1	0
0	1	1	0	0	0	1
0	1	1	1	0	0	1
1	0	0	0	1	0	0
1	0	0	1	1	0	0
1	0	1	0	0	1	0
1	0	1	1	0	0	1
1	1	0	0	1	0	0
1	1	0	1	1	0	0
1	1	1	0	1	0	0
1	1	1	1	0	1	0

SIMULATION WAVEFORM



OBSERVATION

	Inputs				Outputs		
Timing(ns)	a(1)	a(0)	b(1)	b(0)	a_greater_b	a_equal_b	a_less_b
0-100	0	1	0	0	1	0	0
100-200	0	1	0	1	0	1	0
200-300	1	0	0	1	1	0	0
300-400	0	1	1	1	0	0	1

CONCLUSION: - Hence the two bit comparator is implemented in VHDL and the functionality is verified.

PROGRAM-3(f)

AIM: -To write a program in VHDL for implementing the full adder using two half adders and to verify the functionality.

PROGRAM: -

COMPONENT OR2: -

```
library ieee;

use ieee.std_logic_1164.all;

entity or2 is

port(a,b:in std_logic;

c:out std_logic);

end or2;

architecture dataflow of or2 is

begin

c<=a or b;

end dataflow;
```

COMPONENT HALF ADDER: -

```
library ieee;

use ieee.std_logic_1164.all;

entity half_adder is

port(a,b: in std_logic;

s,c: out std_logic);

end half_adder;

architecture dataflow of half_adder is

begin

s<= a xor b;

c<= a and b;

end dataflow;
```

TOP MODULE: -

```
library ieee;

use ieee.std_logic_1164.all;

entity fulladder_halfadder is
```

```
port(x,y,z:in std_logic;
s,c:out std_logic);
end fulladder_halfadder;
architecture structural of fulladder_halfadder is
component half_adder
port(a,b:in std_logic;
s,c:out std_logic);
end component;
component or2
port(a,b:in std_logic;
c:out std_logic);
end component;
signal s1,t1,t2:std_logic;
begin
x1:half_adder port map(x,y,s1,t1);
x2:half_adder port map(s1,z,s,t2);
x3:or2 port map(t1,t2,c);
end structural;
```

TRUTH TABLE

Inputs			Outputs	
x	y	z	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

PROGRAM-4(a)

AIM: -To write a program in VHDL for implementing binary to gray code converter and to verify the functionality.

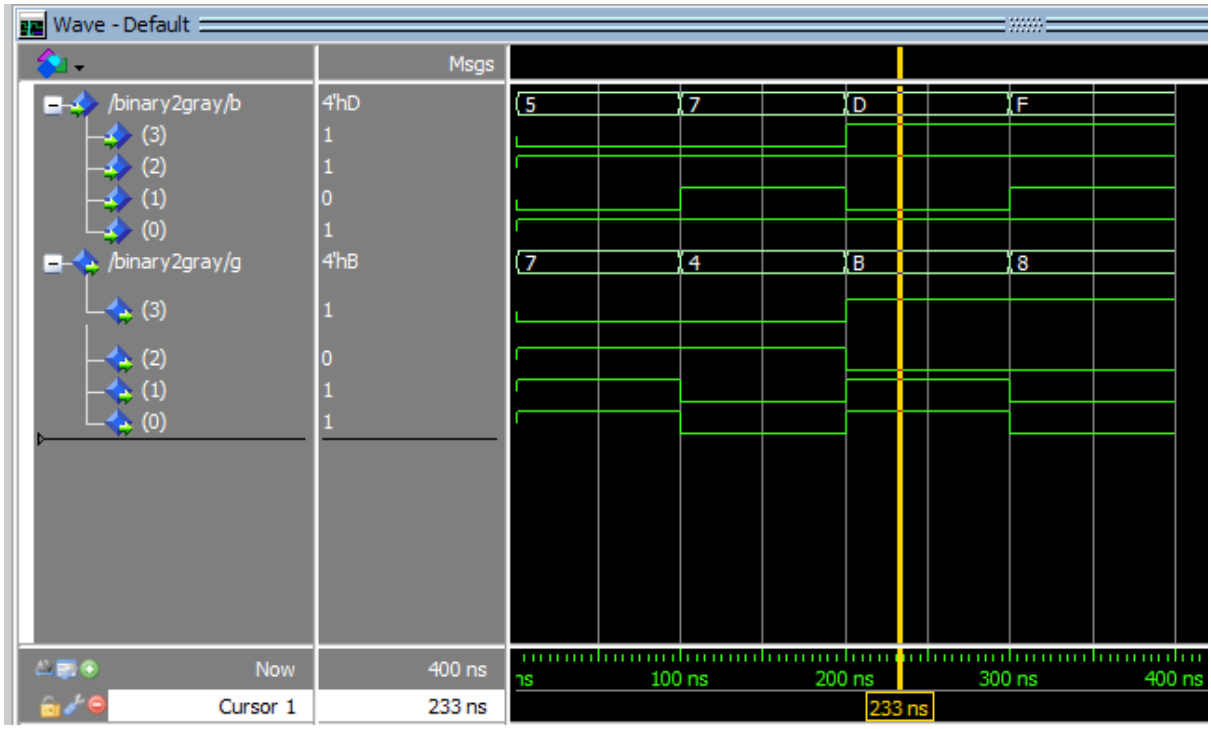
PROGRAM: -

```
library ieee;

use ieee.std_logic_1164.all;

entity binary2gray is

port(b:in std_logic_vector(3 downto 0);

g:out std_logic_vector(3 downto 0));

end binary2gray;

architecture behavioral of binary2gray is

begin

g(3)<=b(3);

g(2)<=b(3) xor b(2);

g(1)<=b(2) xor b(1);

g(0)<=b(1) xor b(0);

end behavioral;
```

TRUTH TABLE

Binary code				Gray code			
b(3)	b(2)	b(1)	b(0)	g(3)	g(2)	g(1)	g(0)
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	1
0	0	1	0	0	0	1	1
0	0	1	1	0	0	1	0
0	1	0	0	0	1	1	0
0	1	0	1	0	1	1	1
0	1	1	0	0	1	0	1
0	1	1	1	0	1	0	0
1	0	0	0	1	1	0	0
1	0	0	1	1	1	0	1
1	0	1	0	1	1	1	1
1	0	1	1	1	1	1	0
1	1	0	0	1	0	1	0
1	1	0	1	1	0	1	1
1	1	1	0	1	0	0	1
1	1	0	1	1	0	0	0

SIMULATION WAVEFORM



OBSERVATION

	Binary code				Gray code			
Timing (ns)	b(3)	b(2)	b(1)	b(0)	g(3)	g(2)	g(1)	g(0)
0-100	0	1	0	1	0	1	1	1
100-200	0	1	1	1	0	1	0	0
200-300	1	1	0	1	1	0	1	1
300-400	1	1	1	1	1	0	0	0

CONCLUSION: - Hence binary to gray code converter is implemented in VHDL and the functionality is verified.

PROGRAM-4(b)

AIM: -To write a program in VHDL for implementing gray to binary code converter and to verify the functionality.

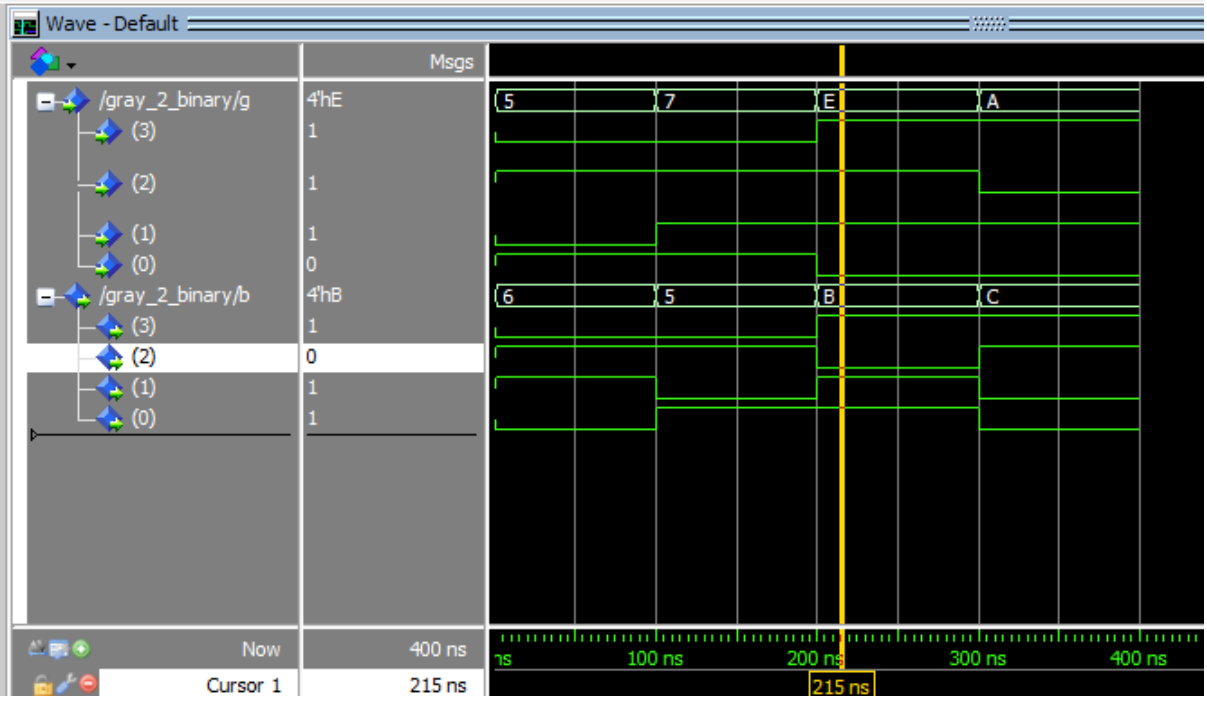
PROGRAM: -

```
LIBRARY ieee;

USE ieee.std_logic_1164.ALL;

entity gray_2_binary is

port(g:in std_logic_vector(3 downto 0);

b:out std_logic_vector(3 downto 0));

end gray_2_binary;

architecture gate_level of gray_2_binary  is

begin

b(3)<=g(3);

b(2)<=g(3) xor g(2);

b(1)<=g(3) xor g(2) xor g(1);

b(0)<=g(3) xor g(2) xor g(1) xor g(0);

end gate_level;
```

TRUTH TABLE

Gray code				Binary code			
g(3)	g(2)	g(1)	g(0)	b(3)	b(2)	b(1)	b(0)
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	1
0	0	1	0	0	0	1	0
0	0	1	1	0	0	1	1
0	1	0	0	0	1	0	0
0	1	0	1	0	1	0	1
0	1	1	0	0	1	1	0
0	1	1	1	0	1	1	1
1	0	0	0	1	0	0	0
1	0	0	1	1	0	0	1
1	0	1	0	1	0	1	0
1	0	1	1	1	0	1	1
1	1	0	0	1	1	0	0
1	1	0	1	1	1	0	1
1	1	1	0	1	1	1	0
1	1	0	1	1	1	1	1

SIMULATION WAVEFORM



OBSERVATION

	Gray code				Binary code			
Timing (ns)	g(3)	g(2)	g(1)	g(0)	b(3)	b(2)	b(1)	b(0)
0-100	0	1	0	1	0	1	1	0
100-200	0	1	1	1	0	1	0	1
200-300	1	1	1	0	1	0	1	1
300-400	1	0	1	0	1	1	0	0

CONCLUSION: - Hence gray to binary code converter is implemented in VHDL and the functionality is verified.

PROGRAM-4(c)

AIM: -To write a program in VHDL for implementing BCD to seven segment display and to verify the functionality.

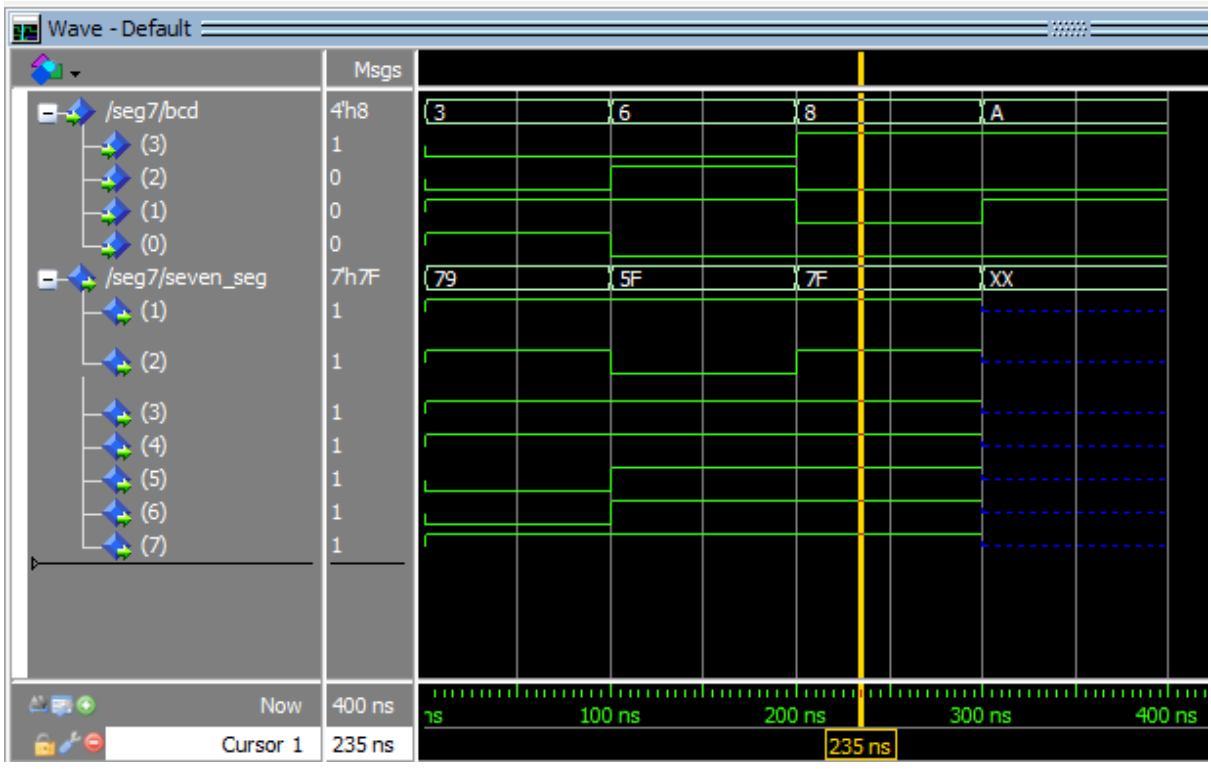
PROGRAM: -

```
library ieee;
use ieee.std_logic_1164.all;
use ieee.std_logic_arith.all;
use ieee.std_logic_unsigned.all;
entity seg7 is
port(bcd:in std_logic_vector(3 downto 0);
seven_seg:out std_logic_vector(1 to 7));
end seg7;
architecture arch_segment7 of seg7 is
begin
process(bcd)
begin
case bcd is
when "0000"=> seven_seg <="1111110";
when "0001"=> seven_seg <="0110000";
when "0010"=> seven_seg <="1101101";
when "0011"=> seven_seg <="1111001";
when "0100"=> seven_seg <="0110011";
when "0101"=> seven_seg <="1011011";
when "0110"=> seven_seg <="1011111";
when "0111"=> seven_seg <="1110000";
when "1000"=> seven_seg <="1111111";
when "1001"=> seven_seg <="1110011";
when others => seven_seg <="-------";
end case;
end process;
end arch_segment7;
```

TRUTH TABLE

Input lines				Output lines							Display
A	B	C	D	A	b	c	d	e	f	g	Pattern
0	0	0	0	1	1	1	1	1	1	0	0
0	0	0	1	0	1	1	0	0	0	0	1
0	0	1	0	1	1	0	1	1	0	1	2
0	0	1	1	1	1	1	1	0	0	1	3
0	1	0	0	0	1	1	0	0	1	1	4
0	1	0	1	1	0	1	1	0	1	1	5
0	1	1	0	1	0	1	1	1	1	1	6
0	1	1	1	1	1	1	0	0	0	0	7
1	0	0	0	1	1	1	1	1	1	1	8
1	0	0	1	1	1	1	0	0	1	1	9

SIMULATION WAVEFORM



OBSERVATION

	Input lines				Output lines						
Timing (ns)	**A**	**B**	**C**	**D**	**a**	**b**	**c**	**d**	**e**	**f**	**g**
0-100	0	0	1	1	1	1	1	1	0	0	1
100-200	0	1	1	0	1	0	1	1	1	1	1
200-300	1	0	0	0	1	1	1	1	1	1	1
300-400	1	0	1	0	-	-	-	-	-	-	-

CONCLUSION: - Hence BCD to seven segment display is implemented in VHDL and the functionality is verified.

PROGRAM-5(a)

AIM: -To write a code in VHDL for implementing the 4x1 multiplexer and to observe the waveforms.

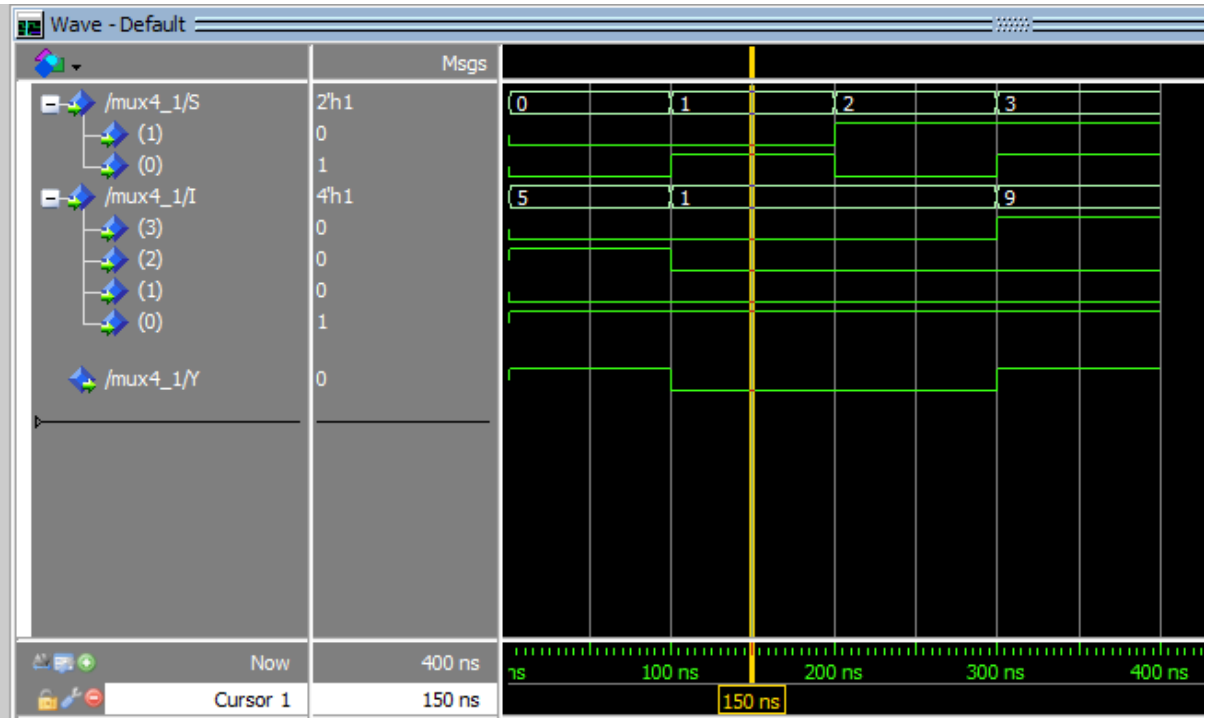
PROGRAM: -

```
library ieee;

use ieee.std_logic_1164.all;

entity mux4_1 is

port(S:in std_logic_vector(1 downto 0);

I:in std_logic_vector(3 downto 0);

Y:out std_logic);

end mux4_1;

architecture arch_mux of mux4_1 is

begin

process(S,I)

begin

if(S<="00") then Y<=I(0);

elsif (S<="01") then Y<=I(1);

elsif (S<="10") then Y<=I(2);

else Y<=I(3);

end if;

end process;

end arch_mux;
```

TRUTH TABLE

Selection line		**Output**
S1	S0	Y
0	0	I(0)
0	1	I(1)
1	0	I(2)
1	1	I(3)

SIMULATION WAVEFORM



OBSERVATION

Selection line		**Input line**				**Output**
S(1)	S(0)	I(3)	I(2)	I(1)	I(0)	Y
0	0	0	1	0	1	1
0	1	0	0	0	1	0
1	0	0	0	0	1	0
1	1	1	0	0	1	1

CONCLUSION: - Hence the 4x1 multiplexer is implemented in VHDL and the functionality is verified.

PROGRAM-5(b)

AIM: -To write a code in VHDL for implementing the 1x4 demultiplexer and to observe the waveforms.

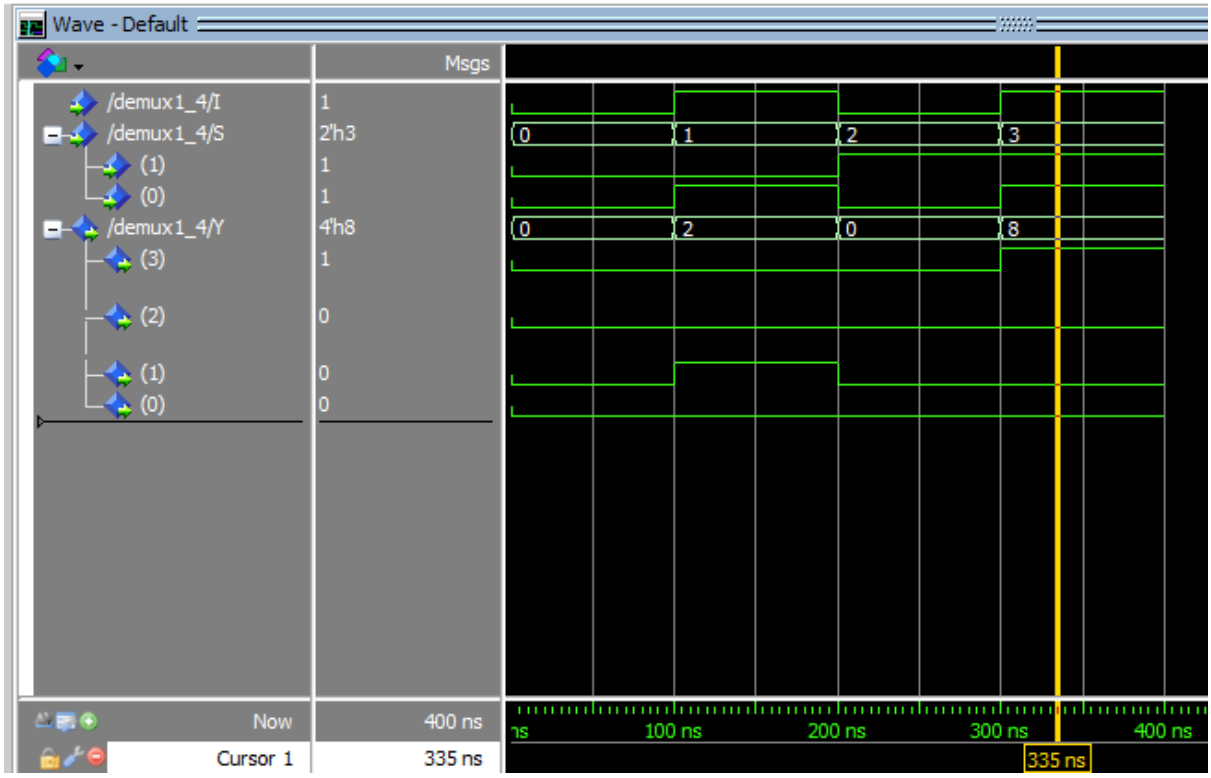
PROGRAM: -

```
library ieee;
use ieee.std_logic_1164.all;
use ieee.std_logic_arith.all;
use ieee.std_logic_unsigned.all;
entity demux1_4 is
port(I:in std_logic;
S:in std_logic_vector(1 downto 0);
Y:out std_logic_vector(3 downto 0));
end demux1_4;
architecture arch_demux of demux1_4 is
begin
process(I,S)
begin
if(S<="00")then
Y(3)<='0';
Y(2)<='0';
Y(1)<='0';
Y(0)<=I;
elsif(S<="01")then
Y(3)<='0';
Y(2)<='0';
Y(1)<=I;
Y(0)<='0';
elsif(S<="10")then
Y(3)<='0';
Y(2)<=I;
Y(1)<='0';
Y(0)<='0';
else
Y(3)<=I;
Y(2)<='0';
Y(1)<='0';
Y(0)<='0';
end if;
end process;
end arch_demux;
```

TRUTH TABLE

Input	Selection line		Output			
I	S1	S0	Y(3)	Y(2)	Y(1)	Y(0)
I	0	0	0	0	0	I
I	0	1	0	0	I	0
I	1	0	0	I	0	0
I	1	1	I	0	0	0

SIMULATION WAVEFORM



CONCLUSION: - Hence the 1x4 demultiplexer is implemented in VHDL and the functionality is verified.

PROGRAM-6(a)

AIM: -To write a code in VHDL for implementing the 2 to 4 decoder and to observe the waveforms.

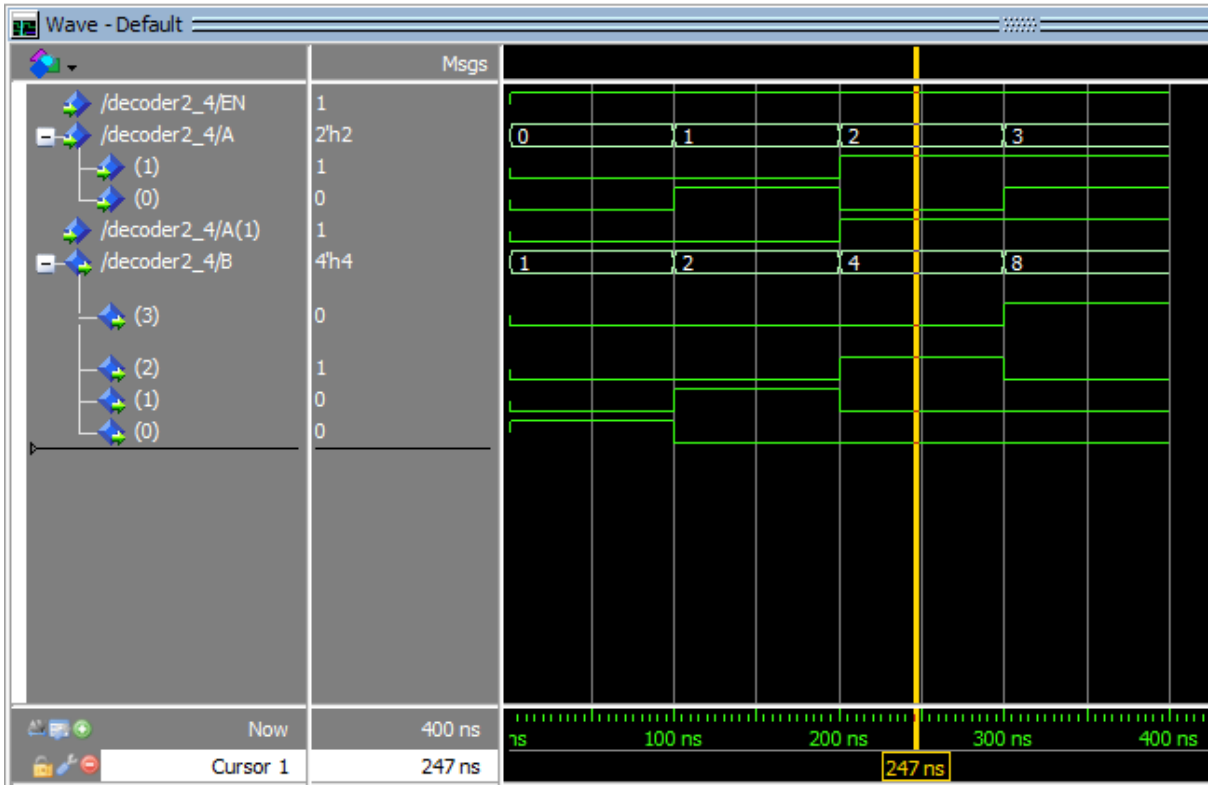
PROGRAM: -

```
library ieee;
use ieee.std_logic_1164.all;
use ieee.std_logic_arith.all;
use ieee.std_logic_unsigned.all;
entity decoder2_4 is
port(EN:in std_logic;
A:in std_logic_vector(1 downto 0);
B:out std_logic_vector(3 downto 0));
end decoder2_4;
architecture arch_decoder of decoder2_4 is
begin
process(EN,A)
begin
if(EN='1') then
case A is
when"00"=>B<="0001";
when"01"=>B<="0010";
when"10"=>B<="0100";
when"11"=>B<="1000";
when others=>B<="0000";
end case;
else
B<="1111";
end if;
end process;
end arch_decoder;
```

TRUTH TABLE

Inputs			Outputs			
EN	A1	A0	B(3)	B(2)	B(1)	B(0)
0	X	X	1	1	1	1
1	0	0	0	0	0	1
1	0	1	0	0	1	0
1	1	0	0	1	0	0
1	1	1	1	0	0	0

SIMULATION WAVEFORM



OBSERVATION

	Inputs			Outputs			
Timing (ns)	EN	A1	A0	B(3)	B(2)	B(1)	B(0)
0-100	1	0	0	0	0	0	1
100-200	1	0	1	0	0	1	0
200-300	1	1	0	0	1	0	0
300-400	1	1	1	1	0	0	0

CONCLUSION: - Hence the 2 to 4 decoder is implemented in VHDL and the functionality is verified.

PROGRAM-6(b)

AIM: -To write a code in VHDL for implementing the 4 to 2 encoder and to observe the waveforms.

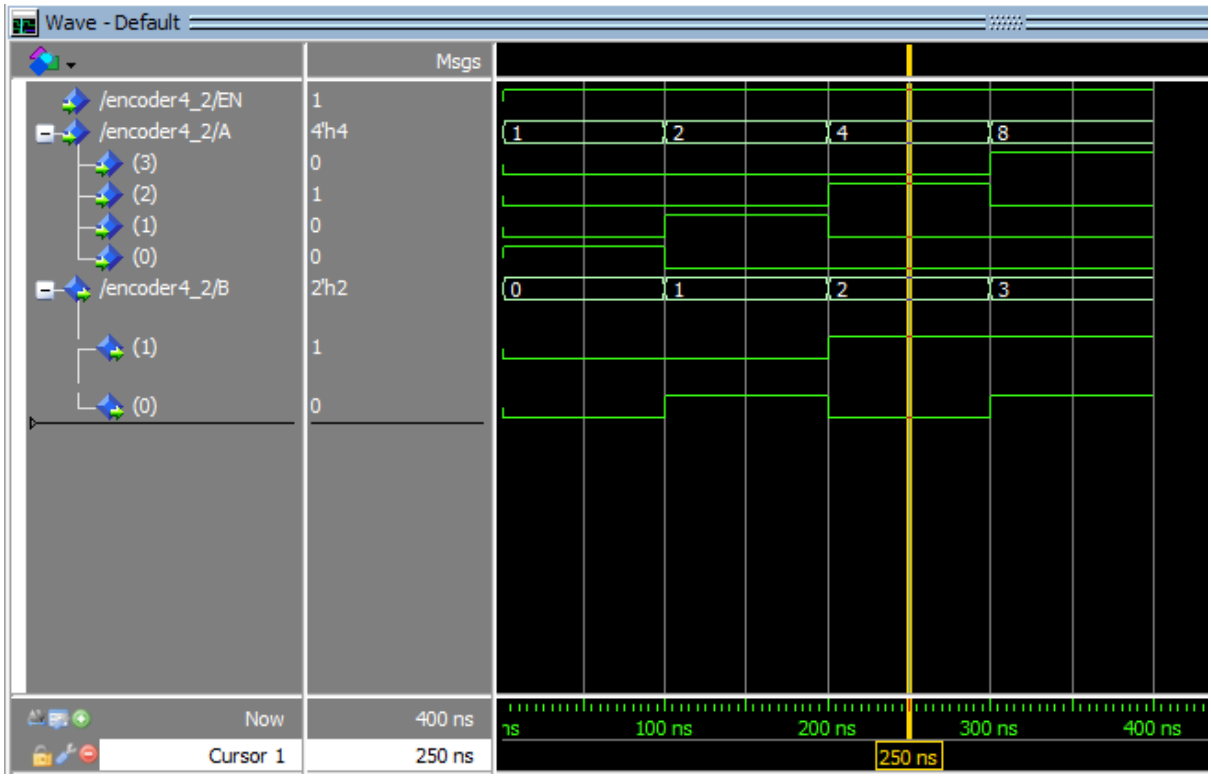
PROGRAM: -

```
library ieee;
use ieee.std_logic_1164.all;
use ieee.std_logic_arith.all;
use ieee.std_logic_unsigned.all;
entity encoder4_2 is
port(EN:in std_logic;
A:in std_logic_vector(3 downto 0);
B:out std_logic_vector(1 downto 0));
end encoder4_2;
architecture arch_encoder of encoder4_2 is
begin
process(EN,A)
begin
if(EN='1') then
case A is
when"0001"=>B<="00";
when"0010"=>B<="01";
when"0100"=>B<="10";
when"1000"=>B<="11";
when others=>B<="00";
end case;
else
B<="11";
end if;
end process;
end arch_encoder;
```

TRUTH TABLE

Inputs					**Outputs**	
EN	A(3)	A(2)	A(1)	A(0)	B(1)	B(0)
0	X	X	X	X	1	1
1	0	0	0	1	0	0
1	0	0	1	0	0	1
1	0	1	0	0	1	0
1	1	0	0	0	1	1

SIMULATION WAVEFORM



OBSERVATION

	Inputs					**Outputs**	
Timing (ns)	EN	A(3)	A(2)	A(1)	A(0)	B(1)	B(0)
0-100	1	0	0	0	1	0	0
100-200	1	0	0	1	0	0	1
200-300	1	0	1	0	0	1	0
300-400	1	1	0	0	0	1	1

CONCLUSION: - Hence the 4 to 2 encoder is implemented in VHDL and the functionality is verified.

PROGRAM-7(a)

AIM: - To write a code in VHDL for implementing the D flip-flop and to verify the functionality.

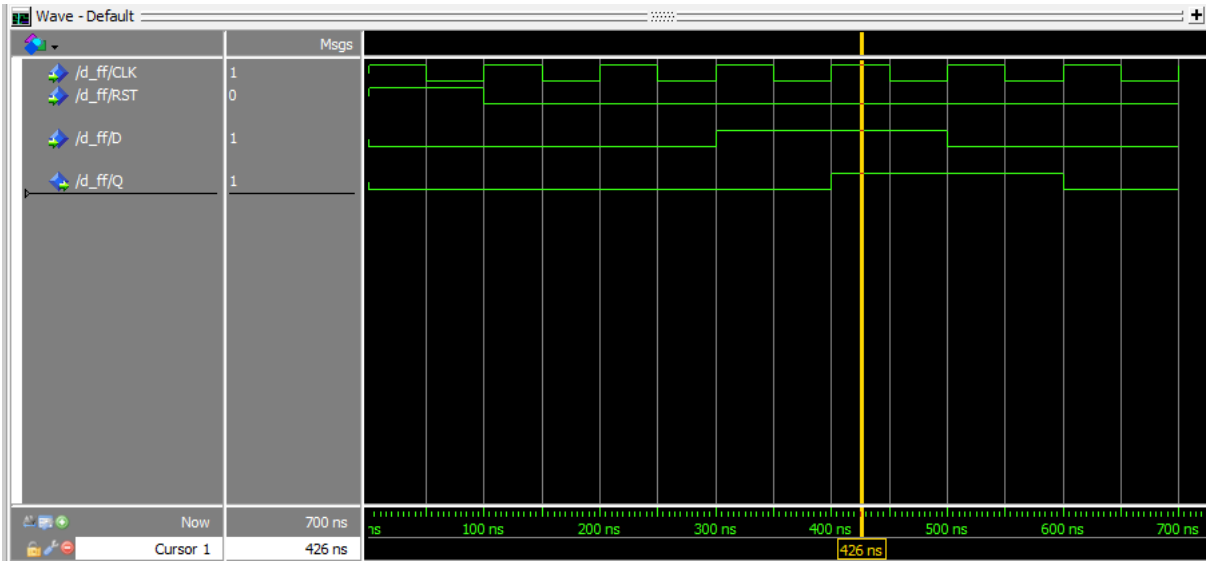
PROGRAM: -

```
library ieee;
use ieee.std_logic_1164.all;
use ieee.std_logic_arith.all;
use ieee.std_logic_unsigned.all;
entity D_FF is
port(D,CLK,RST:in std_logic;
Q:out std_logic);
end D_FF;
architecture arch_D_FF of D_FF is
begin
process(D,CLK,RST)
begin
if(RST='1')then
Q<='0';
elsif(rising_edge(CLK)) then
Q<=D;
end if;
end process;
end arch_D_FF;
```

TRUTH TABLE

Present State (Q_n)	**Input D**	**Next State** (Q_{n+1})
0	0	0
0	1	1
1	0	0
1	1	0

SIMULATION WAVEFORM



OBSERVATION

Timing(ns)	Present State (Q_n)	Timing(ns)	Input D	Timing(ns)	Next State (Q_{n+1})
100-200	0	100-200	0	200-300	0
200-300	0	200-300	0	300-400	0
300-400	0	300-400	1	400-500	1
400-500	1	400-500	1	500-600	1
500-600	1	500-600	0	600-700	0

CONCLUSION: - Hence the D flip-flop is implemented in VHDL and the functionality is verified.

PROGRAM-7(b)

AIM: -To write a code in VHDL for implementing the SR flip-flop and to verify the functionality.

PROGRAM: -

```
library ieee;
use ieee.std_logic_1164.all;
use ieee.std_logic_arith.all;
use ieee.std_logic_unsigned.all;
entity SRFF is
port(s:in std_logic;
r:in std_logic;
clk:in std_logic;
rst:in std_logic;
q:out std_logic;
qb:out std_logic);
end SRFF;
architecture arch_SRFF of SRFF is
begin
process(s,r,clk,rst)
begin
if(rst='1')then
q<='0';
qb<='0';
elsif(clk='1' and clk'event) then
if(s/=r) then
q<=s;
qb<=r;
elsif(s='1' and r='1') then
q<='Z';
qb<='Z';
end if;
```

end if;

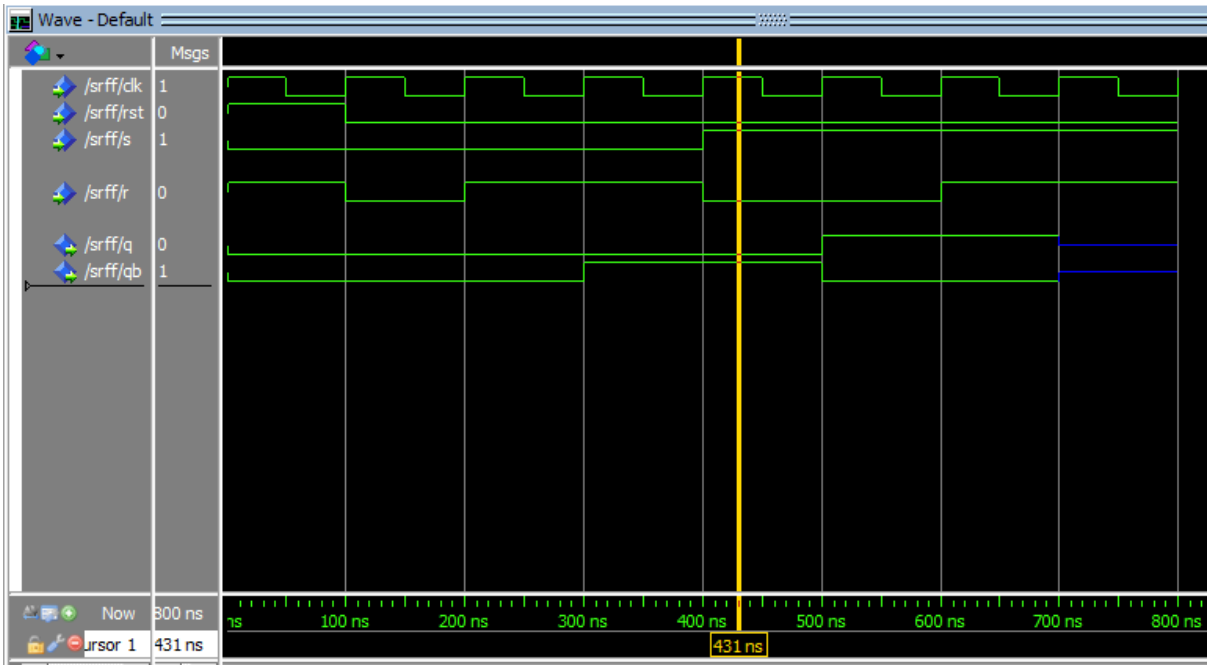
end process;

end arch_SRFF;

TRUTH TABLE

Present State (Q_n)	Input		Next State (Q_{n+1})
	S	R	
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	X
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	X

SIMULATION WAVEFORM



.

OBSERVATION

Timing(ns)	Present State (Q_n)	Timing(ns)	Input		Timing(ns)	Next State (Q_{n+1})
			S	R		
100-200	0	100-200	0	0	200-300	0
200-300	0	200-300	0	1	300-400	0
300-400	0	300-400	0	1	400-500	0
400-500	0	400-500	1	0	500-600	1
500-600	1	500-600	1	0	600-700	1
600-700	1	600-700	1	1	700-800	Z

CONCLUSION: - Hence the SR flip-flop is implemented in VHDL and the functionality is verified.

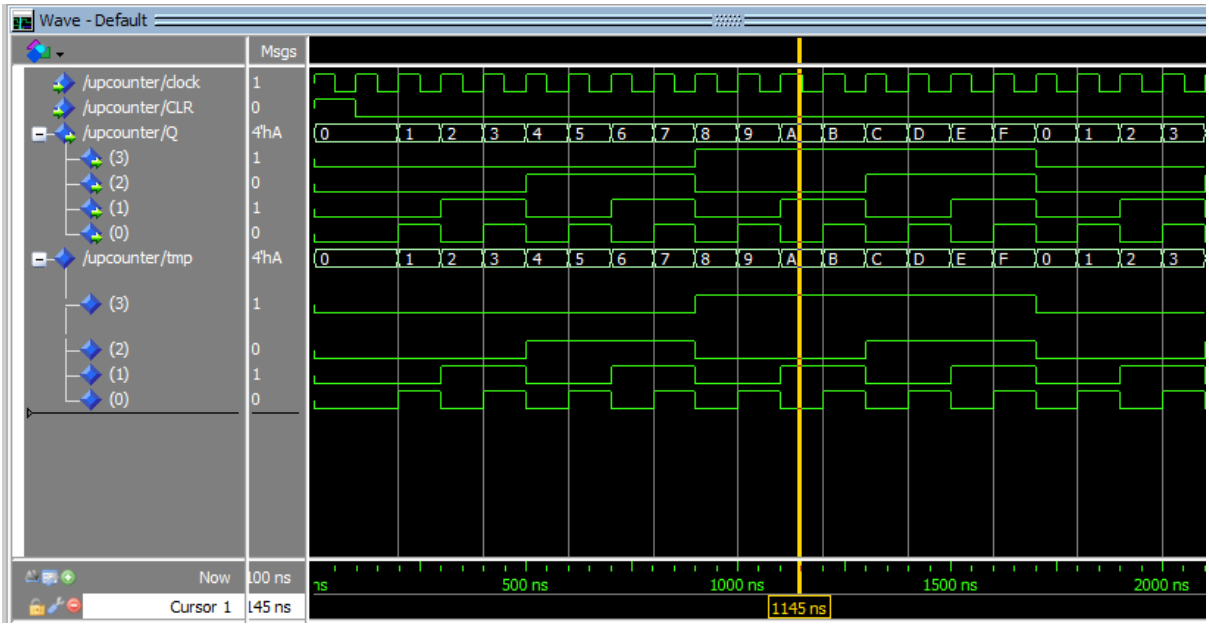
PROGRAM-8

AIM: -To write a code in VHDL for implementing the 4 bit up counter and to observe the waveforms.

PROGRAM: -

```
library ieee;
use ieee.std_logic_1164.all;
use ieee.std_logic_arith.all;
use ieee.std_logic_unsigned.all;
entity upcounter is
port(clock,CLR:in std_logic;
Q:out std_logic_vector(3 downto 0));
end upcounter;
architecture behavioral of upcounter is
signal tmp:std_logic_vector(3 downto 0);
begin
process(clock,CLR)
begin
if(CLR='1')then
tmp<="0000";
elsif(clock='1' and clock'event)then
tmp<=tmp+'1';
end if;
end process;
Q<=tmp;
end behavioral;
```

SIMULATION WAVEFORM



CONCLUSION: - Hence the 4-bit up counter is implemented in VHDL and the functionality is verified.

VIVA QUESTIONS

1. What do you mean by Logic Gates?
2. What are the applications of Logic Gates?
3. What is Truth Table?
4. Why we use basic logic gates?
5. Write down the truth table of all logic gates?
6. What do you mean by universal gate?
7. Write truth table for 2 I/P OR, NOR, AND and NAND gate?
8. Implement all logic gate by using Universal gate?
9. Why is they called Universal Gates?
10. Give the name of universal gate?
11. Draw circuit diagram of Half Adder circuit?
12. Draw circuit diagram of Full Adder circuit?
13. Draw Full Adder circuit by using Half Adder circuit and minimum no. of logic gate?
14. Write Boolean function for half adder? Write Boolean function for Full adder?
15. Design the half Adder & Full Adder using NAND-NAND Logic.
16. Draw circuit diagram of Half Subtractor circuit?
17. Draw circuit diagram of Full Subtractor circuit?
18. Draw Full Subtractor circuit by using Half Subtractor circuit and minimum no. of logic gate?
19. Write Boolean function for half Subtractor?
20. Write Boolean function for Full Subtractor?
21. What is Excess-3 code? Why it is called Excess-3 code?
22. What is the application of Excess-3 Code?
23. What is ASCII code?
24. Excess-3 code is Weighted or Unweighted?
25. Out of the possible 16 code combination? How many numbers used in Excess-3 code?
26. What is Demorgan's Law?
27. Show the truth table for Demorgan's Theorem?
28. What is Minterm & Maxterm?

29. How Minterm can be converted in Max term?
30. What is Hybrid function?
31. What is Flip-Flop?
32. What is Latch circuit?
33. Draw a truth –tables of S-R, J-K, D and T?
34. What is the disadvantages of S-R Flip-Flop?
35. How can you remove the problem of S-R Flip –Flop?
36. Make circuit diagram of S-R, J-K, D and T Flip-Flop?
37. What do you understand by Race Aground condition? How it is over come in J-K Flip Flop?
38. Explain the principle of Multiplexer?
39. Draw a circuit diagram of 4: 1 Multiplexer?
40. What are the advantages of Multiplexer?
41. What are the disadvantages of Multiplexer?
42. Make the Truth-table of Multiplexer?
43. Explain about Demultiplexer?
44. Draw a circuit diagram of 1: 4 Demultiplexer?
45. Make a logic diagram of 1: 4 Demultiplexer?
46. What is the application of Demultiplexer?
47. What is the difference between Multiplexer and Demultiplexer?

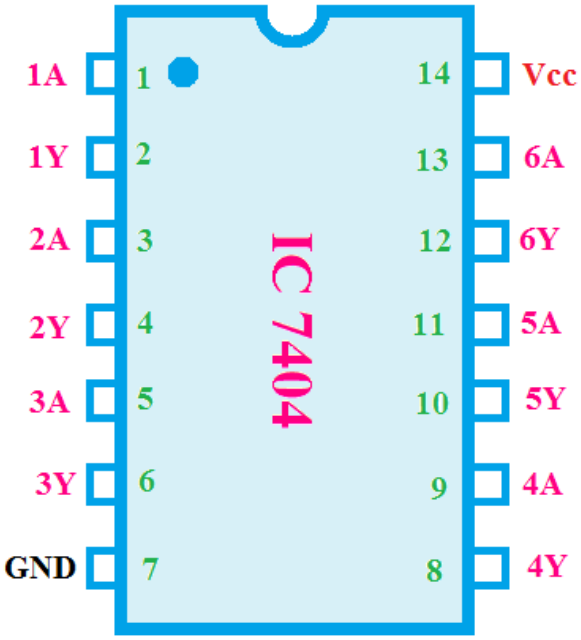
DATA SHEETS OF ICs

IC 7404 Pin Diagram, Circuit Design, Data sheet, application

IC 7404 or IC 74LS04 is a logic gate IC. It consists of six NOT Gates. We know that the NOT gate also called inverters because of it does the complements of the input. When we apply 0 or low signal to the input it gives 1 or high signal in output.

The Pin Diagram of IC 7404:

The IC 7404 consists of fourteen pins each pin are shown below.



Operating Condition of IC 74LS04:

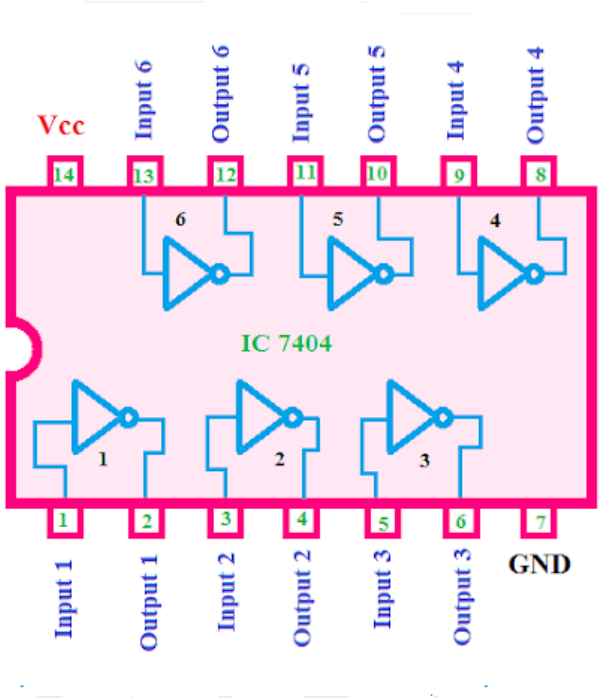
1. The power supply should be given to the IC from 4.5V DC to 5.25V DC.
2. The IC will consider a signal as high when the voltage of the signal is above 2V.
3. The IC will consider a signal as low when the voltage of the signal is below 0.8V.
4. The operating temperature of the IC should be below the 70-degree centigrade.

Characteristics:

1. IC 74LS04 can deliver -0.4 mA current when the output is high.
2. It can deliver 16 mA current when the output is low.
3. When the V_{CC} is 5V and the input signal is 5V then the IC draws 1 mA current.
4. When the V_{CC} is 5V and the input signal is 2.7V then the IC draws 20 to 40 micro-ampere currents.
5. When the V_{CC} is 5V and the input signal is 0.4V then the IC draws -1.6 mA current.

Internal Structure of IC 7404:

IC 7404 consists six NOT Gates as shown in the below figure.



Application of IC 7404:

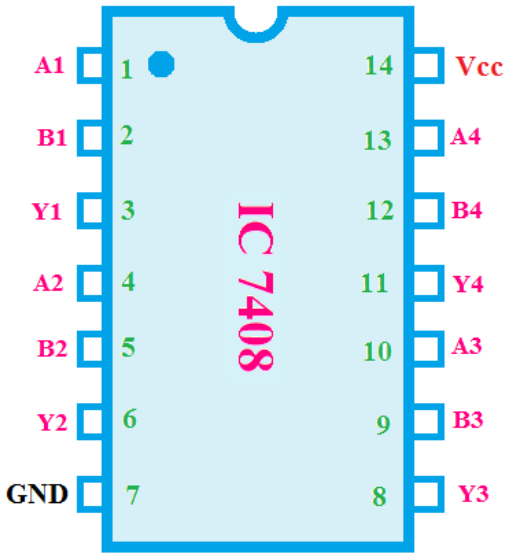
1. IC 7404 is mostly used for digital electronics projects.
2. They are also used in some electronic devices.
3. They are also used in Space instruments.

IC 7408 Pin Diagram, Circuit Design, Data Sheet, Application

IC 7408 is a logic gate IC. It consists of four two-input AND Gates. The AND gate perform logical AND operation. Logic gates come in form of ICs. The all four AND gates are independent. Each gate has three pins two inputs and one output. IC 74HC08, IC DM7408 are AND gate ICs.

Pin diagram of IC 7408:

The IC 7408 has total fourteen pins including ground and V_{CC}. The simple pin diagram is shown below.



Operating Condition of IC DM7408:

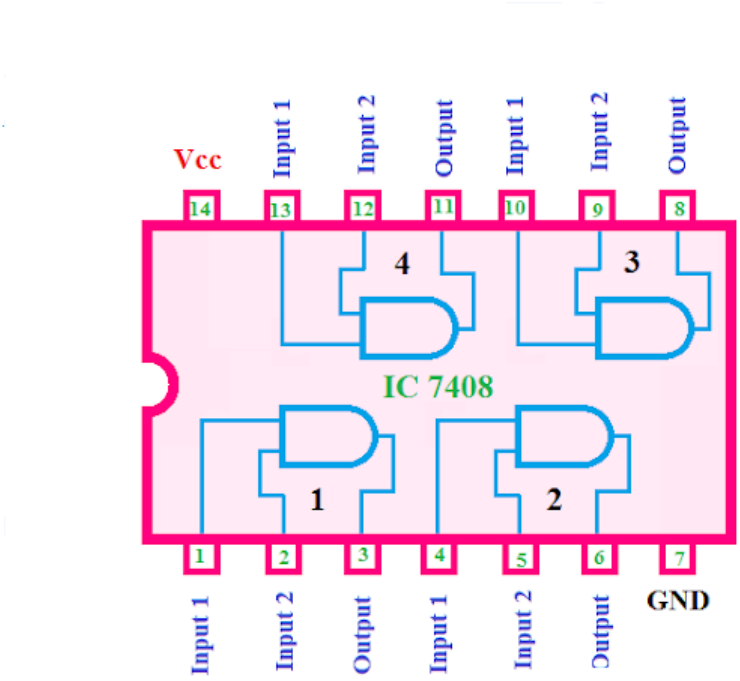
1. The power supply should be given to the IC from 4.5V DC to 5.25V DC.
2. The IC can identify a signal as a high-level signal if the voltage of the signal is above 2V.
3. The IC can identify a signal as a low-level signal if the voltage of the signal is below 0.8V.
4. The IC should be operated below the 70-degree centigrade.

Characteristics:

1. The IC DM7408 can deliver 21 mA current when the output signal is high at maximum V_{CC} Voltage.
2. The IC can deliver 33 mA current when the output signal is low at maximum V_{CC} voltage.
3. When the V_{CC} is maximum and the input signal is 5.5V then the IC draws 1 mA current.
4. When the V_{CC} is maximum and the input signal is 2.7V then the IC draws 20 to 40 micro-ampere currents.
5. When the V_{CC} is maximum and the input signal is 0.4V then the IC draws -1.6 mA current.

The internal structure of IC 7408:

The internal structure of the IC 7408 is shown it consists of four AND Gates.



Application of IC 7408:

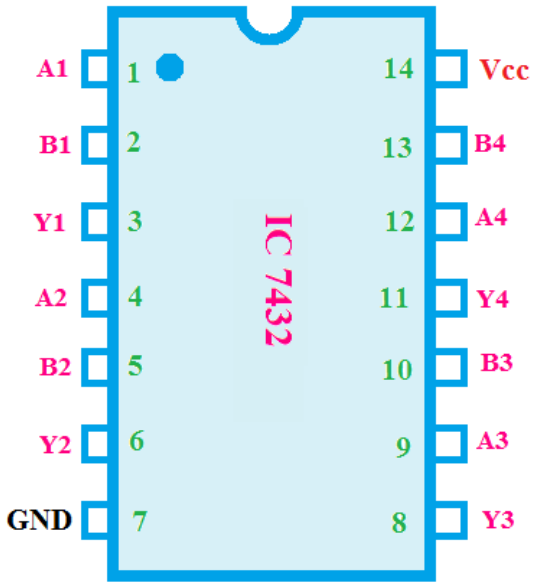
1. IC 7408 is used for digital electronics projects.
2. Used in some electronics devices.

IC 7432 Pin diagram, circuit design, Datasheet, Application

IC 7432 is a logic gate IC which consist of four OR Gates. The OR gate performs logical OR operation. The OR gates come in form of DIP package ICs. Each gate has three terminal two inputs and one output. The ICs are made by CMOS, TTL technology.

Pin Diagram of IC 7432:

The IC 7432 has fourteen pins like other logic gates ICs. The pin diagram is shown below.



Operating Condition of IC 7432:

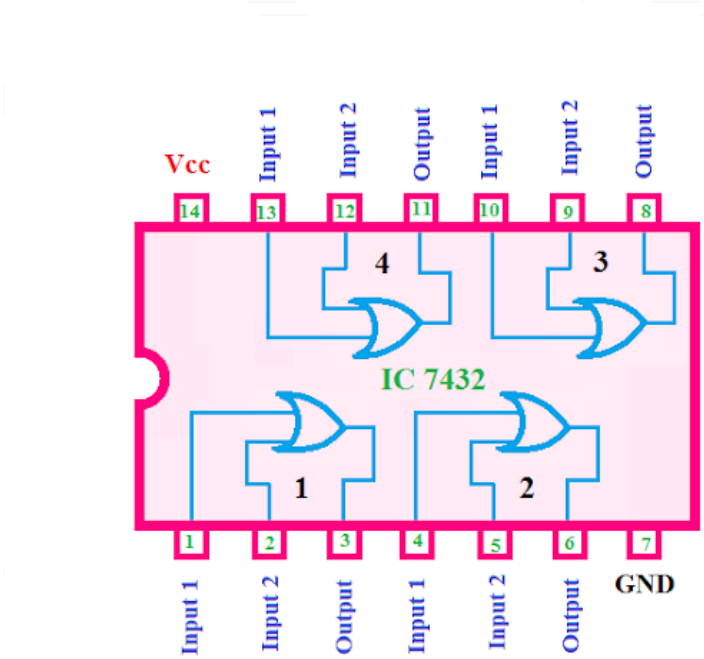
1. The power supply should be given to the IC from 4.5V DC to 5.25V DC
2. The IC will consider a signal as high if the voltage of the signal is above 2V.
3. The IC will consider a signal as low if the voltage of the signal is below 0.8V.
4. The operating temperature of the IC should be below the 70-degree centigrade

Characteristics:

1. IC 74LS04 can deliver -0.4 mA current when the output is high.
2. It can deliver 16 mA current when the output is low.
3. When the V_{CC} is maximum and the input signal is 5V then the IC draws 1 mA current.
4. When the V_{CC} is maximum and the input signal is 2.7V then the IC draws 20 to 40 micro-ampere currents.
5. When the V_{CC} is maximum and the input signal is 0.4V then the IC draws -1.6 mA current.

The internal structure of IC 7432:

The IC 7432 consist of four OR Gates as shown in the below figure. The all OR Gates are independent.



Application of IC 7432:

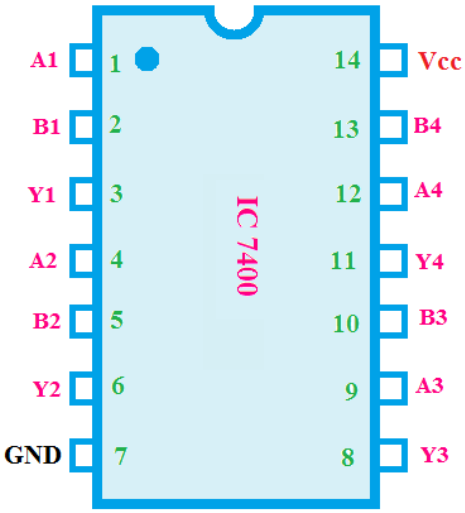
1. IC 7432 are used in digital electronics projects.

IC 7400 Pin Diagram, Circuit design, Datasheet, Application

IC 7400 is fourteen pin Logic Gate IC. The IC 7400 consist of four NAND Gates. The NAND Gate is also called Universal Gate. The NAND gate has a total of three terminals, two inputs terminals, and one output terminal. All NAND Gates are independent. IC 7400 is also called Quad 2-input NAND Gate IC.

Pin diagram of IC 7400:

The IC 7400 has fourteen pins including V_{CC} and ground pins. The simple pin diagram is shown below,



Operating Condition of IC 7400:

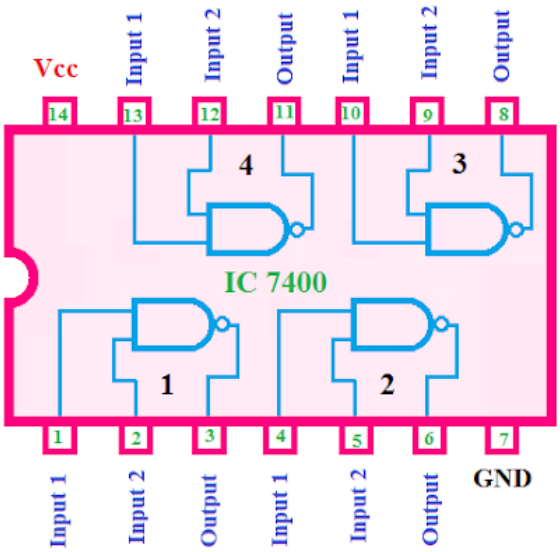
1. The IC 7400 can operate from 4.5V Dc to 5.25V DC voltage. So power supply to the IC should be given in that range.
2. It can identify a signal as a high signal if the signal has the voltage above 2V.
3. It can identify a signal as a low signal if the signal has the voltage below 0.8V.
4. The IC 7400 can withstand up to 70-degree centigrade temperature so the operating temperature should be below that temperature.

Electrical Characteristics:

1. The high-level input current is 20 micro-ampere at maximum V_{CC} and input voltage 2.7V.
2. The low-level input current is -0.36 mA at maximum V_{CC} and input voltage 0.4V
3. Supply current with output HIGH is 1.6 mA at Maximum $V_{CC.}$
4. Supply current with output LOW is 4.4 mA at Maximum $V_{CC.}$
5. The short circuit output current is -20 to -100 mA at maximum $V_{CC.}$

The Internal Structure of IC 7400

The IC 7400 has four independent NAND gates which are shown in the below figure,



Application of IC 7400

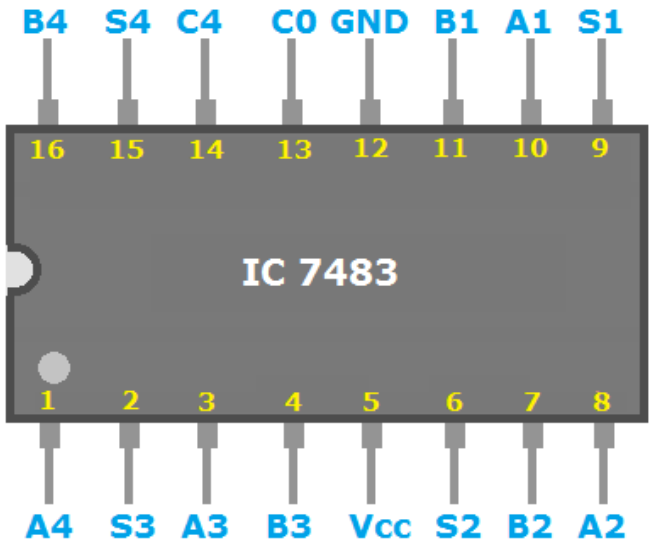
1. NAND gate is a universal gate so it can be used to make others gates like NOT, AND gates etc.
2. They are used in digital electronics projects.

IC 7483 Pin Diagram, Truth Table, Applications

IC 7483 is a digital adder IC that can add two 4-bit numbers. IC 7483 consists of four individual full adder circuits which are internally connected. It has also input and output carry circuit. IC 7483 can be connected directly to the CMOS, NMOS, and TTL circuits.

IC 7483 Pin Diagram

The pin diagram of IC 7483. It has a total of 16 pins.



IC 7483 Pin Diagram

Pin no. **1**(A4), **3**(A3), **8**(A2), **10**(A1) are subjected to give first 4-bit numbers as input 1.

Pin no. **16**(B4), **4**(B3), **7**(B2), **11**(B1) are subjected to give the second 4-bit number as input 2.

Pin no. **15**(S4), **2**(S3), **6**(S2), **9**(S1) are the output pins to collect the data after addition.

Pin no. **14**(C4) is for input carry.

Pin no. **13**(C0) is for output carry.

Pin no. **5** is V_{CC} for Positive power supply.

Pin no. **12** is GND for negative power supply.

IC 7483 Truth Table

This truth table shows the two 4-bit numbers as input and 4-bit output numbers with output carry.

Input Data A				Input Data B				Addition				
A4	A3	A2	A1	B4	B3	B2	B1	C_{out}	S4	S3	S2	S1
1	0	0	0	0	0	1	0	0	1	0	1	0
1	0	0	0	1	0	0	0	1	0	0	0	0
0	0	1	0	1	0	0	0	0	1	0	1	0
0	0	0	1	0	1	1	1	0	1	0	0	0
1	0	1	0	1	0	1	1	1	0	1	0	1
0	1	1	0	0	0	1	1	0	1	0	0	1
1	1	1	0	1	1	1	1	1	1	1	0	1
1	0	1	0	1	1	0	1	1	0	1	1	1

Operating Condition of IC 7483

IC 7483 can operate correctly with 4.75 to 5V DC. It can operate properly at 0 to 70 degrees celcius temperature.

IC 7483 Applications

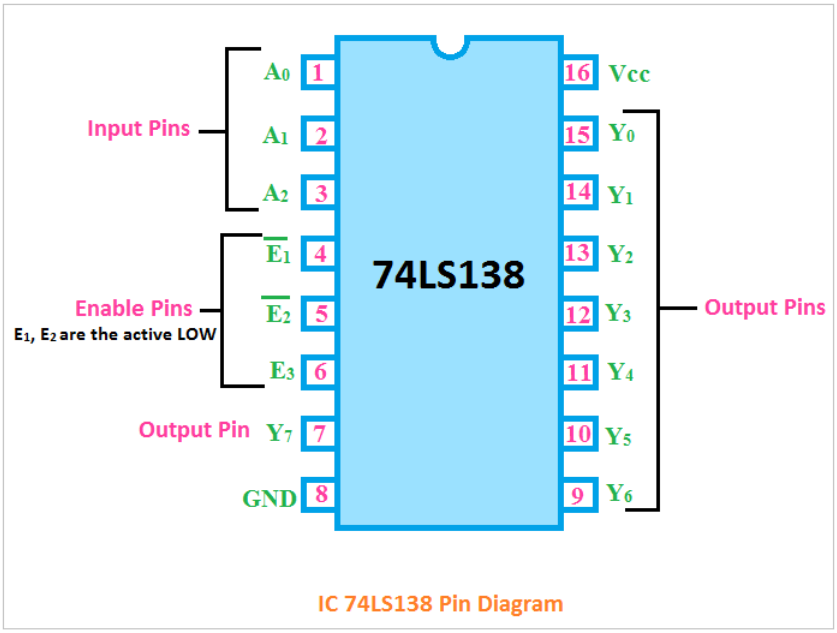
IC 7483 IC used in digital display driver circuits, counter circuits, calculator circuits, matrix keyboards, etc.

IC 74138 Pin Diagram, Truth Table, Logical Circuit, Applications

IC 74138 is a Logical Decoder IC. It also has a demultiplexing facility. The IC 74138 is available in the market with the name of 74LS138. It is a 3 to 8 decoder IC. The internal circuit of this IC is made of high-speed Schottky barrier diode.

IC 74138 Pin Diagram

The IC 74LS138 has a total of sixteen pins.



IC 74LS138 Pin Diagram

1. The pin no. 8 and 16 are the ground and V_{CC}respectively for the power input.

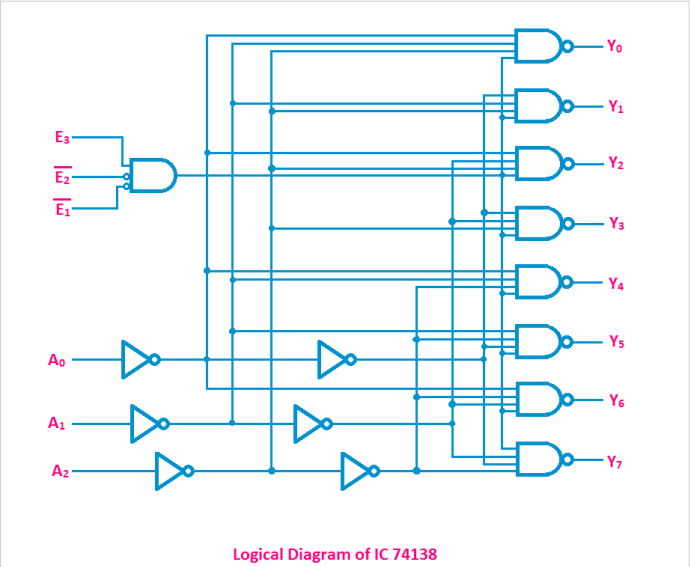
2. There are a total of three input pins (pin no. 1, 2, 3). They are denoted by A_0, A_1, A_2. So, the IC 74LS138 can take three binary input signals.

3. There are three enable input pins E_1, E_2 andE_3, (pin no. 4, 5, 6). E_1 andE_2 are the active LOW pins that mean when low signals are applied to those pins, they will be active.

4. Pin no. 7 and 9 to 15 are the output pins.

Logical Diagram of IC 74138

The logical circuit of the IC 74138 is made using NOT Gate, and AND Gates as shown in the below figure.



Logical Diagram of IC 74138

IC 74138 Truth Table

The truth table of IC 74138 is given below

INPUTS						OUTPUTS							
$\overline{E_1}$	$\overline{E_2}$	E_3	A_0	A_1	A_2	Y_0	Y_1	Y_2	Y_3	Y_4	Y_5	Y_6	Y_7
H	X	X	X	X	X	H	H	H	H	H	H	H	H
X	H	X	X	X	X	H	H	H	H	H	H	H	H
X	X	L	X	X	X	H	H	H	H	H	H	H	H
L	L	H	L	L	L	L	H	H	H	H	H	H	H
L	L	H	H	L	L	H	L	H	H	H	H	H	H
L	L	H	L	H	L	H	H	L	H	H	H	H	H
L	L	H	H	H	L	H	H	H	L	H	H	H	H
L	L	H	L	L	H	H	H	H	H	L	H	H	H
L	L	H	H	L	H	H	H	H	H	H	L	H	H
L	L	H	L	H	H	H	H	H	H	H	H	L	H
L	L	H	H	H	H	H	H	H	H	H	H	H	L

H - High, L - Low, X - Don't Care

IC 74138 Truth Table

Operating Condition of IC SN74LS138

1. The supply voltage or V_{CC} should be given between 4.75V to 5.25V.
2. Operating temperature range should be between 0 to 70-degree centigrade.

Features of IC 74138

1. It is very fast and high-speed IC.
2. It consumes very low power because it consists of low power Schottky diodes.
3. It has a demultiplexing facility.
4. It has a very short propagation delay.
5. Appropriate operating Temperature.

IC 74138 Applications

1. It is a decoder IC, the main application is to decode the digital signal.
2. They are used in digital memory circuits.
3. They are used in data routing applications.
4.They are used for demultiplexing of digital signals.

IC 74147 Pin Diagram, Internal Circuit, Truth Table

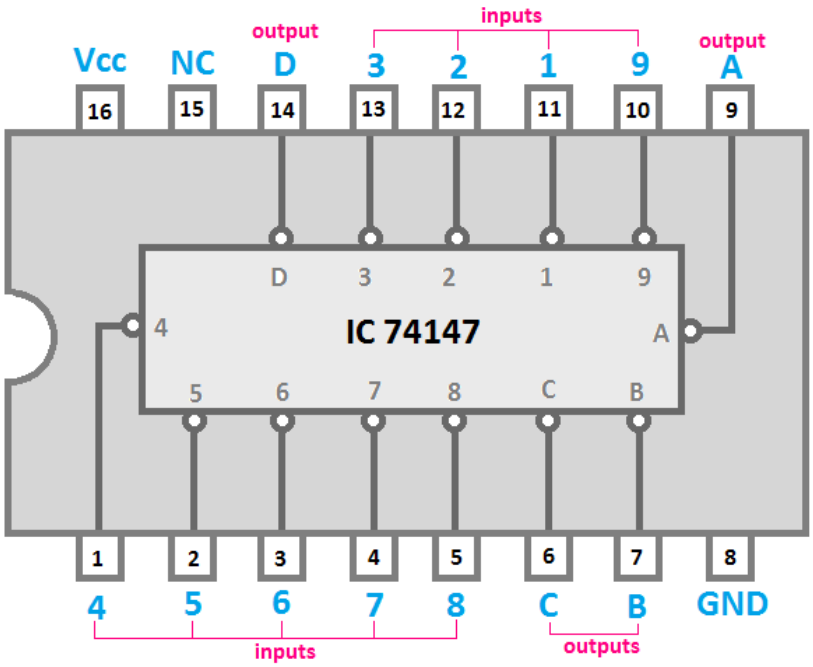
IC 74147 is a digital encoder IC that encodes 9 inputs lines into 4 output lines. It is also known as the Decimal to BCD priority encoder. The priority encoder term is used because it provides encoding for highest order data lines as a first priority. It is made up using Transistor-Transistor Logic (TTL) technology. It is a 10 to 4 encoder IC.

The general specification of this IC are,

- It operates at 4.5V to 5.5 DC voltage.
- It delivers output current from low 70µA to high 8mA
- It operates at the temperature from -55℃ to 70℃
- Logic Case packaging type: DIP
- Mounting Type: Through Hole

IC 74147 Pin Diagram

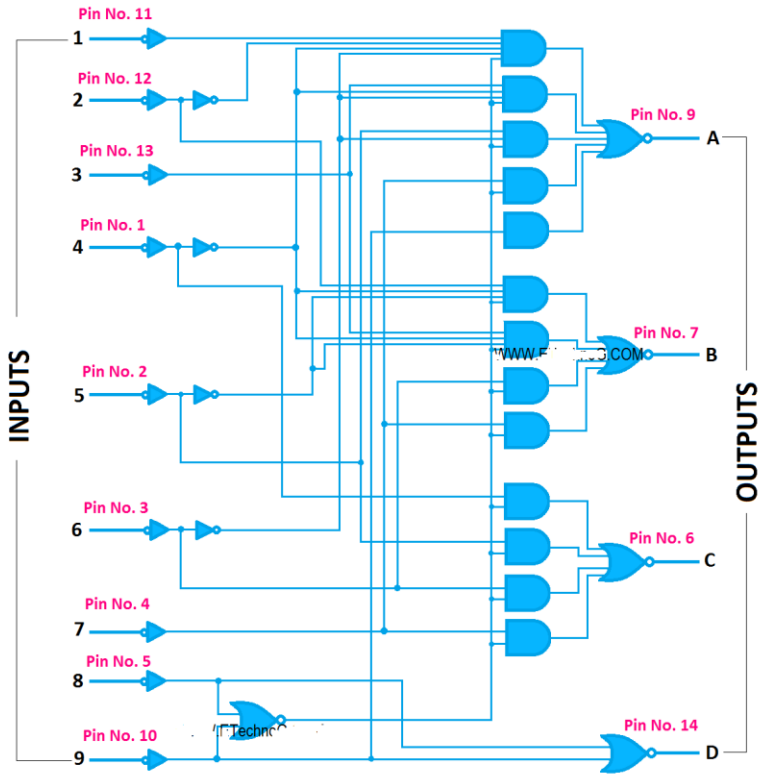
The pin diagram of IC 74147 is shown below



IC 74147 Pin Diagram

IC 74147 Internal Circuit Diagram

Here, you can see the internal circuit diagram of IC 74147



IC 74147 Internal Circuit Diagram

IC 74147 Truth Table/Function Table

The truth table of IC 74147 is shown below.

INPUTS									OUTPUTS			
1	2	3	4	5	6	7	8	9	D	C	B	A
H	H	H	H	H	H	H	H	H	H	H	H	H
X	X	X	X	X	X	X	X	L	L	H	H	L
X	X	X	X	X	X	X	L	H	L	H	H	H
X	X	X	X	X	X	L	H	H	H	L	L	L
X	X	X	X	X	L	H	H	H	H	L	L	H
X	X	X	X	L	H	H	H	H	H	L	H	L
X	X	X	L	H	H	H	H	H	H	L	H	H
X	X	L	H	H	H	H	H	H	H	H	L	L
X	L	H	H	H	H	H	H	H	H	H	L	H
L	H	H	H	H	H	H	H	H	H	H	H	L

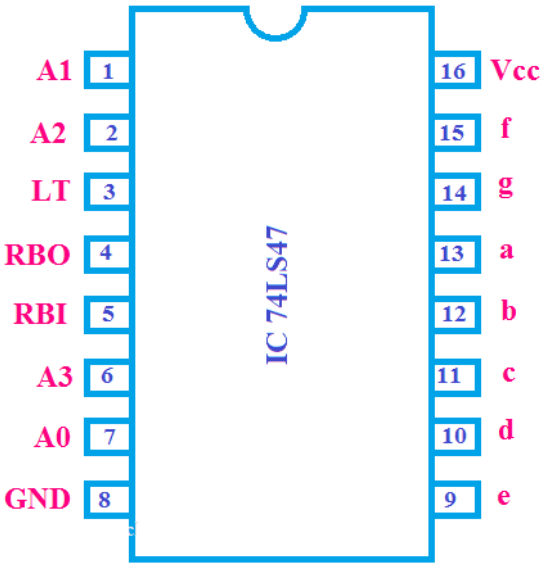
IC 74147 Truth Table/Function Table

Decoder Circuit using IC 7447

BCD means **Binary Coded Decimal**. We can represent each of the decimal numbers by its four-bit binary numbers.

To drive a Seven Segment Display we need BCD to Seven Segment Display Decoder circuit.

Pin Diagram of IC 74LS47:



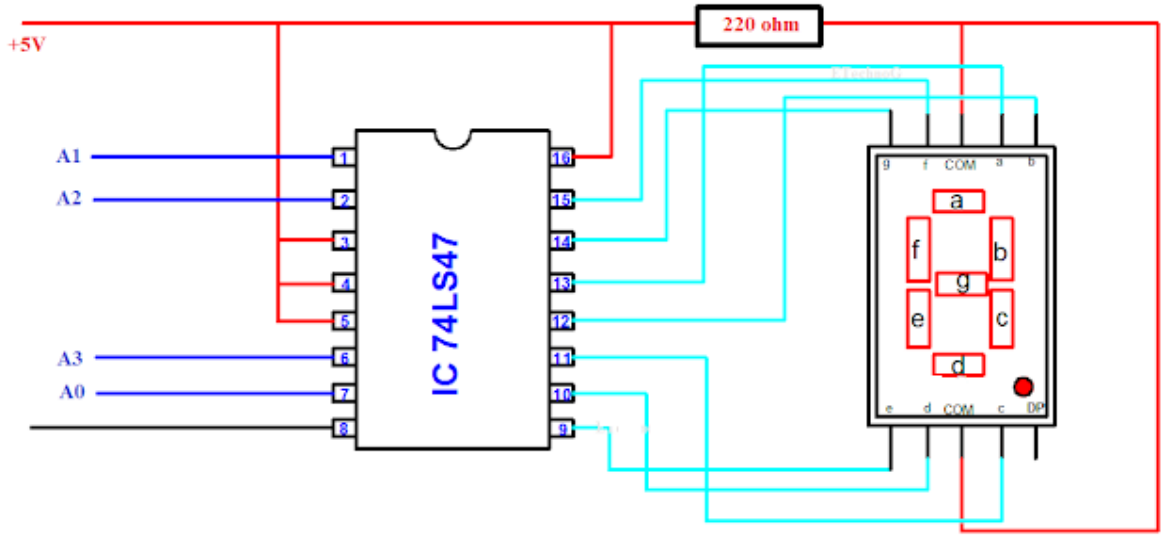
- A0, A1, A2, A3 are the BCD input pins.
- LT - for display test
- RBO, RBI is the Ripple blanking output and Ripple blanking Input pins respectively.
- a, b, c, d, e, f, g are the outputs for seven segment display.

We have four pins to give BCD inputs so we can display from 0 to 15 numbers but we have need two seven segment display. In the below circuit diagram one seven segment display is used so here 0 to 9 numbers can be shown.

The **truth table** is given below for the circuit,

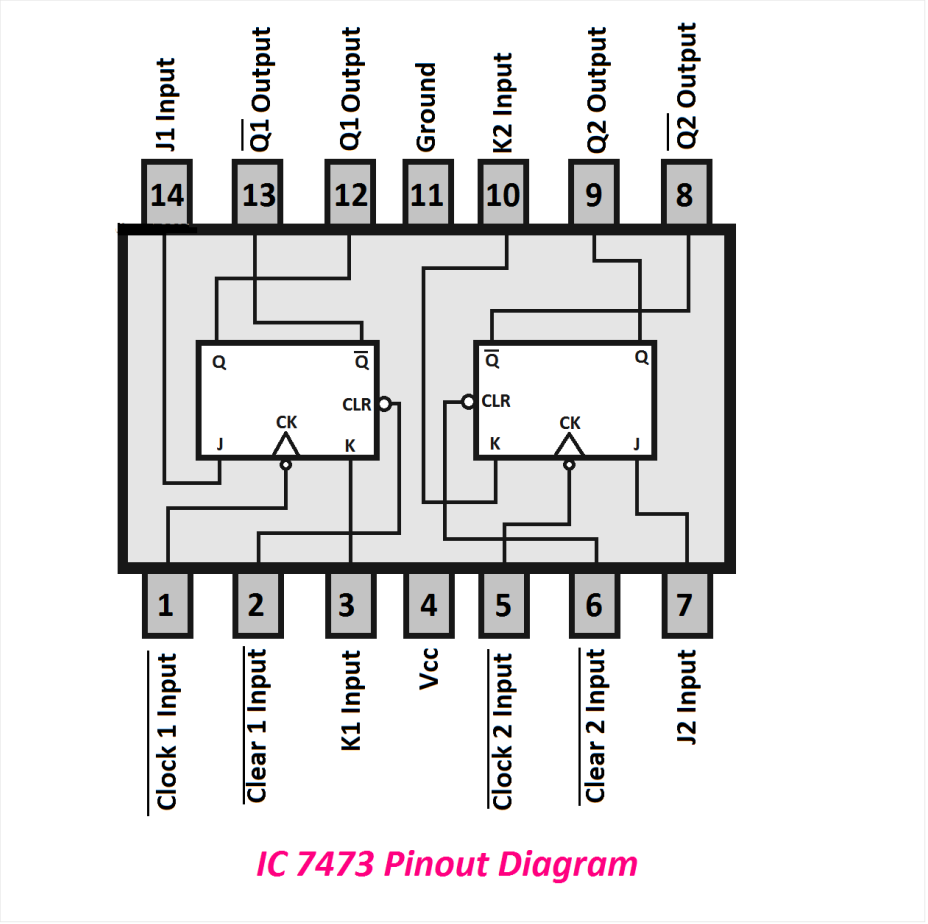
INPUTS						OUTPUTS							
LT	RBO	A3	A2	A1	A0	a	b	c	d	e	f	g	Decimal Number show in display
0	0	0	0	0	0	1	1	1	1	1	1	0	0
0	0	0	0	0	1	0	1	1	0	0	0	0	1
0	0	0	0	1	0	1	1	0	1	1	0	1	2
0	0	0	0	1	1	1	1	1	1	0	0	1	3
0	0	0	1	0	0	0	1	1	0	0	1	1	4
0	0	0	1	0	1	1	0	1	1	0	1	1	5
0	0	0	1	1	0	1	0	1	1	1	1	1	6
0	0	0	1	1	1	1	1	1	0	0	0	0	7
0	0	1	0	0	0	1	1	1	1	1	1	1	8
0	0	1	0	0	1	1	1	1	1	0	1	1	9

Circuit diagram of Decoder Circuit:



IC 7473 PinOut Diagram

The pin diagram of IC 7473 is shown below.



IC 7473 Pinout Diagram

The IC 7473 or mostly known as 74LS73 is a dual JK flip Flop logic gate IC. It is a negative edge-triggered IC. It has two independent JK Flip Flops. They have complementary outputs. These Flip Flops are negative edge triggered. It is also referred to as a JK Master Slave Flip Flop. During the positive transition of the clock input, the data got transferred from the J and K input to the master while during the negative transition the data is transferred from the master to slave. So you may understand the data first transferred from the J and K input to master then it transfer to the slave.

Now, let's see the features of the IC 7473
1. It has two J-K Master-Slave Flip Flop
2. It supports wide operating conditions and a large operating voltage range.
3. The output of the IC 7473 is directly CMOS, NMOS, and TTL Logic.
The IC 74LS73 can be operated from 0 degrees to 70 degrees centigrade temperature. And

when the voltage at its output terminal is reach 2V, it will be considered as output HIGH, and when the voltage decreases to 0.8V then it will be considered as output LOW.

IC 7473 truth table

The truth table of IC 7473 is given below

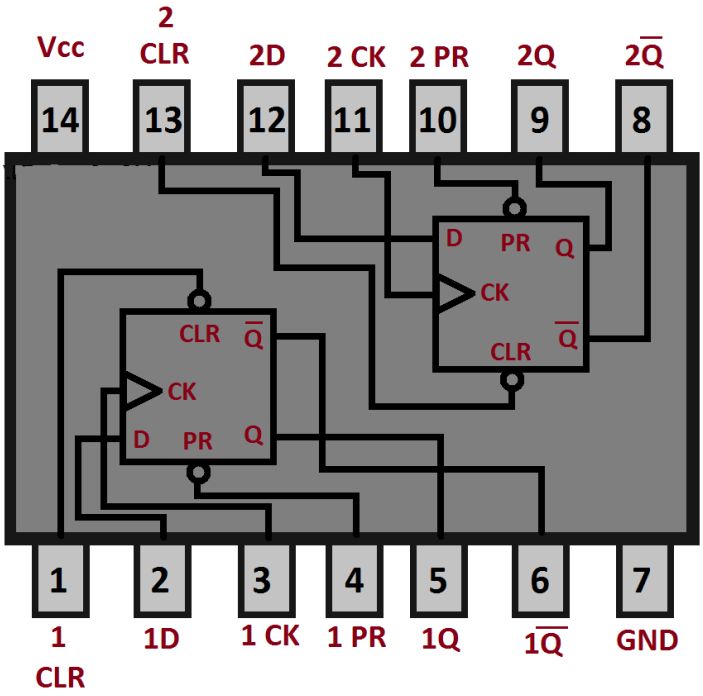
INPUTS				OUTPUTS	
CLR	CLK	J	K	Q	$\overline{Q}$
L	X	X	X	L	H
H	↑	L	L	Q_0	$\overline{Q_0}$
H	↑	H	L	H	L
H	↑	L	H	L	H
H	↑	H	H	Toggle	

H-High logic level, X-Either LOW or HIGH logic level, L-LOW logic level, ↑-Positive going transition of the clock

IC 7473 Truth Table

IC 7474 Pinout Diagram

The pin diagram of IC 7474 is shown below



IC 7474 Pinout Diagram

IC 7474 or mostly known as IC 74LS74 is a dual D Flip Flop positive edge-triggered IC. It has two independent D Flip Flops with complementary outputs. It has D input and Q output. The data in the D input may be changed during the high or low clock but it does not affect the output and the delay times also do not affect. The IC 7474 can be operated up to 7V voltage and 0 to +70 degrees centigrade temperature. The main features of the IC 74LS74 are, it provides very fast switching, low propagation delay, large operating mode, etc.

IC 7474 truth table

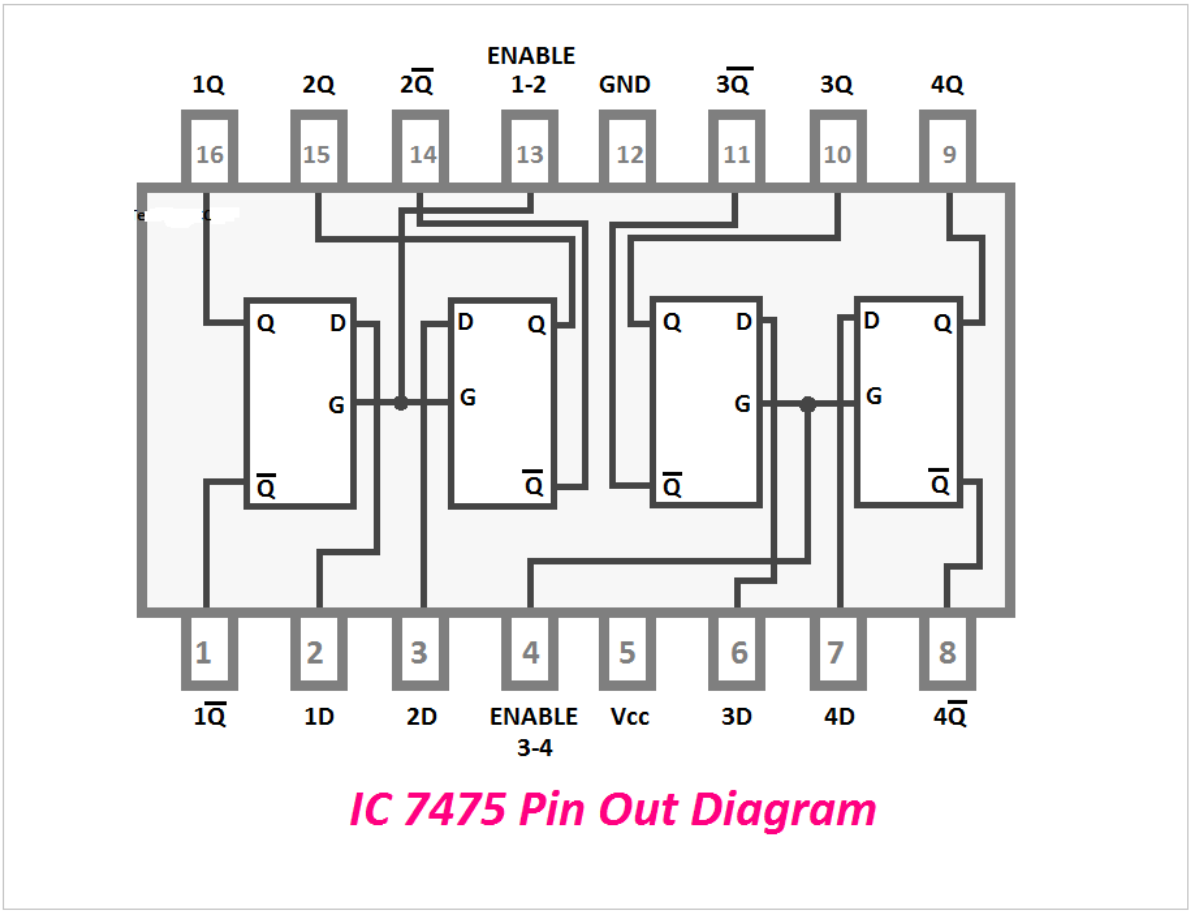
The truth table of IC 7474 is given below

INPUTS				OUTPUTS	
PR	CLR	CLK	D	Q	$\overline{Q}$
L	H	X	X	H	L
H	L	X	X	L	H
L	L	X	X	H	H
H	H	↑	H	H	L
H	H	↑	L	L	H
H	H	L	X	Q_0	$\overline{Q_0}$

H-High logic level, X-Either LOW or HIGH logic level, L-LOW logic level, ↑-Positive going transition of the clock IC 7474 Truth Table

IC 7475 Pinout Diagram

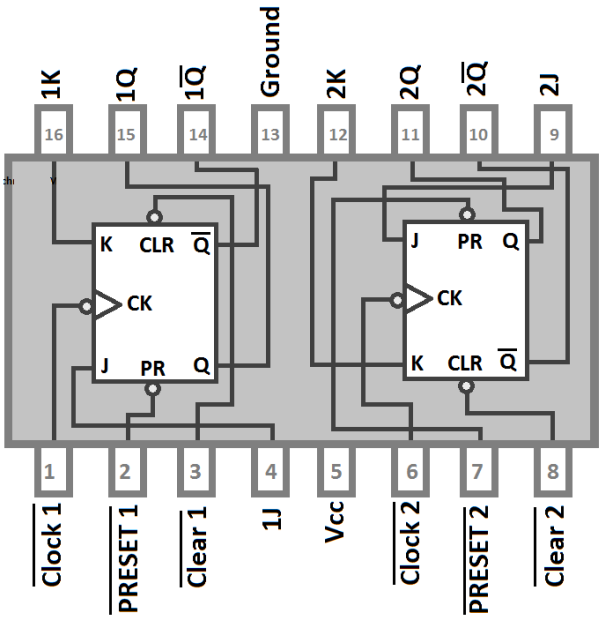
The pin diagram of IC7475 is shown below.



The IC 7475 mostly known as 74HC75 is a 16 pin IC that contains four independent and transparent D latches. You will be seen in the pin diagram the two latches have a common enable which means the first two latches have the same enable and the second two latches have the same enable. Among the different inputs, the D and clock inputs are synchronous inputs whereas Set and Reset inputs are asynchronous inputs. The main features of the IC 7475 are, it has 4-bit bistable latches, low operating voltage ranges, and wide operating conditions. It can be operated from 4.75 to 5.25V voltage ranges and 0 to +70 degree centigrade temperature.

IC 7476 Pinout Diagram

The pin diagram of IC 7476 is shown below.



IC 7476 Pinout Diagram

The IC 7476 commonly known as IC 74LS76 is a dual JK flip IC with Set and Clear Input. The Clock, Clear inputs are active low inputs. During the High-Low Clock Transition, input data is transferred to the output. It is also a 16 pin IC. The IC 7476 can be operated from 4.75 to 5.25V and 0 to +70 degrees centigrade temperature. It works with standard TTL voltage and provides a very fast switching speed.

IC 7476 truth table

The truth table of IC 7476 is given below

INPUTS					OUTPUTS	
PR	CLR	CLK	J	K	Q	$\overline{Q}$
L	H	X	X	X	H	L
H	L	X	X	X	L	H
L	L	X	X	X	H	H
H	H	↑	L	L	Q_0	$\overline{Q_0}$
H	H	↑	H	L	H	L
H	H	↑	L	H	L	H
H	H	↑	H	H	Toggle	

H-High logic level, X-Either LOW or HIGH logic level, L-LOW logic level, ↑-Positive going transition of the clock

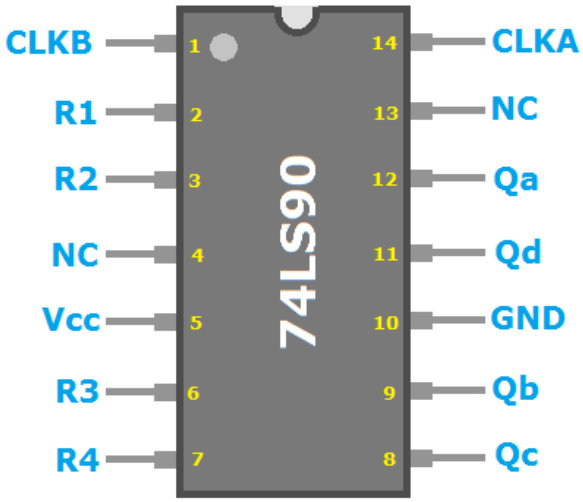
IC 7476 Truth Table

IC 7490 Pin Diagram, Truth Table, Internal Circuit, Application

IC 7490 is a decade counter IC which can generate output code in BCD.
It is an Asynchronous Decade Counter IC.
IC 7490 can count the binary numbers from 0000 to 1001. After 1001 it gets reset and again starts counting. As the IC 7490 gets reset after counting ten numbers, it is called MOD-10 Decade Counter.

IC 7490 Pin Diagram

The pin diagram of IC 7490 is shown below. Here the actual IC name is 74LS90.



IC 7490 Pin Diagram

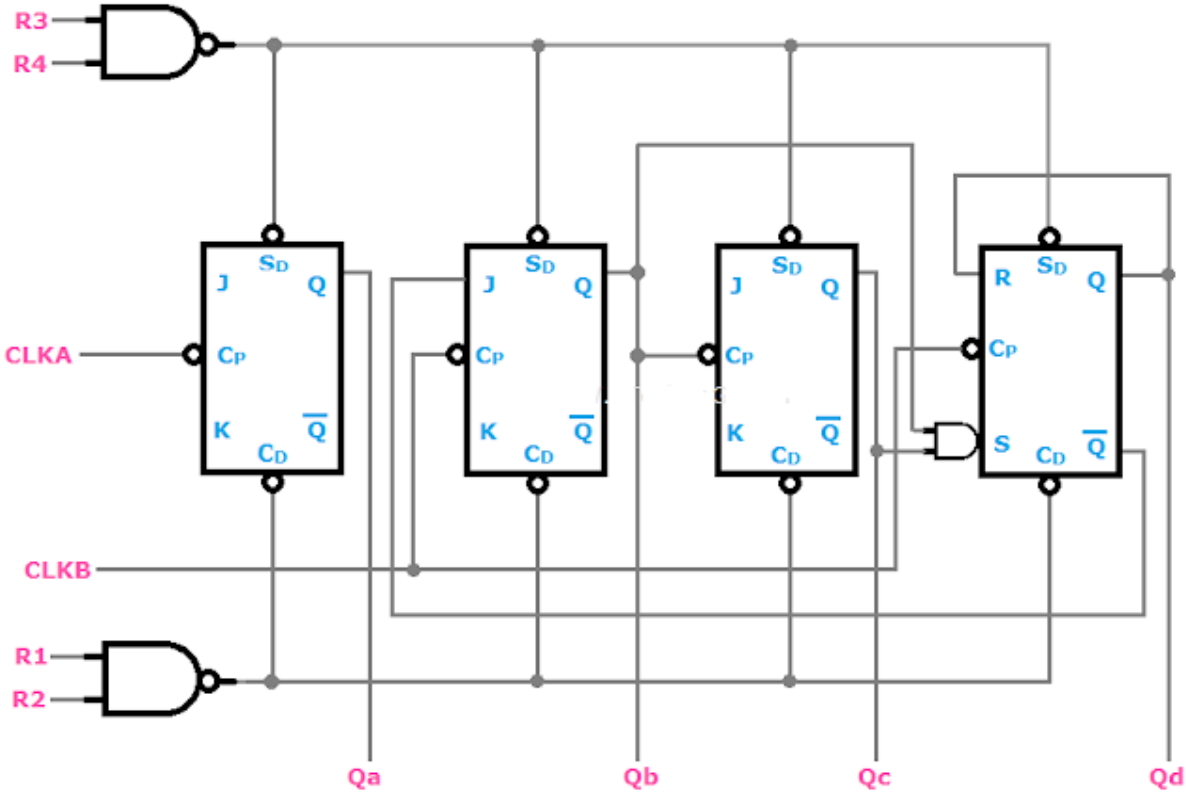
IC 7490 Truth Table

The truth table of IC 7490 is shown below. After counting ten number (0 to 9) it gets reset and start counting again.

Count	Qd	Qc	Qb	Qa
(Start) 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
(New Cycle) 10	0	0	0	0

IC 7490 Internal Circuit

The internal circuit of IC 7490 is shown below. It consists of four numbers of Master-Slave JK flip-flops that are internally connected.



IC 7490 Internal Circuit

The first flip-flop is independent and is driven by the CLKA pin. Another three flip-flops are connected to the CLKB pin.

IC 7490 Applications

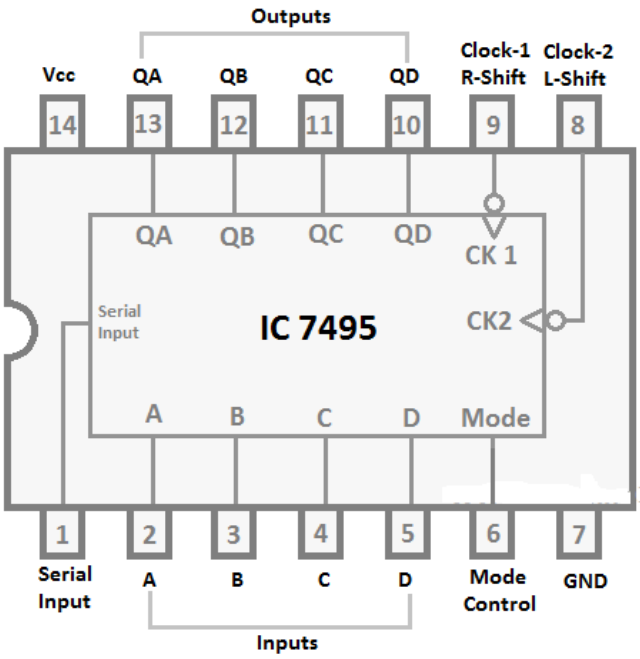
1. Mainly, IC 7490 is used in the digital counter circuits.
2. IC 7490 also used in digital timers and clocks.
3. IC 7490 is used in automatic controller circuits.

IC 7495 Pin Diagram, Internal Circuit, Truth Table

IC 7495 is a shift register IC. It is also known as IC74LS95. It is a 4-bit device and having both serial and parallel synchronous operating modes. In this article, we are going to see a pin diagram of IC 7495, IC 7495 internal circuit, IC 7495 truth table, or functional table. The main application of this IC 7495 is encryption and decryption in digital circuits.

IC 7495 Pin Diagram

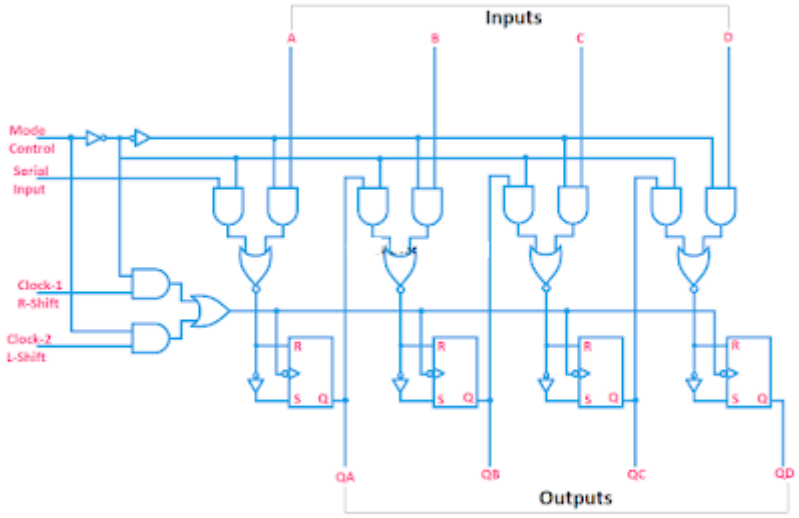
The pin diagram of IC 7495 is shown below.



IC 7495 Pin Diagram

IC 7495 Internal Circuit Diagram

The internal circuit diagram of IC 7495is shown below.



IC 7495 Internal Circuit

The left and right shifts are operated by separate clocks. The R-shift is operated by clock 1 and the left shift is operated by clock 2. You can also see the mode control terminal is connected to the AND gate which is also connected to the Clock 2 terminal. The four outputs have come from four RS Flip Flop Circuits. You can see, the mode control terminal is connected through the clamp diodes. These clamp diodes helped to limit the high-speed termination effect. Here, all the shifts and parallel loads are synchronous.

IC 7495 Truth Table

The functional table or truth table of IC 7495 is shown below.

Inputs								Outputs			
Mode Control	Clocks		Serial	Parallel				QA	QB	QC	QD
	2(L)	1(R)		A	B	C	D				
H	H	X	X	X	X	X	X	Q_{A0}	Q_{B0}	Q_{C0}	Q_{D0}
H	↓	X	X	a	b	c	d	a	b	c	d
H	↓	X	X	Q_B	Q_C	Q_D	d	Q_{Bn}	Q_{Cn}	Q_{Dn}	d
L	L	H	X	X	X	X	X	Q_{A0}	Q_{B0}	Q_{C0}	Q_{D0}
L	X	↓	H	X	X	X	X	H	Q_{An}	Q_{Bn}	Q_{Cn}
L	X	↓	L	X	X	X	X	L	Q_{An}	Q_{Bn}	Q_{Cn}
↑	L	L	X	X	X	X	X	Q_{A0}	Q_{B0}	Q_{C0}	Q_{D0}
↓	L	L	X	X	X	X	X	Q_{A0}	Q_{B0}	Q_{C0}	Q_{D0}
↓	L	H	X	X	X	X	X	Q_{A0}	Q_{B0}	Q_{C0}	Q_{D0}
↑	H	L	X	X	X	X	X	Q_{A0}	Q_{B0}	Q_{C0}	Q_{D0}
↑	H	H	X	X	X	X	X	Q_{A0}	Q_{B0}	Q_{C0}	Q_{D0}

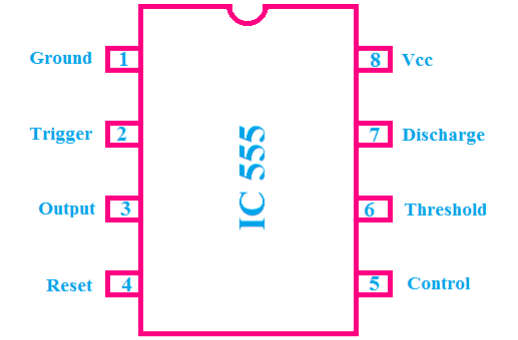
IC 7495 Truth Table

IC 555 Applications, Pin Diagram, Internal Circuit Diagram

The **IC 555** is very useful and popular IC. The IC 555 basically a timer IC. The IC 555 is consisting of a timer circuit and the timer circuit contains three five kilo Ohm resistors.

Pin Diagram of IC 555:

The IC 555 is available in so many packages. Here the block diagram of 8 pin DIP package IC 555 is given below.



IC 555 Pin Diagram
8 Pin DIP Package

The function of Each Pin of IC 555:

Pin no.1: The pin 1 provides the ground terminal for the IC to give the power supply.

Pin no.2: The pin 2 is called Trigger. It is connected to the inverting terminal of the second OpAmp or Comparator. As the non-inverting terminal of the second comparator is connected to the 1/3Vcc point so if we decrease the voltage of the trigger pin below the 1/3 Vcc then the output of the comparator will be high and the circuit will be triggered. Usually, we connect the trigger pin with the V_{CC} and when we want to change the current state of the comparator, we should decrease the voltage of the trigger pin.

Pin no.3: The Pin 3 is called output. The pin 3 is to be connected to the load that means the IC can drive the load through the output pin. The IC can deliver up to 200mA of current through thispin.

Pinno.4: The pin no.4 is used to reset the flip-flop circuit.

Pin no.5: The pin no.5 is directly connected to the inverting terminal of the first comparator. This pin is used to control the input voltage of the first comparator by applying external voltage.

Pin no.6: As the inverting terminal of the first comparator is connected to the 2/3 Vcc point so when the voltage of the non-inverting terminal is greater than2/3 Vcc the output of the first comparator will be high. The pin no.6 or threshold pin is used to give the threshold voltage which is greater than 2/3 Vcc.
Actually, the pin no.2 or Trigger and pin no.6 or Threshold pin are used to control the output

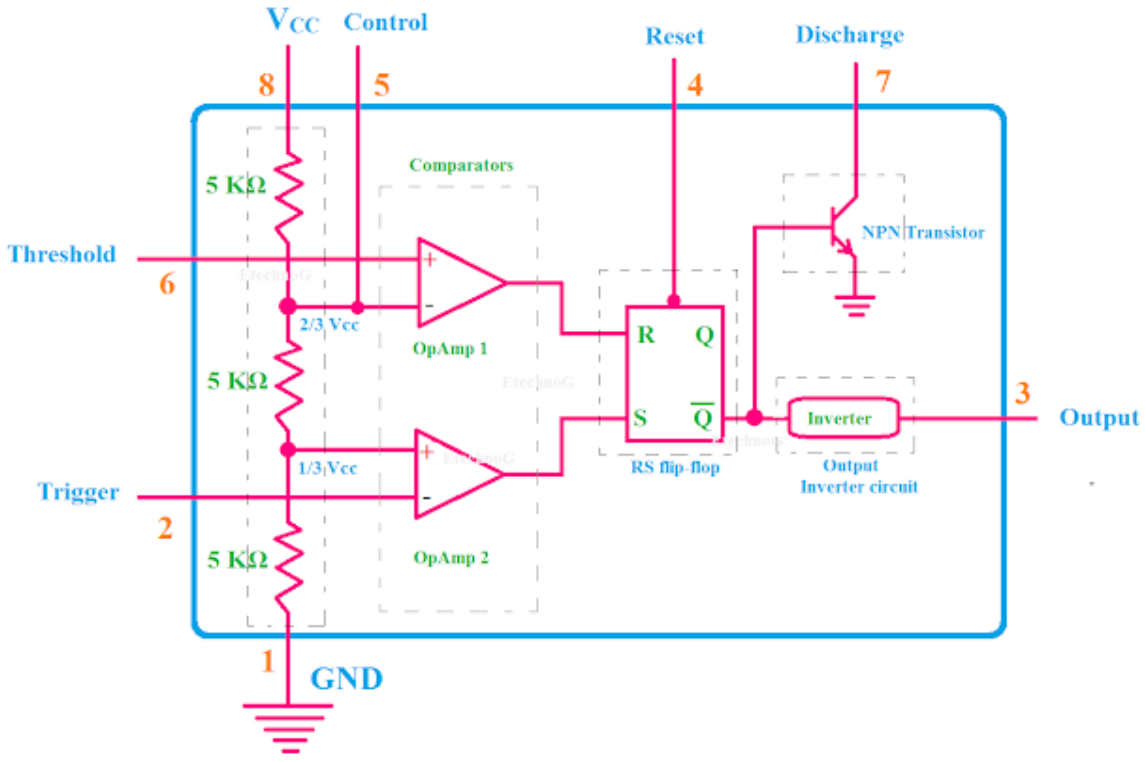
of the comparators.

Pin no.7: The pin no.7 is called discharge pin. It is connected to the collector pin of the transistor.

Pin no.8: The pin no.8 or Vcc is used to give power supply to the IC555. The voltage from 4.5V to 15V can be given to the IC555.

The internal block diagram of IC 555:

You can see in the below figure the internal circuit has no. of blocks. The first block is the voltage divider circuit. The voltage divider circuit is made by three five kiloOhm resistors connected in series. The next block is comparators. Here the comparator using OpAmp is used. There are two comparators in the circuit. The next block is the RS Flip-flop circuit.
The outputs of the comparators are given to the flip-flop circuit. Then the output of the flip-flop circuit is going to the output inverter circuit. The function of the inverter circuit is that it inverse the output of the flip-flop that means if the output of the flip-flop is high then the output of the inverter will be low. The pin no.3 or output pin of the IC is connected to the output of the inverter circuit. A discharge circuit using NPN transistor also connected to the output of the flip-flop circuit.



IC 555 Block Diagram

Applications of IC 555:

The IC 555 is mainly used for timing related applications. Like,

1. The IC 555 is used for Tone generation.
2. It is used to make an alarm circuit.
3. They also used for frequency division applications.
4. The IC 555 is used as a relaxation oscillator.
5. They are also used in digital counter circuits.
6. IC 555 is widely used for electronics projects.