# BHUBANANANDA ORISSA SCHOOL OF ENGINEERING , CUTTACK

# DEPARTMENT OF ELECTRONICS AND TELECOMMUNICATION ENGINEERING

# LAB MANUAL

Year & Semester: 2nd Year, III Semester
Subject Code/Name: PR-3, DIGITAL ELECTRONICS LAB

# EXPERIMENTNo-1

**OBJECTIVE:-**Verify Truth Table of Logic Gates (AND,OR,NOT,NAND&NOR Gates).

**EQUIPMENT REQUIRED:-**All the basic gate mention in the fig.

**THEORY:-**There are different gates available in digital electronic filed. Which is used to perform the different tasks they are as follow:-
ANDGATE:-This gate is used for the multiplication of two binary digits
. ORGATE: - This gate is used for the addition of two binary digits.
NOTGATE:-This gate gives the compliment of given binary digit.
NANDGATE:-This gate is used for giving compliment of multiplication of two binary digits. NORGATE: -This gate is used for giving compliment of the addition of two binary digits
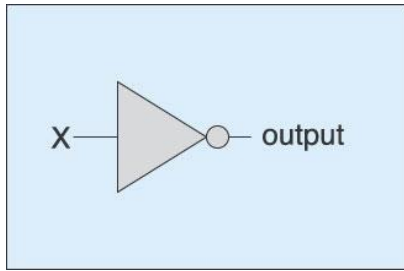
**DAIGRAMS:-**

*2-input AND gate*

Input$_A$ ——
Input$_B$ —— —— Output

| A | B | Output |
|---|---|--------|
| 0 | 0 | 0 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

*2-input OR gate*

Input$_A$ ——
Input$_B$ —— —— Output

| A | B | Output |
|---|---|--------|
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 1 |

**1- input NOT gate**                    **2-input NAND gate**
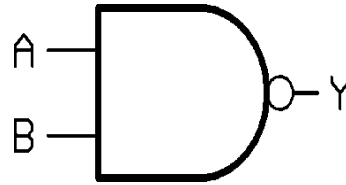
| A | output |
|---|--------|
| 1 | 0 |
| 0 | 1 |

| A | B | output |
|---|---|--------|
| 0 | 0 | 1 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |

2- **Input NOR gate**
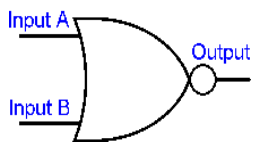


| A | B | output |
|---|---|--------|
| 0 | 0 | 1 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 0 |

**PROCEDURE:-**

1. Place the IC on IC Trainer Kit.

2. Connect $V_{CC}$ and ground to respective pins of IC Trainer Kit.

3. Connect the inputs to the input switches provided in the IC Trainer Kit.

4. Connect the outputs to the switches of O/P LEDs,

5. Apply various combinations of inputs according to the truth table and observe condition of LEDs.

6. Disconnect output from the LEDs and note down the corresponding multimeter voltage readings for various combinations of inputs.

**RESULT:-**Success fully verify the truth table of logic gate.

**PRECAUTIONS:-**1)All the connection should be tight.
    2) It should be care that the values of the components of the circuit is does not
      exceed to their ratings (maximum value).
    3) Before the circuit connection it should be checkout working condition
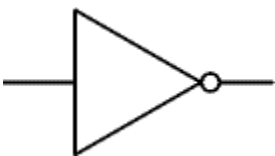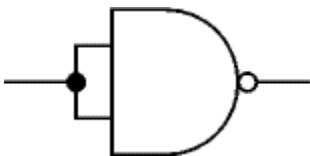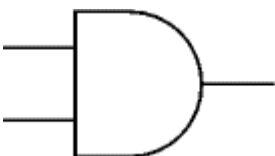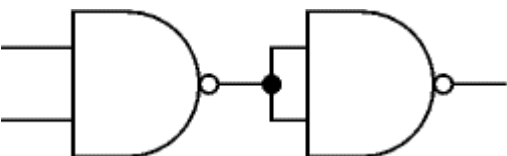     fall the Component.

## EXPERIMENTNo-2
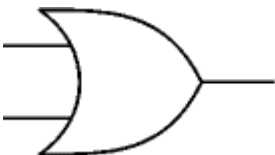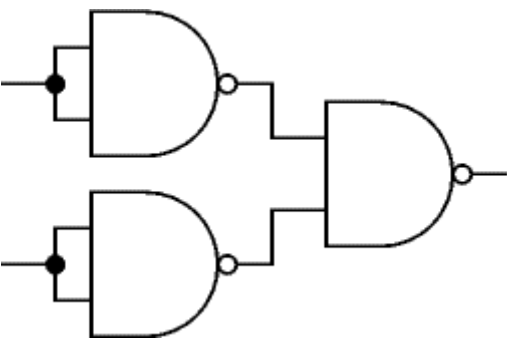
**OBJECTIVE:-**Design Basic Gates Using NAND gates.

**EQUIPMENT  REQUIRED:-**NAND gate ,connecting wires etc.

**THEORY:-** NAND GATE is a universal gate. It is so called as because by using of this gate we can make any gate like not, or , and etc. by help of this gate we can also make multiplexers and de mux .

## NAND gate equivalents

The table below shows the NAND gate equivalents of NOT, AND,OR and NOR gates:

| Gate | Equivalent in NAND gates |
|---|---|
| NOT |  |
| AND |  |
| OR |  |

| NOR |  |  |
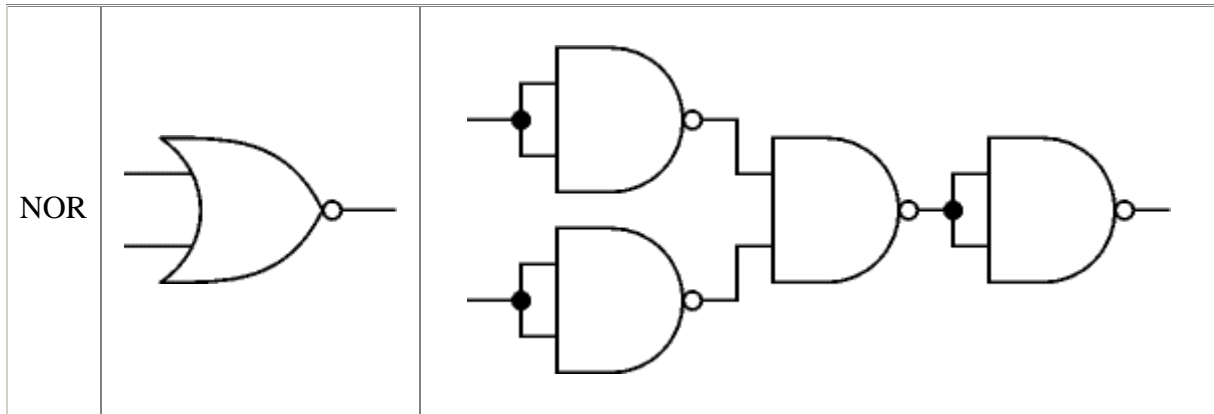
.

**PROCEDURE:-**

1. Place the IC on IC Trainer Kit.
2. Connect $V_{CC}$ and ground to respective pins of IC Trainer Kit.
3. Connect the inputs to the input switches provided in the IC Trainer Kit.
4. Connect the outputs to the switches of O/P LEDs,
5. Apply various combinations of inputs according to the truth table and observe condition of LEDs.
6. Disconnect output from the LEDs and note down the corresponding multimeter voltage readings for various combinations of inputs.

**RESULT:-**Designing of basic gates by using of NAND gate is success fully done.

**PRECAUTIONS:-**1)All the connection should be tight.
2) It should be care that the values of the components of the circuit is does not exceed to their ratings (maximum value).
3) Before the circuit connection it should be checkout working condition of all the Component.

**OBJECTIVE:-**Design Basic Gates Using NOR gates.

**EQUIPMENT  REQUIRED**:-NOR gate, connecting wires

etc.

**THEORY:-** Like NAND gates, NOR gates are so-called "universal gates" that can be combined to form any other kind of logic gate. For example, the first embedded system, Apollo Guidance Computer, was built exclusively from NOR gates, about 5,600 in total for the later versions. Today, integrated circuits are not constructed exclusively from a single type of gate. Instead, EDA tools are used to convert the description of a logical circuit to a net list of complex gates (standard cells) or transistors (full custom approach).

# NOR GATE

A NOR gate is logically an inverted OR gate. By itself has the following truth table:



Truth Table

| Input A | Input B | Output Q |
|---------|---------|----------|
| 0 | 0 | 1 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 0 |

# Making other gates by using NOR gates

A NOR gate is a universal gate, meaning that any other gate can be represented as a combination of NOR gates.

## NOT

Truth Table

| Input A | Output Q |
|---------|----------|

7

This is made by joining the inputs of a NOR gate. As a NOR | 0 | 1 | gate
Is equivalent to an OR gate leading to NOT gate, this | 1 | 0 |
Automatically sees to the "OR" part of the NOR gate,
Eliminating it from consideration and leaving only the NOT part.

**Desired Gate**          **NOR Construction**
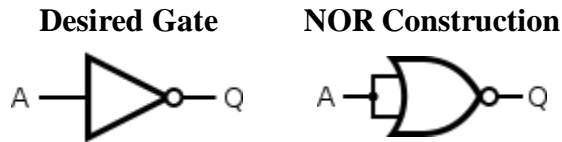


## OR

The OR gate is simply a NOR gate followed by a NOT gate.

**Desired Gate**          **NOR Construction**
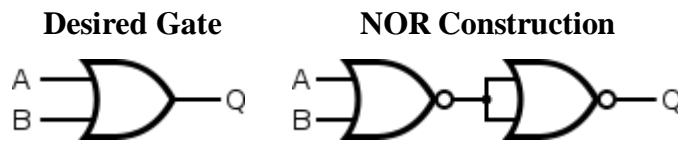


Truth Table

| Input A | Input B | Output Q |
|---------|---------|----------|
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 1 |

## AND

An AND gate gives a 1output when both inputs are1;a NOR gate gives a 1output only when both inputs are 0. Therefore, an AND gate is made by inverting the inputs to a NOR gate.

**Desired Gate**          **NOR Construction**
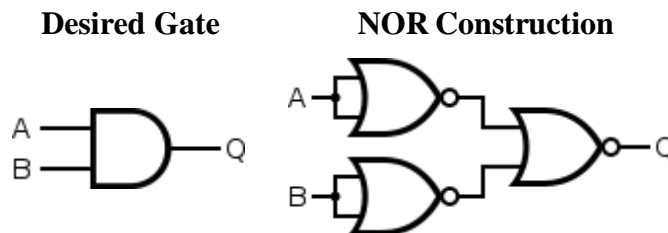
Truth Table

| Input A | Input B | Output Q |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

## **NAND**

A NAND gate is made using an AND gate in series with a NOT gate:

**Desired Gate**                    **NOR Construction**
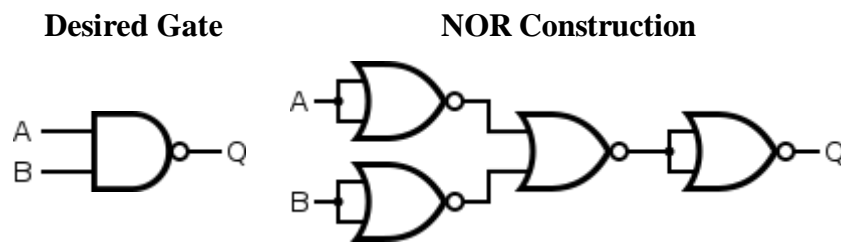


Truth Table

| Input A | Input B | Output Q |
|---|---|---|
| 0 | 0 | 1 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |

## **XOR**

An XOR gate is made by connecting the output of 3 NOR gates (connected as an AND gate)and the output of a NOR gate to here spective inputs of a NOR gate. This expresses the logical formula (AANDB) NOR (ANORB). This construction entails a propagation delay three times that of a single NOR gate.

**Desired Gate**                    **NOR Construction**



Truth Table

9

| Input A | Input B | Output Q |
|:---:|:---:|:---:|
| 0 | 0 | 1 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

| Input A | Input B | Output Q |
|:---:|:---:|:---:|
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |

## XNOR

An XNOR gate can be constructed from four NOR gates implementing the expression "(A NOR N) NOR (B NOR N) where N = A NOR B". This construction has a propagation delay three times that of a single NOR gate, and uses more gates.

**Desired Gate**          **NOR Construction**
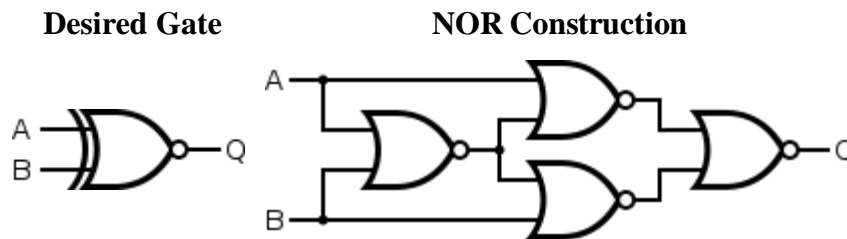


**RESULT**:-Designing of basic gates by using of NOR gate is successfully done.

**PRECAUTIONS:-**1)The continuity of the connecting terminals should be checked before going
2) It should be care that the values of the components of the circuit is does not
exceed to their ratings (maximum value).
3) Before the circuit connection it should be checkout working condition
of all the Component.

## EXPERIMENTNo-4

**OBJECTIVE:**-Verify De morgan's theorem.

**EQUIPMENT REQUIRED:**-All the basic gate mention in the fig.

**THEORY: -** The law is named after <u>Augustus De Morgan</u> (1806–1871) who introduced a formal version of the laws to classical <u>propositional logic</u>. De Morgan's formulation was influenced by algebraization of logic undertaken by <u>GeorgeBoole</u>,which later cemented De Morgan's claim to the find. Although a similar observation was made by <u>Aristotle</u> and was known to Greek and Medieval logicians(in the 14th century <u>William of Ockham</u> wrote down the words that would result by reading the laws out), De Morgan is given credit for stating the laws formally and in corporating them in to the language of logic. De Morgan's Law scan be proved easily , and may even seem trivial. N one the less, these laws are helpful in making valid inferences in proofs and deductive arguments.

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

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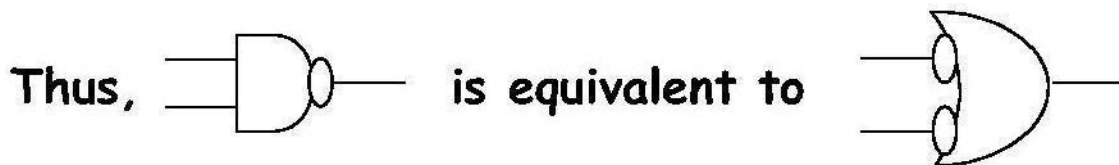


These can be generalized to more than two variables : to

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

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

**Synthes is of logic circuits:-**
Many problems of logic design can be specified using a truth table .Give such a table,can you design the logic circuit?

Design a logic circuit with three inputs A, B, Can done output F such that F=1 only whena majority of the inputs is equal to 1.

| A | B | C | F |
|---|---|---|---|
| 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 |
| 0 | 1 | 0 | 0 |
| 0 | 1 | 1 | 1 |
| 1 | 0 | 0 | 0 |
| 1 | 0 | 1 | 1 |
| 1 | 1 | 0 | 1 |
| 1 | 1 | 1 | 1 |

Sum of product form

$$F=\bar{A}.B.C+A.\bar{B}.C+A.B.\bar{C}+A.B.C$$

**Simplification of Boolean functions:-** Using the theorems of Boolean Algebra, the algebraic forms of functions can often be simplified, which leads to simpler (and cheaper) implementations.

**RESULT**:-Verification of De morgan's theorem is successfully done.

**PRECAUTIONS:-**1)The continuity of the connecting terminals should be checked before going
2) It should be care that the values of the components of the circuit is does not
     exceed to their ratings  (maximum value).
3) Before the circuit connection it should be checkout working condition
     of all the Component.

## EXPERIMENTNo-5

**OBJECTIVE:** **-** Design Half Adder. (a) Using AND/OR/NOT Gates. (b) Using NAND/NOR Gates.

**EQUIPMENT REQUIRED:**- All the basic gate mention in the fig., IC 7486, IC 7432, IC 7408, IC 7400, etc.

**THEORY:-**Half adder is a digital device by using of this device we can able to add two bit data. By addition of this two digit we gate add & carry. n electronics, an **adder** or **summer** is a digital circuit that performs addition of numbers. In modern computers adders reside in the arithmetic logic unit (ALU) where other operations are performed. Although adders can be constructed for many numerical representations, such as Binary- coded decimal or excess-3, the most common adders operate on binary numbers. In cases where two's complement or one's complement is being used to represent negative numbers, it is trivial to modify an adder into an adder- subtractor. Other signed number representations require a more complex adder.

**PROCEDURE:-**
      1. Verify the gates.

      2. Make the connections as per the circuit diagram.

      3. Switch on VCC and apply various combinations of input according to the truth table

      4. Note down the output readings for half/full adder and half/full subtractor sum/difference and the carry/borrow bit for different combination of input.

  **Half Adder using basic gates:-**
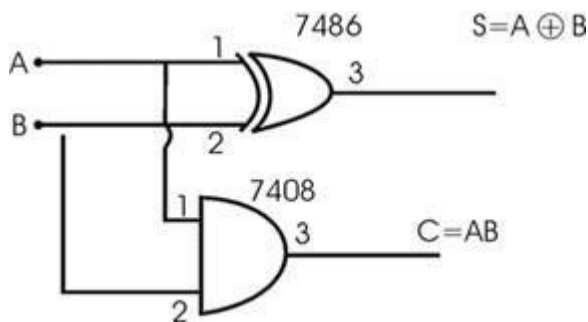


$$S = \overline{A}B + A\overline{B}$$

$$S = A \oplus B$$

$$C = AB$$

**Half Adder using NAND gates only:-**
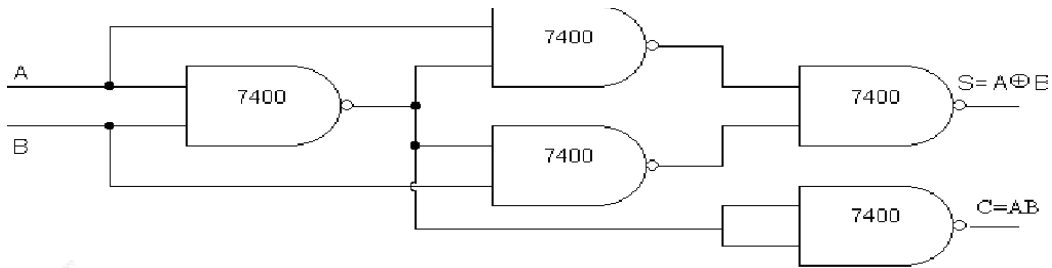
| Half Adder | | | | | |
|---|---|---|---|---|---|
| A | B | S | C | S(V) | C(V) |
|   | 0 | 0 | 0 |   |   |
| 0 | 1 | 1 | 0 |   |   |
| 1 | 0 | 1 | 0 |   |   |
| 1 | 1 | 0 | 1 |   |   |

**RESULT:-**       The designing of half adder is successfully done..

**PRECAUTIONS:-**1)The continuity of the connecting terminals should be checked before going .
          2)  It should be care that the values of the components of the circuit is does not
exceed to their ratings (maximum value).
          3) Before the circuit connection it should be checkout working condition
      of all the Component.

**OBJECTIVE:**-Design full
Adder.

**EQUIPMENT REQUIRED:**- All the basic gate mention in the fig., IC 7486, IC 7432, IC 7408, IC 7400, etc.

**THEORY:-**A **full adder** adds binary numbers and accounts for values carried in as well as out. A one-bit full adder adds three one-bit numbers, often written as *A*, *B*, and $C_{in}$; *A* and *B* are the operands, and $C_{in}$ is a bit carried in (in theory from a past addition). The circuit produces a two-bit output sum typically represented by the signals $C_{out}$ and *S*, where $\text{sum} = 2 \times C_{out} + S$. The one-bit full adder's truth table is:

**PROCEDURE:-**

　　1. Verify the gates.

　　2. Make the connections as per the circuit diagram.

　　3. Switch on VCC and apply various combinations of input according to the truth table

　　4. Note down the output readings for half/full adder and half/full subtractor sum/difference and the carry/borrow bit for different combination of input.
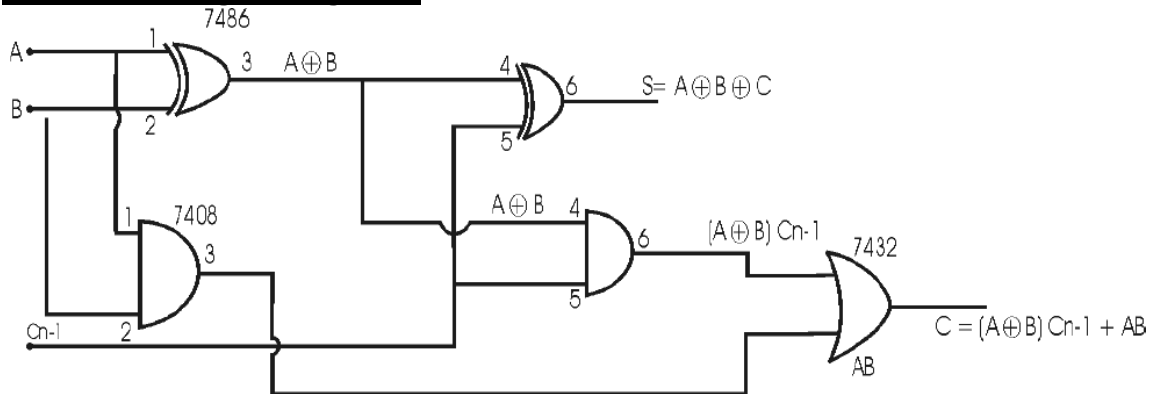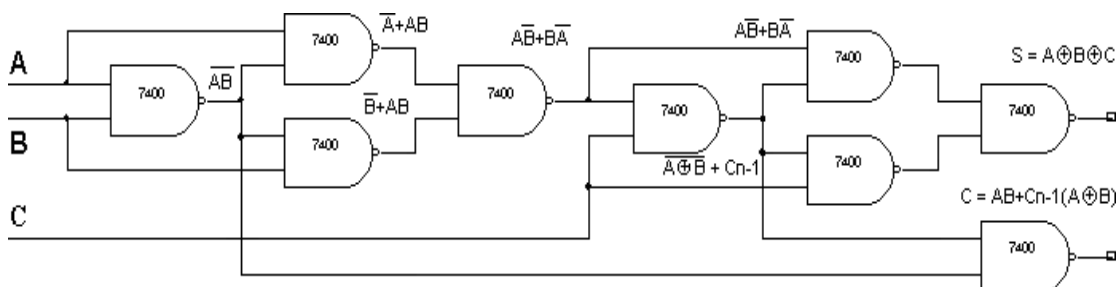
**Full Adder using basic gates:-**



**Full adder using NAND gates only:-**

## OBSERVATION TABLE:-

| | | Full Adder | | | | |
|---|---|---|---|---|---|---|
| A | B | Cn-1 | S | C | S(V) | C(V) |
| 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 | | |

**RESULT:-**The designing of Full adder is successfully done..

**PRECAUTIONS:-**1)The continuity of the connecting terminals should be checked before going .
2) It should be care that the values of the components of the circuit is does not
exceed  to their ratings (maximum value).
3) Before the circuit connection it should be check out working condition
of all the Component.

**OBJECTIVE:-**Design Half subtractor..

**EQUIPMENT REQUIRED:-** All the basic gate mention in the fig., IC 7486, IC 7432, IC 7408, IC 7400, etc.

**THEORY:-**In electronics, a **subtractor** can be designed using the same approach as that of an adder. The binary subtraction process is summarized below. As with an adder, in the general case of calculations on multi-bit numbers, three bits are involved in performing the subtraction for each bit of the difference: the minuend ($X_i$), subtrahend ($Y_i$), and a borrow in from the previous (less significant) bit order position ($B_i$). The outputs are the difference bit ($D_i$) and borrow bit $B_{i+1}$.

The half-subtractor is a combinational circuit which is used to perform subtraction of two bits. It has two inputs, X (minuend) and Y (subtrahend) and two outputs D (difference) and B (borrow).

**PROCEDURE:-**

    1. Verify the gates.

    2. Make the connections as per the circuit diagram.

    3. Switch on VCC and apply various combinations of input according to the truth table

    4. Note down the output readings for half/full adder and half/full subtractor sum/difference and the carry/borrow bit for different combination of input.

**Using X-OR and basic gates:-**
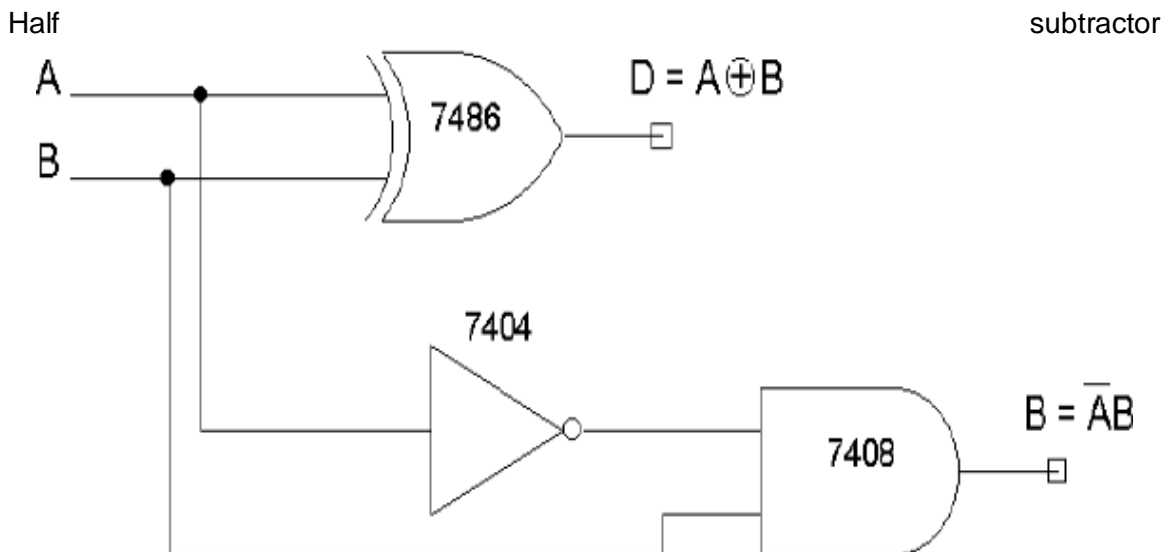
Half                                                                                                    subtractor



**Using only NAND gates:-**

## OBSERVATION TABLE:-

| Half Subtractor | | | | | |
|---|---|---|---|---|---|
| A | B | D | B | D(V) | B(V) |
| 0 | 0 | 0 | 0 | | |
| 0 | 1 | 1 | 1 | | |
| 1 | 0 | 1 | 0 | | |
| 1 | 1 | 0 | 0 | | |

**RESULT:-**The designing of Half substractor is successfully done..

**PRECAUTIONS:-**1)The continuity of the connecting terminals should be checked before going .
2) It should be care that the values of the components of the circuit is does not
exceed to their ratings (maximum value).
3) Before the circuit connection it should be check out working condition
of all theComponent.

**OBJECTIVE:**-Design full subtractor.

.

**EQUIPMENT REQUIRED:**- All the basic gate mention in the fig., IC 7486, IC 7432, IC 7408, IC 7400, etc.

**THEORY:-** In electronics, a **subtractor** can be designed using the same approach as that of an adder. The binary subtraction process is summarized below. As with an adder, in the general case of calculations on multi-bit numbers, three bits are involved in performing the subtraction for each bit of the difference: the minuend ($X_i$), subtrahend ($Y_i$), and a borrow in from the previous (less significant) bit order position ($B_i$). The outputs are the difference bit ($D_i$) and borrow bit $B_{i+1}$.

The Full_subtractor is a combinational circuit which is used to perform subtraction of three bits .It has three inputs, X(minuend) and Y(subtrahend) and Z (subtrahend) and two outputs D (difference) and B (borrow).

Easy way to write truth table

**D=X-Y-Z** (don't bother about sign)

**B = 1 If X< (Y+Z)**

**PROCEDURE:-**

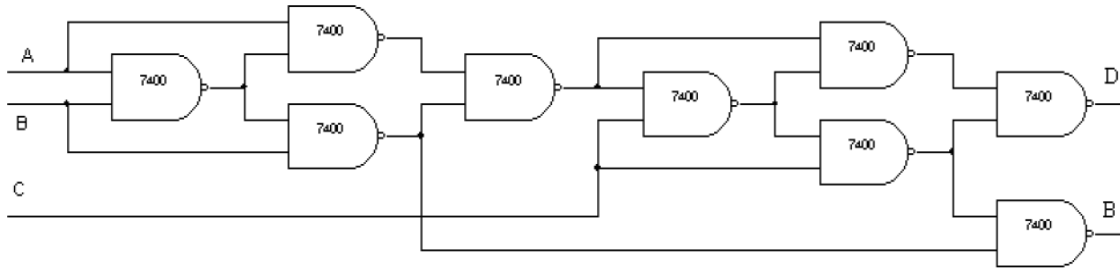      1. Verify the gates.

      2. Make the connections as per the circuit diagram.

      3. Switch on VCC and apply various combinations of input according to the truth table

      4. Note down the output readings for half/full adder and half/full subtractor sum/difference and the carry/borrow bit for different combination of input.

**Using X-OR and basic gates:-**
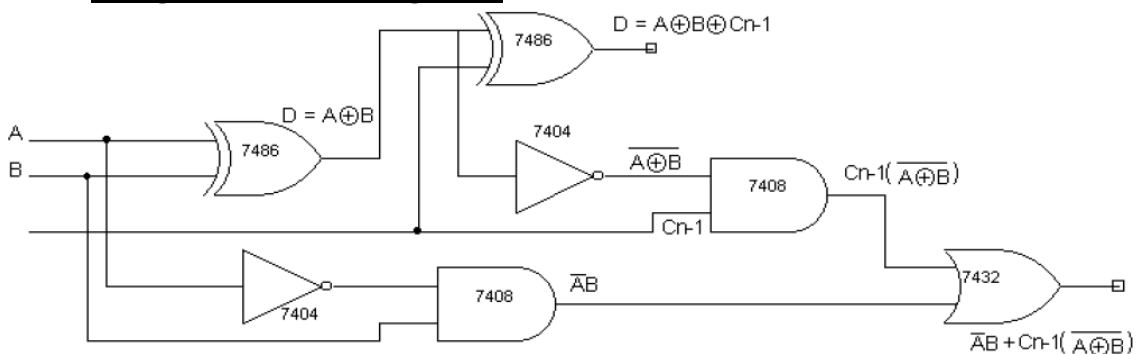


**Using only NAND gates:-**

**OBSERVATION  TABLE:-**

| Full Subtractor | | | | | | |
|---|---|---|---|---|---|---|
| A | B | Cn-1 | D | B | D(v) | B(v) |
| 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 | | |

**RESULT:-**The designing of Full substractor is successfully done..

**PRECAUTIONS:-**1)The continuity of the connecting terminals should be checked before going .
2) It should be care that the values of the components of the circuit is does not
exceed to their ratings (maximum value).
3) Before the circuit connection it should be check out working condition
of all theComponent.

**OBJECTIVE:**-Verify the operation of magnitude comparator (7485IC)

**Apparatus Required:-**IC7486,IC7404,IC7408,etc

**Theory:-** in electronics, a **comparator** is a device that compares two voltages or currents and switches its output to indicate which is larger. It is used in Analog-to-digital converter (ADCs).

**Procedure:-**
1. Verify the gates.

2. Make the connections as per the circuit diagram.

3. Switch on V cc .

4. Applying I/p and Check for the outputs.

5. The voltmeter readings of outputs are taken and tabulated in tabularcolumn.

6. The o/p is verified.

**One bit comparator:-**
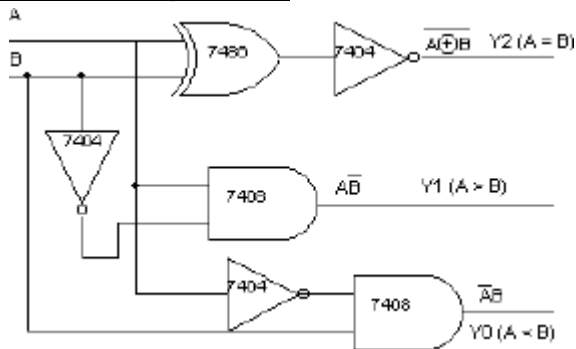


| A | B | Y1 (A>B) | Y2 (A = B) | Y3 (A < B) |
|---|---|----------|------------|------------|
| 0 | 0 | 0 | 1 | 0 |
| 0 | 1 | 0 | 0 | 1 |
| 1 | 0 | 1 | 0 | 0 |
| 1 | 1 | 0 | 1 | 0 |

**Two bit comparator:-**

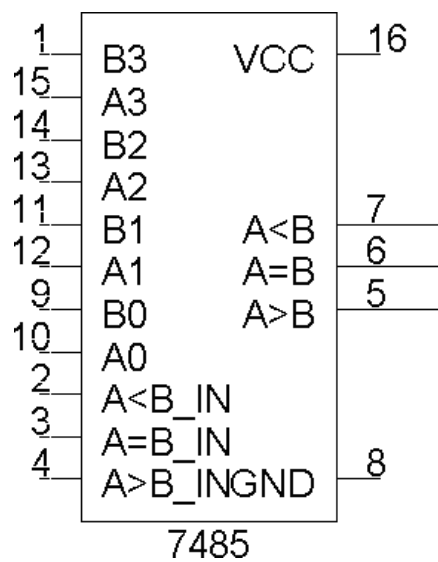| A1 | A0 | B1 | B0 | Y1 (A > B) | Y2 (A = B) | Y3 (A < B) |
|----|----|----|----|------------|------------|------------|
| 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| 0 | 0 | 0 | 1 | 0 | 0 | 1 |
| 0 | 0 | 1 | 0 | 0 | 0 | 1 |
| 0 | 0 | 1 | 1 | 0 | 0 | 1 |
| 0 | 1 | 0 | 0 | 1 | 0 | 0 |
| 0 | 1 | 0 | 1 | 0 | 1 | 0 |
| 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 |



7485

**4-bit comparator:-**



7485
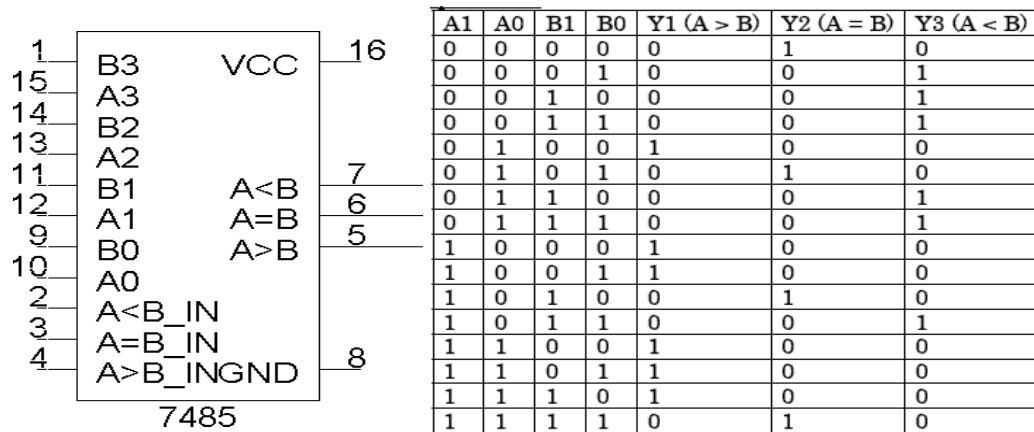
**RESULT:-**We successfully verify the comparator.

**PRECAUTIONS:-**1)The continuity of the connecting terminals should be checked before going .
   2) It should be care that the values of the components of the circuit is does not
   exceed  to their ratings (maximum value).
  3) Before the circuit connection it should be check out working condition
   of all theComponent.

**OBJECTIVE:**-Verify the Truth Table of  RS Flip-flop ,JK F/F, D F/F & T type F/F.

**Apparatus Required:-**IC7410,IC7400,etc.

**Theory:-**In electronics, a **flip-flop** is a circuit that has two stable states and can be used to store state information. The circuit can be made to change state by signals applied to one or more control inputs and will have one or two outputs. A circuit incorporating flip-flops has the attribute of *state*; its output depends not only on its current input, but also on its previous inputs. Such a circuit is described as sequential logic. Where a single input is provided, the circuit changes state every time a pulse appears on the input signal. Since the flip-flop retains the state after the signal pulses are removed, one type of flip-flop circuit is also called a "latch". Other types of flip-flops may have inputs that set a particular state, set the opposite state, or change states, depending on which input is pulsed. Flip-flops are used as data storage elements, for counting of pulses ,and for synchronizing randomly-timed input signals to some reference timing signal.Flip-flops are a fundamental building block of digital electronics systems used in computers, communications, and many other types of systems.

**CIRCUIT  DIAGRAM:-**
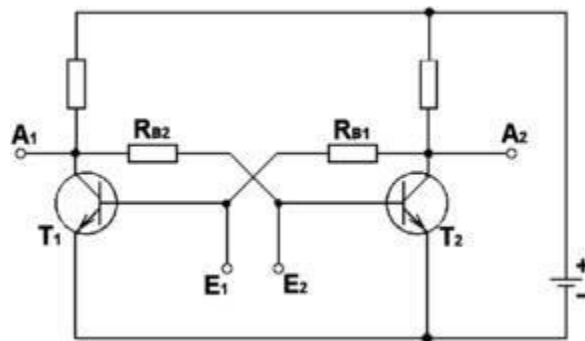


**Flip-flop types:-**
Flip-flops can be divided into common types: the **RS** ("set-reset"), **D** ("data" or "delay"), **T** ("toggle"), and **JK** types are the common ones. The behavior of a particular type can be described by what is termed the characteristic equation ,which derives the "next"(i.e. ,after the next clock pulse) output, $Q_{next}$, in terms of the input signal(s) and/or the current output, $Q$.

**Simple set –reset latches:-**
When using static gates as building blocks, the most fundamental latch is the simple *SR latch*, where S and R stand for *set* and *reset*. It can be constructed from a pair of cross-coupled NOR logic gates. The stored bit is present on the output marked Q. While the S and R inputs are both low, feedback maintains the Q and Q outputs in a constant state , with Q the complement of Q . If S (*Set*) is pulsed high while R (*Reset*) is held low, then the Q output is forced high, and stays high when Sreturns to low; similarly, if R is pulsed high while S is held low, then the Q output is forced low,and stays low when R returns to low.
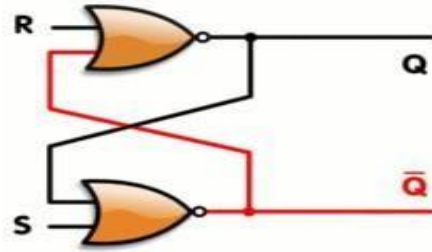
**TRUTH TABLE:-**

| SR latch operation |
| --- |

| S | R | Action |
|---|---|---|
| 0 | 0 | No Change |
| 0 | 1 | Q =0 |
| 1 | 0 | Q =1 |
| 1 | 1 | Restricted combination |

**SR NAND latch:-**This is an alternate model of the simple SR latch built with NAND (not AND) logic gates. *Set* and *reset* now become active low signals, denoted S and R respectively. Otherwise, operation is identical to that of the SR latch. Historically, SR-latches have been pre dominant despite the notational in convenience of active - low inputs .This is because NAND gatesare cheaper to produce than NOR gates in the diode-transistor logic (DTL), transistor- transistor logic (TTL) families, and complementary metal–oxide semiconductor (CMOS) logic families.
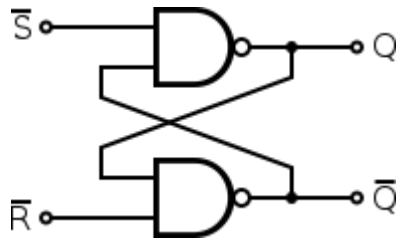


**Gate SR latch:-**
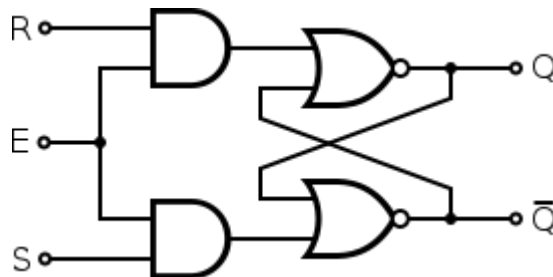


**D flip-flop:-**
The D flip-flop is the most common flip-flop in use today. It is better known as *data* or *delay* flip-flop(as its output Q looks like a delay of input D).The Q output takes on the state of the D input at the moment of a positive edge at the clock pin (or negative edge if the clock input is active low).[23]It is called the D flip-flop for this reason , since the output takes the value of the D

input or *data* input, and *delays* it by one clock cycle. The D flip-flop can be interpreted as a primitive memory cell, zero-order hold, or delay line. Whenever the clock pulses , the value of $Q_{next}$ is D and $Q_{p\,rev}$ otherwise.
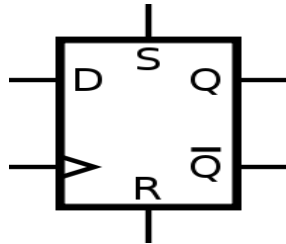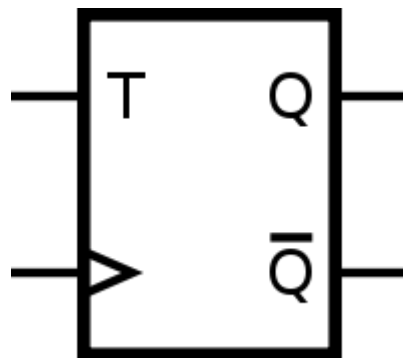


**Truth Table:-**

| | Inputs | | | Outputs | |
|---|---|---|---|---|---|
| S | R | D | > | Q | Q' |
| 0 | 1 | X | X | 0 | 1 |
| 1 | 0 | X | X | 1 | 0 |
| 1 | 1 | X | X | 1 | 1 |

## T flip-flop:-

If the T input is high ,the T flip-flop changes state ("toggles") whenever the clock input is strobed . If the T input is low, the flip-flop holds the previous value. This behavior is described by the characteristic equation:

$$Q_{next} = T \oplus Q = T\overline{Q} + \overline{T}Q$$ (expanding the XOR operator)



**T flip-flop operation**

| Characteristic table | | | | Excitation table | | | |
|---|---|---|---|---|---|---|---|
| *T* | *Q* | *Q_next* | Comment | *Q* | *Q_next* | *T* | Comment |

25

| 0 | 0 | 0 | **Hold state(no clk)** | 0 | 0 | **0 No change** |
|---|---|---|---|---|---|---|
| **0** | **1** | **1** | **Hold state(no clk)** | **1** | **1** | **0 No change** |
| **1** | **0** | **1** | **toggle** | **0** | **1** | **1 Complement** |
| **1** | **1** | **0** | **toggle** | **1** | **0** | **1Complement** |

## JK flip-flop:-

The JK flip-flop augments the behavior of the SR flip-flop (J=Set, K=Reset) by interpreting the S = R = 1 condition as a "flip" or toggle command. Specifically, the combination J=1,K= 0is a command to set the flip-flop ; the combination J =0,K= 1is a command to reset the flip-flop; and the combination J = K = 1 is a command to toggle the flip-flop, i.e., change its output to the logical complement of its current value. Setting J = K = 0 does NOT result in a D flip-flop, but rather, will hold the current state. To synthesize a D flip-flop, simply set K equal to the complement of J. The JK flip-flop is therefore a universal flip-flop, because it can be configured to work as an SR flip-flop, a D flip-flop, or a T flip-flop .NOTE: The flip-flop is positive-edge triggered (rising clock pulse) as seen in the timing diagram .The characteristic equation of the JK flip-flop is:

$$Q_{next} = J\overline{Q} + \overline{K}Q$$
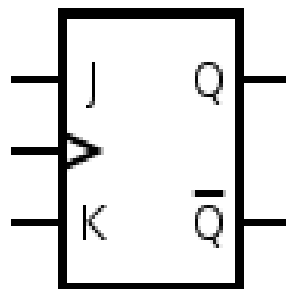


Characteristic table

| J | K | $Q_{next}$ | Comment |
|---|---|---|---|
| 0 | 0 | Q | hold state |
| 0 | 1 | 0 | reset |
| 1 | 0 | 1 | set |

Excitation table

| Q | $Q_{next}$ | J | K | Comment |
|---|---|---|---|---|
| 0 | 0 | 0 | X | No change |
| 0 | 1 | 1 | X | Set |
| 1 | 0 | X | 1 | Reset |

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 1 | Q' | toggle | | 1 | 1 | X | 0 | No change |

**Procedure:-**

1.  Connections are made as per circuit diagram.

2.  The truth table is verified for various combinations of inputs

**RESULT:-**We successfully verify the different flip-flops.

**PRECAUTIONS:-**1)The continuity of the connecting terminals should be checked before going .
   2) It should be care that the values of the components of the circuit is does not
   exceed to their ratings (maximum value).
   3) Before the circuit connection it should be check out working condition
   of all the Component.

# EXPERIMENT No- 11

**OBJECTIVE:**-Design ¾ bit Counter & verify truth table

**Apparatus Required:-**IC7408,IC7476,IC7490,IC74192,IC74193,IC7400,IC7416,IC 7432etc.

**Procedure:-**

1. Connections are made as per circuit diagram.

2. Clock pulses are applied one by one at the clock I/P and the O/P is observed at QA, QB & QC for IC 7476.

3. Truth table is verified.

**Procedure (IC74192,IC 74193):-**

1. Connections are made as per the circuit diagram except the connection from output of NAND gate to the load input

2. The data (0011) = 3 is made available at the data I/p A, B, C & Drespectively.

3. The load pin made low so that the data 0011 appears at QD, QC, QB & QA respectively.

4. Now connect the output of the NAND gate to the load input

5. Clock pulses are applied to "count up" pin and the truth table is verified

6. Now apply (1100)=12 for 12 to 5 counter and remaining is same as for 3 to 8 counter.

7. The pin diagram of IC 74192 is same as that of 74193. 74192 can be configured to count between 0 and 9 in either direction. The starting value can be any number between 0 and 9
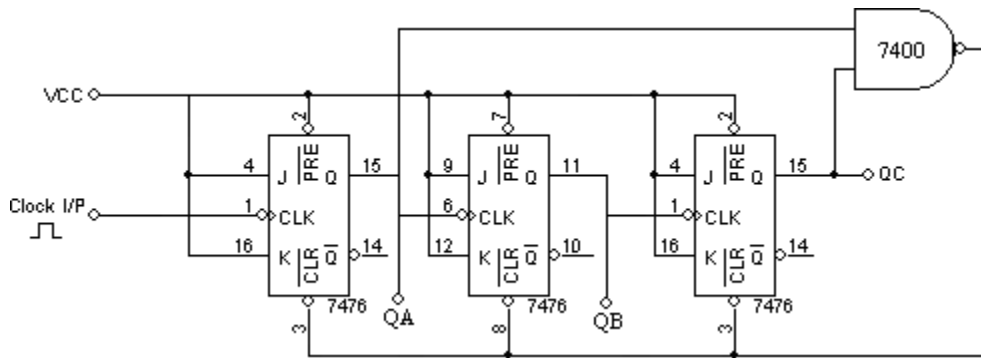
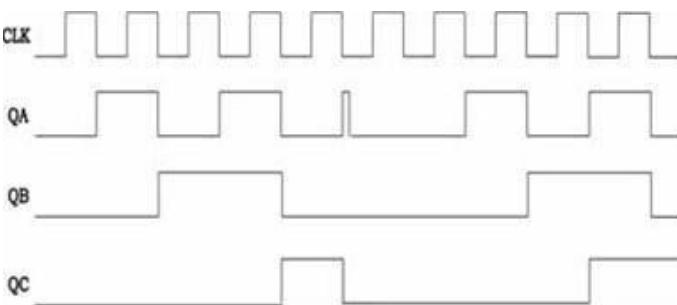**Circuit Diagram:-3-Bit Asynchronous Down Counter:-**
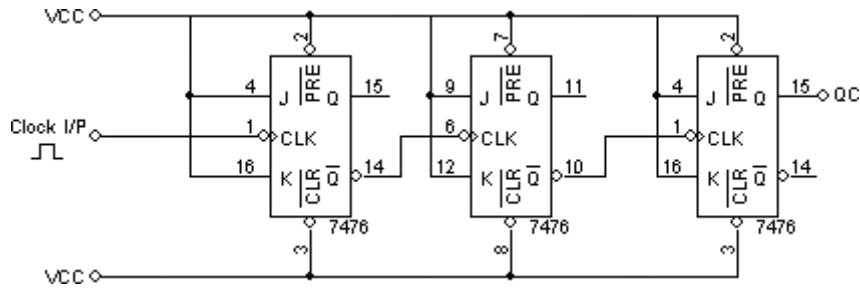
| 3-bit Asynchronous down counter | | | |
|---|---|---|---|
| Clock | QC | QB | QA |
| 0 | 1 | 1 | 1 |
| 1 | 1 | 1 | 0 |
| 2 | 1 | 0 | 1 |
| 3 | 1 | 0 | 0 |
| 4 | 0 | 1 | 1 |
| 5 | 0 | 1 | 0 |
| 6 | 0 | 0 | 1 |
| 7 | 0 | 0 | 0 |
| 8 | 1 | 1 | 1 |
| 9 | 1 | 1 | 0 |
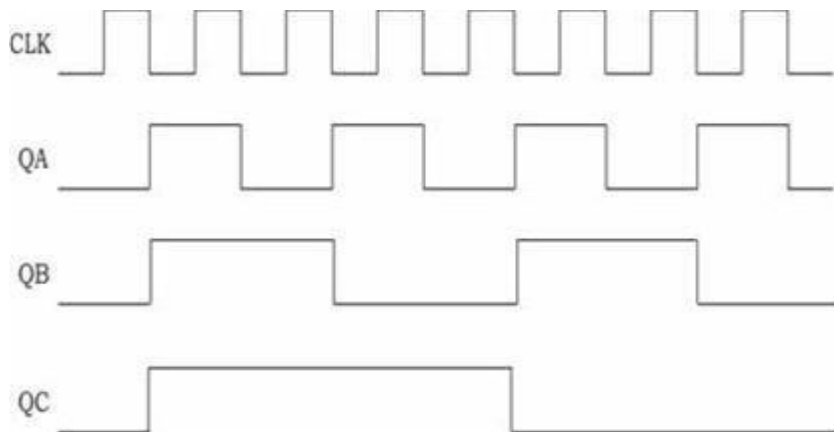
**Mod 5 Asynchronous Counter:-**



| Mod 5 Asynchronous counter | | | |
|---|---|---|---|
| Clock | QC | QB | QA |
| 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 1 |
| 2 | 0 | 1 | 0 |
| 3 | 0 | 1 | 1 |
| 4 | 1 | 0 | 0 |
| 5 | 0 | 0 | 0 |

**Mod 3 Asynchronous Counter:-**

29
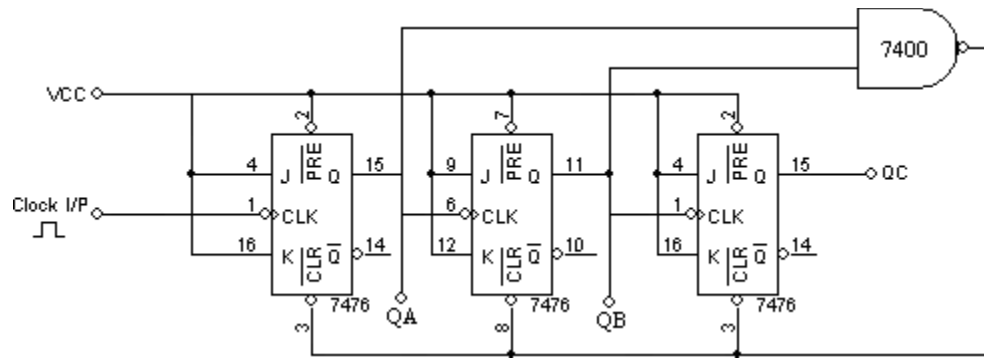
| Mod 3 Asynchronous counter | | | |
|---|---|---|---|
| Clock | QC | QB | QA |
| 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 1 |
| 2 | 0 | 1 | 0 |
| 3 | 0 | 0 | 0 |
| 4 | 0 | 0 | 1 |
| 5 | 0 | 1 | 0 |
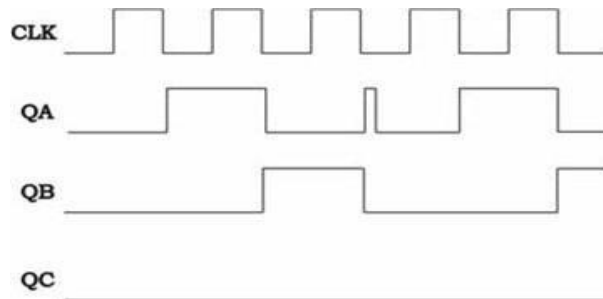


### 3-bit Synchronous Counter:-
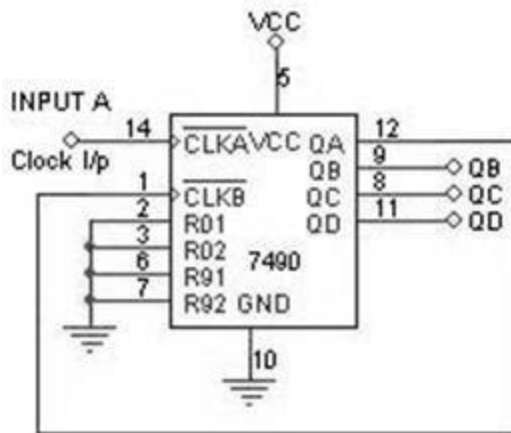


| Clock | QD | QC | QB | QA |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 | 1 |
| 2 | 0 | 0 | 1 | 0 |
| 3 | 0 | 0 | 1 | 1 |
| 4 | 0 | 1 | 0 | 0 |
| 5 | 0 | 1 | 0 | 1 |
| 6 | 0 | 1 | 1 | 0 |
| 7 | 0 | 1 | 1 | 1 |
| 8 | 1 | 0 | 0 | 0 |
| 9 | 1 | 0 | 0 | 1 |
| 10 | 0 | 0 | 0 | 0 |

**RESULT:-**We successfully verify the designing of counters.

**PRECAUTIONS:-**1)The continuity of the connecting terminals should be checked before going .
2) It should be care that the values of the components of the circuit is does not exceed to their ratings (maximum value).
3) Before the circuit connection it should be check out working condition of all theComponent.

## EXPERIMENT No- 12

**OBJECTIVE:**-Design shift Register & verify truth table .used as a serial/parallel shift .Resistor.

**Apparatus Required:-**IC7495,etc.

**Procedure:-**

### Serial In Parallel Out:-

1. Connections are made as per circuit diagram.

2. Apply the data at serial I/p

3. Apply one clock pulse at clock 1(Right Shift) observe this data at QA.

4. Apply the next data at serial I/p.

5. Apply one clock pulse at clock 2, observe that the data on QA will shift to QB and thenew data applied will appear at QA.

6. Repeat steps 2 and 3 till all the 4 bits data are entered one by one into the shift register.
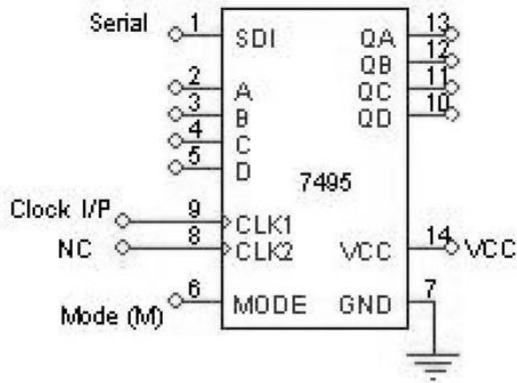
Pin diagram:                    Truth Table:-
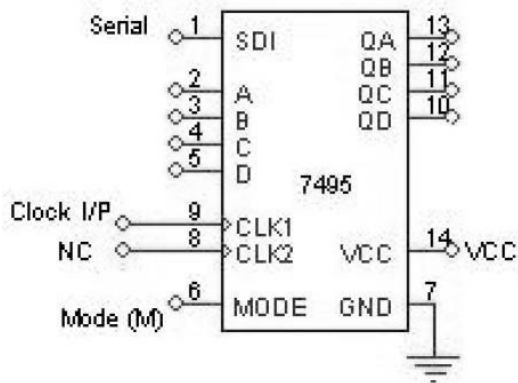


| Clock | Serial i/p | QA | QB | QC | QD |
|-------|-----------|----|----|----|----|
| 1 | 0 | 0 | X | X | X |
| 2 | 1 | 1 | 0 | X | X |
| 3 | 1 | 1 | 1 | 0 | X |
| 4 | 1 | 1 | 1 | 1 | 0 |

### Serial In Serial Out:-

1. Connections are made as per circuit diagram.

2. Load the shift register with 4 bits of data one by one serially.

3. At the end of 4$_{th}$ clock pulse the first data „ d0 "appears at QD.

4. Apply another clock pulse ;the second data„ d1"appears at QD.

5. Apply another clock pulse ; the third data appears at QD.

6. Application of next clock pulse will enable the 4th data „d3" to appear at QD .Thus the data applied serially at the input comes out serially at QD

Pin Diagram:                              Truth Table:-



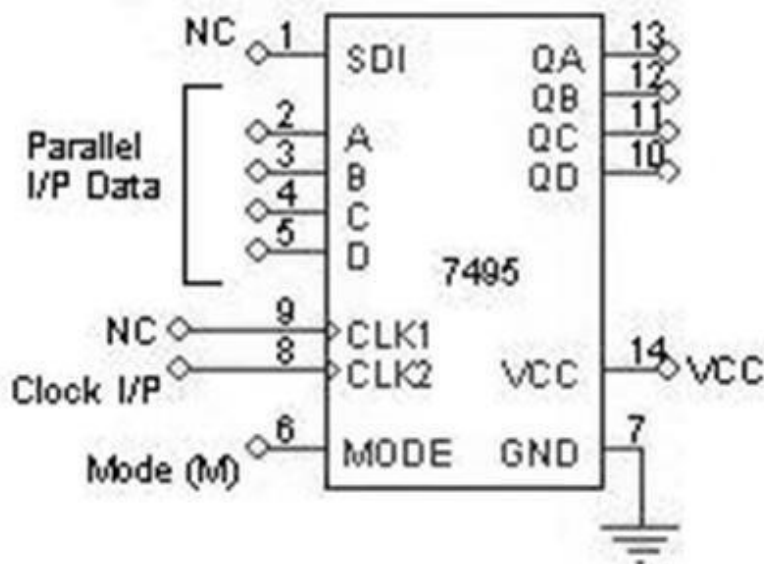| Clock | Serial i/p | QA | QB | QC | QD |
|-------|-----------|----|----|----|----|
| 1 | do=0 | 0 | X | X | X |
| 2 | d1=1 | 1 | 0 | X | X |
| 3 | d2=1 | 1 | 1 | 0 | X |
| 4 | d3=1 | 1 | 1 | 1 | 0=do |
| 5 | X | X | 1 | 1 | 1=d1 |
| 6 | X | X | X | 1 | 1=d2 |
| 7 | X | X | X | X | 1=d3 |

## Parallel In Parallel Out:-

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 QA, Q B, QC and QD respectively.

**Truth Table:-**

| Clock | Parallel i/p | | | | Parallel o/p | | | |
|---|---|---|---|---|---|---|---|---|
| | A | B | C | D | QA | QB | QC | QD |
| 1 | 1 | 0 | 1 | 1 | 1 | 0 | 1 | 1 |

**Parallel In Serial Out:-**

1. Connections are made as per circuit diagram.

2. Apply the desired4bit data at A,B,C and D.

3. Keeping the mode control M=1apply one clock pulse .The data applied at A,B, C andD will appear at QA, QB, QC and QD respectively.

4. Now mode control M=0.Apply clock pulses one by one and observe the data coming out serially at QD.
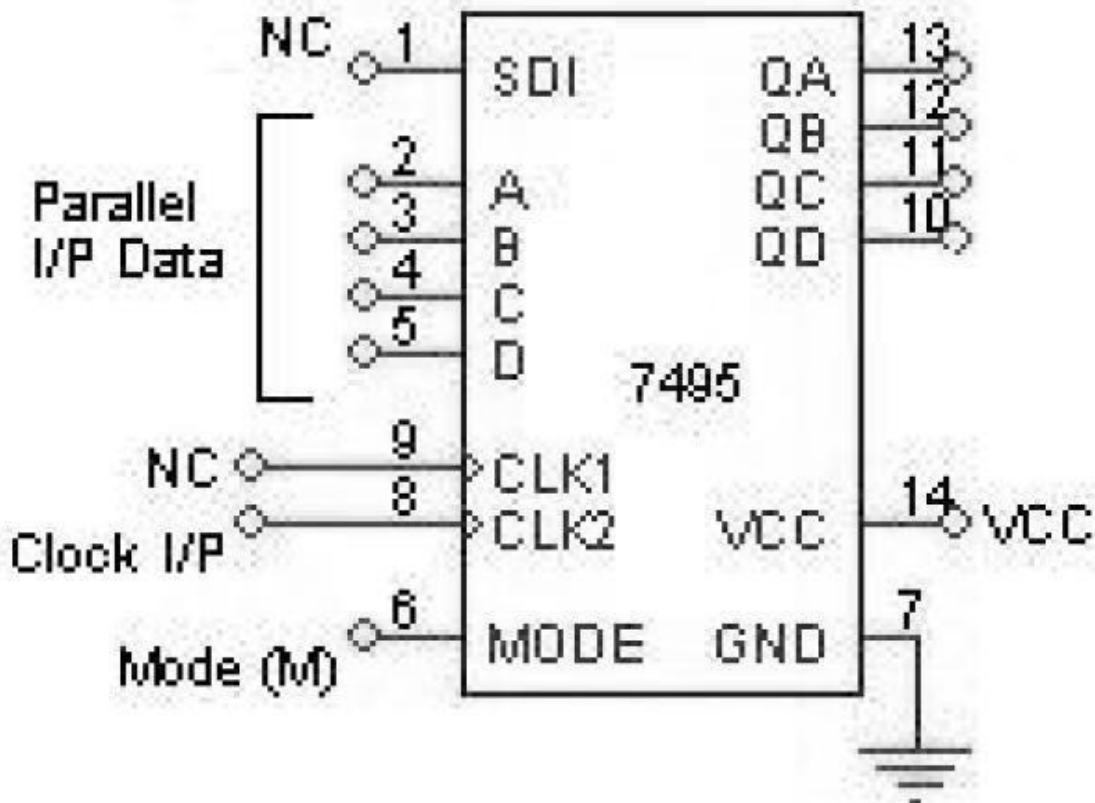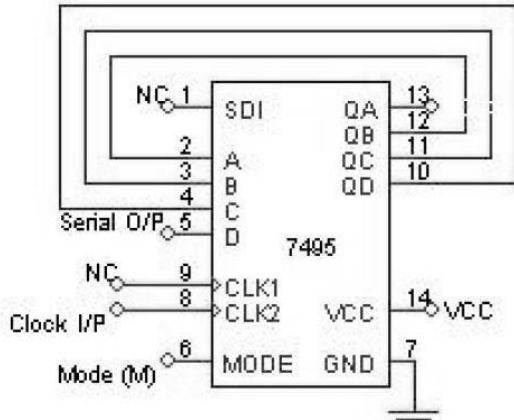
**Truth Table:-**

| Mode | Clock | Parallel i/p | | | | Parallel o/p | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | A | B | C | D | QA | QB | QC | QD |
| 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 |

**Left Shift:-**

1. Connections are made as per circuit diagram

2. Apply the first data at D and apply one clock pulse .This data appears at QD.

3. Now the second data is made available at D and one clock pulse applied .The dataappears at QD to QC and the new data appears at QD.

4. Step 3 is repeated until all the 4bits are entered one by one At
the end 4th clock pulse the 4 bits are available at QA, QB, QC and
QD.

Pin diagram:-                                    Truth Table:-



| Clock | Serial i/p | QA | QB | QC | QD |
|-------|-----------|----|----|----|----|
| 1 | 1 | X | X | X | 1 |
| 2 | 0 | X | X | 1 | 0 |
| 3 | 1 | X | 1 | 0 | 1 |
| 4 | 1 | 1 | 0 | 1 | 1 |

**RESULT:-**We successfully verify the designing of shift resistors.

**PRECAUTIONS:-**1)The continuity of the connecting terminals should be checked before going .
2) It should be care that the values of the components of the circuit is does not
exceed  to their ratings (maximum value).
3) Before the circuit connection it should be check out working condition
of all the Component.