**Lingaya’s Vidyapeeth, Faridabad**

(Deemed to be University under Section 3 of UGC Act, 1956)

**Digital Electronics Lab**

**EC-253**

**LAB File**

**B.TECH 2nd  Year [C.S.E]**



Nachauli, Old Faridabad-Jasana Road,

Faridabad 121002,  Phone: (129) 2201008, 2201009;

WebSite: www.lingayasuniversity.edu.in

Submitted to:SubmittedBy**:**

Monika Mishra NAME –S Kashaf Alam

                                                                                   ROLL No – 22CS75CL

                                                                                      SEMESTER – 3rd

**INDEX**

|  |  |
| --- | --- |
| **S.NO** | **EXPERIMEMT** |
| 1. | Study of TTL gates – AND; OR; NOT;  NAND; NOR; EX-OR; EX-NOR. |
| 2. | Implementation of 4- bit full adder, using 7483 IC |
| 3 | To verify the operation of comparator (1- bit and 2- bit ) |
| 4 | (a)To verify the operation of multiplexer |
|  | (b)to verify the operation of demultiplexer |
| 5 | Design and realize a given function using K- maps and verify its performance |
| 6 | To verify the truth tables of S-R; J-K; T and D type flip flops. |
| 7 | (a) to design and verify 4- bit synchronous |
|  | counter. |
|  | (b) to design and verify 4- bit asynchronous counter. |
| 8 | To analyse the circuit and truth table of 4- bit SIPO shift register. |
| 9 | To study and verify table of half and full subtractor |
| 10 | Verify binary to gray and gray to binary conversion using NAND gates only. |

**EXPERIMENT - 1**

**Aim:-**Study of TTL gates – AND; OR; NOT; NAND; NOR; EX-OR; EX-NOR

. **Apparatus Required:-**

Digital lab kit, single strand wires, breadboard, TTL IC’s

|  |  |
| --- | --- |
| **Gates** | **IC NO.** |
| AND | 7408 |
| OR | 7432 |
| NAND | 7400 |
| NOR | 7402 |
| NOT | 7404 |
| XOR | 74136 |

Theory:-

Logic gates are idealized or physical devices implementing a Boolean function, which it performs a logical operation on one or more logical inputs and produce a single output. Depending on the context, the term may refer to an ideal logic gate, one that has for instance zero rise time and unlimited fan out or it may refer to anon-ideal physical device.

**The main hierarchy is as follows:-**

1. Basic Gates
2. Universal Gates
3. Advanced Gates

Procedure:-

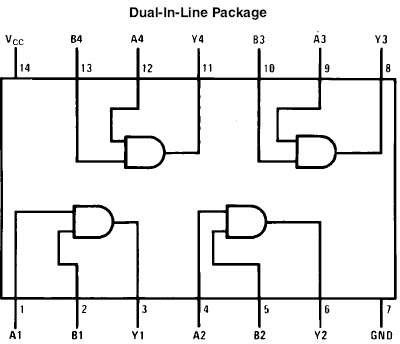
* + Place the breadboard gently on the observationtable.
  + Fix the IC which is under observation between the half shadow line of breadboard, so there is no shortage ofvoltage.
  + Connect the wire to the main voltage source (Vcc) whose other end is connected to last pin of the IC (14 place from the notch).
  + Connect the ground of IC (7th place from the notch) to the ground terminal provided on the digital labkit.
  + Give the input at any one of the gate of the ICs i.e. 1st, 2nd, 3rd, 4th gate by using connecting wires.(In accordance to IC provided).
  + Connect output pins to the led on digital labkit.
  + Switch on the power supply.

If led glows red then output is true, if it glows green output is false, which is numerically denoted as 1 and 0 respectively. The Color can change based on the IC manufacturer it’s just verification of the Truth Table not the color change.

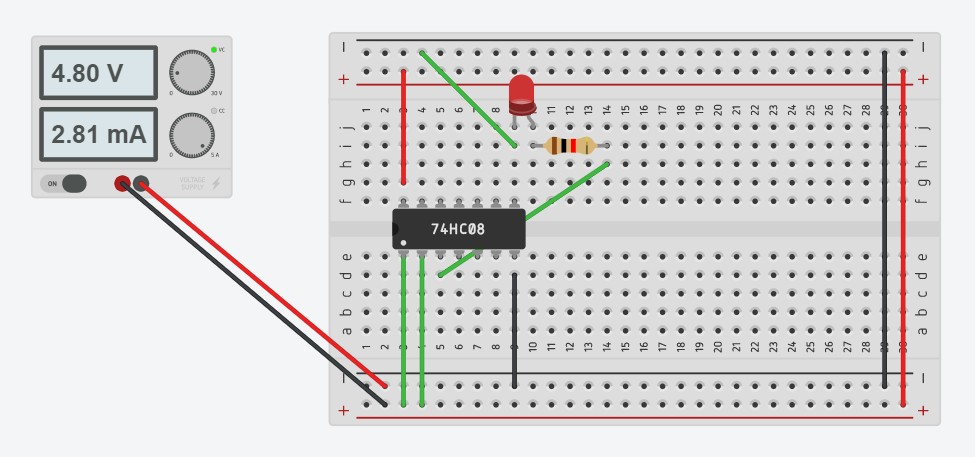
Basic Gates

1. **AND gate: -** Function of AND gate is to give the output true when both the inputs are true. In all the other remaining cases output becomes false. Following table justifies the statement:-

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

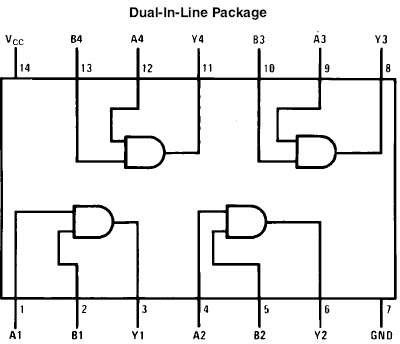


IC 7408

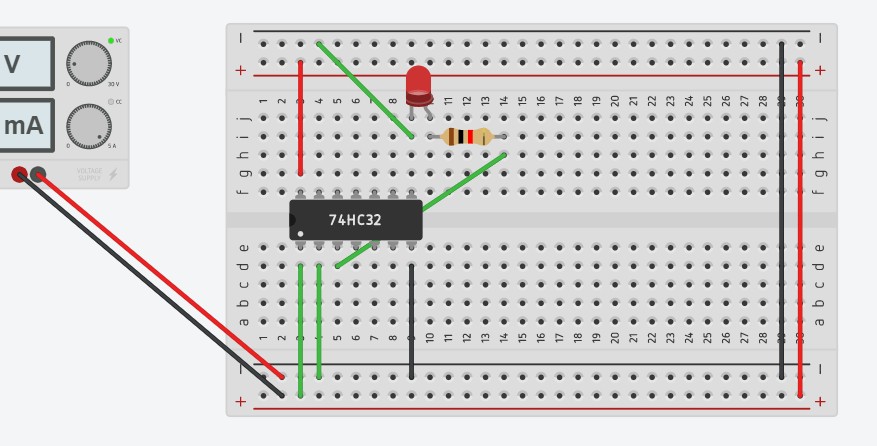


1. **OR gate: -** Function of OR gate is to give output true when one of the either inputs are true .In the remaining case output becomes false. Following table justify the statement:-

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

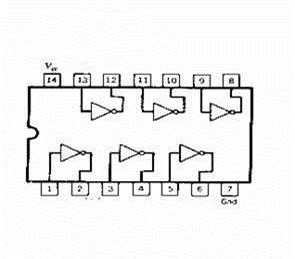


IC 7432



1. **NOT gate: -**Function of NOR gate is to reverse the nature of the input .It converts true input to false and vice versa. Following table justifies the statement :-

|  |  |
| --- | --- |
| Input | Output |
| 1 | 0 |
| 0 | 1 |

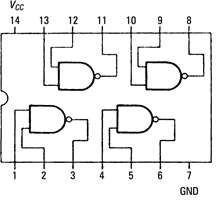


IC 7404

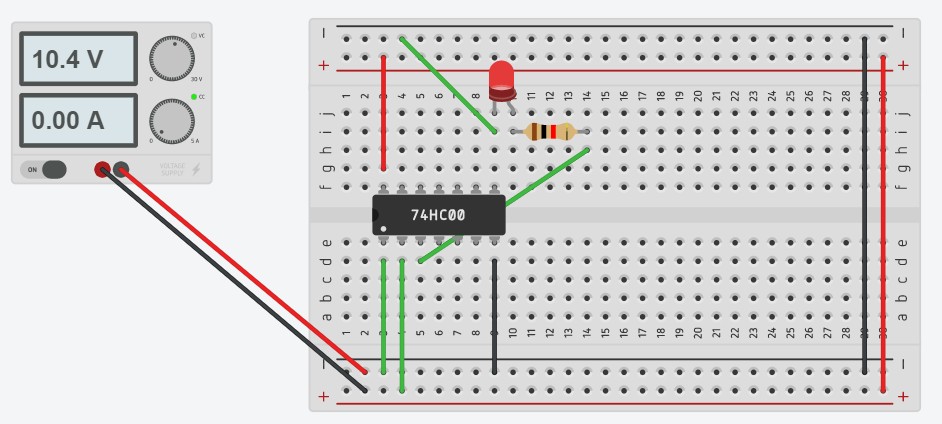
Universal Gate: -

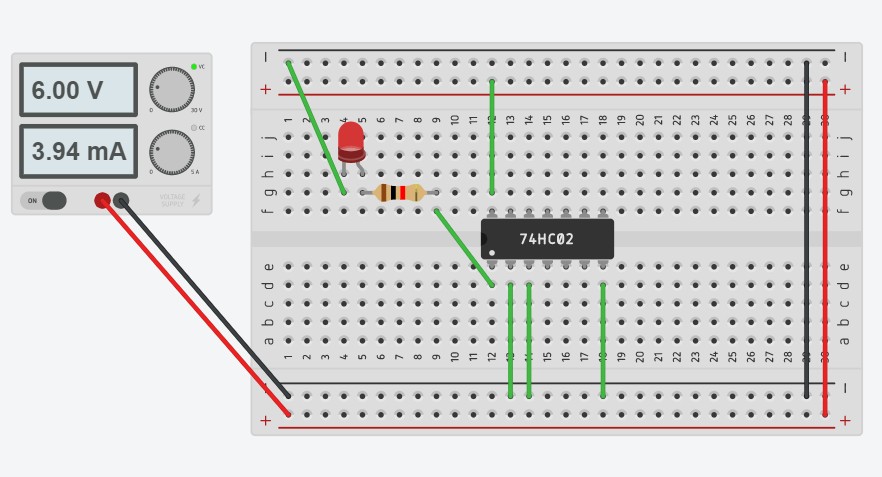
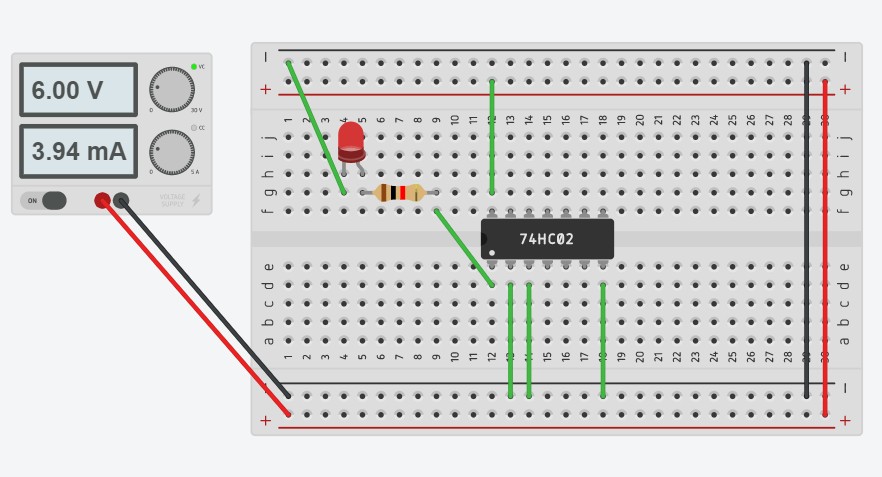
1. **NAND gate: -**Function of NAND gate is to give true output when one of the two provided input are false. In the remaining output is true case .Following table justifies the statement :-

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



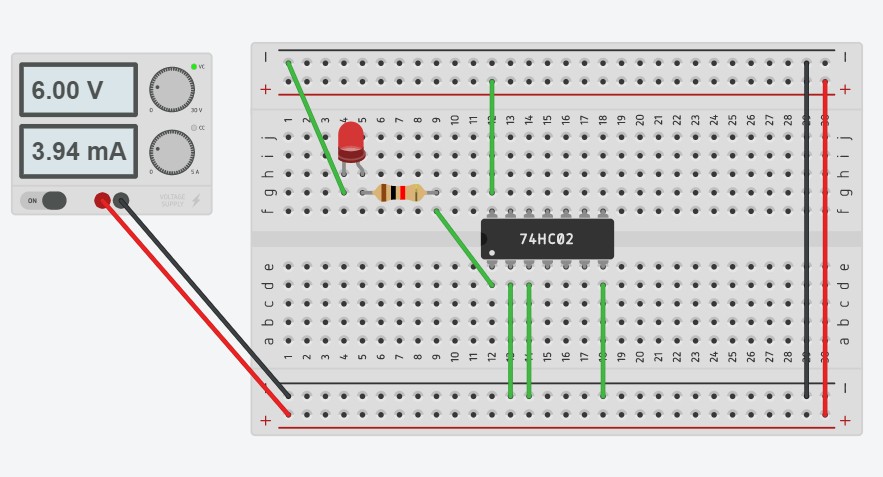
IC 7400





**NOR gate: -** NOR gate gives the output true when both the two provided input are false. In all the other cases output remains false. Following table justifies the statement :-

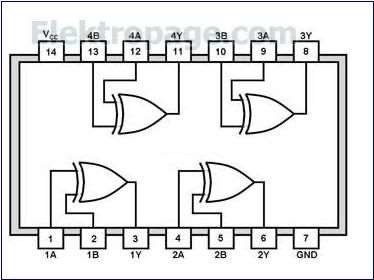
|  |  |  |
| --- | --- | --- |
| **Input A** | **Input B** | **Output** |
| 1 | 1 | 0 |
| 1 | 0 | 0 |
| 0 | 1 | 0 |
| 0 | 0 | 1 |



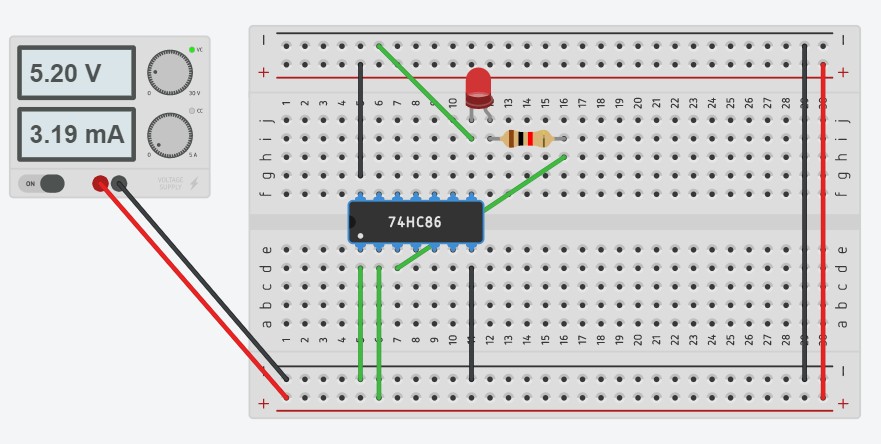
**Advanced Gates**

**XOR gate: -** The function of XOR gate is to give output true only when both the inputs are true. Following table explain this:-

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



IC 7413



Results :

*All gates are verified. Observed output matches theoretical concepts.*

Precautions:

* All connections should be made neat andtight.
* Digital lab kits and ICs should be handled with utmostcare.
* While making connections main voltage should be kept switchedoff.
* Never touch live and naked wires.

Experiment -2

Aim:**-**Implementation of 4-bit full adder, using 7483 IC.

Apparatus Required:-

**1.digital trainer kit**

**2. IC 7483**

**3. Connecting Wires**

Theory:-

**Adder**: -An adder is a logic circuit which adds two or three bits at a time and give sum and carry as the result.

Parallel Adder:-

A n-bit parallel adder can be constructed using number of full adders circuit connected in parallel the carry output of each is connected to the carry input of the next higher-order adder. Since all the bits of the augends and addend are fed into the adder circuits simultaneously and the additions in each position are known as parallel adder.

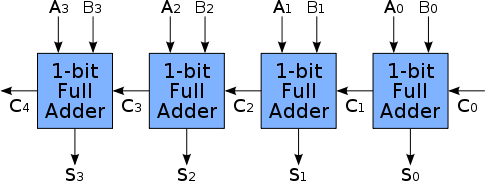
A3 A2 A1 A0 →

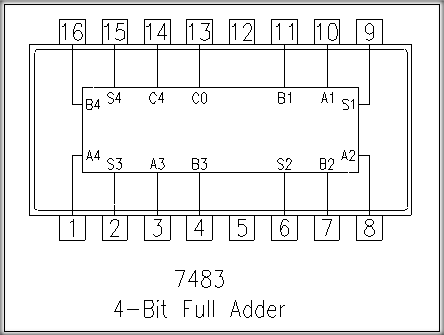
Augends bits B3 B2 B1 B0 →

Addend bits S3 S2 S1 S0 →

Sum bits

LOGIC DIAGRAM OF BCD ADDER





Procedure:

**1.** Connect ground and Vcc to 7483 IC from trainer kit through patch cords.

1. Connect inputs A0, A1, A2, A3 and B0, B1, B2, B3 to logic input switches.
2. Connect carry in from pin no.13 to ground so that carry input (CY1) will be logic‘0’state.
3. Connect S0, S1, S2, S3 and carry out (CY0) from pin nos. 9, 6, 2, 15 and 14 to the outputdisplay.5.Verify truth tables for different combinations ofinputs.

**TRUTH TABLE:-**

The Truth table operation of the 4-bit Adder is shown below:

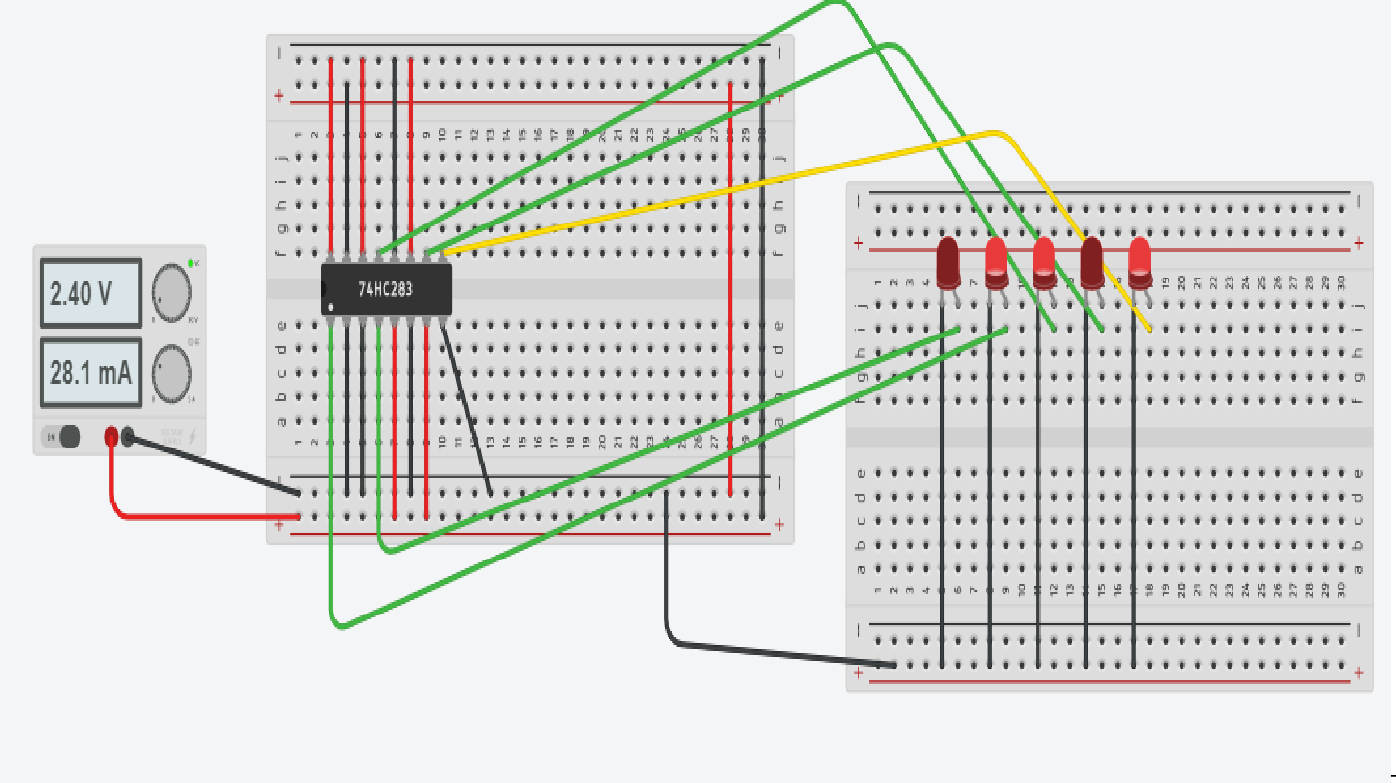
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| INPUTS | | | | | | | | | OUTPUTS | | | | |
| A0 | A1 | A2 | A3 | B0 | B1 | B2 | B3 | CY1 | S0 | S1 | S2 | S3 | CY0 |
| 1 | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 1 |

Result :-

For various combinations of selected input lines, observed the LED output and verified the truth table.

**Precautions:-**

* 1. All ICs should be checked before starting theexperiment.
  2. All the connection should betight.
  3. Always connect ground first and then connect Vcc.
  4. Suitable type wire should be used for different types ofcircuit.
  5. The kit should be off before change theconnections.
  6. After completed the experiments switch off the supply of theapparatus.

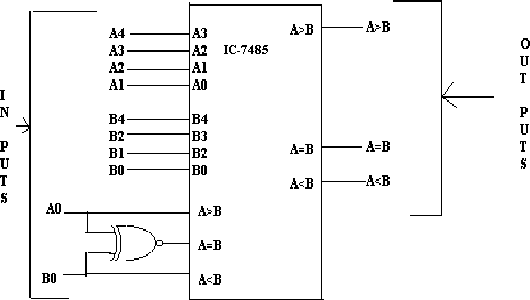


EXPERIMENT 3

**AIM:-** To verify the operation of Comparator.

**APPARATUS**:-IC 7485, Connecting wires, Ex-NOR gate, Digital trainer kit

CIRCUIT DIAGRAM:-



**THEORY**:-

Four bit comparators are available in msi (7485) which can compare straight binary and natural bcd codes . These ic’s can be cascaded to compare words of greater lengths of without external gates . The a > b , a = b and a< b outputs of a stage handling less significant bit are connected to the corresponding a > b, a < b and a = b cascading inputs of the the next stage handling more significant bits.Thestagehandlingtheleastsignificentbitsmusthavea=binputconnectedtologic1levelanda

> b and a < b inputs connected to logic 0 and 1 level.

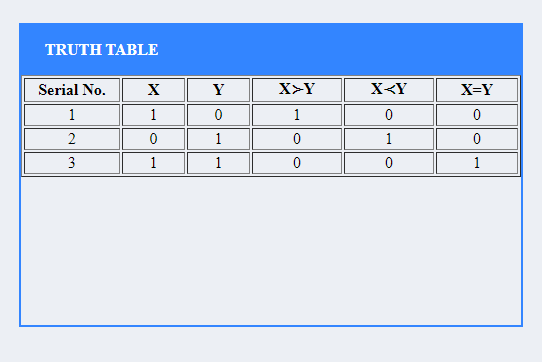
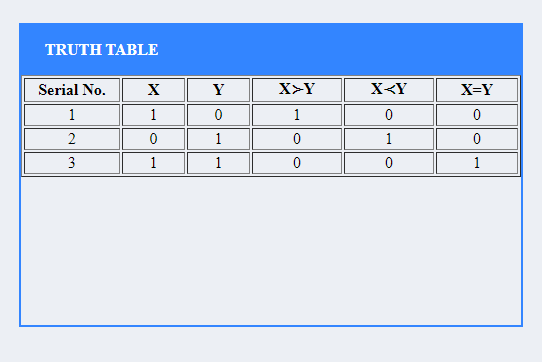
**INPUT**

Figures shows the block diagram of n-bit comparator .it receives two n-bit numbers a and b as inputs and out puts and a > b , a < b , a =b , depending upon the relative magnitude of two numbers , one of out put will be high .

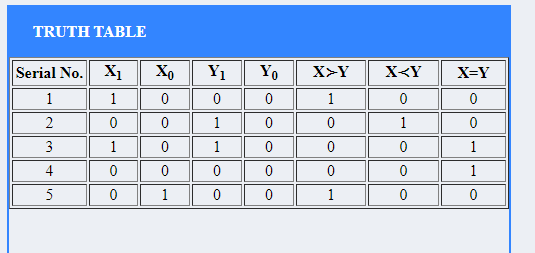
|  |  |  |  |
| --- | --- | --- | --- |
| A1 | AO | B1 | BO |
| 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 1 |

|  |  |  |  |
| --- | --- | --- | --- |
| 0 | 0 | 1 | 0 |
| 0 | 0 | 1 | 1 |
| 0 | 1 | 0 | 0 |
| 0 | 1 | 0 | 1 |
| 0 | 1 | 1 | 0 |
| 0 | 1 | 1 | 1 |
| 1 | 0 | 0 | 0 |
| 1 | 0 | 0 | 1 |
| 1 | 0 | 1 | 0 |
| 1 | 0 | 1 | 1 |
| 1 | 1 | 0 | 0 |
| 1 | 1 | 0 | 1 |
| 1 | 1 | 1 | 0 |
| 1 | 1 | 1 | 1 |

**OUTPUT-1 BIT COMPARATOR**



**OUTPUT-2-BIT COMAPARATOR**



**PROCEDURE:-**

1. Connect the input to the circuit as per the Pin outdiagram.
2. Consequently apply theinput.
3. Check the output according to the tablegiven.

PRECAUTIONS :-

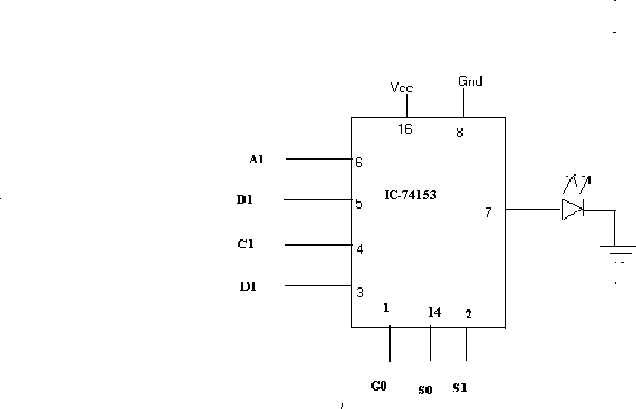
1. The connections should betight.
2. The pins of IC should not beshorted
3. Circuit Should be check by prescribedauthority

**RESULT :- The truth table of comparator is verified .**

EXPERIMENT -4(A)

**AIM:-** To verify the operation of Multiplexer

**APPARATUS:-** IC-74153,Connecting wires , high/low pulses , power supply ,LED.

CIRCUIT DIAGRAM:-

**TRUTHTABLE :-**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| S.No. | Strobe Go | Select I/P | | Output Y |
| S1 | S1 |
| 1. | 0 | 0 | 0 | A1 |
| 2. | 0 | 0 | 1 | B1 |
| 3. | 0 | 1 | 0 | C1 |
| 4. | 0 | 1 | 1 | D1 |

18 **|** P a g e

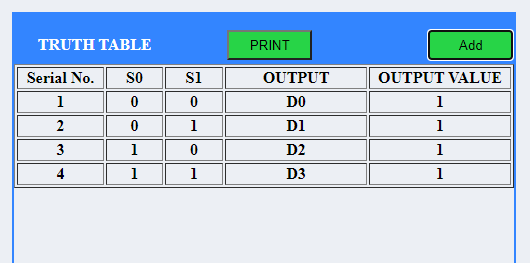
**THEORY** :- Multiplexer means many into one i.e. multiplexer is a logic circuit which has many inputs but single output . A multiplexer accepts several data inputs but allow only one of them at a time to get through to the output . The inputs and outputs are indicated by means of broad allow to indicate that there may be one or more inputs. Depending upon the digital code applied at the select inputs ,one out of the N data sources (D0,D1,---------Dn-1) is selected and transmitted the single output channel . A 4 to 1 line multiplexer has four inputs but only single output .

To perform 4 to 1 line multiplexer experiment . We have used IC 74153 .It has 4 line inputs (A1,B1,C1,D1) and only one outputs Y1.Go is the strobe input (active low )S0 and S1 are select input lines ,select one out of four inputs at output for e.g. S0,S1 =00 then A1 will be selected .

PROCEDURE :-

1. Connect circuit according to S.No. 1 of truth table.
2. Connect output of the circuit to output indicator. .
3. Switch on the instrument by using ON/OFF switch provided on the frontpanel
4. Verify the truth table for other sets of input and observe the output indicator , compare that output with truth table.

**OUTPUT-**



**PRECAUTIONS**:-

* 1. The connections should betight.
  2. The pins of IC should not beshorted
  3. Circuit should be check by prescribedauthority.

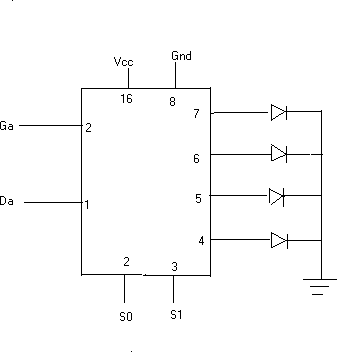
**RESULT :-** The truth table for multiplexer is verified

EXPERIMENT 4(B)

**AIM:- To verify the operation of Demultiplexer**

**APPARATUS:- IC-74155, Connecting wires , high/low pulses ,power supply ,LED.**

CIRCUIT DIAGRAM: -



**IC**

**TRUTH TABLE:-**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| S.No. | Strobe Ga | Select I/P | | Output  Y |
| S0 | S1 |
| 1. | 0 | 0 | 0 | Y0 |
| 2. | 0 | 0 | 1 | Y1 |
| 3. | 0 | 1 | 0 | Y2 |
| 4. | 0 | 1 | 1 | Y3 |

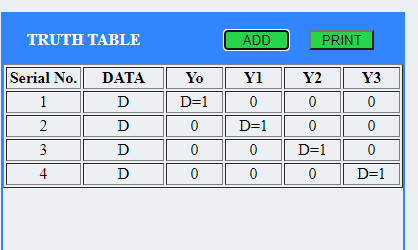
**THEORY**:- Demultiplexer means One into many i.e. demultiplexer is a logic circuit which has single input but many output . It accepts a single I/P and distribute it over several outputs. The block diagram of a demultiplexer is shown in circuit diagram . The select I/P code determines to which O/P the data I/P will be transmitted .A 1 to 4 line demultiplexer has 1 input & 4 outputs.

To perform 1 to 4 demultiplexer experiment . We used IC 74155.It has 2 data select line inputs (S0 & S1) and strobe input Ga (active low).

PROCEDURE:-

**OUTPUT-**

1. Connect the circuit according to S.N. 1 of truth table given above.
2. Switch on the instrument by using ON/OFF switch provided on the frontpanel.
3. Apply data inputs at data inputs Da & also connect Ga to ‘O’ level.
4. Verify the truth table for other sets of input and compare that output with truthtable



PRECAUTIONS:-

* 1. The connections should betight.
  2. The pins of IC should not beshorted
  3. Circuit Should be check by prescribedauthority.

**RESULT:- The truth table of demultiplexer is verified.**

EXPERIMENT 5

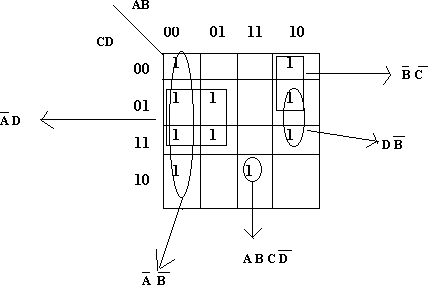
**AIM:- Design and realize a given function using K-Maps and verify its performance.**

**APPARATUS :-** NAND gate IC, Connecting wires.

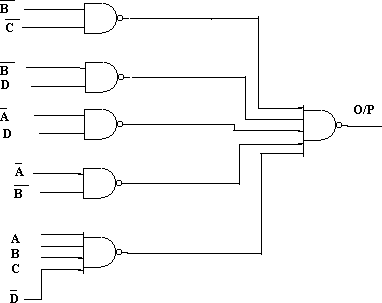
**\_ \_ \_ \_ \_ \_**

**EQUATIONFORMED**:- f(A,B,C,D)=ABCD + BC + BD + AD +AB

**K-MAP & LOGIC DIAGRAM FORMED**



**CIRCUIT DIAGRAM:-**



**THEORY: -The k-map of equation is shown in fig. The equation is minimized in the following steps:**

1. Encircl 1 in cell 14 which can’t be combined with any other 1.Theterm

**corresponding to this is ABCD.**

1. There are at least two possible ways for every 1 forming groups of two adjacent ones. therefore we ignore it for the time being & go to nextstep.

**3 There is only one possible group of 4 adjacent ones involving each of the cell 8,11,5 or 7 & 2.these are(8,9,0,1) , (11,9,1,3) ,(5,7,3,1)&(2,3,1,0) resp.encircle these groups. The terms corresponding to these groups are B C, B D,A D & A B resp.**

**since all the ones have been encircled therefore the minimized equation is :**

**\_ \_ \_ \_ \_ \_ \_ f(A,B,C,D)=ABCD + BC + BD + AD + AB**

**Procedure:-**

1. Connect the circuit according to circuitdiagram
2. Verify the result by making various combinations of logic 0 and logic1.

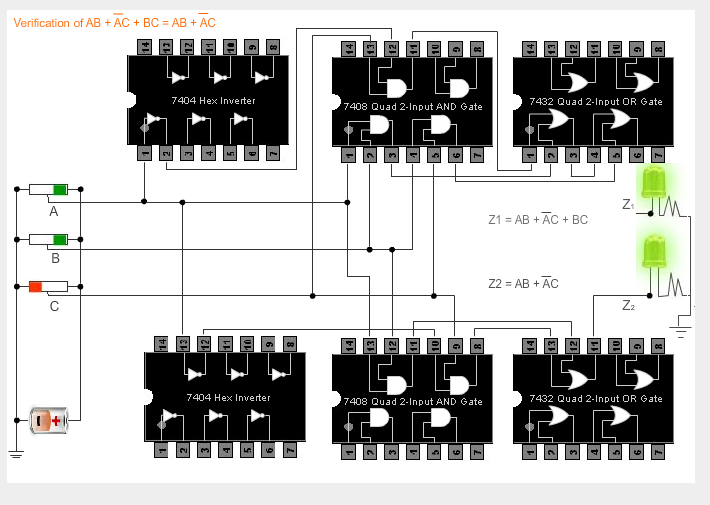
PRECAUTION:

1. The connections should betight.
2. The pins of IC should not beshorted
3. Circuit should be check by prescribedauthority.

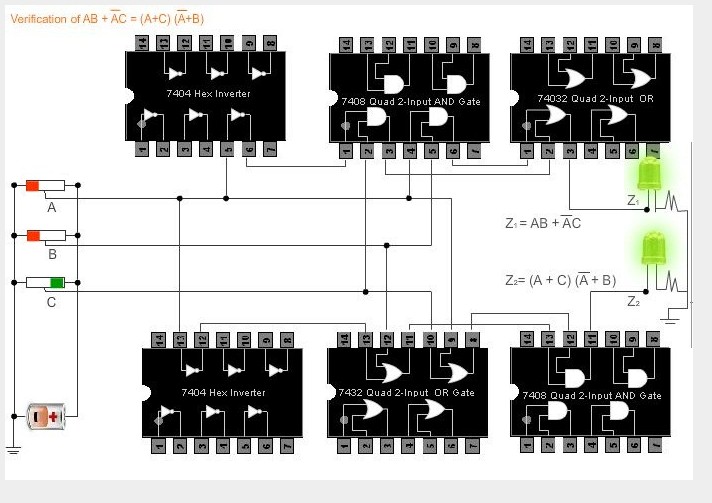
**RESULT:- The minimized equation is verified with the help of logic gates.**

**OUTPUT-**

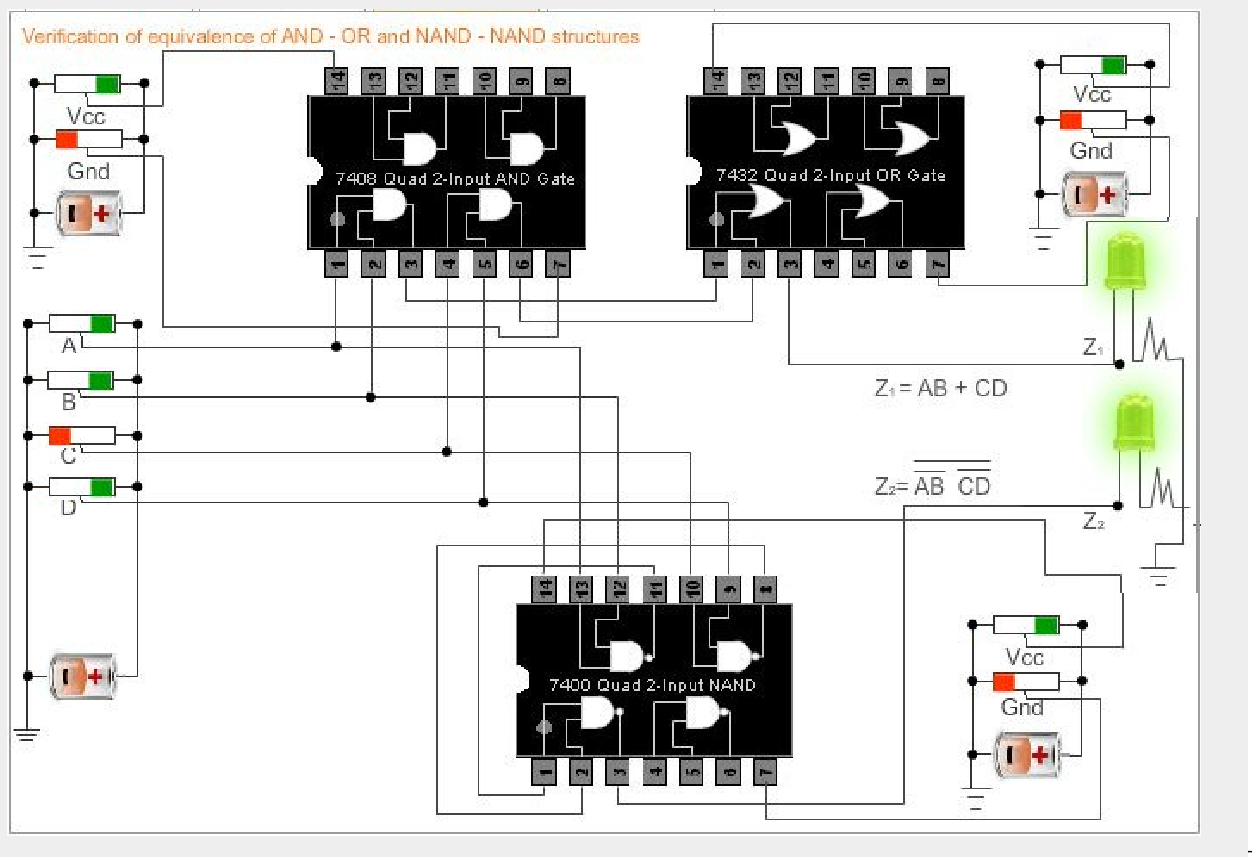
EXPERIMENT 1



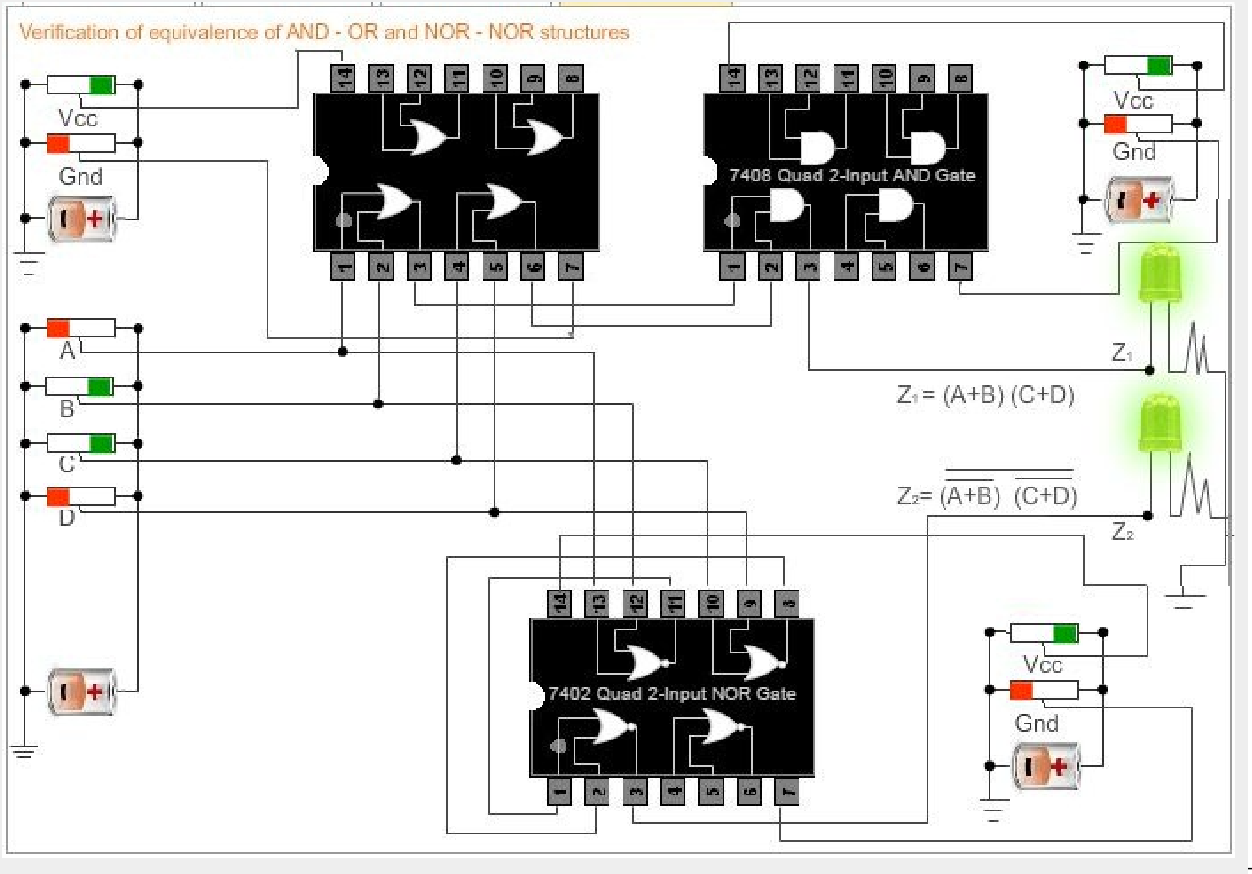
EXPERIMENT 2



EXPERIMENT 3-



EXPERIMENT 4-



**EXPERIMENT 6**

**AIM:-** To verify the truth table of S-R, J-K, T, D Flip-flops.

**APPARATUS:-** NAND gate, Low / High Pulse, clock, connecting wires.

**Theory:-**In case of sequential circuits the effect of all previous inputs on the outputs is represented by a state of the circuit. Thus, the output of the circuit at any time depends upon its current state and the input. These also determine the next state of the circuit. The relationship that exists among the inputs, outputs, present states and next states can be specified by either the state table or the state diagram.

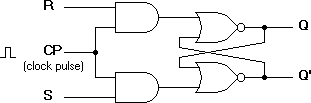
**State Table:-**The state table representation of a sequential circuit consists of three sections labelledpresent state next stateand output. T he p r e sen t s t a t e

d e s i gn a t e s t h e s t a t e o f f l i p - f l op s be f o re t h e occurrence of a clock pulse. The next state shows the states of flip-flops after the clock pulse, and the output section lists the value of the output variables during the present state.

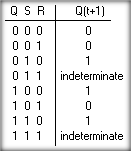
**Flip-Flop:-**The basic one bit digital memory circuit is known as flip-flop.It can store either 0or 1. Flip-flops are classifieds according to the number of inputs.

**S-R Flip-Flop:-** The circuit is similar to SR latch except enable signal is replaced by clock pulse.

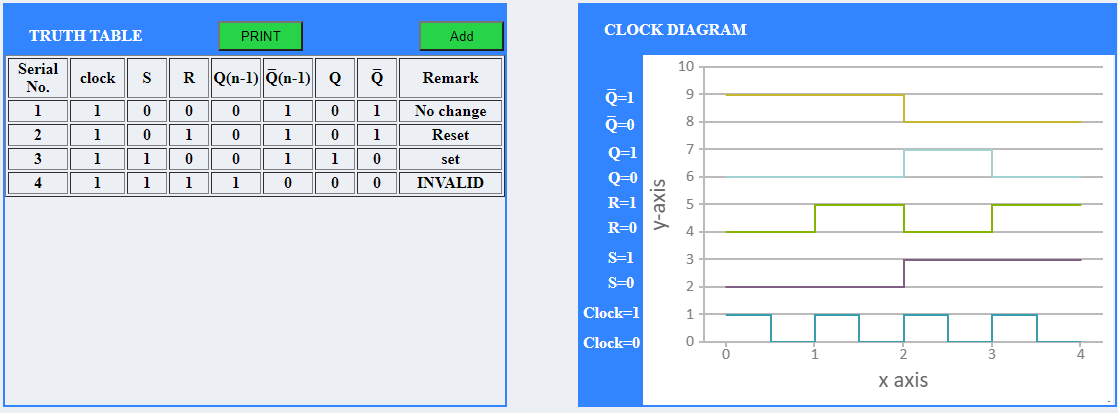
**Logic Diagram**



**Characteristic table for S-R flip flop**

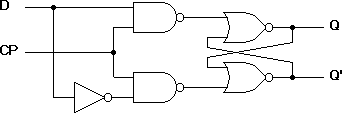


**OUTPUT-**



**D Flip-Flop:-The modified clocked SR flip-flop is known as D-flip-flop.From the truth tableof SR flip-flop we see that the output of the SR flip-flop is in unpredictable state when the inputsare same and high. In many practical applications, these input conditions are not required. These inputconditions can be avoided by making then complement of each other.**

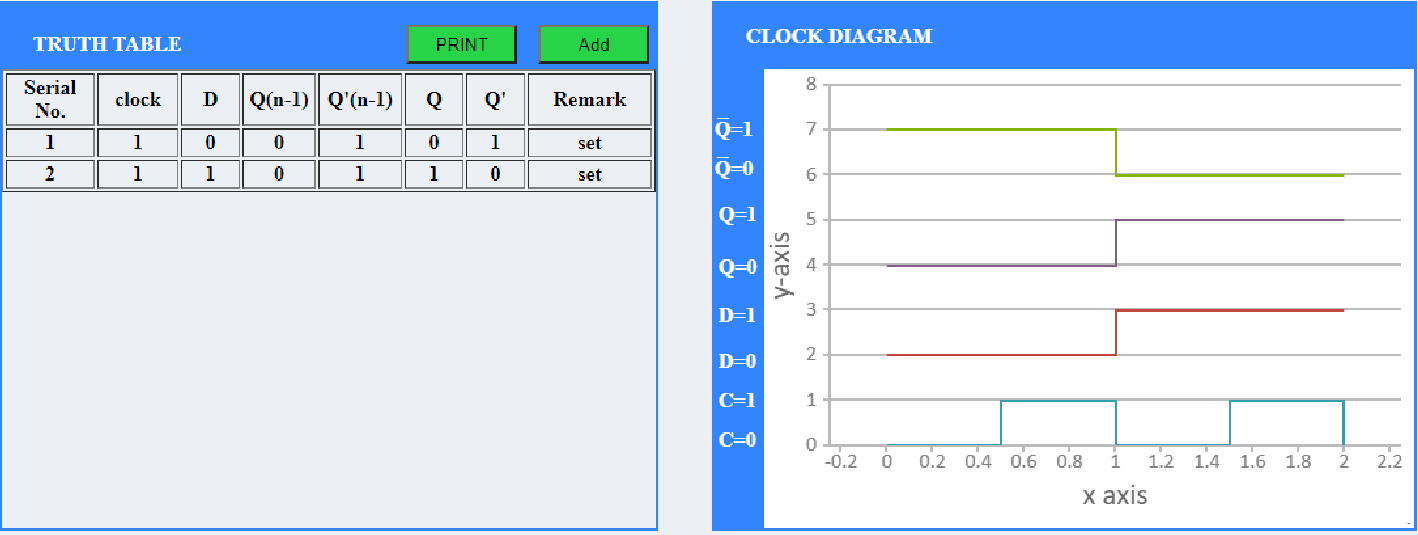
**Logic Diagram**



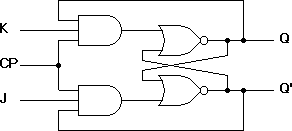
**Characteristic table for D flip flop**



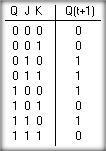
OUTPUT-

**J-K Flip-Flop:- In a RS flip-flop the input R=S=1 leads to an indeterminate output. The RSflip-flop circuit may be re-joined if both inputs are 1 than also the outputs are complement of each other.**

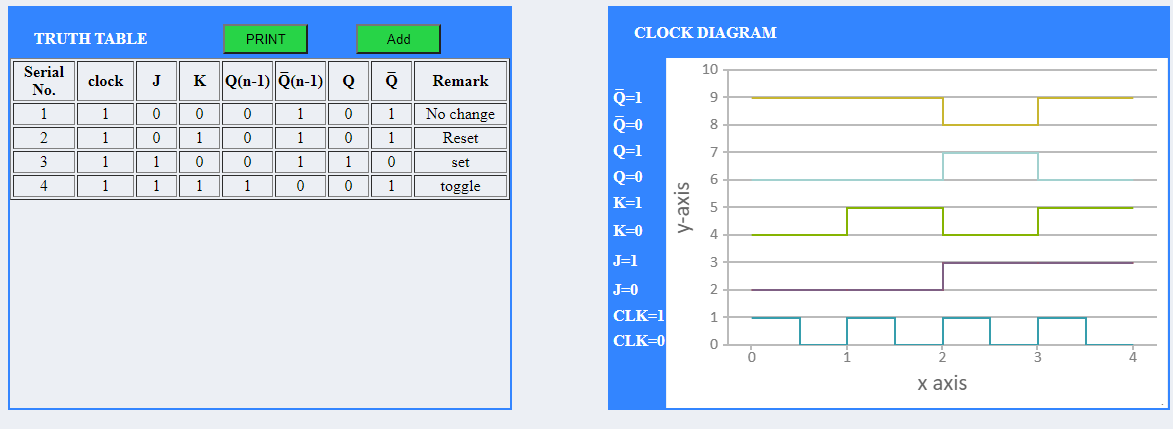
**Logic Diagram**



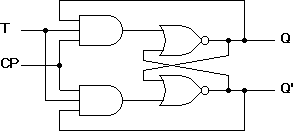
**Characteristic table for J-K flip flop**



OUTPUT-



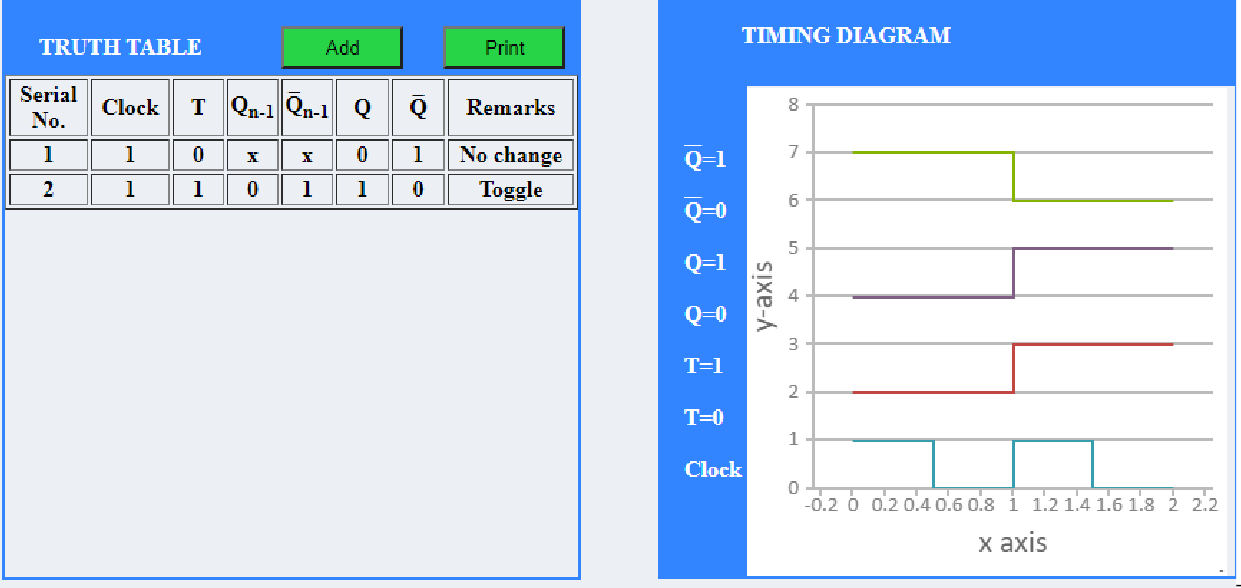
**T Flip-Flop:-T flip-flop is known as toggle flip-flop. The T flip-flop is modification of the J-Kflip-flop.BoththeJKinputsoftheJKflip-fl o p a r e h e l d at**

**l o gic 1 an d t h e c l oc k s i gn a l continuous to change. Logic Diagram**

**Characteristic table for T flip flop**



Output: -



**Procedure:-**

1. Connectionsare madeaspercircuitdiagram.
2. Verify truth-tables for various combinations of input.

3.

**RESULT:- Study and verified truth-tables of various flip-flops.**

**Procedure:-**

* 1. All the IC’s should be checked before use theapparatus.
  2. All LED’s should bechecked.
  3. All connections should betight.
  4. Always connect GROUND first and thenVcc
  5. The circuit should be off before change the connections.6.After completing the experiment switch off the supply toapparatus.

EXPERIMENT 7(A)

Aim:**- Design, and verify the 4-bit synchronous counter.**

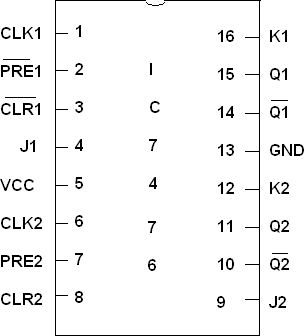
**Apparatus:-**

|  |  |  |  |
| --- | --- | --- | --- |
| S.No. | COMPONENT | SPECIFICATION | QTY. |
| 1. | JK FLIP FLOP | IC 7476 | 2 |
| 2. | 4 I/P AND GATE | IC 7411 | 1 |
| 3. | OR GATE | IC 7432 | 1 |
| 4. | XOR GATE | IC 7486 | 1 |
| 5. | NOT GATE | IC 7404 | 1 |
| 6. | IC TRAINER KIT | - | 1 |
| 7. | PATCH CORDS | - | 35 |

**Theory:-**

**Synchronous counter:** -A simple way of implementing the logic for each bit of an ascending counter (which is what is depicted in the image to the right) is for each bit to toggle when all of the less significant bits are at a logic high state. For example, bit 1 toggles when bit 0 is logic high; bit 2 toggles when both bit 1 and bit 0 are logic high; bit 3 toggles when bit 2, bit 1 and bit 0 are all high; and so on.

Synchronous counters can also be implemented with hardware finite state machines.Which are more complex but allow for smoother, more stable transitions? Hardware-based counters are of this type and they can be implemented using the IC 7476



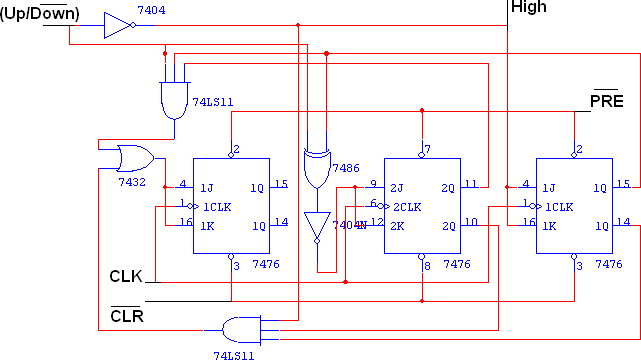
PIN DIAGRAM FOR IC 7476:

TRUTH TABLE:-

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Input**  **Up/Down** | **PresentState**  **QA QB QC** | | | **NextState**  **QA+1 Q B+1 QC+1** | | | **JA** | **A** | **KA** | **JB** | **B** | **KB** | **JC** | **C** | **KC** |
| **0** | **0** | **0** | **0** | **1** | **1** | **1** | **1** |  | **X** | **1** |  | **X** | **1** |  | **X** |
| **0** | **1** | **1** | **1** | **1** | **1** | **0** | **X** |  | **0** | **X** |  | **0** | **X** |  | **1** |
| **0** | **1** | **1** | **0** | **1** | **0** | **1** | **X** |  | **0** | **X** |  | **1** | **1** |  | **X** |

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **0** | **1** | **0** | **1** | **1** | **0** | **0** | **X** | **0** | **0** | **X** | **X** | **1** |
| **0** | **1** | **0** | **0** | **0** | **1** | **1** | **X** | **1** | **1** | **X** | **1** | **X** |
| **0** | **0** | **1** | **1** | **0** | **1** | **0** | **0** | **X** | **X** | **0** | **X** | **1** |
| **0** | **0** | **1** | **0** | **0** | **0** | **1** | **0** | **X** | **X** | **1** | **1** | **X** |
| **0** | **0** | **0** | **1** | **0** | **0** | **0** | **0** | **X** | **0** | **X** | **X** | **1** |
| **1** | **0** | **0** | **0** | **0** | **0** | **1** | **0** | **X** | **0** | **X** | **1** | **X** |
| **1** | **0** | **0** | **1** | **0** | **1** | **0** | **0** | **X** | **1** | **X** | **X** | **1** |
| **1** | **0** | **1** | **0** | **0** | **1** | **1** | **0** | **X** | **X** | **0** | **1** | **X** |
| **1** | **0** | **1** | **1** | **1** | **0** | **0** | **1** | **X** | **X** | **1** | **X** | **1** |
| **1** | **1** | **0** | **0** | **1** | **0** | **1** | **X** | **0** | **0** | **X** | **1** | **X** |
| **1** | **1** | **0** | **1** | **1** | **1** | **0** | **X** | **0** | **1** | **X** | **X** | **1** |
| **1** | **1** | **1** | **0** | **1** | **1** | **1** | **X** | **0** | **X** | **0** | **1** | **X** |
| **1** | **1** | **1** | **1** | **0** | **0** | **0** | **X** | **1** | **X** | **1** | **X** | **1** |

**TRUTH TABLE OF 4 BIT SYNCHRONOUS COUNTERS**

LOGIC DIAGRAM

Procedure:-

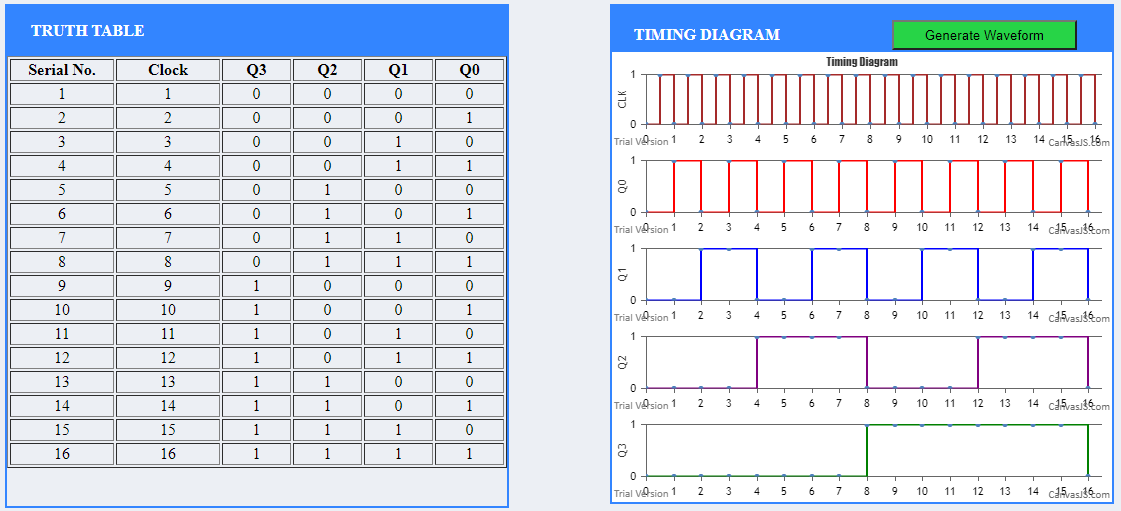
* + 1. Connections are given as per circuitdiagram.
    2. Logical inputs are given as per circuitdiagram.
    3. Observe the output and verify the truthtable.

Result:**-Study of 4 bit synchronous counters and verified its truth table.**

Precautions:-

* + - 1. All ICs should be checked before starting theexperiment.
      2. All the connection should betight.
      3. Always connect ground first and then connect Vcc.
      4. Suitable type wire should be used for different types ofcircuit.
      5. The kit should be off before change theconnections.
      6. After completed the experiments switch off the supply of theapparatus

**OUTPUT-**



**EXPERIMENT 7(B)**

AIM:- **To design and verify 4 bit ripple (asynchronous counter).**

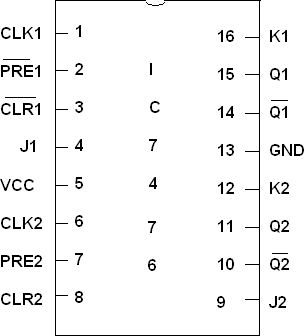
APPARATUS REQUIRED:-

|  |  |  |  |
| --- | --- | --- | --- |
| S.No. | COMPONENT | SPECIFICATION | QTY. |
| 1. | JK FLIP FLOP | IC 7476 | 2 |
| 2. | NAND GATE | IC 7400 | 1 |
| 3. | IC TRAINER KIT | - | 1 |
| 4. | PATCH CORDS | - | 30 |

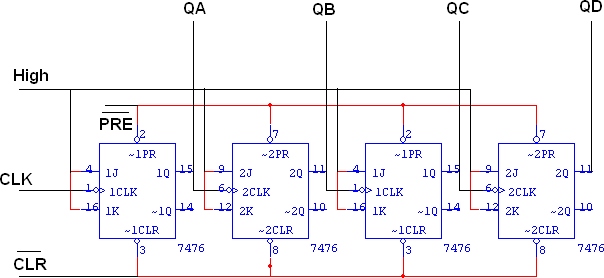
**THEORY:-**

A counter is a register capable of counting number of clock pulse arriving at its clock input. Counter represents the number of clock pulses arrived. A specified sequence of states appears as counter output. This is the main difference between a register and a counter. There are two types of counter, synchronous and asynchronous. In synchronous common clock is given to all flip flop and in asynchronous first flip flop is clocked by external pulse and then each successive flip flop is clocked by Q or Q output of previous stage. A soon the clock of second stage is triggered by output of first stage. Because of inherent propagation delay time all flip flops are not activated at same time which results in asynchronous operation.

PIN DIAGRAM FOR IC 7476:



LOGIC DIAGRAM FOR 4 BIT RIPPLE COUNTER:



TRUTH TABLE:-

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **CLK** | **QA** | **QB** | **QC** | **QD** |
| **0** | **0** | **0** | **0** | **0** |
| **1** | **1** | **0** | **0** | **0** |
| **2** | **0** | **1** | **0** | **0** |
| **3** | **1** | **1** | **0** | **0** |
| **4** | **0** | **0** | **1** | **0** |
| **5** | **1** | **0** | **1** | **0** |
| **6** | **0** | **1** | **1** | **0** |
| **7** | **1** | **1** | **1** | **0** |
| **8** | **0** | **0** | **0** | **1** |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **9** | **1** | **0** | **0** | **1** |
| **10** | **0** | **1** | **0** | **1** |
| **11** | **1** | **1** | **0** | **1** |
| **12** | **0** | **0** | **1** | **1** |
| **13** | **1** | **0** | **1** | **1** |
| **14** | **0** | **1** | **1** | **1** |
| **15** | **1** | **1** | **1** | **1** |

Procedure:-

1. Connections are given as per circuitdiagram.
2. Logical inputs are given as per circuitdiagram.
3. Observe the output and verify the truthtable.

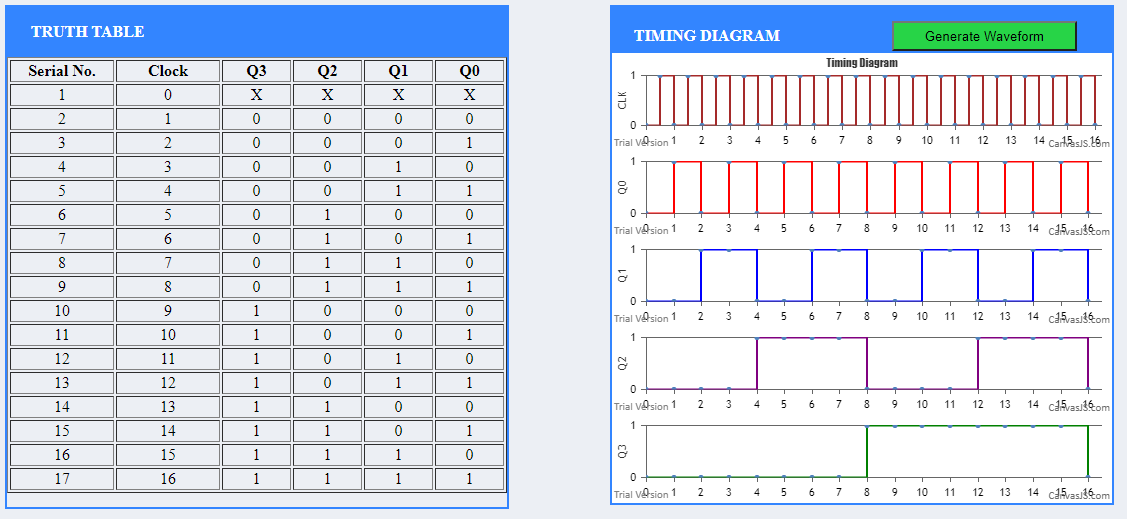
**Result: -**

**Study of 4 bit synchronous counter and verified its truth table**

Precautions:-

1. AllICsshouldbecheckedbeforestartingtheexperiment.
2. All the connection should betight.
3. AlwaysconnectgroundfirstandthenconnectVcc.
4. Suitable type wire should be used for different typesof circuit.
5. The kit should be off before change theconnections.
6. Aftercompletedtheexperimentsswitchoffthesupplyoftheapparatus

OUTPUT-



EXPERIMENT 8

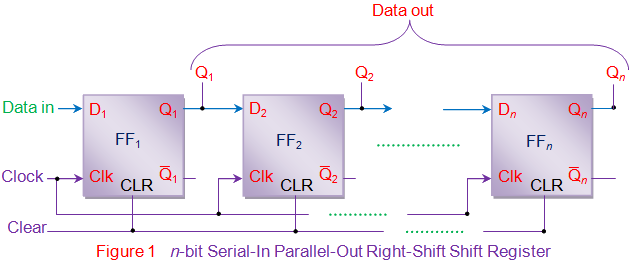
Aim

**To analyze the circuit and truth table of 4-bit SIPO (serial input parallel output) shift register by using IC 7474 (D flip flop).**

Theory

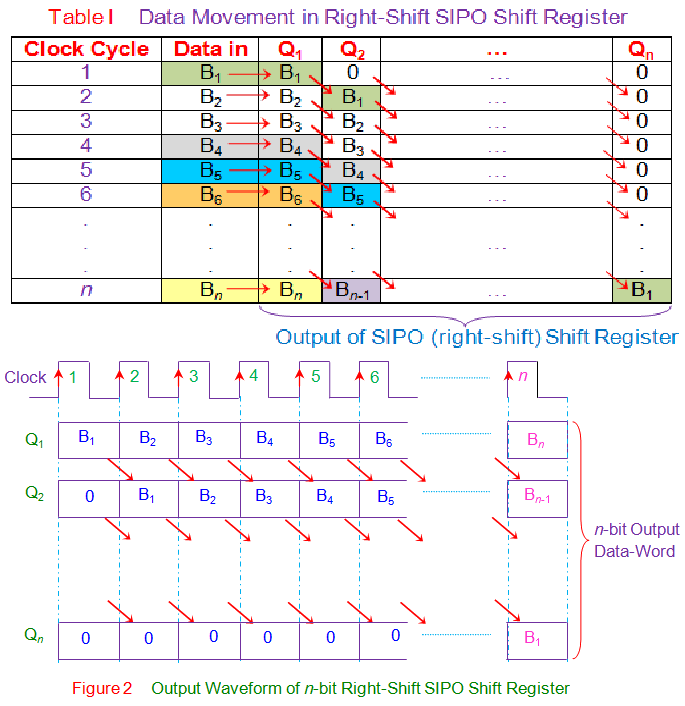
**Introduction**

In Serial In Parallel Out (SIPO) shift registers, the data is stored into the register serially while it is retrieved from it in parallel-fashion. Figure 1 shows an n-bit synchronous SIPO shift register sensitive to positive edge of the clock pulse. Here the data word which is to be stored (Data in) is fed serially at the input of the first flip-flop (D1 of FF1). It is also seen that the inputs of all other flip-flops (except the first flip-flop FF1) are driven by the outputs of the preceding ones like the input of FF2 is driven by the output of FF1. In this kind of shift register, the data stored within the register is obtained as a parallel-output data word (Data out) at the individual output pins of the flip-flops (Q1 to Qn).



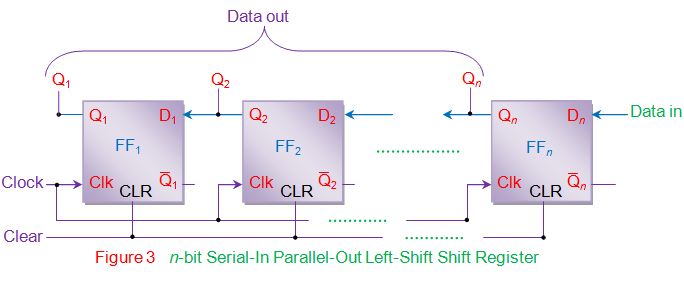
In general, the register contents are cleared by applying high on the clear pins of all the flip-flops at the initial stage. After this, the first bit, B1 of the input data word is fed at the D1 pin of FF1.This bit (B1) will enter into FF1, get stored and thereby appears at its output Q1 on the appearance of first leading edge of the clock. Further at the second clock pulse, the bit B1 right- shifts and gets stored into FF2 while appearing at its output pin Q2 while a new bit, B2 enters into FF1. Similarly at each clock pulse the data within the register moves towards right by a single bit while a new bit of the input word enters into the register. Meanwhile one can extract the bits stored within the register in parallel-fashion at the individual flip-flop outputs.

Analyzing on the same grounds, one can note that the n-bit input data word is obtained as an n- bit output data word from the shift register at the rising edge of the nth clock pulse. This working of the shift-register can be summarized as in Table I and the corresponding waveforms are given by figure 2.

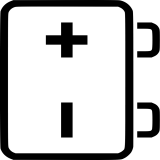


In the right-shift SIPO shift-register, data bits shift from left to right for each clock pulse. However if the data bits are made to shift from right to left in the same design, one gets a left- shift SIPO shift-register as shown by figure 3. Nevertheless the basic working principle remains the same except the fact that now Bn down to B1 is stored in Qn down to Q1 i.e. Q1 = B1, Q2 =

B2 … Qn = Bn at the nth clock pulse.



Procedure

Step-1) Connect the supply(+5V) to the circuit.

Step-2) Keep the Reset and Preset as active-high signals . Step-3) Apply the data at serial input .



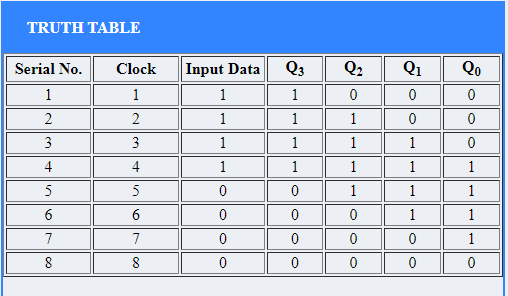
Step-4) Press clock pulse and observe this data at LED Q3. Step-5) Then press "ADD" button to add data in the given truth table.

Step-6) Apply the next data at serial input.

Step-7) Press clock pulse and observe that the data at LED Q3will shift to LED Q2 and the new data applied will appear at Q3.

Step-8) Repeat steps 3 to 5 till all the 4 bits appear at the output of shift register. Step-9) Press the "Print" button after completing your simulation to get your results.

Output-



Experiment 9

***AIM-To verify the truth table of half subtractor and full subtractor.***

**APPARATUS**- ***the ICs of XOR, NOT and AND gates and of full subtractor.and the ICs of XOR, AND, NOT and OR gates for half adder, LEDS.***

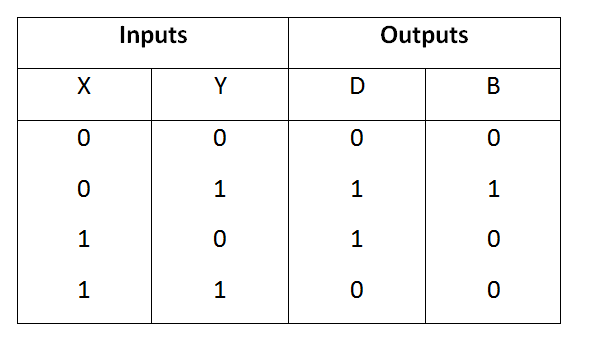
**THEORY-** Subtractor circuits take two binary numbers as input and subtract one binary number input from the other binary number input. Similar to adders, it gives out two outputs, difference and borrow (carry-in the case of Adder). There are two types of subtractors.

1. Half Subtractor
2. Full Subtractor
3. Half Subtractor

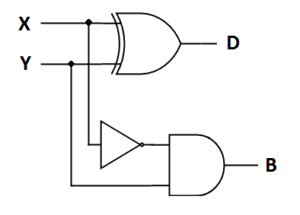
The half-subtractor is a combinational circuit which is used to perform subtraction of two bits. It has two inputs, X (minuend) and Y (subtrahend) and two outputs D (difference) and B (borrow). The logic symbol and truth table are shown below.



**Figure-1:Logic Symbol of Half subtractor**



**Figure-2:Truth Table of Half subtractor**



**Figure-3: Circuit Diagram of Half subtractor**

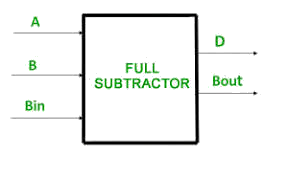
From the above truth table we can find the boolean expression.

**D = X** ⊕ **Y B = X' Y**

From the equation we can draw the half-subtractor circuit as shown in the figure 3.

1. Full Subtractor

A full subtractor is a combinational circuit that performs subtraction involving three bits, namely minuend, subtrahend, and borrow-in. It accepts three inputs: minuend, subtrahend and a borrow bit and it produces two outputs: difference and borrow. The logic symbol and truth table are shown below.



**Figure-4: Logic Symbol of Full subtractor**

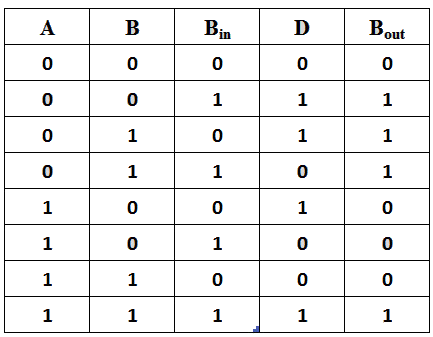


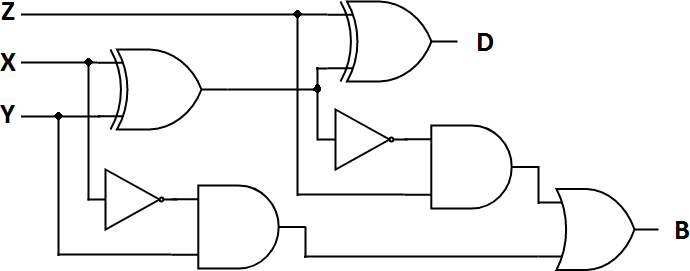
Figure-5: Truth Table of Full subtractor

From the above truth table we can find the boolean expression.

**D = A** ⊕ **B** ⊕ **Bin**

**B = A' Bin + A' B + B Bin**

From the equation we can draw the Full-subtractor circuit as shown in the figure 6.



Procedure-

**Figure-6: Circuit Diagram of Full subtractor**

1. HALF SUBTRACTOR

Step-1) Connect the Supply(+5V) to the circuit.

Step-2) First press "ADD" button to add basic state of your output in the given table. Step-3) Press the switches to select the required inputs "A" and "B".

Step-4) Press "ADD" button to add your inputs and outputs in the given table. Step-5) Repeat steps 3&4 for next state of inputs and their corresponding outputs.

Step-6) Press the "PRINT" button after completing your simulation to get your results.

1. **FULL SUBTRACTOR**

Step-1) Connect the Supply(+5V) to the circuit.

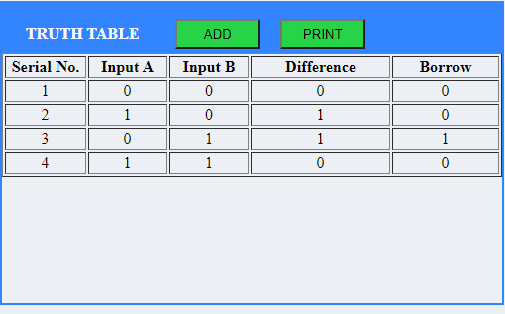
Step-2) First press "ADD" button to add basic state of your output in the given table. Step-3) Press the switches to select the required inputs "A" and "B" and "Carry Input". Step-4) Press "ADD" button to add your inputs and outputs in the given table.

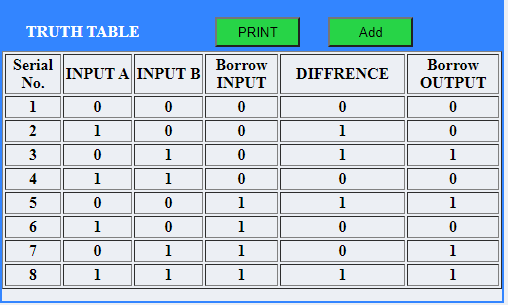
Step-5) Repeat steps 3&4 for next state of inputs and their corresponding outputs. Step-6) Press the "PRINT" button after completing your simulation to get your results.

**RESULT:- The truth table of Full Subtractor is verified.**

OUTPUT-

HALF SUBTRACTOR-





**FULL SUBTRACTOR-**

EXPERIMENT 10

**AIM-** Verify Binary to Gray and Gray to Binary conversion using NAND gates only.

**Apparatus*- NAND gates, LEDs display.***

Theory-

Introduction

Binary Numbers is default way to store numbers, but in many applications binary numbers are difficult to use and a variation of binary numbers is needed. This is where Gray codes are very useful.

Gray code has property that two successive numbers differ in only one bit because of this property gray code does the cycling through various states with minimal effort and used in K- maps, error correction, communication etc.

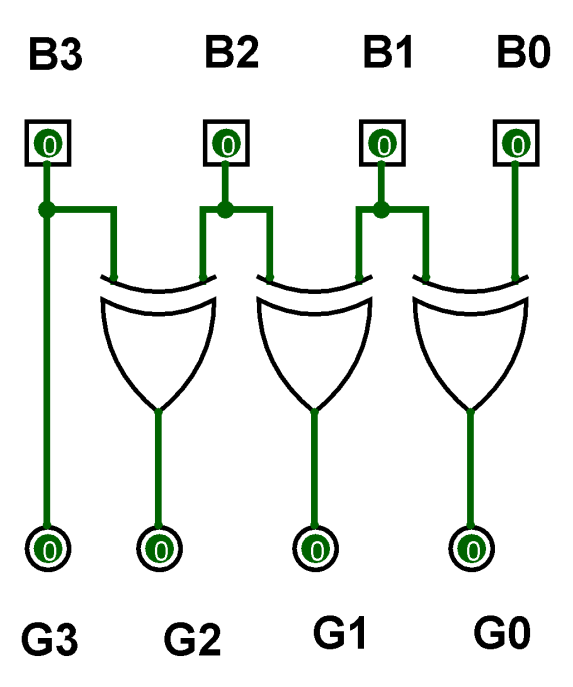
In computer science many a times we need to convert binary code to gray code and vice versa. This conversion can be done by applying following rules :

1. Binary to Gray conversion :
   1. The Most Significant Bit (MSB) of the gray code is always equal to the MSB of the given binary code.
   2. Other bits of the output gray code can be obtained by Ex-ORing binary code bit at that index and previous index.

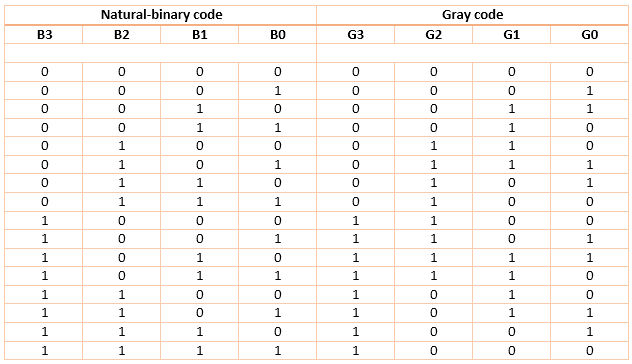
There are four inputs and four outputs. The input variable are defined as B3, B2, B1, B0 and the output variables are defined as G3, G2, G1, G0. From the truth table, combinational circuit is designed.The logical expressions are defined as :

**B3 = G3**

**B2** ⊕ **B3 = G2 B1** ⊕ **B2 = G1 B0** ⊕ **B1 = G0**



**Figure-1: Binary to Gray Code Converter Circuit**



**Figure-2: Binary to Gray Code Converter Truth Table**

1. ***Gray to binary conversion :***
2. The Most Significant Bit (MSB) of the binary code is always equal to the MSB of the given binary number.
3. Other bits of the output binary code can be obtained by checking gray code bit at that index. If current gray code bit is 0, then copy previous binary code bit, else copy invert of previous binary code bit.

There are four inputs and four outputs. The input variable are defined as G3, G2, G1, G0 and the

output variables are defined as B3, B2, B1, B0. From the truth table, combinational circuit is designed.The logical expressions are defined as :

**G0 G1 G2 G3 =B0**



⊕



⊕



⊕

**G1 G2 G3 = B1**



⊕



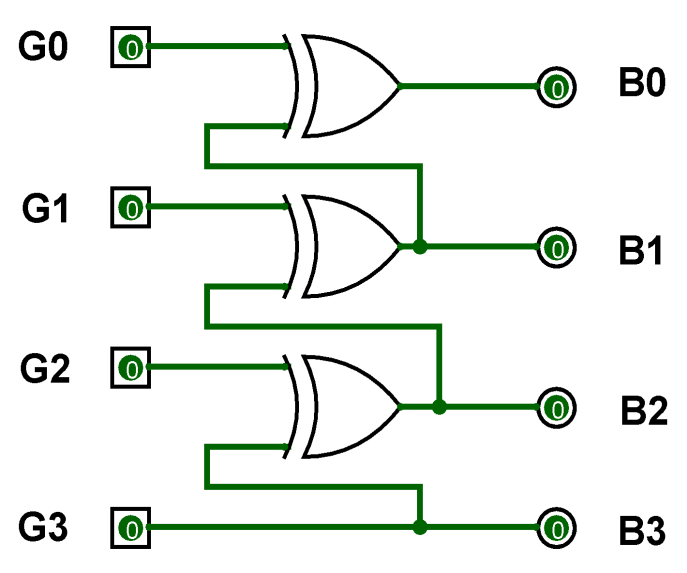
⊕

**G2 G3 = B2**

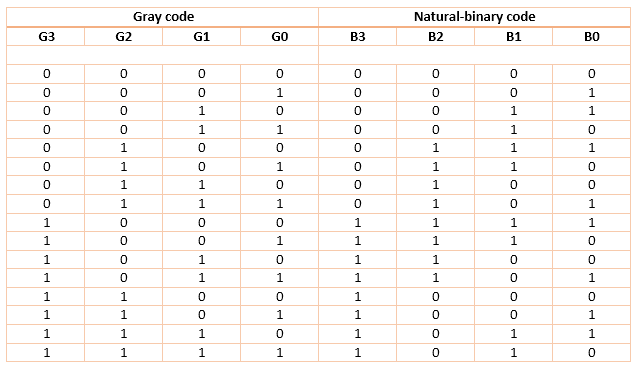


⊕

**G3 = B3**

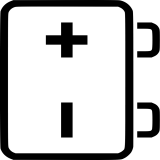


**Figure-3: Gray to Binary Code Converter Circuit**



**Figure-4: Gray to Binary Code Converter Truth Table**

Procedure-



Step-1) Connect the supply(+5V) to the circuit.

Step-2) Press different switches for different combination of inputs. Step-3) Then press "Add" button to add values to the Truth Table .

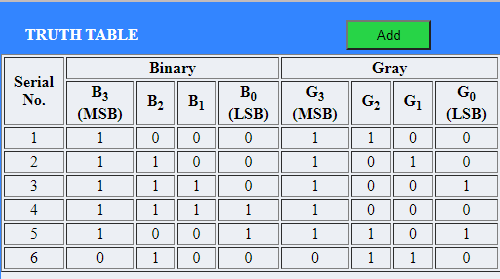
Step-4) Repeat step 2 and step 3 for another set of data.

Step-5) Press "Print" button after completing your simulation to get your results.

**RESULT:-** The truth table of binary to gray code and gray code to binary has been verified.

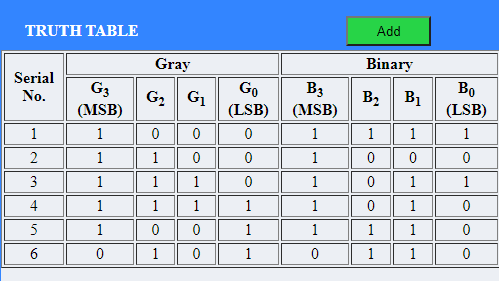
Output-

BINARY TO GRAY CODE CINVERTER-



GRAY TO BINARY CODE CONVERTER-

53



54