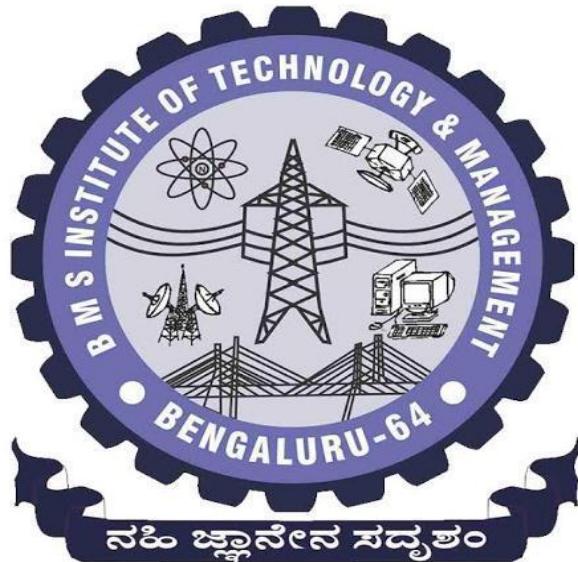


**BMS Institute of Technology & Management**  
**Yelahanka, Bangalore-560 064**



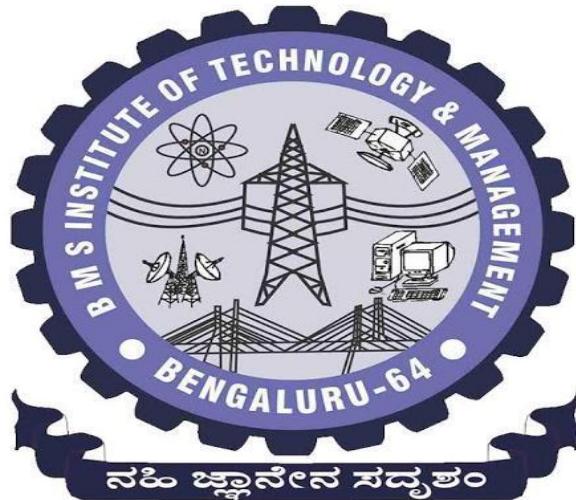
**ANALOG AND DIGITAL ELECTRONICS  
LABORATORY  
(15CSL37)**

**Laboratory Manual**

For  
**III Semester BE, ISE**

**Department of Information Science & Engineering  
BMS Institute of Technology & Mgmt Yelahanka,  
Bangalore-560 064**

**BMS Institute of Technology & Management**  
**Yelahanka, Bangalore-560 064**



**ANALOG AND DIGITAL  
ELECTRONICS LABORATORY  
(15CSL37)**  
**Laboratory Manual**

Prepared by  
Ms. Vinutha K & Ms. Ashwini N

Reviewed by  
Dr. Manjunath T N

**Department of Information Science & Engineering**  
**BMS Institute of Technology & Mgmt**  
**Yelahanka, Bangalore-560 064**

### **Vision of the Institute**

*To emerge as one of the finest technical institutions of higher learning to develop engineering professionals who are technically competent, ethical and environment friendly for betterment of society.*

### **Mission of the Department**

*Accomplish stimulating learning environment through high quality academic instruction, innovation and industry-institute interface.*

## **Department of Information Science and Engineering**

### **Vision of the Department**

*Emerge as centre of learning in the field of information science & engineering with technical competency to serve the society.*

### **Mission of the Department**

*To provide excellent learning environment through contemporary teaching methods, innovation, mentoring and industry institute interaction.*

### **Program Educational Objectives:**

Graduates of the program will,

**PEO-1:** Successful professional career in Information Technology Industry.

**PEO-2:** Pursue higher studies for contemporary knowledge in IT industry.

**PEO-3:** Exhibit professionalism and team work with social concern.

## **PROGRAMME OUTCOMES**

Program outcomes are narrower statements that describe what students are expected to know and be able to do by the time of graduation. These relate to the skills, knowledge and behavior. Bachelor of Engineering Graduation students of Information science and Engineering program at B M S Institute of Technology will attain the following program outcomes.

### **Program Outcomes:**

**After the successful completion of the course, the graduate will be able to**

**PO1: Engineering knowledge:** Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.

**PO2: Problem analysis:** Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.

**PO3: Design/development of solutions:** Design solutions for complex engineering

problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.

**PO4: Conduct investigations of complex problems:** Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.

**PO5: Modern tool usage:** Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.

**PO6: The engineer and society:** Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.

**PO7: Environment and sustainability:** Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.

**PO8: Ethics:** 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 multidisciplinary settings.

**PO10: Communication:** Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.

**PO11: Project management and finance:** Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.

**PO12: Life-long learning:** Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

### **Course outcomes**

<b>CO1</b>	<b>Gaining Knowledge on various parts of a computer</b>
<b>CO2</b>	<b>Able to draw flowcharts and write algorithms</b>
<b>CO3</b>	<b>Able design and development of C problem solving skills</b>
<b>CO4</b>	<b>Able design and develop modular programming skills</b>
<b>CO5</b>	<b>Able to trace and debug a program</b>

## **ANALOG AND DIGITAL ELECTRONICS LABORATORY [As per Choice Based Credit System (CBCS) scheme]**

<b>Laboratory Code</b>	<b>15CSL37</b>	<b>IA Marks</b>	<b>20</b>
<b>Number of Lecture Hours/Week</b>	<b>01I + 02P</b>	<b>Exam Marks</b>	<b>80</b>
<b>Total Number of Lecture Hours</b>	<b>40</b>	<b>Exam Hours</b>	<b>03</b>

**CREDITS – 02**

### **Laboratory Experiments:**

#### **PART-A (Analog Experiments)**

1. a) Design and construct a Schmitt trigger using Op-Amp for given UTP and LTP values and demonstrate its working.
- b) Design and implement a Schmitt trigger using Op-Amp using a simulation package for two sets of UTP and LTP values and demonstrate its working.
2. a) Design and construct a rectangular waveform generator (Op-Amp relaxation oscillator) for given frequency and demonstrate its working.
- b) Design and implement a rectangular waveform generator (Op-Amp relaxation oscillator) using a simulation package and demonstrate the change in frequency when all resistor values are doubled.
3. Design and implement an Astable multivibrator circuit using 555 timer for a given frequency and duty cycle.

**NOTE:** Hardware and Software results need to be compared.

#### **PART-B (Digital Experiments)**

4. Design and implement Half adder, Full Adder, Half Subtractor, Full Subtractor using basic gates.
5. a) Given a 4-variable logic expression, simplify it using Entered Variable Map and realize the simplified logic expression using 8:1 multiplexer IC.
- b) Design and develop the Verilog /VHDL code for an 8:1 multiplexer. Simulate and verify its working.
6. a) Design and implement code converter I)Binary to Gray (II) Gray to Binary Code using basic gates.
7. Design and verify the Truth Table of 3-bit Parity Generator and 4-bit Parity Checker using basic Logic Gates with an even parity bit.
8. a) Realize a J-K Master / Slave Flip-Flop using NAND gates and verify its truth table.
- b) Design and develop the Verilog / VHDL code for D Flip-Flop with positiveedge triggering. Simulate and verify its working.
9. a) Design and implement a mod-n ( $n < 8$ ) synchronous up counter using J-K Flip-Flop ICs and demonstrate its working.
- b) Design and develop the Verilog / VHDL code for mod-8 up counter. Simulate and verify its working.
10. Design and implement an asynchronous counter using decade counter IC to count up from 0 to n ( $n \leq 9$ ) and demonstrate on 7-segment display (using IC-7447).
11. Generate a Ramp output waveform using DAC0800 (Inputs are given to DAC through IC74393 dual 4-bit binary counter).

#### **Study experiment**

12. To study 4-bitALU using IC-74181.

### **Course objectives:**

This laboratory course enables students to get practical experience in design, assembly and evaluation/testing of

- Analog components and circuits including Operational Amplifier, Timer, etc.
- Combinational logic circuits.

- Flip - Flops and their
- Counters and Registers using Flip-
- Synchronous and Asynchronous Sequential
- A/D and D/A

### **Course outcomes:**

On the completion of this laboratory course, the students will be

- Use various Electronic Devices like Cathode ray Oscilloscope, Signal generators, Digital Trainer Kit, Multimeters and components like Resistors, Capacitors, Op amp and Integrated Circuit.
- Design and demonstrate various combinational logic
- Design and demonstrate various types of counters and Registers using Flip-flops
- Use simulation package to design circuits.
- Understand the working and

### **Graduate Attributes (as per NBA):**

1. Engineering Knowledge
2. Problem Analysis
3. Design/Development of Solutions
4. Modern Tool Usage

### **Conduction of Practical Examination:**

- 1 . All laboratory experiments (1 to 11 nos) are to be included for practical examination.
- 2 . Students are allowed to pick one experiment from the lot.
- 3 . Strictly follow the instructions as printed on the cover page of answer script.
- 4 . Marks distribution:
  - a ) For questions having part a only- Procedure + Conduction + Viva:**20 + 50 +10 =80 Marks**
  - b ) For questions having part a and b
    - Part a- Procedure + Conduction + Viva:**10 + 35 +05= 50 Marks**
    - Part b- Procedure + Conduction + Viva:**10 + 15 +05= 30 Marks**
- 5 . Change of experiment is allowed only once and marks allotted to the procedure part to be made zero.

### **Conduction of Lab:**

**Any simulation package like MultiSim / P-spice /Equivalent software may be used.** Faculty-in-charge should demonstrate and explain the required hardware components and their functional Block diagrams, timing diagrams etc. Students have to prepare a write-up on the same and include it in the Lab record and to be evaluated.

**Laboratory Session-1:** Write-upon analog components; functional block diagram, Pin diagram (if any), waveforms and description. The same information is also taught in theory class; this helps the students to understand better.

**Laboratory Session-2:** Write-upon Logic design components, pin diagram (if any), Timing diagrams, etc. The same information is also taught in theory class; this helps the students to understand better.

**Note:** **These TWO Laboratory sessions** are used to fill the gap between theory classes and practical sessions. Both sessions are to be evaluated for 20 marks as lab experiments.

## DO'S AND DON'TS

DO'S	DON'TS
Be regular to the Lab	Do not come late to the Lab
Follow proper Dress Code	Do not throw connecting wires on the Floor
Wear your College ID card	Do not operate the IC trainer kits without permission
Avoid unnecessary talking while doing the experiment	Avoid loose connection and short circuits
Take the signature of the lab in charge before taking the components	Do not interchange the ICs while doing the experiment
Handle the trainer kit properly	Do not panic if you do not get the output
Keep your work area clean after completing the experiment.	
After completion of the experiment switch off the power and return the components	
Arrange your chairs and tables before leaving the laboratory.	

## RULES AND REGULATIONS:

- **Lab Teams:**
  - Lab teams consisting of three students will be formed during the first lab session. It is expected that all team members will contribute to all the lab work.
- **Laboratory Preparation:**
  - Each student is responsible for maintaining his/her own laboratory observation Notebook. Each student is required to perform pre-lab work and enter it into his/her notebook.
- **Lab Work:**
  - Each lab team must be checked by the faculty. Check-out will be used to confirm that the actual lab work as recorded in the lab notebook has been completed and that the lab station has been properly cleaned up. The faculty will initial and date all the data acquired during the lab period. All pages signed by the allotted faculty on the same date in the lab itself.
- **Lab Completion:**
  - Each experiment should be completed during the lab period. If a group is unable to complete the lab work, they may complete it in the Break time, if granted permission by the instructor. The work must be checked to verify that all laboratory exercises are complete.
  - In case a student is ABSENT for a particular lab session, he/she has to compulsorily finish the experiment before the next lab session and get the observation signed All lab work should be completed before the next laboratory period.
  - **5 MARKS WILL BE DEDUCTED FROM THE RECORD FOR THE PARTICULAR EXPERIMENT IF THE STUDENT IS ABSENT FOR THE LAB.**
- **Lab Grading**
  - The faculty will examine your notebooks during lab period and assign a grade: 1. Execution (2 marks per lab): Each student should read the lab material and finish the observation before the lab. Pre-lab work should be turned in at the beginning of each lab session and should execute the experiments.
  - 2. Viva (2 marks per lab): There will be viva for each lab. The questions in the viva come from the lab material. No viva marks will be given if you are more than 10 minutes late.
  - 3. Lab record (5 marks per lab): Students will write a lab report according to the format specified and turn it in at the beginning of the next lab session. Late lab reports will not be accepted.

## **RULES FOR MAINTAINING LABORATORY RECORD:**

1. Put your name, USN and subject on the outside front cover of the record. Put that same information on the first page inside.
2. Update Table of Contents every time you start each new experiment or topic
3. Always use pen and write neatly and clearly
4. Start each new topic (experiment, notes, calculation, etc.) on a right-side (odd numbered) page
5. Obvious care should be taken to make it readable, even if you have bad handwriting
6. Date to be written every page on the top right side corner
7. On each right side page
8. On each left side page
  - Pin diagrams
  - Circuit diagram
  - Tables
  - Graphs

Use labels and captions for figures and tables

## INTRODUCTION TO ADE LAB

### ELECTRONIC COMPONENTS

#### 1. RESISTORS:

A resistor is a component of an electrical circuit that resists the flow of electrical current. A resistor has two terminals across which electricity must pass, and is designed to drop the voltage of the current as it flows from one terminal to the next. A resistor is primarily used To create and maintain a known safe current within an electrical component. Resistance is measured in ohms, after Ohm's law. A 1000 Ohm resistor is typically shown as 1K-Ohm (kilo Ohm), and 1000 K-Ohms is written as 1M-Ohm (mega ohm).

Resistor Colour Code:



Figure 1: A Resistor

Colour	1 <sup>st</sup> Band	2 <sup>nd</sup> Band	3 <sup>rd</sup> Band (multiplier)	4 <sup>th</sup> Band (tolerance)
Black	0	0	$\times 10^0$	
Brown	1	1	$\times 10^1$	$\pm 1\%$
Red	2	2	$\times 10^2$	$\pm 2\%$
Orange	3	3	$\times 10^3$	
Yellow	4	4	$\times 10^4$	
Green	5	5	$\times 10^5$	$\pm 0.5\%$
Blue	6	6	$\times 10^6$	$\pm 0.25\%$
Violet	7	7	$\times 10^7$	$\pm 0.1\%$
Gray	8	8	$\times 10^8$	$\pm 0.05\%$
White	9	9	$\times 10^9$	
Gold			$\times 0.1$	$\pm 5\%$
Silver			$\times 0.01$	$\pm 10\%$

#### 2. Capacitors:

The capacitor's capacitance (C) is a measure of the amount of charge (Q) stored on each plate for a given potential difference or voltage (V) which appears across the plates. In SI units, a capacitor has a capacitance is measured in farad (F).

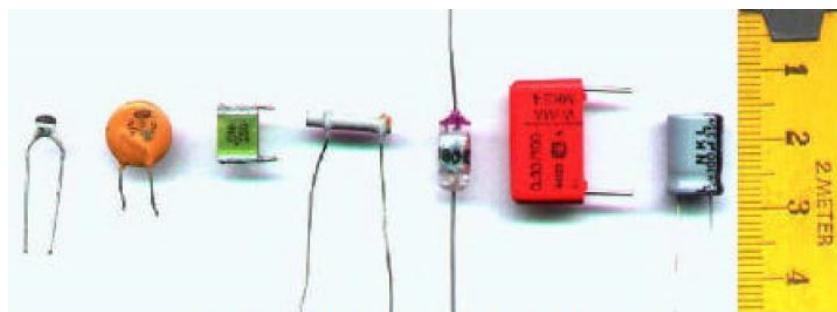
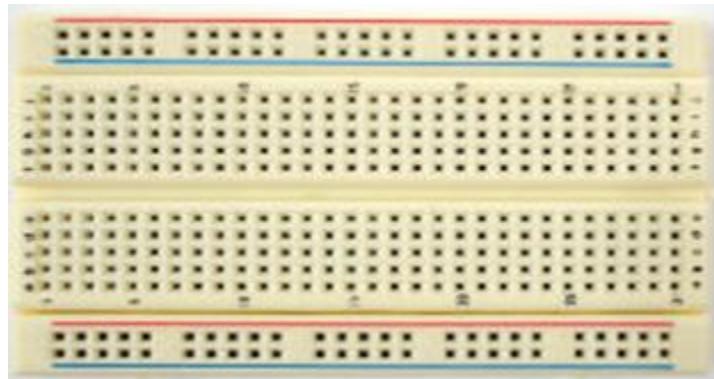


Figure 2: different capacitor models

#### 3. Breadboard:

A breadboard is a material or a device used to build a prototype of an electronic circuit. The breadboard has many strips of metal (copper usually) which run underneath the board. The metal strips are laid out as shown These strips connect the holes on the top of the board. This makes it easy to connect components together to build circuits. The legs of components are placed in the holes (the sockets). The holes are made so that they will hold the component in

place. Each hole is connected to one of the metal strips running underneath the board. The long top and bottom row of holes are usually used for power supply connections.



### DIGITAL COMPONENTS (BASIC GATES)

#### **NOT GATE:**

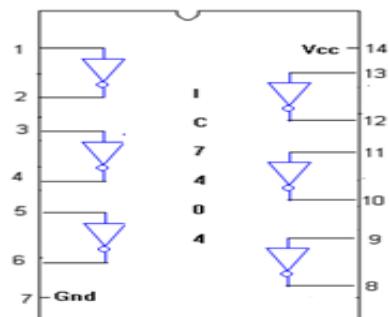
##### SYMBOL:



##### **TRUTH TABLE :**

A	$\bar{A}$
0	1
1	0

##### PIN DIAGRAM:



#### X-OR GATE:

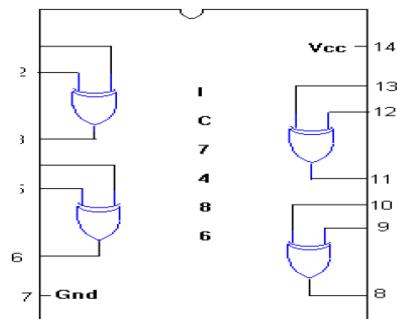
##### SYMBOL :



##### **TRUTH TABLE :**

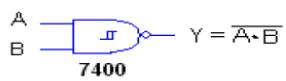
A	B	$\bar{A}B + A\bar{B}$
0	0	0
0	1	1
1	0	1
1	1	0

##### PIN DIAGRAM :



2-INPUT NAND GATE:

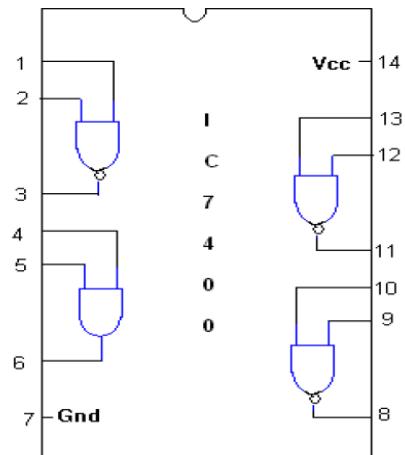
SYMBOL:



TRUTH TABLE

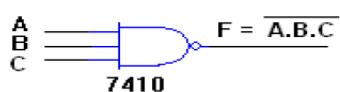
A	B	$\overline{A \cdot B}$
0	0	1
0	1	1
1	0	1
1	1	0

PIN DIAGRAM:



3-INPUT NAND GATE :

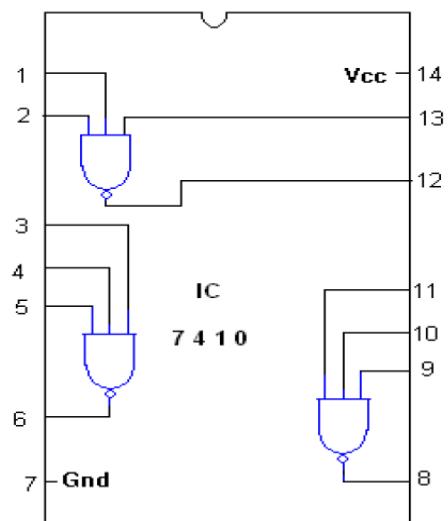
SYMBOL :



TRUTH TABLE

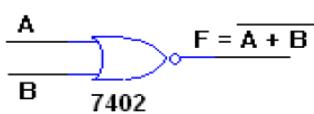
A	B	C	$\overline{ABC}$
0	0	0	1
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	0

PIN DIAGRAM:

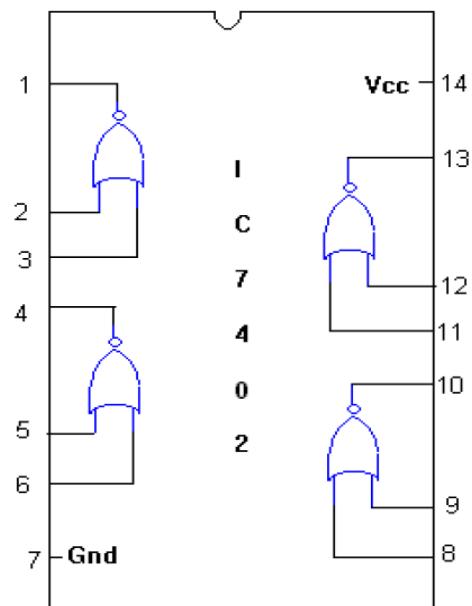


## NOR GATE:

SYMBOL :



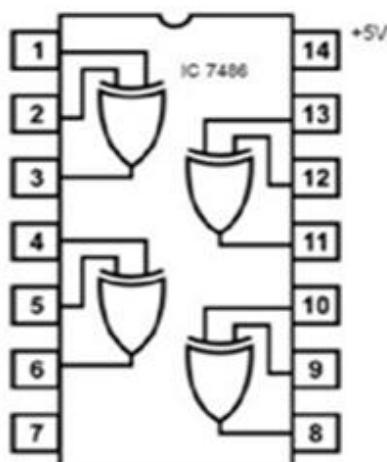
PIN DIAGRAM :



TRUTH TABLE

A	B	$\overline{A+B}$
0	0	1
0	1	1
1	0	1
1	1	0

IC 7486 Pin Diagram



TRUTH TABLE OF EX-OR GATE

Logic inputs		Logic output
A	B	$Y=A \oplus B$
0	0	0
0	1	1
1	0	1
1	1	0

## **Experiment -1** **SCHMITT TRIGGER**

**1.a. AIM :** Design and construct a Schmitt trigger circuit using op-amp for the given UTP and LTP values and demonstrate its working..

**COMPONENTS REQUIRED :** IC  $\mu$ A 741, Resistor of  $10K\Omega$ ,  $90K\Omega$ , DC regulated power supply, Signal generator, CRO

### **DESIGN 1 :**

From theory of Schmitt trigger circuit using op-amp, we have the trip points,

$$UTP = \frac{R_1 V_{ref}}{R_1 + R_2} + \frac{R_2 V_{sat}}{R_1 + R_2} \text{ where } V_{sat} \text{ is the positive saturation of the opamp} = 90\% \text{ of } V_{cc}$$

$$\& LTP = \frac{R_1 V_{ref}}{R_1 + R_2} - \frac{R_2 V_{sat}}{R_1 + R_2}$$

Hence given the LTP & UTP values to find the  $R_1$ ,  $R_2$  &  $V_{ref}$  values, the following design is used.

$$UTP + LTP = \frac{2R_1 V_{ref}}{R_1 + R_2} \quad \dots \dots \dots (1)$$

$$UTP - LTP = \frac{2R_1 V_{sat}}{R_1 + R_2} \quad \dots \dots \dots (2)$$

Let  $V_{sat} = 10V$ ,  $UTP = 4V$  &  $LTP = 2V$ , then equation (2) yields  $R_1 = 9R_2$

Let  $R_2 = 10K\Omega$ , then  $R_1 = 90K\Omega$

$$\text{From equation (1) we have } V_{ref} = \frac{(UTP + LTP)(R_1 + R_2)}{2R_1} = 3.33V$$

### **DESIGN 2 :**

1) Let given  $UTP = 2.5 V$ ,  $LTP = 1V$ . Assume  $V_{sat} = 12 V$

$$UTP = \frac{R_1 V_{ref}}{R_1 + R_2} + \frac{R_2 V_{sat}}{R_1 + R_2} \text{ where } V_{sat} \text{ is the positive saturation of the opamp}$$

$$\& LTP = \frac{R_1 V_{ref}}{R_1 + R_2} - \frac{R_2 V_{sat}}{R_1 + R_2}$$

Hence given the LTP & UTP values to find the  $R_1$ ,  $R_2$  &  $V_{ref}$  values, the following design is used.

$$UTP + LTP = \frac{2R_1 V_{ref}}{R_1 + R_2} \quad \dots \dots \dots (1)$$

$$UTP - LTP = \frac{2R_1 V_{sat}}{R_1 + R_2} \quad \dots \dots \dots (2)$$

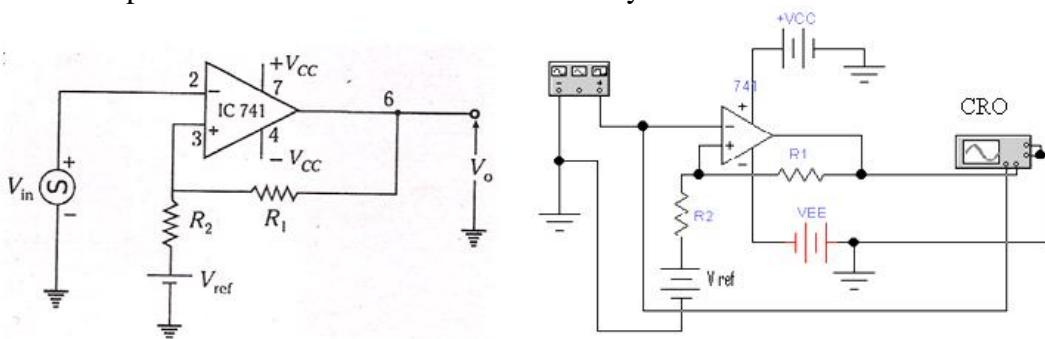
Let  $V_{sat} = 12V$ ,  $UTP = 2.5 V$  &  $LTP = 1V$ , then equation (2) yields  $R_1 = 15R_2$

Let  $R_2 = 1K\Omega$ , then  $R_1 = 15K\Omega$

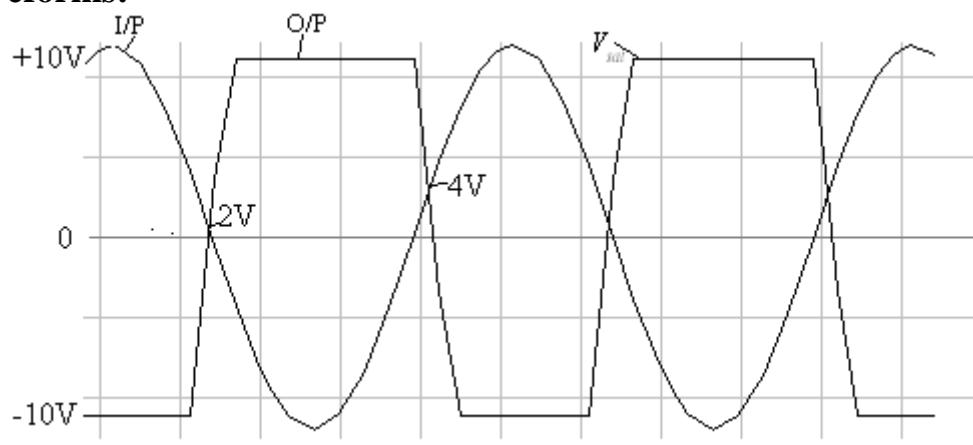
$$\text{From equation (1) we have } V_{ref} = \frac{(UTP + LTP)(R_1 + R_2)}{2R_1} = 1.88V$$

**PROCEDURE :**

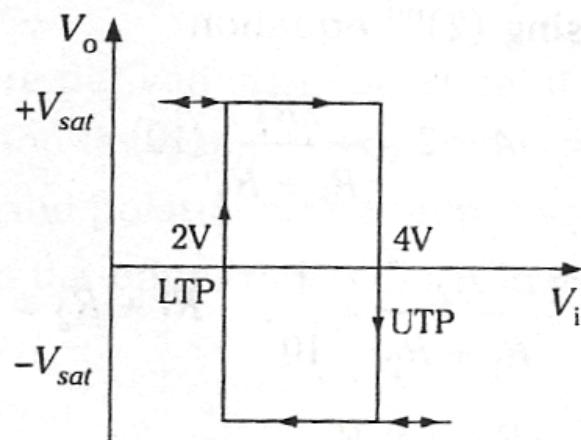
1. Before doing the connections, check all the components using multimeter.
2. Make the connection as shown in circuit diagram.
3. Using a signal generator apply the sinusoidal input waveform of peak-to-peak amplitude of 10V, frequency 1kHz.
4. Keep the CRO in dual mode; apply input ( $V_{in}$ ) signal to the channel 1 and observe the output ( $V_o$ ) on channel 2 which is as shown in the waveform below. Note the amplitude levels from the waveforms.
5. Now keep CRO in X-Y mode and observe the hysteresis curve.



**Fig: Circuit Diagram and actual connections of Schmitt Trigger Circuit**  
**Waveforms:**



CRO in DUAL mode



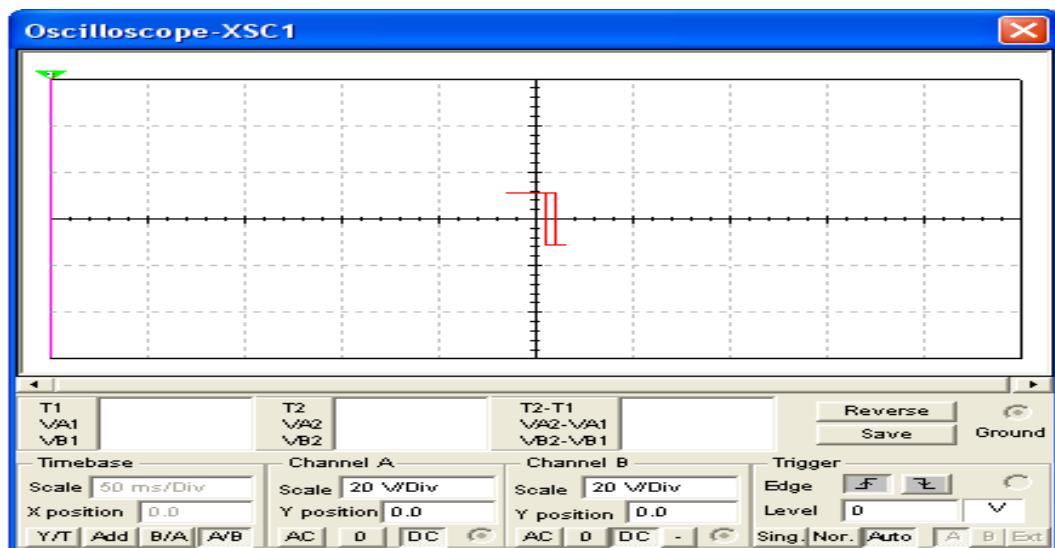
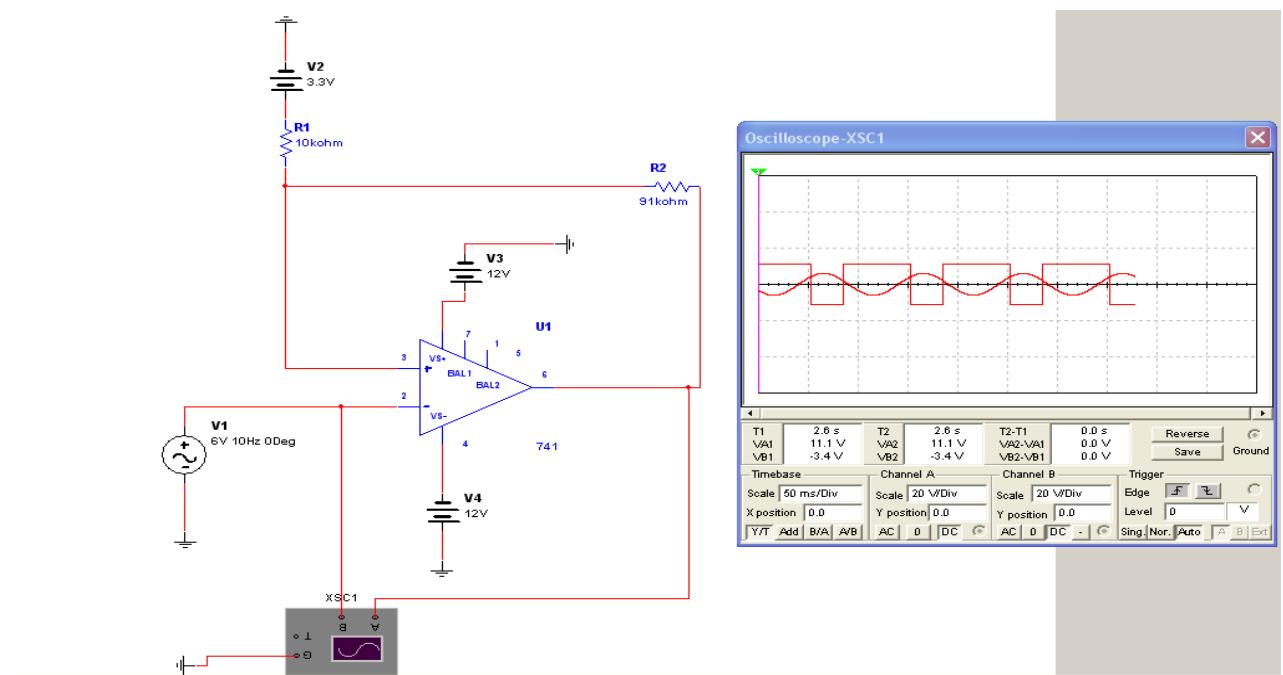
CRO in X-Y mode showing the Hysteresis curve

## THEORY:

Schmitt Trigger converts an irregular shaped waveform to a square wave or pulse. Here, the input voltage triggers the output voltage every time it exceeds certain voltage levels called the upper threshold voltage  $V_{UTP}$  and lower threshold voltage  $V_{LTP}$ . The input voltage is applied to the inverting input. Because the feedback voltage is aiding the input voltage, the feedback is positive. A comparator using positive feedback is usually called a Schmitt Trigger. Schmitt Trigger is used as a squaring circuit, in digital circuitry, amplitude comparator, etc.

**1b. Design and implement a Schmitt trigger using Op-Amp using a simulation package or two sets of UTP and LTP values and demonstrate its working.**

**Schmitt Trigger**



## Experiment -2

### OP-AMP AS A RELAXATION OSCILLATOR

**a. AIM :** Design and construct a rectangular waveform generator (op-amp relaxation oscillator) for a given frequency and demonstrate its working...

#### COMPONENTS REQUIRED:

Op-amp  $\mu$ A 741, Resistor of  $1\text{K}\Omega$ ,  $10\text{K}\Omega$ ,  $20\text{k}\Omega$  Potentiometer, Capacitor of  $0.1\text{ }\mu\text{F}$ , Regulated DC power supply, CRO

#### DESIGN :

The period of the output rectangular wave is given as  $T = 2RC \ln\left(\frac{1+\beta}{1-\beta}\right)$  ----- (1)

Where,  $\beta = \frac{R_1}{R_1 + R_2}$  is the feedback fraction

If  $R_1 = R_2$ , then from equation (1) we have  $T = 2RC \ln(3)$

Another example, if  $R_2=1.16 R_1$ , then  $T = 2RC$  ----- (2)

Example: Design for a frequency of  $1\text{kHz}$  (implies  $T = \frac{1}{f} = \frac{1}{10^3} = 10^{-3} = 1\text{ms}$ )

Use  $R_2=1.16 R_1$ , for equation (2) to be applied.

Let  $R_1 = 10\text{k}\Omega$ , then  $R_2 = 11.6\text{k}\Omega$  (use  $20\text{k}\Omega$  potentiometer as shown in circuit figure)

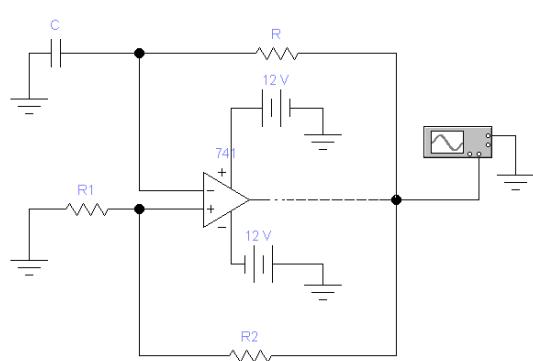
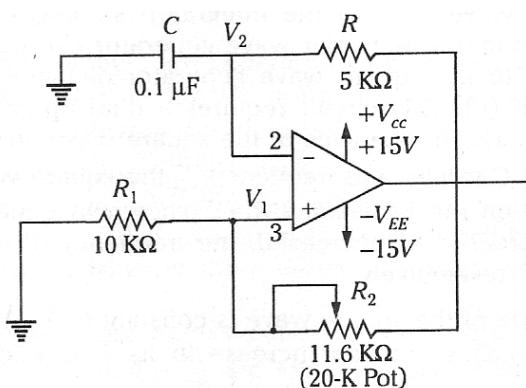
Choose next a value of  $C$  and then calculate value of  $R$  from equation (2).

Let  $C=0.1\mu\text{F}$  (i.e.,  $10^{-7}$ ), then  $R = \frac{T}{2C} = \frac{10^{-3}}{2 \times 10^{-7}} = 5\text{K}\Omega$

The voltage across the capacitor has a peak voltage of  $V_c = \frac{R_1}{R_1 + R_2} V_{sat}$

#### PROCEDURE :

1. Before making the connections check all the components using multimeter.
2. Make the connections as shown in figure and switch on the power supply.
3. Observe the voltage waveform across the capacitor on CRO.
4. Also observe the output waveform on CRO. Measure its amplitude and frequency.



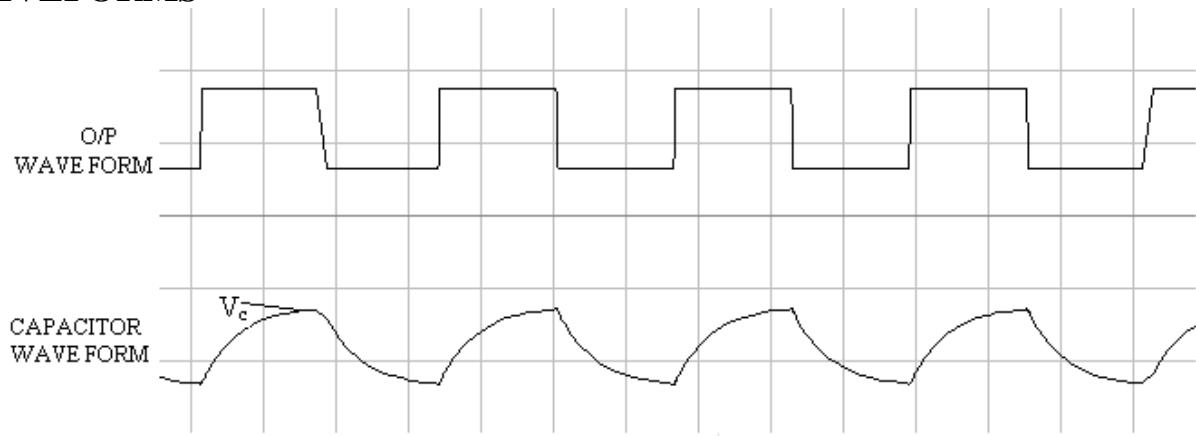
Circuit Diagram & actual connections

#### Values

$C=0.1\mu\text{F}$

$R_1 = 10\text{k}\Omega$ ,  $R_2 = 11.6\text{ k}\Omega$ ,  $R = 4.7\text{k}/5.1\text{k}\Omega$

## WAVEFORMS



### RESULT:

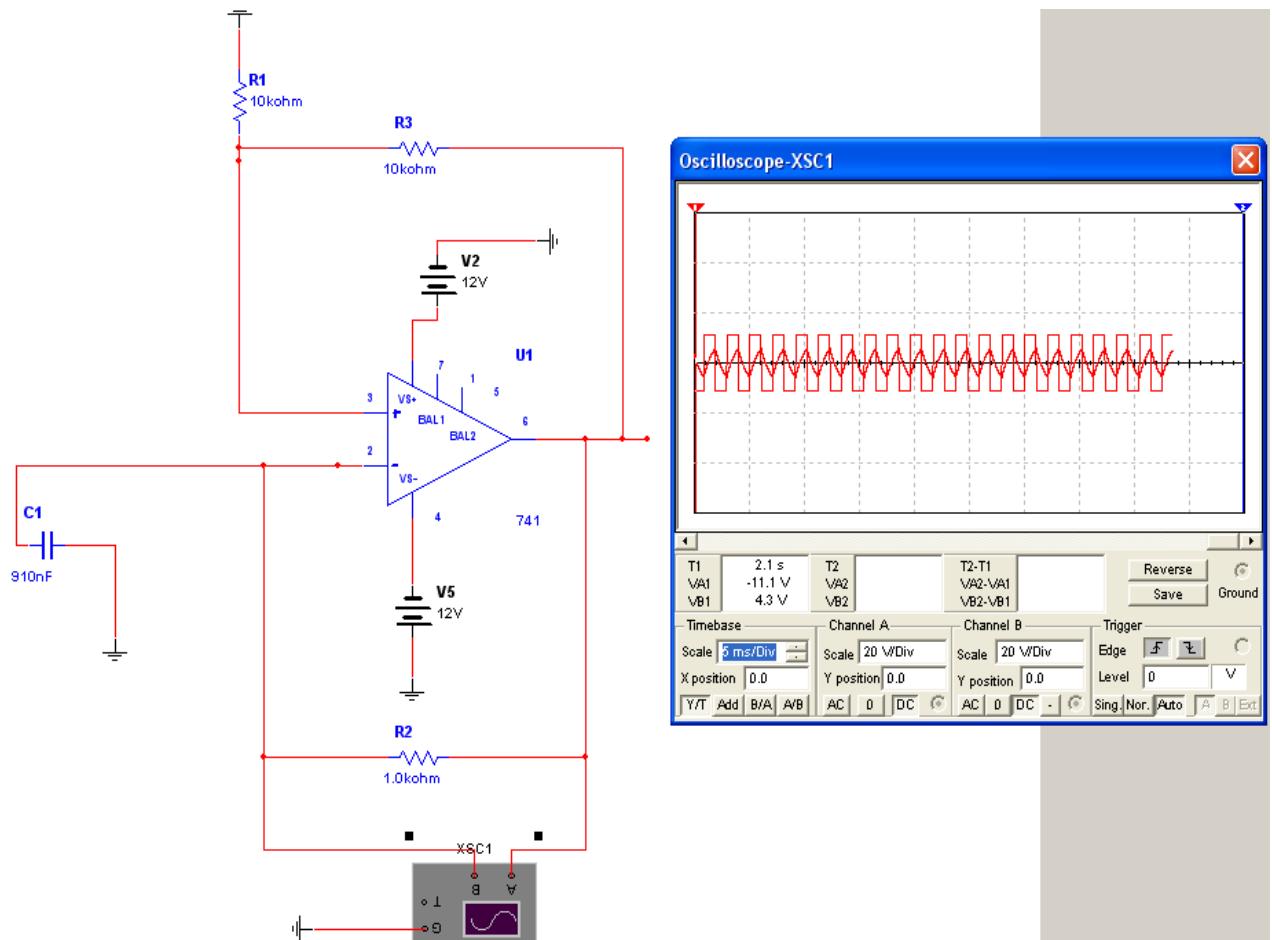
The frequency of the oscillations = .....Hz.

### THEORY:

Op-Amp Relaxation Oscillator is a simple Square wave generator which is also called as a Free running oscillator or Astable multivibrator or Relaxation oscillator. In this figure the op-amp operates in the saturation region. Here, a fraction  $(R_2/(R_1+R_2))$  of output is fed back to the noninverting input terminal. Thus reference voltage is  $(R_2/(R_1+R_2)) V_o$ . And may take values as  $+(R_2/(R_1+R_2)) V_{sat}$  or  $-(R_2/(R_1+R_2)) V_{sat}$ . The output is also fed back to the inverting input terminal after integrating by means of a low-pass RC combination. Thus whenever the voltage at inverting input terminal just exceeds reference voltage, switching takes place resulting in a square wave output.

**b. Design and implement a rectangular waveform generator (Op-Amp relaxation Oscillator) using simulation package and demonstrate the changes in frequency when all resistor values are doubled.**

### Relaxation oscillator



## **Experiment -3**

### **ASTABLE MULTIVIBRATOR USING 555 TIMER**

**AIM :** Design and implement an astable multivibrator using 555 Timer for a given frequency and duty cycle.

**COMPONENTS REQUIRED:** 555 Timer IC, Resistors of  $3.3\text{K}\Omega$ ,  $6.8\text{K}\Omega$ , Capacitors of  $0.1\ \mu\text{F}$ ,  $0.01\ \mu\text{F}$ , Regulated power supply, CRO.

**DESIGN :** Given frequency ( $f$ ) =  $1\text{KHz}$  and duty cycle =  $60\% (=0.6)$

$$\text{The time period } T = 1/f = 1\text{ms} = T_{\text{on}} + T_{\text{off}}$$

Where  $T_{\text{on}}$  is the time the output is high and  $T_{\text{off}}$  is the time the output is low.

From the theory of astable multivibrator using 555 Timer(refer Malvino), we have

$$T_{\text{on}} = 0.693 R_B C \quad \dots\dots(1)$$

$$T_{\text{off}} = 0.693 (R_A + R_B)C \quad \dots\dots(2)$$

$$T = T_{\text{on}} + T_{\text{off}} = 0.693 (R_A + 2 R_B) C$$

Duty cycle =  $T_{\text{on}} / T = 0.6$ . Hence  $T_{\text{on}} = 0.6T = 0.6\text{ms}$  and  $T_{\text{off}} = T - T_{\text{on}} = 0.4\text{ms}$ .

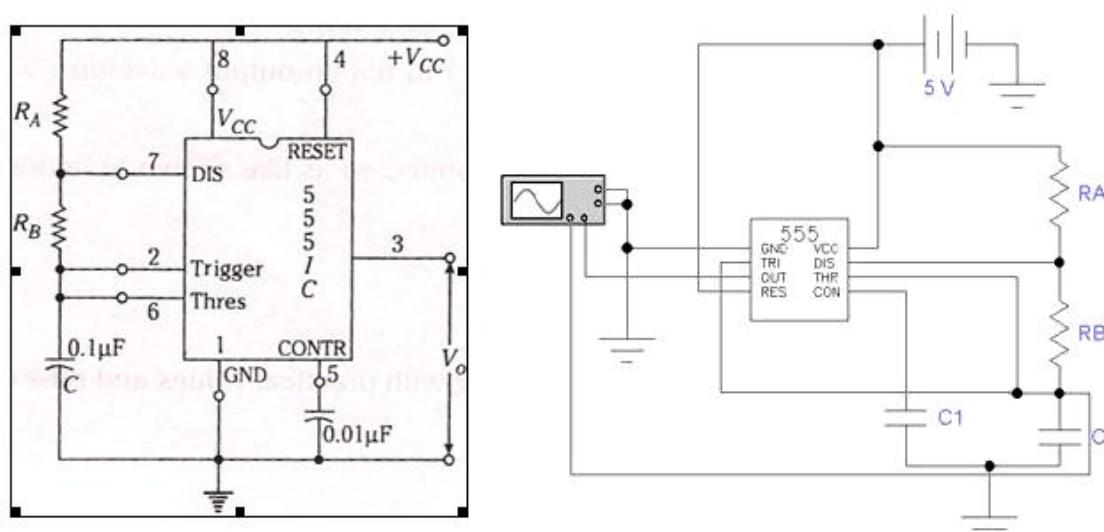
Let  $C=0.1\ \mu\text{F}$  and substituting in the above equations,

$R_B = 5.8\text{K}\Omega$  (from equation 1) and  $R_A = 2.9\text{K}\Omega$  (from equation 2 &  $R_B$  values).

The  $V_{CC}$  determines the upper and lower threshold voltages (observed from the capacitor voltage waveform) as  $V_{UT} = \frac{2}{3}V_{CC}$  &  $V_{LT} = \frac{1}{3}V_{CC}$ .

Note: The duty cycle determined by  $R_A$  &  $R_B$  can vary only between 50 & 100%. If  $R_A$  is much smaller than  $R_B$ , the duty cycle approaches 50%.

Example 2: frequency =  $1\text{kHz}$  and duty cycle =  $75\%$ ,  $R_A = 7.2\text{k}\Omega$  &  $R_B = 3.6\text{k}\Omega$ , choose  $R_A = 6.8\text{k}\Omega$  and  $R_B = 3.3\text{k}\Omega$ .



Circuit Diagram and actual connections

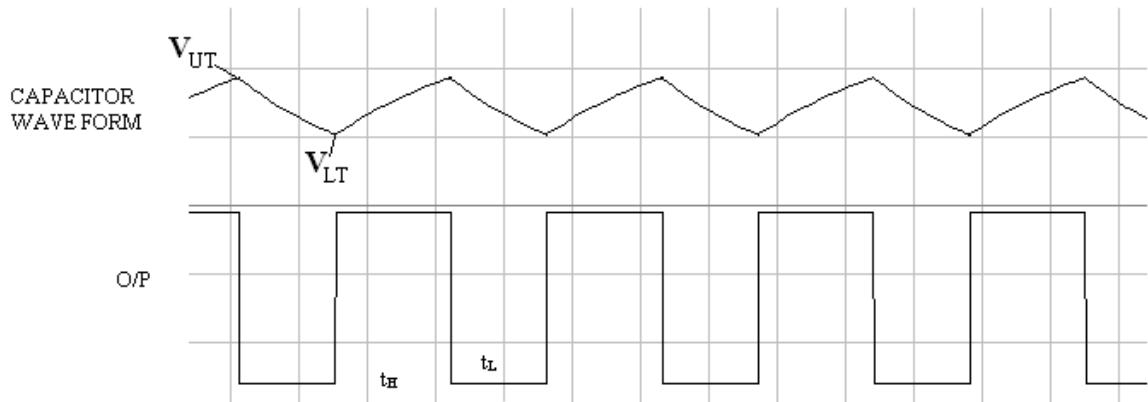
### PROCEDURE :

1. Before making the connections, check the components using multimeter.
2. Make the connections as shown in figure and switch on the power supply.
3. Observe the capacitor voltage waveform at 6<sup>th</sup> pin of 555 timer on CRO.
4. Observe the output waveform at 3<sup>rd</sup> pin of 555 timer on CRO (shown below).
5. Note down the amplitude levels, time period and hence calculate duty cycle.

### RESULT:

The frequency of the oscillations = .....Hz.

### WAVEFORMS



### THEORY:

Multivibrator is a form of oscillator, which has a non-sinusoidal output. The output waveform is rectangular. The multivibrators are classified as: **Astable or free running multivibrator**: It alternates automatically between two states (low and high for a rectangular output) and remains in each state for a time dependent upon the circuit constants. It is just an oscillator as it requires no external pulse for its operation. **Monostable or one shot multivibrator**: It has one stable state and one quasi stable. The application of an input pulse triggers the circuit time constants. After a period of time determined by the time constant, the circuit returns to its initial stable state. The process is repeated upon the application of each trigger pulse. **Bistable Multivibrators**: It has both stable states. It requires the application of an external triggering pulse to change the output from one state to other. After the output has changed its state, it remains in that state until the application of next trigger pulse. Flip flop is an example.

## Experiment -4

### ADDER SUBTRACTOR

**Aim:** Design and implement Half adder, Full Adder, Half Subtractor, Full Subtractor using basic gates.

**Components Used:** IC 7404, IC 7408, IC 7432, IC 7486 , Patch Chords, Power chords, Trainer kit.

#### **THEORY:**

##### HALF ADDER:

A half adder has two inputs for the two bits to be added and two outputs one from the sum ‘S’ and other from the carry ‘c’ into the higher adder position. Above circuit is called as a carry signal from the addition of the less significant bits sum from the X-OR Gate the carry out from the AND gate.

##### FULL ADDER:

A full adder is a combinational circuit that forms the arithmetic sum of input it consists of three inputs and two outputs. A full adder is useful to add three bits at a time but a half adder cannot do so. In full adder sum output will be taken from X-OR Gate, carry output will be taken from OR Gate.

##### HALF SUBTRACTOR:

The half subtractor is constructed using X-OR and AND Gate. The half subtractor has two input and two outputs. The outputs are difference and borrow. The difference can be applied using X-OR Gate, borrow output can be implemented using an AND Gate and an inverter.

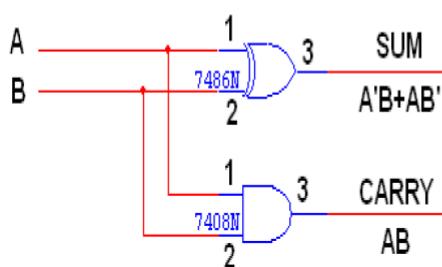
##### FULL SUBTRACTOR:

The full subtractor is a combination of X-OR, AND, OR, NOT Gates. In a full subtractor the logic circuit should have three inputs and two outputs. The two half subtractor put together gives a full subtractor .The first half subtractor will be C and A B. The output will be difference output of full subtractor. The expression AB assembles the borrow output of the half subtractor and the second term is the inverted difference output of first X-OR.

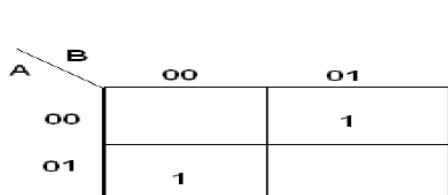
#### HALF ADDER

##### **LOGIC DIAGRAM:**

##### HALF ADDER TRUTH TABLE:

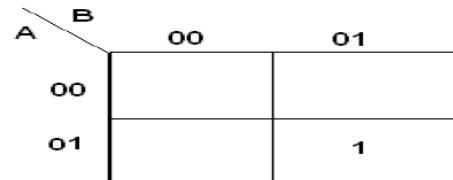


A	B	CARRY	SUM
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0



**K-Map for SUM:**

$$\text{SUM} = A'B + AB'$$



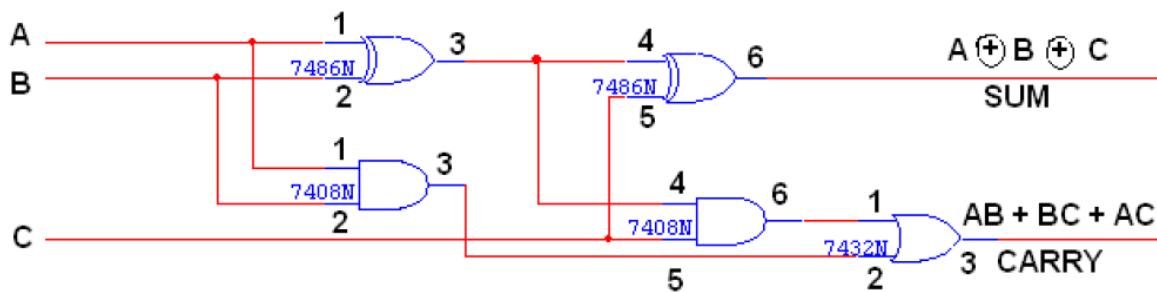
**K-Map for CARRY:**

$$\text{CARRY} = AB$$

### **FULL ADDER**

#### **LOGIC DIAGRAM:**

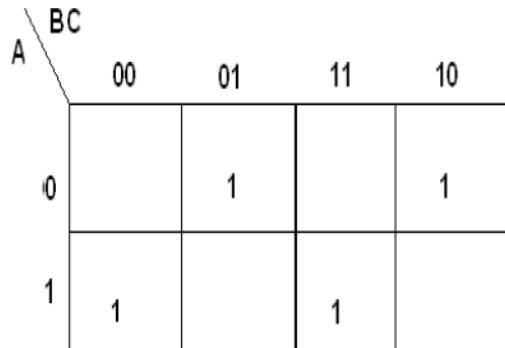
#### **FULL ADDER USING TWO HALF ADDER**



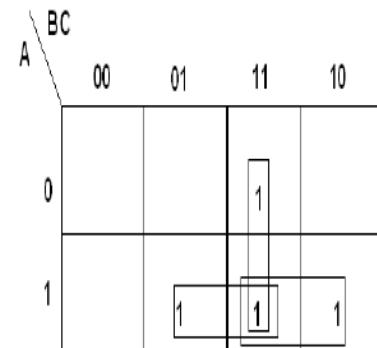
#### **TRUTH TABLE:**

A	B	C	CARRY	SUM
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

**K-Map for SUM:**



**K-Map for CARRY:**

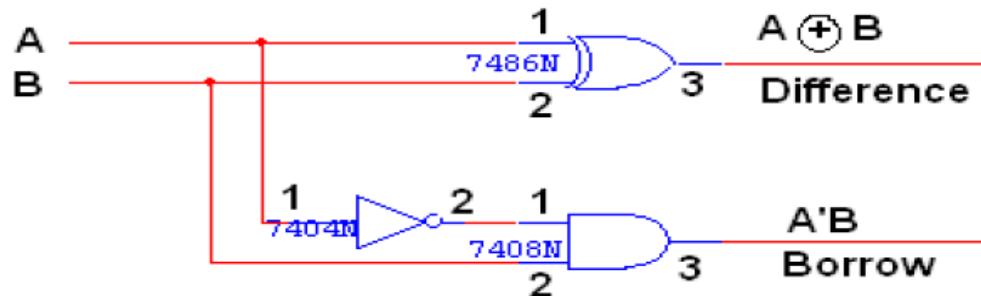


$$\text{SUM} = A'B'C + A'BC' + ABC' + ABC$$

$$\text{CARRY} = AB + BC + AC$$

### HALF SUBTRACTOR **LOGIC DIAGRAM:**

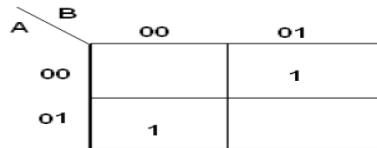
### HALF SUBTRACTOR



### **TRUTH TABLE:**

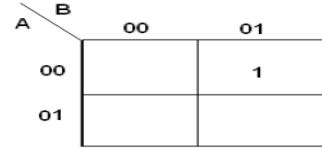
A	B	BORROW	DIFFERENCE
0	0	0	0
0	1	1	1
1	0	0	1
1	1	0	0

**K-Map for DIFFERENCE:**



**DIFFERENCE =  $A'B + AB'$**   
**FULL SUBTRACTOR**

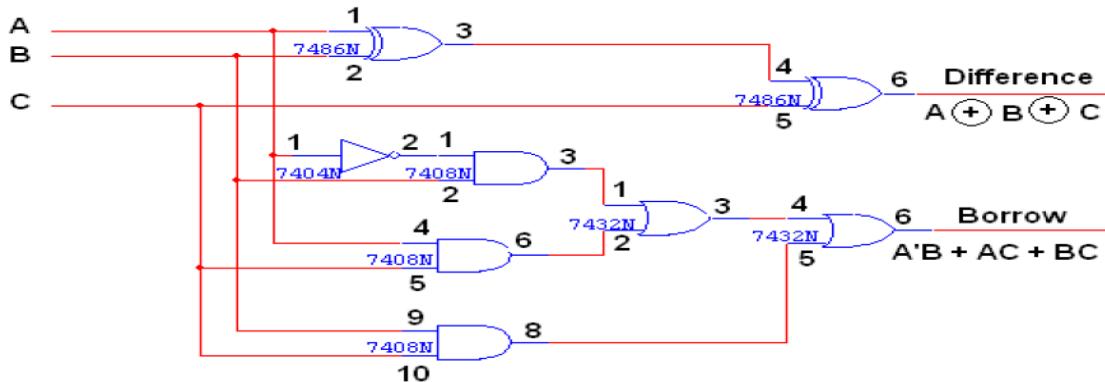
**K-Map for BORROW:**



**BORROW =  $A'B$**

**LOGIC DIAGRAM:**

**FULL SUBTRACTOR**



**TRUTH TABLE**

A	B	C	BORROW	DIFFERENCE
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	1	0
1	0	0	0	1
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1

		BC			
		00	01	11	10
A	0		1		1
	1	1		1	

		BC			
		00	01	11	10
A	0	1		1	
	1			1	

$$\text{Difference} = A'B'C + A'BC' + AB'C' + ABC \quad \text{Borrow} = A'B + BC + A'C$$

**PROCEDURE:**

- (i) Connections are given as per circuit diagram.
- (ii) Logical inputs are given as per circuit diagram.
- (iii) Observe the output and verify the truth table.

**RESULT:** Truth Tables are verified.

## Experiment -5 Multiplexer

a) Given a four variable expression, simplify using Entered Variable Map (EVM) and realize the simplified logic using 8:1 MUX.

a) E.g.,

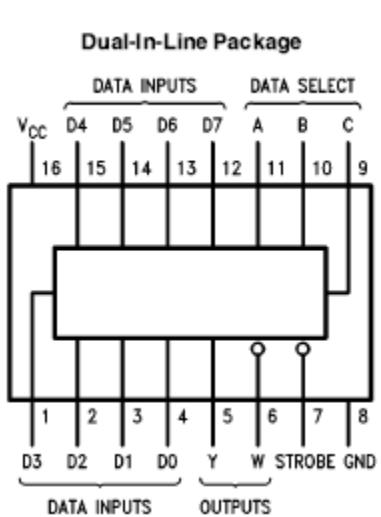
Simplify the function using MEV technique

$$f(a,b,c,d) = \sum m(2,3,4,5,13,15) + d(\bar{8},\bar{9},\bar{10},\bar{11})$$

Decimal	LSB	f	MEV map entry
0 } <sub>0</sub>	0000	0	0-----Do
1	0001	0	
1 } <sub>2</sub>	0010	1	1-----D1
9	0011	1	
2 } <sub>4</sub>	0100	1	1-----D2
5	0101	1	
3 } <sub>6</sub>	0110	0	0-----D3
7	0111	0	
4 } <sub>8</sub>	1000	X	X-----D4
9	1001	X	
5 } <sub>10</sub>	1010	X	X-----D5
11	1011	X	
6 } <sub>12</sub>	1100	0	d----D6
13	1101	1	
7 } <sub>14</sub>	1110	0	d----D7
15	1111	1	

**Components Used:** IC 74LS151, Patch Chords, Power chords, Trainer kit.

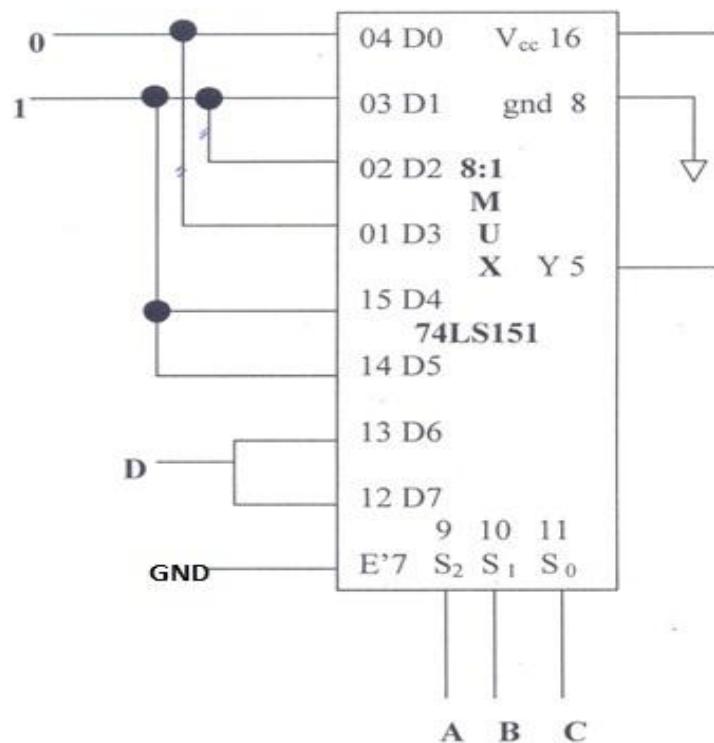
**Pin Diagram of Ics Used:**



Rules for entering values in a MEV Map:

Rule No.	MEV	Function	Entry in MEV Map	Comments
1	0	0	0	If function equals 0 for both values of MEV, enter 0 in appropriate cell of MEV Map
	1	0		
2	0	1	1	If function equals 1 for both values of MEV, enter 1 in appropriate cell of MEV Map
	1	1		
3	0	0	MEV	If function equals MEV for both values of MEV, enter MEV in appropriate cell of MEV Map
	1	1		
4	0	1	$\overline{\text{MEV}}$	If function is complement of MEV, enter $\overline{\text{MEV}}$ in appropriate cell of MEV Map
	1	0		
5	0	x	x	If function equals x for both values of MEV, enter x in appropriate cell of MEV Map
	1	x		
6	0	x	0	If $f = x$ for $\text{MEV} = 0$ and $f = 0$ for $\text{MEV} = 1$ , enter 0 in appropriate cell of MEV Map
	1	0		
7	0	0	0	If $f = 0$ for $\text{MEV} = 0$ and $f = x$ for $\text{MEV} = 1$ , enter 0 in appropriate cell of MEV Map
	1	x		
8	0	x	1	If $f = x$ for $\text{MEV} = 0$ and $f = 1$ for $\text{MEV} = 1$ , enter 1 in appropriate cell of MEV Map
	1	1		
9	0	1	1	If $f = 1$ for $\text{MEV} = 0$ and $f = x$ for $\text{MEV} = 1$ , enter 1 in appropriate cell of MEV Map
	1	x		

**Circuit Diagram:**



**Procedure:**

- (1) Verify all components and patch chords whether they are in good condition or not.
- (2) Make connection as shown in the circuit diagram.
- (3) Give supply to the trainer kit.
- (4) Provide input data to circuit via switches.
- (5) Verify truth table sequence and observe outputs.

**RESULT:** Truth Tables are verified.

**7b. Design and develop the verilog/VHDL code for 8:1 MUX. Simulate and verify its working.**

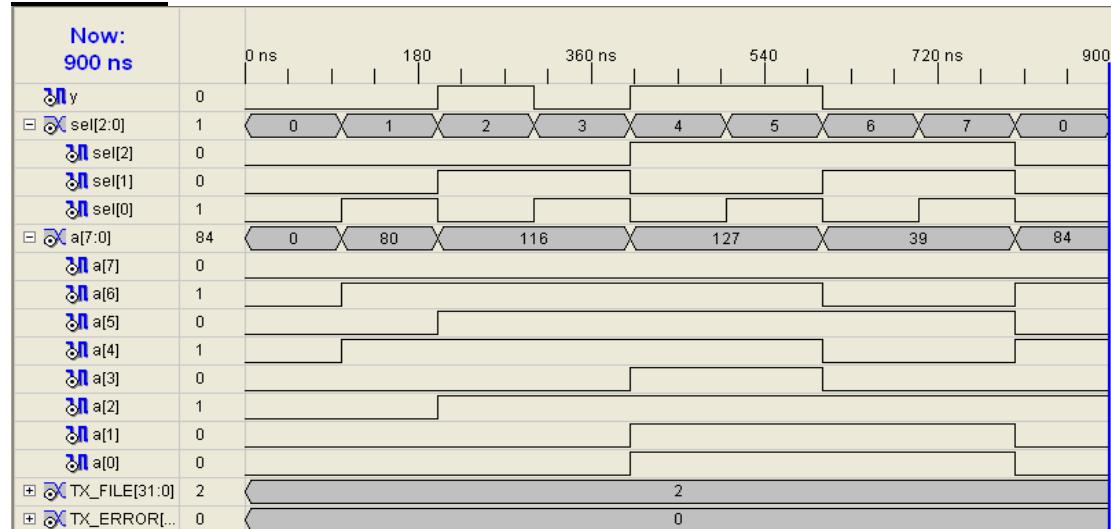
**Program:-**

```
Module mux(sel,a,y);
Input[2:0] sel;
Input[7:0] A;
Output Y;
Reg Y;
Always @(sel,A)
Begin
    Case (sel)
        3' b000:y=A[0];
        3' b001:y=A[1];
        3' b010:y=A[2];
        3' b011:y=A[3];
        3' b100:y=A[4];
        3' b101:y=A[5];
        3' b110:y=A[6];
        3' b111:y=A[7];
    End case
End
End Module
```

**TruthTable**

INPUTS			OUTPUTS
SEL (2)	SEL (1)	SEL (0)	Zout
0	0	0	I(0)
0	0	1	I(1)
0	1	0	I(2)
0	1	1	I(3)
1	0	0	I(4)
1	0	1	I(5)
0	1	1	I(6)
1	1	1	I(7)

**Waveform:**



## Experiment -6

**Aim:** Design and implement code converter I)Binary to Gray (II) Gray to Binary Code using basic gates.

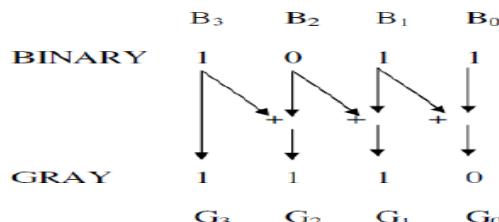
APPARATUS REQUIRED: - IC Trainer Kit, patch chords, IC 7486, etc

### **THEORY:**

A code converter is a circuit that makes the two systems compatible even though each uses a different binary code. Each one of the four maps represents one of the four outputs of the circuit as a function of the four input variables. A two-level logic diagram may be obtained directly from the Boolean expressions derived by the maps. The input variable are designated as B<sub>3</sub>, B<sub>2</sub>, B<sub>1</sub>, B<sub>0</sub> and the output variables are designated as C<sub>3</sub>, C<sub>2</sub>, C<sub>1</sub>, Co. from the truth table, combinational circuit is designed. The Boolean functions are obtained from K-Map for each output variable.

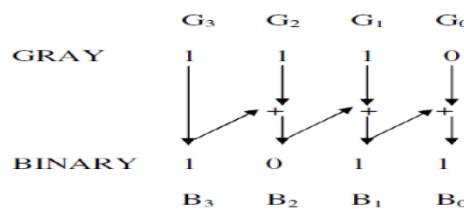
### **BINARY TO GRAY CODE CONVERSION:**

Steps: The example shows the steps involved in conversion of a binary code to its gray code  
 Binary code taken for the example is 1011. In the conversion process the most significant bit (MSB) of the binary code is taken as the MSB of the Gray code. The bit positions G<sub>2</sub>, G<sub>1</sub> and G<sub>0</sub> is obtained by adding (B<sub>3</sub>, B<sub>2</sub>), (B<sub>2</sub>, B<sub>1</sub>) and (B<sub>1</sub>, B<sub>0</sub>) respectively, ignoring the carry generated.  
 From the K-Map simplification



### **GRAY TO BINARY CODE CONVERSION STEPS:**

The example shows the steps involved in conversion of a Gray code to binary code. Gray code taken for the example is 1110. In the conversion process the most significant bit (MSB) of the Gray code is taken as the MSB of the binary code. The bit positions B<sub>2</sub>, B<sub>1</sub> and B<sub>0</sub> is obtained by adding (G<sub>3</sub>, G<sub>2</sub>), (G<sub>2</sub>, G<sub>1</sub>) and (G<sub>1</sub>, G<sub>0</sub>) respectively, ignoring the carry generated.



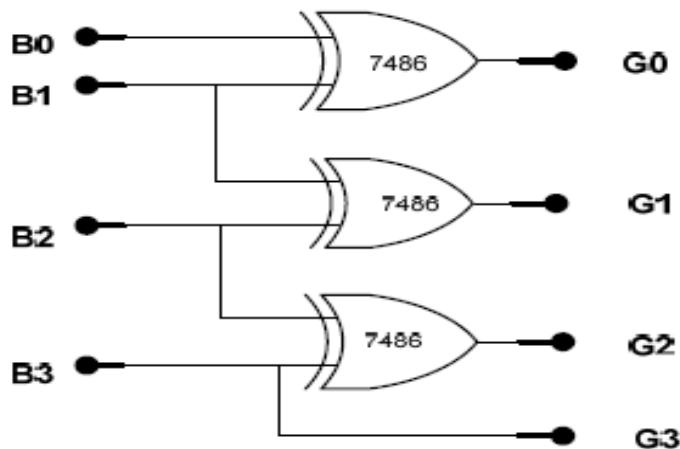
## Binary to Gray Conversion:

### Binary to Gray:

Truth Table:

B3	B2	B1	B0	G3	G2	G1	G0
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	1	1	1	0	0	0

### Circuit diagram using EX-OR Gates:



For binary to Gray code conversion the following Boolean expressions are obtained,

$$G_3 = B_3$$

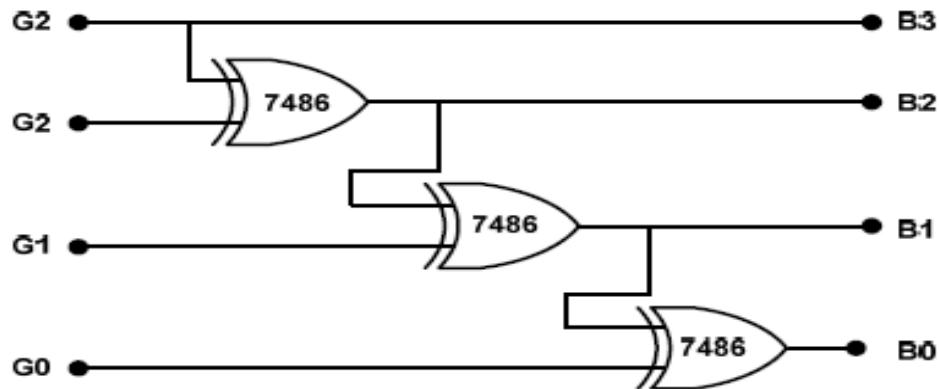
$$G_2 = B_3 \oplus B_2$$

$$G_1 = B_2 \oplus B_1$$

$$G_0 = B_1 \oplus B_0$$

**Gray to Binary Conversion:****Truth-table:**

G3	G2	G1	G0	B3	B2	B1	B0
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	1
0	0	1	1	0	0	1	0
0	0	1	0	0	0	1	1
0	1	1	0	0	1	0	0
0	1	1	1	0	1	0	1
0	1	0	1	0	1	1	0
0	1	0	0	0	1	1	1
1	1	0	0	1	0	0	0
1	1	0	1	1	0	1	1
1	1	1	0	1	0	1	1
1	0	1	0	1	1	0	0
1	0	1	1	1	1	0	1
1	0	0	1	1	1	1	0
1	1	0	1	0	1	1	1
1	0	0	0	1	1	1	1

**Circuit Diagram using EX-OR Gates**

From the K-Map simplification for Gray code to binary code conversion the following Boolean expressions are obtained,

$$B_3 = G_3$$

$$B_2 = G_3 \oplus G_2$$

$$B_1 = G_3 \oplus G_2 \oplus G_1$$

$$B_0 = G_3 \oplus G_2 \oplus G_1 \oplus G_0$$

]

**PROCEDURE:**

- (i) Connections were given as per circuit diagram.
- (ii) Logical inputs were given as per truth table
- (iii) Observe the logical output and verify with the truth tables.

**RESULT:**

Binary to Gray and Gray to Binary converters are designed, constructed using logic Gates and their truth table was verified.

## Experiment -7

**Aim:** Design and verify the Truth Table of 3-bit Parity Generator and 4-bit Parity Checker using basic Logic Gates with an even parity bit.

### **Theory:**

- Parity Generator: It is combinational circuit that accepts an n-1 bit stream data and generates the additional bit that is to be transmitted with the bit stream. This additional or extra bit is termed as a parity bit.  
In even parity bit scheme, the parity bit is ‘0’ if there are even number of 1s in the data stream and the parity bit is ‘1’ if there are odd number of 1s in the data stream.  
In odd parity bit scheme, the parity bit is ‘1’ if there are even number of 1s in the data stream and the parity bit is ‘0’ if there are odd number of 1s in the data stream. Let us discuss both even and odd parity generators.
- Parity Check: It is a logic circuit that checks for possible errors in the transmission. This circuit can be an even parity checker or odd parity checker depending on the type of parity generated at the transmission end. When this circuit is used as even parity checker, the number of input bits must always be even.

When a parity error occurs, the ‘sum even’ output goes low and ‘sum odd’ output goes high. If this logic circuit is used as an odd parity checker, the number of input bits should be odd, but if an error occurs the ‘sum odd’ output goes low and ‘sum even’ output goes high.

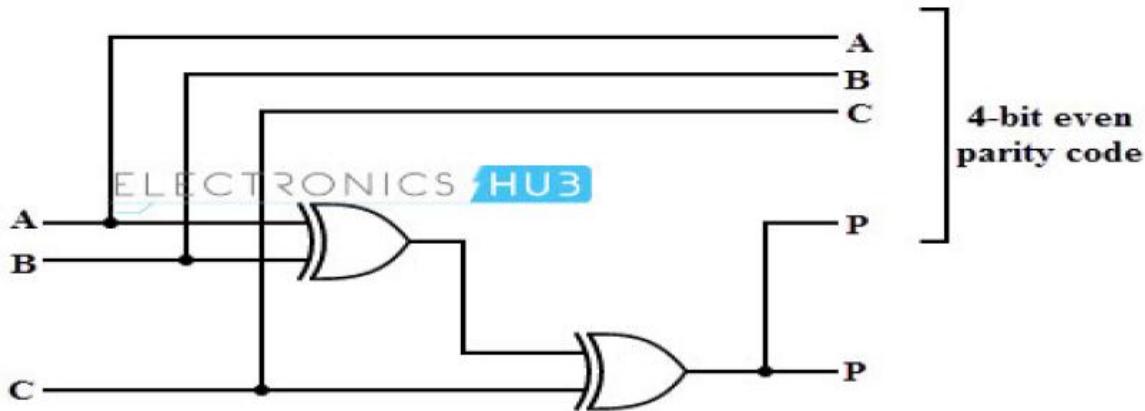
<b>3-bit message</b>			<b>Even parity bit generator (P)</b>
<b>A</b>	<b>B</b>	<b>C</b>	<b>Y</b>
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1

Even Parity Generator Truth Table

From the above truth table, the simplified expression of the parity bit can be written as

$$\begin{aligned}
 P &= \bar{A} \bar{B} C + \bar{A} B \bar{C} + A \bar{B} \bar{C} + A B C \\
 &= \bar{A} (\bar{B} C + B \bar{C}) + A (\bar{B} \bar{C} + B C) \\
 &= \bar{A} (B \oplus C) + A (\bar{B} \oplus \bar{C}) \\
 P &= A \oplus B \oplus C
 \end{aligned}$$

The above expression can be implemented by using two Ex-OR gates. The logic diagram of even parity generator with two Ex – OR gates are shown below.



The three bit message along with the parity generated by this circuit which is transmitted to the receiving end where parity checker circuit checks whether any error is present or not.

### Odd Parity Checker Logic Circuit:

- It is a logic circuit that checks for possible errors in the transmission. This circuit can be an even parity checker or odd parity checker depending on the type of parity generated at the transmission end. When this circuit is used as even parity checker, the number of input bits must always be even.
- When a parity error occurs, the ‘sum even’ output goes low and ‘sum odd’ output goes high.
- If this logic circuit is used as an odd parity checker, the number of input bits should be odd, but if an error occurs the ‘sum odd’ output goes low and ‘sum even’ output goes high.

### Even Parity Checker:

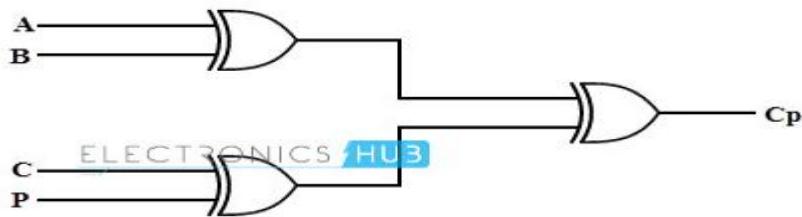
- Consider that three input message along with even parity bit is generated at the transmitting end. These 4 bits are applied as input to the parity checker circuit which checks the possibility of error on the data. Since the data is transmitted with even parity, four bits received at circuit must have an even number of 1s.
- If any error occurs, the received message consists of odd number of 1s. The output of the parity checker is denoted by PEC (parity error check).
- The below table shows the truth table for the even parity checker in which  $PEC = 1$  if the error occurs, i.e., the four bits received have odd number of 1s and  $PEC = 0$  if no error occurs, i.e., if the 4-bit message has even number of 1s.

4-bit received message				Parity error check $C_p$
A	B	C	P	
0	0	0	0	0
0	0	0	1	1
0	0	1	0	1
0	0	1	1	0
0	1	0	0	1
0	1	0	1	0
0	1	1	0	0
0	1	1	1	1
1	0	0	0	1
1	0	0	1	0
1	0	1	0	0
1	0	1	1	1
1	1	0	0	0
1	1	0	1	1
1	1	1	0	1
1	1	1	1	0

Even Parity Checker Truth Table

$$\begin{aligned}
 PEC &= \overline{A} \ \overline{B} (\overline{C} D + \overline{C} \overline{D}) + \overline{A} B (\overline{C} \overline{D} + C D) + A \ B (\overline{C} D + C \overline{D}) + A \ \overline{B} (\overline{C} \overline{D} + C D) \\
 &= \overline{A} \ \overline{B} (C \oplus D) + \overline{A} B (\overline{C} \oplus D) + A \ B (C \oplus D) + A \ \overline{B} (\overline{C} \oplus \overline{D}) \\
 &= (\overline{A} \ \overline{B} + A \ B) (C \oplus D) + (\overline{A} B + A \ \overline{B}) (\overline{C} \oplus \overline{D}) \\
 &= (A \oplus B) \oplus (C \oplus D)
 \end{aligned}$$

The above logic expression for the even parity checker can be implemented by using three Ex-OR gates as shown in figure.



**Even Parity Checker Logic Circuit**

**PROCEDURE:**

- (i) Connections were given as per circuit diagram.
- (ii) Logical inputs were given as per truth table
- (iii) Observe the logical output and verify with the truth tables.

**RESULT:**

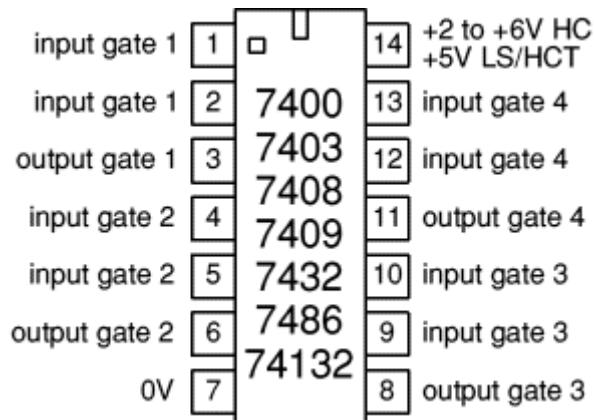
3-bit Parity Generator and 4-bit Parity Checker are designed, constructed using logic Gates and their truth table was verified.

## Experiment -8

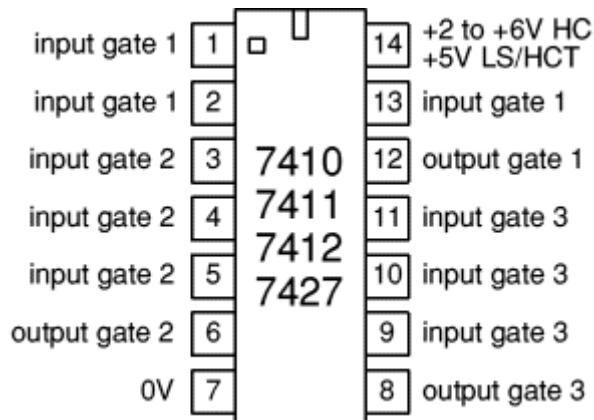
**Aim: 8a. Realize a J-K Master/Slave FF using NAND gates and verify its truth table.**

**Components used:** IC 74LS00, IC 74LS10, IC 74LS20, Power chords, Patch chords, Trainer kit.

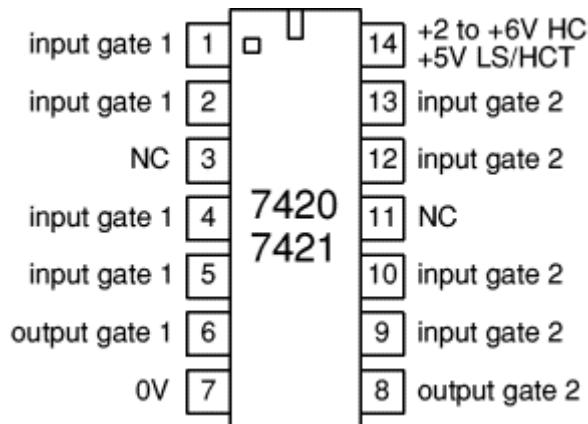
**Pin Details of the ICs: 7400**



**IC-7410**



**IC-7420**



### Theory:

The circuit below shows the solution. To the RS flip-flop we have added two new connections from the Q and Q' outputs back to the original input gates. Remember that a NAND gate may have any number of inputs, so this causes no trouble. To show that we have done this, we change the designations of the logic inputs and of the flip-flop itself. The inputs are now designated J (instead of S) and K (instead of R). The entire circuit is known as a *JK flip-flop*.

In most ways, the JK flip-flop behaves just like the RS flip-flop. The Q and Q' outputs will only change state on the falling edge of the CLK signal, and the J and K inputs will control the future output state pretty much as before. However, there are some important differences.

Since one of the two logic inputs is always disabled according to the output state of the overall flip-flop, the master latch cannot change state back and forth while the CLK input is at logic 1. Instead, the enabled input can change the state of the master latch *once*, after which this latch will not change again. This was not true of the RS flip-flop.

If both the J and K inputs are held at logic 1 and the CLK signal continues to change, the Q and Q' outputs will simply change state with each falling edge of the CLK signal. (The master latch circuit will change state with each *rising* edge of CLK.) We can use this characteristic to advantage in a number of ways. A flip-flop built specifically to operate this way is typically designated as a *T* (for *Toggle*) flip-flop. The lone T input is in fact the CLK input for other types of flip-flops.

The JK flip-flop *must* be edge triggered in this manner. Any level-triggered JK latch circuit will oscillate rapidly if all three inputs are held at logic 1. This is not very useful. For the same reason, the T flip-flop must also be edge triggered. For both types, this is the only way to ensure that the flip-flop will change state only once on any given clock pulse.

Because the behavior of the JK flip-flop is completely predictable under all conditions, this is the preferred type of flip-flop for most logic circuit designs. The RS flip-flop is only used in applications where it can be guaranteed that both R and S cannot be logic 1 at the same time.

At the same time, there are some additional useful configurations of both latches and flip-flops. In the next pages, we will look first at the major configurations and note their

properties. Then we will see how multiple flip-flops or latches can be combined to perform useful functions and operations.

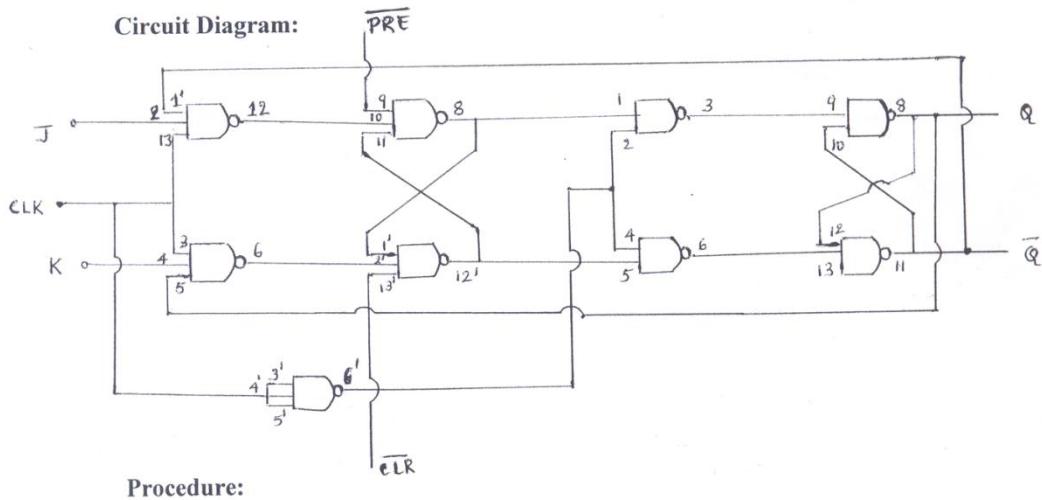
### **Master Slave Flip Flop:**

The control inputs to a clocked flip flop will be making a transition at approximately the same times as triggering edge of the clock input occurs. This can lead to unpredictable triggering.

A JK master flip flop is positive edge triggered, whereas slave is negative edge triggered. Therefore master first responds to J and K inputs and then slave. If  $J=0$  and  $K=1$ , master resets on arrival of positive clock edge. High output of the master drives the K input of the slave. For the trailing edge of the clock pulse the slave is forced to reset. If both the inputs are high, it changes the state or toggles on the arrival of the positive clock edge and the slave toggles on the negative clock edge. The slave does exactly what the master does.

Function Table:

Clk	J	K	Q	---	comment
			$Q_0$	$Q$	
	0	0	$Q_0$	---	No change
	0	1	0	1	Reset
	1	0	1	0	Set
	1	1	$Q_0$	$Q_0$	toggle



- (1) Verify all components and patch chords whether they are in good condition or not.
- (2) Make connection as shown in the circuit diagram.
- (3) Give supply to the trainer kit.
- (4) Provide input data to circuit via switches.
- (5) Verify truth table sequence and observe outputs.

### RESULT:

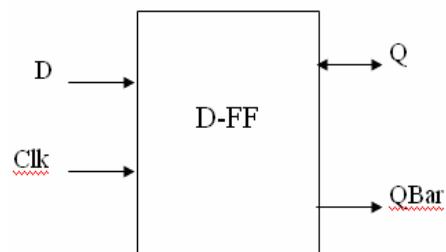
JK Master/Slave circuit is designed, constructed using NAND gates and their truth table was verified.

**8b. Design and develop the verilog/VHDL code for DFF with positive edge triggering. Simulate and verify its working.**

**Program:-**

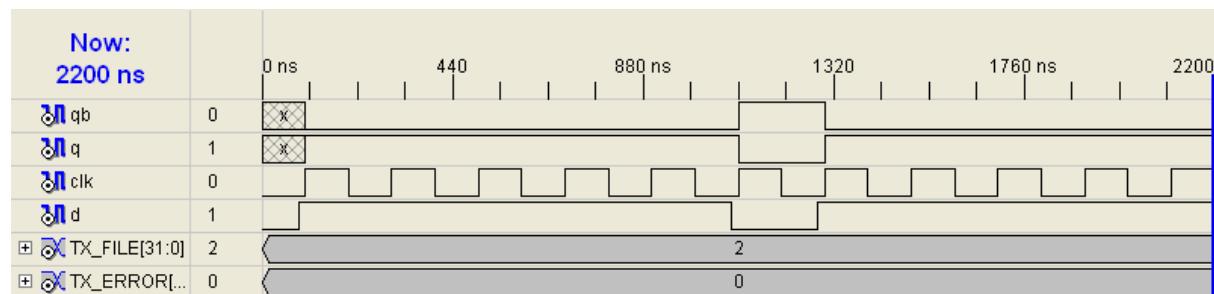
```
Module diff (clk,cl,q,qb)
Input clk,cl;
Output q,qb;
Log q,qb;
always@(posedge clk)
begin
    q=d;
    qb=nq;
end
end Module
```

**Truth Table:**



TruthTable		
INPUT	OUTPUTS	
D	Q	QB
0	0	1
1	1	0

**Waveform:**

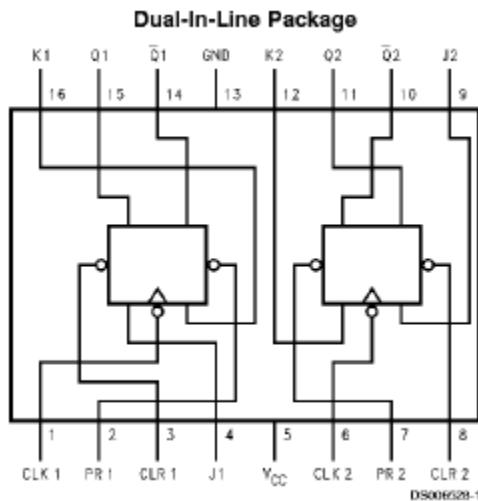


## Experiment -9

**Aim:** 9a) Design and implement a mod n ( $n < 8$ ) synchronous up counter using JK FF IC's and demonstrate its working.

**Components used:** IC 74LS76, IC 74LS08, Patch chords, power chords, and Trainer kit.

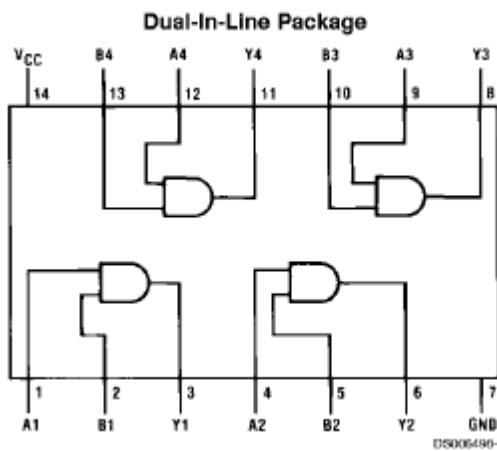
**Pin diagram of 7476**



### Function Table

Inputs					Outputs	
PR	CLR	CLK	J	K	Q	$\bar{Q}$
L	H	X	X	X	H	L
H	L	X	X	X	L	H
L	L	X	X	X	H	H
					(Note 1)	(Note 1)
H	H	↑	L	L	$Q_0$	$\bar{Q}_0$
H	H	↑	H	L	H	L
H	H	↑	L	H	L	H
H	H	↑	H	H	Toggle	

### IC:7408



## Function Table

$$Y = AB$$

Inputs		Output
A	B	Y
L	L	L
L	H	L
H	L	L
H	H	H

H = High Logic Level

L = Low Logic Level

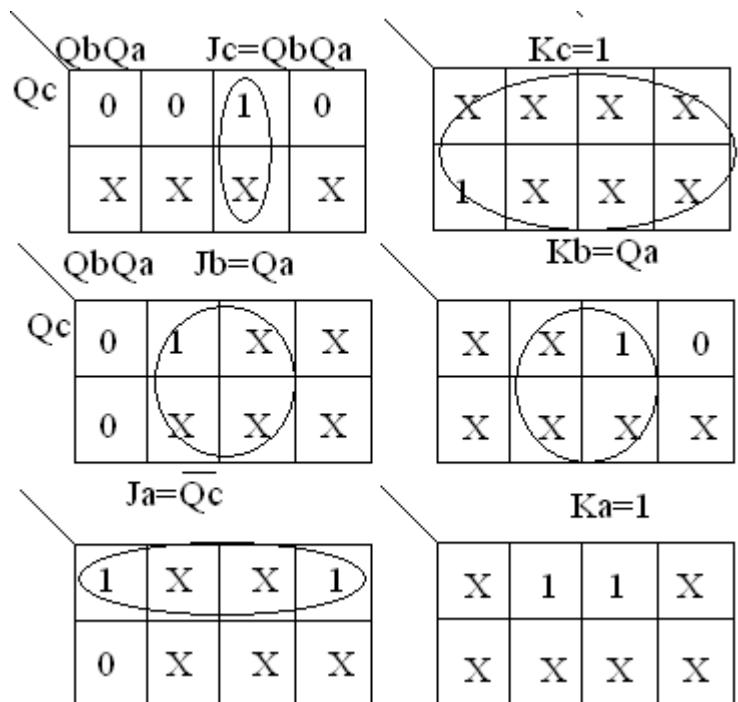
### Theory:

The ripple counter requires a finite amount of time for each flip flop to change state. This problem can be solved by using a synchronous parallel counter where every flip flop is triggered in synchronism with the clock, and all the output which are scheduled to change do so simultaneously.

The counter progresses counting upwards in a natural binary sequence from count 000 to count 100 advancing count with every negative clock transition and get back to 000 after this cycle.

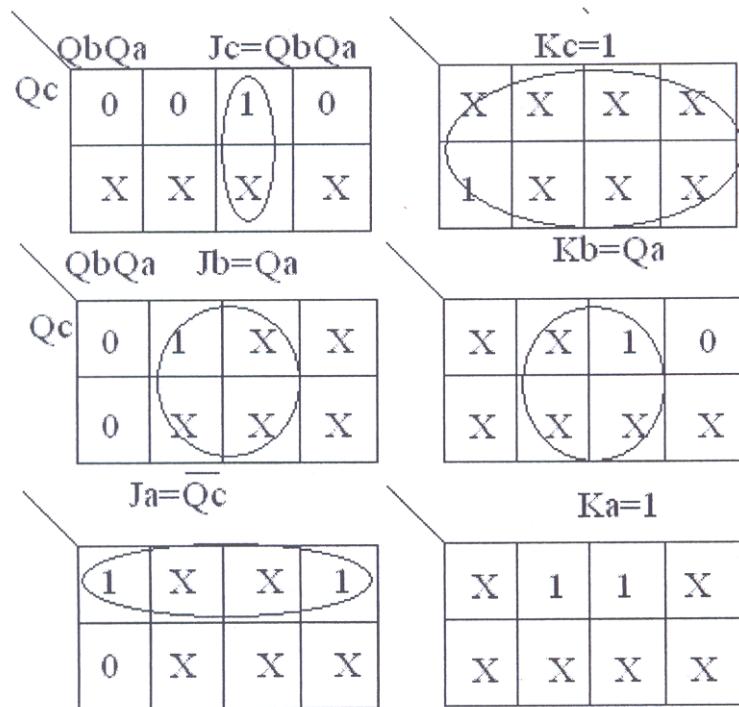
	Q <sub>n</sub>	Q <sub>n+1</sub>	J	K
Circuit	0	0	0	X
	0	1	1	X
	1	0	X	1
	1	1	X	0

Diagram:

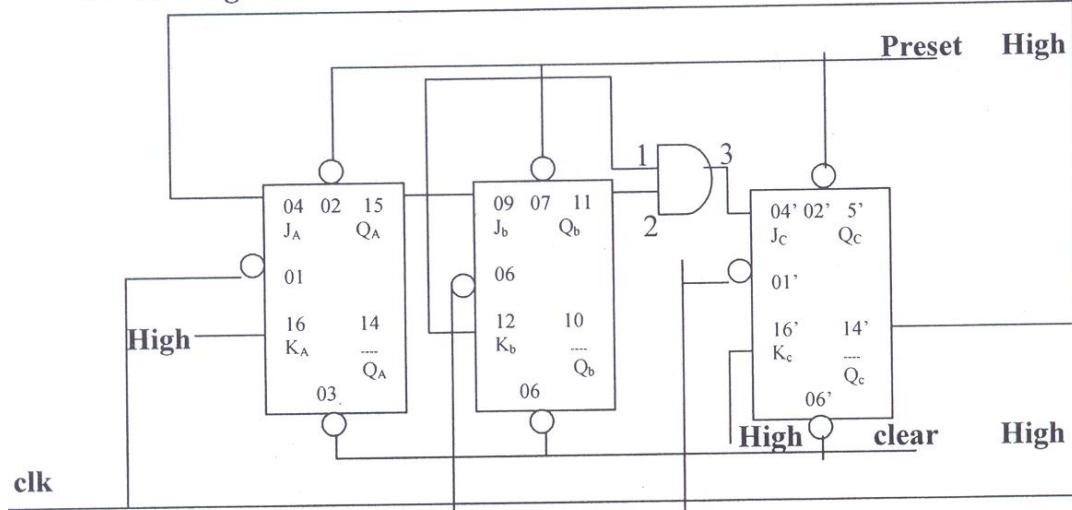


**Transition Table:**

Present State			Next State			Jc	Kc	Jb	Kb	Ja	Ka
Qc	Qb	Qa	Qc+1	Qb+1	Qa+1						
0	0	0	0	0	1	0	x	0	x	1	x
0	0	1	0	1	0	0	x	1	x	x	1
0	1	0	0	1	1	0	x	x	0	1	x
0	1	1	1	0	0	1	x	x	1	x	1
1	0	0	0	0	0	x	1	0	x	0	x
1	0	1	x	x	x	x	x	x	x	x	x
1	1	0	x	x	x	x	x	x	x	x	x
1	1	1	x	x	x	x	x	x	x	x	x

**K-Maps:**


**Circuit Diagram:**



**Procedure:**

- (1) Verify all components and patch chords whether they are in good condition or not.
- (2) Make connection as shown in the circuit diagram.
- (3) Give supply to the trainer kit.
- (4) Provide input data to circuit via switches.
- (5) Verify truth table sequence and observe outputs.

**RESULT:**

A mod n ( $n < 8$ ) synchronous up counter using JK FF IC's are designed, constructed and truth table was verified.

**9b. Design and develop the verilog/VHDL code for mod 8 up counter simulate and verify its working.**

**Program:-**

Module Vc ( Clk,Reset, Q)

Input clk,Reset;

Output[3:0]Q;

Reg[3:0]Q;

Always @ (posedge clk)

Begin

If (reset)

    Q=4'b000

Else

If (Q==4'b0111)

    Q=4'b000

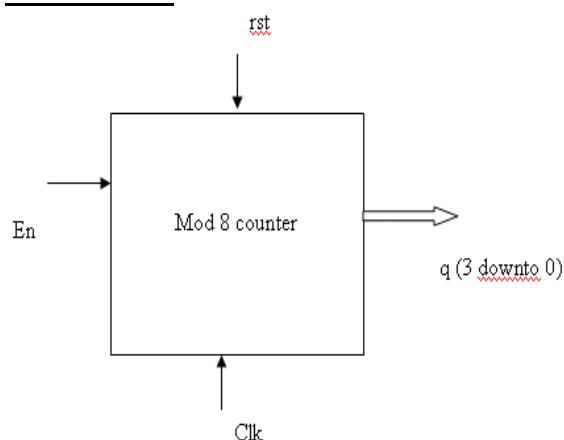
Else

    Q=Q+1;

End

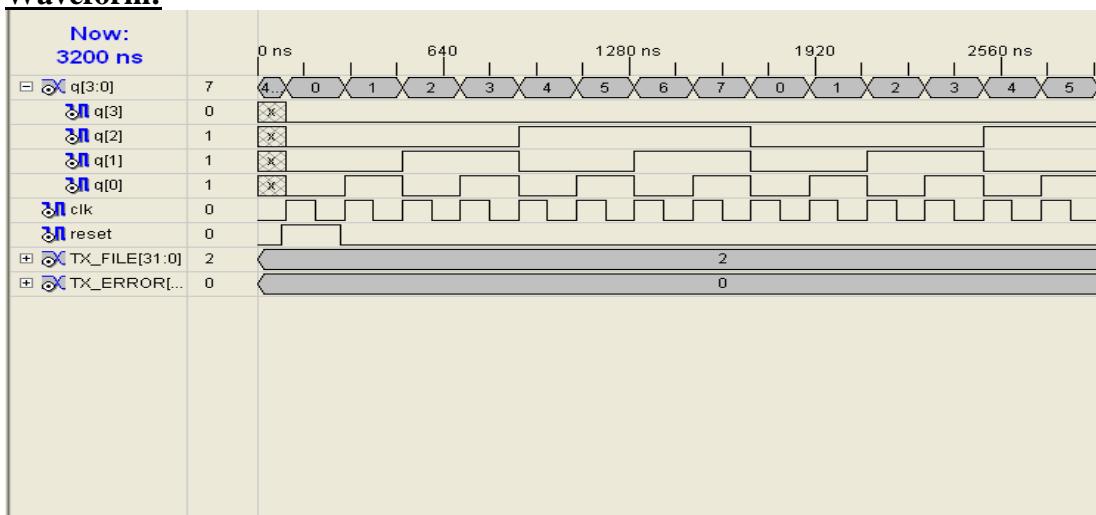
End Module;

**Truth Table**



rst	Clk	En	Q
1	X	0	0000
0	1	1	0001
0	1	1	0010
0	1	1	0011
0	1	1	0100
0	1	1	0101
0	1	1	0110
0	1	1	0111

**Waveform:**



## Experiment -10

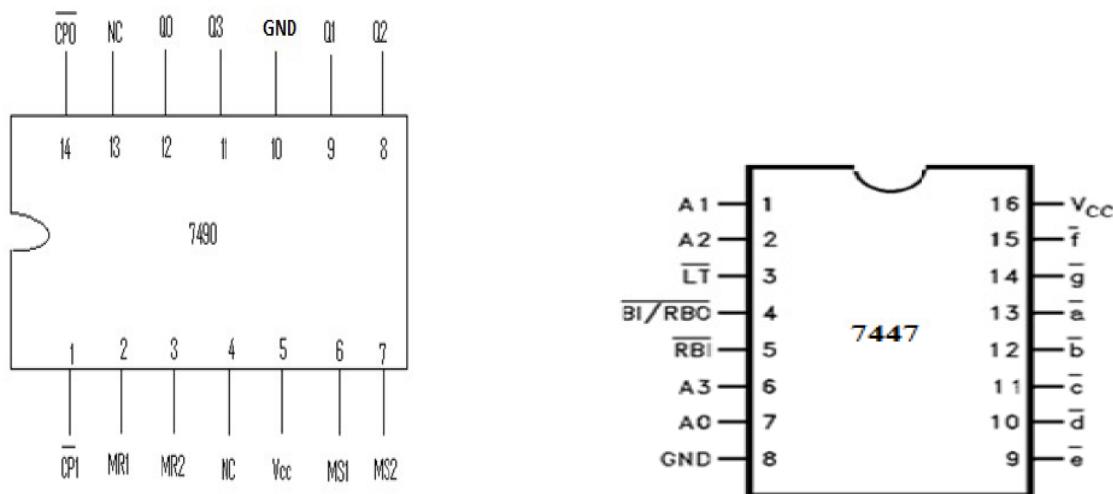
**Aim:** 10a) Design and implement asynchronous counter using decade counter IC to count up from 0 to n ( $n \leq 9$ ) and demonstrate on 7-segment display.

**Components used:** IC 74LS90, IC 7447, Patch chords, Power chords and Trainer kit.

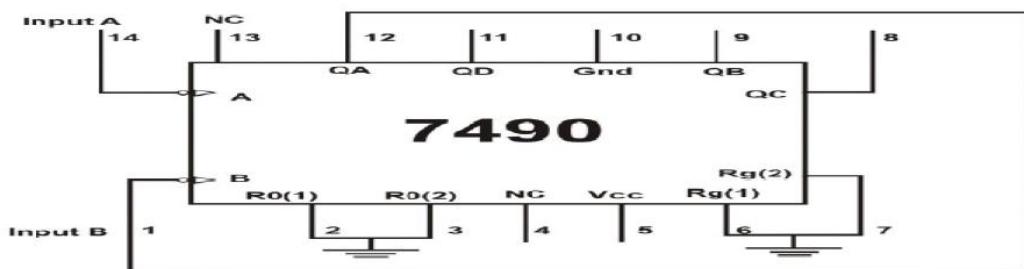
### Theory:

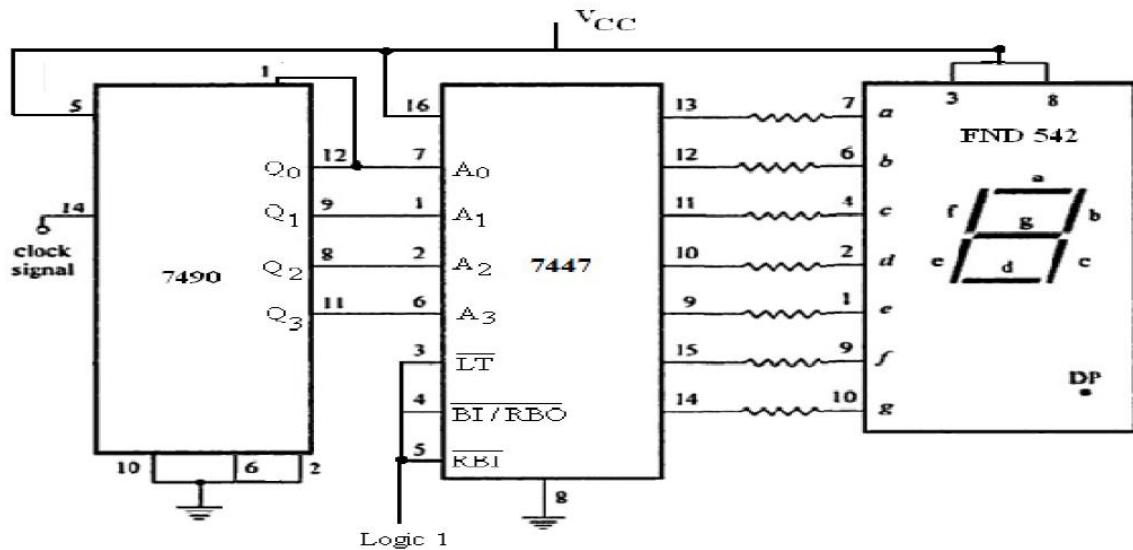
Asynchronous counter is a counter in which the clock signal is connected to the clock input of only first stage flip flop. The clock input of the second stage flip flop is triggered by the output of the first stage flip flop and so on. This introduces an inherent propagation delay time through a flip flop. A transition of input clock pulse and a transition of the output of a flip flop can never occur exactly at the same time. Therefore, the two flip flops are never simultaneously triggered, which results in asynchronous counter operation.

### Pin Diagram



### Circuit Diagram:





**Function Table:**

Clock	$Q_a$	$Q_b$	$Q_c$	$Q_d$
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
8	0	1	1	1
9	1	0	0	0

**Result:** To verify truth table.

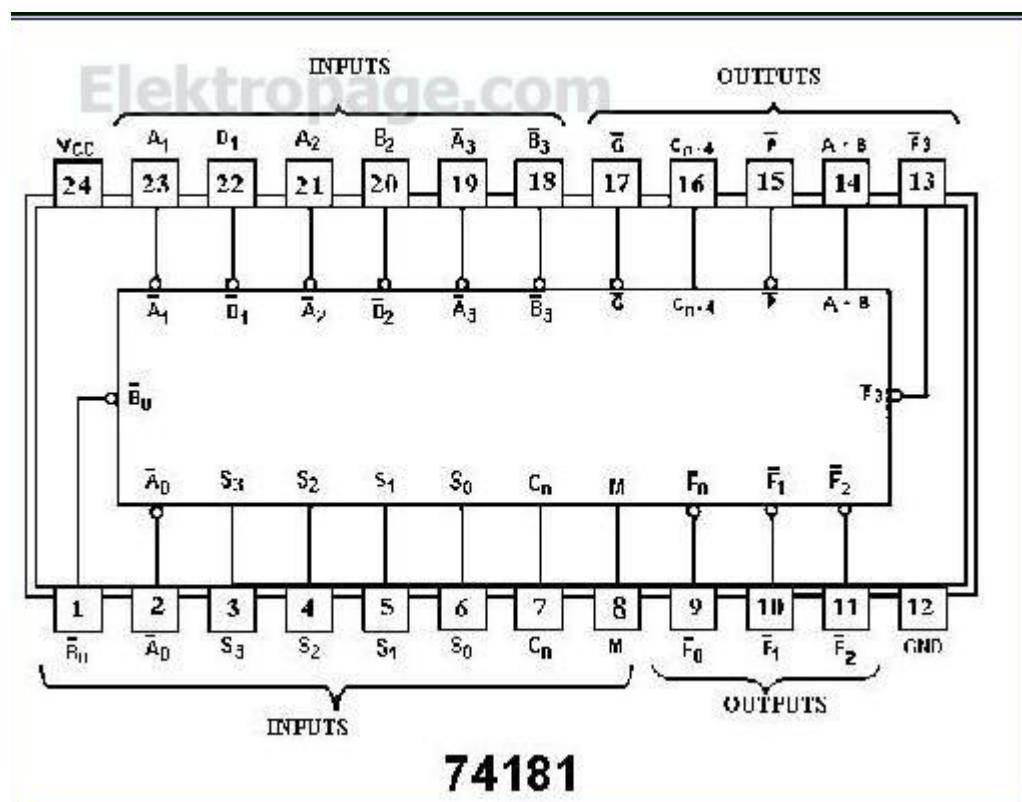
## Experiment -12

**Aim:** 12) To study 4-bitALU using IC-74181.

**Apparatus Required:** IC 74181, etc.

**Theory:**

The 74181 is a 7400 series medium-scale integration (MSI) TTL integrated circuit, containing the equivalent of 75 logic gates and most commonly packaged as a 24-pin DIP. The 4-bit wide ALU can perform all the traditional add / subtract / decrement operations with or without carry, as well as AND / NAND, OR / NOR, XOR, and shift. Many variations of these basic functions are available, for a total of 16 arithmetic and 16 logical operations on two four-bit words. Multiply and divide functions are not provided but can be performed in multiple steps using the shift and add or subtract functions. Shift is not an explicit function but can be derived from several available functions including  $(A+B)$  plus A.



**PIN DESCRIPTION**

PIN NO.	SYMBOL	NAME AND FUNCTION
1, 22, 20, 18	$\bar{B}_0$ to $\bar{B}_3$	operand inputs (active LOW)
2, 23, 21, 19	$\bar{A}_0$ to $\bar{A}_3$	operand inputs (active LOW)
6, 5, 4, 3	$S_0$ to $S_3$	select inputs
7	$C_0$	carry input
8	M	mode control input
9, 10, 11, 13	$\bar{F}_0$ to $\bar{F}_3$	function outputs (active LOW)
12	GND	ground (0 V)
14	A=B	comparator output
15	$\bar{P}$	carry propagate output (active LOW)
16	$C_{in4}$	carry output
17	$\bar{G}$	carry generate output (active LOW)
24	V <sub>cc</sub>	positive supply voltage

**FUNCTION TABLES**

MODE SELECT INPUTS				ACTIVE HIGH INPUTS AND OUTPUTS		MODE SELECT INPUTS				ACTIVE LOW INPUTS AND OUTPUTS	
$s_3$	$s_2$	$s_1$	$s_0$	LOGIC ( $M=H$ )	ARITHMETIC <sup>(2)</sup> ( $M=L; C_n=H$ )	$s_3$	$s_2$	$s_1$	$s_0$	LOGIC ( $M=H$ )	ARITHMETIC <sup>(2)</sup> ( $M=L; C_n=L$ )
L	L	L	L	$\bar{A}$	A	L	L	L	L	$\bar{A}$	A minus 1
L	L	L	H	$\bar{A} + \bar{B}$	$A + B$	L	L	L	H	$\bar{AB}$	$AB$ minus 1
L	L	H	L	$\bar{AB}$	$A + \bar{B}$	L	L	H	L	$\bar{A} + B$	$A\bar{B}$ minus 1
L	L	H	H	logical 0	minus 1	L	L	H	H	logical 1	minus 1
L	H	L	L	$\bar{AB}$	$A + AB$	L	H	L	L	$\bar{A} + B$	$A + (A + \bar{B})$
L	H	L	H	$\bar{B}$	$(A + B) + A\bar{B}$	L	H	L	H	$\bar{B}$	$AB + (A + B)$
L	H	H	L	$A \oplus B$	$A - B$ minus 1	L	H	H	L	$\bar{A} \oplus \bar{B}$	$A - B$ minus 1
L	H	H	H	$\bar{AB}$	$A\bar{B}$ minus 1	L	H	H	H	$A + \bar{B}$	$A + \bar{B}$
H	L	L	L	$\bar{A} + B$	$A + AB$	H	L	L	L	$\bar{AB}$	$A + (A + B)$
H	L	L	H	$\bar{A} \oplus \bar{B}$	$A + B$	H	L	L	H	$A \oplus B$	$A + B$
H	L	H	L	$B$	$(A + \bar{B}) + A\bar{B}$	H	L	H	L	$B$	$A\bar{B} + (A + B)$
H	L	H	H	$AB$	$AB$ minus 1	H	L	H	H	$A + B$	$A + B$
H	H	L	L	logical 1	$A + A^{(1)}$	H	H	L	L	logical 0	$A + A^{(1)}$
H	H	L	H	$A + \bar{B}$	$(A + B) + A$	H	H	L	H	$\bar{AB}$	$AB + A$
H	H	H	L	$A + B$	$(A + \bar{B}) + A$	H	H	H	L	$AB$	$A\bar{B} + A$
H	H	H	H	A	A minus 1	H	H	H	H	A	A

**Notes to the function tables**

1. Each bit is shifted to the next more significant position.
2. Arithmetic operations expressed in 2s complement notation.

H = HIGH voltage level

L = LOW voltage level

**Notes to the function tables**

1. Each bit is shifted to the next more significant position.
2. Arithmetic operations expressed in 2s complement notation.

H = HIGH voltage level

L = LOW voltage level

**PROCEDURE:**

1. Connections are made as shown in the Circuit diagram.
2. Change the values of the inputs and verify at least 5 functions given in the function table

**PRECAUTIONS:**

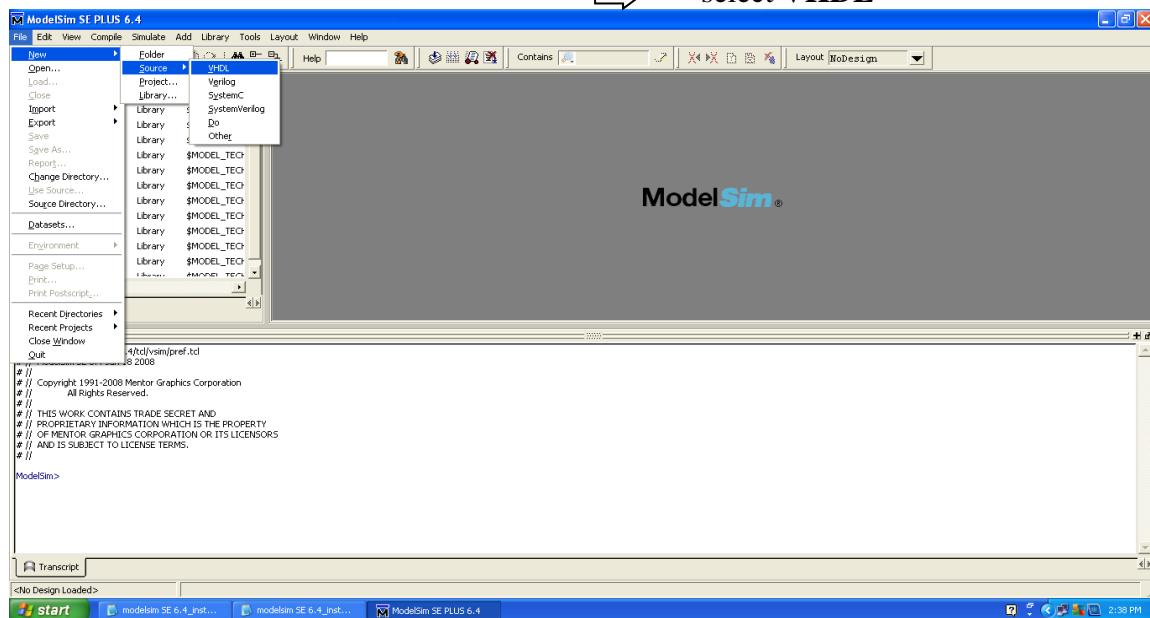
1. Make the connections according to the IC pin diagram.
2. The connections should be tight.
3. The  $V_{cc}$  and ground should be applied carefully at the specified pin only.

## Logic Design with Model Sim Software (VHDL)

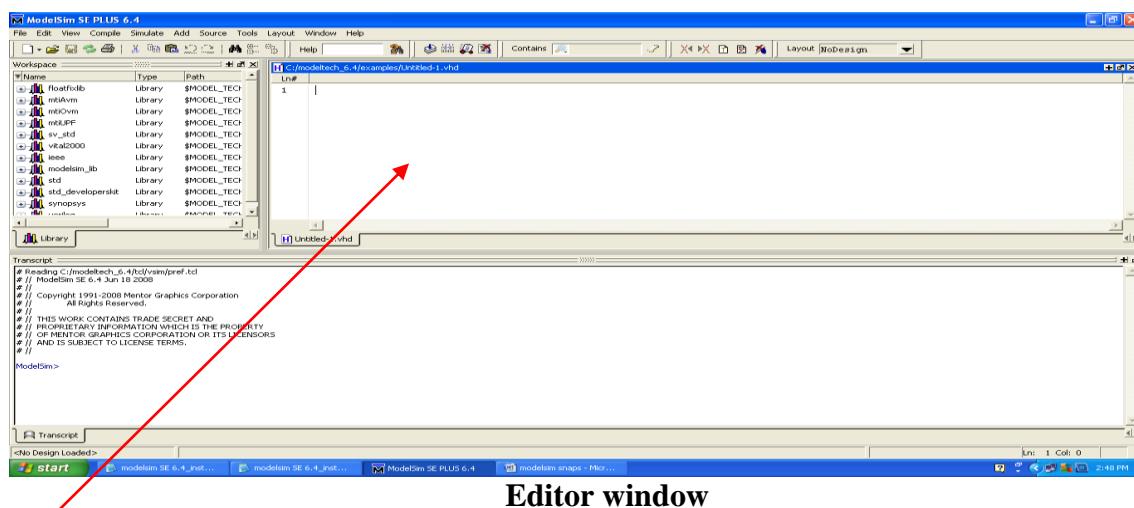
### Steps to write VHDL code and to simulate in Model Sim

1. Double click on ModelSim icon on your desktop.
2. Goto **File menu**, select **New**,

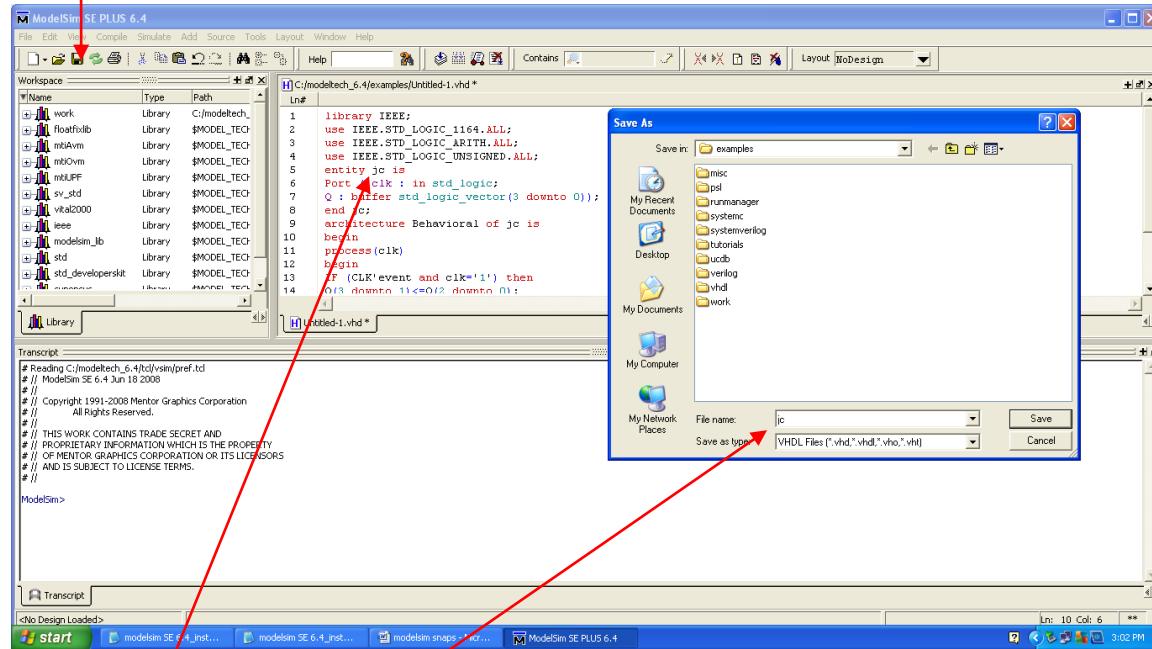
↳ select source      ↳ select VHDL



3. You will get text editor where you can type VHDL code and save.

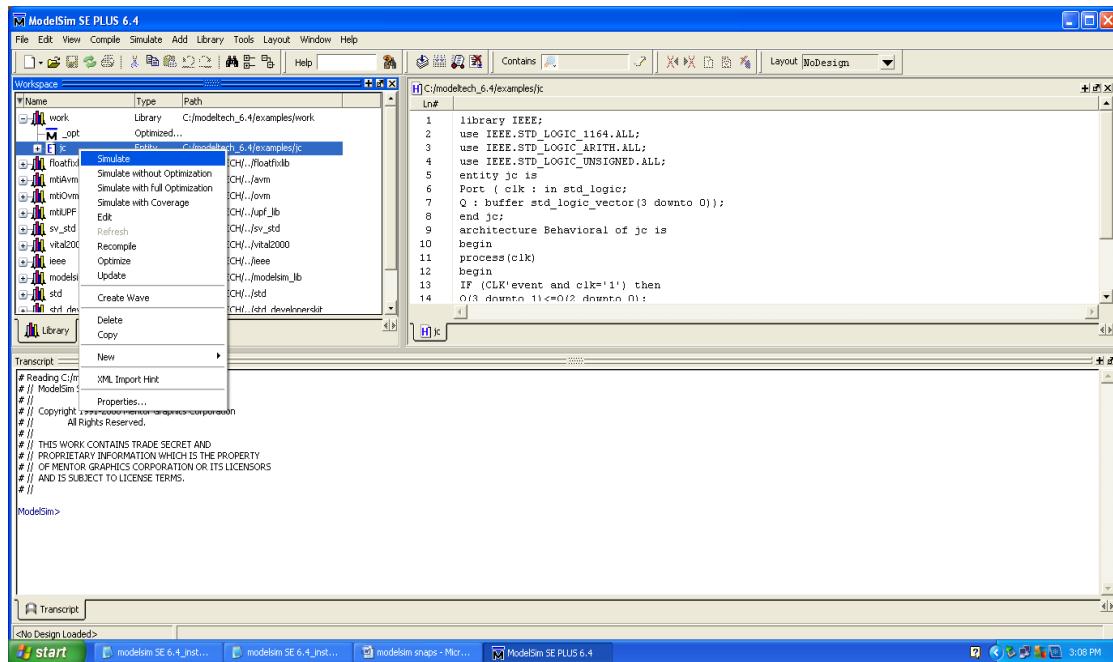


**Click on save button**



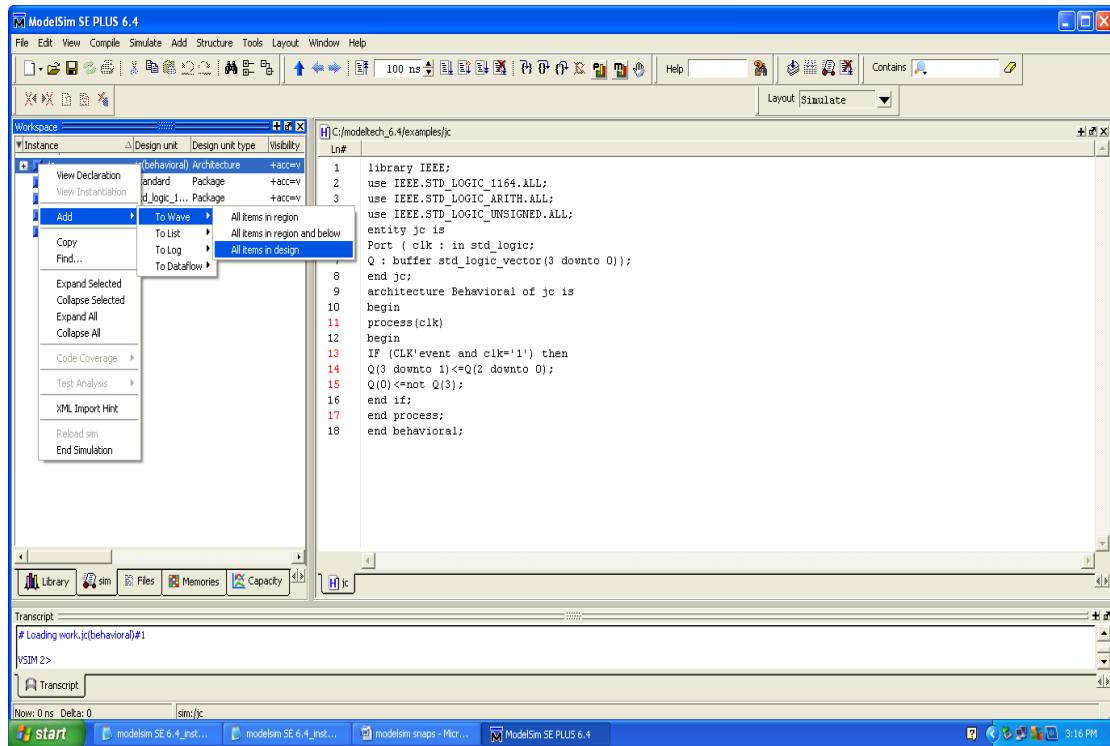
**Entity name and file name should be same.**

4. After saving click on View and select Workspace.
5. Click on work, right click on your file, choose simulate option.

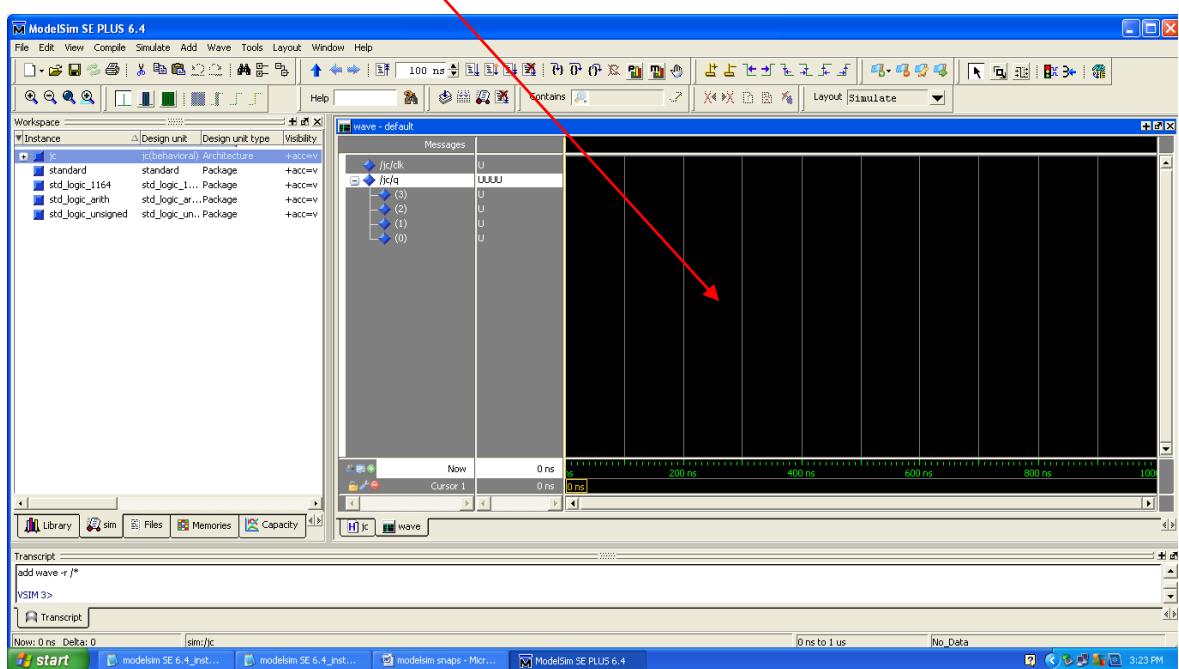


## 6. Select work space instance, right click on file,

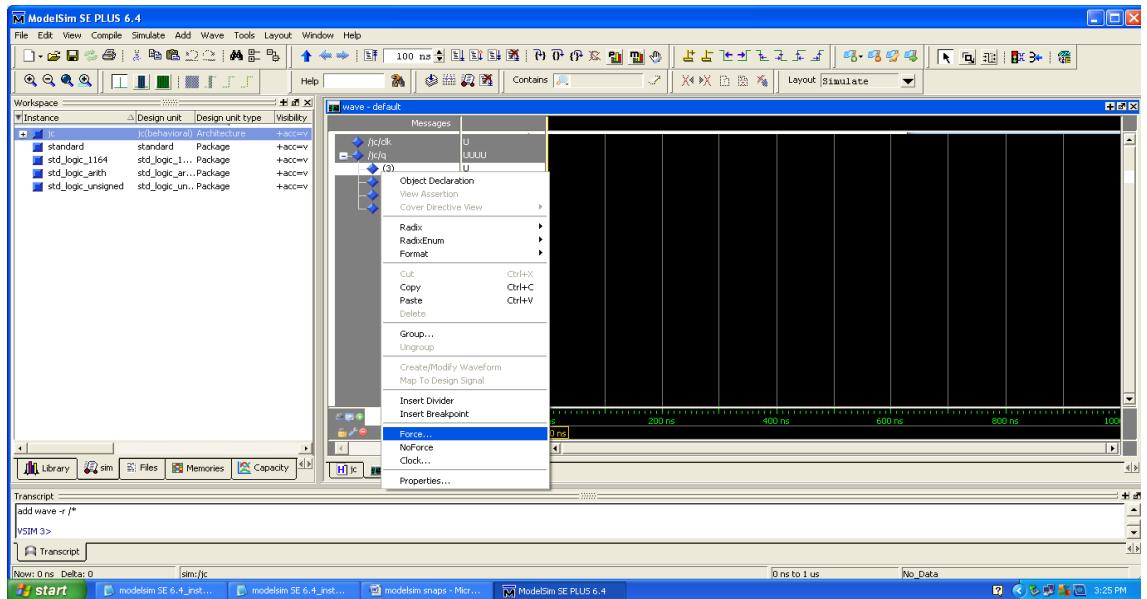
ADD To wave All items in design

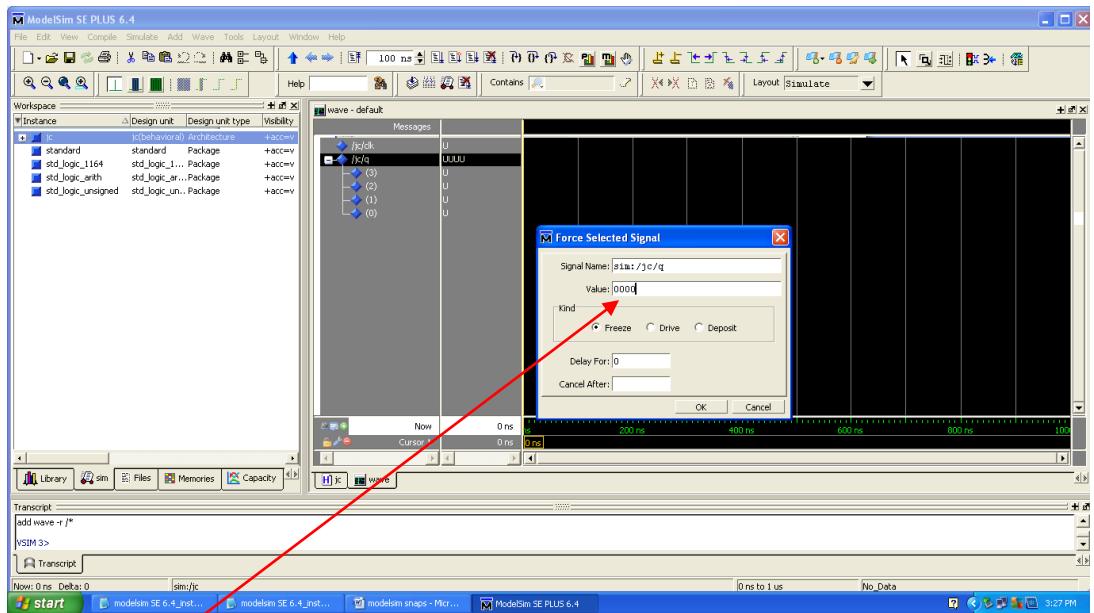


You will get waveform window

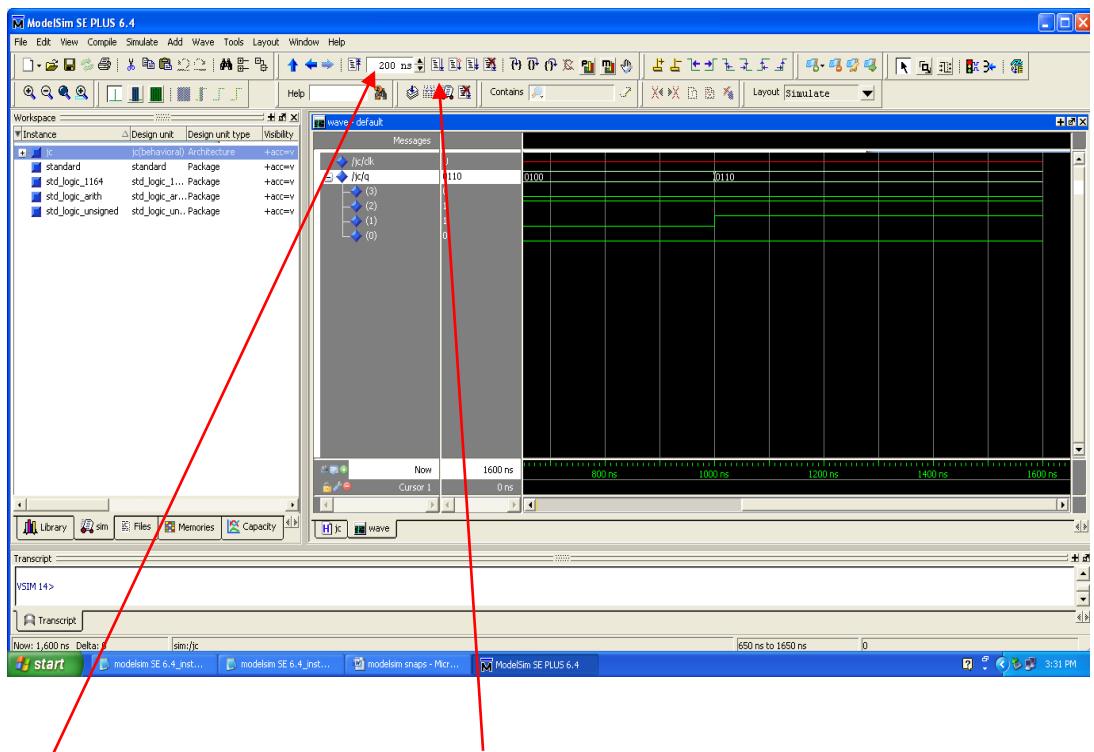


To assign the values, right click on entity and select Force.





Assign the values for all the bits and click **OK**.



Set the **time delay**

Click on **Run** button

**VIVA QUESTIONS ANALOG:**

1. Draw the basic structure of an N channel junction field effect transistor.
2. Why is FET known as a unipolar device?
3. What are the advantages and disadvantages of JFET over BJT?
4. What is a channel?
5. Distinguish between JFET and MOSFET.
6. What is an effect of cascading?
7. What are all the factors affecting the bandwidth of the RC Coupled amplifier?
8. Explain bypass capacitor?
9. What is meant by coupling capacitor?
10. Why does amplifier gain reduce?
11. Explain the different regions in frequency response?
12. State the types of distortions in amplifier?
13. What is cross over distortion? How it can be eliminated?
14. Define noise?
15. Draw the symbol of JFET and MOSFET.
16. What are the two modes of MOSFET?
17. Define pinch-off voltage
18. What is feedback and what are feedback amplifiers?
19. What is meant by positive and negative feedback?
20. What are the advantages and disadvantages of negative feedback?
21. Differentiate between voltage and current feedback in amplifiers?
22. What is the type of feedback used in an op- amp Schmitt trigger?
23. Give the expression for the frequency of oscillations in an op-amp sine wave oscillator?
24. What are the conditions for sustained oscillations or what is Barkhausen criterion
25. What are the classifications of Oscillators?
26. What are the types of feedback oscillators?
27. Define Piezo-electric effect?
28. Draw the equivalent circuit of crystal oscillator?
29. How does an oscillator differ from an amplifier?

**VIVA QUESTIONS ON LOGIC DESIGN:**

1. Why NAND & NOR gates are called universal gates?
2. Realize the EX – OR gates using minimum number of NAND gates.
3. Give the truth table for EX-NOR and realize using NAND gates?
4. Compare TTL logic family with CMOS family?
5. Which logic family is fastest and which has low power dissipation?
6. What are the different methods to obtain minimal expression?
7. What is a Min term and Max term
8. State the difference between SOP and POS.
9. What is meant by canonical representation?
10. What is K-map? Why is it used?
11. What is a multiplexer?
12. What is a de-multiplexer?
13. What are the applications of multiplexer and de-multiplexer?
14. Derive the Boolean expression for multiplexer and de-multiplexer.
15. In a 2n to 1 multiplexer how many selection lines are there?

16. Implement an 8:1 mux using 4:1 mux?
17. What is a comparator?
18. What are the applications of comparator?
19. Derive the Boolean expressions of one bit comparator and two bit comparators.
20. How do you realize a higher magnitude comparator using lower bit comparator?
21. Design a 2 bit comparator using a single Logic gates?
22. Design an 8 bit comparator using a two numbers of IC 7485?
23. What are the applications of decoder?
24. What is the difference between decoder & encoder?
25. For n- 2n decoder how many i/p lines & how many o/p lines?
26. Using 3:8 decoder and associated logic, implement a full adder
27. What is the difference between decoder and de-mux?
28. What are the different types of LEDs?
29. What are the applications of LEDs?
30. What is a priority encoder?
31. What is the difference between Flip-Flop & latch?
32. Give examples for synchronous & asynchronous inputs?
33. What are the applications of different Flip-Flops?
34. What is the advantage of Edge triggering over level triggering?
35. What is the relation between propagation delay & clock frequency of flip-flop?
36. What is race around in flip-flop & how to overcome it?
37. Convert the J K Flip-Flop into D flip-flop and T flip-flop
38. List the functions of asynchronous inputs?
39. What is the necessity for sequence generation?
40. What are PISO, SIPO, and SISO with respect to shift register?
41. Differentiate between serial data & parallel data
42. What is the significance of Mode control bit?
43. What is a ring counter?
44. What is a Johnson counter?
45. How many Flip-flops are present in IC 7495?
46. What is an asynchronous counter?
47. How is it different from a synchronous counter?
48. Realize asynchronous counter using T flip-flop
49. What are synchronous counters?
50. What are the advantages of synchronous counters?
51. What is an excitation table?
52. Write the excitation table for D, T FF
53. Design mod-5 synchronous counter using T FF
54. What is a presettable counter?
55. What are the applications of presettable counters?
56. Write the circuit for preset value of 0100 and N=5 (up counter)
57. What is a decade counter?
58. What do you mean by a ripple counter?