VISION AND MISSION OF THE CS&E DEPARTMENT

Vision
To develop technical professionals acquainted with recent trends and technologies of computer science to serve as valuable resource for the nation/society.

Mission:
Facilitating and exposing the students to various learning opportunities through dedicated academic teaching, guidance and monitoring.

VISION AND MISSION OF THE INSTITUTE

Vision
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 the society.

Mission
Accomplish stimulating learning environment through high quality academic instruction, innovation and industry-institute interface.
Instructions to students

1. Students Leave their foot wares outside.
2. Students keep their bags in the rack.
3. Students must taken care of their valuable things.
4. Students must bring Observation book, record and manual along with pen, pencil, and eraser etc., no borrowing from others.
5. Students must handle the trainer kit and other components carefully, as they are expensive.
6. Before entering to lab, must prepare for Viva for which they are going to conduct experiment.
7. Before switch on the trainer kit, must show the connections to one of the faculties or instructors.
8. After the completion of the experiment should return the components to the respective lab instructors.
9. Before leaving the lab, should check whether they have switch off the power supplies and keep their chairs properly.

DO’S AND DON’TS

- Be regular to the Lab Do not come late to the Lab
- 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.
Lab Grading

The faculty will examine your notebooks during lab period and assign a grade based upon the quality and contents of your pre-lab work.

There will be eleven lab experiments and a one internal lab test of 100 marks (Reduced to 20). Each lab experiment is worth **up to 20 points**. Each lab contains four parts as follows.

1. **Observation (5 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. Late observations will not be accepted.

2. **Viva (5 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 (10 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.

4. At the end of the semester all notebooks will be collected for a final grade by the faculty.

5. **Penalty for incomplete work**: If any of the 3 parts is missed, a score of zero will be reported by the faculty for that lab.
**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:
   - Title of experiment
   - Aim/Objectives
   - Components Required
   - Theory
   - Procedure described clearly in steps
   - Result
8. On each left side page:
   - Pin diagrams
   - Circuit diagram
   - Tables
   - Graphs
9. Use labels and captions for figures and tables.
10. Attach printouts and plots of data as needed. Stick printouts (A4 Size) on the right side of the lab record.
11. Strictly observe the instructions given by the Teacher/ Lab Instructor.
ANALOG AND DIGITAL ELECTRONICS LABORATORY

[As per Choice Based Credit System (CBCS) scheme]
(Effective from the academic year 2015 -2016)

SEMESTER - III

Laboratory Code : 15CSL37
IA Marks : 20

Number of Lecture Hours/Week : 01I + 02P
Exam Marks : 80

Total Number of Lecture Hours : 40
Exam Hours: 03

Course objectives:

These laboratory courses enable 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 operations
- Counters and Registers using Flip-flops.
- Synchronous and Asynchronous Sequential Circuits.
- A/D and D/A Converters.

Descriptions (if any)

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.
# CONTENTS

<table>
<thead>
<tr>
<th>Sl No.</th>
<th>Experiment</th>
<th>Page No.</th>
</tr>
</thead>
</table>
| 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. | 32-35 |
| 2.     | a. Design and construct a rectangular waveform generator (Op-Amp relaxation oscillator) for given frequency.  
        b. Design and implement a rectangular waveform generator (Op-Amp relaxation oscillator) using a simulation package and observe the change in frequency when all resistor values are doubled. | 36-38 |
| 3.     | Design and implement an astable multivibrator circuit using 555 timer for a given frequency and duty cycle. | 39-42 |
| 4.     | Design and implement Half adder, Full Adder, Half Subtractor, Full Subtractor using basic gates. | 43-48 |
| 5.     | a. Given any 4-variable logic expression, simplify using Entered Variable Map and realize the simplified logic expression using 8:1 multiplexer IC.  
        b. Write the Verilog / VHDL code for an 8:1 multiplexer. Simulate and verify its working. | 49-52 |
| 6.     | a) Design and implement code converter I) Binary to Gray II) Gray to Binary Code using basic gates. | 53-57 |
| 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. | 58-62 |
        b. Write the Verilog / VHDL code for D Flip-Flop with positive-edge triggering. Simulate and verify its working. | 63-66 |
| 9.     | a. Design and implement a mod-n (n<8) synchronous up counter using J-K Flip-Flop ICs & demonstrate its working.  
        b. Write the Verilog / VHDL code for mod-8 up counter. Simulate and verify its working | 67-70 |
| 10.    | Design and implement an asynchronous counter using decade counter IC to count up from 0 to n (n<=9) and demonstrate on 7 segment display (using IC 7447). | 71-73 |
| 11.    | Generate a Ramp output waveform using DAC0800 (Inputs are given to DAC through IC74393 dual 4-bit binary counter). | 74-76 |
| 12.    | Study experiment: To study 4-bit ALU using IC-74181. | 77-80 |
|        | VIVA QUESTIONS | 81-83 |
Course outcomes:

On the completion of this laboratory course, the students will be able to:

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

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.
1. INTRODUCTION

A Digital design is also called as Logic design, the goal of design is to build systems. Digital design is engineering and engineering means “problem solving”. Digital design is concerned with the design of digital electronics circuits. Digital circuits are employed in the design and construction of systems such as digital computers, data communication, digital recording, and many other applications that require digital hardware.

Digital system have such a prominent role in everyday life that we refer to the present technological period as the digital age. Digital systems are used in communication, business transactions, traffic control, space guidance, medical treatment, weather monitoring, the internet, and many other commercial, industrial and scientific enterprises.

We have digital telephones, digital television, digital versatile discs, digital cameras, and digital computers. The most striking property of the digital computer is its generality. It can follow a sequence of instructions, called a program, that operates on given data. The user can specify and change the program or the data according to the specific need. Because of this flexibility, general-purpose digital computers can perform a variety of information processing tasks that range over a wide spectrum of applications. One characteristic of digital system is their ability to manipulate discrete elements of information. Early digital computers were used for numeric computations.

In this case, the discrete elements used were the digits. From this application the term digital computer emerged. Discrete elements of information are represented in digital system by physical quantities called signals. Electronic devices called transistors predominate in the circuitry that implements these signals. The signals in most present day electronic digital system use just two discrete values and so it is called as binary. A binary digit, called a bit, has two values: 0 and 1. Discrete elements of information are represented with groups of bits called binary codes.

The logic design lab is an introduction to digital electronics where the students conduct simple experiments and design simple digital electronic circuits and verify the outputs using digital trainer kits. The entire set of experiments are classified into two subdivisions combinational circuit design and Sequential circuit design.

Combinational circuits are digital circuits that do not have memory, i.e., their output at any given time depend only on the present set of inputs that is given to the circuit, whereas a sequential circuits are those whose outputs depend not only on the present inputs given, but
Advantages:

One advantage of digital circuits when compared to analog circuits is signals represented digitally can be transmitted without degradation due to noise. For example, a continuous audio signal, transmitted as a sequence of 1s and 0s, can be reconstructed without error provided the noise picked up in transmission is not enough to prevent identification of the 1s and 0s. An hour of music can be stored on a compact disc using about 6 billion binary digits.

In a digital system, a more precise representation of a signal can be obtained by using more binary digits to represent it. While this requires more digital circuits to process the signals, each digit is handled by the same kind of hardware. In an analog system, additional resolution requires fundamental improvements in the linearity and noise characteristics of each step of the signal chain.

Computer-controlled digital systems can be controlled by software, allowing new functions to be added without changing hardware. Often this can be done outside of the factory by updating the product's software. So, the product's design errors can be corrected after the product is in a customer's hands.

Information storage can be easier in digital systems than in analog ones. The noise-immunity of digital systems permits data to be stored and retrieved without degradation. In an analog system, noise from aging and wear degrade the information stored. In a digital system, as long as the total noise is below a certain level, the information can be recovered perfectly.

Disadvantages:

In some cases, digital circuits use more energy than analog circuits to accomplish the same tasks, thus producing more heat which increases the complexity of the circuits such as the inclusion of heat sinks. In portable or battery-powered systems this can limit use of digital systems.

For example, battery-powered cellular telephones often use a low-power analog front-end to amplify and tune in the radio signals from the base station. However, a base station has grid power and can use power-hungry, but very flexible software radios. Such base stations can be easily reprogrammed to process the signals used in new cellular standards. Digital circuits are sometimes more expensive, especially in small quantities.

Most useful digital systems must translate from continuous analog signals to discrete digital signals. This causes quantization errors. Quantization error can be reduced if the system stores enough digital data to represent the signal to the desired degree of fidelity. The
Nyquist-Shannon sampling theorem provides an important guideline as to how much digital data is needed to accurately portray a given analog signal.

In some systems, if a single piece of digital data is lost or misinterpreted, the meaning of large blocks of related data can completely change. Because of the cliff effect, it can be difficult for users to tell if a particular system is right on the edge of failure, or if it can tolerate much more noise before failing. Digital fragility can be reduced by designing a digital system for robustness. For example, a parity bit or other error management method can be inserted into the signal path. These schemes help the system detect errors, and then either correct the errors, or at least ask for a new copy of the data. In a state-machine, the state transition logic can be designed to catch unused states and trigger a reset sequence or other error recovery routine.

Digital memory and transmission systems can use techniques such as error detection and correction to use additional data to correct any errors in transmission and storage.

On the other hand, some techniques used in digital systems make those systems more vulnerable to single-bit errors. These techniques are acceptable when the underlying bits are reliable enough that such errors are highly unlikely. A single-bit error in audio data stored directly as linear pulse code modulation (such as on a CD-ROM) causes, at worst, a single click. Instead, many people use audio compression to save storage space and download time, even though a single-bit error may corrupt the entire song.
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.

1. DIODE

Diodes must be connected the correct way round, the diagram may be labeled a or + for anode and k or - for cathode (yes, it really is k, not c, for cathode!). The cathode is marked by a line painted on the body. Diodes are labeled with their code in small print; you may need a magnifying glass to read this on small signal diodes.

Testing a diode with a DIGITAL Multimeter

1. Digital Multimeter has a special setting for testing a diode, usually labeled with the diode symbol.
2. Connect the red (+) lead to the anode and the black (-) to the cathode. The diode should conduct and the meter will display a value (usually the voltage across the diode in mV, 1000mV = 1V).
3. Reverse the connections. The diode should NOT conduct this way so the meter will display "off the scale" (usually blank except for a 1 on the left).
2. **TRANSISTOR**

A transistor is a semiconductor device used to amplify or switch electronic signals and electrical power. It is composed of semiconductor material usually with at least three terminals for connection to an external circuit. A voltage or current applied to one pair of the transistor’s terminals changes the current through another pair of terminals. Because the controlled (output) power can be higher than the controlling (input) power, a transistor can amplify a signal. Today, some transistors are packaged individually, but many more are found embedded in integrated circuits.

**Testing a transistor with a Multimeter**

Set a digital Multimeter to diode test and as described above for testing a diode.

**Test each pair of leads both ways** (six tests in total):

1. The base-emitter (BE) junction should behave like a diode and conduct one way only.
2. The base-collector (BC) junction should behave like a diode and conduct one way only.
3. The collector-emitter (CE) should not conduct either way.

**Types of transistor:**

There are two types of standard transistors, NPN and PNP, with different circuit symbols. The letters refer to the layers of semiconductor material used to make the transistor. Most transistors used today are NPN because this is the easiest type to make from silicon.

**SL 100 is an NPN transistor.**

![Testing an NPN transistor](image)
3. RESISTORS

A register is a memory device that can be used to store more than one bit of information. A register is usually realized as several flip-flops with common control signals that control the movement of data to and from the register. „Common refers to the property that the control signals apply to all flip-flops in the same way „. A register is a generalization of a flip-flop. Where a flipflop stores one bit, a register stores several bits „. The main operations on a register are the same as for any storage devices, namely ‹Load or Store: Put new data into the register ‹Read: Retrieve the data stored in the register (usually without changing the stored data It is used to control the voltages and the currents in your circuit.

4. CAPACITORS

A capacitor (originally known as a condenser) is a passive two-terminal electrical component used to store electrical energy temporarily in an electric field. The forms of practical capacitors vary widely, but all contain at least two electrical conductors (plates) separated by a dielectric (i.e. an insulator that can store energy by becoming polarized). The conductors can be thin films, foils or sintered beads of metal or conductive electrolyte, etc. The non-conducting dielectric acts to increase the capacitor's charge capacity. Materials commonly used as dielectrics include glass, ceramic, plastic film, air, vacuum, paper, mica, and oxide layers. Capacitors are widely used as parts of electrical circuits in many common electrical devices. Unlike a resistor, an ideal capacitor does not dissipate energy. Instead, a capacitor stores energy in the form of an electrostatic field between its plates.
5. 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. To use the bread board 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.

6. INTEGRATED CIRCUIT

An Integrated Circuit (IC) consists of many basic electronic components. An integrated circuit or monolithic integrated circuit (also referred to as an IC, a chip, or a microchip) is a set of electronic circuits on one small plate ("chip") of semiconductor material, normally silicon. This can be made much smaller than a discrete circuit made from independent electronic components. ICs can be made very compact, having up to several billion transistors and other electronic components in an area the size of a human fingernail.
7. VOLTMETERS

A voltmeter, also known as a voltage meter, is an instrument used for measuring the potential difference, or voltage, between two points in an electrical or electronic circuit. Some voltmeters are intended for use in direct current (DC) circuits; others are designed for alternating current (AC) circuits. Specialized voltmeters can measure radio frequency (RF) voltage.

A basic analog voltmeter consists of a sensitive galvanometer (current meter) in series with a high resistance. The internal resistance of a voltmeter must be high. Otherwise it will draw significant current, and thereby disturb the operation of the circuit under test. The sensitivity of the galvanometer and the value of the series resistance determine the range of voltages that the meter can display.

8. AMMETER

Ammeter means Ampere-meter which measures ampere value. Ampere is the unit of current so an ammeter is a meter or an instrument which measures current.

Working Principle of Ammeter

The main principle of ammeter is that it must have a very low resistance and also inductive reactance. Now, why do we need this? can't we connect an ammeter in parallel? The answer to this question is it has very low impedance because it must have very low amount of voltage drop across it and must be connected in series connection because current is same in the series circuit. Also due to very low impedance the power loss will be low and if it is connected in parallel it becomes almost a short circuited path and all the current will flow through ammeter as a result of high current the instrument may burn. So due to this reason it must be connected in series. For an ideal ammeter, it must have zero impedance so that it has
zero voltage drop across it so the power loss in the instrument is zero. But the ideal is not achievable practically.

9. POTENTIOMETER

This is a very basic instrument used for comparing emf two cells and for calibrating ammeter, voltmeter and watt-meter. The basic working principle of potentiometer is very very simple. Suppose we have connected two battery in head to head and tale to tale through a galvanometer. That means the positive terminals of both battery are connected together and negative terminals are also connected together through a galvanometer as shown in the figure.

Here in the figure it is clear that if the voltage of both battery cells is exactly equal, there will be no circulating current in the circuit and hence the galvanometer shows null deflection. The working principle of potentiometer depends upon this phenomenon.
10. OSCILLOSCOPE

An oscilloscope is easily the most useful instrument available for testing circuits because it allows you to see the signals at different points in the circuit. The best way of investigating an electronic system is to monitor signals at the input and output of each system block, checking that each block is operating as expected and is correctly linked to the next.

The screen of this oscilloscope has 8 squares or divisions on the vertical axis, and 10 squares or divisions on the horizontal axis. Usually, these squares are 1 cm in each direction.

Setting up the CRO
i. Before you switch the oscilloscope on, check that all the controls are in their 'normal' positions.
   1. all push button switches are in the OUT position
   2. all slide switches are in the UP position
   3. all rotating controls are CENTRED

ii. Check through all the controls and put them in these positions:

   iii. Set both VOLTS/DIV controls to 1 V/DIV and the TIME/DIV control to 2 s/DIV, its slowest setting:

   iv. Switch ON, red button, top center:
The green LED illuminates and, after a few moments, you should see a small bright spot, or trace, moving fairly slowly across the screen.

v. Find the Y-POS 1 control:

The Y-POS 1 allows you to move the spot up and down the screen. For the present, adjacent trace so that it runs horizontally across the centre of the screen.

vi. Now investigate the INTENSITY and FOCUS controls:

When these are correctly set, the spot will be reasonably bright but not glaring, and as sharply focused as possible. (The TR control is screwdriver adjusted. It is only needed if the spot moves at an angle rather than horizontally across the screen with no signal connected.)

Adjusting the INTENSITY control changes the brightness of the oscilloscope display. The FOCUS should be set to produce a bright clear trace. If required, TR can be adjusted using a small screwdriver so that the oscilloscope trace is exactly horizontal when no signal is connected.

vii. The TIME/DIV control determines the horizontal scale of the graph which appears on the oscilloscope screen.

viii. The VOLTS/DIV controls determine the vertical scale of the graph drawn on the oscilloscope screen.
The diagram shows a lead with a BNC plug at one end and crocodile clips at the other. Adjust VOLTS/DIV and TIME/DIV until you obtain a clear picture of the signal, which should look like this:

**DC/AC/GND slide switches:** In the DC position, the signal input is connected directly to the Y-amplifier of the corresponding channel, CH I or CH II. In the AC position, a capacitor is connected into the signal pathway so that DC voltages are blocked and only changing AC signals are displayed.

In the GND position, the input of the Y-amplifier is connected to 0 V. This allows you to check the position of 0 V on the oscilloscope screen.

**Trace selection switches:** The settings of these switches control which traces appear on the oscilloscope screen.

**11.OP AMP:(OPERATIONAL AMPLIFIER)**

Electronic amplifiers covert a signal that carries a low amount of energy, whether it is audio or video, into a signal with a high amount of energy. The signal is not modified, only amplified. In the case of sound, the signal output is greater in magnitude. In other words, if a radio is not loud, an amplifier can make it loud.

Amplifiers can also work with light signals to make them brighter. For example, by increasing the voltage, a signal can become brighter, as in a lamp which may have two or three brightness settings.

Some other examples of amplifiers are speakers, home stereo systems or public address systems.
An Op-Amp can be conveniently divided into four main blocks:

1. An Input Stage or Input Diff. Amp.
2. The Gain Stage
3. The Level Translator
4. An Output Stage

Note: It can be used to perform various mathematical operations such as Addition, Subtraction, Integration, Differentiation, log etc.

Where:
\( V^+ \): non-inverting input
\( V^- \): inverting input
\( V_{out} \): output
\( V_{S+} \): positive power supply (sometimes also \( V_{DD}, V_{CC}, \) or \( V_{CC+} \))
\( V_{S-} \): negative power supply (sometimes also \( V_{SS}, V_{EE}, \) or \( V_{CC-} \))

**12. OP-AMP RELAXATION OSCILLATOR**

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 \( \frac{R_2}{R_1+R_2} \) of output is fed back to the noninverting input terminal. Thus reference voltage is \( \frac{R_2}{R_1+R_2} \) \( V_o \). And may take values as \( +\frac{R_2}{R_1+R_2} \) \( V_{sat} \) or \( -\frac{R_2}{R_1+R_2} \) \( V_{sat} \). 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.

### Circuit Diagram

**Design:**

The period of the output rectangular wave is given as

\[
T = \frac{2RC \ln (1+\beta/1-\beta)}{1} 
\]

Where,

\[
\beta = \frac{R_1}{R_1 + R_2} \text{ is the feedback fraction}
\]

If \( R_1 = R_2 \), then from equation (1) we have

\[
T = \frac{2RC \ln(3)}{2}
\]

Design for a frequency of 1 kHz (implies \( T = 1\text{ms} \)) Let \( C = 0.1\mu\text{F} \)

Then calculating \( R \) as

\[
R = \frac{1\text{ms}}{2 \times 0.1\mu\text{F} \ln(3)} = 10^{-3} \times 2 \times 0.1 \times 10^{-6} \times 1.099 = 5 \times 10^3
\]

\( = 5\text{K} \) Select \( R = 4.7\text{KΩ} \)

The voltage across the capacitor has a peak voltage of \( V_c = (R_1/R_1 + R_2) V_{\text{saturated}} \)

### Waveforms

**Waveform Diagram**

13. **SIGNAL/FUNCTION GENERATOR**

A function generator is a device that can produce various patterns of voltage at a variety of frequencies and amplitudes. It is used to test the response of circuits to common input signals. The electrical leads from the device are attached to the ground and signal input terminals of the device under test.

Most function generators allow the user to choose the shape of the output from a small number of options.

- Square wave - The signal goes
- Sine wave - The signal curves like a sinusoid from high to low voltage.
• Triangle wave - The signal goes from high to low voltage at a fixed rate.

The amplitude control on a function generator varies the voltage difference and low voltage of the output signal. The frequency control of a function generator controls the rate at which output signal oscillates.

Most function generators allow the user to choose the shape of the output from a small number of options.

• Square wave - The signal goes directly from high to low voltage.
• Sine wave - The signal curves like a sinusoid from high to low voltage.
• Triangle wave - The signal goes from high to low voltage at a fixed rate.

The amplitude control on a function generator varies the voltage difference between the high and low voltage of the output signal. The frequency control of a function generator controls the rate at which output signal oscillates.

Switch on the function generator and adjust the output level to produce a visible signal on the oscilloscope screen. Adjust TIME/DIV and VOLTS/DIV to obtain a clear display and investigate the effects of pressing the waveform shape buttons.

The rotating FREQUENCY control and the RANGE switch are used together to determine the frequency of the output signal.

14. MULTIVIBRATOR

Multivibrator is a form of oscillator, which has a non-sinusoidal output. The output waveform is rectangular. The multivibrators are classified as

1. **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.

2. **Monostable or one shot multivibrators:** It has one stable state and one quasi stable. The application of an input pulse triggers the circuit time constants and the output goes to the quazi stable state, 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.

3. **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.

![Circuit Diagram](image)

Connect the pin 2 to the CRO to get the capacitor waveform check the amplitude from the waveform to get the UTP and LTP values.

Connect pin 3 to CRO to get the output. Find out the \( T_H \) and \( T_L \) values.
Waveforms

(15) 555 TIMER

The 555 timer IC was introduced in the year 1970 by Signetic Corporation and gave the name SE/NE 555 timer. It is basically a monolithic timing circuit that produces accurate and highly stable time delays or oscillation. When compared to the applications of an op-amp in the same areas, the 555IC is also equally reliable and is cheap in cost. Apart from its applications as a monostable multivibrator and astable multivibrator, a 555 timer can also be used in dc-dc converters, digital logic probes, waveform generators, analog frequency meters and tachometers, temperature measurement and control devices, voltage regulators etc. The timer IC is setup to work in either of the two modes – one-shot or monostable or as a free-running or astable multivibrator. The SE 555 can be used for temperature ranges between – 55°C to 125°C. The NE 555 can be used for a temperature range between 0°C to 70°C.

The important features of the 555 timer are:

- It operates from a wide range of power supplies ranging from + 5 Volts to + 18 Volts supply voltage.
- Sinking or sourcing 200 mA of load current.
- The external components should be selected properly so that the timing intervals can be made into several minutes along with the frequencies exceeding several hundred kilo hertz.
- The output of a 555 timer can drive a transistor-transistor logic (TTL) due to its high current output.
- It has a temperature stability of 50 parts per million (ppm) per degree Celsius change in temperature, or equivalently 0.005 %/ °C.
- The duty cycle of the timer is adjustable.
- The maximum power dissipation per package is 600 mW and its trigger and reset inputs has logic compatibility. More features are listed in the datasheet.
IC Pin Configuration

The 555 Timer IC is available as an 8-pin metal can, an 8-pin mini DIP (dual-in-package) or a 14-pin DIP. The pin configuration is shown in the figures.

This IC consists of 23 transistors, 2 diodes and 16 resistors. The use of each pin in the IC is explained below. The pin numbers used below refers to the 8-pin DIP and 8-pin metal can packages. These pins are explained in detail, and you will get a better idea after going through the entire post.

**Pin 1: Grounded Terminal:** All the voltages are measured with respect to the Ground terminal.

**Pin 2: Trigger Terminal:** The trigger pin is used to feed the trigger input when the 555 IC is set up as a monostable multivibrator. This pin is an inverting input of a comparator and is responsible for the transition of flip-flop from set to reset. The output of the timer depends on the amplitude of the external trigger pulse applied to this pin. A negative pulse with a dc level greater than Vcc/3 is applied to this terminal. In the negative edge, as the trigger passes
through Vcc/3, the output of the lower comparator becomes high and the complimentary of Q becomes zero. Thus the 555 IC output gets a high voltage, and thus a quasi stable state.

**Pin 3: Output Terminal:** Output of the timer is available at this pin. There are two ways in which a load can be connected to the output terminal. One way is to connect between output pin (pin 3) and ground pin (pin 1) or between pin 3 and supply pin (pin 8). The load connected between output and ground supply pin is called the *normally on load* and that connected between output and ground pin is called the *normally off load*.

**Pin 4: Reset Terminal:** Whenever the timer IC is to be reset or disabled, a negative pulse is applied to pin 4, and thus is named as reset terminal. The output is reset irrespective of the input condition. When this pin is not to be used for reset purpose, it should be connected to + Vcc to avoid any possibility of false triggering.

**Pin 5: Control Voltage Terminal:** The threshold and trigger levels are controlled using this pin. The pulse width of the output waveform is determined by connecting a POT or bringing in an external voltage to this pin. The external voltage applied to this pin can also be used to modulate the output waveform. Thus, the amount of voltage applied in this terminal will decide when the comparator is to be switched, and thus changes the pulse width of the output. When this pin is not used, it should be bypassed to ground through a 0.01 micro Farad to avoid any noise problem.

**Pin 6: Threshold Terminal:** This is the non-inverting input terminal of comparator 1, which compares the voltage applied to the terminal with a reference voltage of 2/3 Vcc. The amplitude of voltage applied to this terminal is responsible for the set state of flip-flop. When the voltage applied in this terminal is greater than 2/3Vcc, the upper comparator switches to +Vs and the output gets reset.

**Pin 7: Discharge Terminal:** This pin is connected internally to the collector of transistor and mostly a capacitor is connected between this terminal and ground. It is called discharge terminal because when transistor saturates, capacitor discharges through the transistor. When the transistor is cut-off, the capacitor charges at a rate determined by the external resistor and capacitor.

**Pin 8: Supply Terminal:** A supply voltage of + 5 V to + 18 V is applied to this terminal with respect to ground (pin 1).
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.

Logic gates are electronic circuits that operate on one or more input signals to produce an output signal. The gates are blocks of hardware that produces the equivalent of logic 1 or logic 0 output signals if input logic requirements are satisfied. Gate INPUTS are driven by voltages having two nominal values, e.g. 0V and 5V representing logic 0 and logic 1 respectively. The OUTPUTS of a gate provides two nominal values of voltage only, e.g. 0V and 5V representing logic 0 and logic 1 respectively. There is always a time delay between an input being applied and the output responding. The different types of logic gates are as follows:

1. **NOT GATE**: It has one input and one output. The output is the complement of the input.
2. **OR GATE**: The gate has two inputs and one output. The output is logic ‘1’ when either of the inputs or both the inputs are at logic ‘1’.
3. **AND GATE**: The gate has two inputs and one output. The output is logic ‘1’ only when both the inputs are high.
4. **NAND GATE**: It is an AND gate followed by a NOT gate. It is the complement of AND gate. The output is logic ‘0’ when both the inputs are at logic ‘1’, else the output is always in the high state.
5. **NOR GATE**: It is an OR gate followed by a NOT gate. It is the complement of OR gate. The output is logic ‘1’ when both the inputs are at logic ‘0’, else the output is always in the low state.
6. **EXOR GATE**: It is logic gate whose output is in the high state when both the inputs are not same. When the both the inputs are high and when both are low, the output is low.
7. **EXNOR GATE**: It is logic gate whose output is in the high state when the both the inputs are high and when both are low. The output is low when both the inputs are not same.
**PROCEDURE:**

1. Place the IC in the socket of the trainer kit.
2. Make the connections for the gate as shown in the circuit diagram.
3. Verify the Truth Table.
4. Repeat the above steps for other gates in the different IC chips.

(1) **Implementation of Logic Gates**

1. **NOT Gate: - [IC7404]**

   ![Not Gate Symbol and Truth Table](image)
   
   **Truth Table**
   
<table>
<thead>
<tr>
<th>A</th>
<th>Output (A)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

2. **OR Gate: - [IC7432]**

   ![Or Gate Symbol and Truth Table](image)
   
   **Truth Table**
   
<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>Y</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

3. **AND GATE: - [IC7408]**

   ![And Gate Symbol and Truth Table](image)
   
   **Truth Table**
   
<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>Y</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

4. **NAND GATE: - [IC7400]**

   ![Nand Gate Symbol and Truth Table](image)
   
   **Truth Table**
   
<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>Y</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>
5. NOR GATE: - [IC 7402]

**Symbol**

\[ Y = A + B \]

**Truth Table**

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>Y</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

6. XOR GATE: - [IC 7486]

**Symbol**

\[ Y = A \oplus B \]

**Truth Table**

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>Y</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

7. EX-NOR GATE: - [IC 4077]

**Symbol**

\[ Y = \overline{A \oplus B} \]

**Truth Table**

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>Y</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

(2) Implementation of Basic Gates Using Universal Gates

**Theory:**
NAND and NOR gates are called as universal gates because all the other basic gates can be realized using only NAND or NOR gates.

**PROCEDURE:**
1. Place the IC in the socket of the trainer kit.
2. Make the connections for the gate as shown in the circuit diagram.
3. Verify the Truth Table.
4. Repeat the above steps for other gates in the different IC chips.
1. NAND GATE AS
   (a) AND GATE

   **LOGIC DIAGRAM**
   
   ![NAND gate diagram](image)
   
   **TRUTH TABLE**
   
<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>Y</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

(b) OR GATE

   **LOGIC DIAGRAM**
   
   ![OR gate diagram](image)
   
   **TRUTH TABLE**
   
<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>Y</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

(c) NOT GATE

   **LOGIC DIAGRAM**
   
   ![NOT gate diagram](image)
   
   **TRUTH TABLE**
   
<table>
<thead>
<tr>
<th>I/P (A)</th>
<th>D/P (A)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

(d) NOR GATE

   **LOGIC DIAGRAM**
   
   ![NOR gate diagram](image)
   
   **TRUTH TABLE**
   
<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>Y</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

(e) EX-OR GATE

   **LOGIC DIAGRAM**
   
   ![EX-OR gate diagram](image)
   
   **TRUTH TABLE**
   
<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>Y</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>
(f) **EX-NOR GATE**

- **LOGIC DIAGRAM**
- **TRUTH TABLE**

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>Y</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

2. **NOR Gate As**

(a) **AND GATE**

- **LOGIC DIAGRAM**
- **TRUTH TABLE**

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>Y</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

(a) **OR GATE**

- **LOGIC DIAGRAM**
- **TRUTH TABLE**

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>Y</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

(b) **NOT GATE**

- **LOGIC DIAGRAM**
- **TRUTH TABLE**

<table>
<thead>
<tr>
<th>I/P (A)</th>
<th>O/P (A)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>
(c) NAND GATE

**LOGIC DIAGRAM**

![NAND Gate Diagram]

**TRUTH TABLE**

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>Y</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

(d) EX-OR GATE

**LOGIC DIAGRAM**

![EX-OR Gate Diagram]

**TRUTH TABLE**

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>Y</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

(a) EX-NOR GATE

**LOGIC DIAGRAM**

![EX-NOR Gate Diagram]

**TRUTH TABLE**

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>Y</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>
Experiment No.1a

To design and implement a Schmitt trigger using Op-Amp for given UTP and LTP values & demonstrate its working.

Description:
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 UTP and lower threshold voltage 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.

Open loop gain of op amp is very high (ideally infinite). Any small difference between $V_{IN}$ and $V_{INV}$ results into saturation of output voltage $\pm V_{SAT}$. Value of $V_{SAT}$ is limited by the supply voltage of op amp.

Components Required:
Op amp IC µA 741, Resistor of 10KΩ, 110KΩ, DC regulated power Supply, trainer kit (+12v & -12v is given to Op amp from this), Signal generator, CRO.

Design:
From theory of Schmitt trigger circuit using op-amp, we have the trip points $UTP= \frac{R2 \times V_{ref}}{R1+R2} + \frac{R1 \times V_{sat}}{R1+R2}$

$LTP= \frac{R2 \times V_{ref}}{R1+R2} - \frac{R1 \times V_{sat}}{R1+R2}$

Where $V_{sat}$ is the positive saturation of the opamp=90% of $V_{cc}$

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

$UTP + LTP= \frac{2R2 \times V_{ref}}{R1+R2} -------(1)$

$UTP - LTP= \frac{2R1 \times V_{sat}}{R1+R2} -------(2)$

Let $V_{sat}=12v$, $UTP=4v$ and $LTP=2v$ then eq(2) yields

$R2=11R1$ From Eq(1) we have $V_{ref}=(UTP+LTP)(R1+R2) / 2R2= 3.27v$

Let $R1=10k$ then $R2= 110k$
Circuit Diagram for Schmitt Trigger

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(Vin) signal to the channel 1 and observe the output (Vo) 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.

Result waveform

Hysteresis curve

CRO in X-Y mode showing the Hysteresis curve
Experiment No.1b

To implement a Schmitt trigger using Op-Amp using a simulation package for two sets of UTP and LTP values & demonstrate its working.

Components to be placed in the schematic:
IC uA 741, Resistor of 10KΩ, 110KΩ, DC regulated power supply, Signal generator, CRO

Type of analysis: TIME DOMAIN (TRANSIENT)  Run to time: 100msec
step size: 0.1msec
UTP=4v  LTP=2v  Vref=3.3v

Hysteresis Curve:
To get the Hysteresis curve
1. Choose Axis settings from Plot menu
2. Under X-axis select axis variable
3. Select the input wave
4. Click Apply and ok
5. From the output check the UTP and LTP values
6. When Vin > UTP The o/p changes from +vsat(10v) to –vsat(-10v)
7. When vin< LTP the o/p changes from -vsat(10v) to +vsat(-10v)

If UTP= 1V and LTP= -3V then Vref= -2.2v

For UTP= -1V LTP= -3V then Vref=-2.2v

Hysteresis Curve
Experiment No.2a

Design and construct a rectangular waveform generator (op-amp relaxation Oscillator) for given frequency and demonstrate its working

**Description:**
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 \( \frac{R_1}{R_1+R_2} \) of output is fed back to the noninverting input terminal. Thus reference voltage is \( \frac{R_1}{R_1+R_2} V_o \). And may take values as + \( \frac{R_1}{R_1+R_2} \) Vsat or - \( \frac{R_1}{R_1+R_2} \) Vsat. 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.

**Components Required:**
Op-amp μA 741, Resistor of 10KΩ, 4.7KΩ, Capacitor of 0.1 μF, digital trainer kit (+12v & -12v is given to Op amp from this), CRO.

**Circuit Diagram**

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

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

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

Design for a frequency of 1 kHz (implies \( T = 1ms \) ) Let \( C = 0.1\mu F \)

Then calculating \( R \) as \( R = \frac{T}{2C \ln(3)} = 1 \times 10^{-3}/2 \times 0.1 \times 10^{-6} \times 1.099 = 5 \times 10^3 \)

= 5K Select \( R = 4.7K\Omega \)

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.

**Waveforms**

![Waveform Diagram](image)

**Result:**

The frequency of the oscillations = ___ Hz.
Experiment No.2b

To implement a rectangular waveform generator (Op-Amp relaxation oscillator) using a simulation package and observe the change in frequency when all resistor values are doubled.

Type of analysis: TIME DOMAIN (TRANSIENT)

Run to time: 10ms, Step size: 0.01ms

Waveforms with resistor values doubled

\[ T=2\text{ms} = \text{frequency }=500\text{hz} \]
Experiment No.3

To design and implement an astable multivibrator using 555 Timer for a given frequency and duty cycle.

Description:

Multivibrator is a form of oscillator, which has a non-sinusoidal output. The output waveform is rectangular. The multivibrators are classified as

i) **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.

ii) **Monostable or one shot multivibrators**: It has one stable state and one quasi stable. The application of an input pulse triggers the circuit time constants and the output goes to the quasi stable state, 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.

iii) **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.

Components Required:

555 Timer IC, Resistors of 3.3KΩ, 6.8KΩ, Capacitors of 0.1 μF, 0.01 μF, digital trainer kit(used to give +5v power supply to 555 IC),CRO.
**Design:**

Frequency = 1 kHz and duty cycle = 75%, \( RA = 7.2\, \text{k}\Omega \) & \( RB = 3.6\, \text{k}\Omega \),

Duty cycle = \( t_H / T = 0.75 \). Hence \( t_H = 0.75T = 0.75\text{ms} \) and \( t_L = T - t_H = 0.25\text{ms} \). Let \( C = 0.1\mu\text{F} \) and substituting in the above equations,

So \( RB = \frac{t_L}{0.693 \times C} = \frac{0.25 \times 10^{-3}}{0.693 \times 0.1 \times 10^{-6}} = 3.6\, \text{k}\Omega \)

\( RA = \frac{(t_H - 0.693 \times RB \times C)}{0.693 \times C} \)

\( = \frac{0.75 \times 10^{-3} \times 0.693 \times 3.6 \times 103 \times 0.1 \times 10^{-6}}{0.693 \times 0.1 \times 10^{-6}} = 7.2\, \text{k}\Omega \)

Choose \( RA = 6.8\, \text{k}\Omega \) and \( RB = 3.3\, \text{k}\Omega \).

**Note:**

The duty cycle determined by \( RA \) & \( RB \) can vary only between 50 % & 100 %. If \( RA \) is much smaller than \( RB \), the duty cycle approaches 50%.

**Circuit Diagram:**

Connect the pin 2 to the CRO to get the capacitor waveform check the amplitude from the waveform to get the UTP and LTP values.

Connect pin 3 to CRO to get the output. Find out the TH and TL values.

**Procedure:**

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

Given frequency \( f = 1 \text{KHz} \) and duty cycle = 60%

\( (=0.6) \) The time period \( T = \frac{1}{f} = 1 \text{ms} = t_H + t_L \)

Where \( t_H \) is the time the output is high and \( t_L \) is the time the output is low.

For an astable multivibrator using 555 Timer we have

\[
\begin{align*}
    t_L &= 0.693 \times R_B \times C \\
    t_H &= 0.693 \times (R_A + R_B) \times C
\end{align*}
\]

Duty cycle \( = \frac{t_H}{T} = 0.6 \). Hence \( t_H = 0.6 \times T = 0.6 \text{ms} \) and \( t_L = T - t_H = 0.4 \text{ms} \).

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

\[ R_B = 3.6 \text{ K\Omega} \text{ (from equation 2) and} \]

\[ R_A = 7.2 \text{ K\Omega} \text{ (from equation 1 \& R_B values).} \]

Choose \( R_A = 6.8 \text{k\Omega} \) and \( R_B = 3.3 \text{k\Omega} \).

The \( V_{cc} \) determines the upper and lower threshold voltages (observed from the capacitor voltage waveform) as \( V_{tr} = \frac{2}{3} V_{cc} \) & \( V_{t\ell} = \frac{1}{3} V_{cc} \).

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%.

Result:

The frequency of the oscillations = 1KHz.

Waveforms

\[ V_{UL} \]

\[ V_{LT} \]

\[ \text{O/P} \]

\[ t_r \quad t_c \]

Result:
Threshold voltage | \( V_{UT} \) | \( V_{LT} \)  
---|---|---  
theoretical | \( \frac{2}{3} \times V_{cc} = 3.3V \) | \( \frac{1}{3} \times V_{cc} = 1.6V \)  
practical | | |  

**Note:**

Each division in oscilloscope is

0.2 Time = no of div in x-axis x
time base Amplitude = no of div in
y-axis x volt/div Duty cycle =

\[(Ton/Ton + Toff) \times 100\]

<table>
<thead>
<tr>
<th>Duty cycle</th>
<th>Duty cycle</th>
<th>Ton</th>
<th>Toff</th>
</tr>
</thead>
<tbody>
<tr>
<td>Theoretical</td>
<td>75%</td>
<td>0.75ms</td>
<td>0.25ms</td>
</tr>
<tr>
<td>Practical</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Experiment No 4.

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

Components required:-

<table>
<thead>
<tr>
<th>Sl.No</th>
<th>Name of The Component</th>
<th>IC Number</th>
<th>Quantity</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>AND gate</td>
<td>7408</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>OR gate</td>
<td>7432</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>Not gate</td>
<td>7404</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>EXOR gate</td>
<td>7486</td>
<td>3</td>
</tr>
<tr>
<td>5</td>
<td>NAND gate</td>
<td>7400</td>
<td>3</td>
</tr>
<tr>
<td>6</td>
<td>NOR gate</td>
<td>7402</td>
<td>3</td>
</tr>
<tr>
<td>7</td>
<td>Patch chords</td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>Trainer Kit</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

(a) ADDER

Theory:
An Adder is a circuit which performs addition of binary numbers producing sum and carry. An half adder is a digital circuit which performs addition of two binary numbers which are one bit each and produces a sum and a carry(one bit each). A full adder is a digital circuit which performs addition of three binary numbers (one bit each), to produce a sum and a carry(one bit each). A Full adders are basic block of any adder circuit as they add two numbers along with the carry from the previous addition.

The basic rules of binary addition are:

\[
\begin{align*}
0+0 &= 0 \\
0+1 &= 1 \\
1+0 &= 1 \\
1+1 &= (10)_2
\end{align*}
\]

Column by column addition, similar to decimal addition is performed a logic circuit known as half adder adds two 1 bit signals. In actual addition there is often a third bit, the carry bit that must be added. Hence to add 3 bits at a time a logic circuit known as a Full adder is used.
Half Adder

(a) Half Adder Using Logic Gates

**Block Diagram**

**Logic Diagram**

**Truth Table**

(b) Half Adder Using NAND Gates

(c) Half Adder Using NOR Gates

Full Adder

**Block Diagram**
(a) Full Adder Using Logic Gates

\[
\text{SUM} = \overline{A} \overline{B} \overline{Cin} + \overline{A} B \overline{Cin} + A \overline{B} \overline{Cin} + A B \overline{Cin}
\]

\[
= (\overline{A} B + A B)Cin + (A B + A B)\overline{Cin}
\]

\[
\text{SUM} = (A \oplus B)Cin + (A \oplus B)\overline{Cin}
\]

\[
= A \oplus B \oplus C
\]

\[
c\text{arry} = (A \oplus B) \text{Cin} + A B
\]

\[
= A B + B C + C A
\]

(b) Full Adder Using NAND Gates

(c) Full Adder Using NOR Gates

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>Cin</th>
<th>Sum</th>
<th>Cout</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>
Procedure:

1. Place the IC in the socket of the trainer kit.
2. Make the connections as shown in the circuit diagram.
3. Verify the truth table for half adder and full adder circuits using basic and universal gates.

(b) SUBTRACTORS

Theory:

Subtractors are digital circuits which perform subtraction of binary numbers to produce a difference and a borrow if any. A half subtractor subtracts two one bit numbers to give their difference and a borrow if any. A full subtractor subtracts two one bit numbers along with a borrow (from previous stage) to generate a difference and a borrow.

Half Subtractor

Block Diagram

(a) Half Subtractor using Logic Gates

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>Diff</th>
<th>Borrow</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Logic Diagram

Difference = A ⊕ B

Borrow = \overline{A}B

(b) Half Subtractor Using NAND gates

Difference = A ⊕ B

Borrow = \overline{A}B
(c) Half Subtractor Using NOR gates

![Half Subtractor Using NOR gates diagram]

Borrow = \( \bar{A}\bar{B} \)

---

Full Subtractor

**Block Diagram**

![Full Subtractor Block Diagram]

(a) Full Subtractor Using Logic Gates

\[
\text{Diff} = \overline{A} \overline{B} \text{Bin} + A \overline{B} \text{Bin} + A \overline{B} \text{Bin} + A B \text{Bin} \\
= (\overline{A} \overline{B} + A B) \text{Bin} + (A B + A \overline{B}) \text{Bin} \\
= (A \oplus B) \text{Bin} + (A \oplus B) \overline{\text{Bin}} \\
\text{Diff} = A \oplus B \oplus \text{Bin}
\]

\[
\text{Bout} = \overline{A} \overline{B} \text{Bin} + A \overline{B} \text{Bin} + A \overline{B} \text{Bin} + A B \text{Bin} \\
= \overline{A} [B \text{Bin} + B \overline{\text{Bin}}] + [A + \overline{A}] B \text{Bin} \\
\text{Bout} = \overline{A} (B \oplus \text{Bin}) + B \text{Bin}
\]

![Full Subtractor Using Logic Gates diagram]

Truth Table

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>Bin</th>
<th>Diff</th>
<th>Borrow</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

(b) Full Subtractor Using NAND Gates

![Full Subtractor Using NAND Gates diagram]
(c) Full Subtractor Using NOR Gates

Procedure:
1. Place the IC in the socket of the trainer kit.
   (complex boolean Expression’s are simplified by using Kmaps).
2. Make the connections as shown in the circuit diagram.
3. Verify the truth table for half subtractor and full subtractor circuits using basic and universal gates.

   Result: Realized both half Adder and Subtractor and full Adder and Subtractor circuits using basic and universal.
Experiment No.5a

Given any 4-variable logic expression, simplify using Entered Variable Map and realize the simplified logic expression using 8:1 multiplexer IC.

Description:
The term multiplex means “many to one”. A multiplexer (MUX) has n inputs. Each line is used to shift digital data serially. There is a single output line. One of the data stored in the n input line is transferred to the output based on the valued of control bits. An n to 1 multiplexer requires m control bits where \( n \leq 2^m \).

To construct an 4 variable function we require a 16(\(2^4\)) to 1 multiplexer, whereas using an entered variable map method a 4 variable expression can be realized using 8(\(2^3\)) to 1 multiplexer

Components Used:
IC 74 LS151, patch chords, power chords, trainer kit.

<table>
<thead>
<tr>
<th>S.No</th>
<th>Components</th>
<th>Quantity</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Trainer Kit</td>
<td>01</td>
</tr>
<tr>
<td>2</td>
<td>Patch Chords</td>
<td>20</td>
</tr>
<tr>
<td>3</td>
<td>IC74151</td>
<td>01</td>
</tr>
<tr>
<td>4</td>
<td>IC7404</td>
<td>01</td>
</tr>
</tbody>
</table>

Pin Diagrams: IC 74LS151
Example:

Simplify the following function using EVM technique

\[ f(a,b,c,d) = \sum m(2,3,4,5,13,15) + \sum d(8,9,10,11) \]

<table>
<thead>
<tr>
<th>Decimal</th>
<th>ABCD</th>
<th>f</th>
<th>MEV entry</th>
<th>Data</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0000</td>
<td>0</td>
<td>0</td>
<td>Do</td>
</tr>
<tr>
<td>1</td>
<td>0001</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>0010</td>
<td>1</td>
<td>1</td>
<td>D1</td>
</tr>
<tr>
<td>3</td>
<td>0011</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>0100</td>
<td>1</td>
<td>1</td>
<td>D2</td>
</tr>
<tr>
<td>5</td>
<td>0101</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>0110</td>
<td>0</td>
<td>0</td>
<td>D3</td>
</tr>
<tr>
<td>7</td>
<td>0111</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>1000</td>
<td>X</td>
<td></td>
<td>D4</td>
</tr>
<tr>
<td>9</td>
<td>1001</td>
<td>X</td>
<td></td>
<td>D5</td>
</tr>
<tr>
<td>10</td>
<td>1010</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>1011</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>1100</td>
<td>0</td>
<td>D</td>
<td>D6</td>
</tr>
<tr>
<td>13</td>
<td>1101</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>1110</td>
<td>0</td>
<td>D</td>
<td>D7</td>
</tr>
<tr>
<td>15</td>
<td>1111</td>
<td>1</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Circuit Diagram

Procedure:
1. Verify all components & patch chords whether they are in good condition or not.
2. Make connections 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 & observe outputs.

Result:
The truth table is verified
Experiment No.5b

Write the verilog /VHDL code for 8:1 MULTIPLEXER. Simulate and verify its working.

Description:
An 8:1 multiplexer has 8 inputs and one output. The data stored in one of these 8 input lines is transferred serially to the output based on the value of the selection bits.

Truth table:

<table>
<thead>
<tr>
<th>SEL (2)</th>
<th>SEL (1)</th>
<th>SEL (0)</th>
<th>Zout</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>I(0)</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>I(1)</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>I(2)</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>I(3)</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>I(4)</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>I(5)</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>I(6)</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>I(7)</td>
</tr>
</tbody>
</table>

Algorithm for 8 to 1 multiplexer: Input/output
I is a one dimensional array of size 8 Sel is a one dimensional array of size 3 Zout is output

Method
If the value of sel is 000 zout = I[0] If the value of sel is 001 zout = I[1] If the value of sel is 010 zout = I[2] If the value of sel is 011 zout = I[3]
If the value of sel is 100 zout = I[4] If the value of sel is 101 zout = I[5] If the value of sel is 110 zout = I[6] Else the value of zout is I[7]

VHDL code for 8 to 1 MUX (behavioral modeling):
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;  // includes the standard library

entity mux1 is
    Port ( I : in std_logic_vector(7 downto 0);
           sel : in std_logic_vector(2 downto 0); // Input and output is declared as
           ports zout : out std_logic);
end mux1;

architecture Behavioral of mux1 is begin
    zout<= I(0) when sel="000" else  // Based on the value of selection the value
           // of data I(1) when sel="001" else  // stored in the array I is stored in
           zout
           I(2) when sel="010" else I(3) when sel="011" else I(4) when sel="100" else
           I(5) when sel="101" else I(6) when sel="110" else I(7);
end Behavioral;

OUTPUT:
Experiment No 6.

Design and implement code converter I) Binary to Gray II) Gray to Binary

Code using basic gates.

Description:

Gray Code is one of the most important codes. It is a non-weighted code which belongs to a class of codes called minimum change codes. In this codes while traversing from one step to another step only one bit in the code group changes. In case of Gray Code two adjacent code numbers differs from each other by only one bit.

Binary to gray code conversion is a very simple process. There are several steps to do this types of conversions.

Steps given below elaborate on the idea on this type of conversion.

1. The M.S.B. of the gray code will be exactly equal to the first bit of the given binary number.

2. Now the second bit of the code will be exclusive-or of the first and second bit of the given binary number, i.e if both the bits are same the result will be 0 and if they are different the result will be 1.

3. The third bit of gray code will be equal to the exclusive-or of the second and third bit of the given binary number. Thus the Binary to gray code conversion goes on. One example given below can make your idea clear on this type of conversion.

Gray code to binary conversion is again very simple and easy process. Following steps can make your idea clear on this type of conversions.

1. The M.S.B. of the binary number will be equal to the M.S.B of the given gray code.

Now if the second gray bit is 0 the second binary bit will be same as the previous or the first bit. If the gray bit is 1 the second binary bit will alter. If it was 1 it will be 0 and if it was 0 it will be 1

1. This step is continued for all the bits to do Gray code to binary
conversion. One example given below will make your idea clear.

Components required:
IC 7486, Patch Cords & digital trainer Kit.

<table>
<thead>
<tr>
<th>Sl.No</th>
<th>COMPONENT</th>
<th>SPECIFICATION</th>
<th>QTY.</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>X-OR GATE</td>
<td>IC 7486</td>
<td>1</td>
</tr>
<tr>
<td>2.</td>
<td>AND GATE</td>
<td>IC 7408</td>
<td>1</td>
</tr>
<tr>
<td>3.</td>
<td>OR GATE</td>
<td>IC 7432</td>
<td>1</td>
</tr>
<tr>
<td>4.</td>
<td>NOT GATE</td>
<td>IC 7404</td>
<td>1</td>
</tr>
<tr>
<td>5.</td>
<td>IC TRAINER KIT</td>
<td>-</td>
<td>1</td>
</tr>
<tr>
<td>6.</td>
<td>PATCH CORDS</td>
<td>-</td>
<td>35</td>
</tr>
</tbody>
</table>

1) BINARY TO GRAY CONVERSION

\[
\begin{array}{cccc}
0 & 0 & 1 & 1 \\
0 & 0 & 1 & 1 \\
0 & 0 & 1 & 1 \\
0 & 0 & 1 & 1 \\
0 & 0 & 1 & 1 \\
0 & 1 & 0 & 1 \\
0 & 1 & 0 & 1 \\
0 & 1 & 0 & 1 \\
0 & 1 & 0 & 1 \\
0 & 1 & 0 & 1 \\
0 & 1 & 0 & 1 \\
0 & 1 & 1 & 0 \\
0 & 1 & 1 & 0 \\
1 & 0 & 0 & 1 \\
1 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 \\
1 & 1 & 1 & 1 \\
\end{array}
\]
BOOLEAN EXPRESSIONS:
G3=B3  
G2=B3 ⊕ B2

G1=B1 ⊕ B2  
G0=B1 ⊕ B0

**Circuit Diagram:** BINARY TO GRAY CODE
TRUTH TABLE:

I) RAY TO BINARY CONVERSION

<p>| | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

B3 = G3

<p>| | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

B2 = G3 ⊕ G2

<p>| | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

B1 = G3 ⊕ G2 ⊕ G1

<p>| | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

B0 = G3 ⊕ G2 ⊕ G1 ⊕ G0

BOOLEAN EXPRESSIONS:

B3 = G3  B2 = G3 ⊕ G2  B1 = G3 ⊕ G2 ⊕ G1  B0 = G3 ⊕ G2 ⊕ G1 ⊕ G0

Circuit Diagram: Gray to Binary
TRUTH TABLE:

<table>
<thead>
<tr>
<th>Gray</th>
<th>Binary</th>
</tr>
</thead>
<tbody>
<tr>
<td>G3</td>
<td>G2</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

PROCEDURE:

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

RESULT: Binary to gray code conversion and vice versa is realized using EX-OR gates.
Experiment No 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.

Description:
The parity generating technique is one of the most widely used error detection techniques for the data transmission. In digital systems, when binary data is transmitted and processed, data may be subjected to noise so that such noise can alter 0s (of data bits) to 1s and 1s to 0s. Hence, parity bit is added to the word containing data in order to make number of 1s either even or odd. Thus it is used to detect errors, during the transmission of binary data. The message containing the data bits along with parity bit is transmitted from transmitter node to receiver node.

I) 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.

Even Parity Generator
Let us assume that a 3-bit message is to be transmitted with an even parity bit. Let the three inputs A, B and C are applied to the circuits and output bit is the parity bit P. The total number of 1s must be even, to generate the even parity bit P.

II) Parity Checker
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).

Components Required:
IC 7486, Patch Cords & Digital Trainer Kit.

PARITY GENERATOR: TRUTH TABLE:
The K-map simplification for 3-bit message even parity generator is

<table>
<thead>
<tr>
<th>3-bit message</th>
<th>Even parity bit generator (P)</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>B</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>
From the above truth table, the simplified expression of the parity bit can be written as

\[
P = \overline{A} B C + \overline{A} B \overline{C} + A \overline{B} \overline{C} + A B C
\]

\[
= \overline{A} (B C + B \overline{C}) + A (\overline{B} \overline{C} + B C)
\]

\[
= \overline{A} (B \oplus C) + A (\overline{B} \oplus C)
\]

\[
P = A \oplus B \oplus C
\]

To generate the even parity bit for a 4-bit data, three Ex-OR gates are required to add the 4-bits and their sum will be the parity bit.

**Circuit diagram: parity generator**

1) **PARITY CHECKER:**

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.
The above truth table can be simplified using K-map as shown below.

\[
P_{EC} = \overline{A} \overline{B} (C \overline{D} + \overline{C} \overline{D}) + \overline{A} B (C \overline{D} + C \overline{D}) + A B (C \overline{D} + C \overline{D}) + A \overline{B} (C \overline{D} + C \overline{D})
\]
\[
= \overline{A} \overline{B} (C \oplus D) + \overline{A} B (C \oplus D) + A B (C \oplus D) + A \overline{B} (C \oplus D)
\]
\[
= (\overline{A} B + A \overline{B}) (C \oplus D) + (A B + A \overline{B}) (C \oplus D)
\]
\[
= (A \oplus B) \oplus (C \oplus D)
\]

The above logic expression for the even parity checker can be implemented by using three Ex-OR gates as shown in figure. If the received message consists of five bits, then one more Ex-OR gate is required for the even parity checking.

**Circuit diagram:**
PROCEDURE:

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

RESULT: 3 bit parity generator and 4 bit parity checker is verified.
Experiment No.8a:

Realize a J-K Master/Slave FF using NAND gates and verify its truth table

Description:
A flip-flop is a device very much like a latch in that it is a bistable multivibrator, having two states and a feedback path that allows it to store a bit of information. The difference between a latch and a flip-flop is that a latch is asynchronous, and the outputs can change as soon as the inputs do (or at least after a small propagation delay). A flip-flop, on the other hand, is edge-triggered and only changes state when a control signal goes from high to low or low to high.

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.

Components used:
IC 74 LS00, IC 74LS10, patch chords, trainer kit.

<table>
<thead>
<tr>
<th>S.No</th>
<th>Components</th>
<th>Quantity</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Trainer kit.</td>
<td>01</td>
</tr>
<tr>
<td>2</td>
<td>Patch chords</td>
<td>03</td>
</tr>
<tr>
<td>3</td>
<td>IC 74LS00</td>
<td>01</td>
</tr>
<tr>
<td>4</td>
<td>IC 74LS10</td>
<td>02</td>
</tr>
</tbody>
</table>
Truth table:

<table>
<thead>
<tr>
<th>Clk</th>
<th>J</th>
<th>K</th>
<th>Q</th>
<th>Q bar</th>
<th>comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>Q0</td>
<td>Q bar</td>
<td>No change</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>Reset</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>Set</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>Q0</td>
<td>Q0</td>
<td>toggle</td>
</tr>
</tbody>
</table>

CIRCUIT DIAGRAM:

Procedure:
1. Verify all components & patch chords whether they are in good condition or not
2. Make connections 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 & observe outputs.

Conclusion: Truth table is verified
Experiment No.8b

Write the verilog/ VHDL code for D Flip-Flop with positive- edge triggering. Simulate and verify its working.

Description:
D- flip flop is a data flip flop the truth table summarizes the operation of the D – flip flop. It has a single input D and two output Q and Q’

Truth table:

<table>
<thead>
<tr>
<th>clk</th>
<th>D(input)</th>
<th>Q(output)</th>
<th>Q bar</th>
</tr>
</thead>
<tbody>
<tr>
<td>-</td>
<td>0</td>
<td>No change</td>
<td></td>
</tr>
<tr>
<td>-</td>
<td>1</td>
<td>No change</td>
<td></td>
</tr>
<tr>
<td>↑</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>↑</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

Algorithm:
Clock and D is the input to the flip flop, Q and Qbar is the output of the system For every positive transition of the clock
1. The value of D is stored in Q and the
2. The value of complement of D is stored in Qbar

VHDL code for D Flip Flop counter:
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
entity dflip is
Port ( D,Clk : in std_logic;  // D, clk is declared as
input ports Q : out std_logic;  // Q and Qbar is declared
as output ports
Qbar : out
std_logic:=‘1’); end dflip;
arithmetic Behavioral of
dflip is begin
process(clk) // process uses for sequential circuits begin
if rising_edge(clk)
   then Q<= D;
   Qbar<= not D; end
if;
end process; end Behavioral;

Results:

<table>
<thead>
<tr>
<th>Name</th>
<th>Value at 16.23 ns</th>
</tr>
</thead>
<tbody>
<tr>
<td>clk</td>
<td>A 0</td>
</tr>
<tr>
<td>D</td>
<td>A 1</td>
</tr>
<tr>
<td>D</td>
<td>A 0</td>
</tr>
<tr>
<td>Qbar</td>
<td>A 1</td>
</tr>
</tbody>
</table>
Experiment No.9a

Design and implement a mod n (n<8) synchronous up counter using JK Flip Flop ICs and demonstrate its working.

Description:
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.

Components Used:
IC 74 LS76, IC 74LS08, patch chords, trainer kit.

Pin Diagrams: 74LS76

Synchronous counter design:
To successfully design synchronous counters we may employ the following six basic steps:

1. Create the state transition diagram.

2. Create a present state-next state table (often referred to as the next state table).

3. Expand the table to form the transition table for each flip-flop in the circuit. The transition table shows the flip-flop inputs required to make the counter go from present state to the desired next state. This is also referred to as the excitation table.
4. Determine the logic functions of the J and K inputs as a function of the present states.
5. Analyse the counter to verify the design.
6. Construct and test the counter.

**Function Table:**

<table>
<thead>
<tr>
<th>Inputs</th>
<th>Outputs</th>
</tr>
</thead>
<tbody>
<tr>
<td>PR</td>
<td>CLR</td>
</tr>
<tr>
<td>L</td>
<td>H</td>
</tr>
<tr>
<td>H</td>
<td>L</td>
</tr>
<tr>
<td>L</td>
<td>L</td>
</tr>
<tr>
<td>H</td>
<td>H</td>
</tr>
<tr>
<td>H</td>
<td>H</td>
</tr>
<tr>
<td>H</td>
<td>H</td>
</tr>
<tr>
<td>H</td>
<td>H</td>
</tr>
</tbody>
</table>

---

**Functional Truth Table for J-K Flip Flop:**

<table>
<thead>
<tr>
<th>J</th>
<th>K</th>
<th>Qn</th>
<th>Qn+1</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

---

**State Synthesis Table for JK Flip Flop**

<table>
<thead>
<tr>
<th>Present state</th>
<th>Next state</th>
<th>J</th>
<th>K</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>X</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>X</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>X</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>X</td>
<td>0</td>
</tr>
</tbody>
</table>
Experiment No. 9b

Write the verilog/ VHDL code for mod-8 up counter. Simulate and verify its working.

Description:
A modulus 8 counter has 8 unique states. The input of counter is summarized in the truth table below.

Truth Table:

<table>
<thead>
<tr>
<th>rst</th>
<th>clock</th>
<th>Q2</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>x</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Algorithm:

Input : clock and reset
Output: Q
Method:
Execute the code repeated if clk or rst changes begin
1. If rst is high counter remains stable at state 0
2. If rst is low and if clock occurs the counter progresses through state 0 to 7 end

VHDL Code for MOD-8 Counter:

```vhdl
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_ARITH.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;

entity mod8 is
    Port ( rst, clk : std_logic ;
            q : buffer std_logic_vector(2 downto 0) := "000" );
end mod8;

architecture Behavioral of mod8 is
begin

end Behavioral;
```


as it is end mod8;  // used for input and as output and it should store a
value architecture Behavioral of mod8 is
process(clk,rst) is // process uses for sequential circuits executes the code with in if clk or
rst
begin  //changes
if (CLK'event and clk='0') then
if(rst='1') then q<="000";
else
q <= q+1;
end if;
end if;
end process; end Behavioral;

**Result:**

<table>
<thead>
<tr>
<th>clk</th>
<th>A 0</th>
</tr>
</thead>
<tbody>
<tr>
<td>q</td>
<td>A[5]</td>
</tr>
</tbody>
</table>
| q[2] A 1
| q[1] A 0
| q[0] A 1
| rst A 0

![Waveform Diagram]

---

Dept. of CS&E, BMSIT&M

Page 70
Experiment No.10

Design and implement asynchronous counter using decade counter IC to count up from 0 to n (n≤9) and demonstrate on seven segment display (using IC-7447).

Description:
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.

Components Used:
IC 74 LS90, IC 7447 (BCD to seven segment decoder), patch chords, trainer kit.
1. Cp0 (pin 14) to be connected to clock
2. Cp1 (pin 1) to be connected to Q0. (The output of the first flip flop drives the second clock)

Pin Diagram: 74LS90 / 74LS47

Pin Names Description of 7447:
A0–A3 = BCD Inputs
RBI = Ripple Blanking Input (Active
LOW) LT= Lamp Test Input (Active LOW)
RBO =Ripple Blanking Output (Active LOW) a –g =Segment Outputs (Active LOW)

**Pin Names Description of 7490:**
R1 and R2-clear all flipflop (high active and low for not active) S1 and S2- set all flip flop (high active and low for not active) CLKA-clock pulse to first flip flop
CLKB-clock pulse to second flip flop (output of first flip flop clock for second filp flop)

**Circuit Diagram**

For mod 9
connect Q0 and Q3 to reset(clear) through an AND gate. Reset should not be connected to the switch

For mod8 : Connect Q3 to reset
For mod7 : Connect Q2, Q1,Q0 to reset through an And Gate
For Mod 6 : Connect Q2 and Q1 to reset through an AND gate
For mod 5: Connect Q0 and Q2 to reset through an AND gate

For Mod 4 Connect Q2 to reset For mod 3

Connect Q1 and Q0 to reset through an AND gate

For mod 2 Connect Q1 to reset Function

Table:

<table>
<thead>
<tr>
<th>Clock</th>
<th>Q3</th>
<th>Q2</th>
<th>Q1</th>
<th>Q0</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>6</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>7</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>8</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>9</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

Procedure:
1. Verify all components & patch chords whether they are in good condition or not.
2. Make connections 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 & observe outputs.

Result:
mod n<=9 counter implemented using the decade counter Ic
Experiment No.11

Generate a Ramp output waveform using DAC0800 (Inputs are given to DAC through IC74393 dual 4-bit binary counter)

Description:

DAC 0800

- To convert the digital signal to analog signal a Digital-to-Analog Converter (DAC) has to be employed.
- The DAC will accept a digital (binary) input and convert to analog voltage or current.
- Every DAC will have "n" input lines and an analog output.
- The DAC require a reference analog voltage (Vref) or current (Iref) source.
- The smallest possible analog value that can be represented by the n-bit binary code is called resolution.
- The resolution of DAC with n-bit binary input is 1/2^n of reference analog value.
- Every analog output will be a multiple of the resolution.
- The DAC0800 require a positive and a negative supply voltage in the range of ±5V to ±18V.
- It can be directly interfaced with TTL, CMOS, PMOS and other logic families.
- For TTL input, the threshold pin should be tied to ground (VLC =0V).
- The reference voltage and the digital input will decide the analog output current, which can be converted to a voltage by simply connecting a resistor to output terminal or by using an op-amp I to V converter.

Components required:

IC DAC0800, IC 74LS393, OPAMP IC µA741, patch chords, digital trainer kit, bread board, single stand wires.
Pin diagram & block diagram of dac0800

Pin diagram of 74LS393 binary counter IC

Connection diagram:
\[2^n = 2^8 = 256 \text{ different o/p levels in } 2^8 - 1 = 255 \text{ steps.}

\textbf{Result :}

\[V_o\]
Study experiment

Experiment No 12

To study 4-bit ALU using IC-74181.

Description:
ALU stands for the arithmetic and logical unit and is one of the important units in almost all the calculating machines these days be it with the hand-held mobile, or computers. All the computational work in the system are carried out by this unit. The typical ALU sizes are:
- 4-bit ALU: ALU that processes two 4-bit numbers.
- 8-bit ALU: ALU that processes two 8-bit numbers.
- Still in the latest systems ALU sizes are 16, 32, 64-bit etc.

Figure 12.1 shows the block diagram of a typical ALU.

In figure 12.1, the 1x2 selector on the left is as a mode selector to select one of the two units i.e. either the arithmetic unit or the logical unit. The function select lines are then used to select one of the many functions of arithmetic or the logical type.

MSI package for ALU:- IC 74181 a 4-bit Arithmetic and logical unit:
Function of ALU as seen from figure 12.3:

1. The ALU has two 4-bit input lines A3-A0, B3-B0, and a 4-bit function select lines S3-S0.
2. One mode select line „M” that is used to select ALU for either arithmetic or the logical function.

3. It has four output lines f3-f0.

4. Carry-in Cn is used in cascade mode.

5. When the size is to be increased to 8-bit operations two 74181 can be cascaded and Cn+4 will be used as input to Cn line of next stage. The cascading is shown in figure-12.4.

Figure-12.4: Cascading of ALU IC 74181 ALU

The function table of IC 74181 ALU is given in figure-12.5. It shows the functions selected depending on the Cn, M and S3-S0 lines.

Figure-12.5: Function Table
**Equipment Required:**
IC 74181, LEDs, Power Supply, CRO, Multimeter

**Circuit diagram:**
LEDs are connected at the input A and B lines and the select lines to indicate the value of the inputs A and B. The LEDs at the select lines are used to specify the function of the ALU. The LEDs at the output are used to test and verify the output. The whole implementation is shown in figure 12.6 is

![Circuit Diagram](image)

**Procedure:**
1. Keep the datasheet of IC 74181 ready.
2. Insert the IC on the Breadboard.
3. Make connections as shown in figure-12.6
4. Verify the connections

**Result:**
The above circuit when connected to power supply gives correct result as per the function table.
Sample Viva Questions

1. Why operational amplifier is called by its name?
2. Explain the advantages of OPAMP over transistor amplifiers.
3. List the OPAMP ideal characteristics.
4. Give the symbol of OPAMP
5. Explain the various applications of OPAMP
6. Define UTP and LTP
7. Mention the applications of schmitt trigger
8. What is a square wave generator/ Regenerative comparator?
9. Give the hysterisis curve of a schmitt trigger
10. What is a bipolar and unipolar devices? Give examples
11. Define resolution
12. Explain the need of D/A and A/D converters.
13. List the different types of A/D and D/A converters
14. What is a multivibrators?
15. What is a bistable multivibrators?
16. Give the applications of monostable and astable multivibrators
17. Explain the working of 555 timer as astable and monostable multivibrator
18. Why astable multivibrator is called as free running multivibrator
19. Define duty cycle.
20. List the applications of 555 timer
21. Explain 555 timer as astable multivibrator to generate a rectangular wave of duty cycle of less than 0.5
22. Define a logic gate.
23. What are basic gates?
24. Why NAND and NOR gates are called as universal gates?
25. State De morgans theorem
26. Give examples for SOP and POS
27. Explain how transistor can be used as NOT gate
28. Realize logic gates using NAND and NOR gates only
29. List the applications of EX-OR and EX-NOR gates
30. What is a half adder?
31. What is a full adder?
32. Differentiate between combinational and sequential circuits. Give examples
33. Give the applications of combinational and sequential circuits
34. Define flip flop
35. What is an excitation table?
36. What is race around condition?
37. How do you eliminate race around condition?
38. What is minterm and max term?
39. Define multiplexer/data selector
40. What is a demultiplexer?
41. Give the applications of mux and demux
42. What is an encoder and decoder?
43. Comparemux and encoder
44. Comparerdemux and decoder
45. What is a priority encoder?
46. What are counters? Give their applications.
47. Compare synchronous and asynchronous counters
48. What is modulus of a number?
49. What is a shift register?
50. What does LS stand for, in 74LS00?
51. What is positive logic and negative logic?
52. What are code converters?
53. What is the necessity of code conversions?
54. What is gray code?
55. Realize the Boolean expressions for
   a Binary to gray code conversion
   b Gray to binary code conversion
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 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?