UNDERGRADUATE COMPUTER ENGINEERING LABORATORY MANUAL

(VOL. 1)

Electric Circuit Theory

Analog Electronic Circuit

Software Development Technique





UNIVERSITY OF ILORIN

FACULTY OF ENGINEERING & TECHNOLOGY

DEPARTMENT OF COMPUTER ENGINEERING

ISBN: 978-978-8090-84-2

© April 2016

A.T. Ajiboye
Department of Computer Engineering
University of Ilorin

J.F. Opadiji
Department of Computer Engineering
University of Ilorin

S.A. Olatunji
Department of Computer Engineering
University of Ilorin

O. O. Labiran
Department of Computer Engineering
University of Ilorin

H.O. Mahmud
Department of Computer Engineering
University of Ilorin

O.F. Adebayo
Department of Computer Engineering
University of Ilorin

Foreword

This instruction manual is the first in the series of the undergraduate Computer Engineering laboratory manuals designed to facilitate a practical-oriented approach to learning by students of Computer Engineering in Nigerian universities. In developing this manual, requirements in the Basic Minimum Academic Standard (BMAS) as defined by both the National Universities Commission (NUC) and the Council for the Regulation of Engineering in Nigeria (COREN) for courses to be audited by Computer Engineering undergraduates were considered.

This volume focuses on experiments in Electric Circuit Theories, Analog Electronic Circuits and Software Development Techniques. A total of six laboratory sessions are to be performed by students in each of these areas.
At the end of each laboratory session, each student is expected to write a report on the experiments conducted in the workbook provided and submit to laboratory instructors for assessment.

It is our desire that the use of this manual in undergraduate Computer Engineering laboratories will achieve our goal of imparting students with the necessary hands-on experience that will make them productive Computer Engineers in the future.

J.F. Opadiji, Dr.Eng, R.Eng.

Table of Contents

Student Information Page	5
Attendance Page	6
Laboratory Safety Rules and Operating Procedures	7
General Laboratory Safety Information	8
Troubleshooting Hints	9
Guidelines for Laboratory Report Notebook	10
Laboratory Grading Policy	12
UILCPE/ECT 1 – Direct Current (DC) Measurement	13
UILCPE/ECT 2 – Alternating Current (AC) Measurement	18
UILCPE/ECT 3 – Network Analysis Methods	23
UILCPE/ECT 4 – First Order Circuits	27
UILCPE/ECT 5 – Sinusoidal Steady State	31
UILCPE/ECT 6 – Series and Resonance	34
UILCPE/AEC 1 – Diode Clipping Circuits	38
UILCPE/AEC 2 – Clamping Circuits	42
UILCPE/AEC 3 – Rectifier Circuits	46
UILCPE/AEC 4 – Transistor Biasing	54
UILCPE/AEC 5 – Transistor Small Signal Amplifiers	57
UILCPE/AEC 6 – Transistor as a Switch	61
UILCPE/SDT 1 – Introduction to Computer Programming	65
UILCPE/SDT 2 – Java IDE	67
UILCPE/SDT 3 – Flow Control Arrays and Classes	71
UILCPE/SDT 4 – Methods and Interfaces	77
UILCPE/SDT 5 – Input/Output Streams in Java	81
UILCPE/SDT 6 – Graphical User Interface	84
APPENDIX 1 – Standard Resistor Colour Code	88

STUDENT INFORMATION PAGE



Student Passport Photograph

SURNAME: _____

FIRST NAME: _____

OTHER NAME: _____

MATRICULATION NUMBER: _____

ACADEMIC SESSION: _____

CURRENT LEVEL: _____

ATTENDANCE PAGE

DATE	EXPERIMENT NUMBER	EXPERIMENT TITLE	SIGNATURE (Technologist-In-Charge)
	UILCPE/ECT 1	Direct Current (DC) Measurement	
	UILCPE/ECT 2	Alternating Current (AC) Measurement	
	UILCPE/ECT 3	Network Analysis Methods	
	UILCPE/ECT 4	First Order Circuits	
	UILCPE/ECT 5	Sinusoidal Steady State	
	UILCPE/ECT 6	Series and Resonance	
	UILCPE/AEC 1	Diode Clipping Circuits	
	UILCPE/AEC 2	Clamping Circuits	
	UILCPE/AEC 3	Rectifier Circuits	
	UILCPE/AEC 4	Transistor Biasing	
	UILCPE/AEC 5	Transistor Small Signal Amplifiers	
	UILCPE/AEC 6	Transistor as a Switch	
	UILCPE/SDT 1	Introduction to Computer Programming	
	UILCPE/SDT 2	Java IDE	
	UILCPE/SDT 3	Flow Control Arrays and Classes	
	UILCPE/SDT 4	Methods and Interfaces	
	UILCPE/SDT 5	Input/Output Streams in Java	
	UILCPE/SDT 6	Graphical User Interface	

LABORATORY SAFETY RULES AND OPERATING PROCEDURES

1. Note the location of the Circuit Breaker to shut off power in an emergency. Note the location of the nearest exit to your location.
2. Students are allowed in the laboratory only when the laboratory instructor (Technologist) is present.
3. Food and drinks are strictly prohibited in the laboratory. In the case where it becomes necessary to drink water, keep open drinks away from the laboratory benches.
4. Report any broken equipment or defective parts to the laboratory instructor. Do not open, remove the cover, or attempt to repair any equipment.
5. When the laboratory exercise is over, all equipment must be turned off and returned to the designated location.
6. All equipment, tool and instruments are the property of the University and must not be taken out of the laboratory.
7. Do not move instruments from one laboratory station to another laboratory station.
8. Do not tamper with or remove security straps, locks, or other security devices.
9. When touching the FPGA development boards or any other circuit boards, please do not touch the solid-state parts on the board but handle the board from its edge.
10. Lateness to laboratory sessions will not be condoned. Students who arrive later than thirty (30) minutes into the session will not be allowed into the laboratory.

I have read and understood these rules and procedures. I agree to abide by the rules and procedures at all times while in the laboratory, failure to follow these rules and procedures will result in my immediate dismissal from the laboratory and additional disciplinary action may be taken.

_____ _____

SIGNATURE **DATE**

GENERAL LABORATORY SAFETY INFORMATION

Introduction
The danger of injury or death from electrical shock, fire, or explosion is present while conducting experiments in this laboratory. To work safely, it is important that you understand the prudent practices necessary to minimize the risks and what to do if there is an accident.

Electrical Shock
Avoid contact with conductors in energized electrical circuits. The typical cannot let-go (the current in which a person cannot let go) current is about 30mA (OSHA). Muscle contractions can prevent the person from moving away from the energized circuit. Possible death can occur as low as 50mA. For a person that is wet the body resistance can be as low as 1000 ohms. A voltage of 50V can result in death.

Do not touch someone who is being shocked while still in contact with the electrical conductor or you may also be electrocuted. Instead, shut down the power source or switch off the electric circuit breaker if the power source is inaccessible.

Make sure your hands are dry. The resistance of dry, unbroken skin is relatively high and thus reduces the risk of shock. Skin that is broken, wet, or damp with sweat has a low resistance.

When working with an energized circuit, work with only your right hand, keeping your left hand away from all conductive material. This reduces the likelihood of an accident that results in current passing through your heart.

Be cautious of rings, watches, and necklaces. Skin beneath a ring or watch is damp, lowering the skin resistance. Shoes covering the feet are much safer than sandals.

Fire
Some electronic components can become extremely hot and cause severe burns if touched. In case of fire outbreak, turn off the power supply and notify the laboratory instructor. Avoid using fire extinguishers on electronic equipment.

Explosions
When using electrolytic/tantalum capacitors, be careful to observe proper polarity and do not exceed the voltage rating. Electrolytic and tantalum capacitors can explode and cause injury.

First Aid
A first aid kit is always available in the laboratory. Consult the laboratory instructor if you need to make use of the first aid kit and proceed to University Clinic if needed.

TROUBLESHOOTING HINTS

1. Be sure that the power is turned on.

2. Be sure that the supply voltages are correct and the ground connections are common. For some power supplies, the 'ON' button needs to be clicked in order to turn on the output channel.

3. Be sure the circuit you built is identical to that in the diagram. Do a node-by-node check if needed.

4. Be sure the resistors, capacitors and inductors used in your circuit all have the correct values. You can get that information using a digital multimeter, color code or the nominal value printed on the component.

5. Be sure that the experiment is set up correctly and you are measuring the correct parameter

GUIDELINES FOR LABORATORY REPORT NOTEBOOK

The laboratory workbook contains the record of all works pertaining to the experiment. This record should be sufficiently complete so that anyone of similar technical background can perform the experiment by simply following your experimental procedures. Record everything directly into the workbook during the experiment.

Organization in your workbook is important. Descriptive headings should be used to separate and identify the various parts of the experiment. Record the data in chronological order. A neat, organized and complete record of an experiment is just as important as the experimental work.

1. **Heading:** The experiment number should be at the top of each page. Your name and date should be at the top of the first page of each day's experimental work.
2. **Title**: The experiment title should be clearly stated.
3. **Objective(s):** A brief but complete statement of what you intend to find out or verify in the experiment should be at the beginning of each experiment
4. **Diagram(s):** A circuit diagram should be drawn and labeled so that the actual experiment circuitry could be easily duplicated at any time in the future. Be especially careful to record all circuit changes made during the experiment.
5. **Equipment List:** List those items of equipment which have a direct effect on the accuracy of the data. It may be necessary later to locate specific items of equipment for rechecks if discrepancies develop in the results.
6. **Procedure(s):** In general, lengthy explanations of procedures are unnecessary. Short commentaries alongside the corresponding data may be used. Keep in mind the fact that the experiment must be reproducible from the information given in your notebook.
7. **Data:** Think carefully about what data is required and prepare suitable data tables. Record instrument readings directly. Do not use calculated results in place of direct data; however, calculated results may be recorded in the same table with the direct data. Data tables should be clearly identified and each data column labeled and headed by the proper units of measure.
8. **Calculations:** Not always necessary but equations and sample calculations are often given to illustrate the treatment of the experimental data in obtaining the results.
9. **Graphs:** Graphs are used to present large amounts of data in a concise visual form. Though not required for every practical session, they are often an efficient analytical tool. Data to be presented in graphical form should be plotted in the laboratory so that any questionable data points can be checked while the experiment is still set up. The grid lines in the notebook can be used for most graphs. If special graph paper is required, affix the graph permanently into the notebook. Give all graphs a short descriptive title. Label and scale the axes. Use units of measure. Label each curve if more than one on a graph.

10. Results: The results should be presented in a form which makes the interpretation easy. Large amounts of numerical results are generally presented in graphical form. Tables are generally used for small amounts of results. Theoretical and experimental results should be on the same graph or arranged in the same table in a way for easy correlation of these results.
11. **Answers to Questions:** Some laboratory sessions will ask questions based on the experiments you just performed. The answer to the questions are usually gotten using the data acquired during the experiment. These questions will be graded so take time to answer them carefully. Be sure that you use only the acquired data to answer the questions
12. **Conclusion:** This is your interpretation of the results of the experiment as an engineer. Be brief and specific. Give reasons for important discrepancies.

LABORATORY GRADING POLICY

1. Students must follow the Guidelines for Laboratory Report Notebook for submission of the laboratory report.
2. Each student must submit one REPORT per laboratory experiment.
3. The EXACT DURATION of a laboratory session will be decided by the laboratory instructor prior to the laboratory experiment.
4. Students should not skip the laboratory periods. SPECIAL PERMISSION must be obtained from the LABORATORY INSTRUCTOR to do so. Otherwise, grade points will be DEDUCTED from the student's total grade earned for the scheduled laboratory experiment.
5. Each laboratory session will be graded out of 100 points. The average of all the scores attained for all sessions in a semester will be the scores for the student in the course for that semester.
6. Students who arrive for a laboratory session more than thirty (30) minutes after commencement of session will not be allowed in and will be given a zero (0) grade for that session
7. All laboratory report MUST be submitted before the deadline as decided by the laboratory instructor.

If required, Lab policy may be modified at the discretion of the lab Instructor

EXPERIMENT #: UILCPE/ECT 1

EXPERIMENT TITLE: Direct Current (DC) Measurement

OBJECTIVES

- To understand the basics of Direct Current (DC) circuits.
- To use a digital multimeter to measure DC voltage, current and resistance.
- To understand the valid measurement condition for a digital multimeter.

EQUIPMENT REQUIRED

1. Breadboard
2. DC power supply
3. Digital multimeter

BACKGROUND

DC Circuit Basics

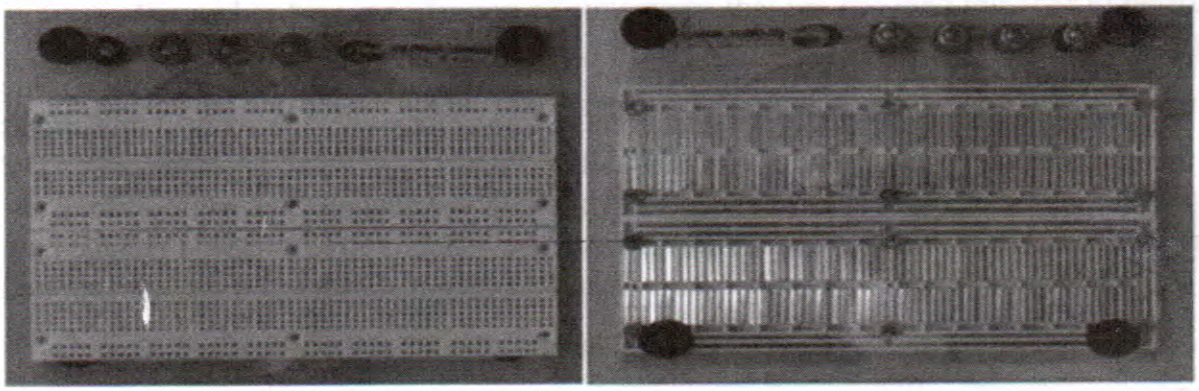
A DC circuit is an electrical circuit that consists of any combination of electronic components powered by DC voltage source(s). The voltages and currents in this circuit are invariant with time. A DC circuit is usually powered by a DC voltage source or a DC current source. There are basic concepts and laws that are fundamental to circuit analysis. These laws are Ohm's law, Kirchhoff's Current Law (KCL) and Kirchhoff's Voltage Law (KVL). In addition, the voltage divider rule and the current divider rule are often applied to simplify the circuit analysis.

Breadboard

A breadboard is also referred to as a solderless breadboard or a plugboard. It is used to build temporary circuits for testing or to experiment new circuit ideas. It has many holes, which can be used to plug in resistors, capacitors, inductors, ICs, and etc. A typical breadboard is shown in Figure 1.1(a). The backside of the bread board, Figure 1.1(b), has strips of metal connecting the holes on the front side. The holes connected by same metal strip form one common node in a circuit. Different components at a given node are connected by pushing in a corresponding end of each component into holes connected to the same node. It is also noticed that some common nodes are longer than most of the five-hole nodes. They are typically used for power supply connections or for those nodes to which many components are connected. A jumper wire can also be used to combine two nodes into one. They are used for DC or AC power supply connections. To connect a binding post onto the breadboard, a wire with long-enough metal exposed is inserted into the hole at the bottom of the post followed by tightening the plastic cap to ensure good connection. The other end of the wire is then plugged into one of the long common nodes on the breadboard.

Here are a few tips for using the breadboard:

1. Never build your circuit without a breadboard, even for the easiest circuit configurations.
2. Always use the binding posts and side-lines (long common nodes) for power supply connections.
3. It is recommended to use black wires for ground and red (or other colors if there are multiple voltages needed) for positive voltage (or other DC/AC voltages).
4. Keep the jumper wires short and flat on the board, so that the circuit doesn't look cluttered.
5. Route jumper wires around the chips, so that it makes it easy to change the chips.
6. You could trim or bend the resistor/capacitor/inductor lead, so that they will fit in snugly and won't get pulled out by accident.
7. A wire should be used to connect the probe of an oscilloscope onto the breadboard, since the probe connection might loosen the existing connection of your components.

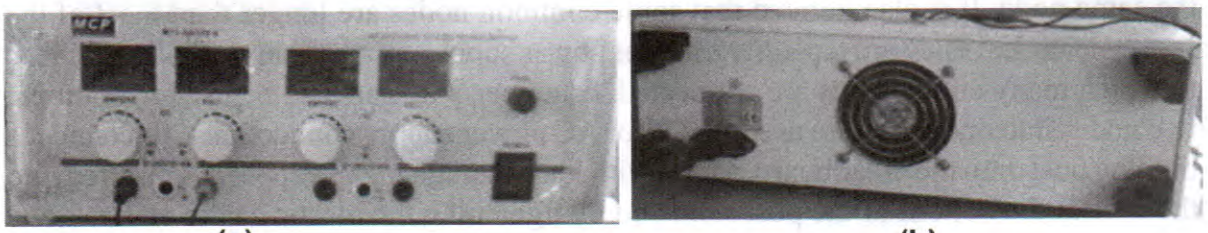


(a) (b)

Figure 1.1: Breadboard (a) Front view (b) Back view

Power Supply

A Power supply is a device that supplies DC or AC voltage and current to a circuit by providing the required voltage levels where needed. A typical example of power supply unit is shown in Figure 1.2.



(a) (b)

Figure 1.2: Power supply. (a) Front; (b) Back.

Digital Multimeter

A digital multimeter is an electronic instrument that is used to measure voltage, current, resistance and other electrical and electronics parameters.

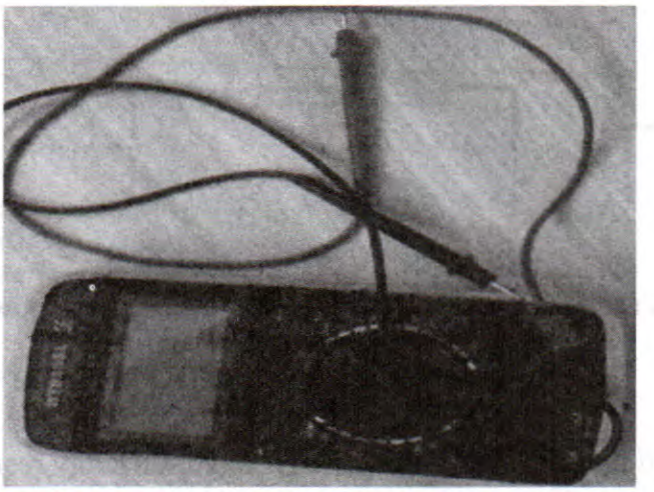


Figure 1.3: Digital Multimeter

Tips for using the Digital Multimeter (DMM):

1. Voltage is measured by placing the DMM in parallel with the Device Under Test (DUT) on which the voltage is to be measured, as shown in Figure 1.4 (a). First connect the DMM input connector (red probe) to 1000V/600V input, and select DCV or ACV. The ground probe (black probe) is connected to the corresponding low input. Then place the probes on two sides of the DUT. Preferably, the red probed should be connected to the side with higher potential. But sometimes it is hard to decide in a circuit with multiple DC voltage sources as for which end of the DUT has higher potential. It is fine to just randomly pick a side. If the reading ends up being negative, that means the red probe is connected to the lower-potential end.
2. Current is measured by inserting the DMM into the circuit and letting the current being measured go through the DMM, as shown in Figure 1.4(b). First connect the DMM input connector (red probe) to 400mA or 10A inputs, and select DCI or ACI. The ground probe (black probe) is connected to the corresponding LOW input. Then insert the probes into the branch being measured. Preferably, current should flow into the DMM from the red probed and flow out from the black probe. But sometimes it is hard to decide the current direction in a circuit with multiple DC voltage sources. It is fine to assume a direction. If the reading ends up being negative, that means the current flows in the opposite direction from your assumption.
3. Resistance measurement is done similar to the voltage measurement by placing the DMM across the resistor to be measured, as shown in Figure 1.4(c) and use the Ω button. The ground probe (black probe) is connected to the corresponding LOW input. Note that the resistor being measured should be disconnected from the rest of the circuit during this measurement.
4. DMM usually has an internal resistance of 10MΩ, as shown in the circuit in Figure 1.6 (e). The measurement result will be inaccurate if the resistor for which the voltage or current being measured is comparable in value to the internal resistance.

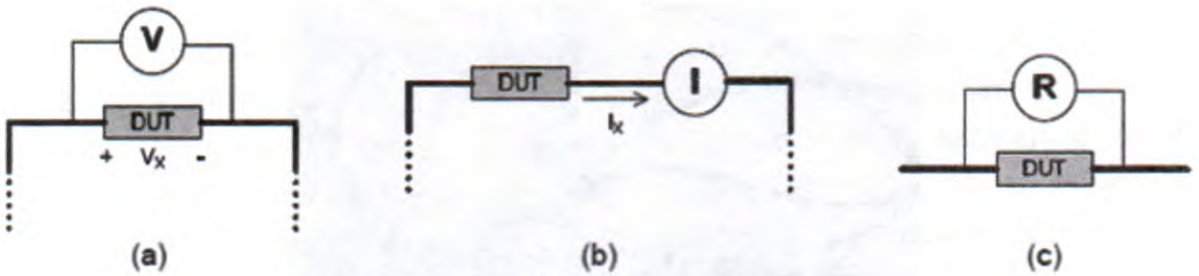


Figure 1.4: Measurement using DMM, (a) Voltage; (b) Current; © Resistance

PREPARATION

For the circuits in Figure 1.6, use 9V as input voltage. Pick any resistor values between 1KΩ and 56KΩ and assign the same values to all resistors with the same name and different values to those with different names. Ask the laboratory instructor for available resistors and capacitors.

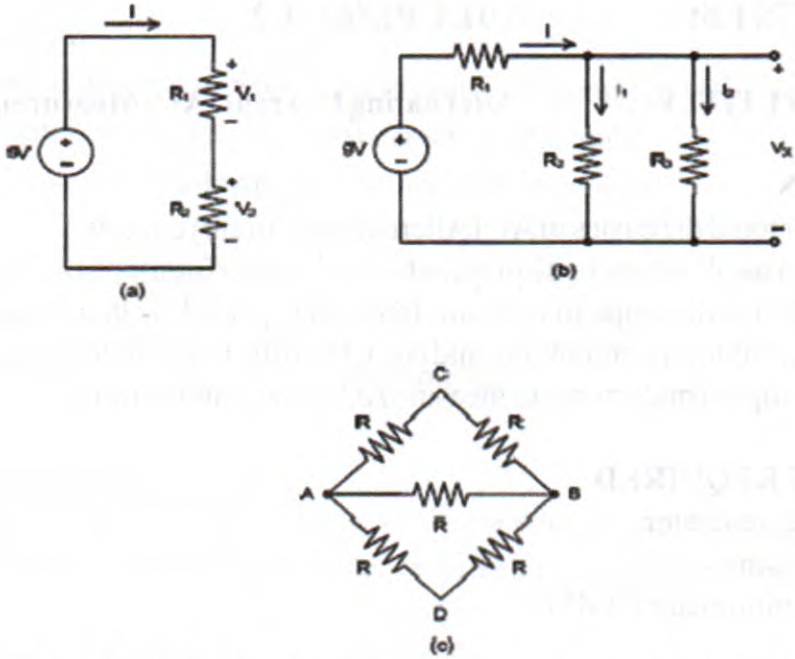
1. For the circuit in Figure 1.6(a), calculate:

a) V1, V2 and I using Ohm's Law.
b) V1, V2 using Voltage Divider Rule.
c) Compare the result in (a) and (b).
d) Add V1 and V2, compare the value obtained with the supplied voltage.

2. For the circuit in Figure 1.6(b), calculate:

a) I1, I2, I and VX using Ohm's Law.
b) I1, I2 using Current Divider Rule.
c) Compare the result in (a) and (b).
d) Add I1 and I2, compare the value obtained with the total current in the circuit.

3. For the circuit in Figure 1.6(c), calculate the equivalent resistance between points AB and points CD. (Hint: Rearrange the resistors in the circuit or use Y-Δ transformation)



Figures 1.6: Circuits to verify (a) Voltage Divider Rule (b) Current Divider Rule (c) Equivalent Resistance

EXPERIMENT

1. Measure the resistance of each resistor used in this experiment using a DMM. Compare the nominal value with the one measured. The nominal value of a resistor can also be obtained from its color bands or color code.
2. Build the circuits in Figure 1.6 on a breadboard.
3. For circuit (a), use a DMM to measure I, V1 and V2.
4. For circuit (b), use a DMM to measure I, I1, I2 and VX.
5. For circuit (c), use a DMM to measure RAB and RCD.

EXPERIMENT #: UILCPE/ECT 2

EXPERIMENT TITLE: Alternating Current (AC) Measurement

OBJECTIVES

- To understand the basics of AC (Alternating Current) circuits
- To use an oscilloscope to display and record a waveform
- To use an oscilloscope to measure frequency, period, voltage (magnitude, peak-to-peak, maximum, minimum, and etc.), DC offset, etc. of the waveform
- To use a digital multimeter to measure AC voltage and current

EQUIPMENT REQUIRED

1. Function generator
2. Oscilloscope
3. Digital multimeter (DMM)

BACKGROUND

AC Circuit Basics

An Alternating Current is a signal that is a function of time. In AC circuits, the voltage and current sources are time-varying, with the most common being a sinusoidal variation. Other time-varying applications, depending on the application, include the square wave and the triangular wave.

A sinusoidal AC voltage or current is described by its amplitude, frequency and phase, for example,

$$V_{magnitude} = V_{peak-to-peak} \times \frac{1}{2} = V_{rms} \quad (2.1)$$

$$I_{magnitude} = I_{peak-to-peak} \times \frac{1}{2} = I_{rms} \quad (2.2)$$

For sine waveforms,

$$V_{magnitude} = V_{peak-to-peak} \times \frac{1}{2} = V_{rms} \times \sqrt{2} \quad (2.3)$$

$$I_{magnitude} = I_{peak-to-peak} \times \frac{1}{2} = I_{rms} \times \sqrt{2} \quad (2.4)$$

For triangular waveforms,

$$V_{magnitude} = V_{peak-to-peak} \times \frac{1}{2} = V_{rms} \times \sqrt{3} \quad (2.5)$$

$$I_{magnitude} = I_{peak-to-peak} \times \frac{1}{2} = I_{rms} \times \sqrt{3} \quad (2.6)$$

Function Generator

A function generator is electronic equipment used to generate electrical waveforms. It is widely used in development and testing of electronic system, for example, as a signal source to test amplifiers, or to introduce an error signal into a control loop.

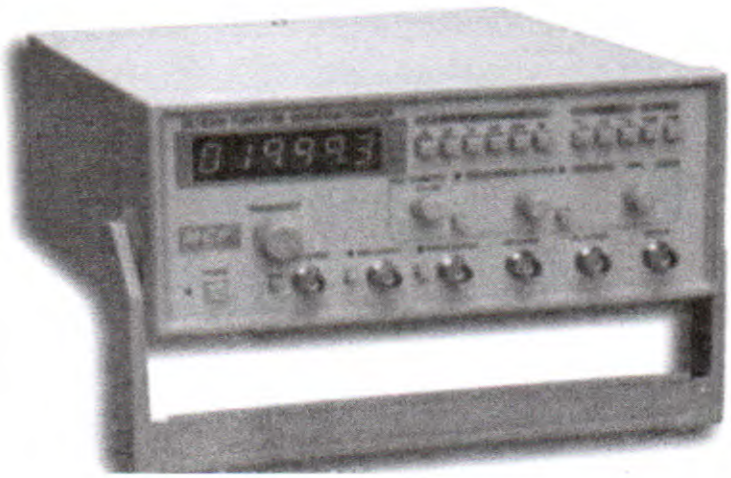


Figure 2.1: Function Generator

Oscilloscope

An oscilloscope is an electronic testing instrument that allows observation of signal voltages. Usually a two-dimensional graph of one or more electrical potentials is displayed using the vertical or 'Y' axis with time along the horizontal or 'X' axis. In most instances, an oscilloscope shows events that repeat with either no change or very slow changes. Thus, the exact wave shape of an electrical signal can be observed. In addition to the amplitude of the signal, an oscilloscope can show distortion, the time between two events (such as pulse width, period, rise time, and etc), and relative timing between two related signals.

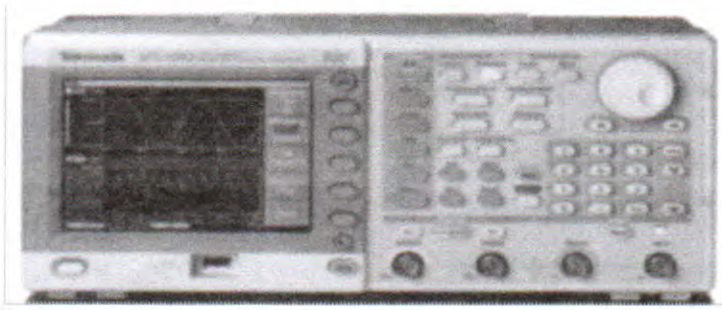


Figure 2.2: Oscilloscope

PREPARATION

1. Calculate the RMS voltage of the following waveforms with 10 Vpeak-to-peak:

a. Sine wave

b. Square wave

c. Triangle wave

2. Calculate the period of a waveform with the frequency of:

a. 100 Hz

b. 1KHz

c. 100KHz

EXPERIMENT

Waveform Display and Measurement

1. Click 'Default Settings' button located beside the LCD screen on oscilloscope to clear all the settings from the previous user. For the same purpose, click the 'Default' button on the function generator, which is located on the left of the numeric keypad, and then click Enter on the right of the keypad.
2. Set the function generator output to sine wave.
3. Connect Ch.1 of the oscilloscope to the function generator output. Polarities (positive and negative) of both probes need to be matched, i.e. positive to positive, negative to negative. Turn on Channel 1 output on function generator by clicking the 'On' button of channel 1. Click 'Autoset' button on the oscilloscope to automatically find the waveform on the LCD screen.
4. Click on the different 'Function' selection buttons on the function generator. Set the waveform back to sinusoidal.
5. Various measurement options for all four channels can be set on the oscilloscope. To do this, click 'Measure' button in 'Wave Inspector' section and press 'Add Measurement' button to bring up the menu. Use the 'Multipurpose a' knob to select the source (Ch1, Ch2, Ch3 or Ch4, whichever is available) and 'Multipurpose b' knob to select the measurement type. Using the 'Multipurpose b' knob highlight the frequency, period, amplitude, mean and RMS, and click on 'OK Add Measurement' each time to bring up the readings on the screen. When finished, click on the 'Menu Off' button to turn off the menu. (Note: the amplitude definition might be different from that in your text book. Make sure to check the definition on the oscilloscope when selecting this measurement.)
6. Vary the frequency and then the amplitude on the function generator. It can be done by clicking on the 'Frequency/Period' button or the 'Amplitude/High' button, then either type in the number using the numeric keypad or adjust the knob above that. Use 'Autoset' button or manually adjust the scale vertically or horizontally when needed. (Note: the readings on the oscilloscope will not be accurate if the waveform is not properly displayed on the LCD screen, e.g. too few cycles or the waveform exceeds the screen in the vertical direction.)

7. Vary the channel scale knob in the 'Vertical' section and the time scale knob in the 'Horizontal' section on the oscilloscope.
8. Now, with all the measurement displayed on the screen, set the frequency of the input signal to 1KHz and magnitude to 5V. (Try to keep about 4 - 5 cycles displayed on the oscilloscope screen by adjusting both the vertical and horizontal scale knobs in order to obtain an accurate reading.)
9. Adjust the 'level' knob on oscilloscope, move it back in, so that it is slightly below the peak and the waveform till it becomes stable.
10. Keeping the level the same, slowly decrease the amplitude of the input signal on the function generator by adjusting the knob above the numeric keypad when 'Amplitude' is high-lighted on the screen (if not, click on the 'Amplitude/High' button).
11. Adjust the level arrow back to the zero and the signal amplitude back to 5 V.

Voltage Measurement

1. With a sine wave of 5V magnitude from function generator, set DMM to measure AC voltage and place it in parallel with the probe of the oscilloscope. Compare the readings on the oscilloscope with the readings on the DMM.
2. Switch the waveform to a square waveform first, and then to a triangle (ramp) waveform. In each case what is the AC voltage on DMM? Describe and compare the measured voltage on the DMM and the calculated values in PREPARATION.

Frequency Measurement

1. Set the waveform back to sine wave on the function generator, record the frequency and period readings from the oscilloscope.
2. Change the frequency on the function generator to 100Hz and 100KHz. Record the frequency and period readings. Compare the measured values with those calculated in PREPARATION.

EXPERIMENT #: UILCPE/ECT 3

EXPERIMENT TITLE: Network Analysis Methods

OBJECTIVES

- To analyze a resistive circuit using node or mesh analysis.
- To understand Thevenin's and Norton's theorems.
- To verify the superposition principle.

EQUIPMENT REQUIRED

1. Breadboard
2. DC power supply
3. Digital multimeter (DMM)

BACKGROUND

Electrical circuit analysis is the process of finding the voltages across and the currents through every component in the network. The following techniques are normally used for resistive circuit analysis.

Nodal analysis is a method of determining the voltage at the nodes in an electrical circuit with respect to a reference node, using Kirchhoff's Current Law (KCL).

Mesh analysis is a method that is used to solve for current through any component in a planar circuit using Kirchhoff's Voltage Law (KVL). In some cases, one method is clearly preferred over another. For example, when the circuit contains only voltage sources (or current sources), it is probably easier to use mesh analysis (or node analysis). It is often helpful to consider which method is more appropriate for the problem solution and make the selection.

Thevenin's theorem, also called Thevenin's equivalent, states that if we identify a pair of terminals in any circuit made up of both independent and dependent sources and resistors, the circuit can be replaced by an independent voltage source Voc in series with a resistor Rth. This series combination of Voc, the Thevenin's voltage, and Rt, the Thevenin's resistance, is equivalent to the original circuit in the sense that when the same load is connected across the terminals, the same voltage and current will be obtained from the terminals of the load. This equivalence holds for all possible values of load resistances. Voc is the open-circuit voltage of the original circuit across the terminals. Rth can be found by either of the two methods listed below. One is to find the short-circuit current isc, then find Rth using

$$R_{th} = \frac{V_{ac}}{i_{sc}}$$

The other method usually used in less complicated circuits involves deactivating the sources in the circuit, i.e. replacing all independent voltage sources with short circuits and all independent current sources with open circuits, and finding the equivalent resistance, which is Rth. Dependent current and voltage source(s) are not replaced with open circuits or short circuits.

Norton's theorem, also referred as Norton equivalent, is a dual of Thevenin's theorem. If we identify a pair of terminals in any circuit made up of both independent and dependent sources and resistors, the circuit can be replaced by a parallel combination of an ideal current source isc and a conductance Gn, where isc is the short-circuit current at the terminals in the original circuit and Gn is the ratio of the short-circuit current to the open-circuit voltage at the terminals in the original circuit.
The four parameters Voc, Rth, isc and Gn are related by

$$V_{oc} = i_{sc}R_{th}$$

$$G_n = \frac{1}{R_{th}}$$

For linear circuits containing two or more independent sources, the superposition principle can also be used for circuit analysis. The voltage across (or the current through) any element can be obtained by adding algebraically all the individual voltages (or currents) caused by each independent source acting alone, with all the other independent voltage sources replaced by short circuits and all the other independent current sources replaced by open circuits.

PREPARATION

For the circuit in Figure 3.1, pick any resistor values between 1KΩ and 50KΩ, such that not all resistors are the same

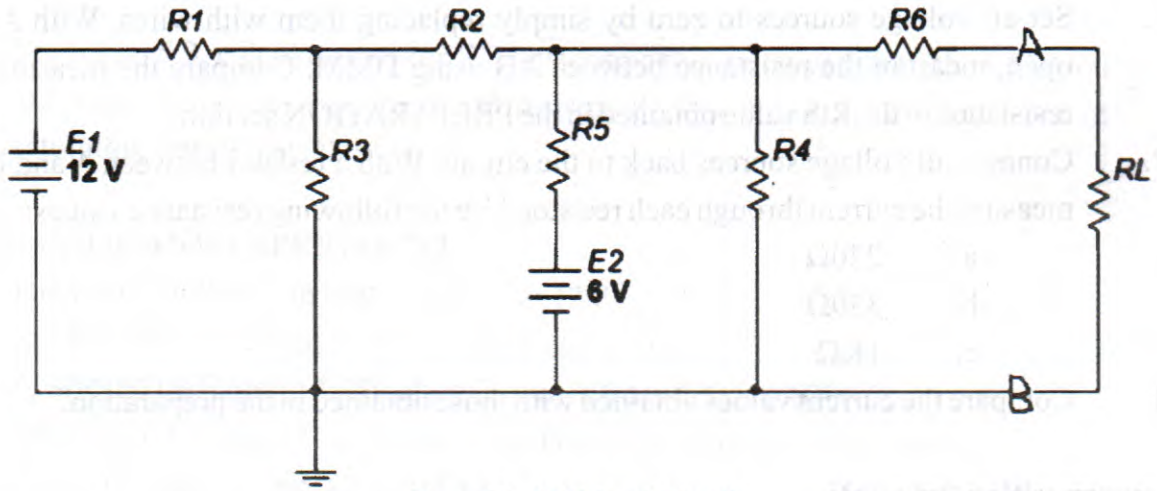


Figure 3.1: DC Network Circuit

1. Short AB, as shown in Figure 3.1. Use mesh analysis to calculate the voltage across each resistor and the current through AB.
2. Leave AB open, as shown in Figure 3.1 Use nodal analysis to calculate the voltage across each resistor and that across AB.
3. Find Thevenin's and Norton's Equivalent (Voc, Isc and Rth) using the results from step 1 and 2.
4. If a resistor is connected between nodes A and B, use Thevenin's theorem to calculate the current through this resistor using the following values: 1KΩ, 330Ω and 220Ω.
5. Leave AB open. Find the voltage across AB caused by E1 only. In order to do so, E2 need to be set to zero by simply replacing it with a wire (short). Repeat this step and find the VAB caused by E2 only. Verify that the sum of these two voltages found using each source individually equals to the voltage found in step 2.

EXPERIMENT

Build the circuit in Figure 3.1 on the breadboard.

Mesh Analysis and Nodal Analysis

1. Short AB by connecting a wire across nodes A and B. Measure the voltage across each resistor and the current through AB.
2. Leave AB open. Measure the voltage across each resistor and the voltage between AB.
3. Compare the results obtained from step 1 and 2 with those from PREPARATION section.

Thevenin's and Norton's Theorems

1. Set all voltage sources to zero by simply replacing them with wires. With AB open, measure the resistance between AB using DMM. Compare the measured resistance to the Rth value obtained in the PREPARATION section.
2. Connect all voltage sources back to the circuit. With a resistor between A and B, measure the current through each resistor. Use the following resistance values:
    a. 220Ω
    b. 330Ω
    c. 1KΩ
3. Compare the current values obtained with those obtained in the preparation.

Superposition Principle

With AB open, measure the voltage across AB caused by each voltage source individually. Compare the results with those obtained from PREPARATION section.

EXPERIMENT #: UILCPE/ECT 4

EXPERIMENT TITLE: First Order Circuits

OBJECTIVES

- To study the step response of first order circuits.
- To understand the concept of time constant.

EQUIPMENT REQUIRED

1. Breadboard
2. Function generator
3. Oscilloscope
4. Digital multimeter (DMM)

BACKGROUND

First-order transient circuits are described by a first order differential equation. Firan inductor or a capacitor, i.e. RL or RC circuits.

For a step voltage/current source input, the output can be expressed as

$$X(t) = X(\infty)[X(0) - X(\infty)] \times e^{\frac{t}{\tau}} \quad (4.1)$$

Where, $X(0)$ is the circuit response at t = 0, and $X(\infty)$ is the response at t = ∞. The parameterτ is called time constant of the circuit and gives the time required for the response (i) to rise from zero to 63% (or $1 - \frac{1}{e}$) of its final steady value as shown in Figure 4.1 (a), or (ii) to fall to37% (or $\frac{1}{e}$) of its initial value as shown in Figure 4.1 (b). Therefore, the smaller the value of τ, the faster the circuit response is.

For RC circuit

$$\tau = RC \quad (4.2)$$

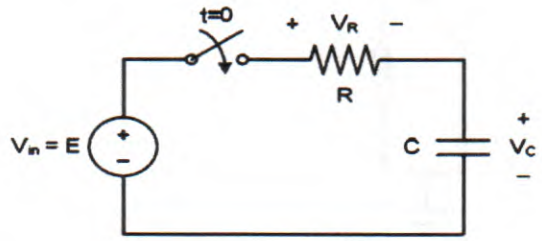
For RL circuit

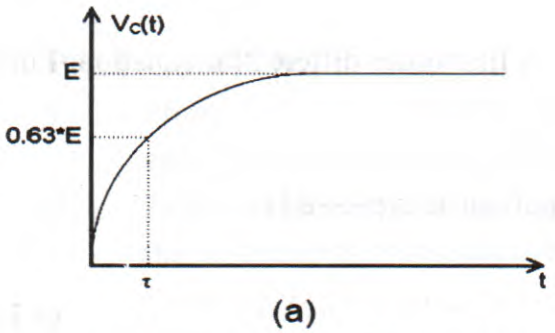
$$\tau = \frac{L}{R} \quad (4.3)$$

Applying the equations above, the voltage responses across the capacitor and the resistor in Figure 4.1 can be written as:

$$V_c(t) = E\left(1 - e^{\frac{t}{RC}}\right), \text{for } t \geq 0 \qquad (4.4)$$

$$V_R(t) = Ee^{\frac{t}{RC}}, \quad \text{for } t \geq 0 \qquad (4.5)$$





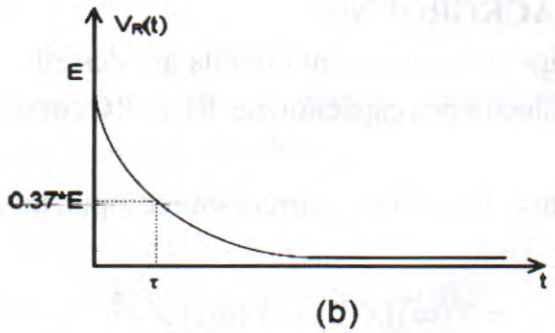


Figure 4.1: A first order circuit and its responses. (a) voltage over the capacitor; (b) voltage over the resistor.

PREPARATION

Figure 4.2 and Figure 4.3 show an RC and an RL circuit.

For all circuits, R = 1KΩ, C = 0.1uF, L = 100mH.

Step Voltage Input

1. For the circuits in Figure 4.2 using step voltage sources, derive the analytical expressior $V_{out}(t)$ for $t \geq 0$, when $V_{in}(t) = Eu(t)$.
2. Sketch or plot $V_{out}(t)$ for each circuit

Square Wave Input

1. For the circuits in Figure 4.3 use a square wave input. Assume that $V_{in}(t)$ square wave with amplitude of $\pm E$ and period of $T_s = 10 \times \tau$.
2. Sketch or plot $V_{out}(t)$ for each circuit using superposition. (Show at least five cycles.)

Hint: the square wave can be broken up into a series of step functions with displacement of $T_s/_2$ and alternate polarities. Each of these step function inputs generates an output. Thus the total output response is the summation of all the individual output.

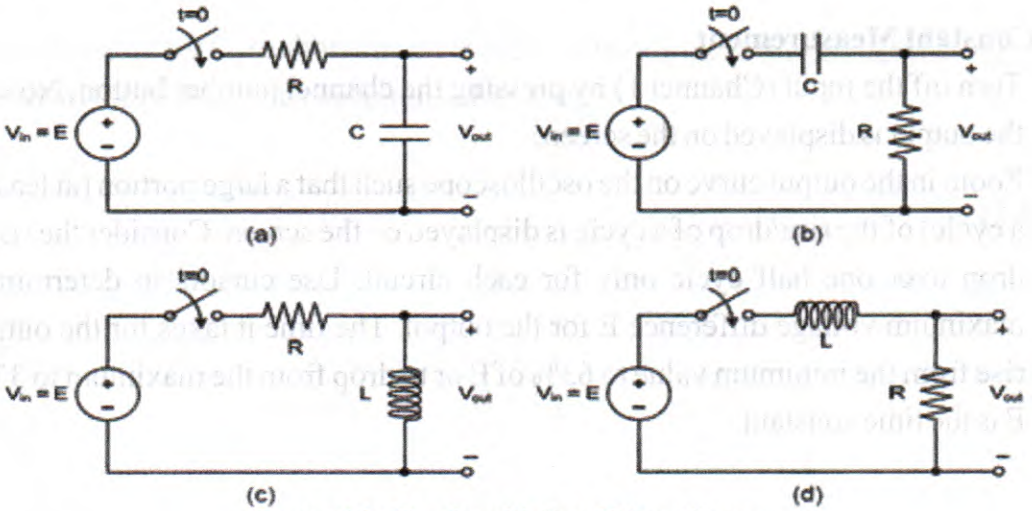


Figure 4.2: Circuits with Step Voltage Sources

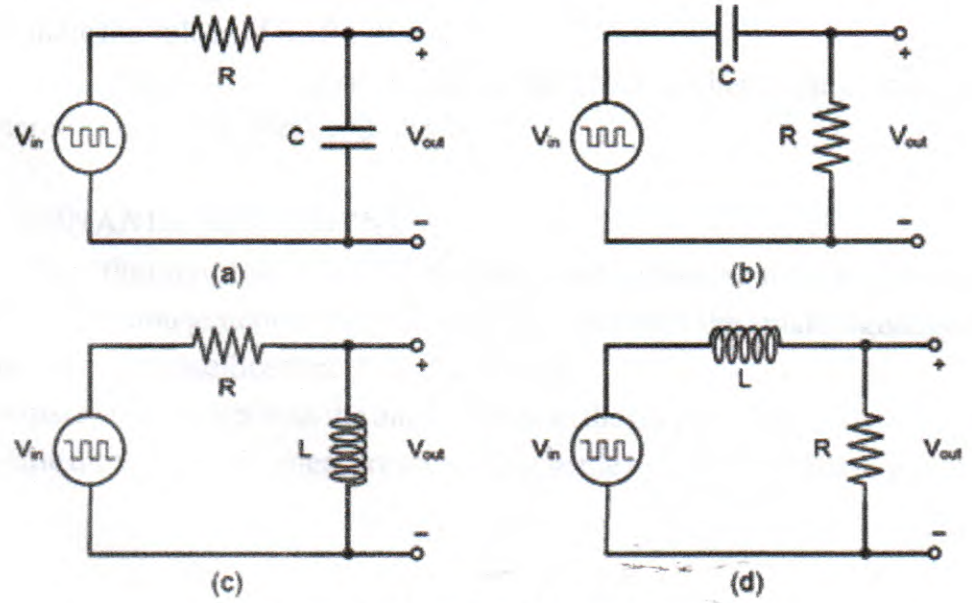


Figure 4.3: Circuits with Square Wave Input

Use the same component values as in the PREPARATION to build the circuits shown in Figure 4.3 (a) – (d).

Square Wave Output

On the oscilloscope, connect Ch1 to the input and Ch2 to the output so that both the input and the output are displayed on the screen. Save the screen image for both the input and the output, preferably to a USB drive. Use the 'Menu' button on the 'Save/Recall' section on the bottom of the oscilloscope screen, then use 'File Utilities' to select or create a folder

to save the image. Press the 'Save Screen Image' button and use the associated buttons next to the screen to select the format or edit file name etc.
Compare these waveforms with the results from PREPARATION.

Time Constant Measurement

1. Turn off the input (Channel 1) by pressing the channel number button. Now only the output is displayed on the screen.
2. Zoom in the output curve on the oscilloscope such that a large portion (at least half a cycle) of the rise/drop of a cycle is displayed on the screen. Consider the rise and drop over one half cycle only for each circuit. Use cursors to determine the maximum voltage difference E for the output. The time it takes for the output to rise from the minimum value to 63% of E or to drop from the maximum to 37% of E is the time constant

EXPERIMENT #: UILCPE/ECT 5

EXPERIMENT TITLE: Sinusoidal Steady State

OBJECTIVES

- To understand and calculate the power factor of a passive circuit
- To verify that resistive components dissipate power
- To verify that reactive components do not dissipate power

EQUIPMENT REQUIRED

1. Breadboard
2. Function generator
3. Oscilloscope
4. Digital multimeter (DMM)

BACKGROUND

The steady-state response is the response that exists after the initial conditions and transient for natural response cease to exist. AC steady state analysis determines the steady state response of a circuit when the inputs are sinusoidal functions. The steady state voltages and currents in the circuit will also be sinusoidal, with the same frequency as the input signal.

The angle $\theta = \theta_V - \theta_I$ referred to as the power factor angle, is involved in the calculation of the average and reactive power. The power factor is the cosine of this angle

$$PF = \cos(\theta_V - \theta_I) = \cos\theta \quad (5.1)$$

A lagging power factor implies that the current lags the voltage, i.e. an inductive load. A leading power factor implies current leads the voltage, i.e. a capacitive load. The power factor (PF) is the ratio of the average power to the apparent power. The average power absorbed by the element is calculated by,

$$P_{AVG} = \frac{V_m \times I_m}{2} \cos(\theta_V - \theta_I) = V_{rms} \times I_{rms} \times \cos(\theta_V - \theta_I) \quad (5.2)$$

and the apparent power is given by,

$$|S| \frac{V_m \times I_m}{2} = V_{rms} \times I_{rms} \quad (5.3)$$

where V_m and I_m are the magnitudes of the voltage and current respectively, Vrms and Irms are the RMS values voltage and current respectively, θV and θI are the phase angles of the voltage and current respectively.

The average power (often simply called 'power') dissipated in a circuit with a periodic input signal of period T is defined as

$$P_{avg} = \frac{1}{T}\int_0^T p(t)dt \tag{5.4}$$

The average power dissipated by a resistor is given by

$$P = I_{rms}{}^2 \times R = V_{rms} \times I_{rms} = \frac{V_{rms}{}^2}{R} \tag{5.5}$$

$$\theta = \left|tan^{-1}\left\{\frac{sin\emptyset}{\frac{V_m}{V_{xm}} - cos\emptyset}\right\}\right| \tag{5.6}$$

and the average power dissipated by reactive components, such as inductors and capacitors, is equal to zero. In other words, reactive components are storage elements and do not dissipate power.

PREPARATION

Consider the circuit in Figure 5.1 with Vin magnitude of 5V and frequency of 3KHz.

1. Calculate the total impedance of the circuit.
2. Calculate the power factor of the circuit.
3. Calculate the voltage and current (they are both complex numbers) in each element in the circuit.
4. Calculate the average power dissipated in each element in the circuit.
5. Calculate the total average power provided by the source.
6. Verify that the power generated by the source equals the total power dissipated in all the components in the circuit.

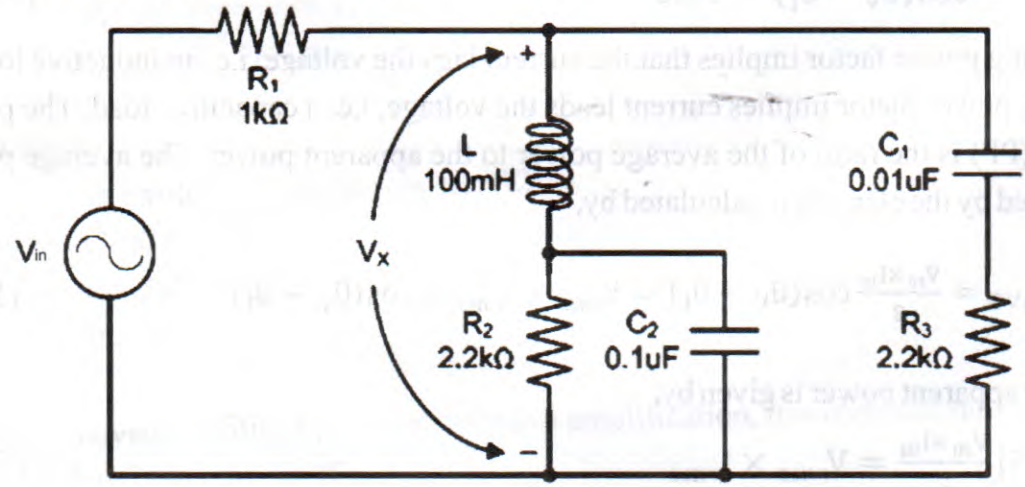


Figure 5.1: AC Network Circuit

EXPERIMENT

1. Build the circuit of Figure 5.1 on breadboard. Set the input voltage to $\pm 5V$ and frequency to 3KHz. Place the Ch1 probe at V_{in} and the Ch2 probe at V_x. Display both V_{in} and V_x on the screen. *(Hint: These two curves should have the same frequency but with a phase shift between each other.)*
2. Measure V_m and V_{xm} using the oscilloscope. Note that $\emptyset$ is measured indirectly. First measure the time difference Δt between the two peaks of V_{in} and V_x Expand the time scale in order to get a better reading with the cursors. Then use the equation $\emptyset = 2\pi f \times \Delta t$ (rad) to calculate $\emptyset$. *(Hint: Convert $\emptyset$ to degrees if needed)*
3. Calculate the power factor using equation 5.6.
4. Measure the voltage and current in each element using a DMM. Compare the result with that from the PREPARATION section.
5. Calculate the average power dissipated in each element using the results (voltage and/or current) from the last step. Compare the result with that from PREPARATION.
6. Measure the total voltage and current provided by the voltage source.
7. Calculate the total power provided by the voltage source. Compare this result with that from PREPARATION.
8. Compare the total power delivered by the voltage source with the total power dissipated by all the elements in the circuit. Explain your result.

EXPERIMENT #: UILCPE/ECT 6

EXPERIMENT TITLE: Series and Resonance

OBJECTIVES

- To study the behavior of series and parallel LC circuits at resonance.
- To understand the resonance frequency, cut-off frequency, bandwidth and quality
- factor of a resonance circuit.
- To determine if a circuit is inductive or capacitive.
- To understand the circuit behavior at resonance.

EQUIPMENT REQUIRED

1. Breadboard
2. Function generator
3. Oscilloscope
4. Digital multimeter (DMM)

BACKGROUND

Resonant circuits form the basis for filters that have better performance than first order (RL, RC) filters in passing desired signal or rejecting undesired signals that are relatively close in frequency. The resonance frequency is defined as the frequency at which the impedance of the circuit is purely real, that is, with zero reactance. For the reactance to be zero, impedance of the inductor must equal that of the capacitor. At resonance, the impedance of a branch with LC in series is equal to zero, which is equivalent to a short circuit, and the admittance of a branch with LC in parallel is equal to zero, which is equivalent to an open circuit. As the frequency increases, the magnitude of an inductive reactance increases, while the magnitude of a capacitive reactance decreases. A circuit is said to be inductive if the total reactance is positive while a circuit is said to be capacitive if the total reactance is negative.

A bandpass RLC circuit will have two cut-off frequencies ω_{c1} and ω_{c2} where the amplitude is $1/\sqrt{2}$ of the maximum value. The cut-off frequency is also called the half-power frequency or 3dB frequency in some cases.

The bandwidth BW (or passband bandwidth) is defined as the difference between the upper and lower cutoff frequencies. In case of a low-pass filter or baseband signal, the bandwidth is equal to its upper cutoff frequency.

Examples of the resonance frequency, cutoff frequencies and the bandwidth are shown in Figure 6.1 for a bandpass and a bandreject filter

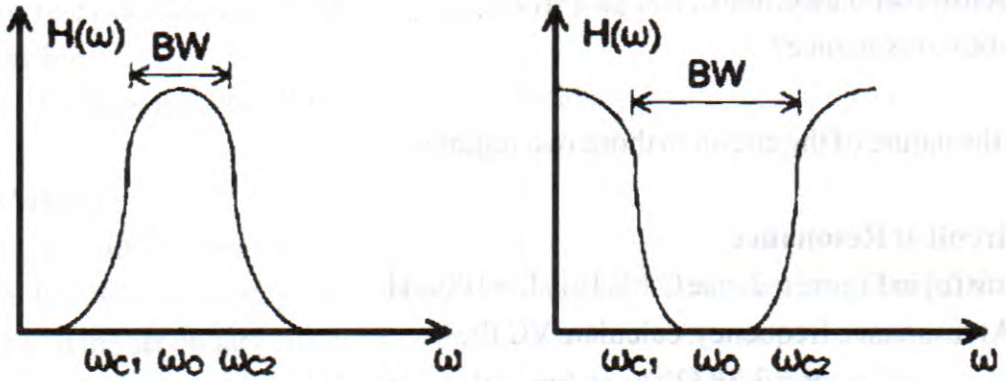


Figure 6.1: Resonance frequency, cutoff frequency and bandwidth

The quality factor or Q factor of the frequency response is described quantitatively in terms of the ratio of the resonance frequency to the bandwidth,

$$Q = \frac{\omega_0}{\beta} \tag{6.1}$$

with both ω_0 and β are in radians. This definition lends itself to laboratory measurement because it is possible to measure both the resonance frequency (ω_0) and the bandwidth (β). It is also defined as an energy ratio,

$$Q = 2\pi \frac{maximum\ energy\ stored}{total\ energy\ lost\ per\ period} \tag{6.2}$$

The steady response of a circuit will in general have a maximum amplitude and phase angle that is different from that of the source. In some cases, the magnitude of the voltage response may exceed that of the voltage source.

PREPARATION

RLC Circuit Basic Measurement

For circuits (a) through (d) in Figure 6.2, use C = 0.1uF, L = 100mH, R = 1KΩ.

1. Derive the transfer function.
2. Find the resonance frequency, cutoff frequencies, bandwidth and Q-factor for each of the circuits.

3. What is the phase relation between the total voltage and current when the frequency is

i. below resonance, and

ii. above resonance?

What is the nature of the circuit in those two regions?

RLC Circuit at Resonance

For circuit (e) in Figure 6.2, use C = 0.1uF, L = 100mH.

1. At resonance frequency, calculate VC if:
   a. R = 3.3KΩ
   b. R = 330Ω
2. Find out whether the magnitude of VC is larger or smaller than Vin. Explain your result.

EXPERIMENT

1. On the breadboard, construct circuit (a) in Figure 6.2. Connect Ch1 to input and Ch2 to output so that both the input and the output are displayed on the oscilloscope.
2. Set the input voltage to $\pm 5V$ and frequency to the value calculated in PREPARATION. For circuits (a) and (c), vary the frequency from the theoretical value to get a maximum output voltage. The frequency at this maximum voltage Vmax is the resonance frequency. For circuits (b) and (d), vary the frequency from the theoretical value to get a minimum output voltage. The frequency at this minimum voltage Vmin is the resonance frequency.
3. If $V_{cutoff} = \frac{V_{max}}{\sqrt{2}}$, vary the frequency until the output voltage equals V_{cutoff} and this is known as the cut-off frequency. (Hint: There should be two (2) cut-off frequencies for each case.)
4. Calculate the bandwidth by subtracting the two (2) cut-off frequencies.
5. Calculate the Q-factor using equation 6.1.
6. Repeat step 1 through 5 for circuits (b) to (d) in Figure 6.2.
7. Compare your result with that from PREPARATION.
8. Build circuit (e) in Figure 6.2. Connect Ch1 to input and Ch2 to output so that both the input and the output are displayed on the oscilloscope. Set the input voltage to $\pm 5V$ and frequency to the resonance frequency found in step 1 for circuit (b) in Figure 6.2.
2. Find the magnitude for VC for the following cases.

a. R = 3.3KΩ

b. R = 330Ω

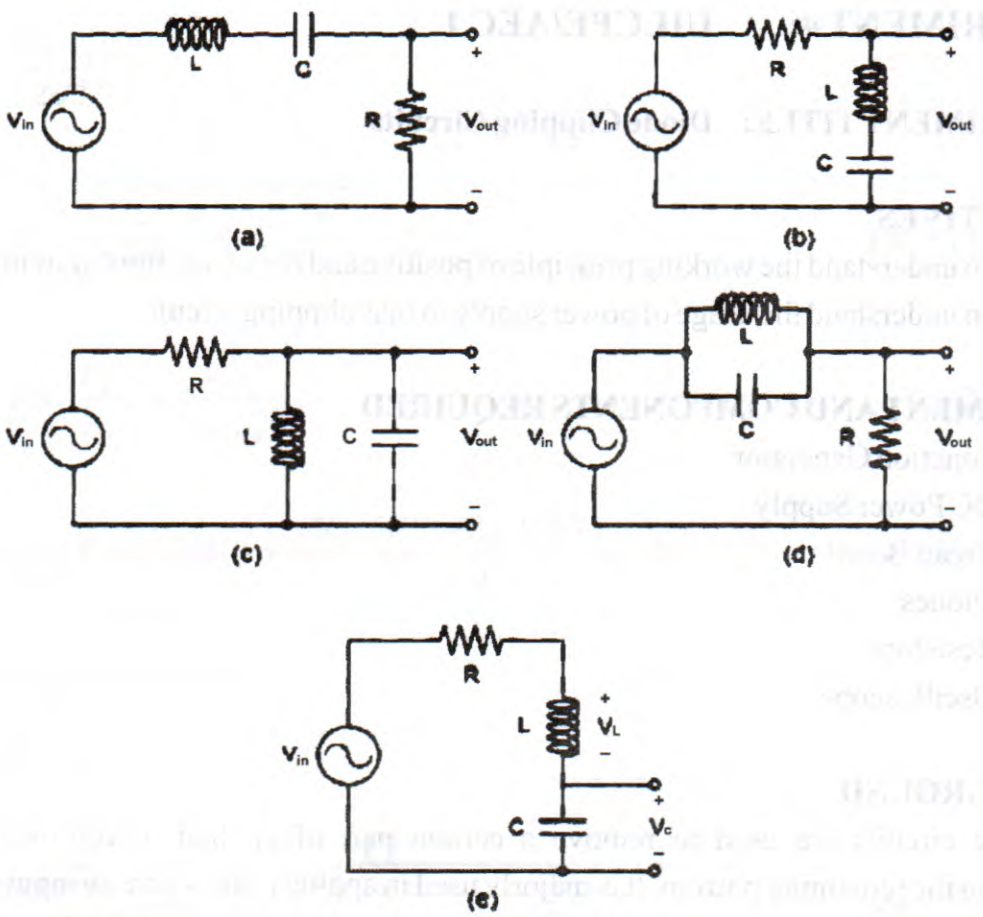


Figure 6.2: RLC Resonance Circuits

EXPERIMENT #: UILCPE/AEC 1

EXPERIMENT TITLE: Diode Clipping Circuits

OBJECTIVES

- To understand the working principle of positive and negative clipper circuits
- To understand the usage of power supply to bias clipping circuits

EQUIPMENT AND COMPONENTS REQUIRED

1. Function Generator
2. DC Power Supply
3. Bread Board
4. Diodes
5. Resistors
6. Oscilloscope

BACKGROUND

Clipping circuits are used to remove a certain part of applied waveform without distorting the remaining portion. It is majorly used in applications where an input voltage should not be allowed to exceed a maximum value and also in wave shaping of signal (function generation). Below are the various circuits and design equations for common clipping circuits using diodes.

Case I: Positive Clipping Circuit:

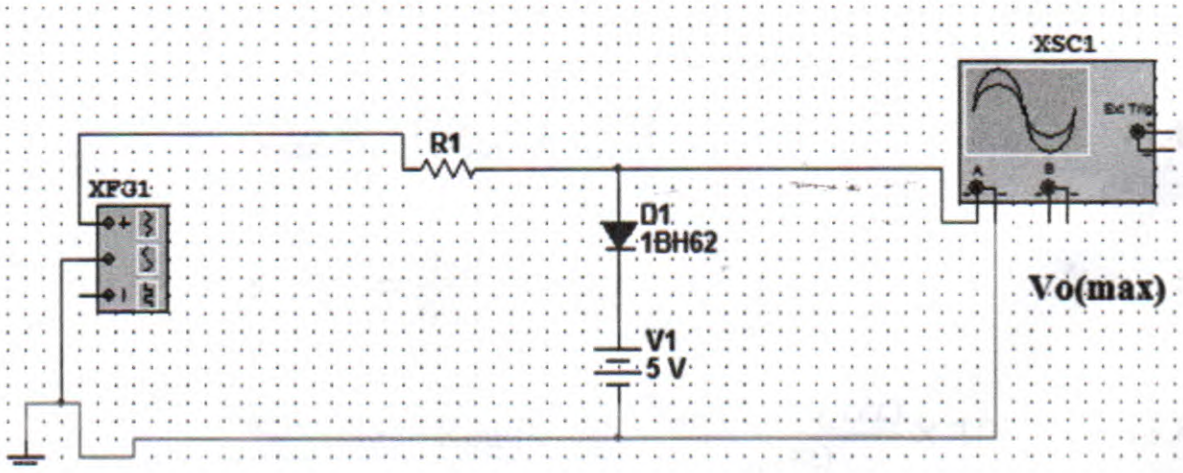


Figure 1.1: Diode Positive Clipping Circuit

Resistance value can be calculated using the relationship below

$$R = \sqrt{R_f * R_r} \quad (1.1)$$

Where R_f - Diode forward resistance = 100Ω

R_r - Diode reverse resistance= 10kΩ

The Reference voltage, V_R is given by

$$V_R = V_0max - V_r \quad (1.2)$$

Where:

V_r (Diode drop) = 0.7V (Silicon Diode)

V_0max is the maximum output voltage:

$$V_R = V_0max - V_r \quad (1.3)$$

In Figure 1.1, V_R= 5V

Case II: Negative Clipping Circuit:

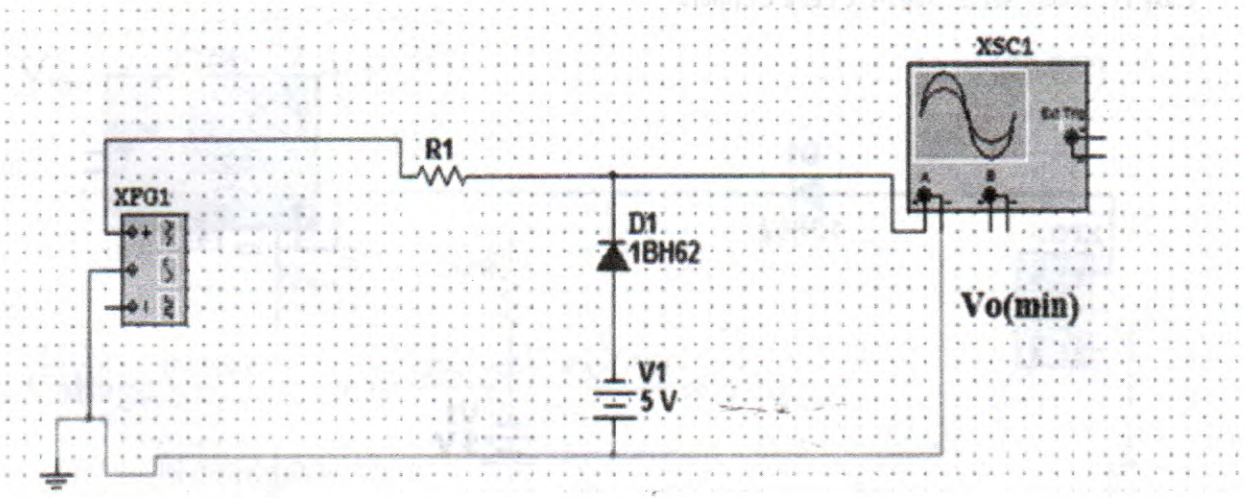


Figure 1.2: Diode Negative Clipping Circuit

The value of R is determined using the formula given in case I. The negative output voltage is calculated using the expression below

$$V_R = V_0min - Vr \quad (1.4)$$

In Figure 1.2, V_R= 5V

Where:

Vr (Diode drop) = 0.7V (Silicon Diode)

A sinusoidal input Vi of 1 KHz and of amplitude 10V P-P is applied at the input

Case III: Diode Series Clipping / Positive Peak Clipper:

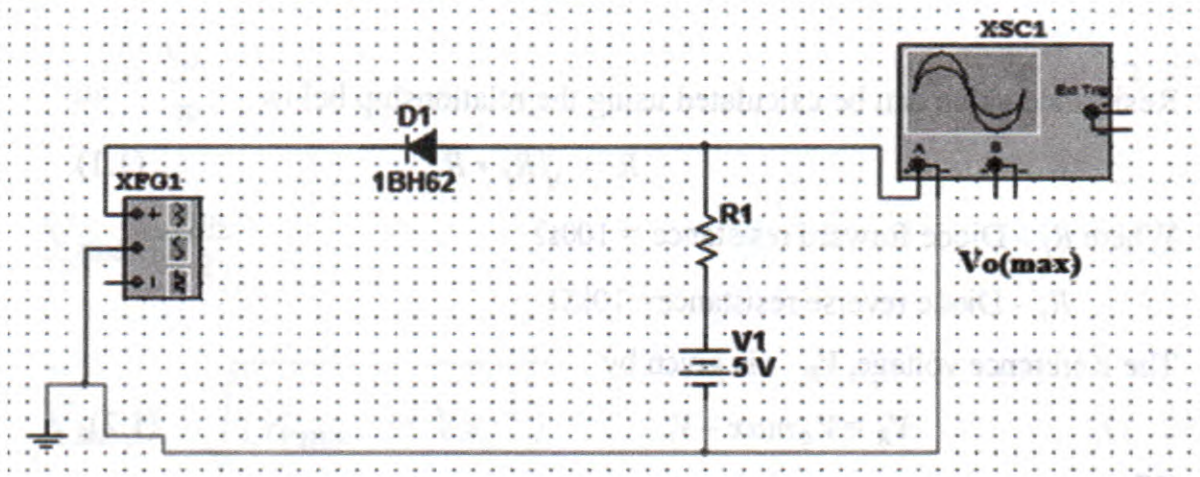


Figure 1.3: Diode Series Clipping/Positive Peak Clipper

The value of R is determine using the formula given in case I and the negative output voltage, The values for maximum output voltage $V_0 max$ is given by

$$V_o max = V_R \tag{1.5}$$

Case IV: Diode Series Negative Peak Clipper:

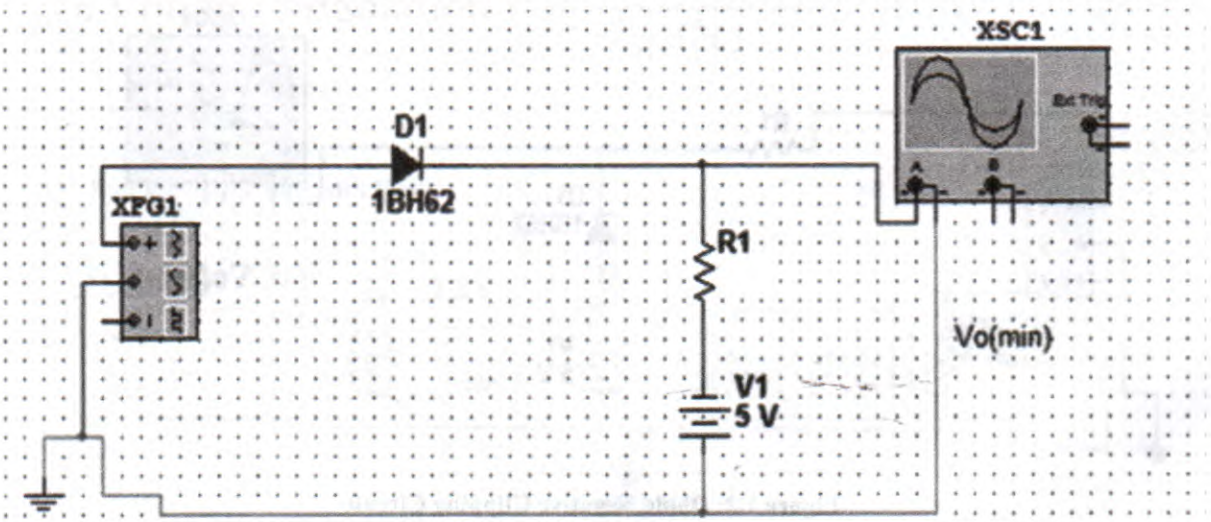


Figure 1.4: Diode Series Negative Peak Clipper Circuit

The value of R is determine using the formula given in case I and the negative output voltage, $V_{o\,min}$ is given by

$$V_{o\,min} = V_R \tag{1.6}$$

Case V: Combination Clipper Circuit

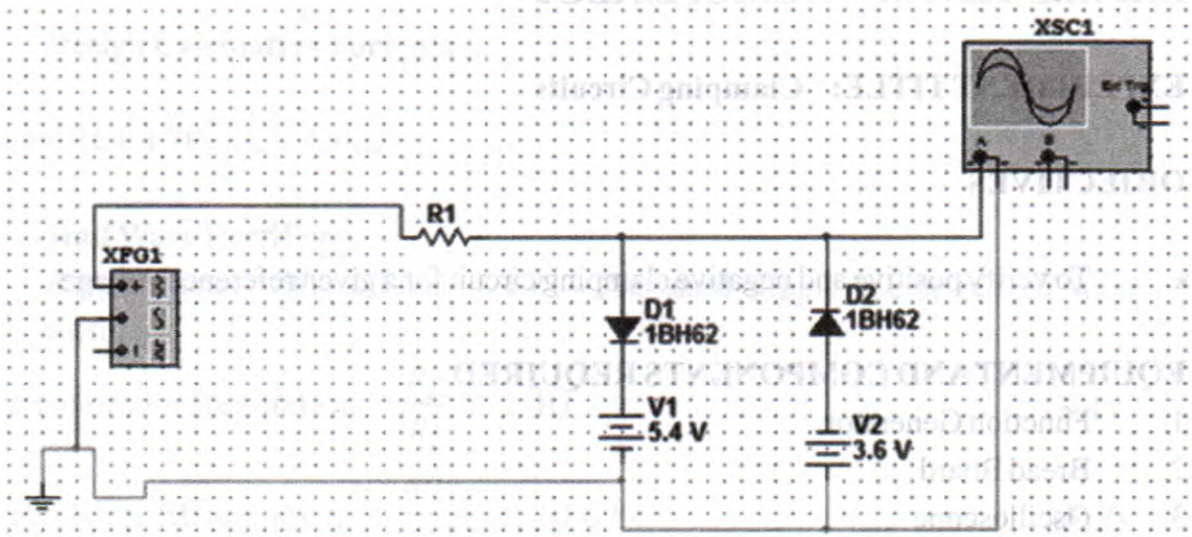


Figure 1.5: Combination Clipper Circuit

The value of R is determine using the formula given in case I and the positive output voltage, $V_o max$ is given by the relationship below

$$V_o max = V_{R1} + V_{r1} \quad (1.7)$$

$$V_o min = V_{R2} + V_{r2} \quad (1.8)$$

PREPARATION

1. From the circuit diagram of Figure 1.1 to Figure 1.5,

R_f = 100Ω and R_r = 100KΩ , calculate R

2. Using the values provided in Figures 1.1 to 1.5 for V_R, calculate the values for $V_o max$ and $V_o min$ where applicable and sketch the output waveform.

EXPERIMENT

A sinusoidal input Vi of 1KHz and of amplitude 10V P-P is applied to the circuit of Figure 1.1 to Figure 1.5 and assume a diode voltage drop of 0.7V. Observe the output signal on the Oscilloscope. (Hint: Connect channel 1 to

EXPERIMENT #: UILCPE/AEC 2

EXPERIMENT TITLE: Clamping Circuits

OBJECTIVES

- To verify positive and negative clamping circuit for a given reference voltage

EQUIPMENT AND COMPONENTS REQUIRED

1. Function Generator
2. Bread Board
3. Oscilloscope
4. Diodes
5. Resistors
6. Capacitors

BACKGROUND

The clamping circuit is made up of resistor, capacitor and a diode. The effect of this circuit is to shift the waveform to a different DC level without distorting the applied waveform. The clamper circuit moves the whole signal up or down to set either the positive peak or negative peak of the signal at the desired level.

The equation for determining the resistance value of a clamping circuit is given in equation 2.1

$$R = \sqrt{R_f * R_r} \tag{2.2}$$

R_f = Diode forward resistance = 100Ω

R_r = Diode reverse resistance = 100kΩ

$$\tau = RC \tag{2.2}$$

Where:

C = Capacitance of the capacitor in Farad

T = period in sec.

The circuits for common clamping circuits using diodes are presented in Figures 2.1 to 2.4.

Case I: Positive Clamping Circuit

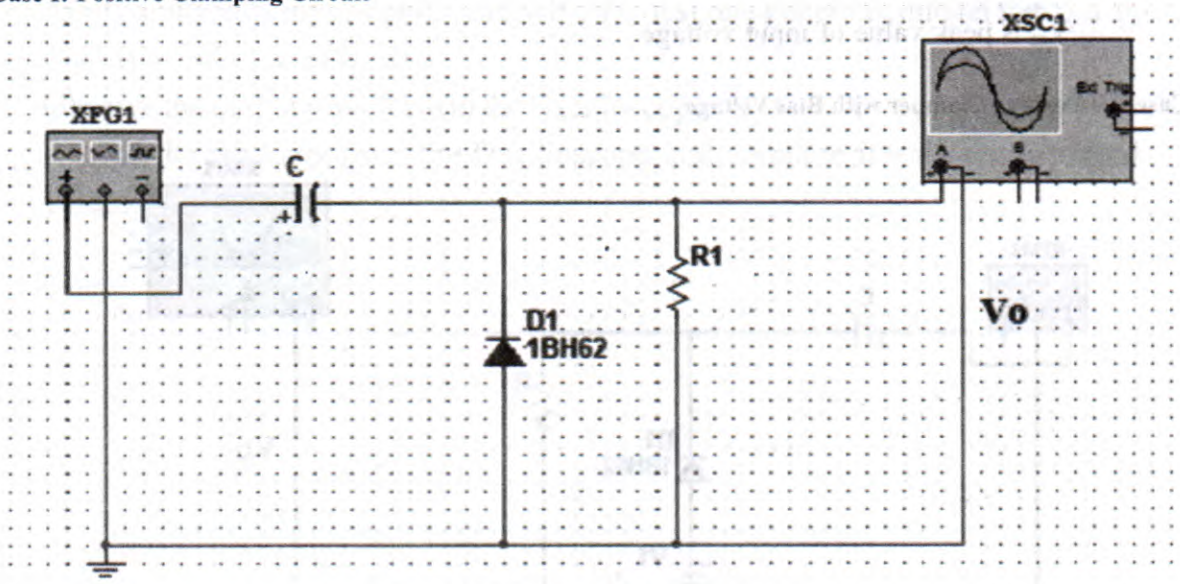


Figure 2.1: Positive Clamping Circuit

$$V_{out} = 2V_m \tag{2.3}$$

Where: V_{out} is the peak to peak amplitude of the output signal.

V_m is peak value of input voltage.

Case II: Negative Clamping Circuit

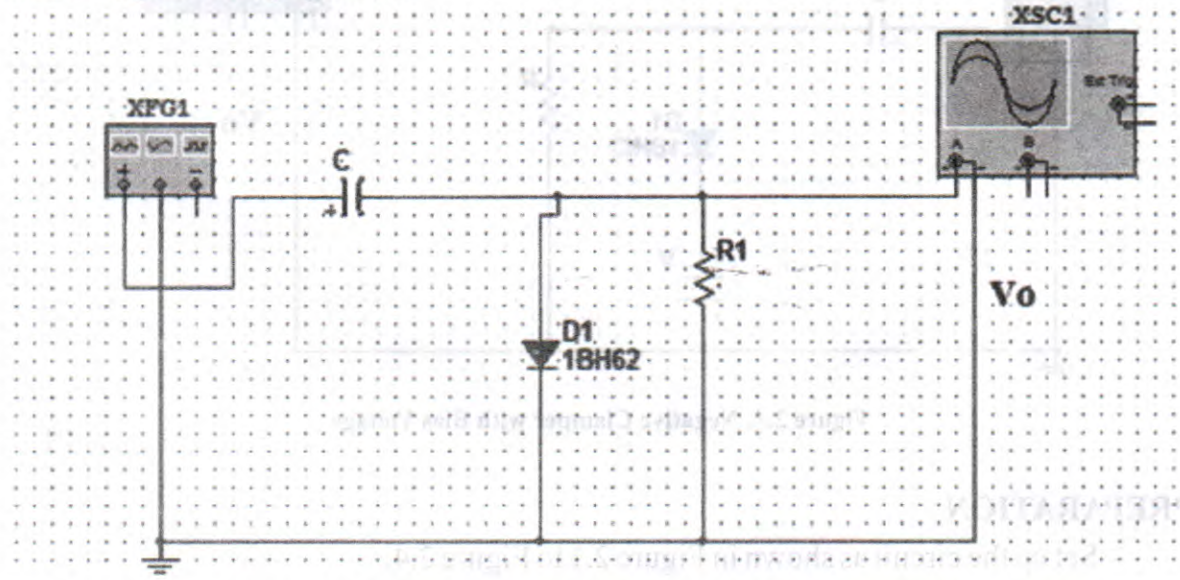


Figure 2.2: Negative Clamping Circuits

$$V_{out} = -2V_m \quad (2.4)$$

Where: V_{out} is the peak to peak amplitude of the output signal

V_m is peak value of input voltage

Case III: Positive Clamper with Bias Voltage

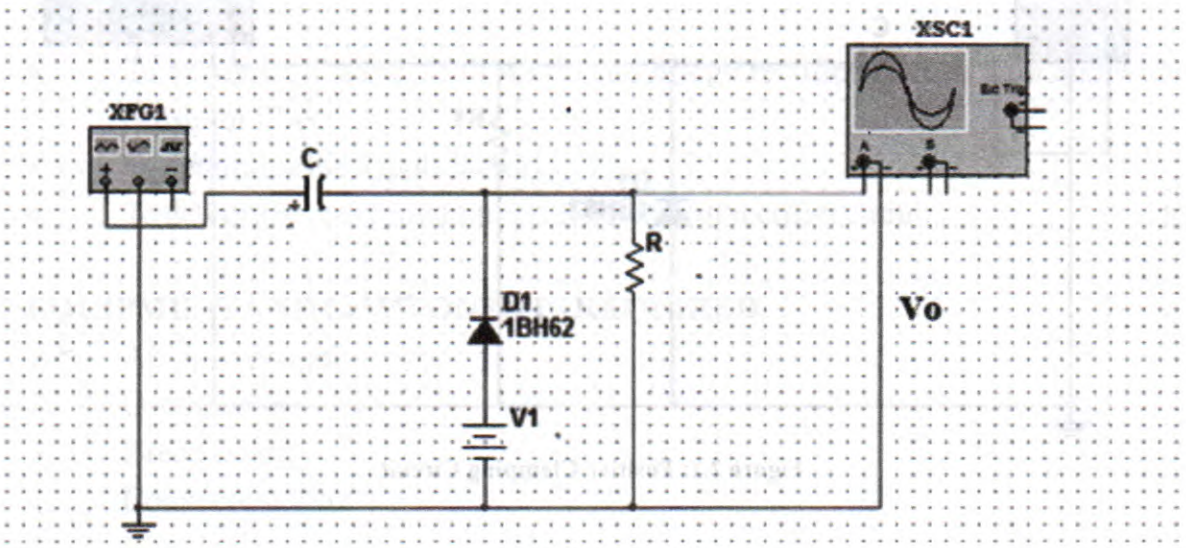


Figure 2.3: Positive Clamper with Bias Voltage

Case IV: Negative Clamper with Bias Voltage

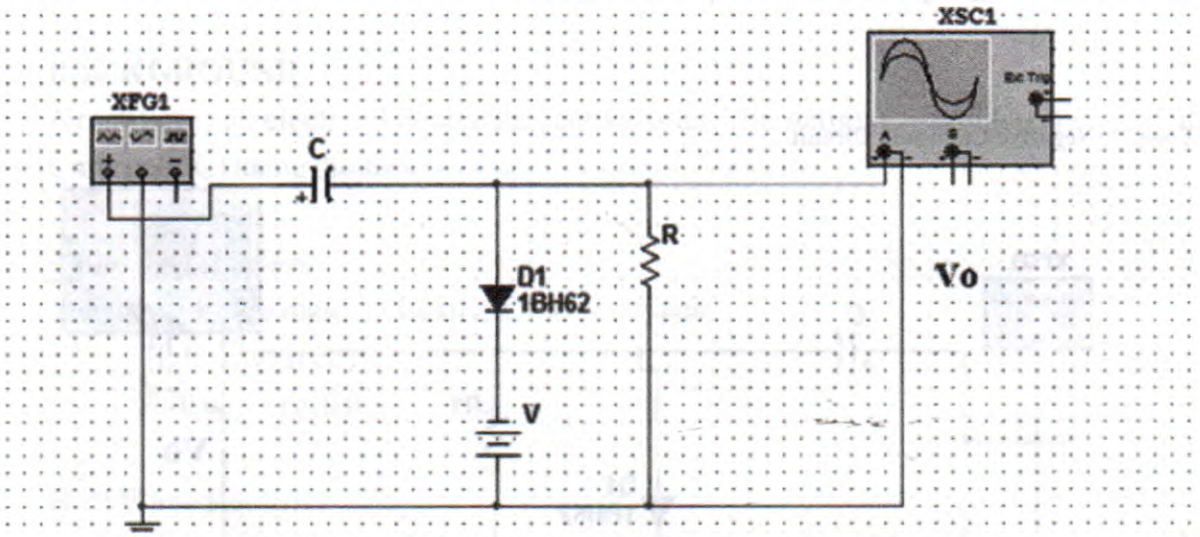


Figure 2.4: Negative Clamper with Bias Voltage

PREPARATION

1. Set up the circuit as shown in Figure 2.1 to Figure 2.4.
2. Take the frequency of the input voltage to be 1KHz and time constant to be 10 times the period, calculate the capacitance of the capacitor.
3. Calculate the output voltage during the positive and negative half cycle
4. Calculate the charging and the discharging time for the capacitors

EXPERIMENT

1. Apply sinusoidal input signal of 15V P-P from signal generator and let the V_{bias} in Case III = 10V, Case IV = 10V
2. Observe the output waveform in the Oscilloscope
3. Record the readings from the Oscilloscope and compare it with the expected values.

EXPERIMENT #: UILCPE/AEC 3

EXPERIMENT TITLE: Rectifier Circuits

OBJECTIVES

- To verify Half wave, full wave, and bridge rectifier circuits with and without capacitor filter
- To determine the Ripple factor
- To determine the regulation and efficiency of rectifier circuits

EQUIPMENT AND COMPONENTS REQUIRED

1. AC power supply
2. Oscilloscope
3. Bread Board
4. Connecting Wire
5. Resistors
6. Diodes
7. Transformer
8. Capacitor(s)

BACKGROUND

A rectifier is an electrical device that converts an Alternating Current (AC) into a Direct Current (DC) using diode(s).

Types of Rectifiers

The rectifiers are mainly classified into two types:

- Half wave rectifier
- Full wave rectifier

Half Wave Rectifier

As the name suggests, the half wave rectifier is a type of rectifier which converts half of the AC input signal (positive half cycle) into pulsating DC output signal and the remaining half signal (negative half cycle) is blocked or lost. In half wave rectifier circuit, we use only a single diode.

Full Wave Rectifier

The full wave rectifier is a type of rectifier which converts the full AC input signal (positive half cycle and negative half cycle) to pulsating DC output signal. Unlike the half wave rectifier, the input signal is not wasted in full wave rectifier. The efficiency of

full wave rectifier is high as compared to the half wave rectifier.

A. Design Calculation Formulae:

Assume RL = 4.7KΩ, C = 220μF

I. Half Wave Rectifier:

1. Ripple Factor without Filter = 1.21 (Standard Ripple Factor of half wave rectifier)
2. Percentage Regulation $= \frac{R_f}{R_L} \times 100$ (3.1)
3. Rectifier Efficiency $\eta = \frac{0.406}{1+\frac{R_f}{R_L}} = 40.6\ \%$ (3.2)
4. Ripple Factor without Filter $\gamma = \frac{1}{2\sqrt{3} f R_L C}$ (3.3)

   f = 50Hz

II. Full Wave Rectifier:

1. Ripple Factor without Filter = 0.48 (Standard Ripple Factor of Full wave center tapped rectifier)
2. Percentage Regulation = $\frac{\pi f}{R_L} \times 100$ (3.4)
3. Rectifier Efficiency $\eta = \frac{0.81}{1+\frac{R_f}{R_L}} = 81\ \%$ (3.5)
4. Ripple Factor without Filter = $\gamma = \frac{1}{4\sqrt{3} f R_L C}$ (3.6)

III. Bridge Rectifier:

1. Ripple Factor without Filter = 0.48 (Standard Ripple Factor of Bridge rectifier
2. Percentage Regulation $= \frac{R_f}{R_L} \times 100$ (3.7)
3. Rectifier Efficiency $\eta = \frac{0.81}{1+\frac{R_f}{R_L}} = 81\ \%$ (3.8)
4. Ripple Factor without Filter = $\gamma = \frac{1}{4\sqrt{3} f R_L C}$ (3.9)

A. Circuit Diagram and relevant Formulas for each Circuit

I. Half wave Rectifier without Filter:

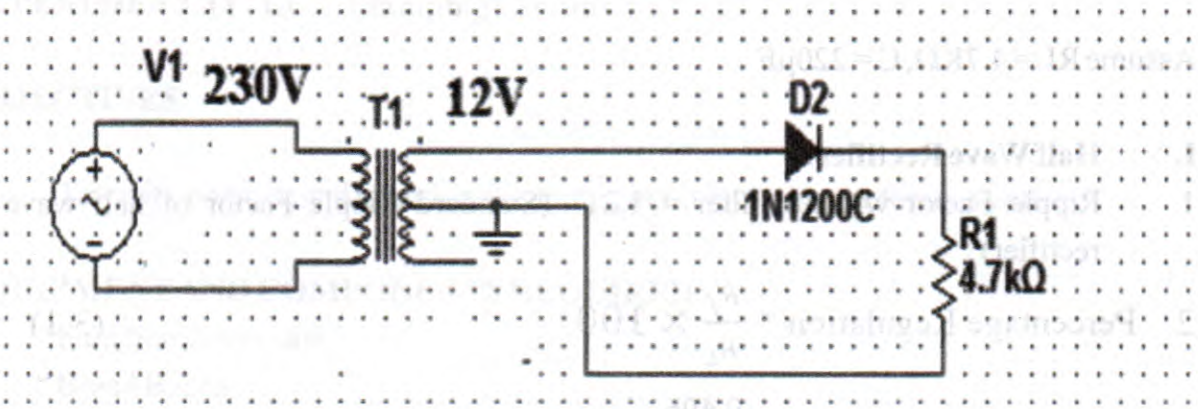


Figure 3.1: Half Wave Rectifier without Filter

$$V_{dc} = \frac{V_m}{\pi} \tag{3.10}$$

V_m = Peak output voltage

$$V_{rms} = \frac{V_m}{2} \tag{3.11}$$

$$V_{ac} = \sqrt{(V_{rms}^2 - V_{dc}^2)}. \tag{3.12}$$

The Ripple Factor, γ is as defined in equation 3.13

$$\gamma = \frac{V_{ac}}{V_{dc}} \tag{3.13}$$

Rectifier Efficiency η is as defined in equation 3.14.

$$\eta = \frac{P_{dc}}{P_{ac}} = \frac{V_{dc}^2}{V_{rms}^2} \tag{3.14}$$

$$\% \text{ Voltage Regulation} = \frac{V_{dc}(NL) - V_{dc}(FL)}{V_{dc}(FL)} \times 100 \tag{3.15}$$

II. Half Wave Rectifier with Filter:

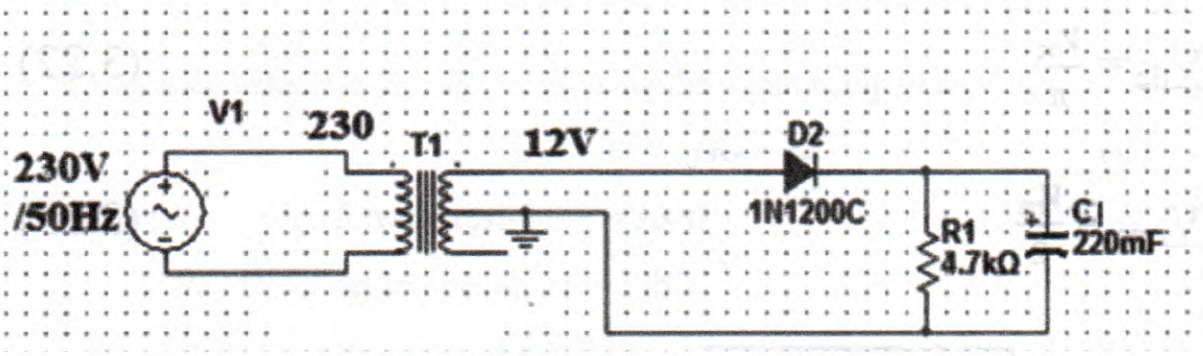


Figure 3.2: Half Wave Rectifier with Filter

$$\gamma = \frac{V_{ac}}{V_{dc}} \tag{3.16}$$

$$V_{dc} = \frac{V_m}{1+\frac{1}{2fR_LC}} \tag{3.17}$$

$$V_{ac} = \frac{V_{rp-p}}{2} \tag{3.18}$$

$$V_{rms} = \sqrt{(V_{dc}^2 + V_{ac}^2)} \tag{3.19}$$

$$\eta = \frac{P_{dc}}{P_{ac}} = \frac{V_{dc}^2}{V_{rms}^2} \tag{3.20}$$

$$\%\ \text{Voltage Regulation} = \frac{V_{dc}(NL) - V_{dc}(FL)}{V_{dc}(FL)} \times 100 \tag{3.21}$$

III. Full Wave Rectifier without Filter:

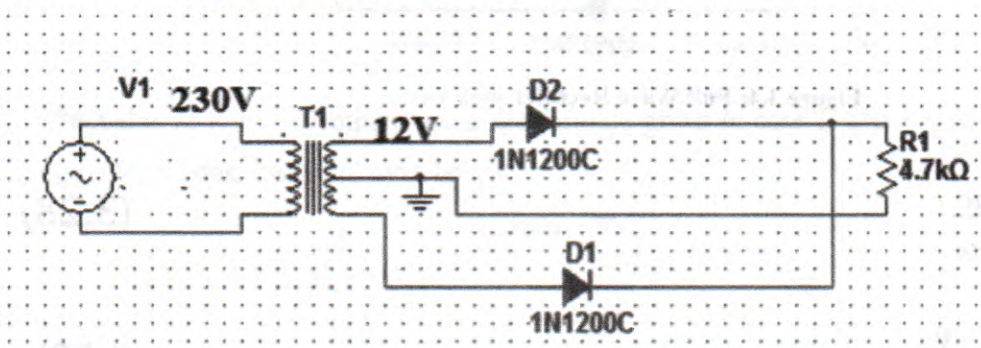


Figure 3.3: Full Wave Rectifier without Filter

$$V_{dc} = \frac{V_m}{\pi} \tag{3.22}$$

$$V_{rms} = \frac{V_m}{2} \tag{3.23}$$

$$V_{ac} = \sqrt{(V_{rms}^2 - V_{dc}^2)} \tag{3.24}$$

$$\gamma = \frac{V_{ac}}{V_{dc}} \tag{3.25}$$

$$\eta = \frac{P_{dc}}{P_{ac}} = \frac{V_{dc}^2}{V_{rms}^2} \tag{3.26}$$

$$\%\ \text{Voltage Regulation} = \frac{V_{dc}(NL) - V_{dc}(FL)}{V_{dc}(FL)} \times 100 \tag{3.27}$$

IV. Full Wave Rectifier with Filter:

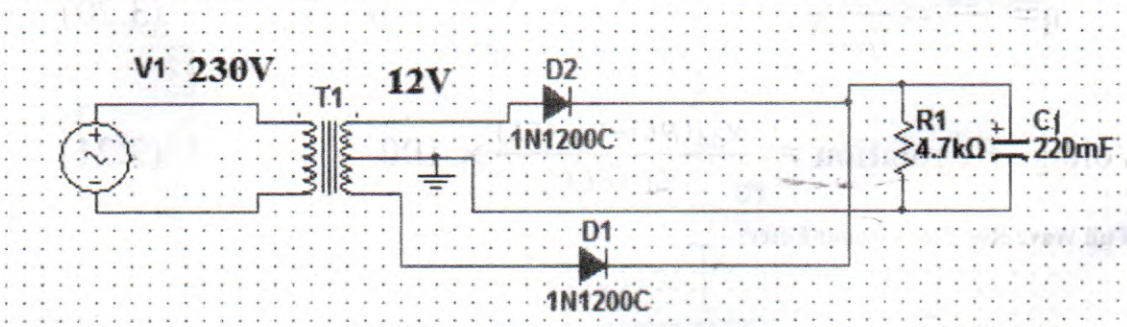


Figure 3.4: Full Wave Rectifier with Filter

$$\gamma = \frac{V_{ac}}{V_{dc}} \tag{3.28}$$

$$V_{dc} = \frac{V_m}{1 + \frac{1}{4fR_LC}} \tag{3.29}$$

$$V_{ac} = \frac{V_{rp-p}}{2\sqrt{3}} \quad (3.30)$$

$$V_{rms} = \sqrt{(V_{dc}^2 + V_{ac}^2)} \quad (3.31)$$

$$\eta = \frac{P_{dc}}{P_{ac}} = \frac{V_{dc}^2}{V_{rms}^2} \quad (3.32)$$

$$\% \text{ Voltage Regulation} = \frac{V_{dc}(NL) - V_{dc}(FL)}{V_{dc}(FL)} \times 100 \quad (3.33)$$

V. Bridge Rectifier without Filter:

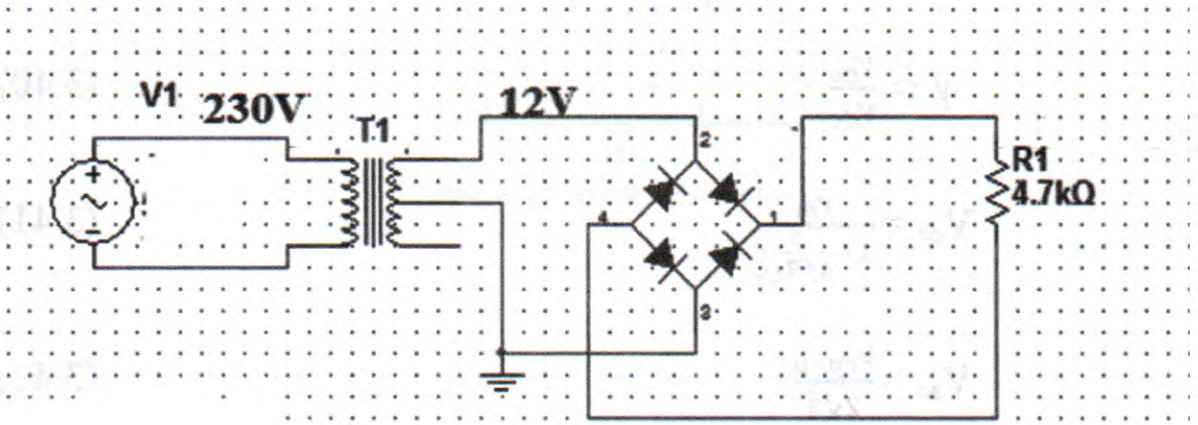


Figure 3.5: Bridge Rectifier without Filter

$$V_{dc} = \frac{2V_m}{\pi} \quad (3.34)$$

$$V_{rms} = \frac{V_m}{\sqrt{2}} \quad (3.35)$$

$$V_{ac} = \sqrt{(V_{rms}^2 - V_{dc}^2)} \quad (3.36)$$

$$\gamma = \frac{V_{ac}}{V_{dc}} \quad (3.37)$$

$$\eta = \frac{P_{dc}}{P_{ac}} = \frac{{V_{dc}}^2}{{V_{rms}}^2} \quad (3.38)$$

$$\%\text{Voltage Regulation} = \frac{V_{dc}(NL) - V_{dc}(FL)}{V_{dc}(FL)} \times 100 \quad (3.39)$$

VI. Bridge Rectifier with Filter:

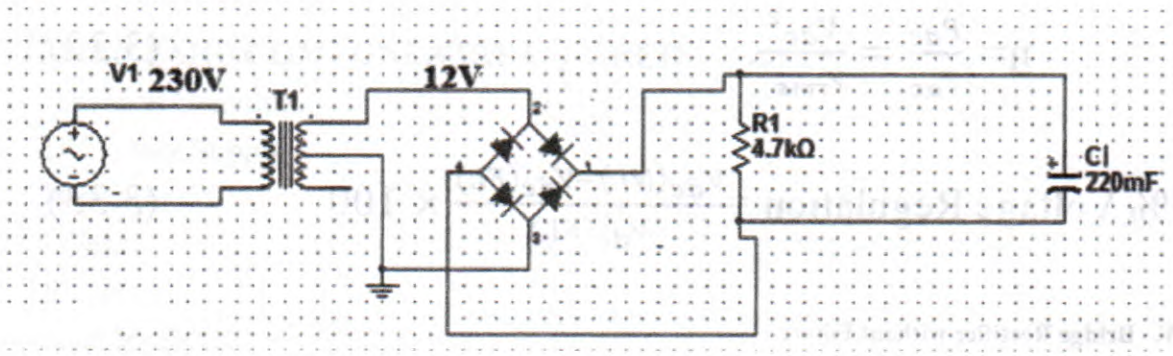


Figure 3.6: Bridge Rectifier with Filter

$$\gamma = \frac{V_{ac}}{V_{dc}} \quad (3.40)$$

$$V_{dc} = \frac{V_m}{1 + \frac{1}{4fR_LC}} \quad (3.41)$$

$$V_{ac} = \frac{V_{rp-p}}{2\sqrt{3}} \quad (3.42)$$

$$V_{rms} = \sqrt{({V_{dc}}^2 + {V_{ac}}^2)} \quad (3.43)$$

$$\eta = \frac{P_{dc}}{P_{ac}} = \frac{{V_{dc}}^2}{{V_{rms}}^2} \quad (3.44)$$

$$\%\text{ Voltage Regulation} = \frac{V_{dc}(NL) - V_{dc}(FL)}{V_{dc}(FL)} \times 100 \quad (3.45)$$

PREPARATION

In the circuit diagrams of Figures 3.1 to 3.6.

1. Use relevant formula to find Vdc and Vrms of both Full wave and Half wave rectifier
2. Find the Ripple factor, Regulation and Efficiency.

EXPERIMENT

Connect the circuit diagrams of Figure 3.1 to 3.6

1. Apply 230V AC supply from the power mains to the primary of the transformer to get 12V AC at the secondary side of the transformer

Observe the voltage across secondary to get Vm and draw the output waveforms.

EXPERIMENT #: UILCPE/AEC 4

EXPERIMENT TITLE: Transistor Biasing

OBJECTIVE

To verify the transistor biasing and its stability

EQUIPMENT AND COMPONENTS REQUIRED

1. Oscilloscope
2. Function Generator
3. Power Supply Unit
4. Bipolar Transistors
5. Breadboard
6. Capacitors
7. Resistors

BACKGROUND

Transistor biasing entails setting a transistor DC operating voltage or current conditions to the suitable level, so that any AC input signal can be amplified. A transistor steady state of operation relies on the base current, collector voltage and collector current. A typical transistor amplifier circuit is shown in Figure 4.1. The load line is obtained by plotting the voltage between the collector and the emitter VCE and the collector current IC as shown in Figure 4.2. The base current IB controls the collector current IC. Figure 4.1: Typical Transistor Amp

$$I_B = \frac{I_C}{\beta} \qquad (4.1)$$

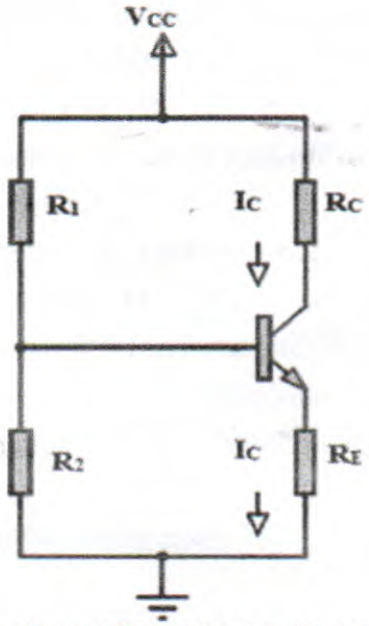


Figure 4.1: Typical Transistor Amplifier Circuit

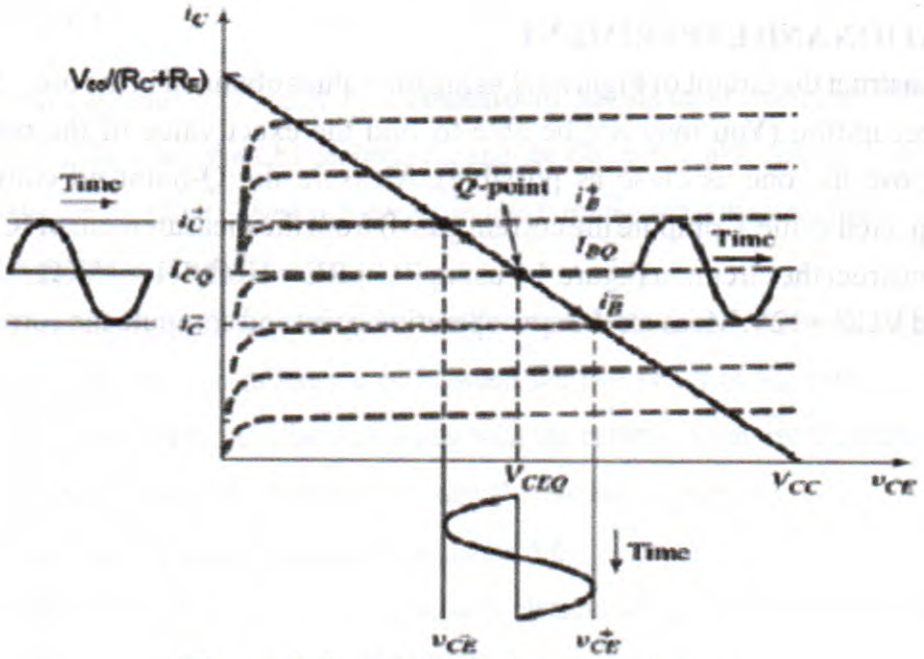


Figure 4.2: Typical DC Load Line and a Q-point

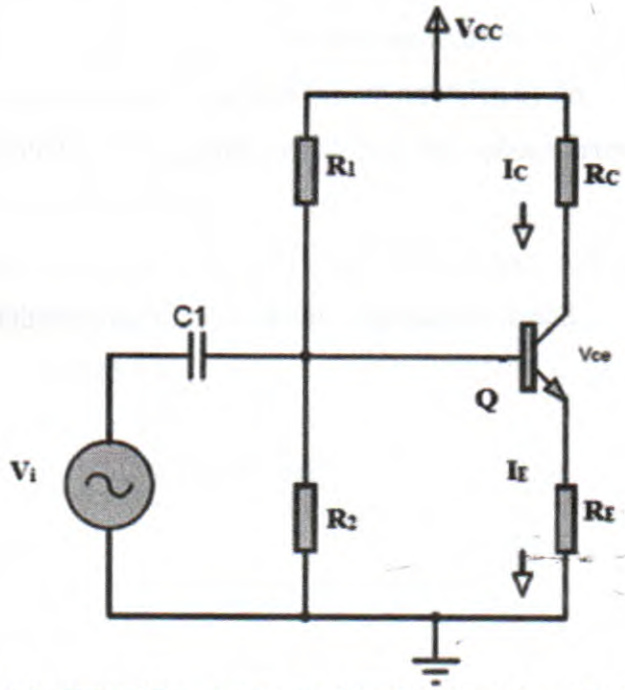


Figure 4.3: Typical Transistor Amplifier Circuit with an AC Input Signal

PREPARATION

Assume VCC = 12 V.

1. Consider the circuit in Fig. 4.1 with IB = 0.025mA, Determine RC and RE at the Q-point of the amplifier. Assume β = 220, VBE = 0.7V and IC ≈ IE.
2. Determine R1 and R2 of Fig. 4.1 Assuming I2 = 25IB

SIMULATION AND EXPERIMENT

1. Construct the circuit of Figure 4.1 using the values obtained in the pre- lab preparation (You may not be able to find the exact value of the resistors but choose the one as close as possible). Measure the Q-point nd compare with expected value. Compute the current gain β from the measurement of IC and IB.
2. Construct the circuit in Figure 4.1 using RC = RE = 1KΩ, R1 = 15KΩ, R2 = 9KΩ and VCC = 12V. Measure the new operating point and compute the current gain β.

EXPERIMENT #: UILCPE/AEC 5

EXPERIMENT TITLE: Transistor Small-Signal Amplifiers

OBJECTIVE

- To verify small signal transistor amplifiers

EQUIPMENT AND COMPONENTS REQUIRED

1. Oscilloscope
2. Function Generator
3. Power Supply Unit
4. Bipolar Transistors
5. Breadboard
6. Capacitors
7. Resistors

BACKGROUND

An important application of bipolar transistors is in small-signal amplifiers. These are systems that accept input signal of small amplitudes (on the order of 100mV) and deliver larger amplitude. The amplifier voltage gain g can be determined using equation 5.1

$$g = \frac{v_{out}}{v_{in}} \qquad (5.1)$$

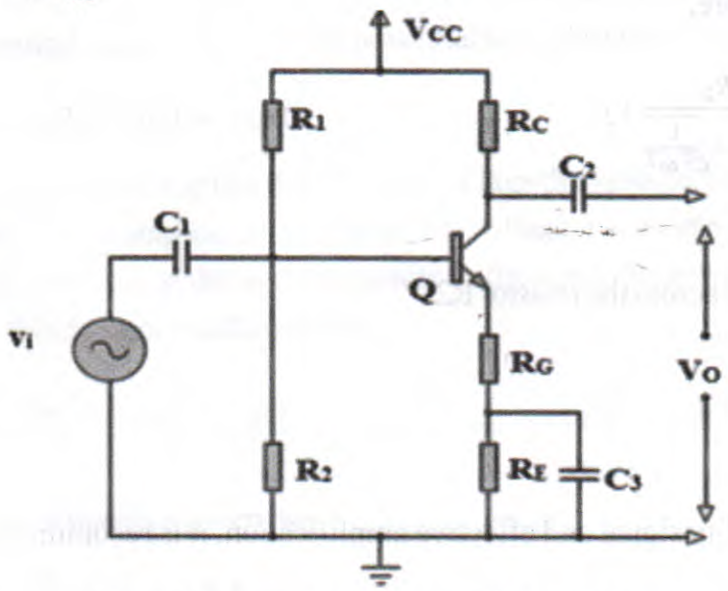


Figure 5.1: Common Emitter Amplifier Circuit

C1 = Input capacitor chosen to block DC voltage from getting to the transistor
C2 = Output capacitor chosen to block any DC voltage from getting to the output
C3 = Emitter capacitor is chosen so that AC voltage around RE is short circuits effectively

Since the entire AC input voltage appears across RG, therefore from Ohm's law:

$$i_E = \frac{V_1}{R_G} \tag{5.2}$$

$$V_o = i_C R_C = i_E R_C \tag{5.3}$$

$$V_o = \frac{R_C}{R_G} v_I \tag{5.4}$$

The AC amplifier gain g is given by

$$g = \frac{V_{output}}{V_{input}} = \frac{R_C}{R_G} \tag{5.5}$$

Note: The amplifier gain g must be less than the current gain β of the transistor.
The circuit in Fig. 5.1 passes the larger frequency of the input voltage vi to the voltage across the resistor R2. Capacitor C1 and resistor R2 act as a voltage divider of the input voltage vi. Therefore,

$$v_r = \frac{R_2}{\sqrt{R^2 + \frac{1}{C^2\omega^2}}} v_I \tag{5.6}$$

Where:

vr = voltage across the resistor R2
C = C1
ω = 2πf
R = R2

To achieved a quality signal and effective amplification, it is recommended to choose C such that;

$$v_r = \frac{v_1}{2} \tag{5.7}$$

Output capacitor C_2 can also be obtained using equation (5.6) and (5.7), where:
vr = voltage across C_2
$R = R_2 = RC$
$v_i = vo$
For the emitter capacitor C_E to short circuit the AC voltage around the emitter resistor, the time constant of the resistor and capacitor is therefore;

$$\tau = RC = \frac{1}{f} \qquad (5.8)$$

Where:

$$R = R_E, \qquad C = C_E$$

PREPARATION

Consider the circuit of Figure 5.1,

1. Calculate the values of R_1, R_2, R_C and R_E.
2. Determine the gain of the amplifier given $R_G = 36\Omega$ and hence the output voltage.
3. Determine the capacitors C_1, C_2 and C_3.

SIMULATION AND EXPERIMENT

1. Construct the circuit of Figure 5.1 with the values obtained in the preparation.
2. Set the function generator to 200Hz, 5Vp-p. Measure the small-signal voltage *vi* and *vo* using an oscilloscope, then compute *g*.
3. Compare your results with the one obtained in the preparation.
4. Double the value of R_G, measure and compute the new amplifier gain g.

EXPERIMENT #: UILCPE/AEC 6

EXPERIMENT TITLE: Transistor as a Switch

OBJECTIVES

- To verify the transistor cut-off and saturation operations
- To study the use of transistor as an electronic switch

EQUIPMENT AND COMPONENTS REQUIRED

1. Oscilloscope
2. Function Generator
3. Power Supply Unit
4. Bipolar Transistors
5. Breadboard
6. Resistors

BACKGROUND

Modern embedded equipment uses binary logic of base 2 (0 and 1) to perform their functions. Transistor is one of the fundamental component configure as a switch. A transistor connected in a common-emitter configuration invert a signal such that if a high-input signal is applied, a low-output signal is obtained and vice-versa. It can also be used to switch ON and OFF some electronics and electromagnetics devices such as DC motors, lamps/LEDs, relays and some mechanical actuators.

In normal transistor switching operation, transistor is saturated such that VCE is approximately 0.2V and $IC \neq \beta I_B$. Due to the characteristics of the PN junctions, the transistor cannot instantly turn on and off. The rise time of the output signal, t_r, is defined as the time it takes the output to rise from 10 percent to 90 percent of its final "ON" state. Likewise, the fall time, t_f, is the time it takes to fall from 90 percent to 10 percent of its final "OFF" state. Delay time, t_s, is the time between when the input has gone to zero and when the output falls to 90 percent of its "ON" value as illustrated in Fig. 6.1.

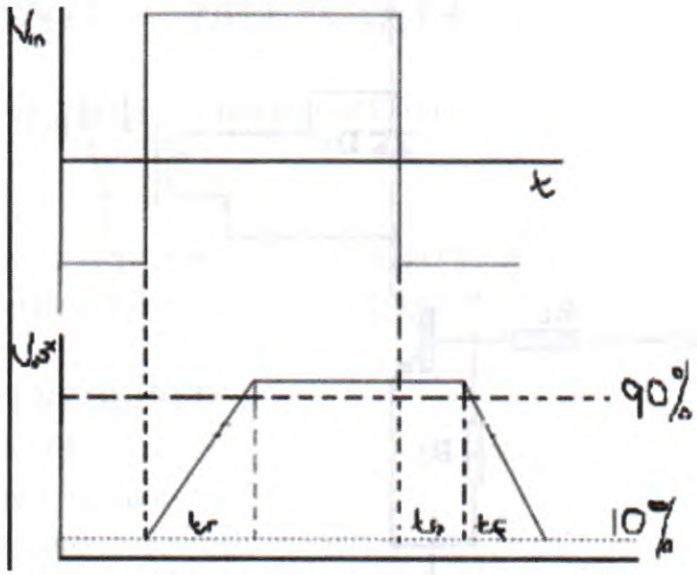


Figure 6.1: Description of the Rise, Delay and Fall time

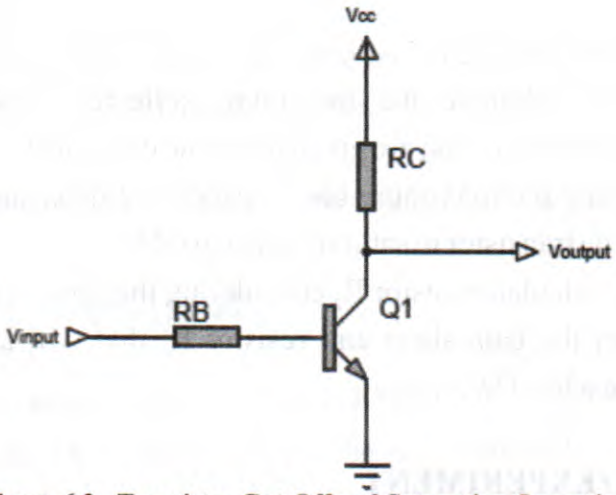


Figure 6.2: Transistor Cut-Off and Saturation Operation

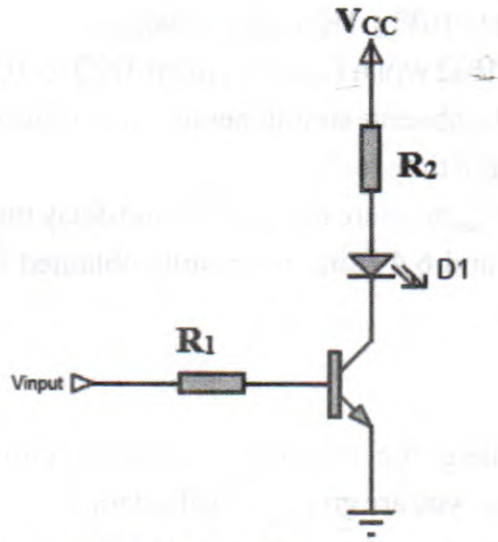


Figure 6.3: Using Transistor to drive LED

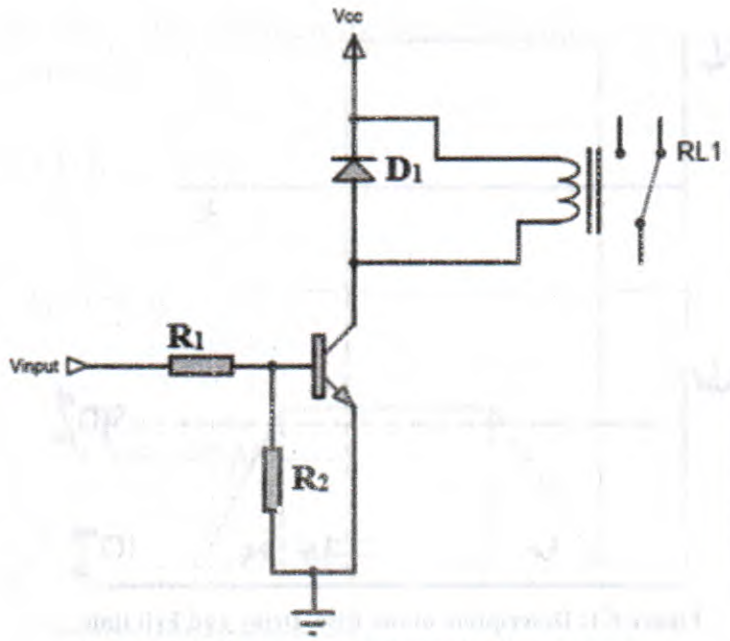


Figure 6.4: Using a Transistor to Control a Relay

PREPARATION

1. From Fig. 6.4, calculate the maximum collector current considering the specification of the relay you are given using the data sheet.
2. From 1, calculate the maximum base current and determine resistor R_1 and R_2 that will keep the transistor in saturation when ON.
3. From Fig. 6.3, calculate resistor R_2 considering the specification of the LED you are given from the data sheet and resistor R_1 that will keep the transistor in saturation state when ON.

SIMULATION AND EXPERIMENT

Construct the circuit shown in Figure 6.2, $V_{CC} = 12V$

1. Set the signal generator to $10V_{p\text{-}p}$, 1KHz square wave.
2. R_C should be fixed at 1KΩ while R_B varies from 1KΩ to 100KΩ.
3. The V_{in} and V_{out} should be observe simultaneously on the oscilloscope.
4. Determine the rise and fall time for $V_{in.}$
5. Determine and sketch V_{out}, measure the rise, fall and delay time.
6. Construct Figures 6.3 and 6.4 using the results obtained from the preparation. State your observation.

PREPARATION

1. From Fig. 6.4, calculate the maximum collector current considering the specification of the relay you are given using the data sheet.
2. From 1, calculate the maximum base current and determine resistor R_1 and R_2 that will keep the transistor in saturation when ON.

3. From Fig. 6.3, calculate resistor R_2 considering the specification of the LED you are given from the data sheet and resistor R_1 that will keep the transistor in saturation state when ON.

SIMULATION AND EXPERIMENT

Construct the circuit shown in Figure 6.2, $V_{CC} = 12V$

1. Set the signal generator to $10V_{p-p}$, 1KHz square wave.
2. R_C should be fixed at 1KΩ while R_B varies from 1KΩ to 100KΩ.
3. The V_{in} and V_{out} should be observe simultaneously on the oscilloscope.
4. Determine the rise and fall time for V_{in}.
5. Determine and sketch V_{out}, measure the rise, fall and delay time.
6. Construct Figures 6.3 and 6.4 using the results obtained from the preparation. State your observation.

EXPERIMENT #: UILCPE/SDT 1

EXPERIMENT TITLE: Introduction to Computer Programming

OBJECTIVES:

- To familiarize the student with the computer environment
- To understand the importance of algorithms in problem solving.

BACKGROUND

Computer Programming is the act of writing special keywords or syntax to solve a particular problem. It involves the procedure of converting algorithm to codes, syntax or language understood by the IDE compiler to generate program file. Programming Language is categorized into three languages namely: low-level language (Machine language (ML)), Assembly language (Mnemonics) (AL) and high level language (HLL). The machine language or low level language are special keywords that are binary bits (0 or 1) which are executed and understood directly by the computer while the Assembly language or Mnemonics are keywords or syntax that are interpreted into the machine language before execution and it is more user-friendly compared to the machine language.

The high level language are keywords or syntax which is English-like language known as source code that are converted by the IDE compiler or interpreter into the machine language known as the object code. The high level language is mostly used for developing complex programs and it is the most user-friendly programming language which can be easily utilized by the developer to interpret algorithm and solve sophisticated problems, example of high level language are PROLOG, C, C#, C++, VISUAL BASIC, FORTRAN and JAVA.

EXPERIMENT

Task One

The first task is to gather basic information about the computer you are using. The instructor will assist you on this task if required. Turn on the computer you are using and get the following information about the system:

1. Operating System
2. Processor type and speed
3. Hard disk capacity
4. List at least five application software on the system

5. List any existing high level programming language on the system

Task Two

1. Open any word processing application on the system you are using and write an algorithm, pseudocode or draw a flowchart for the following problems:

$$\bar{x}_i = \frac{1}{i} \sum_i^n x_i; \quad \forall i = 1 \ldots n \qquad (1.1)$$

1. Sort an array of 50 integers in ascending order of magnitude
2. Multiply two matrices $C_{mp} = A_{mn} \times B_{np}$

Task Three

Installing the NetBeans IDE

1. You are provided with the installation file of the NetBeans IDE. Click on the icon and follow the necessary steps to install the software Please note that
   - It is important to make sure you are using the correct version of the software for your system specification.
   - If you are having any problem installing the software, notify your lab instructor.

EXPERIMENT #: UILCPE/SDT 2

EXPERIMENT TITLE: JAVA IDE

OBJECTIVE:

· To introduce the student to the NetBeans Integrated Development Environment (IDE) and how to write simple Java programs

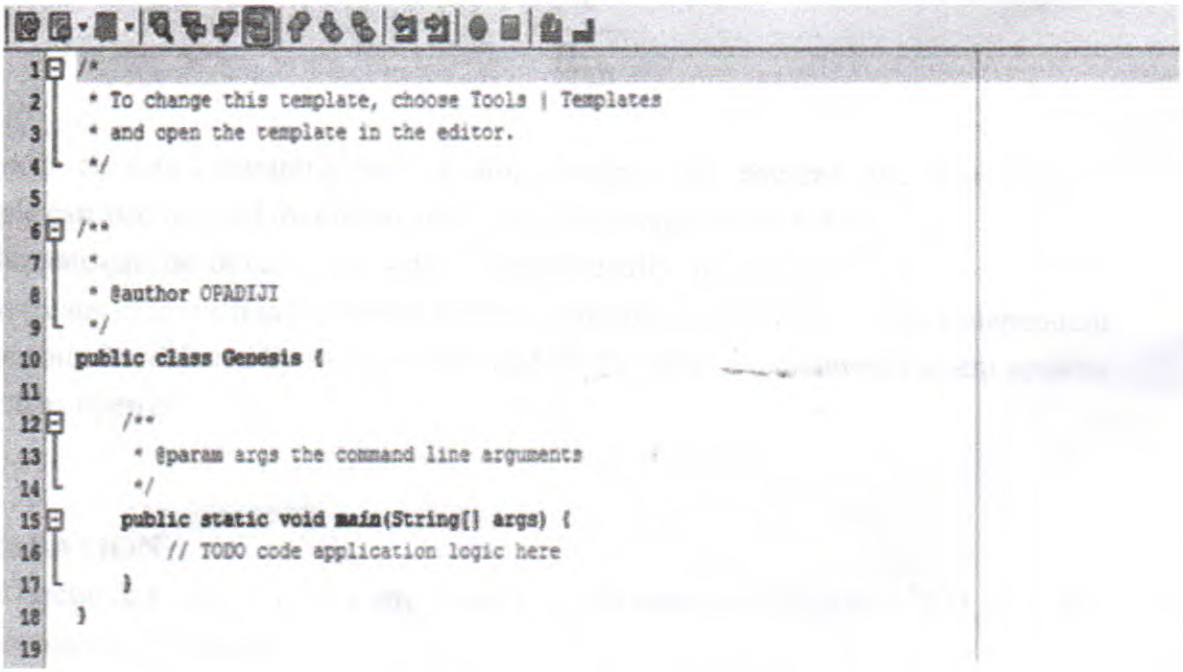
BACKGROUND

Java programming is a high level language. High level languages are similar to regular human languages such as English. Java is a fully object-oriented programming language with classes being its basic building blocks. The NetBeans IDE is one of the most popular development environment for Java and as the one we will be using in this laboratory.

PREPARATION

Students are expected to review their departmental course materials on JAVA programming.

EXPERIMENT



```
1  /*
2   * To change this template, choose Tools | Templates
3   * and open the template in the editor.
4   */
5
6  /**
7   *
8   * @author OPADIJI
9   */
10 public class Genesis {
11
12     /**
13      * @param args the command line arguments
14      */
15     public static void main(String[] args) {
16         // TODO code application logic here
17     }
18 }
19
```

Figure 2.1: NetBeans IDE

Writing your first program

1. Click on the NetBeans icon on your desktop.
2. Go to the file menu and select New Project.
3. In the New project window, select java under category and java application under project, and then click next.
4. Enter a name for the project and uncheck the set name for main class box.
5. To create a new application, go to the file menu and select new file.
6. Select Java under categories and Java Main class under file type.
7. You should see Figure 2.1. Note: the name of the file is the same as the name of the main class created.
8. Type "System.out.println("This is my first java program");" into the "//TODO code application logic here" Section of Figure 2.1.
9. Run the program by pressing the key combination Shift + F6 or Click on run under the run menu.

Writing a GUI Program

We are going to write the same program written in the last task and add GUI capabilities.

1. Type the program below into the editor of your IDE

```
1  import javax.swing.JOptionPane;  //importing a dialogue BOX
2     public class Genesis2 {
3    public static void main(String[] args)
4  {
5    JOptionPane.showMessageDialog(null,"This is my first Java Program");
6    System.exit(0);  //end execution
7    }
8    }
```

Figure 2.2: GUI Program Code

2. Run your program and observe the output.
3. Type the program code in Figure 2.3 into the editor of your IDE.

```
import javax.swing.JOptionPane;   //importing a dialogue box class
public class rectangle {
public static void main(String[] args) {
        String firstinput,Secondinput;double Length;double breath;double Area;
         //open input dialogue box for first input
         firstinput=JOptionPane.showInputDialog("The length of the rectangle is");
         Length=Double.parseDouble(firstinput); //convert string input to double
//open input dialogue box for Second  input
         Secondinput=JOptionPane.showInputDialog("The breath  of the rectangle is");
         breath=Double.parseDouble(Secondinput); //convert string input to double
         Area=Length*breath;  //compute area of rectangle
         JOptionPane.showMessageDialog(null,"The area of the rectangle is"+" "+Area);
         System.exit(0);  //end execution
         }}
```

Figure 2.3: GUI Program to Compute Area of a Rectangle

4. Run your program and observe the output
5. Replace PLAIN_MESSAGE in the program code with the following:
   i. ERROR_MESSAGE
   ii. INFORMATION_MESSAGE
   iii. WARNING_MESSAGE
   iv. QUESTION_MESSAGE
6. Create another file called Rectangle2, type the following code, and observe the output difference with the rectangle program output.

```
import java.text.DecimalFormat;
import javax.swing.JOptionPane;
import javax.swing.JTextArea;
public class rectangle2 {
public static void main(String[] args) {
         DecimalFormat df=new DecimalFormat("0.0");
         JTextArea output =new JTextArea(20,10);
         String firstinput,Secondinput;double Length;double breath;double Area;
          firstinput=JOptionPane.showInputDialog("The length of the rectangle is");
          Length=Double.parseDouble(firstinput);
          Secondinput=JOptionPane.showInputDialog("The breath  of the rectangle is");
          breath=Double.parseDouble(Secondinput);
          Area=Length*breath;
          output.append(df.format(Area));
          JOptionPane.showMessageDialog(null,output,"Area of Rectangle",
          JOptionPane.INFORMATION_MESSAGE);
          System.exit(0);
          }}
```

Figure 2.4: Output with JTextArea Capability

LABORATORY ASSIGNMENT

1. Create a project called Practical. You will use this project as a container for all the exercises you do in the lab all through the semester.
2. Create a simple program that will allow you to print any message on the screen and name the application code1.java.
3. Write two simple java programs that will make use of the input and output dialogue boxes in the javax.swing library.

EXPERIMENT #: **UILCPE/SDT 3**

EXPERIMENT TITLE: **Flow Control, Arrays and Classes**

OBJECTIVES:

- To introduce flow control, arrays and concept of classes in programming.
- To teach the students how to create Java packages using NetBeans.

BACKGROUND

Classes

Classes are like molds used to create objects. Just like every brick that comes from the same mold are identical, all object created from the same class are identical. They have the same properties and understand the same messages. A Java program is constructed as a collection of classes used to create objects in the program. There are two structural parts of a class: Attributes (properties) and Behavior (messages).

Attributes are the structural part of a class that determines the properties of the objects created from the class. They are referred to as the data members of the class.

Behaviours are the structural part of a class that determines the messages which are understood by objects created from the class. They are referred to as the member functions of the class.

Flow Control

Statements in a program are executed in the order in which they are written except the machine is instructed to do otherwise. Keywords used to specify what the next statement to be executed is are called program flow control keywords.

- Branching control keywords
- Looping control keywords

A. if Statement

This control statement is of the form if <conditional expression><execute statement>
if (cgpa>= 4.5) degreeClass = "First Class"; The statement to be executed if conditional expression is true may be a simple statement or a block statement

B. if ... else Statement

This is similar to the if statement except for the fact that there is another statement or group of statements to be executed if the conditional expression is false
if<conditional expression><execute statement> else <execute statement>
if (income >= 200000.00) tax_rate = 0.3; else tax_rate = 0.14;

C. While Statement

The while statement allows the iteration of a statement or group of statements. The conditional expression is tested before any statement is executed

The Syntax is while <conditional expression><execute statement>

D. switch … case Statement

The switch … case statement is used for multiple options

E. Do While Statement

The do … while statement is similar to the while statement except for the fact that statements in the loop are executed at least once before the conditional expression is tested.

do<statement> while <conditional expression>

F. For Loop Statement

Like all looping controls, the for statement requires:

A counter (or control) variable, an initial value of the counter variable, an incrementing or decrementing expression to proceed in the iteration, a stopping condition for the counter variable

The format for the for statement is

For (expression1; expression2; expression3)

expression1 >>> initialization of control variable, expression2 >>> stopping condition for loop

expression3 >>> incrementing or decrementing control character

Arrays

An array is a collection of items that have the same data type, class or belong to the same parent class. All the elements in an array are stored contiguously in memory under the same name.

In order to create an array, one must Declare a variable to hold the array, Create a new array object and assign it to the declared variable Store information in that array. Every element of an array is addressed by an integer subscript, e.g., given an array k having ten elements, the elements are k0, k1, k2, … k9.The first element carries the subscript 0. This is the naming convention in Java. Since subscripts are not written in Java programs, they are written as indices in the form of k(0), k(1), k(2), …, k(9). An index indicates the position of an element in the array. These are examples of array declaration in Java:

- int() quantity; // int quantity(); is also a valid declaration
- double() energy; // double energy(); is also a valid declaration
- String() surname; // String surname(); is also a valid declaration
- Shape() circle; // Shape circle(); is also a valid declaration

PREPARATION

Students are expected to review their departmental course materials on JAVA programming.

Classes

Type and run the code below

```
1  public class classwork3{
2  public static void main(String[] args) {
3  person mydata = new person();
4  mydata.showdata();
5  }
6  }
7  class person{
8  String Myname = "Goodluck Ebele";
9  String Sex = "Male";
10 int Age = 60;
11 public person()
12 {
13 }
14 public void showdata()
15 {
16 System.out.println("The details for the Presidential candidate is" +"\n"+
17 Myname +" \n" + Sex + "\n" + Age );
18 }
19 }
20
```

Figure 3.1: Program Code to write a Class

2. Modify the code by introducing other properties in the class person and also creating more objects from the same class. Print out all the properties for all the objects you created.
3. Add one more class to the program with the name shape and use the class to compute and print out the area of any two dimensional shape of your choice.

Controlling the Flow of Program

Type and run the code below

```
import javax.swing.JOptionPane;
public class operators{
public static void main(String[] args) {
        int x,y;
     String Value1=JOptionPane.showInputDialog("Provide the first value");
       x=Integer.parseInt(Value1);
        String Value2=JOptionPane.showInputDialog("Provide the Second value");
       y=Integer.parseInt(Value2);

      String operator=JOptionPane.showInputDialog("Provide the Operator");
       if(operator.equals("*")){
     int result=(x*y);
      JOptionPane.showMessageDialog(null,"The Multiplication of the two numbers provided is "+result);
     }
      if (operator.equals("+")){
     int result=(x+y);
      JOptionPane.showMessageDialog(null,"The addition of the two numbers provided is "+result);
     }
     if(operator.equals("-")){
     int result=(x-y);
      JOptionPane.showMessageDialog(null,"The Subraction of the two numbers provided is "+result);
     }
      if(operator.equals("/")){
     int result=(x/y);
      JOptionPane.showMessageDialog(null,"The Division  of the two numbers provided is "+result);
      }}
```

Figure 3.2: Program to perform four operators on any two numbers

```
2  import javax.swing.JOptionPane;
3  public class MYCGPA {
4  public static void main(String[] args) {
5  // TODO code application logic here
6  int a1 = 0,a21 = 0,a22 = 0,a3 = 0,apass = 0,StudentPopulation, counter;
7  double cgpa; String entry;
8  entry = JOptionPane.showInputDialog("How many students are in the class");
9  StudentPopulation = Integer.parseInt(entry);
10 // group students based on their cgpa
11 for ( counter = 1; counter <= StudentPopulation; counter++)
12 {
13 entry = JOptionPane.showInputDialog("Enter CGPA for student " + counter);
14 cgpa = Double.parseDouble(entry);
15 if (cgpa >= 4.50) a1++;
16 else
17 if (cgpa >= 3.50) a21++;
18 else if (cgpa >= 2.50) a22++;
19 else if (cgpa >= 1.50) a3++;
20 else if (cgpa >= 1.00) apass++;
21 //counter++;
22 }
23 JOptionPane.showMessageDialog(null,"Number of students in First Class Division = " + a1
24 + "\nNumber of students in Second Class, Upper division = " + a21
25 + "\nNumber of students in Second Class, Lower division = " + a22
26 + "\nNumber of students in Third Class = " + a3
27 + "\nNumber of students in Pass division = " + apass);
28 System.exit(0);
29 }
30 }
```

Figure 3.3: Program to determine the Class of Degree of any CGPA entered

1. Modify the program in Figure 3.2 to give four attempts to the users.
2. Modify the program in Figure 3.3 to decrement the control character.
3. Write a java program that will print out the corresponding letter grade "A – F" for any score entered by the user using the if statement or if … else statement.
4. Write a program to determine the number of odd numbers from a set of 20 integers entered by the program user.
5. Rewrite the programs in the first three tasks using a different program flow control statement to the one you used previously.

Arrays

```
1  import javax.swing.JOptionPane;
2  import javax.swing.JTextArea;
3  public class arrayadd {
4  public static void main(String[] args) {
5  // TODO code application logic here
6  int[][] Firstmatrix = {{3,2,4,5,6,2},{8,4,8,3,1,2},{0,0,7,8,3,2},{2,2,2,1,1,2},{4,4,4,4,7,2},{4,2,4,8,7,2}};
7  int[][] Secondmatrix = {{1,0,8,0,0,3},{4,5,5,5,1,3},{3,1,1,1,2,3},{1,5,6,7,0,3},{8,4,9,9,1,3},{4,7,8,4,7,2}};
8  int[][] thirdmatrix = {{1,0,8,0,0,3},{4,5,5,5,1,3},{3,1,1,1,2,3},{1,5,6,7,0,3},{8,4,9,9,1,3},{4,7,8,4,7,3}};
9  int[][] matrixAddition = new int[6][6];
10 int x = 0, y = 0;
11 JTextArea output = new JTextArea(10, 20);
12 for (x = 0; x < 6; x++)
13 {
14 for (y = 0; y < 6; y++)
15
16 {
17
18 matrixAddition[x][y] = Firstmatrix[x][y] + Secondmatrix[x][y]+thirdmatrix[x][y];
19 output.append(matrixAddition[x][y] + "\t");
20 }
21 output.append("\n\n\n"); //whatever you expect to happen to output.append you can add it
22 }
23 JOptionPane.showMessageDialog(null,output,"Matrix",1);
24 System.exit(0);
25 }
26 }
```

Figure 3.4: Addition of three Matrices

EXPERIMENT

```
1  import javax.swing.JOptionPane;
2  import java.text.DecimalFormat;
3  import javax.swing.JTextArea;
4  public class MovingAverage {
5  public static void main(String[] args) {
6  int k= 0, a = 0;double d =0;
7  String i = "";
8  String [] InputNumber = new String[10];
9  double[] ConvertNumber = new double[10];
10 double[] NumberOutput = new double[10];
11 JTextArea output = new JTextArea(20, 30);
12 DecimalFormat df=new DecimalFormat("0.00");
13 while (a< 1 || a > 10)
14 {
15 i = JOptionPane.showInputDialog(null, "How many names are in the array (1 - 10)?");
16 a = Integer.parseInt(i);
17 }
18 output.append("Number\tmovingAverage\n");
19 for (k = 0; k <a; k++)
20 {
21 InputNumber[k] = JOptionPane.showInputDialog(null,"Enter Text " +(k+1));
22 ConvertNumber[k]=Double.parseDouble(InputNumber[k]);
23 d=ConvertNumber[k]+d;
24 NumberOutput[k]=d/(k+1);
25 output.append((df.format(k+1)+"\t"+df.format(NumberOutput[k])+"\n"));
26 }
27 JOptionPane.showMessageDialog(null,(output),"Text Output",1);
28 System.exit(0);
29 }
30 }
```

Figure 3.5: Program to compute Moving Average

1. Type and run the code in Figure 3.5 and observe the output.
2. Using only the main method, write a program to find the average hourly temperature for each day of the week for the first 31 days in the year given that a user will enter hourly temperature data for your program. Show a formatted output of your result.
3. Create your own package and add the program you wrote in 2 to the package using the NetBeans refactor tool.
4. Rewrite the program in 2 but this time creating a class with a method that will do the computation and format the output.

EXPERIMENT #: UILCPE/SDT 4

EXPERIMENT TITLE: Methods and Interfaces

OBJECTIVES:

- To introduce built- in methods in the Java API
- To introduce the creation of new methods and interfaces
- To introduce Exception and Multi-threading

BACKGROUND

A method is a set of codes which is referred to by name and can be called (invoked) at any point in a program simply by utilizing the method's name. Think of a method as a subprogram that acts on data and often returns a value. Each method has its own name. It can also said to be a collection of statements that are grouped together to perform a specific task. Once a method is created a user can then call it to perform a desired task. Basically there are two ways in which a method is called .It can return a value or returning nothing (no return value).

Interface is very close to class but it is not a class. It can have methods and variables just like the class but the methods declared in interface are by default abstract. A Java class can implement multiple Java Interfaces . Variables declared in an interface are public, static and final by default. An interface definition begins with the keyword interface and contains a set of public abstract methods in addition to methods, interfaces may also contain variables but the variables have to be public final static. An example is given in the code snippet below:

```
public interface workforce
public abstract double compute Income;
```

PREPARATION

This laboratory session is to give the students ample practice with the concepts taught in course materials on Java programming.

EXPERIMENT

Methods

```
1
2  package testPackage;
3  import javax.swing.JOptionPane;
4  import javax.swing.JTextArea;
5  public class StringArrange {
6  public static void main(String[] args) {
7  // TODO code application logic here
8    int k= 0, v = 0; String i = "";
9  String[] text = new String[10];
10 JTextArea output = new JTextArea(12, 20);
11 while (v < 1 || v > 10)
12 {
13 i = JOptionPane.showInputDialog(null, "How many strings are in the array (1 - 10)?");
14 v = Integer.parseInt(i);
15 }
16 output.append("Text arranged in alphabetical order\n");
17 output.append("=================================\n");
18 for (k = 0; k < v; k++)
19 text[k] = JOptionPane.showInputDialog(null, "Enter Text " + (k+1));
20  myString entered_text = new myString(text,v,'a'); //enter a for ascending order
21 text = entered_text.arrange();
22 for (k = 0; k < v; k++)
23 output.append((k+1)+"\t"+text[k]+"\n");
24  JOptionPane.showMessageDialog(null,output,"Text Output",1);
25 System.exit(0);
26 }}
27 class myString{
28 private String[] stringArray = new String[10];
29 private String ps = "";
30 private int count = 0;
31 private char order = 'n';
32 public myString(String[] s, int c, char o){
```

```
32 stringArray = s;
33 count = c;
34 order = o;
35 }
36 public String[] arrange(){
37 int i = 0;
38 if (order == 'a'){
39 do{
40 if ((stringArray[i].compareTo(stringArray[i+1])) > 0){
41 ps = stringArray[i].substring(0);
42 stringArray[i] = stringArray[i+1].substring(0);
43 stringArray[i+1] = ps.substring(0);
44 i = 0;
45 }
46 else i++;
47 }
48 while (i < (count-1));
49 }
50 else
51 do{
52 if ((stringArray[i].compareTo(stringArray[i+1])) < 0)
53 {
54 ps = stringArray[i].substring(0);
55 stringArray[i] = stringArray[i+1].substring(0);
56 stringArray[i+1] = ps.substring(0);
57 i = 0;
58 }
59 else i++;
60 } while (i < (count-1));
61  return stringArray;
62 }}
```

Figure 4.1: Program to arrange Strings in ascending or descending order

1. Type and Run the program code in Fig. 4.1
2. Write a program that will print "get a string from a user" and find out the number of times a given alphabet appears in that string. Output the string, the alphabet and the number of occurrences of the alphabet in the string.
3. Write a program that will compute the active power and the reactive power of a power transformer given its real power and power factor.
4. Implement a program that will compute the area of four different quadrilaterals: square, rectangle, parallelogram and trapezium. Use the interface to compute the area and let the different shapes inherit the quadrilateral class.

Exceptions and Multi-threading

```
1  import javax.swing.JOptionPane;
2  import javax.swing.JTextArea;
3  public class exceptionpractice {
4  public static void main(String[] args) {
5  // TODO code application logic here
6  int k= 0, c = 0;
7  String i = "";
8  String[] text = new String[10];
9  JTextArea output = new JTextArea(20, 30);
10 for(int d=5;d>0;d--){
11 try{ while (c < 1 || c > 10){
12 i = JOptionPane.showInputDialog(null, "How many names are in the array (1 - 10)?");
13 c = Integer.parseInt(i);}
14 } catch (NumberFormatException nfe){
15 JOptionPane.showMessageDialog(null, "Invalid Data Entered...
16 \n\n"+"you have"+" "+ d+" " +"trials left"+" "+nfe.getMessage());
17 } }
18 output.append("Number\tText\n");
19 for (k = 0; k < c; k++) {
20 text[k] = JOptionPane.showInputDialog(null, "Enter Text " + (k+1));
21 output.append((k+1)+"\t"+text[k]+"\n");
22 }
23 JOptionPane.showMessageDialog(null,output,"Text Output",1);
24 System.exit(0); }}
25
```

Figure 4.2: Program to Demonstrate Exception

1. Type and Run the program code in Fig. 4.2.
2. Write a two-threaded program that will allow for any two things to be done simultaneously within a program.

EXPERIMENT #: UILCPE/SDT 5

EXPERIMENT TITLE: Input/Output Streams in Java

OBJECTIVE:

- To introduce the concept of I/O streams in Java

BACKGROUND

Basically we use InputStream to read data from a source and the OutputStream for writing data to a destination. To create an input or output stream, an object of the stream class is created and associated with the source or destination of the data. A method for reading or writing data as defined in the stream class is then called to perform the input or output function. After making use of the stream object, the stream is closed using a close() method as defined in the stream class. Input and output statements are enclosed in a try ... catch block to handle the exceptions that may occur while working with streams and files. These exceptions are defined in the IOException class of the java.io package.

PREPARATION

This laboratory session is to give the students ample practice with the concepts taught in course materials on Java programming.

.EXPERIMENT

```
1  import java.io.*;
2  public class readfile {
3  public static void main(String[] args) {
4  // TODO code application logic here
5  int content = 0;
6  try{
7      FileInputStream mydata = new FileInputStream("\\Users\\Public\\contentfile.txt");
8  while (content != -1)
9  {
10  content = mydata.read();
11  System.out.print(content);
12 }
13 mydata.close();
14 }catch(FileNotFoundException ffe) {
15 System.out.println("Sorry..." + ffe.toString());
16 }
17 catch(IOException ioe) {
18 System.out.println("Unable to read file..."+ioe.getMessage());
19 }
20 }
21 }
```

Program 5.1: Program to demonstrate FileInputStream

```
import java.io.*;
import javax.swing.JOptionPane;
import javax.swing.JTextArea;
public class readfile1 {
public static void main(String[] args) {
// TODO code application logic here
int content = 0;
try{
    JTextArea output = new JTextArea(10, 20);
    FileInputStream mydata = new FileInputStream("\\Users\\Public\\contentfile.txt");
while (content != -1)
{
 content = mydata.read();
output.append(String.valueOf((char)content));
}
JOptionPane.showMessageDialog(null,output);
mydata.close();
}catch(FileNotFoundException ffe) {
System.out.println("Sorry..." + ffe.toString());
}
catch(IOException ioe) {
System.out.println("Unable to read file..."+ioe.getMessage());
}}}
```

Figure 5.2: Program to demonstrate FileInputstream

```
import java.io.*;
public class fileout {
public static void main(String[] args) {
// TODO code application logic here
int content = 0;
try{
FileInputStream mydatain = new FileInputStream("\\Users\\Public\\contentfile.txt");
FileOutputStream mydataout = new FileOutputStream("\\Users\\Public\\mycontentout.txt");
while (content != -1) {
content = mydatain.read();
mydataout.write((byte)content);
}
mydatain.close(); mydataout.close();
}catch(FileNotFoundException ffe) {
System.out.println("Sorry..." + ffe.toString());
}
catch(IOException ioe) { System.out.println("Unable to read file..."+ioe.getMessage());
}
}}
```

Figure 5.3: Program to demonstrate FileOutputStream

1. Type and run the program codes of Figures 5.1, 5.2 and 5.3. State your observations.
2. Create a folder name lab381
3. Create a file containing an array of 1000 randomly generated real numbers named random.txt
4. Duplicate the content of the file into another file named randcopy.txt
5. Rename randcopy.txt (name it copyrand.txt)
6. Delete randcopy.txt
7. Retrieve the content of the file copyrand.txt into an integer array after conversion of each of the numbers to integer
8. Display all the 1000 integers in a text area on a message box.

EXPERIMENT #: UILCPE/SDT 6

EXPERIMENT TITLE: Graphical User Interface

OBJECTIVE:

- To introduce the students to GUI development and building applets

BACKGROUND

Graphic User Interface (GUI) programs differ from normal ways of displaying results because they event driven .What generate the event is user actions such as clicking on a button ,the program then respond to these event as they occur. GUI is of two types namely: Stand-alone application and applet. The formal is a program that runs on its own, without depending on a Web browser while the latter runs in a rectangle area of a webpage to do simple task .A GUI program offers a much richer type of user interface, where the user uses a mouse and keyboard to interact with GUI components such as windows, menus, buttons, check boxes, text input boxes, scroll bars, combo boxes text Area.

PREPARATION

This laboratory session is to give the students ample practice with the concepts taught in course materials on Java programming

EXPERIMENT

Using Jframe in your Netbeans, design the GUI in Figure 6.1 and add the program code in Figure 6.2.

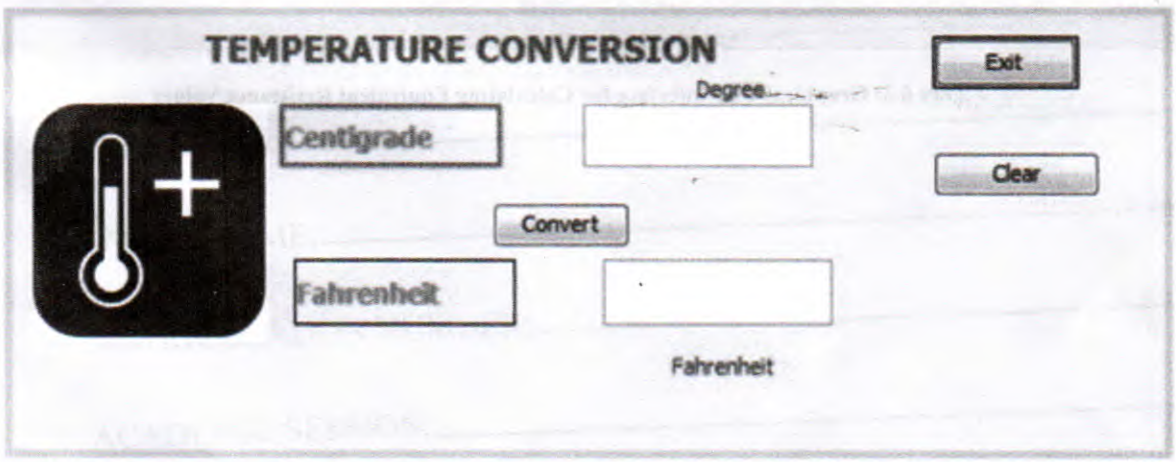


Figure 6.1: Graphical User Interface for Temperature Conversion

```
1  private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
2  double centigrade = Double.parseDouble(jTextField_centigrade.getText());
3  double fahrenheit = ((9*centigrade)/5)+32;
4  jTextField_fahrenheit.setText(String.valueOf(fahrenheit));}
5
6  private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
7  jTextField_fahrenheit.setText("");
8  jTextField_centigrade.setText("");
9  }
10 private void jButton3ActionPerformed(java.awt.event.ActionEvent evt) {
11 System.exit(0);
12     }
```

Figure 6.2: Graphical User interface for Temperature Conversion program code

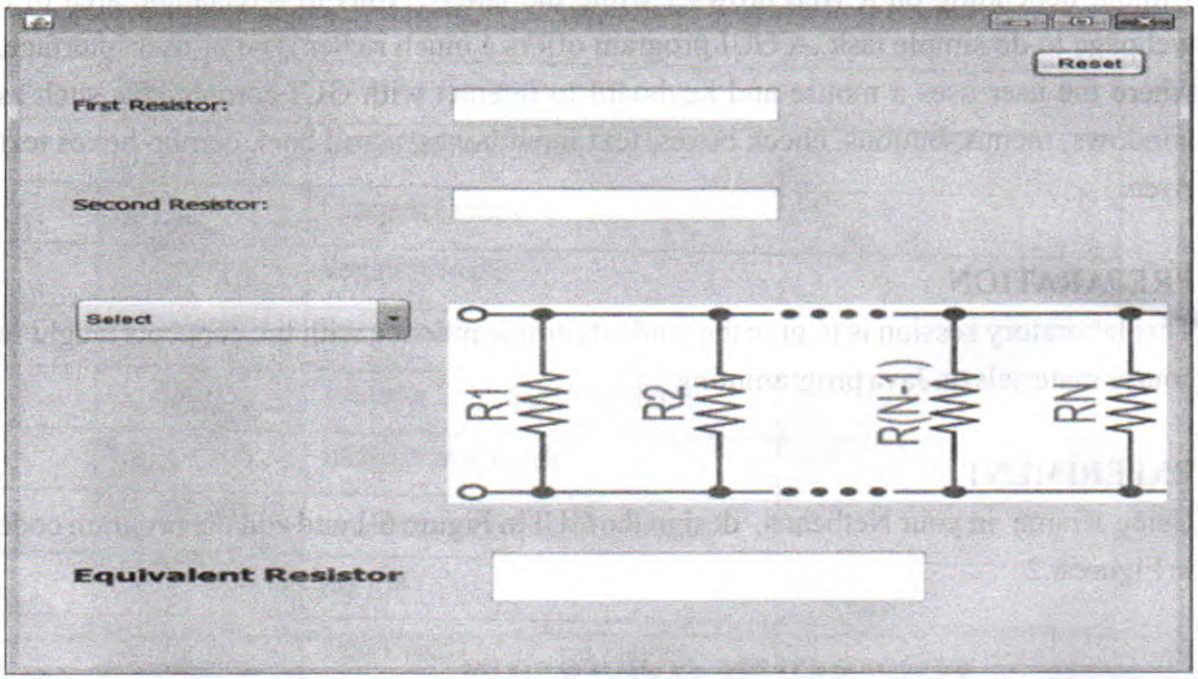


Figure 6.3: Graphical User interface for Calculating Equivalent Resistance Values

```
1   private void equalActionPerformed(java.awt.event.ActionEvent evt) {
2       String collect = (String)equal.getSelectedItem();
3       String parallel ="Parallel";
4       String Series ="Series";
5
6       double convertR1 = Double.parseDouble(R1.getText());
7       double convertR2 = Double.parseDouble(R2.getText());
8       double result;
9
10      if (( collect.equals(parallel)))  {
11         double  firstresult = (1/convertR1) + (1/convertR2);
12         result =(1/firstresult);
13          Display.setText(Double.toString(result));
14  }
15  else{
16        if (( collect.equals(Series))){
17
18              double  firstresult = convertR1 + convertR2;
19              Display.setText(Double.toString(firstresult));
20         }
21
22      };
```

Figure 6.4: Program Code for Calculating the Equivalent Resistance Values

```
1     private void resetbtnActionPerformed(java.awt.event.ActionEvent evt) {
2  R1.setText(null);
3  R2.setText(null);
4  Display.setText(null);
5
6     }
```

Figure 6.5: Program Code for Resetting the Equivalent Resistance Values

1. Write your own Java applet to allow the user to arrange a set of names stored in a text file and print out the names arranged in alphabetical order on a web page.
2. Design a simple frame that contains the following components
   i. 4 labels (Matric Number, Surname, First name, CGPA)

APPENDIX I: STANDARD RESISTOR COLOR CODE

The color code for the resistor value utilizes two digits and a multiplier digit in that order, as shown in Figure A1. The fourth band designates the tolerance.

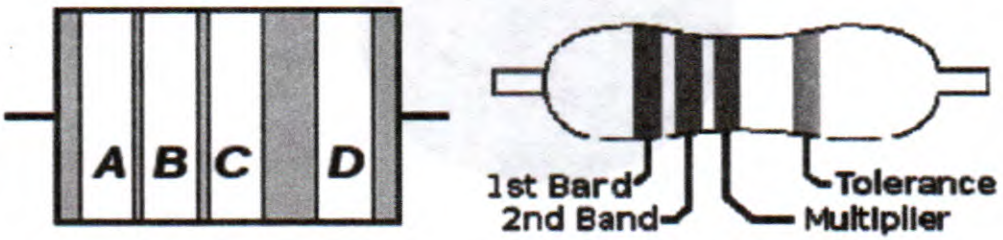


Figure A1: Resistor with four color bands.

The resistance of a resistor with the four bands of color may be written as:

$$R = (A \times 10 + B) \times C \pm D$$

Where A and B are the values of the first and second bands, respectively, C is a multiplier and D is the tolerance. The color code together with multipliers and tolerances are listed in Table A1.

Table A1: Resistor Color Code

Color	Significant figures	Multiplier	Tolerance
Black	0	$\times 10^0$	--
Brown	1	$\times 10^1$	--
Red	2	$\times 10^2$	±2%
Orange	3	$\times 10^3$	--
Yellow	4	$\times 10^4$	--
Green	5	$\times 10^5$	--
Blue	6	$\times 10^6$	--
Violet	7	$\times 10^7$	--
Gray	8	$\times 10^8$	--
White	9	--	--
Gold	--	$\times 10^{-1}$	±5%
Silver	--	$\times 10^{-2}$	±10%
None	--	--	±20%

ii. 4 fields (for each of the labels)
iii. 4 buttons (Add, Delete, Update, Cancel)

3. Build a simple applet that will allow you to write text that you read from a text file in a web browser.

4. The GUI below was designed using Microsoft Visual BASIC. Build a similar GUI using the NetBeans IDE.

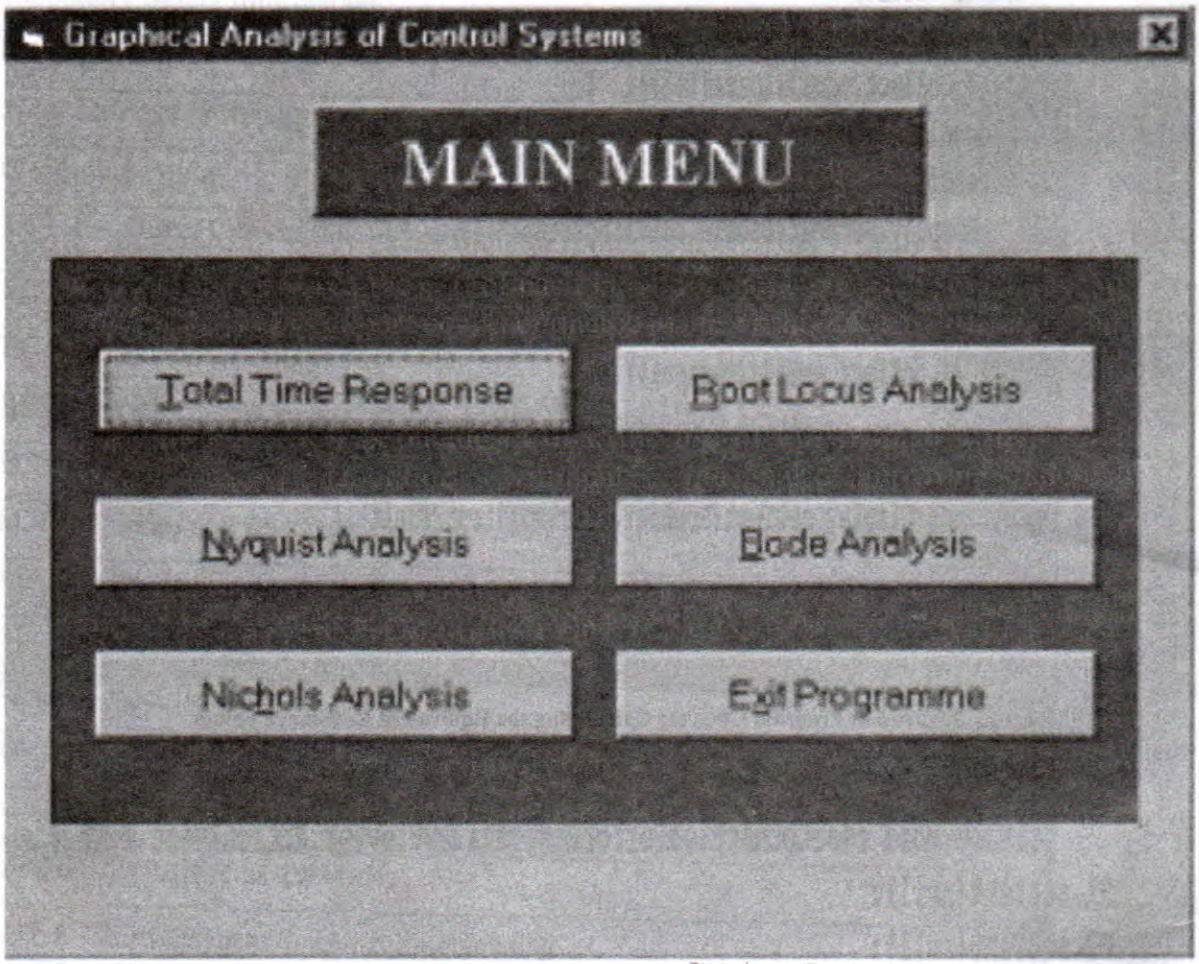
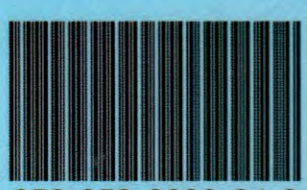


Figure 6.6: GUI Analysis of Control Systems


978-978-8090-84-2