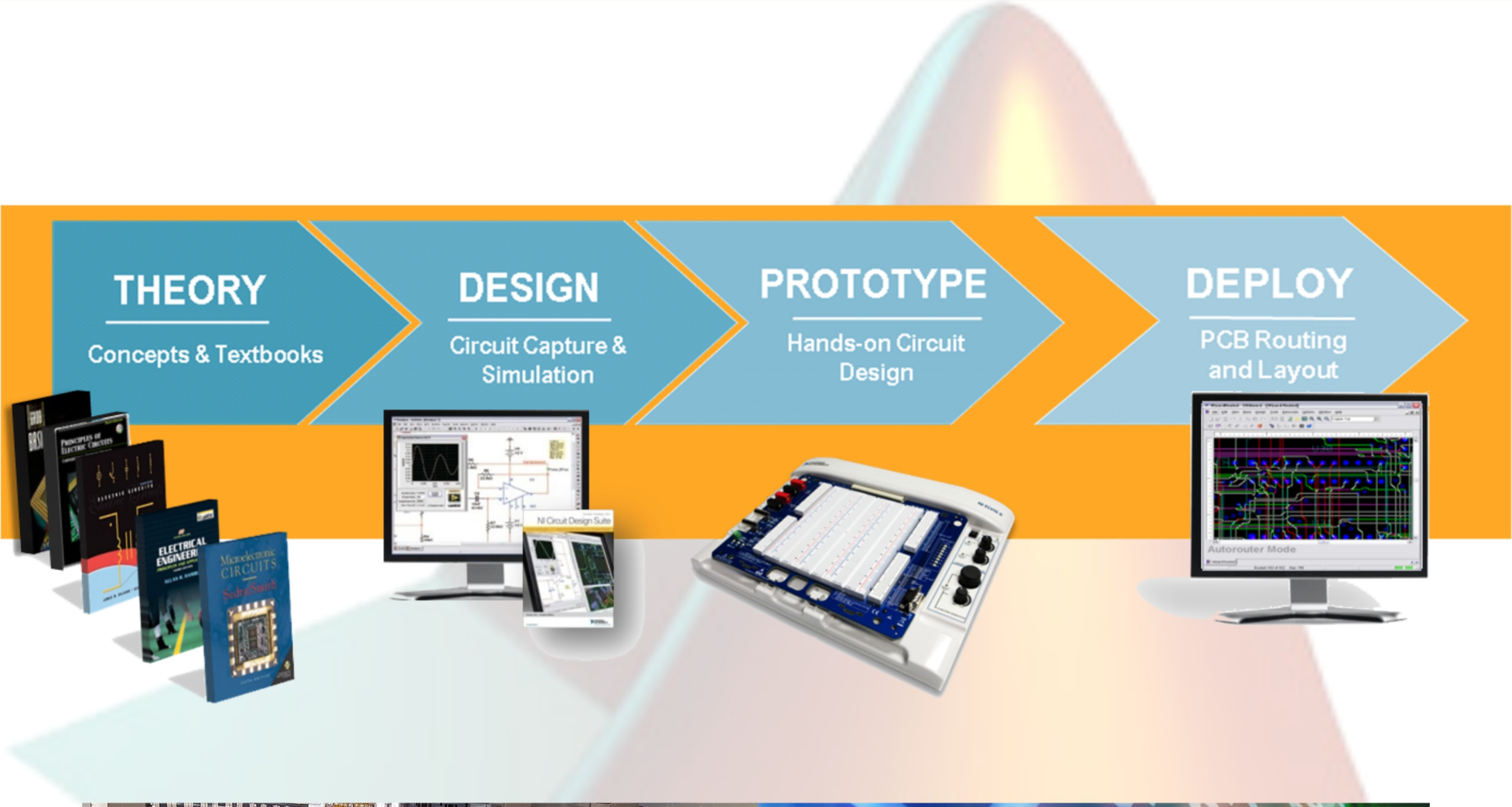
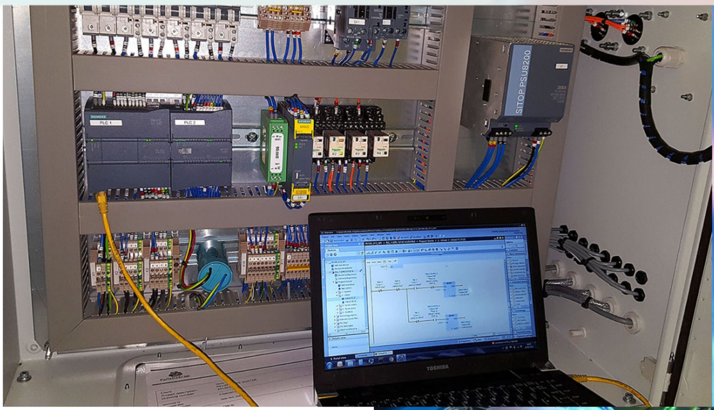
APPLICATION OF COMPUTER IN ELECTRICAL TECHNOLOGY (ET-251)

Lab Manual





Authors

Engr. M. Abbas Khan Abbasi **Engr. M. Sheraz Ahmad**

Lab Manual

Course Title

APPLICATIONS OF COMPUTER

IN

ELECTRICAL TECHNOLOGY

Course Code

(ET 251)

Prepared By:

Engr. Muahammad Abbas Khan Abbasi
Assistant Professor (Computer Engineering)

Engr. Muhammad Sheraz Ahmad
Lecturer (Computer Engineering)

Technical Education & Vocational Training Authority Khyber Pakhtunkhwa (KP-TEVTA),

Pawshawar Pakistan

Department of Electrical Technology

Rs. 160/=

Student Name: ______________________________

Class: ______________________________

Class Roll No: ______________________________

BTE Roll No: ______________________________

Teacher Name: ______________________________

DEDICATION

This manual is dedicated to our dearest Parents,

Families, respected work colleagues and Teachers

Who Motivate, Support and Encourage us

In every aspect of our life

ABOUT THE AUTHORS

Engr. Muhammad Sheraz Ahmad and Engr. Muhammad Abbas khan Abbasi belongs to province Khyber Pakhtunkhwa Hazara region. They obtained BS. Engineering degree in Computers from COMSATS Institute of Information Technology Abbottabad. They are Computer Engineers by Education started professional career in KP-TEVTA as Lecturer Computer Engineering and Aims to write their first manual keeping in view the need of DAE Students studying in Technical Education Institution of Pakistan.

They are teaching subjects related to electrical & computer engineering in Government College of Technology Abbottabad.

ACKNOWLEDGMENT

Allah is very kind, merciful and compassionate. His benevolence and blessings enables us to accomplish this task.

We are thankful to our seniors **Professor Dr. Hasham Khan** and **Professor Engr. Fazl-e-Azeem Ullah KP Technical Education and Vocational Training Authority,** without their enormous help, guidance and encouragement, this manual could not have been a success. We also extend our vote of thanks to them for being a source of inspiration always.

We are deeply thankful to our work colleagues and friend **Engr. Mohsin Akram**, who helped us a lot with his excellent technical skills.

And last but not least, we are thankful to the KP-TEVTA which give us a platform to work in an excellent environment with all possible modern amenities.

PREFACE

This manual contains practicals in accordance to curriculum of KP-BTE for the Subject **ET-251**: Application of Computer in Electrical Technology.

- First Part of Manual Contains Programs in Basic Language to solve electrical equation with practice exercises.
- 2nd Part of Manual Contains PSPICE and Multisim Practical.
- 3rd Part of Manual Contains Practical Related to MTS 8086 Trainer in Assembly language.
- 4th Part of Manual Contains Introduction to PLC.

RECOMENDATION

This practical Lab manual is highly appreciated and recommended by the following officers, who are working under Khyber Pakhtunkhwa Technical Education & Education & Vocational Training Authority in different Polytechnic Institutes and College of technologies.

1. ***Engr. Hafeez Ur Rehman***
   Assistant Professor (Computer Engineering)
   Government College of Technology, Peshawar.
2. ***Engr. Amjid Ali***
   Assistant Professor (Computer Engineering)
   Government College of Technology, Swat.
3. ***Engr. Faisal Jamal Nasir***
   Assistant Professor (Computer Engineering)
   Government Polytechnic Institute, Bunair.
4. ***Engr. Syed Ijlal Hussain Shah***
   Assistant Professor (Computer Engineering)
   Government College of Technology, Abbottabad.
5. ***Engr. Syed Shadab Ali Shah***
   Assistant Professor (Computer Engineering)
   Government Polytechnic Institute, Karak.
6. ***Engr. Hikmat Ullah Khan***
   Assistant Professor (Computer Engineering)
   Government College of Technology, Bannu.
7. ***Engr. Iqbal Munir***
   Assistant Professor (Computer Engineering)
   Government College of Technology, Swat.
8. ***Engr. Fazal Rabi***
   Assistant Professor (Computer Engineering)
   Government College of Technology, Swat
9. ***Engr. Safi Jan***
   Assistant Professor (Computer Engineering)
   Government College of Technology, Nowshera
10. ***Engr. Faizan Ali***
    Lecturer (Electrical Engineering)
    Government College of Technology, Abbottabad.
11. ***Engr. Mohsin Akram***
    Lecturer (Electrical Engineering)
    Government College of Technology, Abbottabad.
12. ***Engr. Zeeshan Lodhi***
    Lecturer (Electrical Engineering)
    Government Polytechnic Institute, Mansehra.
13. ***Mr. Waseem Ullah Khan***
    Lecturer (Computer)
    Government College of Technology, Kohat.

List of Practical's

PRACTICAL NO. 01

GW-BASIC PROGRAM FOR OHM'S LAW

Objective:

- To write a GW-BASIC program which verify ohm's law of electricity (V=IR)
- Find the value of V when I & R are known.

PROGRAM:

```
AUTO
10      CLS
20      REM Program for Ohm's Law
30      INPUT "Enter the value of Current (I) ="; I
40      INPUT "Enter the value of Resistance (R) ="; R
50      LET V=I*R
60      PRINT "The value of Voltage (V) ="; V;"V"
70      END
```

RESULTS

PRACTICE PROGRAMS

1) Write a GW-BASIC program which find the value of I when V & R are known.
2) Write a GW-BASIC program which find the value of R when I & V are known.

RESULTS

PRACTICAL NO. 02

PARALLEL COMBINATION OF RESISTORS

Objective:

- Write a BASIC program which calculate equivalent resistance (R_{eq}) of parallel combination circuit shown in Figure 1.

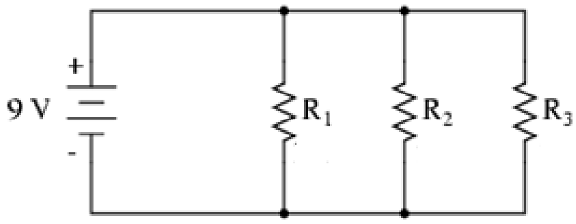


Figure 1: Parallel combination resistive circuit

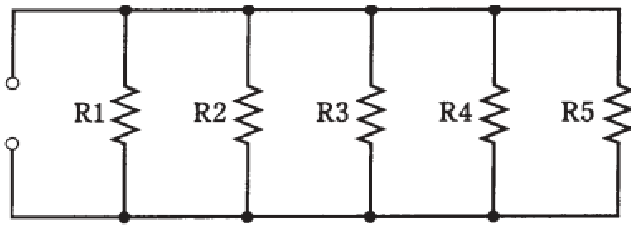
PROGRAM:

```
AUTO
10      CLS
20      REM Program for Parallel Combination Circuit
30      INPUT "Enter the value of resistor: R1 ="; R1
40      INPUT "Enter the value of resistor: R2 ="; R2
50      INPUT "Enter the value of resistor: R3 ="; R3
60      LET Req1=(R1+R2)/(R1*R2)
70      LET Req= (Req1*R3)/(Req1+R3)
80      PRINT "Equivalent Resistance of circuit Req ="; Req;"ohms"
90      END
```

<u>RESULTS</u>

PRACTICE PROGRAM

- Write a GW-BASIC program which find the equivalent resistance of the circuit shown below.



RESULTS

PRACTICAL NO. 03

SERIES COMBINATION OF RESISTORS

Objective:

- Write a BASIC program which calculate equivalent resistance (R_{eq}) of Series combination circuit shown in Figure 3.1.

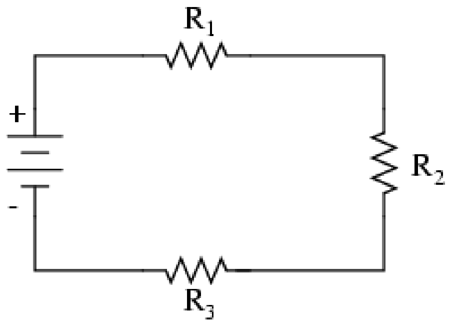


Figure 3.1: Series combination resistive circuit

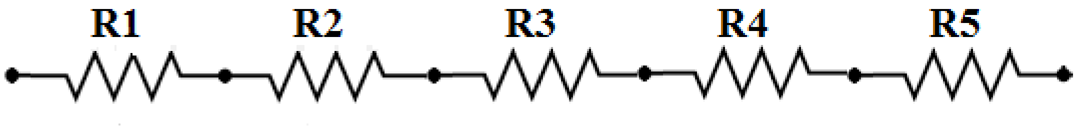
PROGRAM:

```
AUTO
10      CLS
20       REM Program of Series Combination Circuit
30      INPUT "Enter the value of resistor R1 ="; R1
40      INPUT "Enter the value of resistor R2 ="; R2
50      INPUT "Enter the value of resistor R3 ="; R3
60      LET Req= R1+R2+R3
70      PRINT "Equivalent Resistance of Circuit Req="; Req;"ohms"
80      END
```

<u>RESULTS</u>

PRACTICE PROGRAM

- Write a GW-BASIC program which find the equivalent resistance of the circuit shown below.



RESULTS

PRACTICAL NO. 04

PARALLEL COMBINATION OF CAPACITORS

Objective:

- Write a BASIC program which calculate equivalent Capacitance (C_{eq}) of parallel combination circuit shown in Figure 4.1.

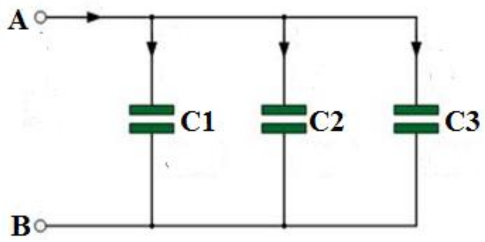


Figure 4.1: Parallel Combination Circuit

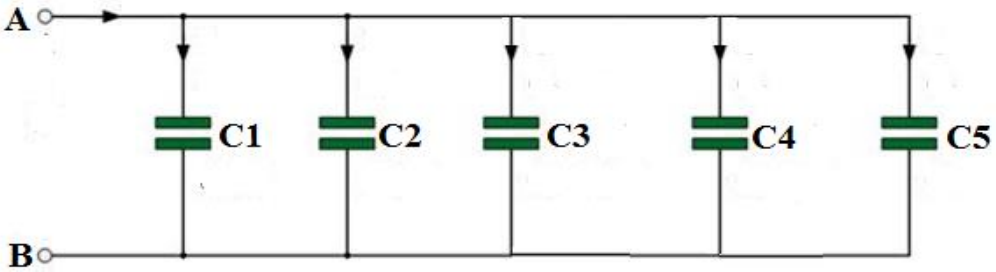
PROGRAM:

```
AUTO
10      CLS
20       REM Program of Parallel Combination of Capacitor Circuit
30      INPUT "Enter the value of Capacitor C1 ="; C1
40      INPUT "Enter the value of Capacitor C2 ="; C2
50      INPUT "Enter the value of Capacitor C3 ="; C3
60      LET Ceq= C1+C2+C3
70      PRINT "Equivalent Capacitance of circuit Ceq ="; Ceq;"F"
80      END
```

RESULTS

PRACTICE PROGRAM

- Write BASIC program to compute equivalent Capacitance of the circuit shown below.



RESULTS

PRACTICAL NO. 05

SERIES COMBINATION OF CAPACITORS

Objective:

- Write a BASIC program which compute equivalent Capacitance (C_{eq}) of Series combination circuit shown in Figure 5.1.

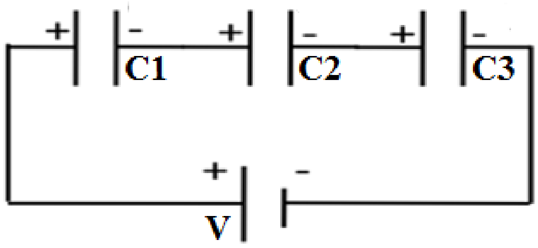


Figure 5.1: SeriesCombination Capacitor Circuit

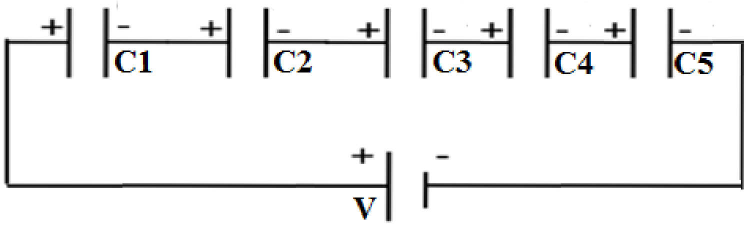
PROGRAM:

```
AUTO
10      CLS
20      REM Program of Series Combination of Capacitor Circuit
30      INPUT "Enter the value of Capacitor C1 ="; C1
40      INPUT "Enter the value of Capacitor C2 ="; C2
50      INPUT "Enter the value of Capacitor C3 ="; C3
60      LET Ceq= ((C1*C2*C3)/(C1*C2+C1*C3+C2*C3))
70      PRINT "Equivalent Capacitance of circuit Ceq ="; Ceq;"F"
80      END
```

<u>RESULTS</u>

PRACTICE PROGRAM

- Write BASIC program to compute equivalent Capacitance of the circuit shown below.



<u>RESULTS</u>

PRACTICAL NO. 06

COMPUTATIONOF CURRENT VOLTAGE& POWER

Objective:

- Write a BASIC program which compute total Current "I", Voltage "V" across each resistor & Power dissipation "P" of Series/Parallel resistor network Figure 6.1.

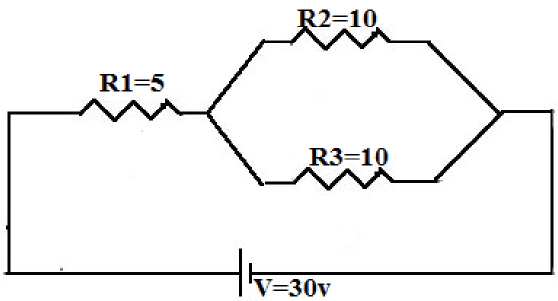


Figure 6.1: Series Parallel resistor network

Formulas

$$I = \frac{V}{Req} \quad Vn = IRn$$

$$P = I_T V$$

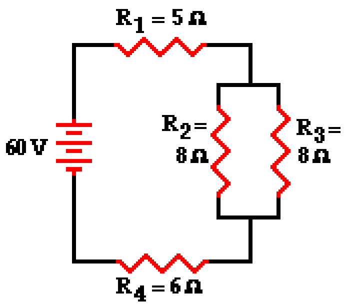
PROGRAM:

```
AUTO
10      CLS
20       REM Program to Compute I, V and P of series parallel resistor network
30      LET V=30
40      LET R1=5: R2=10: R3=10
50      LET Req1= (R2*R3)/(R2+R3)
60      LET Req=R1+Req1
60      LET I= V/Req
70      LET I1=I: I2= I1/2: I3= I1/2:
80      LET V1= I1*R1: V2= I2*R2: V3= I3*R3
90      LET P= I*V
100     PRINT "Total Current I ="; I;"mA"
110     PRINT "Voltage across resistor R1= V1="; V1;"v"
120     PRINT "Voltage across resistor R2= V2="; V2;"v"
130     PRINT "Voltage across resistor R3= V3="; V3;"v"
140     PRINT "Power dissipation P="; P;"Watt"
150     END
```

RESULTS

PRACTICE PROGRAM

- Write BASIC program which computetotal Current "I", Voltage "V"across each resistor & Power dissipation "P" of Series/Parallel resistor network Figure shown below.



Answer:

R_{tot} = 15 Ω	I_{tot} = 4 Amp
I_1 = 4 Amp	V_1 = 20 V
I_2 = 2 Amp	V_2 = 16 V
I_3 = 2 Amp	V_3 = 16 V
I_4 = 4 Amp	V_4 = 24 V

PRACTICE PROGRAM

RESULTS

PRACTICAL NO. 07

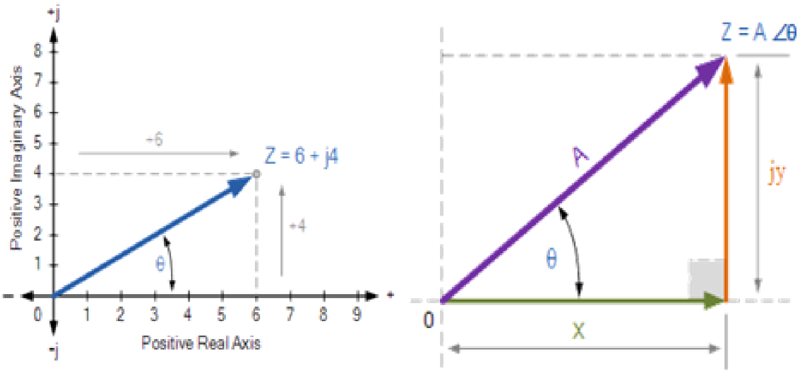
CONVERTION FROM RECTANGULAR TO POLAR FORM

Objective:

- Write a BASIC program which converts a Rectangular form into Polar form.

Theory:

$$\boldsymbol{Rectangular\ form}: x + jy \qquad \boldsymbol{Polar\ form}: A\angle\theta$$



$$\boldsymbol{Conversion\ from\ rectangular\ to\ polar\ form}$$

$$A = \sqrt{x^2 + y^2} \qquad \theta = \tan^{-1}(\frac{y}{x})$$

PROGRAM:

```
AUTO
10      CLS
20      INPUT "Enter the value of x ="; x
30      INPUT "Enter the value of y ="; y
40      LET A=SQR((x)^2+(y)^2)
50      LET Angle=ARCTAN(y/x)
60      PRINT "The value of amplitude A =", A
70      PRINT "The value of Angle (theta) =", Angle;"radians"
80      END
```

RESULTS

PRACTICE PROGRAM

- Write BASIC program which convert (5, 2) to polar coordinates.

RESULTS

PRACTICAL NO. 08

CONVERTION FROM POLAR TO RECTANGULAR FORM

Objective:

- Write a BASIC program which converts a Polar form into Rectangular form.

Theory:

The conversion from polar to rectangular form can be done using the following relations:

$$\boldsymbol{x} = A.\cos\theta \qquad \boldsymbol{y} = A.\sin\theta$$

$$\boldsymbol{Z} = x + jy$$

PROGRAM:

```
AUTO
10      CLS
20      REM Program to convert polar into rectangular form
30      INPUT "Enter the value of amplitude A ="; A
40      INPUT "Enter the value of Angle (theta) ="; Angle
50      LET x=A*COS(Angle)
60      LET y=A*SIN(Angle)
70      PRINT "The value of x-coordinate x ="; x
80      PRINT "The value of y-coordinate y ="; y
90      PRINT "The rectangular form is =";x, "+", y"j"
100     END
```

<u>RESULTS</u>

PRACTICE PROGRAM

- Write BASIC program which convert the polar coordinates (2, 53 ^{o}) to rectangular coordinates rectangular coordinates.

RESULTS

PRACTICAL NO. 09

BASIC OPERATIONS ON IMPEDANCE

Objective:

- Write a BASIC program which perform basic operations (i.e. addition, subtraction, division& multiplication) on Impedance in rectangular form.

Theory:

Addition

$$Z_1 + Z_2 = (x_1 + jy_1) + (x_2 + jy_2) = (x_1 + x_2) + j(y_1 + y_2)$$

Subtraction

$$Z_1 - Z_2 = (x_1 + jy_1) - (x_2 + jy_2) = (x_1 - x_2) + j(y_1 - y_2)$$

Multiplication

$$Z_1 \times Z_2 = (x_1 + jy_1) \times (x_2 + jy_2)$$

Division

$$\frac{Z_1}{Z_2} = \frac{(x_1 + jy_1)}{(x_2 + jy_2)}$$

$$multiply\ numinator\ \&\ denuminator$$

$$by\ conjugate\ (x_2 + jy_2)$$

PROGRAM:

```
AUTO
10      CLS
20      REM Addition of Impedance:
30      INPUT "Enter the value of real part of Z1="; Zr1
40      INPUT "Enter the value of real part of Z2="; Zr2
50      INPUT "Enter the value of Imaginary part of Z1="; Zi1
60      INPUT "Enter the value of Imaginary part of Z2="; Zi2
70      LET Zr=Zr1+Zr2
80      LET Zi=Zi1+Zi2
90      PRINT "The Result of Z1+Z2 =";Zr, "+", Zi"j"
100     END
```

RESULTS

PRACTICE PROGRAM

- Write BASIC program which subtract & multiply two impedances.

RESULTS

PRACTICAL NO. 10

IMPEDANCES OF SERIES RC CIRCUITS

Objective:

- Write a GW-BASIC Program which calculate impedance of series RC Circuit shown in figure 10.1.

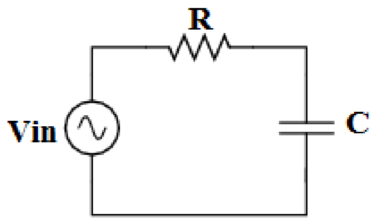


Figure 10.1: Series RC circuit

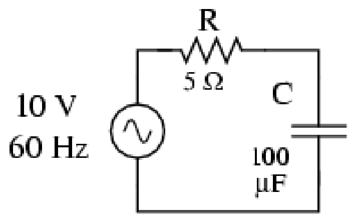
PROGRAM:

```
AUTO
10      CLS
20      REM Impedance of Series RC Circuit:
30      INPUT "Enter the value of resistance R=ZR ="; ZR
40      INPUT "Enter the value of capacitance C =”; C
50      INPUT "Enter the value of frequency f =”; f
60      LET w=2*3.14*f        ; w is angular frequency (omega)
70      LET Zc= 1/(w*C)
80      PRINT “The total Impedance of RC circuit = Z t=ZR+Zc=”; ZR, Zc”j”
90      END
```

RESULTS

PRACTICE PROGRAM

- Repeat above program for known values of circuit elements shown below (Hint: you don’t need to take input on run time).



RESULTS

PRACTICAL NO. 11

IMPEDANCES OF PARALLEL RC CIRCUITS

Objective:

- Write a GW-BASIC program which calculate impedance in polar form of parallel RC Circuit shown in figure 12.

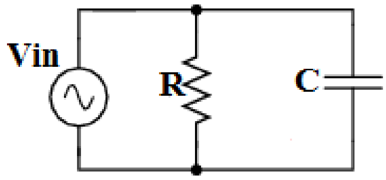


Figure 12: Parallel RC circuit

$$\frac{1}{Z_T} = \sqrt{\left(\frac{1}{R}\right)^2 + \left(\frac{1}{X_C}\right)^2}$$

$$\theta = \tan^{-1}\frac{X_C}{R}$$

PROGRAM:

```
AUTO
10    CLS
20    REM Impedance of Parallel RC Circuit:
30    INPUT "Enter the value of resistance R ="; R
40    INPUT "Enter the value of capacitance C ="; C
50    INPUT "Enter the value of frequency f ="; f
60    LET w = 2*3.14*f
70    LET Xc = 1/(w*C)
80    LET Z1 = SQR((1/R)^2+(1/Xc)^2)
90    LET Zt =1/Z1
100   LET Angle = ARCTAN(XC/R)
110   PRINT "The total Impedance of parallel RC circuit = Zt ="; Zt "ohms"
120   PRINT "Value of angle theta (in radians) Angle ="; Angle "radians"
130   END
```

<u>RESULTS</u>

PRACTICE PROGRAM

- Repeat the above program for R=5Ω, C=100μF and frequency *f*=60Hz.

RESULTS

PRACTICAL NO. 12

IMPEDANCE CALCULATION SERIES RLC CIRCUITS

Objective:

- Write a BASIC program which calculate the total Impedance (in polar form) of the series RLC circuit shown in figure 12.

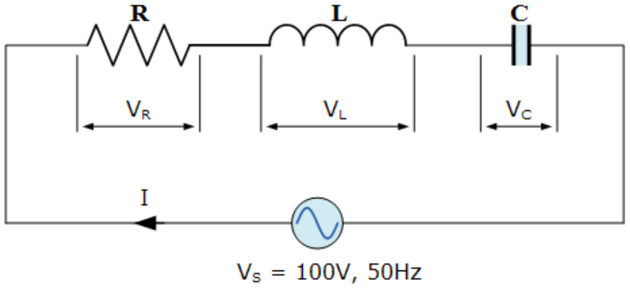


Figure 12: Series RLC circuit

Theory:

For series RLC circuits

$$Z_{eq} = Z_1 + Z_2 + Z_3 + \ldots\ldots\ldots\ldots + Z_n \quad or \quad Z_{eq} = \sqrt{R^2 + (X_L - X_C)^2}$$

$$\theta = \tan^{-1}\frac{X_L - X_C}{R}; \quad X_C = \frac{1}{2\pi f C}; \qquad X_L = 2\pi f L$$

PROGRAM:

```
AUTO
10    CLS
20    REM Program which compute total impedance of series RLC circuit
30    INPUT "Enter the value of Frequency f ="; f
40    INPUT "Enter the value of resistance R="; R
50    INPUT "Enter the value of Inductance L ="; L
60    INPUT "Enter the value of Capacitance C ="; C
70    LET XC=1/(2*3.14*f*C)
80    LET XL=2*3.14*f*L
90    LET Z=SQR(R^2+(XL-XC)^2)
100   LET Angle=ARCTAN((XL-XC)/R)
```

```
110    PRINT "Total impedance of Circuit Zeq ="; Z "ohms"
120    PRINT "Value of angle theta (in radians) Angle ="; Angle "radians"
130    END
```

RESULTS

PRACTICE PROGRAM

- Repeat the above program for R = 200Ω, X_C=150Ω, X_L= 80Ω and *f*= 60Hz.

RESULTS

PRACTICAL NO. 13

IMPEDANCE CALCULATION PARALLEL RLC CIRCUITS

Objective:

- Write a BASIC program which calculate equivalent Impedance (in polar form) of the parallel RLC circuit shown in figure 13.

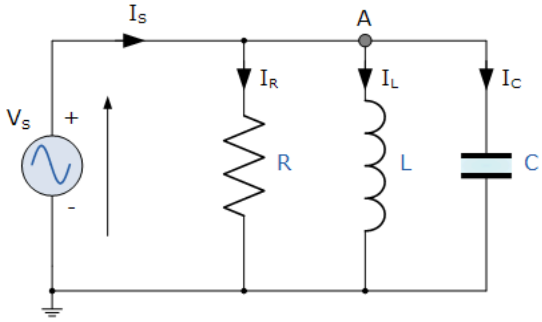


Figure 13: Parallel RLC circuit

Theory:

For Parallel RLC Circuits

$$\frac{1}{Z_{eq}} = \frac{1}{Z_1} + \frac{1}{Z_2} + \frac{1}{Z_3} + \ldots\ldots\ldots\ldots + \frac{1}{Z_n} \quad or \quad \frac{1}{Z_{eq}} = \sqrt{\left(\frac{1}{R}\right)^2 + \left(\frac{1}{X_L} - \frac{1}{X_C}\right)^2}$$

$$\theta = \tan^{-1}\frac{X_L - X_C}{R}; \qquad X_C = \frac{1}{2\pi f C}; \qquad X_L = 2\pi f L$$

PROGRAM:

```
AUTO

10      CLS

20      REM Program to compute total impedance of parallel RLC circuit

30      INPUT "Enter the value of Frequency f ="; f

40      INPUT "Enter the value of resistance R="; R

50      INPUT "Enter the value of Inductance L ="; L

60      INPUT "Enter the value of Capacitance C ="; C

70      LET XC=1/(2*3.14*f*C)

80      LET XL=2*3.14*f*L

90      LET Z1 = SQR((1/R)^2+((1/XL)-(1/XC))^2)

100     LET Zeq = 1/Z1
```

```
110     LET Angle=ARCTAN((XL-XC)/R)
120     PRINT "Total impedance of circuit Zeq ="; Zeq "ohms"
110     PRINT "Value of angle theta (in degree) Angle ="; Angle "degree"
120     END
```

RESULTS

PRACTICE PROGRAM

- Repeat the above program for R = 200Ω, X_C=150Ω, X_L= 80Ω and *f*= 60Hz.

RESULTS

PRACTICAL NO. 14

MAXIMUM AVERAGE POWER TRANSFER OF A.C CIRCUITS

Objective:

- Write a BASIC program which calculate maximum average power transfer of A.C circuits.

Theory:

The maximum power transfer of an AC circuit is calculated by the relation shown below:

$$P_{\max} = \frac{I_L{}^2 R_L}{2} = \frac{|V_{Th}|^2}{8R_{Th}}$$

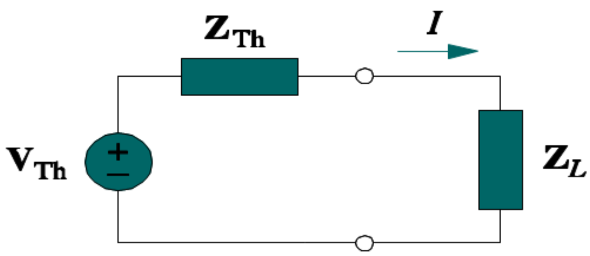
PROGRAM:

```
AUTO
10      CLS
20      REM Maximum Average Power Transfer
30      INPUT "Enter the value of Vth="; Vth
40      INPUT "Enter the value of Rth="; Rth
50      LET Pmax= (Vth^2)/(8*Rth)
60      PRINT “Maximum Average Power Transfer =”; Pmax;“Watt”
70      END
```

<u>RESULTS</u>

PRACTICE PROGRAM

- Repeat the above program for the circuit given below when I=5mA and Z_{th}=50Ω.



RESULTS

PRACTICAL NO. 15

APPARENT POWER & POWER FACTOR OF A.C CIRCUITS

Objective:

- Write a BASIC program which calculates Apparent Power & Power factor of A.C circuits.

Theory:

The apparent power (S) is the product of the rms values of voltage and current, and the power factor is the cosine of the phase difference between voltage and current. Apparent power and power factor of an AC circuits are calculated by the following mathematical relations:

$$\boldsymbol{S} = V_{rms} I_{rms}$$

$$\boldsymbol{pf} = Cos(\theta_v - \theta_i)$$

$$\text{For purely resistive circuits } \theta_v = \theta_i$$

$$So\ pf = Cos(\theta_v - \theta_i) = Cos(0) = 1$$

$$\text{And for purely rective load } \theta_v - \theta_i = \pm 90^0$$

$$So\ pf = 0$$

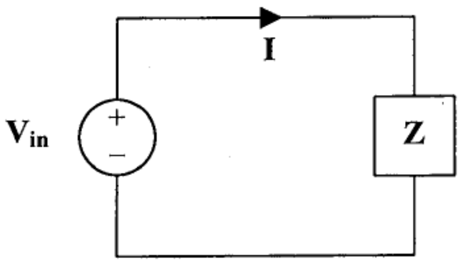
PROGRAM:

```
AUTO
10      CLS
20      REM Apparent Power & Power factor
30      INPUT "Enter the rms value of voltage Vrms ="; Vrms
40      INPUT "Enter the rms value of current Irms ="; Irms
50      INPUT "Enter the value of angle (in radian) Qv="; Qv
60      INPUT "Enter the value of angle (in radian) Qi="; Qi
70      LET S=Vrms*Irms
80      Q=Qv-Qi
90      LET pf=COS(Q)
100     PRINT "Apparent Power (S) ="; S "Watt"
110     PRINT "Power Factor = P.f ="; Pf
120     END
```

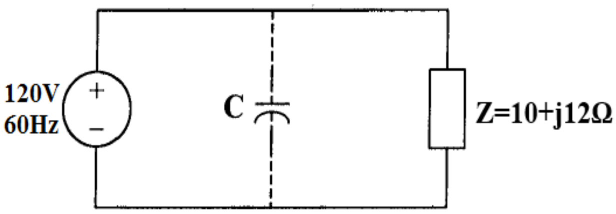
RESULTS

PRACTICE PROGRAMS

- Given the circuit below $V_{in} = 100\angle 30^0$ and $I = 10\angle 60^0$. Write BASIC program which calculate
  a) Apparent power, assuming V_{in} and I are *rms* values.
  b) Power factor



- Write BASIC program which calculate the power factor of circuit shown in Figure below (Hints: First calculate $\theta = \tan^{1}\frac{y}{x}$)



PRACTICE PROGRAM

RESULTS

PRACTICAL NO. 16

INTRODUCTION TO MULTISIM

Objective:

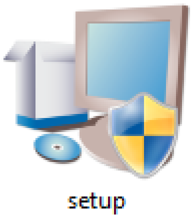
- Familiar with MULTISIM technical software& how it is use to design, simulate and analyze electrical networks.

Introduction

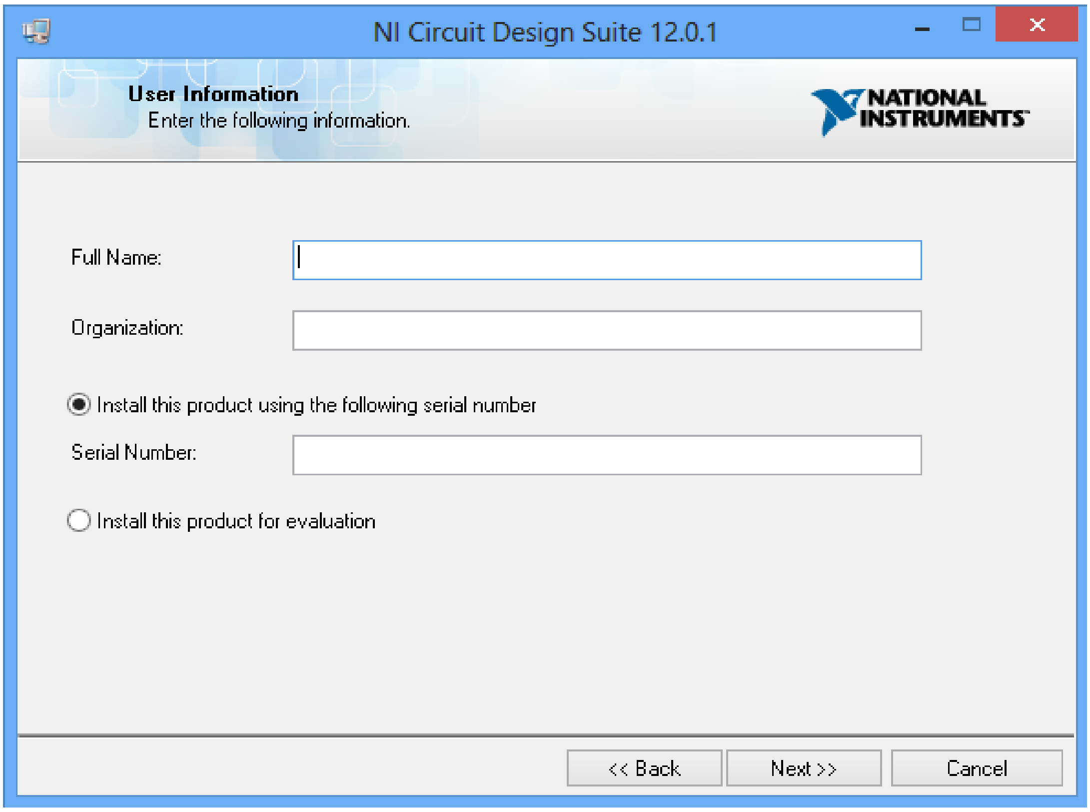
Multisim is a powerful schematic capture and simulation environment that engineers, students, and professors can use to simulate electronic circuits and prototype Printed Circuit Boards (PCBs).

Installation Steps:

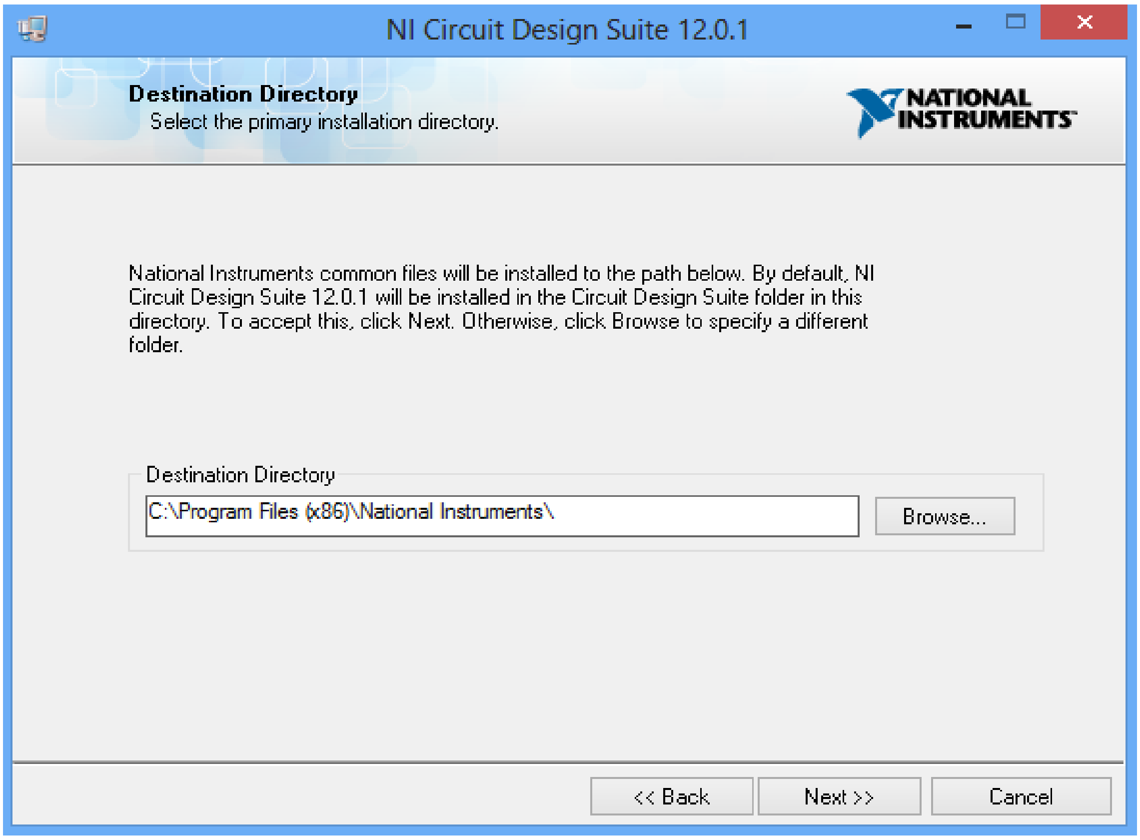
1. Double click on setup file as shown below:



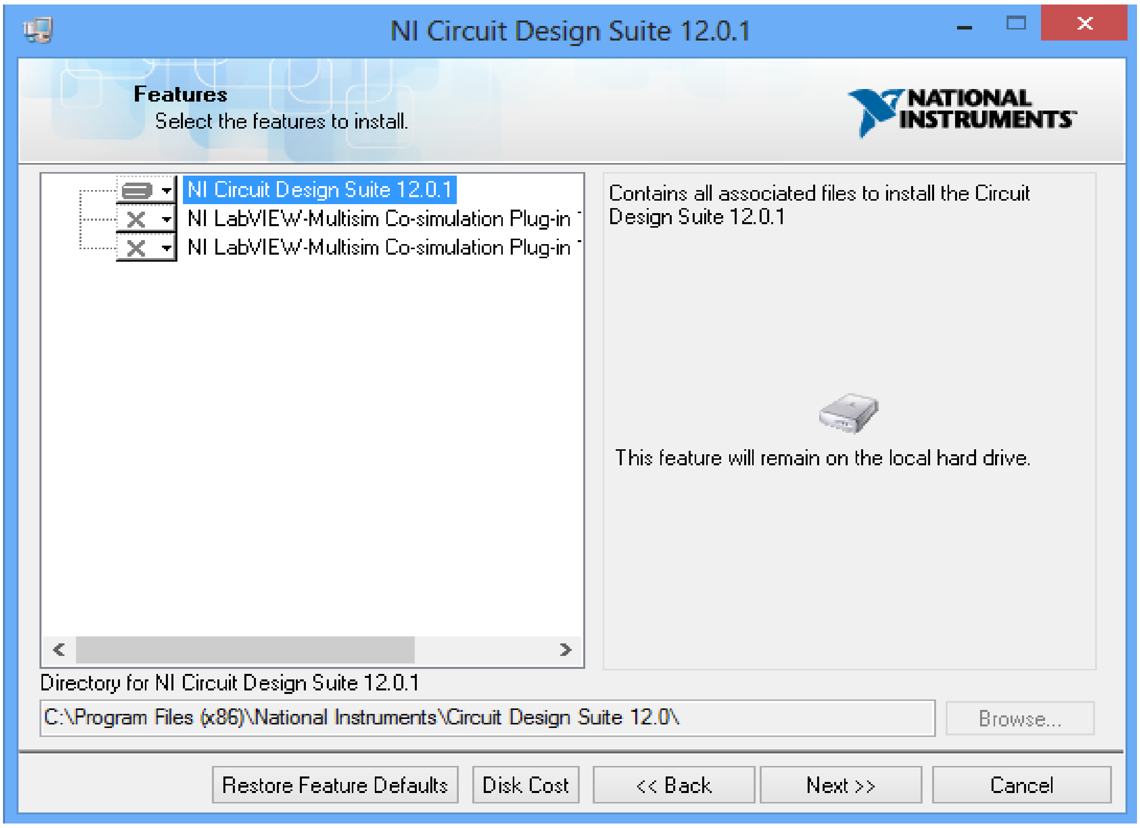
2. Fill up the table entries and check "***install this product using the following serial number***" if you've serial number otherwise check "***install this product for evaluation***" and click on **Next** button.



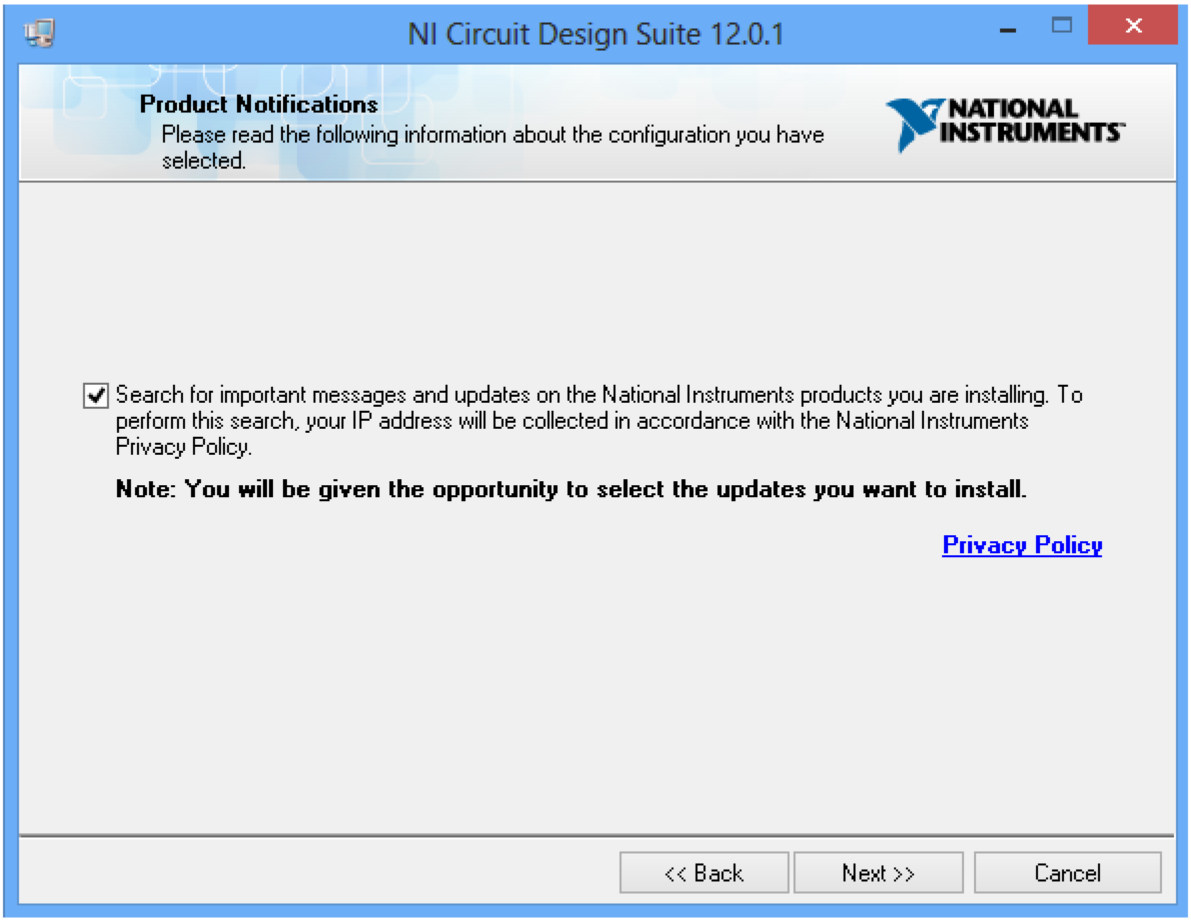
3. Click **<u>Browse…</u>** button and select the location in disk where to install software and click on **<u>Next</u>** button to proceed installation.



4. Select "***NI Circuit Design Suite 12.1***" from the left pan & click on **<u>Next</u>** button.



5. Click **Next** button of figure below and wait for completion of installation.



PRACTICAL NO. 17

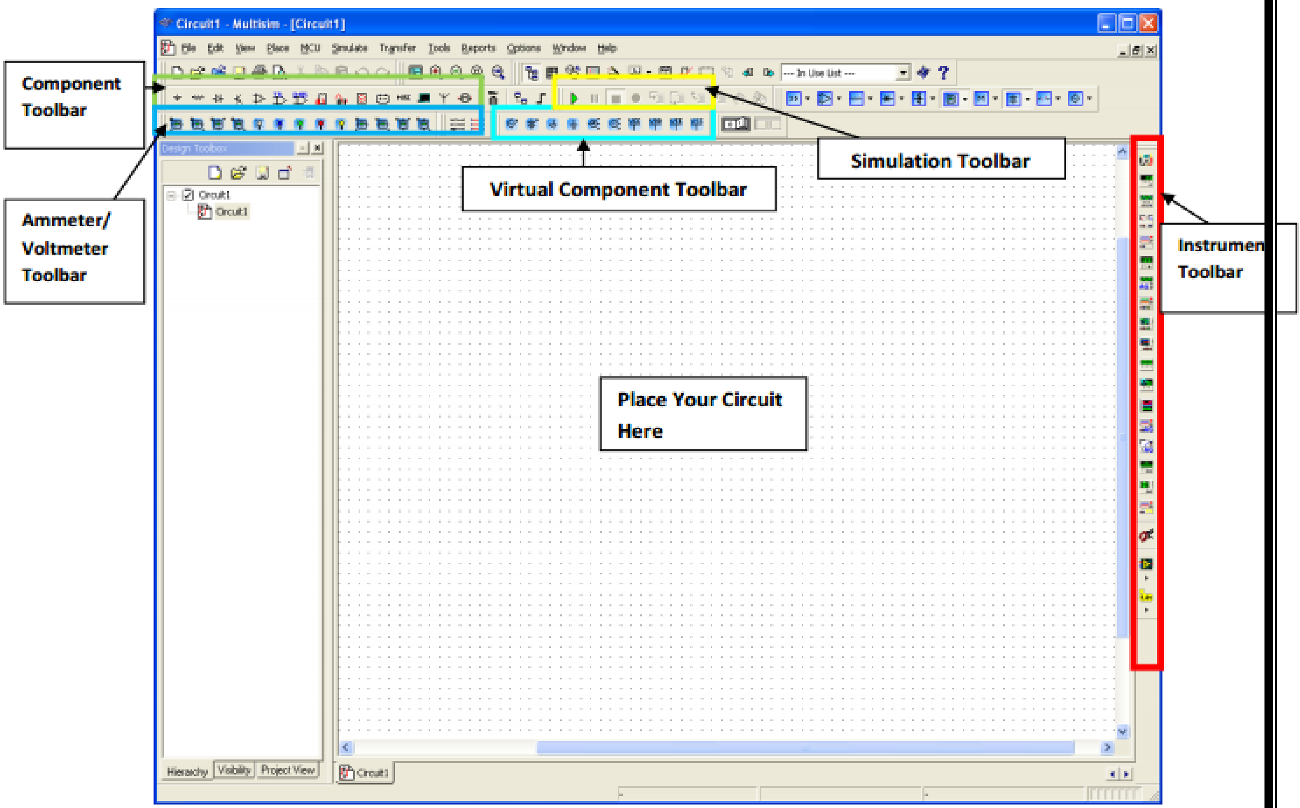
STARTINGTHE MULTISIM

Objective:

- Learn how to Start Multisim and understand the interface of Multisim.

Steps to Start:

i) Click on Start button located on taskbar
ii) Click on All Programs
iii) Click on National Instruments
iv) Click on Circuit Design Suite
v) Select Multisim an interface window shown below will be open.



❖ Open/Create Schematic

A blank schematic Circuit 1 is automatically created. To create a new schematic click on File – New – Schematic Capture. To save the schematic click on **File /Save As**. To open an existing file click on **File/ Open** in the toolbar.

❖ Place Components

To Place Components click on **Place/Components**. On the Select Component Window click on **Group** to select the components needed for the circuit. Click **OK** to place the component on the schematic.

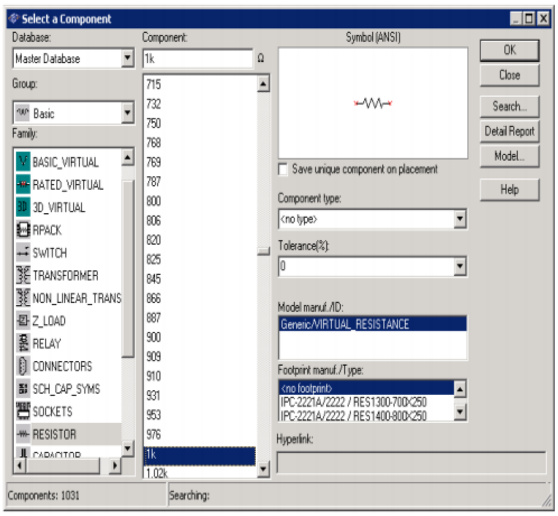


Figure 1: Select Resistor

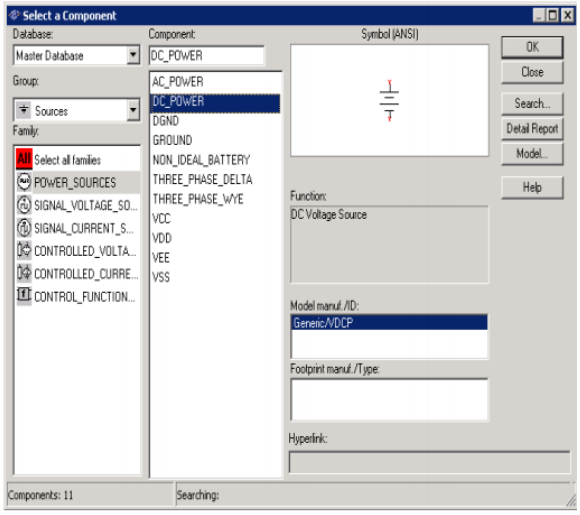


Figure 2: Select DC voltage

For example to select resistors and the DC source shown in Figure 3 click on **<u>Place/ Components.</u>** In **<u>Group</u>** select **<u>Basic</u>** scroll down to Resistors and select the value of the resistor needed to construct the circuit, for this example select 1k. To place DC source click on Sources in **<u>Group</u>** and select DC Source. As shown in Figure 1and Figure 2 respectively.

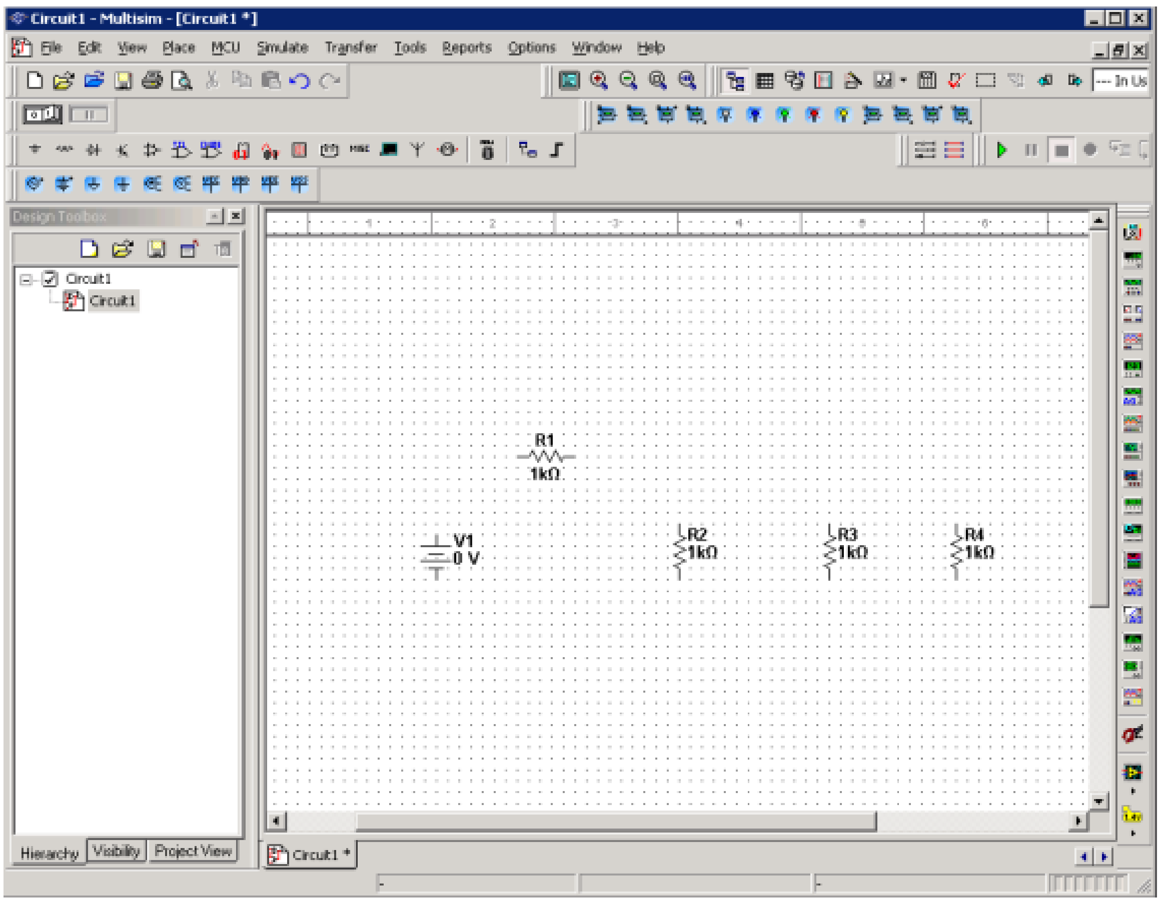


Figure 3: DC Source & Resistors

❖ Rotate Components

To rotate the components right click on the Resistor to flip the component on 90Clockwise (Ctrl +R) and 90 Counter Clockwise (Ctrl+Shift+R).

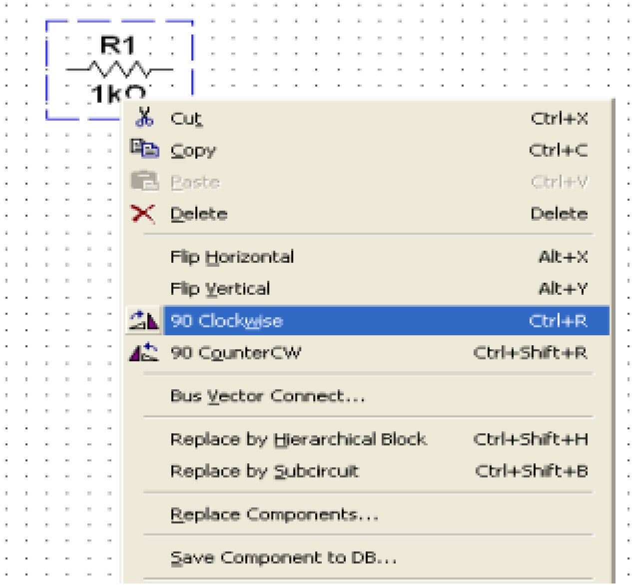


Figure 5: Rotate Components

❖ Place Wire/Connect Components

To connect resistors click on **<u>Place/Wire</u>** drag and place the wire. Components can also be connected by clicking the mouse over the terminal edge of one component and dragging to the edge of another component. Reference Figure 6.

❖ Change Component Values

To change component values double click on the component this brings up a window that display the properties of the component, reference Figure 7. ChangeR1 from 1k Ohm to 10 Ohms, R2 to 20 Ohms, R3 to 30 Ohms, and R4 to 40 Ohms. Also change the DV source from 0 V to 20 V. Figure 8 shows the completed circuit.

❖ Grounding

All circuits must be grounded before the circuit can be simulated. Click on Ground in the toolbar to ground the circuit. If the circuit is not grounded Multisim will not run the simulation.

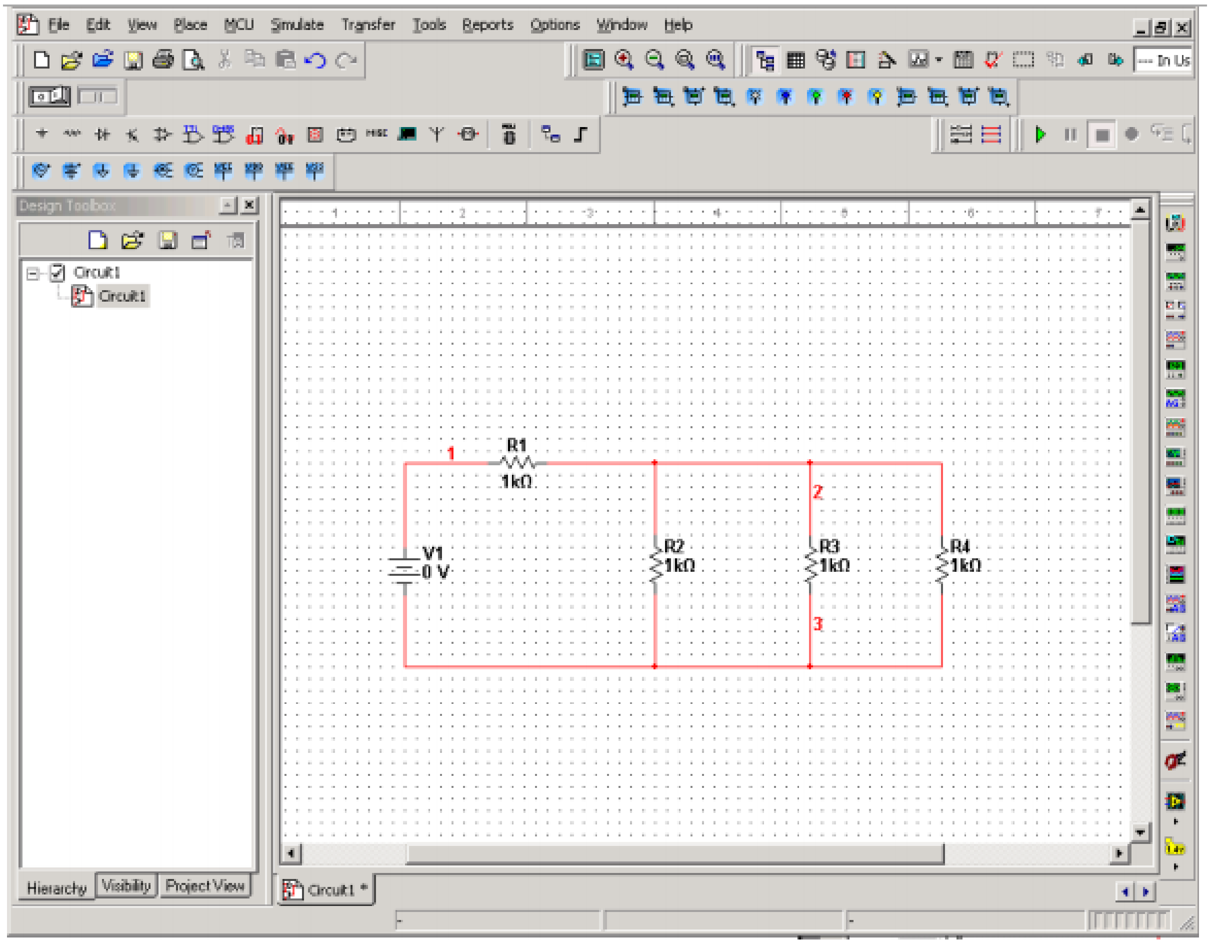


Figure 6: Place/ Wire

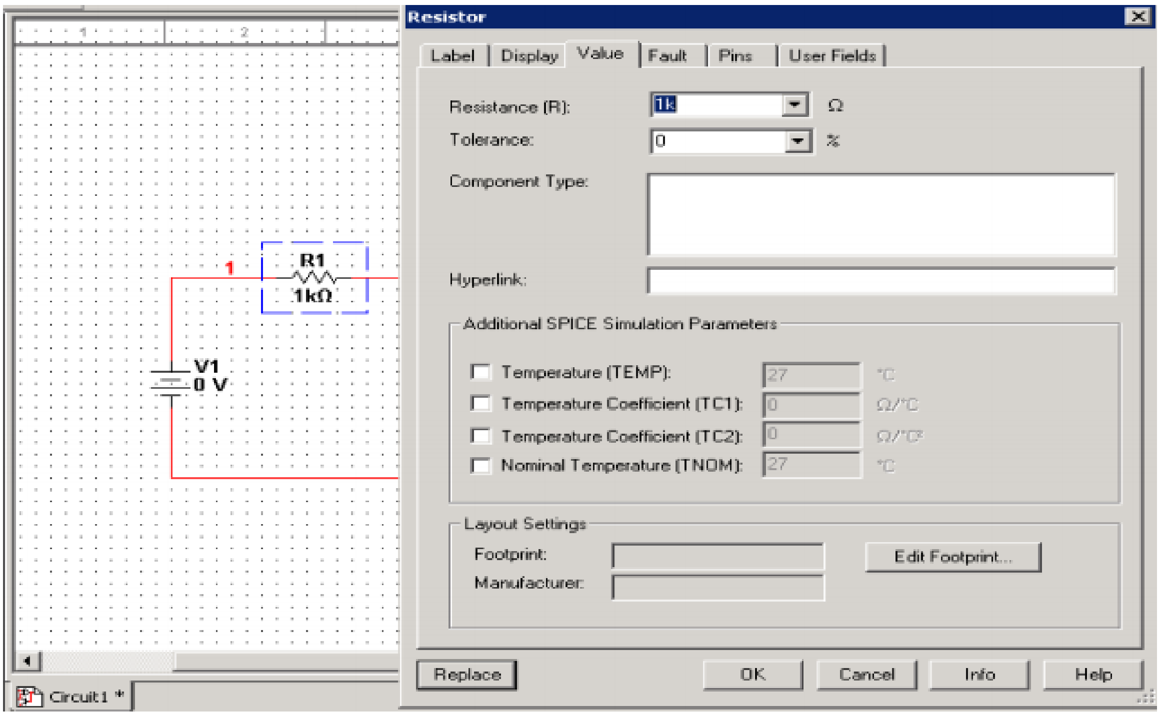


Figure 7: Change Component Values

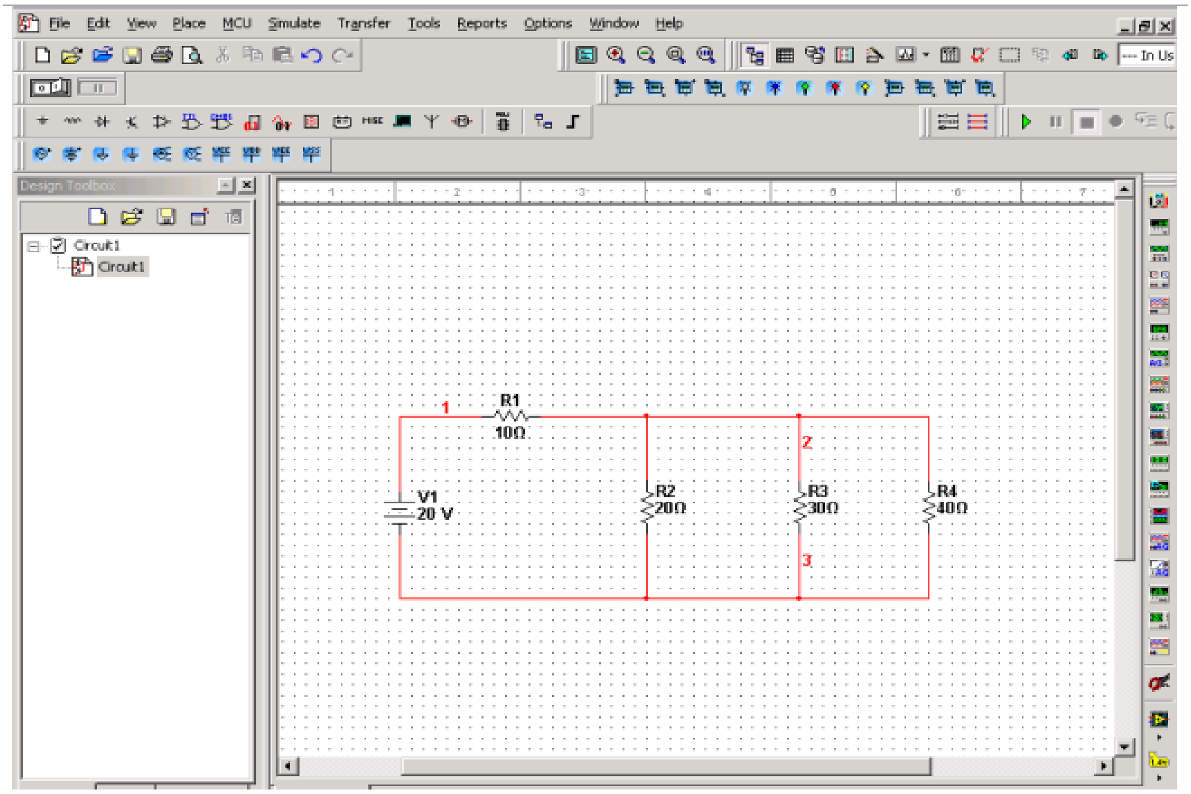


Figure 8: Completed Circuits

Figure 9: Grounding

❖ Simulation

To simulate the completed circuit Click on **Simulate/Run** or F5. This feature can also be accessed from the toolbar as shown in the Figure 10 below.

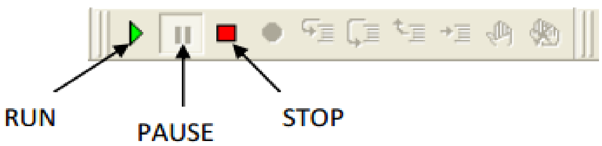


Figure 10: Simulation

❖ Analyzing Components

Multisim offers multiple ways to analyze the circuit using virtual instruments. Some of the basic instruments are:

a) Multimeter
b) Wattmeter
c) Agilent Multimeter
d) Ammeter
e) Voltmeter
f) Oscilloscope etc.

PRACTICAL NO. 18

DESIGN & SIMULATE HALF-WAVE RECTIFIER CIRCUIT

Objective:

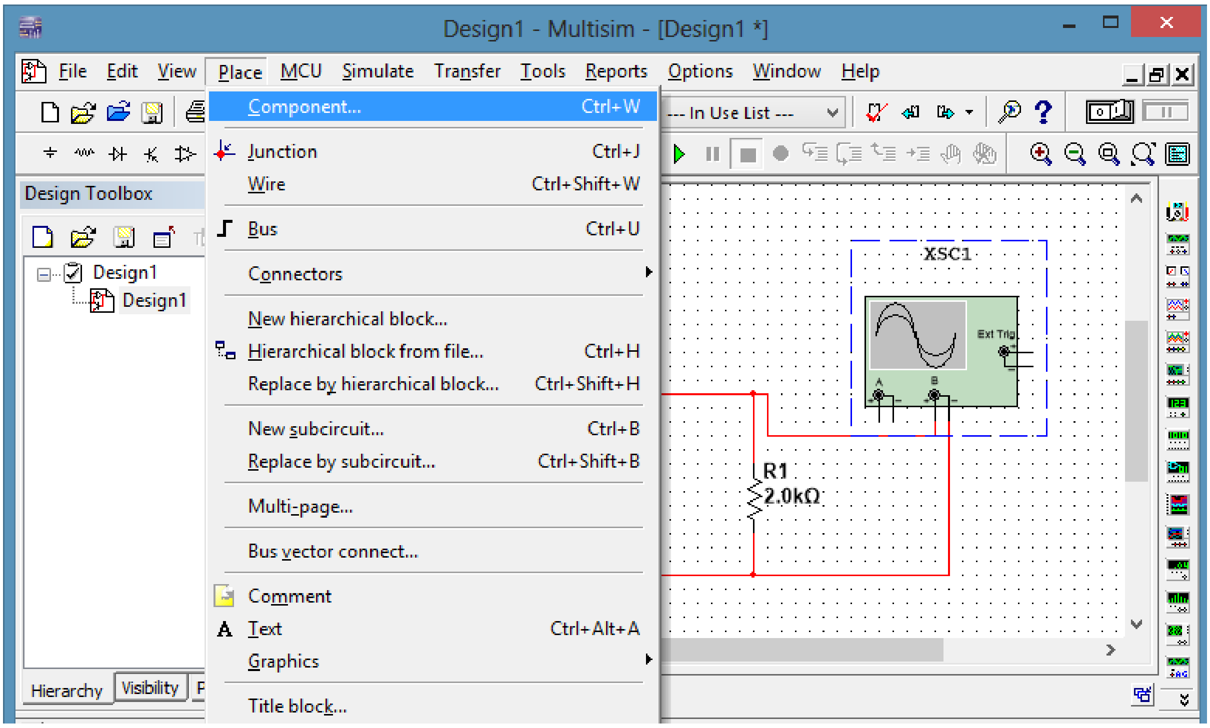
- To design and simulate the half wave rectifier circuit in MULTISIM environment.

Required Equipment:

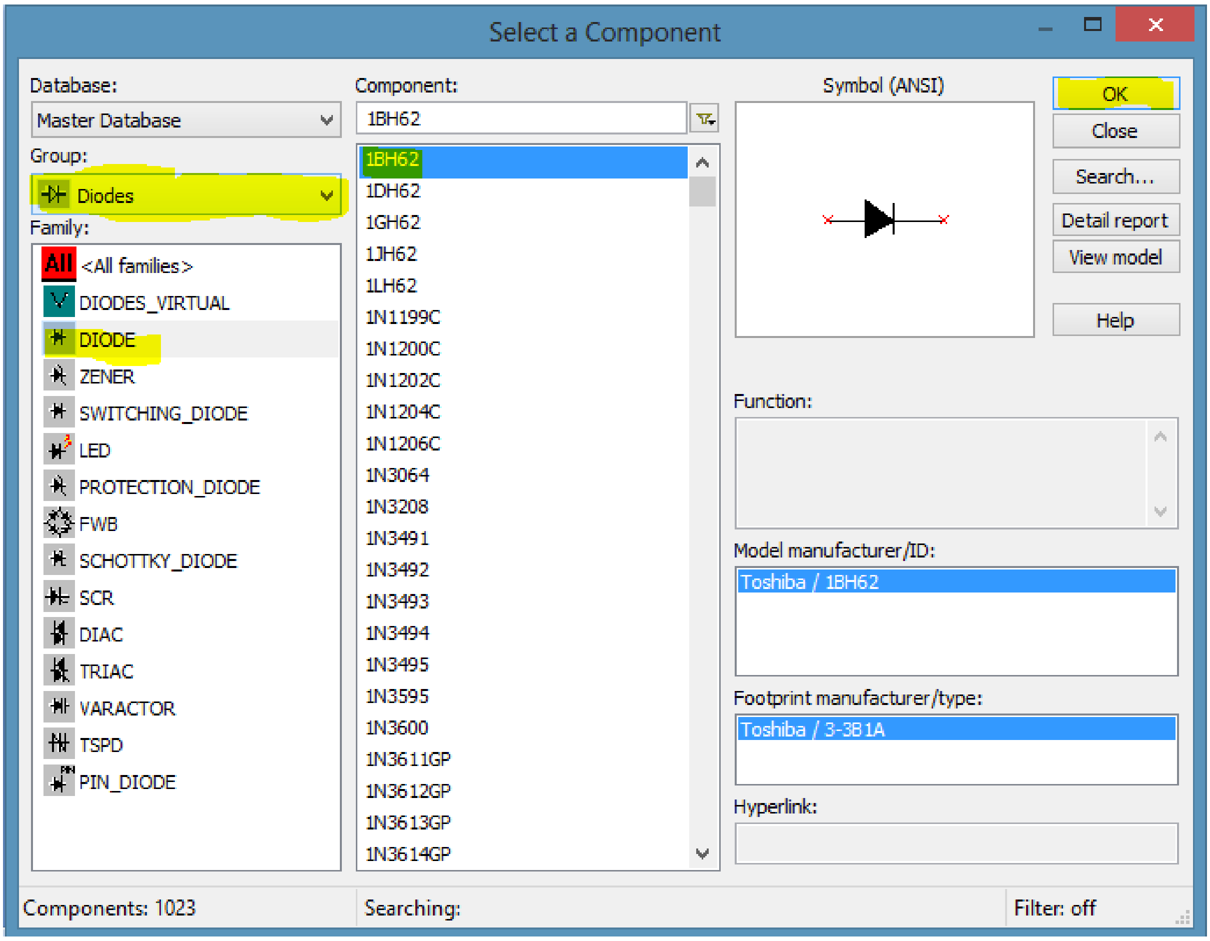
a) Diode=01
b) Resistor=01
c) AC power source
d) Ground
e) Oscilloscope for analyzing output

Procedure:

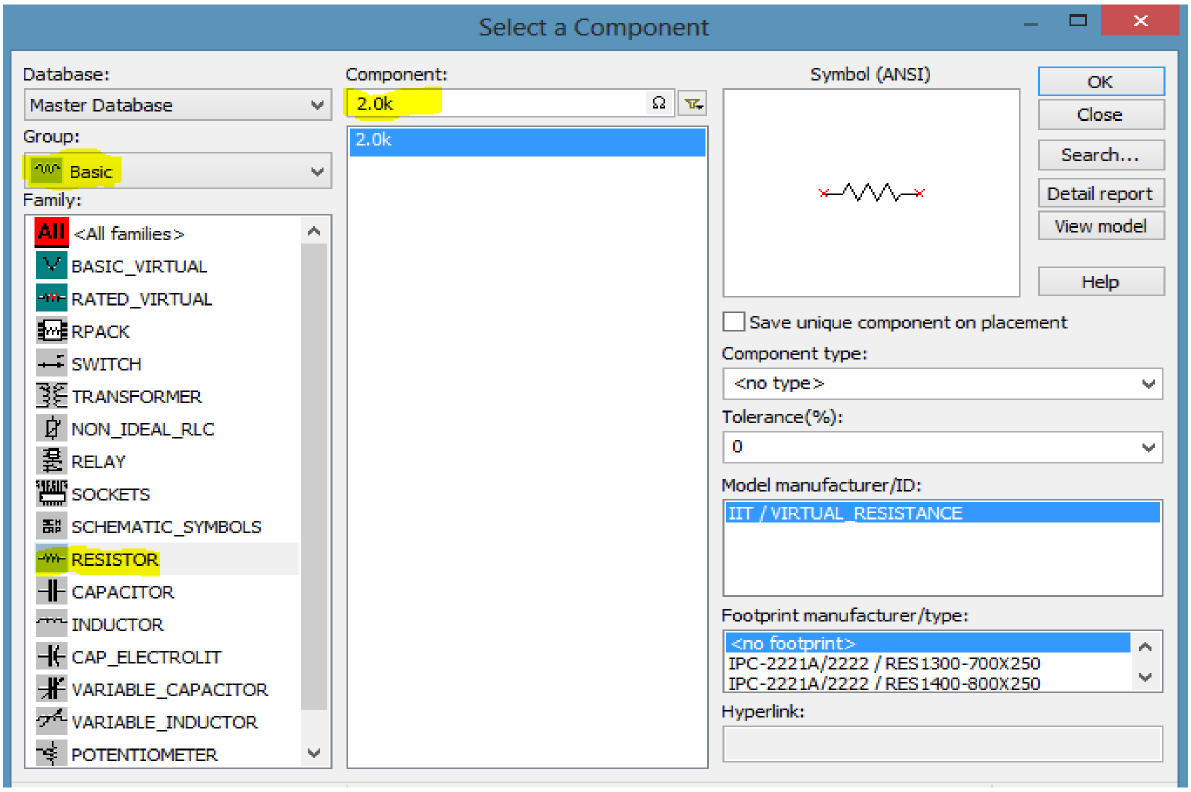
1. Open Multisim and from menu bar follow the path: **Place → Component** as shown in figure below.



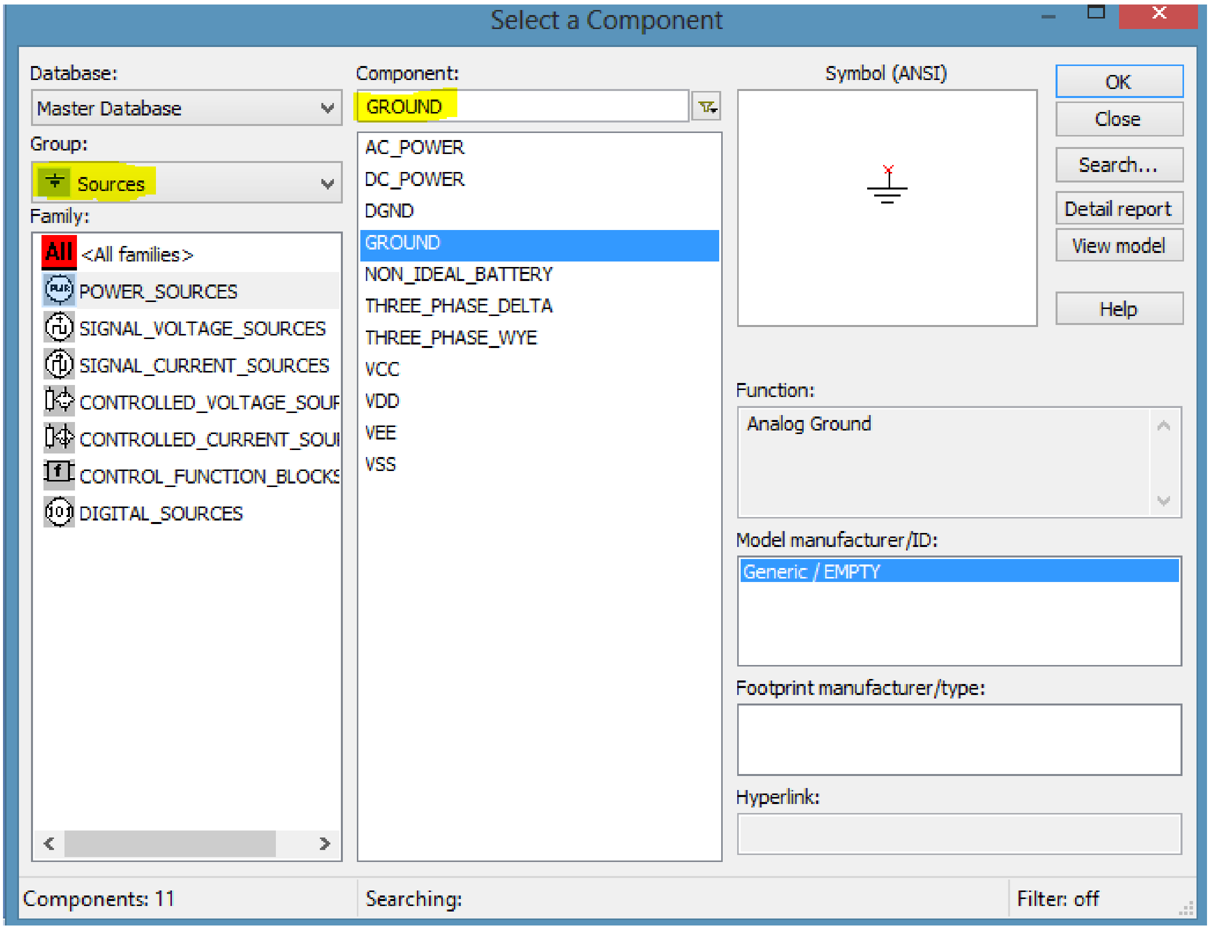
2. From the **Group** select ***Diode***; a list of diode with symbols will appear, choose any one of them (e.g. IBH62, IN6064 etc.) and click **OK** to place it on design board.



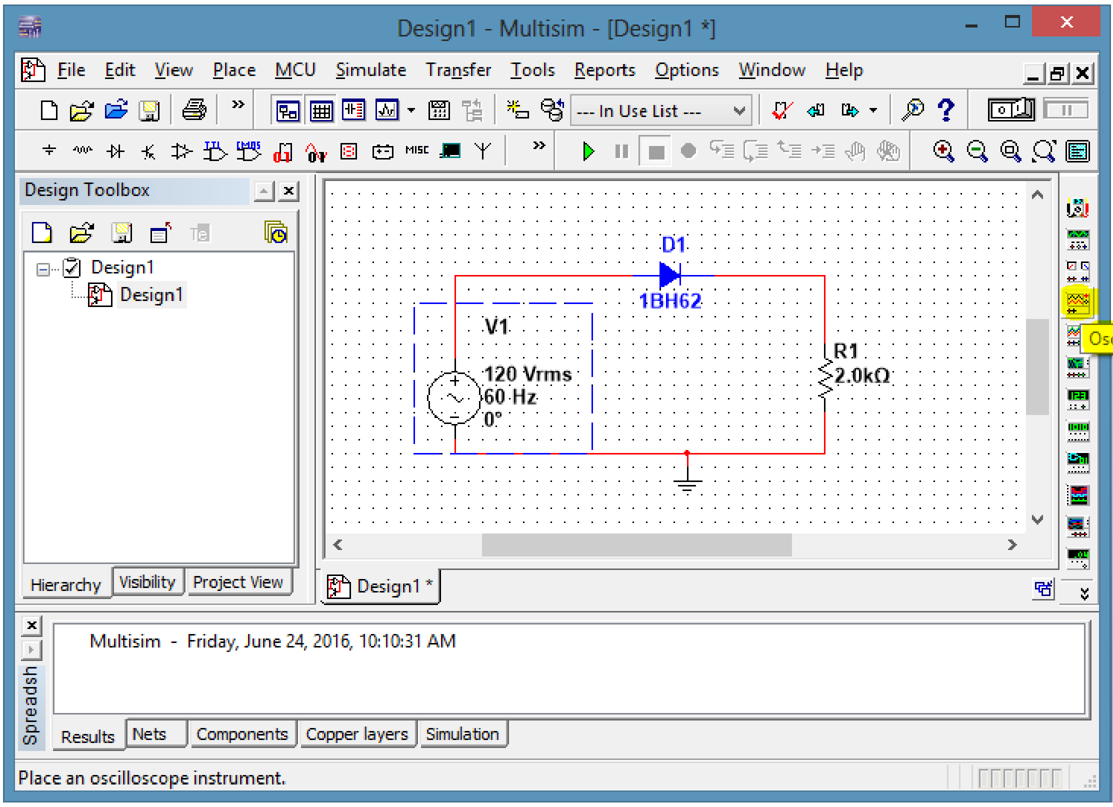
3. From the **Group** select ***Basic***; and form **Family** select ***Resistor*** and set the required value (e.g. 1KΩ, 2KΩ etc.) and click **OK** to place it on design board.

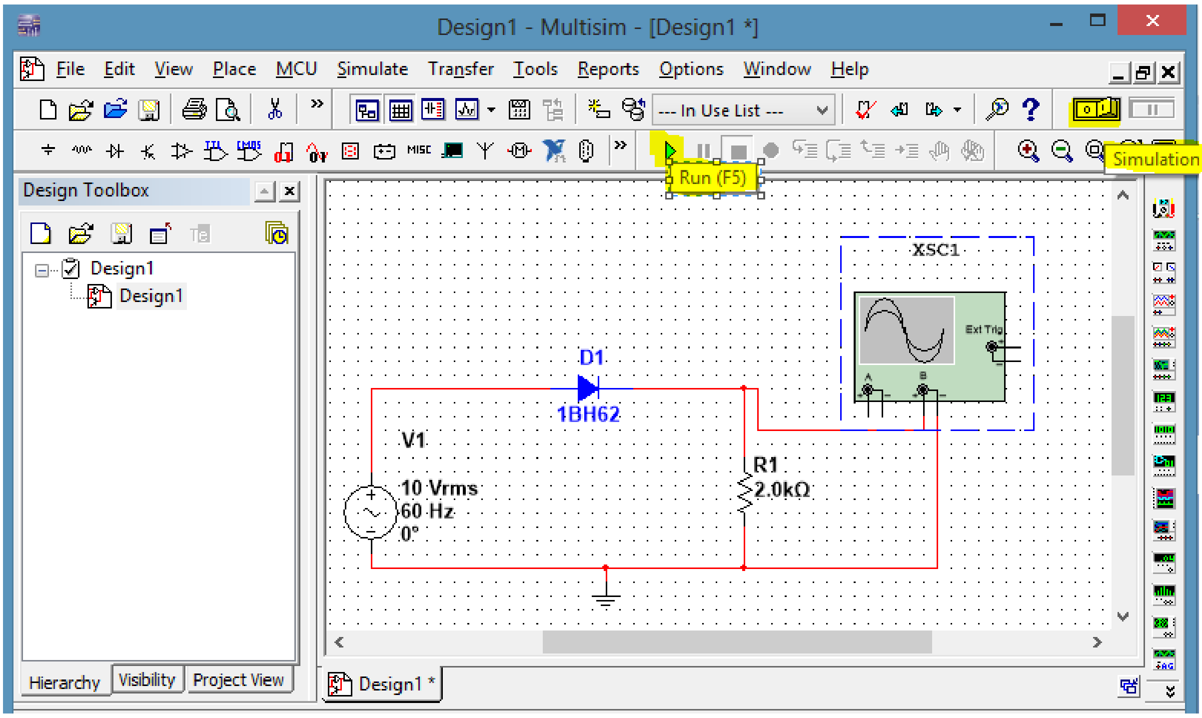


4. Repeat step 2 and select **<u>Sources</u>**; form **<u>Family</u>** select ***Power Sources*** which shows the list of power sources then select **AC_POWER**.

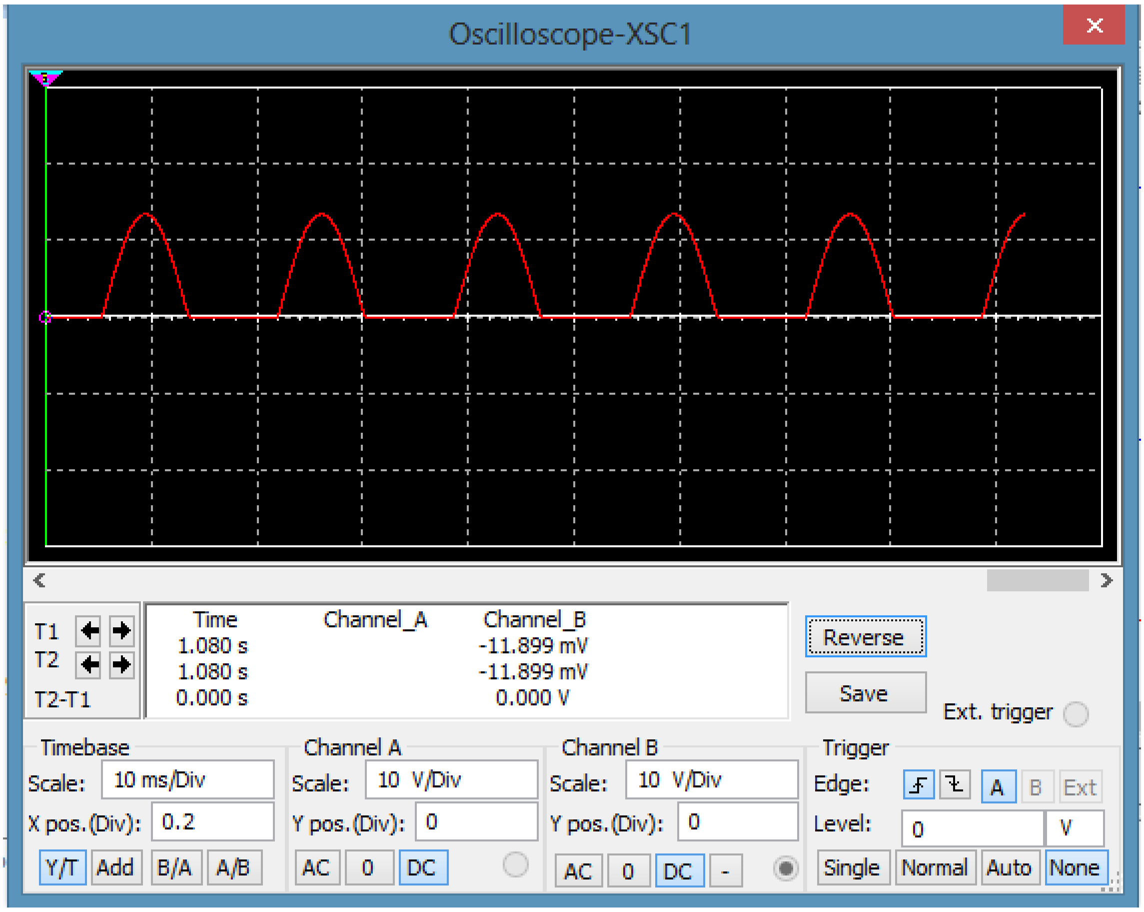


5. Repeat step 4 and select **GROUND** instead of AC_POWER.
6. Connect all components using wire and adjust the value of Power source and other components by ***right click*** or ***double click*** on its icon.
7. Select **<u>Oscilloscope</u>** as shown in figure below and connect across the load resistor as shown in figure below and run the simulation either by ***<u>Run button</u>*** or ***<u>Simulation Switch</u>***.





8. Double Click on **<u>Oscilloscope Icon</u>** to see the output as shown in figure below.


Oscilloscope-XSC1
Time Channel_A Channel_B
T1 1.080 s -11.899 mV
T2 1.080 s -11.899 mV
T2-T1 0.000 s 0.000 V
Reverse
Save
Ext. trigger
Timebase
Scale: 10 ms/Div
X pos.(Div): 0.2
Y/T Add B/A A/B
Channel A
Scale: 10 V/Div
Y pos.(Div): 0
AC 0 DC
Channel B
Scale: 10 V/Div
Y pos.(Div): 0
AC 0 DC -
Trigger
Edge: A B Ext
Level: 0 V
Single Normal Auto None

PRACTICAL NO. 19

IMPLEMENTATION OF SERIES/PARALLEL CIRCUITS USING PSPICE

Objective:

- Familiar with PSPICE technical software& how it is use to implement, simulate and analyze electrical networks.

Introduction

SPICE is a powerful general purpose analog and mixed-mode circuit simulator that is used to verify circuit designs and to predict the circuit behavior. This is of particular importance for integrated circuits. SPICE was originally developed at the Electronics Research Laboratory of the University of California, Berkeley (1975). SPICE stands for Simulation Program for Integrated Circuits Emphasis.

PSpice is a PC version of SPICE (which is currently available from OrCAD Corp. of Cadence Design Systems, Inc.). OrCAD Capture enables fast and intuitive schematic design entry for PCB development or analog simulation using PSpice. The OrCAD student edition is called PSpice AD Lite. Light version of PSpice has the following limitations: Circuits have a maximum of 64 nodes, 10 transistors and 2 operational amplifiers.

SPICE can do several types of circuit analyses, such as

- **Non-linear DC analysis:** Calculates the DC transfer curve.
- **Non-linear transient and Fourier analysis:** Calculates the voltage and current as a function of time when a large signal is applied; Fourier analysis gives the frequency spectrum.
- **Linear AC Analysis:** Calculates the output as a function of frequency. A bode plot is generated.
- Noise analysis

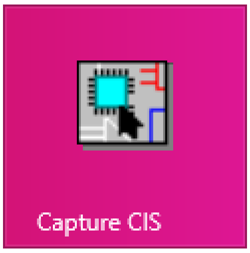
PSpice has analog and digital libraries of standard components (such as NAND, NOR, flip-flops, MUXes, FPGA, PLDs and many more digital components). All analyses can be done at different temperatures, default temperature is 300K. PSpice circuit can contain the following components:

- Independent and dependent voltage and current sources
- Resistors, Capacitors, Inductors, Mutual inductors
- Transmission lines , Operational amplifiers, Switches
- Diodes, Bipolar transistors, MOS transistors
- JFET
- MESFET
- Digital gates etc.

PRACTICAL NO. 20

STARTING THE PSPICE SOFTWARE

To start Orcad Capure, double-click the desktop icon,



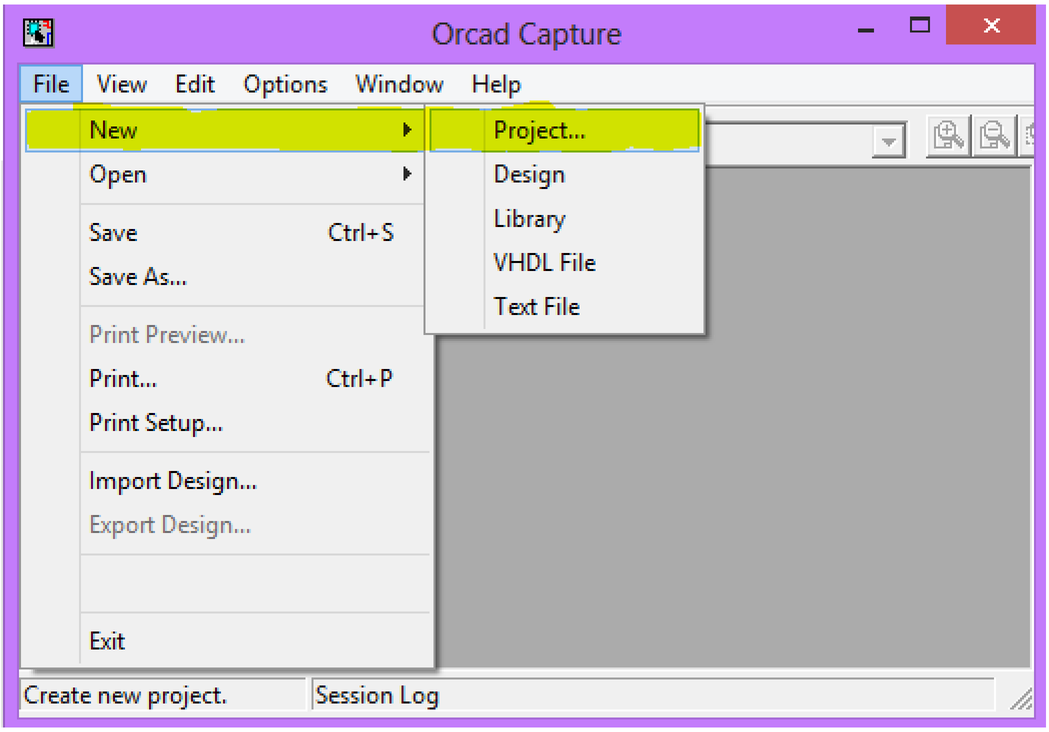
or start ISE from the Start menu by selecting:

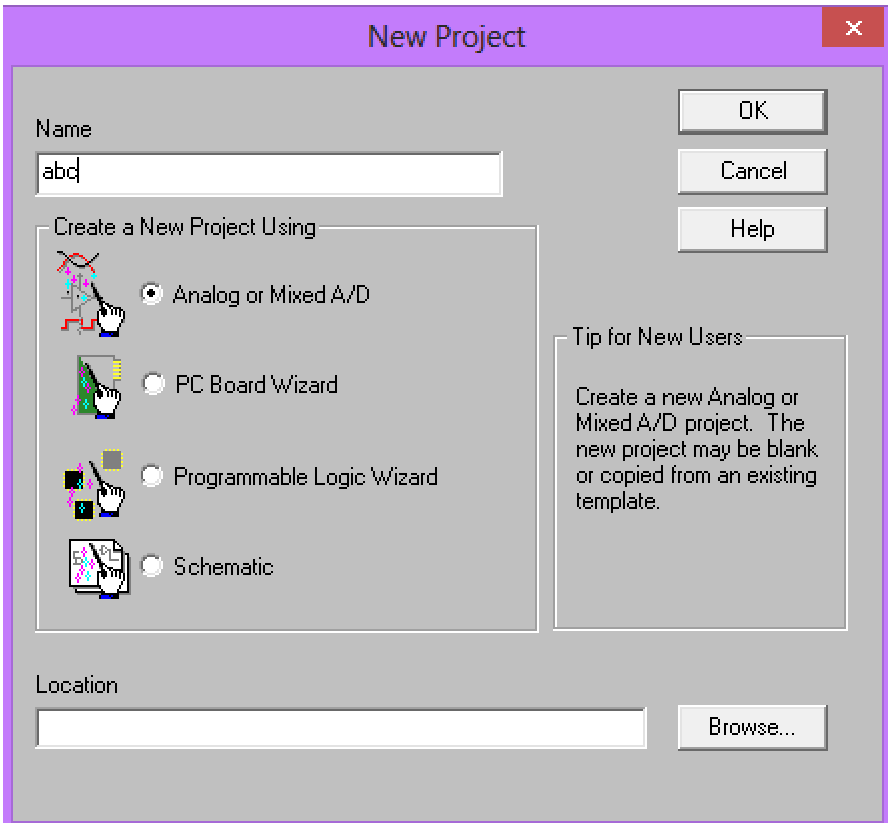
Start ->All Programs ->Orcad 9.2 ->Capture CIS

CREATE A NEW PROJECT

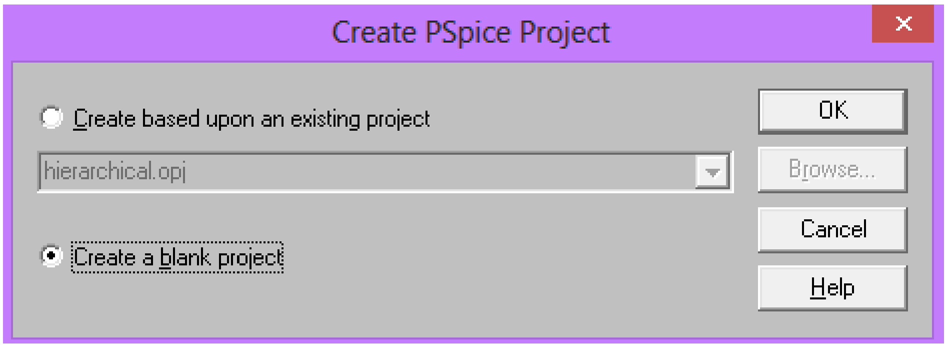
To create a new project:

1. Select **File ->New Project...** The New Project Wizard appears as shown in figure below

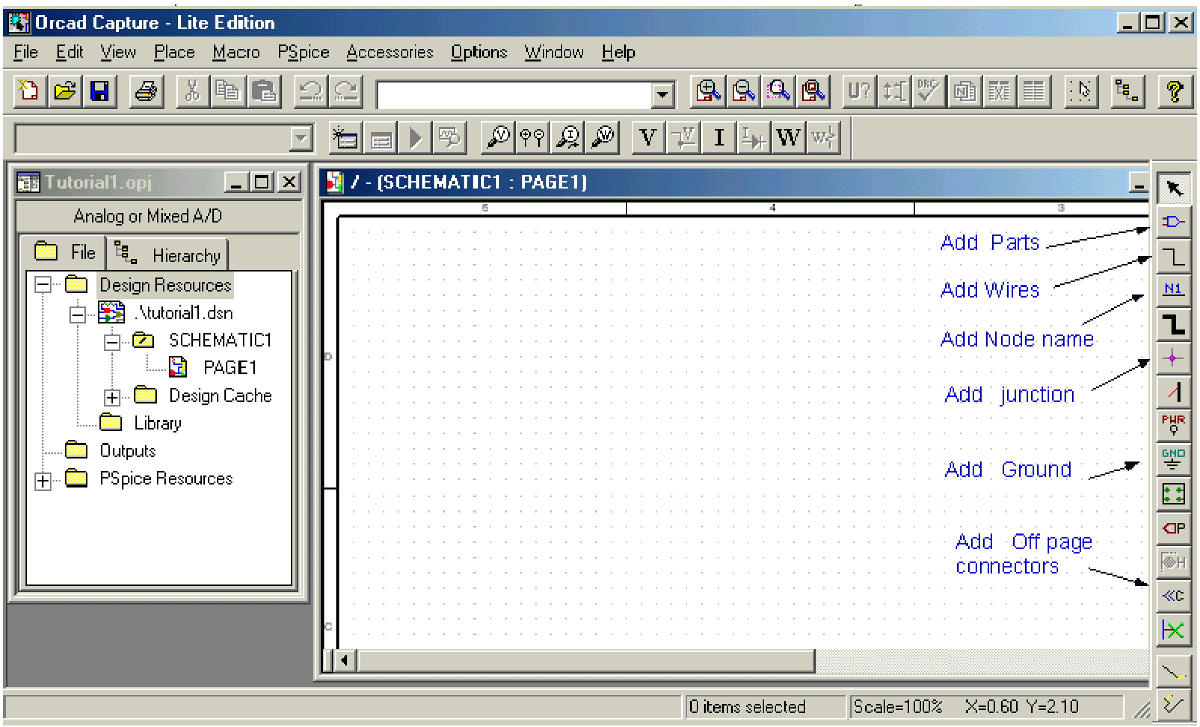




2. Type name of project as **abc**in the Project Name field.
3. Choose **Analog or Mixed A/D** and click on **OK** button.
4. Select Location C:\PROGRAM FILES\ORCAD\Capture.
5. When the **Create PSpice Project** box opens as shown in figure below, select "**Create Blank Project"** and click on **OK.**



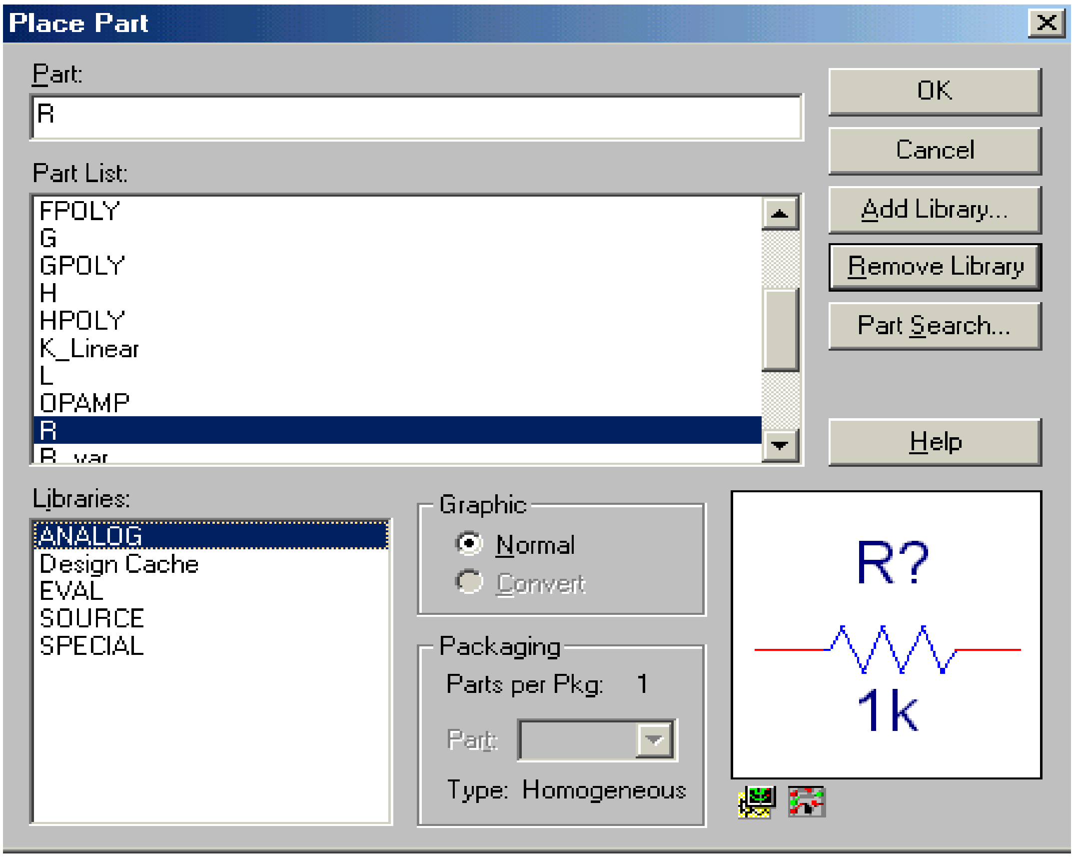
6. A new page will open in the Project Design Manager as shown below.


Orcad Capture - Lite Edition
File Edit View Place Macro PSpice Accessories Options Window Help
Tutorial1.opj
Analog or Mixed A/D
File
Hierarchy
Design Resources
.\tutorial1.dsn
SCHEMATIC1
PAGE1
Design Cache
Library
Outputs
PSpice Resources
/ - (SCHEMATIC1 : PAGE1)
Add Parts
Add Wires
Add Node name
Add junction
Add Ground
Add Off page connectors
0 items selected
Scale=100%
X=0.60 Y=2.10

PRACTICAL NO. 21

PLACE THE COMPONENTS AND CONNECT THE PARTS

1. Click on the Schematic window in Capture.
2. To Place a part go to PLACE/PART menu or click on the Place Part Icon. This will open a dialog box shown below.



3. Select the library that contains the required components.
4. Type the beginning of the name in the Part box. The part list will scroll to the components whose name contains the same letters. If the library is not available, you need to add the library, by clicking on the Add Library button. This will bring up the Add Library window. Select the desired library. For Spice you should select the libraries from the Capture/Library/PSpice folder.

   Analog: contains the passive components (R, L, and C), mutual inductane, transmission line, and voltage and current dependent sources (voltage dependent voltage source E, current-dependent current source F, voltage-dependent current source G and current-dependent voltage source H).

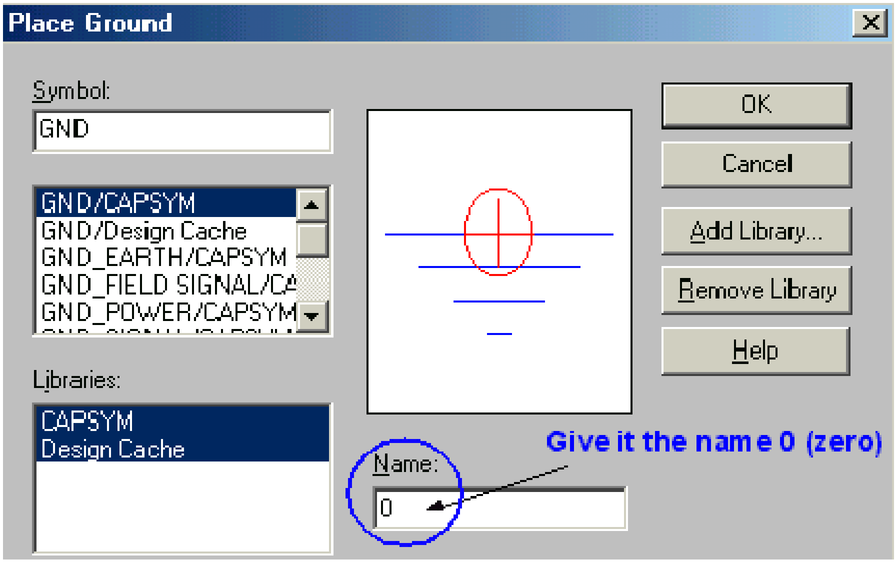
   Source: give the different type of independent voltage and current sources, such as Vdc, Idc, Vac, Iac, Vsin, Vexp, pulse, piecewise linear, etc.

   Eval: provides diodes (D…), bipolar transistors (Q…), MOS transistors, JFETs (J…), real opamp such as the u741, switches (SW_tClose, SW_tOpen), various digital gates and components.

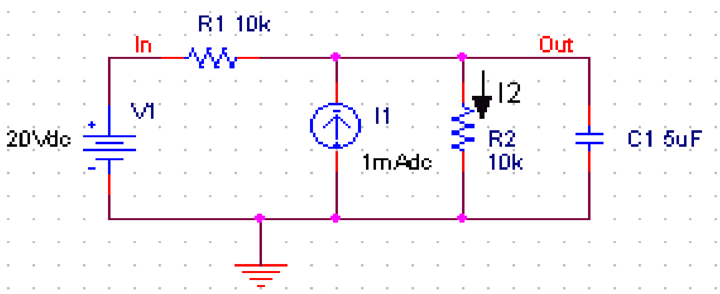
Abm: contains a selection of interesting mathematical operators that can be applied to signals, such as multiplication (MULT), summation (SUM), Square Root (SWRT), Laplace (LAPLACE), arctan (ARCTAN), and many more.
Special: contains a variety of other components, such as PARAM, NODESET, etc.

5. Place the resistors, capacitor (from the Analog library), and the DC voltage and current source. You can place the part by the left mouse click. You can rotate the components by clicking on the R key. To place another instance of the same part, click the left mouse button again. Hit the ESC key when done with a particular element. You can add initial conditions to the capacitor. Double-click on the part; this will open the Property window that looks like a spreadsheet. Under the column, labeled IC, enter the value of the initial condition, e.g. 2V. For our example we assume that IC was 0V (this is the default value).

6. After placing all part, you need to place the Ground terminal by clicking on the **GND** icon. When the Place Ground window opens, select **GND/CAPSYM** and give it the name 0 (i.e. zero). Do not forget to change the name to 0, otherwise PSpice will give an error or "**Floating Node**". The reason is that SPICE needs a ground terminal as the reference node that has the node number or name 0 (zero).



7. Now connect the elements by clicking on the **PLACE WIRE ICON**.
8. You can assign names to nets or nodes using the **Place Net Alias** command (PLACE/NET ALIAS menu). We will do this for the output node and input node. Name these OUT and IN, as shown in circuit.



PRACTICAL NO. 22

ASSIGN VALUES AND NAMES TO THE PARTS/COMPONENTS

1. Change the values of the resistors by double-clicking on the number next to the resistor. Resistor name can also be change. Do the same for the capacitor and voltage and current source.
2. If you haven't done so yet, you can assign names to nodes (e.g. OUT and IN nodes).
3. Save the project.

Netlist

The netlist gives the list of all elements using the simple format:

R_name node1 node2 value

C_namenodexnodey value, etc.

1. Generate netlist by going to the **PSPICE/CREATE NETLIST** menu.
2. Look at the netlist by double clicking on the Output/name.net file in the Project Manager Window (in the left side File window).

Note on Current Directions in elements:

The positive current direction in an element such as a resistor is from node 1 to node 2. Node 1 is either the left pin or the top pin for a horizontal or vertical positioned element (e.g. a resistor). By rotating the element 180 degrees one can switch the pin numbers. To verify the node numbers you can look at the netlist:

e.g. R_R2 node1 node2 10k

e.g. R_R2 0 OUT 10k

Since we are interested in the current direction from the OUT node to the ground, we need to rotate the resistor R2 twice so that the node numbers are interchanged:

R_R2 OUT 0 10k

SPECIFYING THE TYPE OF ANALYSIS AND SIMULATION

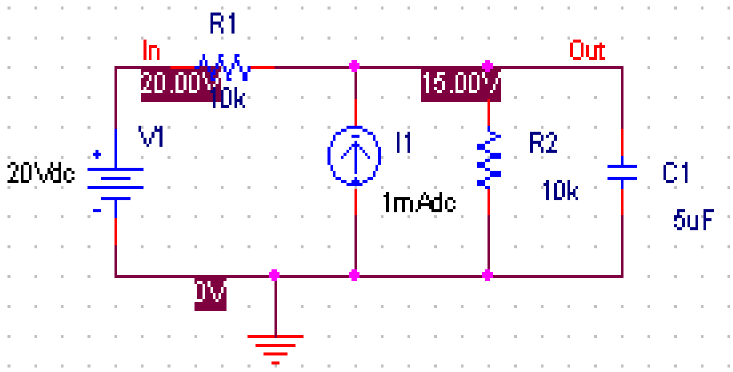
Spice allows the following analysis

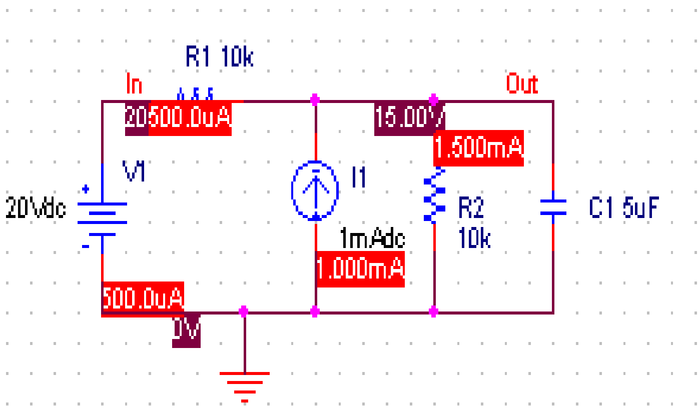
- DC bias
- DC Sweep
- Transient with Fourier analysis
- AC analysis
- Montecarlo/worst case sweep
- Parameter sweep and Temperature sweep.

BIAS or DC Analysis

1. With the schematic open, go to the PSPICE menu and choose **NEW SIMULATION PROFILE.**
2. In the Name text box, type a descriptive name, e.g. Bias.
3. From the **Inherit from List**: select **none** and click Create.

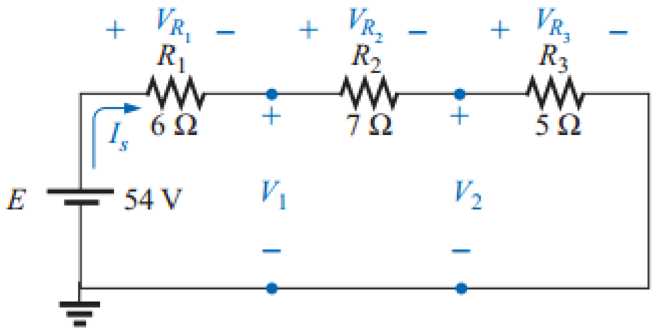
4. When the Simulation Setting window opens, for the **Analysis Type**, choose Bias Point and click OK.
5. Now you are ready to run the simulation: **PSPICE/RUN**
6. A window will open, letting you know if the simulation was successful. If there are errors, consult the Simulation Output file.
7. To see the result of the DC bias point simulation, you can open the Simulation Output file or go back to the schematic and click on the V icon (Enable Bias Voltage Display) and I icon (current display) to show the voltage and currents (see Figure below).
   To check the direction of the current, you need to look at the netlist: the current is positive flowing from node1 to node1.





Practice Lab:

- Design & Simulate the circuit below in Orcad Capure environment.



PRACTICAL NO. 23

INTRODUCTION OF MICROPROCESSOR & BLOCK DIAGRAM OF 8086 MICROPROCESSOR

Objective:

This practical provide the basic understanding of microprocessor, how 8085 microprocessor is different from 8086.

Theory:

A microprocessor also known as central processing unit (CPU) perform the following operations:

- ❖ Arithmetic and logic operations (i.e. Addition, subtraction, AND, OR, etc.)
- ❖ Data transfers (to memory or I/O).
- ❖ Program flow via simple decisions.

It is a complete computation engine that is fabricated on a single chip. The microprocessor is the heart of any normal computer. Figure 20.1 shows the block diagram of microprocessor.

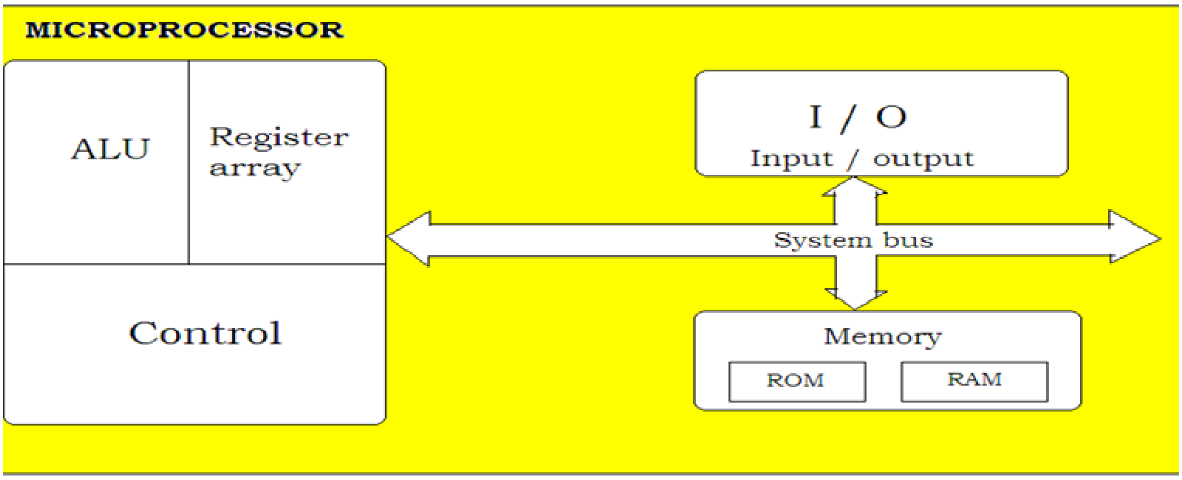


Figure 20.1: Block diagram of microprocessor

ALU: Stands for Arithmetic Logic Unit and responsible for arithmetic (i.e. addition, subtraction etc.) as well as logical (i.e. AND, OR, NOT etc.) operations.

CU: Stands for Control Unit and responsible to control the activities of other components.

Register Array: Used to store temporary data for processing.

Intel 8086 Microprocessor:

The 8086 is a 16-bit microprocessor chip. It supports x86 architecture. Intel 8086, released in 1979. The block diagram of Intel 8086 microprocessor is shown in Figure 20.2.

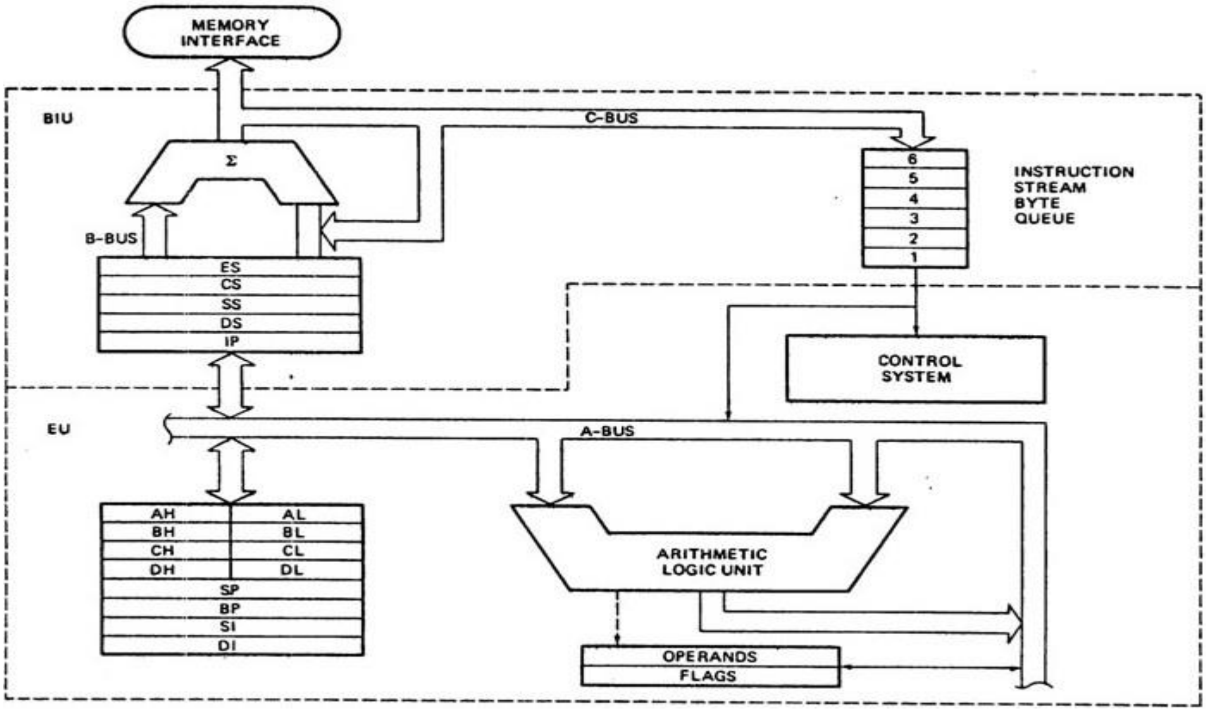


Figure 20.2: Block diagram of Intel 8086 microprocessor

In figure 20.2

- AX, BX, CX &DX are general purpose registers.
- SP, BP, SI & DI are special purpose registers, each have specific task.
- ES, CS, SS & DS are Segment registers, they address a section of memory in a program.
- FLAGS: Used to indicate conditions of the microprocessor.

Main difference between 8085 & 8086 Microprocessor

Intel 8085	Intel 8086
• 8-bit microprocessor. A microprocessor which has n data lines is called an n-bit microprocessor i.e., the width of the data bus determines the size of the microprocessor.	• 16-bit microprocessor.
• Its speed is 3 MHz.	• Its speed can vary between 5, s8, and 10MHz for three different microprocessor.
• It has 5 flag registers.	• It has 9 flag registers.
• It has 16-bit address lines.	• It has 20-bit address lines.
• It has 6500 transistors.	• It has 29000 transistors.

PRACTICAL NO. 24

INTRODUCTION OF MACHINE & ASSEMBLY LANGUAGE

Objective:

The practical provide the basic concept of machine and assembly language.

Machine Language:

It is a Programming language that can be directly understood by a machine (computer) without any conversion (translation). It is only understood by the computer and almost impossible for humans to use because it consist entirely on numbers. Programmers, therefore, use either a high-level programming language or an assembly language. Figure 21.1 shows the level of different languages.

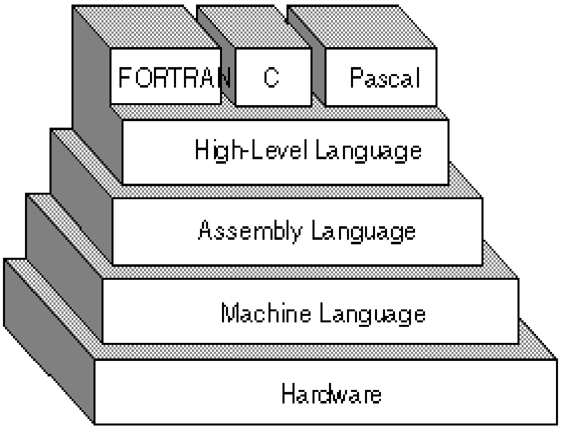
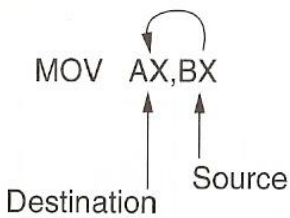


Figure 21.1: Level of programming languages

Assembly Language:

An assembly language contains the same instructions as a machine language, but the instructions and variables have names instead of being just numbers. Programs written in high-level languages are translated into assembly language or machine language by a compiler. Assembly language programs are translated into machine language by a program called an assembler. Every CPU has its own unique machine language. Intel microprocessors used assembly language processing. Assembly language is converted into executable machine code by a utility program referred to as an assembler. A program written in assembly language consists of a series of processor instructions. For example a MOV instruction is most common instruction in assembly language program MOV instruction copies the contents (data) of one register to other register/memory.MOV copies the source into the destination (See Fig below).



PRACTICAL NO. 25

ASSEMBLY LANGUAGE PROGRAM FOR ADDITION & SUBTRACTION

Objective:

Objective of this practical to familiar with simple Assembly language program for addition & subtraction.

ADDITION

Assembly Code	Description
MOV AL, 12H **MOV** BL, 05H **ADD** AL, BL	Put 12H in register AL Put 05H in register BL add BL with AL (i.e. 12H+05H) and store result in AL.
ADD AL, BL **ADD** CX, DI **ADD** CL, 44H **ADD** [BX], AL	AL=AL+BL CX=CX+DI CL=CL+44H Add the contents of memory location which is stored in BX with data stored in AL.

SUBTRACTION

Assembly Code	Description
MOV AL, 12H **MOV** BL, 05H **SUB** AL, BL	Put 12H in register AL Put 05H in register BL Subtract BL from AL (i.e. 12H-05H) and store result in AL.
SUB AL, BL **SUB** CX, DI **SUB** CL, 44H **SUB** [BX], AL	AL=AL-BL CX=CX-DI CL=CL-44H Subtract the contents of memory location which is stored in BX with data stored in AL.

PRACTICAL NO. 26

ASSEMBLY LANGUAGE PROGRAM FOR MULTIPLICATION & DIVISION

Objective:

Objective of this practical to familiar with simple Assembly language program for multiplication & division.

MULTIPLICATION

Assembly Code	Description
MOV BL, 5 **MOV** CL, 10 **MOV** AL, CL **MUL** BL **MOV** DX, AX	Store 5 in register BL Store 10 in register CL Copy the value of CL register into AL register. Multiply AL register with BL register. Result of multiplication is stored in DX

DIVISION

Assembly Code	Description
MOV AL, NUMB **MOV** AH, 0 **DIV** NUMB1 **MOV** ANSQ, AL **MOV** ANSR, AH	Get NUMB Zero-extended Divide by NUMB1 Save quotient Save remainder

PRACTICAL NO. 27

INTRODUCTION TO PLC

Objective:

This Lab provides the basic understanding of PLCs, how they are different from Micro-controller.

Introduction to PLC:

A Programmable Logic Controller (PLC) is a digital computer used for automation of electromechanical processes, such as control of machinery on factory assembly lines, heating, cooling or lighting fixtures. PLCs are used in many industries and machines. Unlike general-purpose computers, the PLC is designed for multiple inputs and output arrangements, extended temperature ranges, immunity to electrical noise, and resistance to vibration and impact. Programs to control machine operation are typically stored in battery-backed or non-volatile memory. A PLC is an example of a hard real time system since output results must be produced in response to input conditions within a bounded time, otherwise unintended operation will result.

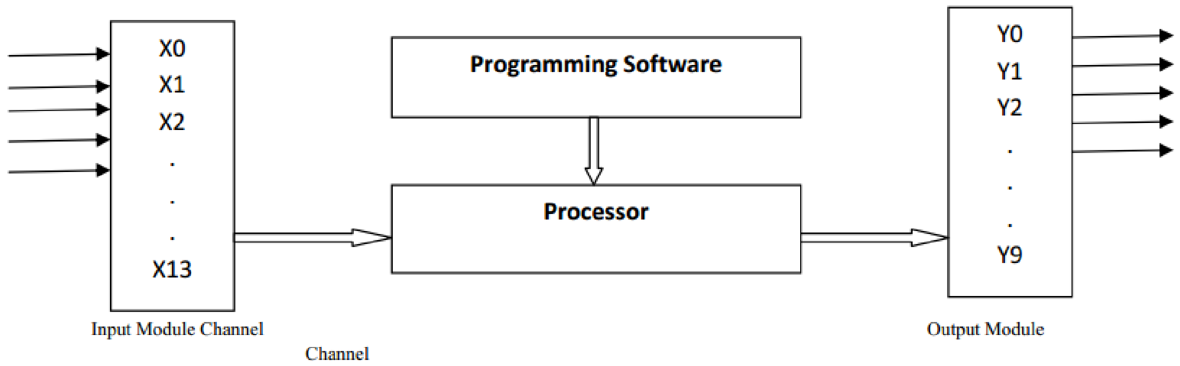


Figure: Basic Structure of Programmable Logic Controller (PLC)

The processor performs arithmetic and logic operations on input variable data and determines the proper state of the output variables. Input module channel examine the state of physical switches and other input devices and put their state into the form suitable for the processor. The PLC is able to accommodate a number of inputs, called channels. The output modules supply ac power to external devices such as motors, lights, solenoids, and so on.

PLC Operation can be considered in two modes, the I/O scan mode and the execution mode. In I/O scan mode, the processor updated all outputs and inputs the state of all inputs one channel at a time. In execution mode, the processor evaluates each rung of the ladder diagram program that is being executed sequentially, starting from the first rung and proceeding to the last rung. Scan Time is an important characteristic of PLC which is how much time is required for one complete cycle of I/O scan and execution. This depends upon number of I/O channels involved, length of the ladder diagram program and the processing speed. A typical maximum scan and execution time is 5 to 20 ms.

Difference between PLC & Microcontroller

PLC and Micro-controller can be chosen for some applications interchangeably.

There major differences are listed below:

1. PLC is a special microcontroller designed for industrial application used for controlling machinery & processes whereas microcontroller is a microprocessor that can be programmed for any type of applications.
2. PLC is programmed using ladder diagram whereas microcontroller is programmed using C or Assembly Language.
3. PLC process is optimized to scan the inputs and determines the outputs based on the logical conditions programmed into it by the user whereas microcontroller is a general purpose IC.
4. PLC come in a package of Inputs units, Output unit containing LEDs & processing unit, whereas microcontroller comes in a simple IC.
5. PLC is more useful in high power applications whereas microcontroller is more suitable to low power applications.
6. PLC is not efficient in extensive calculations whereas microcontroller can perform arithmetic calculations more efficiently.

Be aware of your own worth, use all of your power to achieve it. Create an ocean from a dewdrop. Do not beg for light from the moon, obtain it from the spark within you.

— *Muhammad Iqbal* —