Scilab Textbook Companion for Basic Electronics by A. P. Godse and U. A. Bakshi[1]

Created by
Boston Shullai
engineering
Others
Sri Jayachamarajendra College of Engineering.
College Teacher
None
Cross-Checked by
Bhavani Jalkrish

July 31, 2019

[1]Funded by a grant from the National Mission on Education through ICT, http://spoken-tutorial.org/NMEICT-Intro. This Textbook Companion and Scilab codes written in it can be downloaded from the "Textbook Companion Project" section at the website http://scilab.in

Book Description

Title: Basic Electronics

Author: A. P. Godse and U. A. Bakshi

Publisher: Technical Publications, Pune

Edition: 1

Year: 2006

ISBN: 8184310811

Scilab numbering policy used in this document and the relation to the above book.

Exa Example (Solved example)

Eqn Equation (Particular equation of the above book)

AP Appendix to Example(Scilab Code that is an Appednix to a particular Example of the above book)

For example, Exa 3.51 means solved example 3.51 of this book. Sec 2.3 means a scilab code whose theory is explained in Section 2.3 of the book.

Contents

List of Scilab Codes

List of Figures

Chapter 2

Semiconductor Devices

Scilab code Exa 2.1 `Dc reverse Resistance`

```
//Ex-2.1 Pg-2.18
clc;
clear;
Vf=0.2; //voltage in volts
Vr=60; //voltage in volts
If=60*10^(-3); //current in ampere
I0=0.025*10^(-3); //current in ampere
Rf=Vf/If;//forward resistance
Rr=Vr/I0;//reverse resistance
Rr=Rr*1e-6
printf("the equivalent resistance are Rf=%.3f ohm
    and Rr=%-.1f M ohm",Rf,Rr)
```

Scilab code Exa 2.2 `Dynamic forward resistance`

```
//Chapter-2 Ex-2.2 Pg-2.18
clc
clear;
```

```
disp("refer to the Figure-2.19 given ")
disp("from the characteristics at point P, Vf=0.7V, If=60mA")
Vf=0.7;
If=0.06;
Rf=Vf/If;//DC forward resistance
printf("\n DC forward resistance Rf : %.2f ohm\n",Rf)
disp("as the forward voltage changes from P to Q")
delta_Vf=0.77-.7;
delta_If=(120-60)*10^(-3);
rf=delta_Vf/delta_If;//dynamic forward resistance
printf("\n Dynamic forward resistance rf : %.3f ohm",rf)
```

Scilab code Exa 2.3 `Current through the resistance`

```
//Ex2_3 Pg-2-22
clc
clear;
disp(" Refer to the figure-2.24 shown ")
disp(" since Rf=0 The circuit becomes as shown in figure-2.24(a)")
V=10;//supply voltage
Rf=0;//forward resistance
Rl=1;//load resistance in k ohm
Vin=0.7;//cut in voltage
Il=(V-Vin)/Rl;//applying KVL to the loop
If=Il;
printf("\n \n current through the resistance Il=If = is %.1f mA",If)
Vl=Il*Rl;
printf("\n \n voltage across Rl is %.1f V",Vl)
Pd=If*Vin;
printf("\n \n diode power Pd = %.2f mW",Pd)
```

```
Pl=Il*Vl;
printf("\n \n load power Pl = %.2f mW",Pl)
```

Scilab code Exa 2.4 Forward current through the diode

```
//EX2_4 PG-2.23
clc
disp("Refer the Figure-2.25 shown")
disp("When forward resistance Rf is neglected  then"
   )
disp("the diode behaves as a battery as shown in
   Figure-2.25(a)")
Vf=0.7;//cut-in voltage
V=10;//supply voltage
Rl=500;//load resistance
If=(V-Vf)/Rl;//applying KVL to the circuit
If=If*1e3
printf("\n Forward current is %.2f mA\n",If)
disp("When forward resistance is Rf is 3.2 Ohm then"
   )
disp("the equivalent circuit is as shown in fig
   -2.25(b)")
Rf=3.2;
If=(V-Vf)/(Rl+Rf);//applying KVL to the circuit
If=If*1e3
printf("\n therefore Forward current is %.4f mA",If)
```

Scilab code Exa 2.5 Piece Wise linear characteristic of a Diode

```
//EX2_5 PG-2.23
clc
If=80e-3;//maximum forward current
Rf=0.4;//dynamic resistance
```

```
Vin=0.3;//cut-in voltage for germanium
disp("when forward current is zero then")
Vf=Vin;//voltage across the diode
printf(" voltage across the diode is %1.1f V\n",Vf)
disp("when forward current is 80mA then")
Vf=Vin+If*Rf;
printf(" voltage across the diode is %1.3f V",Vf)
clf()
x=[0 .1 .2 .3 .332];//x-coordinate
y=[0 0 0 0 80];//y-coordinate
plot2d(x,y)
xlabel('voltage across the diode (V) ');
ylabel('current (mA)');
title('Piecewise linear characteristic')
xgrid(color("grey"));
```

Scilab code Exa 2.6 Load resistance Rl

```
//EX2_6 PG-2.28
clc
disp("Refer to the Figure:2.29 shown")
If=25e-3;//current at Q-point
disp("for Q point If=25mA, ie Iq=25mA")
disp("for If=0 A, Vf=Vin=3V at point B")
disp("Now we draw the graph")
disp("The coordinates are Q=(1V,25mA) B=(3V,0mA)")
clf()
x=[ 0 0.5 0.6    1   1.1 ];//x-coordinate
y=[ 0  1   5     25   30 ];//y-coordinate
plot(x,y,)
x1=[0.5   1    3];//x-coordinate
y1=[ 31   25   0];//y-coordinate
plot(x1,y1)
x2=[ 0 1];
y2=[25 25];
```

```
plot2d(x2,y2,style=6)
xlabel('Vf (volts)');
ylabel('If (mA)');
title("Piece-wise linear characteristic")
xgrid(color("red"));
disp("Q-point is denoted by the intersection of two
    lines as shown in the plot")
delta_If=10e-3;//from the graph plotted
delta_Vf=0.9;//from the graph plotted
s=delta_If/delta_Vf;//slope
disp("Therefore load resistance is the reciprocal of
    the slope ")
Rl=1/s;//load resistance
printf("\n required load resistance is %.0f ohm",Rl)
```

Scilab code Exa 2.7 New supply voltage

```
//EX2_7 PG-2.29
clc
Rl=150;//load resistance
If=35e-3;//current at which Q-point appears
disp("We draw the new DC load line in the plot")
s=-1/Rl;//slope
printf("\n slope is %f \n",s)
x=[ 0 0.5  1   1.1 1.15 ];//x-coordinate
y=[ 0  1  25   35   40 ];//y-coordinate
clf()
plot2d(x,y,style=2)
x1=[0 1.1 2.6 6.4 ];//x-coordinate
y1=[42  35   25  0];//y-coordinate
plot2d(x1,y1)
x2=[ 0  1  1.1];//x-coordinate
y2=[35 35  35];//y-coordinate
plot2d(x2,y2,style=5)
x3=[0  1  2.6];//x-coordinate
```

```
y3=[25 25 25];//y-coordinate
plot2d(x3,y3,style=4)
x4=[1.1 1.1 1.1];
y4=[ 0  10   35 ];
plot2d(x4,y4,style=6)
x5=[2.6 2.6];
y5=[0    25];
plot2d(x5,y5,style=7)
xlabel('Vf (volts)');
ylabel('If (mA)');
title("Piece-wise linear characteristic")
xgrid(color("gray"));
disp("The slope passes through the Q-point so the
   equation of the load ")
disp("line is If=(-Vf/Rl)+(Vin/Rl)")
disp("Now from the graph we can see that slope =(
   change in If)/(change in Vf)")
disp("For If=35 mA Vf=1.1 V")
disp("The coordinates are Q=(1.1V,35mA) C=(2.6V,25mA
   ) B=(6.4V,0mA)")
delta_If=10e-3;//change in If
delta_Vf=delta_If/abs(s);//change in Vf and we take
   only the magnitudeof the slope
printf("\n change in Vf is %1.1f V \n",delta_Vf)
disp("This is point C corresponding to If=35-10=35mA
   , as If is change by 10mA" )
disp("and Vf=2.6V ")
disp("We join the Q-point and point C as shown in
   the plot to get the DC load")
disp(" line we extend the line to intersect point B
   ")
disp("the voltage at point B is the new supply
   voltage required")
printf("\n From the plot voltage at point B = 6.7 V"
   )
```

Scilab code Exa 2.8 Pn junction diode current

```
//EX2_8 PG-2.33
clc
V=0.22;//forward bias voltage
T=25+273;//room temperature in degree kelvin
I0=2e-3;//reverse saturation current
n=1;//for germanium diode
k=8.62e-5//Boltzmann's constant
Vt=k*T;
I=I0*(exp(V/(n*Vt)));// diode current
printf("therefore the P-N junction diode current is
    %f A",I)
// in the book they have taken the approximate value
//hence the answer is slighty different. in the book
//Vt=0.02568(approx) whereas Vt=0.0256876(exact
    value)
```

Scilab code Exa 2.9 Maximum forward current

```
//EX2_9 PG-2.36
clc
P1=500e-3;//rated power rating of the diode
T1=27;//temperature in Degree Celsius
Df=4e-3;//Derating factor
disp("At temperature T=27 degree C")
disp("For silicon diode Vf=0.7 V is constant ")
Vf=0.7;
If1=P1/Vf;
printf(" \n Therefore maximum forward current at T
    =27 degree C is %.4f A \n",If1)
disp("At temperature T=77 degree C")
```

```
T2=77;//temperature in degree celsius
P2=P1-(T2-T1)*Df;//rated power rating of the diode
    at T=77 degree C
If2=P2/Vf;
printf("\n Therefore maximum forward current at T=77
    degree C is %.4f A",If2)
```

Scilab code Exa 2.10 `Forward voltage drop`

```
//EX2_10 PG-2.37
clc
T1=27;//initial temperature
Vf1=0.7;//forward voltage
Vtc=-2.3e-3;//voltage temperature coefficient
disp("at T2=25 degree C")
T2=50;
Vf2=Vf1+((T2-T1)*Vtc)
printf("\n therefore forward voltage drop at 50
    degree C is %.4f V \n",Vf2)
disp("at T3=77 degree C")
T3=77;
Vf2=Vf1+((T3-T1)*Vtc)
printf("\n therefore forward voltage drop at 77
    degree C is %.3f V",Vf2)
```

Scilab code Exa 2.11 `Dynamic resistance`

```
//EX2_11 PG-2.38
clc
If=30e-3;//forward current
T1=25+273;//temperature in degree kelvin
disp("Therefore at a temperature of 25 Degree C ")
Rf=26e-3/If;
```

```
printf("\n dynamic resistance is %.3f ohm \n",Rf)
disp("Therefore at a temperature of 75 Degree C ")
T2=75+273//Temperature in degree kelvin
Rf=Rf*(T2/T1);
printf("\n dynamic resistance is %.3f ohm",Rf)
```

Scilab code Exa 2.12 Maximum reverse recovery time

```
//EX2_12 PG-2.43
clc
Tf_min=1;//fall time in micro second
Trr_max=Tf_min/10
printf("the maximum recovery time is %.1f micro sec"
    ,Trr_max)
```

Scilab code Exa 2.13 Derated power rating

```
//EX2_13 PG-2.48
clc
PD_max=320e-3;//maximum power rating
T1=50;//temperature in degree celsius at which
    maximum power rating occurs
DF=2.3e-3;//Derating factor
disp("Therefore the derated power rating at a
    temperature T=100 Degree celsius")
T2=100;
PD=PD_max-DF*(T2-T1);
printf("\n is %.3f W",PD)
```

Scilab code Exa 2.14 Zener resitance

```
//EX2_14 PG-2.52
clc
Vzd=50e-3;//change in Vz
Izd=2.5e-3//change in Iz
Zz=Vzd/Izd;//zener resistance
printf("zener resistance Zz=%.0f ohm",Zz)
```

Scilab code Exa 2.15 voltage across the terminal

```
//EX2_15 PG-2.52
clc
rz=4;//zener diode resistance
Vz=5.1;
Iz=25e-3;
x=Iz*rz;
Vzz=Vz+x;//total terminal voltage across the diode
printf("Therefore total terminal voltage across the
    diode Vz''= %.1fV",Vzz)
```

Scilab code Exa 2.16 maximum zener current

```
//EX2_16 PG-2.52
clc
Vz=6.8//nominal voltage
Pd_max=500;//zener diode power loss in mW at 40
    degree celsius
D=2.6;//Derating factor in mW/degree celsius
T1=40;//Temperature in degree celsius
Izm=Pd_max/Vz;
printf("    At T=40 degree celsius maximum zener
    current Izm=%.2f mA\n\n",Izm)
T2=100;//Temperature in degree celsius
delta_T=T2-T1;//change in temperature
```

```
Pd_derated=Pd_max-D*delta_T
Izm=Pd_derated/Vz;
printf("    At T=100 degree celsius maximum zener
    current Izm=%.3f mA\n\n",Izm)
```

Scilab code Exa 2.17 Zener diode current and power dissipation

```
//EX2_17 PG-2.52
clc
printf("Refer to the figure-2.51 shown\n")
printf(" We apply KVL across the circuit \n")
printf(" Therefore -Iz*R1-Vz+Vin = 0 \n")
Vin=15;
Vz=5.2;
R1=670;
Iz=(Vin-Vz)/R1;//zener diode current
Iz=Iz*1e3;//in mA
Pd=Vz*Iz;//power dissipation
printf(" \n zener diode current Iz:%f mA\n power
    dissipation :%f mW ",Iz,Pd)
```

Chapter 3

Diode applications

Scilab code Exa 3.1 RMS value of current output voltage efficiency

```
//EX3_1 pg-3.14
clc
disp("Refer to the circuit diagram shown in figure
   -3.7")
Rf=75;//diode forward resistance
Rl=10e3;//load resistance
Rs=10;//transformer secondary resistance
Ep=230;//rms value of primary voltage
N2byN1=1/3;//turns ratio
Es=Ep*N2byN1;//rms value of secondary voltage
Esm=sqrt(2)*Es;//peak value of secondary voltage
Im=Esm/(Rs+Rf+Rl);
Im=Im*1e3;
printf("\n Therefore peak value of current is %.2f
   mA \n",Im)
Idc=Im/%pi;
printf("\n Average current is %.3f mA \n",Idc)
Irms=Im/2;//for half wave rectifier
printf("\n rms current is %.3f mA \n",Irms)
Idc1=Idc*1e-3;
Edc=Idc1*Rl;
```

```
printf("\n DC output voltage is %.2f V \n",Edc)
Pdc=Edc*Idc1;//Dc output power
Irms1=Irms*1e-3;
Pac=Irms1^2*(Rs+Rf+Rl);//AC output power
%n=Pdc/Pac*100;//efficiency
printf("\n Efficiency is %.2f %% \n",%n)
disp("Ripple factor for half wave rectifier is 1.21")
```

Scilab code Exa 3.2 Dc output voltage

```
//EX3_2 PG-3.15
clc
disp("Refer to the figure-3.8 shown")
Vin=0;//cut-in voltage for an ideal diode is zero
Rf=0;//forward resistance for an ideaal diode is zero
disp("For an ideal diode")
Vm=15;
Vdc=-Vm/%pi;
printf("\n DC output voltage is %.2f V\n ",Vdc)
disp("-ve sign indicates that voltage is negative wrt ground")
disp("For a silicon diode Vin=0.7V ")
Vin=0.7;
Vdc=-(Vm-Vin)/%pi;
printf("\n DC output voltage is %.2f V\n ",Vdc)
disp("-ve sign indicates that voltage is negative wrt ground")
```

Scilab code Exa 3.3 Dc voltage and load current

```
//EX3_3 PG-3.16
```

```
clc
Rl=1;//load resistance in kohm
Vm=10;
disp("For an ideal diode Vin= 0V")
Vin=0;//for ideal diode
Rf=0;//for ideal diode
Edc=Vm/%pi;
Idc=Edc/Rl;
printf("\n Dc voltage is %.2f V and load current is
   %.2f mA \n",Edc,Idc)
disp("For a silicon diode Vin=0.7 V")
Vin=0.7;
Edc=(Vm-Vin)/%pi;
Idc=Edc/Rl
printf("\n Dc voltage is %.2f V and load current is
   %.2f mA \n",Edc,Idc)
```

Scilab code Exa 3.4 Average output voltage

```
//EX3_4 Pg3.16
clc
Esm=300;//peak rms voltage
Edc=Esm/%pi;//average output voltage
printf("\n average output voltage is %.3f V",Edc)
```

Scilab code Exa 3.5 1Power 2Percentage regulation 3efficiency of rectification and

```
//EX3_5 PG-3.28
clc
Es=30;//rms voltage
Rf=2;
Rs=8;
Rl=1e3;//in kohm
```

```
Esm=sqrt(2)*Es;//peak value of voltage
Im=Esm/(Rf+Rl+Rs);//peak value of current
Idc=2*Im/%pi;//average current for full wave rectifier
P=Idc^2*Rl;
printf(" Power delivered to the load is %.3f W \n",P)
Vdc_noload=2*Esm/%pi;
printf("\n Vnl=%f \n",Vdc_noload)
Vdc_fullload=Idc*Rl;
printf("\n Vfl=%.2f \n",Vdc_fullload)
%reg=(Vdc_noload-Vdc_fullload)/Vdc_fullload*100;
printf("\n percentage regulation is %f %% \n",%reg)
disp("Efficiency of rectification =(DC output)/(AC output)")
x=(1+(Rf+Rs)/Rl)^(-1);
n=8/%pi^2*x*100;
printf("\n Efficiency of rectification is %.1f %% \n",n)
Irms=Im/sqrt(2);
AC_rating=Es*Irms;
TUF=P/AC_rating;
printf("\n TUF of secondary is %.3f",TUF)
//The exact answer for percentage regulation is 1% not .97% as shown in the book ....
//because in the book it has rounded off the value 27.009 to 27 only
```

Scilab code Exa 3.6 Peak to peak value of output voltage

```
//EX3_6 PG-3.29
clc
disp("Refer to the figure -3.16 shown")
Rl=100;
Idc=1;//maximum value of DC current in each diode
```

```
disp("We know Idc=Im/pi for each diode")
Vm=%pi*Rl*Idc;
Vp=2*Vm;
printf("\n maximum peak to peak voltage is %.2f V" ,
   Vp)
```

Scilab code Exa 3.7 1no load Dc voltage 2output voltage at 100mA 3percentage regul

```
//EX3_7 PG-3.30
clc
Rf=1;
Es=10;
Rs=5;
Esm=sqrt(2)*Es;
Edc_nl=2*Esm/%pi;
printf("Therefore o load DC output voltage is %.4f V
    \n",Edc_nl)
Idc=100e-3;
disp("We know that Idc=2Im/pi and Im=Esm/(Rf+rs+rl)"
   )
Im=Idc*%pi/2;
Rl=Esm/Im-Rf-Rs;//load resistance
Edc_ol=Idc*Rl;//DC output voltage at 100mA
printf("\n DC output voltage at 100mA is %.4f V \n "
   ,Edc_ol)
%reg=(Edc_nl-Edc_ol)/Edc_ol*100;
printf("\n percentage regulation is %.2f %%",%reg)
```

Scilab code Exa 3.8 Input power from the transformer

```
//EX3_8 PG-3.31
clc
Pdc=500;//for half wave rectifier
```

```
%n=40.6;//maximum efficiency for half wave rectifier
disp(" We know that efficiency=Pdc/Pac ")
disp("for Half wave rectifier ")
Pac=Pdc/%n*100;
printf("\n AC input power is %.3f W \n",Pac)
disp("for Full wave rectifier ")
%n=81.2;//maximum efficiency for full wave rectifier
Pac=Pdc/%n*100;
printf("\n AC input power is %.3f W \n",Pac)
```

Scilab code Exa 3.9 1Average voltage 2efficiency

```
//EX3_9 PG-3.32
clc
Rf=10;
Rl=100;
Es=30;//transformer rms voltage
Esm=sqrt(2)*Es;//peak value of the voltage
Im=Esm/(Rf+Rl);
Idc=2*Im/%pi;
Edc=Idc*Rl;
printf("Average voltage is %.2f V \n",Edc)
Pdc=Idc^2*Rl;
Irms=Im/sqrt(2);//rms value of the current
Pac=Irms^2*(Rf+Rl);
%n=Pdc/Pac*100;
printf("\n Efficiency is %.2f %%",%n)
```

Scilab code Exa 3.10 Load current and rms value of input current

```
//EX3_10 PG-3.32
clc
Es=100;//rms value of current
```

```
Rf=50;//forward resistance
Rl=950;
Rs=0;//resistance of the transformer secondary which
    is assumed to be 0 ohm
Esm=sqrt(2)*Es;//peak value of the input voltage
Im=Esm/(Rs+Rl+Rf);
Irms=Im/sqrt(2);
printf("rms value of input current is %.4f A \n",
   Irms)
Idc=2*Im/%pi;
printf("\n load value of current is %.4f A",Idc)
```

Scilab code Exa 3.11 resistance of the load and efficiency

```
//EX3_11 PG-3.38
clc
Rf=0.1;
Idc=10;
Rs=0;
Es=30;//rms value of input voltage
Esm=sqrt(2)*Es;//peak value of the input voltage
Im=Idc*%pi/2;//DC output current
disp("We know that Im=Esm/(2Rf+Rs+Rl) for fullwave
   rectifier")
Rl=Esm/Im-2*Rf-0;
printf("\n Therefore load resistance is %.1f ohm \n"
   ,Rl)
Pdc=Idc^2*Rl;//Dc output power rating
Irms=Im/sqrt(2);//rms value of input current
Pac=Irms^2*(2*Rf+Rs+Rl);//Ac input power
%n=Pdc/Pac*100;//efficiency
printf("\n Therefore efficiency is %.2f %% \n",%n)
```

Scilab code Exa 3.12 1Dc load current 2Dc load voltage 3ripple voltage 4PIV rating

```
//EX3_12 PG-3.39
clc
Rl=5e3;
N1toN2=2;//transformer turns ratio
Ep=460;//rms value of primary voltage
Es=Ep/N1toN2;
Esm=sqrt(2)*Es;//peak value of the secondary voltage
Im=Esm/Rl;//We neglect forward diode resistance
Idc=2*Im/%pi;
printf("\n  Therefore DC load current is %f A \n",
   Idc)
Edc=Idc*Rl;
printf("\n  DC load voltage is %.3f V \n",Edc)
Rf=.482;//ripple factor for full bridge rectifier
Vrip=Rf*Edc;//ripple voltage
printf("\n  Therefore ripple voltage is %.1f V \n",
   Vrip)
disp(" Peak value of bridge rectifier=PIV rating of
   each diode")
PIV=Esm;
printf("\n  Therefore PIV rating of each diode is %
   .2f V",PIV)
```

Scilab code Exa 3.13 Load voltage and ripple voltage

```
//EX3_13 PG-3.40
clc
Ep=230; //rms value of primary voltage
N2toN1=1/15;//turns ratio
Rf=0;//diode is ideal
Rs=0;//transformer is ideal
Rl=50;//load resistance
Es=Ep*N2toN1;//rms vaue of primary voltage
```

```
Esm=sqrt(2)*Es;//peak value of input voltage
Im=Esm/(Rs+2*Rf+Rl);
Idc=2*Im/%pi;
Edc=Idc*Rl;//load voltage
printf("\n Therefore load voltage is %.1f V\n",Edc)
Rf=.482;//ripple factor for full wave rectifier
Vrip=Rf*Edc;//ripple voltage
printf("\n ripple voltage is %.4f V",Vrip)
//in the book the ripple voltage has been rounded
   off to
//6.6516V but the actual ans is 6.6539V
```

Scilab code Exa 3.14 `Average load current and efficiency`

```
//EX3_14 PG-3.40
clc
Rf=0;//diode forward resistance
Es=240;//rms value of supply voltage
Rl=48;//load resistance
Im=sqrt(2)*Es/(Rl+Rf);
Idc=2*Im/%pi;
printf(" Average load current is %.3f A \n",Idc)
Pdc=Idc^2*Rl;
Irms=Im/sqrt(2);//rms value of input current
Pac=Irms^2*Rl;
%n=Pdc/Pac*100;//efficiency
printf("\n Therefore efficiency is %.2f %% \n",%n)
```

Scilab code Exa 3.15 `1Dc output voltage 2ripple factor 3efficiency 4PIV`

```
//EX3_15 PG-3.41
clc
Esm=100;//peak value of supply voltage
```

```
Rf=25;//diode forward voltage
Rl=950;//load resistance
Rt=(2*Rf)+Rl;//total resistance
Im=Esm/Rt;
Idc=2*Im/%pi;
Edc=Idc*Rl;
printf("\n DC output voltage is %f V \n",Edc)
Irms=Im/sqrt(2);//rms value of input current
x=(Irms/Idc)^2-1;
Rf=sqrt(x);//ripple factor;
printf("\n Therefore ripple factor is %.4f \n",Rf)
Pdc=Idc^2*Rl;
Pac=Irms^2*(2*Rf+Rl);//Ac input power
%n=Pdc/Pac*100;//efficiency
printf("\n Therefore efficiency is %.2f %% \n",%n)
disp("Peak value of bridge rectifier=PIV rating of each diode")
PIV=Esm;
printf("\n Therefore PIV rating of each diode is %.2f V",PIV)
//In the book the answer for Edc=57.5985V which is wrong because they have taken
//Rf=50 ohm instead of 25 ohm as given in the question similarly
//the efficiency=73.3417% in the book is wrong
//the correct answer for efficiency is 80.97%
```

Scilab code Exa 3.16 `1Average Dc voltage 2Average Dc current 3frequency of output`

```
//EX3_16 PG-42
clc
Rl=100;//load resistance
Es=230;//rms value of input voltage
Rf=0;//ideal diode resistance
Rs=0;//neglecting transformer resistance
```

```
f=50;//frequency of the supply
Esm=sqrt(2)*Es;//peak value of the input voltage
Edc=2*Esm/%pi;//as Rf=0 ohm
printf("\n Average DC voltage is %.0f V \n",Edc)
Im=Esm/Rl;
Idc=2*Im/%pi;
printf("\n Therefore DC load current is %.2f A \n",
   Idc)
f=2*f;//frequency of output waveform
printf("\n frequency of output waveform is %.0f Hz",
   f)
```

Scilab code Exa 3.17 `1Ripple voltage 2ripple factor`

```
//EX3_17 PG-3.56
clc
Edc=12;//output DC voltage
f=50;//frequency
Idc=50e-3;
C=100e-6;//filter capacitor
Rl=2e3;//load resistance
Vr=Edc/(2*f*C*Rl);//rms value of ripple voltage
printf("\n rms value of ripple voltage is %.1f V \n
   ",Vr)
Rf=(4*sqrt(3)*f*C*Rl)^(-1)*100;//ripple factor
printf("\n ripple factor is %.2f %% \n",Rf)
```

Scilab code Exa 3.18 `1Dc output voltage 2rms value of of ripple voltage 3ripple fa`

```
//EX3_18 PG-3.56
clc
Es=120;//rms value of input voltage
f=50;//frequency
```

```
Idc=50e-3;
C=100e-6;//filter capacitor
Esm=sqrt(2)*Es;
Edc=Esm-Idc/(4*f*C);
printf("\n DC output voltage is %.4f V \n",Edc)
Vr=Idc/(4*sqrt(3)*f*C);//rms value of ripple voltage
printf("\n rms value of ripple voltage is %.4f V \n
   ",Vr)
 Rf=Vr/Edc;
printf("\n ripple factor is %f \n",Rf)
```

Scilab code Exa 3.19 Load voltage and ripple voltage

```
//EX3_19 PG-3.56
clc
Ep=230;//rms value of primary voltage
N1toN2=10;//turns ratio
Rl=50;//load resistance
f=50;//frequency of the supply in Hz
Es=Ep/N1toN2;//rms vaue of secondary voltage or the
   input voltage
Esm=sqrt(2)*Es;//peak value of input voltage
Im=Esm/(Rl);
Idc=2*Im/%pi;
Edc=Idc*Rl;//load voltage
printf("\n Therefore load voltage is %.1f V \n",Edc)
Rf=.48;//ripple factor for full wave rectifier
   without filter
Vrip=Rf*Edc;//ripple voltage
printf("\n Ripple factor is 0.48")
printf("\n ripple voltage is %.4f V \n",Vrip)
disp("If the capacitor filter is used then")
C=470e-6;//filter capacitor
Edc=Esm-Idc/(4*f*C);
printf("\n new DC load voltage is %.2f V \n",Edc)
```

```
Rf=((4*sqrt(3)*f*C*Rl))^(-1);// new ripple factor
Vrip=Rf*Edc;//new ripple voltage
printf("\n Ripple factor is %.4f",Rf)
printf("\n rms value of ripple voltage is %.4f V \n
    ",Vrip)
// in the book the new ripple factor is 3.256e-3
    which is wrong
//the actual answer is 0.1228 hence the new ripple
    voltage is 3.4544V
// not 0.09156V as shown in the book
```

Scilab code Exa 3.20 Value of capacitor C

```
//EX3_20 PG-3.57
clc
Rf=.01;//ripple factor in percentage
Rl=2;//load resistance in kohm
f=50;//frequency
disp("the %ripple factor=Rf=((4*sqrt(3)*f*C*Rl))
    ^(-1)*100 ")
C=((4*sqrt(3)*f*Rf*Rl))^(-1)*100;//filter capacitor
printf("\n the filter capacitor is %.3f mF",C)
```

Scilab code Exa 3.21 Ripple factor

```
//EX3_21 PG-3.58
clc
Idc=100e-3;//average current
C=500e-6;//filter capacitor
Esm=18;//peak voltage
f=50;//frequency of the supply in Hz
Edc=Esm-Idc/(4*f*C);
Rl=Edc/Idc;//load resistance
```

```
Rf=(4*sqrt(3)*f*C*Rl)^(-1)*100;//ripple factor
printf("\n ripple factor is %.2f %% \n",Rf)
```

Scilab code Exa 3.22 surge current

```
//EX3_22 PG-3.58
clc
f=50;//frequency
C=1000e-6;//filter capacitor
Rl=500;//load resistance
Vrms=120;//rms value of voltage
T1=1e-3;//conduction period of diode,T1=1ms
disp("conduction period of diode,T1=1ms")
Esm=sqrt(2)*Vrms;//peak value of input voltage
disp("Edc=Esm-Idc/(2*f*C) and Idc=Edc/Rl")
Edc=Esm/(1+(2*Rl*f*C)^(-1));//output dc voltage
Idc=Edc/Rl;
T=1/f;
//Idc*T=Ip*T1
//Ip is the surge current
Ip=Idc*T/T1;
printf("\n hence the diode should be rated for a
    minimum surge\n current of %.2f A \n",Ip)
```

Scilab code Exa 3.23 Rms value of ripple voltage and peak to peak voltage

```
//EX3_23 PG-3.59
clc
Rf=0.1;//riplle facto
Edc=10;
Vrip=Rf*Edc;//rms value of voltage
printf("\n rms value of ripple voltage is %.0f V \n
    ",Vrip)
```

```
Vp_p=2*sqrt(2)*Vrip;
printf("\n peak to peak voltage is %.4f V ",Vp_p)
```

Scilab code Exa 3.24 Dc output voltage and ripple factor

```
//EX3_24 PG-3.59
clc
Es=230;//rms value of input voltage
f=50;//frequency
Idc=50e-6;
Rl=100;//load resistance
C=1000e-6;//filter capacitor
Esm=sqrt(2)*Es;
Edc=2*Esm/%pi;
printf("\n Therefore DC output voltage is %.2f V \n"
    ,Edc)
Idc=Edc/Rl;
disp("if the capacitor filter C=1000e-6 is use then
    ")
Rf=(4*sqrt(3)*f*C*Rl)^(-1);//ripple factor
printf("\n ripple factor is %.4f \n",Rf)
Edc=Esm-Idc/(4*f*C);
printf("\n Therefore new DC load voltage is %.4f V \
    n",Edc)
```

Scilab code Exa 3.25 Value of the capacitance needed

```
//EX3_25 Pg-3.60
clc
Es=230;
f=50//frequency
Rf=.005;//ripple factor
Il=0.5;//average load current
```

```
Esm=sqrt(2)*Es;//peak value of input voltage
disp(" For a half wave rectifier Ripple factor=(2*
   sqrt(3)*f*C*Rl)^(-1)")
Edc=Esm/%pi//for half wave rectifier
Rl=Edc/Il;
C=(2*sqrt(3)*f*Rf*Rl)^(-1);//for half wave rectifier
C=C*1e3;
printf("\n  Therefore capacitance required is %.3f
   mF",C)
```

Scilab code Exa 3.26 Ripple factor

```
//EX3_26 PG-3.60
clc
Rl=1000;
C=500e-3
f=50;
Rf=(4*sqrt(3)*f*C*Rl)^(-1);//ripple factor
Rf=Rf*1e6;
printf("\n ripple factor is %.2f 10^(-6) \n",Rf)
```

Scilab code Exa 3.27 minimum value of capacitor

```
//EX3_27 PG-3.60
clc
Il=12e-3;//load current
Es=200;//rms voltage
Rf=0.02;//riplle factor
Esm=sqrt(2)*Es;//peak value of input voltage
Edc=2*Esm/%pi;
Idc=Il;
Rl=Edc/Idc;//load resistance
f=50;//frequency of the supply in Hz
```

```
disp(" For a half wave rectifier Ripple factor=1/(2*
    sqrt(3)*f*C*Rl)")
C=(4*sqrt(3)*f*Rf*Rl)^(-1);//filter capacitor
printf("\n  Therefore minimum value of capacitance
    required is %.3f microF",C*1e6)
//C=9.619 microF not 9.622 microF
```

Scilab code Exa 3.28 Value of the capacitor required

```
//EX3_28 PG-3.61
clc
Es=230;//rms voltage
f=50;
Il=10e-3;//load current
Rf=.01;//ripple factor
Esm=sqrt(2)*Es;//peak value of input voltage
Edc=2*Esm/%pi;//for full wave
Rl=Edc/Il;
C=(4*sqrt(3)*f*Rf*Rl)^(-1);//for full wave rectifier
C=C*1e6;
printf("\n Therefore capacitance required is %.2f
    microF",C)
```

Scilab code Exa 3.29 1Average Dc current 2Average Dc voltage 3Ripple voltage

```
//EX3_29 PG-3.61
clc
Rl=2e3;//load resistance
Es=200;//rms voltage
f=50;
Esm=sqrt(2)*Es;//peak value of input voltage
Rf=0;//ideal diodes
Rs=0;
```

```
Ism=Esm/(Rf+Rs+Rl);
Idc=2*Ism/%pi;
printf("\n Therefore Average DC load current is %.2f
    A \n",Idc)
Edc=Idc*Rl;
printf("\n Therefore average DC voltage is %.0f V \n
   ",Edc)
Rf=0.48;//ripple factor
Vrip=Rf*Edc;// ripple voltage
printf("\n rms value of ripple voltage is %.1f V \n
   ",Vrip)
disp("if a filter capacitor C=500 microF is used
   then")
C=500e-6;//capacitor filter
Rf=(4*sqrt(3)*f*C*Rl)^(-1);//for full wave rectifier
Vrip=Rf*Edc;//new ripple voltage
printf("\n rms value of new ripple voltage is %.4f V
    \n ",Vrip)
```

Scilab code Exa 3.30 Load regulation

```
//EX3_30 PG-3.67
clc
Vnl=10;//no load output voltage
Vfl=9.8;//full output voltage
LR=Vnl-Vfl;//load regulation
%LR=(Vnl-Vfl)/Vfl*100;//percentage load regulation
printf("\n percentage load regulation is +%.2f %%",
   %LR)
```

Scilab code Exa 3.31 Percentage load regulation

```
//EX3_31 PG-3.67
```

```
clc
LR=3e-3;//load regulation
Vnl=15;//no load voltage or maximum voltage
Vfl=Vnl-LR;//full load voltage
%LR=(Vnl-Vfl)/Vfl*100;//percentage load regulation
printf("\n percentage load regulation is +%.2f %%",
   %LR)
```

Scilab code Exa 3.32 Source regulation and percentage source regulation

```
//EX3_32 PG-3.67
clc
Vhl=10+.3;//high line voltage
Vll=10-.3;//low line voltage
SR=Vhl-Vll;//source regulation
Vnom=10;//nominal load voltage
%SR=SR/Vnom*100;//percentage source regulation
printf("\n percentage source regulation is %.0f %%",
   %SR)
```

Scilab code Exa 3.33 Range of input voltage

```
//EX3_33 PG-3.70
clc
disp("Refer to the figure -3.45 shown")
Vz=6.1;//zener voltage
Iz_min=2.5e-3;//minimum zener current
Iz_max=25e-3;//maximum zener current
rZ=0;//ideal zener diode
R=2.2e3;
Rl=1e3;//loadd resistance
Il=Vz/Rl;
//For minimum input voltage(Vin_min)
```

```
Iz=Iz_min
I=Iz_min+Il;
Vin_min=Vz+I*R;
printf("\n minimum input voltage(Vin_min) is %.2f V
   \n",Vin_min)
//For maximum input voltage(Vin_max)
I=Iz_max+Il;
Vin_max=Vz+I*R;
printf("\n maximum input voltage(Vin_max) is %.2f V
   \n",Vin_max)
printf("\n range of input voltage is from %.3f V to
   %.2f V \n",Vin_min,Vin_max)
```

Scilab code Exa 3.34 design of zener diode

```
//EX3_34 PG-3.70
clc
Vo=5;//output voltage
Vin_min=12-3;//min input voltage
Vin_max=12+3;//max input voltage
Iz_min=10e-3;//minimum zener current
Il=20e-3;//load current
Pz=500e-3;//Zener wattage
Vz=Vo;//zener voltage
disp("Step1 : Maximum power dissipation correesponds
    to Iz_max")
Iz_max=Pz/Vz;
printf("\n maximum current that should flow through
   the zener diode is %.1f A \n",Iz_max)
disp ("Step2 : We know that Il is constant")
//for Vin_max, Iz=Iz_max
I=Il+Iz_max;
Rmin=(Vin_max-Vz)/I;
printf("\n minimum resistance required is %.2f ohm \
   n",Rmin)
```

```
disp("Iz is maximum when R=Rminimum")
disp("Step3 : for calculation of Rmax I must be minimum ie I=Iz_min ")
I=Il+Iz_min
Rmax=(Vin_min-Vz)/I;
printf("\n maximum resistance required is %.2f ohm \n",Rmax)
printf("\n Thus R must be greater than %.2f ohm and less than \n %.2f ohm for proper regulation \n",Rmin,Rmax)
```

Scilab code Exa 3.35 `Minimum and maximum load current`

```
//EX3_35 PG-3.72
clc
disp("refer to the figure -3.47 shown")
Vz=10;//output voltage
Vin=20;//input voltage
Iz_max=25e-3;//maximum zener current
Iz_min=5e-3;//minimum zener current
R=300;
Rz=0;//zener resistance
I=(Vin-Vz)/R;
//for Il_min Iz=Iz_max
Il_min=I-Iz_max;//minimum load current
printf("\n minimum load current is %.2f mA \n",Il_min*1e3)
//for Il_max , Iz=Iz_min
Il_max=I-Iz_min;//maximum load current
printf("\n maximum load current is %.2f mA \n",Il_max*1e3)
Rl_min=Vz/Il_max;//minimum load resistance
printf("\n minimum load resistance is %.3f ohm \n",Rl_min)
// in the book in the question it given that Iz_max
```

```
=50mA
//but during the solution Iz_max is taken as 25mA I
    have taken Iz_max=25mA
// in this program
```

This code can be downloaded from the website wwww.scilab.in This code

can be downloaded from the website wwww.scilab.in

Scilab code Exa 3.38 Design of zener diode

```
//EX3_38  Pg-3.75
clc
Vo=5;
Il=20e-3;
Pz=500e-3;
Rl=Vo/Il;
Il_min=Il;//minimum load current
Il_max=Il;//maximum load current
Iz_max=Pz/Vo;//maximum zener current
Iz_min=5e-3;//minimum zener current
V=12;//input DC voltage
Vin_min=12-3;//min input voltage
Vin_max=12+3;//max input voltage
Rmax=(Vin_min-Vo)/(Il_max+Iz_min);
printf("\n maximum resistance required is %.0f ohm \
    n",Rmax)
Rmin=(Vin_max-Vo)/(Il_min+Iz_max);
printf("\n minimum resistance required is %.2f ohm \
    n",Rmin)
printf("\n So series resistance must be selected
    between %.2f ohm to %.0f ohm \n",Rmin,Rmax)
```

Scilab code Exa 3.39 Design of Voltage regulator

```
//EX3_39 Pg-3.76
clc
Vo=10;
Il_min=0;//mainimum load current
Il_max=10e-3;//maximum load current
Iz_max=50e-3;//maximum zener current
Iz_min=2e-3;//minimum zener current
Vin_min=20;//min input voltage
Vin_max=30;//max input voltage
Rl_min=Vo/Il_max;
Rmax=(Vin_min-Vo)/(Il_max+Iz_min);
printf("\n maximum resistance required is %.2f ohm \
    n",Rmax)
Rmin=(Vin_max-Vo)/(Il_min+Iz_max);
printf("\n minimum resistance required is %.0f ohm \
    n",Rmin)
printf("\n So series resistance must be selected
    between %.0f ohm to %.2f ohm \n",Rmin,Rmax)
```

This code can be downloaded from the website wwww.scilab.in

Scilab code Exa 3.40 series resistance and diode current

```
//EX3_40  Pg-3.76
clc
Vo=24;
Il_min=0;//minimum load current
Pz=600e-3;
Vin=32;//input voltage
```

```
Iz_max=Pz/Vo;
Rmin=(Vin-Vo)/(Il_min+Iz_max);
printf("\n minimum resistance required is %.0f ohm \
   n",Rmin)
printf("\n As Vin and Il are not changing R=Rmin=%.0
   f ohm\n is sufficient to work as a regulator\n",
   Rmin)
disp("For Rl=1200 ohm")
Rl=1200;
Il=Vo/Rl;
printf(" \n load current is: %.2f A \n",Il)
R=Rmin
It=(Vin-Vo)/R;
Iz=It-Il;
printf("\n zener current is :%.3f A \n",Iz)
printf(" As Iz=Iz_min=%.3f A, the circuit will work
   as a regulator",Iz)
```

This code can be downloaded from the website wwww.scilab.in

Scilab code Exa 3.41 Design of zener regulator

```
//EX3_41 PG-3.79
clc
Pd=400e-3;
Vz=8.1;//output voltage
Vo=Vz;
Zz=8;
Vin=15;
Izm=Pd/Vz;
Rmax=(Vin-Vz)/Izm;
printf("\n maximum series resistance is %.3f ohm \n"
   ,Rmax)
Iz_min=5e-3;//we select the minimum zener current
```

```
Il_max=Izm-Iz_min;//maximum load current
printf("\n maximum load current is %.7f A \n",Il_max)
Rl=Vz/Il_max;//load resistance
deltaVin=.1*Vin;//change in input voltage is equal to 10% of the original input voltage
R=Rmax;//series resistance
x=(Rl*Zz)/(Rl+Zz);
deltaVo=(deltaVin*x)/(R+x);
Sv=deltaVo/deltaVin;//voltage stability factor
printf("\n voltage stability factor  is %.3f \n",Sv)
SR=deltaVo/Vo*100;// line regulation for a 10% change in Vin
printf("\n line regulation is %.4f %% \n",SR)
deltaIL=Il_max;
y=(R*Zz)/(R+Zz)
deltaVo=deltaIL*y;
LR=deltaVo/Vo*100;//load regulation
printf("\n load regulation is %.4f %% \n",LR)
z=(Rl*Zz)/(Rl+Zz)
RR=z/(R+z);//ripple rejection ratio
printf("\n ripple rejection ratio is %.3f \n",RR)
Ro=y;//output resistance
printf("\n output resistance is %.3f ohm \n",Ro)
```

This code can be downloaded from the website wwww.scilab.in

Chapter 4

Transistors

Scilab code Exa 4.1 Base current

```
//EX-4_1 PG-4.11
clc
Ie=12e-3;//emitter current
Ic=Ie/1.02;//collector current
Ib=Ie-Ic;//base current
printf("\n Therefore base current is %.0f microA \n"
   ,Ib*1e6)
```

Scilab code Exa 4.2 Dc current gain and voltage gain

```
//EX4_2 Pg-4.14
clc
printf("When beta_dc=50 ")
beta_dc=50;
alpha_dc=beta_dc/(1+beta_dc);
printf("\n alpha_dc=%.4f \n\n",alpha_dc)
printf(" When beta_dc=190 ")
beta_dc=190;
```

```
alpha_dc=beta_dc/(1+beta_dc);
printf("\n alpha_dc=%.4f \n\n",alpha_dc)
printf(" When alpha_dc=0.995 ")
alpha_dc=0.995;
beta_dc=alpha_dc/(1-alpha_dc);
printf("\n beta_dc=%.0f \n\n",beta_dc)
printf(" When alpha_dc=0.9765 ")
alpha_dc=0.9765;
beta_dc=alpha_dc/(1-alpha_dc);
printf("\n beta_dc=%.2f \n",beta_dc)
```

Scilab code Exa 4.3 Dc voltage gain and current gain

```
//EX4_3,Pg-4.15
clc
Ib=20e-6;
Ie=0.0064;
beta_dc=Ie/Ib-1;
alpha_dc=beta_dc/(1+beta_dc);
printf("\n beta_dc=%.0f \n \n alpha_dc=%f \n",
   beta_dc,alpha_dc)
Ic=beta_dc*Ib;//collector current
printf("\n collector current is %.3f mA \n",Ic*1e3)
```

Scilab code Exa 4.4 current gain and emitter current

```
//EX4_4 PG4.15
clc
Icbo=5e-6;//leakage current
Ic=20e-3;//collector current
Ie=Ic/0.996;//Ic=o.996Ie
alpha_dc=Ic/Ie;
printf("\n alpha_dc=%.3f \n",alpha_dc)
```

```
disp("We know that Ic=alpha_dc*Ie+Icbo ")
disp("Therefore emitter current is Ie=(Ic-Icbo)/
   alpha_dc")
Ie=(Ic-Icbo)/alpha_dc;//emitter current
Ie=Ie*1e3;
printf("\n emitter current is %.2f mA",Ie)
```

Chapter 5

Biasing methods

Scilab code Exa 5.1 Calculation of Ib Ic and Vce and the region of operation

```
//EX5_1 PG 5.4
clc
disp("  Refer to the figure -5.5 shown")
disp("i) When Rb=300 kohm")
disp("   Base emitter junction is not reverse biased
    ")
disp("   Assume transistor is operating in active
   region")
Rb=300e3;
Rc=2e3;//collector resistance
Vcc=10;//supply voltage
Vbe=0.7;//base emitter voltage
disp("   We know that Vcc=Ib*Rb+Vbe ")
Ib=(Vcc-Vbe)/Rb;//since Vcc=Ib*Rb+Vbe
printf("\n     base current Ib is: %.2f microA \n",Ib
   *1e6)
Beta=100;
Ic=Beta*Ib;//colector current in active region
printf("\n     Collector current is %.1f mA \n",Ic*1
   e3)
disp("   Applying KVL around collector loop ie Vcc=
```

```
    Ic*Rc+Vce")
Vce=Vcc-Ic*Rc;//since Vcc=Ic*Rc+Vce
printf("\n    Now Vce= %.1f V \n\n",Vce)
printf("    Since Vce=3.8 V collector to base
    junction is reverse biased and we can say \n
    that our assumption that transistor is in active
    region is justified")
printf("\n\n ii)When Rb=150 kohm\n\n")
disp("    base emitter junction is not reverse biased
    ")
Rb=150e3;
disp("    assume transistor is operating in active
    region")
disp("    Applying KVL around base loop ie Vcc=Ib*Rb+
    Vbe")
Ib=(Vcc-Vbe)/Rb;//since Vcc=Ib*Rb+Vbe
printf("\n    base current Ib is: %.0f microA \n",Ib
    *1e6)
Ic=Beta*Ib;//colector current in active region
printf("\n    Collector current is %.1f mA \n",Ic*1
    e3)
disp("    Applying KVL around collector loop ie Vcc=
    Ic*Rc+Vce")
Vce=Vcc-Ic*Rc;//since Vcc=Ic*Rc+Vce
printf("\n    Therefore Vce= %.1f V \n\n",Vce)
printf("    Collector voltage Vce has to be +ve or
    zero but Vce=-2.4 V hence \n    transistor is not
     in active region but it is in saturation region\
    n\n ")
Vbe_sat=0.8;//base saturation voltage
Vce_sat=0.2;//collector saturation voltage
disp("    Applying KVL around base loop ie Vcc=Ib*Rb+
    Vbe_sat")
Ib=(Vcc-Vbe_sat)/Rb;//since Vcc=Ib*Rb+Vbe_sat
printf("\n    base current Ib is: %.0f microA \n",Ib
    *1e6)
disp("    Applying KVL around collector loop ie Vcc=
    Ic*Rc+Vce_sat")
```

```
Ic=(Vcc-Vce_sat)/Rc//since Vcc=Ic*Rc+Vce_sat
printf("\n     Collector current is %.1f mA \n",Ic*1e3)
printf("     To justify transistor is in saturation then \n     Ib must be greater than (Ic/Beta)")
x=Ic/Beta
 printf("\n\n     Now Ib=%.0fnmicroA \n \n    (Ic/Beta)=%.0f microA \n",Ib*1e6,x*1e6)
if (Ib>x) then//x=(Ic/Beta)
disp("    Hence transistor in saturation region is satisfied ")
end
```

Scilab code Exa 5.2 Minimum and maximum values of Ic and Vce

```
//EX5_2 PG-5.6
clc
Vbe=0.7;//base emitter voltage for silicon
Vcc=12;//supply voltage
Rb=150e3;
Rc=2e3
hFE_min=50;//minimum voltage gain
hFE_max=60;//maximum voltage gain
Ib=(Vcc-Vbe)/Rb;//since Vcc=Ib*Rb+Vbe
printf("\n base current is %.8f A \n",Ib)
printf("\n for hFE_min=50")
Ic=hFE_min*Ib
printf("\n Ic=%.3f A \n",Ic*1e3)
Vce=Vcc-Ic*Rc
printf(" Vce=%.4f V \n",Vce)
printf("\n\n for hFE_max=60")
Ic=hFE_max*Ib
printf("\n Ic=%.2f A \n",Ic*1e3)
Vce=Vcc-Ic*Rc
printf(" Vce=%.0f V \n",Vce)
```

Scilab code Exa 5.3 Calculation of Q point values

```
//EX5_3 PG5.8
clc
disp("Refer to the figure -5.8 shown")
Vbe=0.7;//base emitter voltage for silicon
Vcc=12;//supply voltage
Rb=100e3;
Rc=10e3;
Beta=100;//voltage gain
Ib=(Vcc-Vbe)/((1+Beta)*Rc+Rb);//since Vcc=Ib*Rb+Vbe
printf("\n base current is %.2f microA \n",Ib*1e6)
Ic=Beta*Ib
printf("\n Ic=%.3f mA \n",Ic*1e3)
Vce=Vcc-(Ib+Ic)*Rc
printf("\n Vce=%.4f V \n",Vce)
```

Scilab code Exa 5.4 Calculation of minimum and maximum values of Ic and Vce

```
//EX5_4 PG-5.9
clc
Vbe=0.7;//base emitter voltage for silicon
Vcc=12;//supply voltage
Rb=150e3;
Rc=2e3
hFE_min=50;
hFE_max=60;
printf("i) for hFE_min=50")
Beta=hFE_min;//minimum voltage gain
Ib=(Vcc-Vbe)/((1+Beta)*Rc+Rb);//since Vcc=Ib*Rb+Vbe
printf("\n    base current is %.2f microA \n",Ib*1e6
  )
```

```
Ic=Beta*Ib
printf("     Ic=%.3f mA \n",Ic*1e3)
Vce=Vcc-(Ib+Ic)*Rc
printf("     Vce=%.3f V \n",Vce)
printf("\n     for hFE_max=60")
Beta=hFE_max;//maximum voltage gain
Ib=(Vcc-Vbe)/((1+Beta)*Rc+Rb);//since Vcc=Ib*Rb+Vbe
printf("\n     base current is %.2f microA \n",Ib*1e6
   )
Ic=Beta*Ib
printf("     Ic=%.3f mA \n",Ic*1e3)
Vce=Vcc-(Ib+Ic)*Rc
printf("     Vce=%.3f V \n",Vce)
```

Scilab code Exa 5.5 Calculation of Vce and Ic

```
//EX5_5 PG-5.11
clc
disp("Refer to the figure -5.13 shown")
Vbe=0.7;//base emitter voltage for silicon
Vcc=10;//supply voltage
R1=10e3;
Rc=1e3;
R2=5e3;
Re=500;
Beta=100;//voltage gain
Vb=R2*Vcc/(R1+R2);//base voltage
printf("\n Vb=%.2f V\n",Vb)
Ve=Vb-Vbe;//emitter voltage
printf("\n Ve=%.2f V \n",Ve)
Ie=Ve/Re;
printf("\n Ie=%.2f mA \n",Ie*1e3)
Ib=5.26e-3/(1+Beta);//Ie=0.00526 A=5.26 mA
printf("\n Ib=%.2f microA \n",Ib*1e6)
Ic=Beta*Ib;
```

```
printf("\n Ic=%.3f mA \n",Ic*1e3)
disp("We apply KVL to the collector circuit")
disp("Vcc−Ic*Rc−Vce−Ie*Re=0")
Vce=Vcc-Ic*Rc-Ie*Re;//since Vcc−Ic*Rc−Vce−Ie*Re=0
printf("\n Vce=%.2f V \n",Vce)
```

Scilab code Exa 5.6 Determination of Ic Ve and Vce

```
//EX5_6 PG−5.12
clc
printf("Refer to the figure −5.13 shown in the
    question no 5.5\n")
//We must find the value of Ic,Ve,Vce using exact
    analysis
Vbe=0.7;//base emitter voltage for silicon
Vcc=10;//supply voltage
R1=10e3;
Rc=1e3;
R2=5e3;
Re=500;
Beta=100;//voltage gain
Vt=R2*Vcc/(R1+R2);//thevenin's voltage
printf("\n Vt=%.2f V\n",Vt)
Rb=R1*R2/(R1+R2);
printf("\n Rb=%.0f ohm \n",Rb)
Ib=(Vt-Vbe)/(Rb+(1+Beta)*Re);
printf("\n Ib=%.3f microA \n",Ib*1e6)
Ic=Beta*Ib;
printf("\n Ic=%.2f mA \n",Ic*1e3)
Vce=Vcc-Ic*Rc-(Ic+Ib)*Re;//since Vcc−Ic*Rc−Vce−Ie*Re
   =0
printf("\n Vce=%.3f V \n",Vce)
//the ans for Ve in the book is 2.648V whereas in
   output it is 2.638V because
// in the book the values has been rounded off so
```

```
    that the final answer is
//2.648V same is the case for Rb,Ib andIc
```

Scilab code Exa 5.7 Minimum and maximum values of Ic and Vce

```
//EX5_7 PG-5.13
clc
Vbe=0.7;//base emitter voltage for silicon
Vcc=12;//supply voltage
R1=10e3;
R2=2e3
Re=470;
Rc=2e3
hFE_min=50;
hFE_max=60;
Vb=R2*Vcc/(R1+R2);//base voltage
printf("\n Vb=%.0f V\n",Vb)
Ve=Vb-Vbe;//emitter voltage
printf("\n Ve=%.1f V \n",Ve)
Ie=Ve/Re;
printf("\n Ie=%.2f mA \n",Ie*1e3)
printf("\ni) for hFE_min=50")
Beta=hFE_min;
Ib=Ie/(Beta+1);
Ic=Beta*Ib;
printf("\n    Ic=%.3f mA \n",Ic*1e3)
Vce=Vcc-Ic*Rc-Ve;
printf("    Vce=%.3f V \n",Vce)
disp("ii)for hFE_max=60")
Beta=hFE_max;
Ib=Ie/(Beta+1);
Ic=Beta*Ib;
printf("     Ic=%.3f mA \n",Ic*1e3)
Vce=Vcc-Ic*Rc-Ve;
printf("     Vce=%.2f V \n",Vce)
```

Scilab code Exa 5.8 Design of a fixed biased circuit

```
//EX5_8 PG-5.16
clc
disp("refer to the figure -5.17 shown")
Vbe=0.7;//base emitter voltage for silicon
Vcc=10;//supply voltage
Beta=100;//voltage gain
Vce=5;//colector to emitter voltage
Ic=5e-3;//collector current
disp("We apply KVL to the collector circuit ie Vcc-
    Vce-Ic*Rc=0")
Rc=(Vcc-Vce)/Ic;//since Vcc-Vce-Ic*Rc=0
printf("\n Rc=%.0f kohm \n",Rc*1e-3)
Ib=Ic/Beta;//base current
printf("\n Ib=%.2f microA \n",Ib*1e6)
disp("We apply KVL to the base circuit ie Vcc-Vbe-Ib
    *Rb=0")
Rb=(Vcc-Vbe)/Ib;//since Vcc-Vbe-Ib*Rb=0
printf("\n Rb=%.0f kohm \n",Rb*1e-3)
disp("the standard value of Rb=200k ohm")
```

This code can be downloaded from the website wwww.scilab.in

Scilab code Exa 5.9 Design of Collector to base bias circuit

```
//EX5_9 PG-5.17
clc
disp("refer to the figure -5.19 shown")
Vbe=0.7;//base emitter voltage for silicon
```

```
Vcc=16;//supply voltage
Beta=100;//voltage gain
Vce=5;//colector to emitter voltage
Ic=5e-3;//collector current
Ib=Ic/Beta;//base current
printf("\n Ib=%.0f microA \n",Ib*1e6)
Rc=(Vcc-Vce)/(Ic+Ib);//since Vcc-Vce-Ic*Rc=0
printf("\n Rc=%.3f kohm \n",Rc*1e-3)
disp("Rc=2 kohm standard value")
disp("We apply KVL to the input circuit ie Vce-Vbe-
   Ib*Rb=0")
Rb=(Vce-Vbe)/Ib;//since Vce-Vbe-Ib*Rb=0
printf("\n Rb=%.0f kohm \n",Rb*1e-3)
disp("the standard value of Rb=91 kohm")
```

This code can be downloaded from the website wwww.scilab.in

Scilab code Exa 5.10 Design of voltage divider bias circuit

```
//EX5_10 PG-5.18
clc
disp("Refer to the figure-5.20 shown")
Vbe=0.7;//base emitter voltage for silicon
Vcc=12;//supply voltage
Beta=100;//voltage gain
Vce=5;//colector to emitter voltage
Ve=3;//assumption
Ic=3e-3;//collector current
Ib=Ic/Beta;//base current
printf("\n Ib=%.0f microA \n",Ib*1e6)
Ie=Ic+Ib;//emitter current
printf("\n Ie=%.2f mA \n",Ie*1e3)
Re=Ve/Ie;
printf("\n Re=%.0f ohm \n",Re)
```

```
printf(" the standard value of Re=910 ohm")
Re=910;//standard value
Ve=Ie*Re;
printf("\n\n Ve=%.3f V \n",Ve)
Rc=(Vcc-Ve-Vce)/Ic
printf("\n Rc=%.0f ohm \n",Rc)
printf(" the lower side standard value is selected
   to reduce Ic*Rc and increase Vce ")
Vb=Ve+Vbe
printf("\n\n Therefore Vb=%.5f V \n",Vb)
I=10*Ib
printf("\n I=%.1f mA \n",I*1e3)
R2=Vb/I;
printf("\n R2=%.0f ohm \n",R2)
printf(" the standard value of R2=11 kohm\n")
disp("the lower side standard is selected to satisfy
    I>=10*Ib")
R2=11e3;
I=Vb/R2;
printf("\n I=%.4f mA \n",I*1e3)
R1=(Vcc-Vb)/(I+Ib)
printf("\n R1=%.3f kohm \n",R1*1e-3)
printf(" the standard value of R1=22kohm\n")
disp("The lowest standard value is selected to
   satisfy I>=10*Ib")
```

This code can be downloaded from the website wwww.scilab.in

Scilab code Exa 5.11 Stability factor

```
//EX5_11 PG-5.26
clc
//for base bias take the figure as shown the
   Example5.1
```

```
Beta=100;
S=1+Beta;//stability factor
printf("\n For base bias: stability factor=%.0f \n",
   S)
//for collector to base bias the figure as shown the
    Example5.3
Beta=100;
Rc=10e3;
Rb=100e3
S=(1+Beta)/(1+Beta*(Rc/(Rc+Rb)));
printf("\n For collector to base bias: stability
   factor=%.0f \n",S)
//for voltage divider  bias take the figure as shown
    the Example5.5
Re=500;
R1=10e3;
R2=5e3;
Rb=R1*R2/(R1+R2);//R1 and R2 are in parallel
S=(1+Beta)/(1+Beta*(Re/(Re+Rb)));
printf("\n For voltage divider bias: stability
   factor=%.2f \n",S)
```

Scilab code Exa 5.12 Resistance Rb

```
//EX5_12 PG-5.28
clc
disp("Refer to the figure -5.25 shown")
Icbo1=2e-6;//at a temperature T1=25 degree celsius
Vbb=5;
Vbe=0.1
disp("Icbo doubles for every 10 degree Celsius")
T1=25;//temperature in degree celsius
T2=80;//temperature in degree celsius
Icbo2=Icbo1*2^((T2-T1)/10);//at a temperature T2=80
   degree celsius
```

```
printf("\n Therefore Icbo2=%.2f microA \n",Icbo2*1e6)
disp("Apply KVL to the base circuit Vbb=Vbe+Icbo2*Rb we get")
Rb=(Vbb-Vbe)/Icbo2;
printf("\n Rb=%.3f kohm \n",Rb*1e-3)
```

Scilab code Exa 5.13 Calculation of R1 and R3

```
//EX5_13 PG-5.29
clc
disp("Refer to the figure -5.26 shown")
Vcc=15;//supply voltage
Beta=100;//voltage gain
Vbe=0.6;//base emitter voltage
Ic=2e-3
Vce=3;
R4=600;
R2=10e3;
Ib=Ic/Beta;
Ie=Ic+Ib;//collector current
printf("\n Apply KVL to the collector side Vcc=Ic*R3+Vce+Ie*R4")
R3=(Vcc-Vce-Ie*R4)/Ic;//since Vcc=Ic*R3+Vce+Ie*R4
printf("\n Therefore R3=%.3f kohm \n",R3*1e-3)
printf("\n Apply KVL to the base side I*R2=Vbe+Ie*R4")
I=(Vbe+Ie*R4)/R2;//since I*R2=Vbe+Ie*R4
printf("\n Therefore I=%.4f mA \n",I*1e3)
printf("\n Apply KVl to the potential divider side we get Vcc=(I+Ib)*R1+I*R2")
R1=(Vcc-I*R2)/(I+Ib);//since Vcc=(I+Ib)*R1+I*R2
printf("\n therefore R1=%.3f kohm \n",R1*1e-3)
```

Scilab code Exa 5.14 Calculation of Re Vce and stability factor

```
//EX5_14 PG-5.30
clc
disp("Refer to the figure -5.27 shown")
Vcc=12;//supply voltage
Beta=100;//voltage gain
Vbe=0.7;//base emitter voltage
Ic=2e-3;
Ib=Ic/Beta;
R1=50e3;
R2=5e3;
Rc=2e3;
disp("apply KVl to the potential divider side we get
    Vcc-(I+Ib)*R1-I*R2=0 we get")
I=(Vcc-R1*Ib)/(R1+R2);//since Vcc-(I+Ib)*R1-I*R2=0
Vb=R2*I;
Ie=Ib+Ic;
Re=(Vb-Vbe)/(Ib+Ic);//Vb=Vbe+Re*Ie
printf("\n Re=%.2f ohm \n",Re)
disp("Apply KVL to the collector side Vcc-Ic*Rc-Vce+
   Ie*Re=0")
Vce=Vcc-Ic*Rc-Ie*Re;//since Vcc-Ic*Rc-Vce+Ie*Re=0
printf("\n Therefore Vce=%.1f V \n",Vce)
Rb=R1*R2/(R1+R2);//R1 and R2 are in parallel
S=(1+Beta)/(1+Beta*(Re/(Re+Rb)));
printf("\n stability factor=%.3f \n",S)
```

Scilab code Exa 5.15 Determination of percent change in Q values

```
//EX5_15 PG-5.31
clc
```

```
disp("Refer to the figure -5.28 shown")
Vbe=0.7;
Rb=100e3;
Rc=600;
Vcc=12;//supply voltage
T1=25;//temperature in degree celsius
T2=75;//temperature in degree celsius
disp("at T1=25 degree celsius , applying KVL to the
    base circuit we get")
disp("Vcc-Ib*Rb-Vbe=0")
Beta=100;//voltage gain at T1=25 degree celsius
Ib=(Vcc-Vbe)/Rb;//since Vcc-Ib*Rb-Vbe=0
Ic=Beta*Ib;
printf("\n Therefore Ic=%.1f mA \n",Ic*1e3)
disp("Apply KVL to the collector side Vcc-Ic*Rc-Vce
    =0")
Vce=Vcc-Ic*Rc;//since Vcc-Ic*Rc-Vce=0
printf("\n Therefore Vce=%.2f V \n",Vce)
Beta=125;//voltage gain at T1=25 degree celsius
Ic1=Beta*Ib;
Vce1=Vcc-Ic1*Rc;//since Vcc-Ic*Rc-Vce=0
printf("\n At 75 degree celsius Vce=%.3f\n",Vce1)
Ic_change=(Ic1-Ic)*100/Ic;//percentage in Ic
printf("\n change in Ic=%.0f %% (an increase)\n",
    Ic_change)
Vce_change=(Vce1-Vce)*100/Vce;//percentage in Vce
printf("\n change in Vce=%.2f %% (a decrease) \n",
    Vce_change)
```

Chapter 6

Power control Devices

Scilab code Exa 6.1 Finding of various ratings of SCR

```
//EX6_1 PG-6.15
clc
Es=20;//rms value of the supply voltage
Ep=sqrt(2)*Es;//peak value
printf("\n Therefore peak value of the input voltage is %.4f V \n",Ep)
printf("\n Therefore forward and reverse blocking voltge of SCR>%.4f V \n",Ep)
Rl=30;//load resistance
Ih=5e-3;//holding current
Vtm=1.7;//state voltage drop
Vl_peak=Ep-Vtm;
printf("\n Vl_peak=%.4f V \n",Vl_peak)
Il_peak=Vl_peak/Rl;
printf("\n Il_peak=%.4f A \n",Il_peak)
Il_rms=Il_peak/2;
printf("\n rms value of current flowing through the SCR is %.3f A \n",Il_rms)
//SCR current rating should be greater than Il_rms
Es_off=Vtm+Ih*Rl;//voltage which cause SCR to switch off
```

```
printf("\n voltage which cause SCR to switch off is %.2f V \n",Es_off)
```

Scilab code Exa 6.2 Designing values of R1 R2 and R3

```
//EX6_2 PG-6.18
clc
disp("Refer to the figure -6.19 shown")
Es=25;//rms value of the supply voltage
Vd1=0.7;//diode drop
Vg=0.75;//SCR triggering voltage
alpha1=10;//minimum phase angle
alpha2=90;//maximum phase angle
Ep=sqrt(2)*Es;//peak value
Es1=Ep*sind(alpha1)
Es2=Ep*sind(alpha2)
Vt=Vd1+Vg;//voltage across R3
printf("\n Position of R2 at top to trigger at 10 degree ")
Vr1=Es1-Vt;//since Vt=Vr2+Vr3
//I1_min>>Ig ie 200 microA
I1_min=1e-3;
R1=Vr1/I1_min;
printf("\n Therefore R1=%.4f kohm \n",R1*1e-3)
printf(" We use R1=4.7 kohm standard value \n\n")
R1=4.7e3;//standard value R1
x=Vt/I1_min;//x=R2+R3
printf(" Position of R2 at bottom to trigger at 90 degree ")
Vr3=Vt;
I1=Es2/(R1+x);
R3=Vr3/I1;
printf("\n Therefore R3=%.2f ohm \n",R3)
printf(" We use R3=270 ohm standard value \n")
R3=270;//standard value R3
```

```
R2=x-R3;//since x=R2+R3
printf("\n Therefore R2=%.2f kohm \n",R2*1e-3)
printf(" We use the pot=1.5 kohm standard value for
   precise judgement")
```

Scilab code Exa 6.3 Frequency of oscillations

```
//EX6_3 PG-6.33
clc
Rt=5;//resistance in kohm
Ct=0.1;//capacitance in micro farad
n=0.58;//standoff ratio
T=Rt*Ct*log(1/(1-n));//time period in seconds
fo=1/T;//frequency of oscillations
fo=fo;
printf("\n Therefore frequency of oscillation is %.3
   f kHz \n",fo)
```

Scilab code Exa 6.4 Frequency of oscillations

```
//EX6_4 PG-6.34
clc
Vbb=20;
Ct=0.1e-6;//capacitance
Rt=10e3;//resistance
Vv=1.5;//valley potential
Vd=0.7;//cut in voltage of diode
n=0.6;//stand off ratio
Vp=n*Vbb+Vd;
disp("We know that Vp=Vbb(1-exp(-T/(Rt*Ct)))")
x=(Vp-Vv)/Vbb;//x=(1-exp(-T/(Rt*Ct))
y=1-x;//y=exp(-T/(Rt*Ct)
z=Rt*Ct;
```

```
T=-log(y)*z;//time period
fo=1/T;//frequency of oscillations
fo=fo*1e-3;
printf("\n Therefore frequency of oscillation is %.3
   f kHz \n",fo)
```

Scilab code Exa 6.5 Range of Rt

```
//EX6_5 PG-6.34
clc
Vbb=30;
Vv=0.8;//valley potential
Iv=15e-3
Vd=0.7;//cut in voltage of diode
n=0.33;//stand off ratio
Vp=18;
Ip=35e-6;
printf(" For turn ON")
printf("\n  Rt<(Vbb-Vp)/Ip")
Rt=(Vbb-Vp)/Ip;
printf("\n  therefore Rt<%.0f Ohm \n",Rt)
printf(" \n  For turn OFF")
printf("\n  Rt>(Vbb-Vv)/Iv")
Rt1=(Vbb-Vv)/Iv
printf("\n  therefore Rt>%.0f Ohm \n",Rt1)
printf("\n  So range of Rt is %.3f kohm< Rt <%.2f
   kohm \n",Rt1*1e-3,Rt*1e-3)
```

Chapter 7

Junction field effect transistors

Scilab code Exa 7.1 Drain current

```
//Ex7_1 PG-7.13
clc
Idss=10e-3;
Vgs_off=-4;
printf("For Vgs =0 V")
Vgs=0;
Id=Idss;
printf("\n Idss=%.0f mA \n",Id*1e3)
printf("\n For Vgs =-1 V")
Vgs=-1;
Id=Idss*(1-Vgs/Vgs_off)^2;
printf("\n Idss=%.3f mA \n",Id*1e3)
printf("\n For Vgs =-4 V")
Vgs=-4;
Id=Idss*(1-Vgs/Vgs_off)^2;
printf("\n Idss=%.0f A \n",Id)
//In the book this example is mention as Example-4_3
//but it is the first example in this chapter so I'
    ve taken this as Example-7_1
```

Scilab code Exa 7.2 1maximum and minimum od output voltage 2voltage gain

```
//Ex7_2 PG-7.17
clc
Vi=50e-3//input supply
Rd=5e3;
Yfs_max=4000e-6;
Yfs_min=1000e-6;
disp(" For Yfs_max=4000e-6")
Id_delta=Yfs_max*Vi;
printf("\n Change in Id is +/- %.1f mA \n",Id_delta*1e3)
Vo=Id_delta*Rd;//output voltage
Av=Vo/Vi;//voltge gain
printf("\n Voltage gain is %.0f \n",Av)
disp(" For Yfs_min=1000e-6")
Id_delta=Yfs_min*Vi;
printf("\n Change in Id is +/- %.2f mA \n",Id_delta*1e3)
Vo=Id_delta*Rd;//output voltage
Av=Vo/Vi;//voltge gain
printf("\n Voltage gain is %.0f \n",Av)
//In the book this example is mention as Example-7_1
//but  it is the second example in this chapter so I've taken this as Example-7_2
```

Chapter 8

Amplifiers

Scilab code Exa 8.1 Gain in dB scale

```
//Ex8_1 PG-8.3
clc
Av=100;//voltage gain
G=20*log10(Av);//gain in decibel
printf("\n Therefore gain in decibel=%.0f dB \n",G)
```

Scilab code Exa 8.2 Maximum voltage gain

```
//Ex8_2 PG-8.5
clc
Av=200;//gain at cut-off frequencies
Avm=Av*sqrt(2);//maximum voltage gain
printf("\n Therefore maximum voltage gain=%.2f \n",
   Avm)
```

Scilab code Exa 8.3 Gain of an amplifier

```
//Ex8_3 PG-8.6
clc
Amid=100;//mid-band gain
f1=1e3;//loer cut-off frequency
f=20;//frquency at which the gain of the amplifier
    should be found
A=Amid/sqrt(1+(f1/f)^2)
printf("\n Therefore  the gain of the amplifier at f
     = 20Hz is %.0f \n",A)
```

Scilab code Exa 8.4 Gain of an amplifier

```
//Ex8_4 PG-8.7
clc
G=200;//3dB gain
f2=20e3;//higher cut-off frequency
Amid=G*sqrt(2);//mid-band gain
f=100e3;//frquency at which the gain of the
    amplifier should be found
A=Amid/sqrt(1+(f/f2)^2)
printf("\n Therefore  the gain of the amplifier at f
    =100kHz is %.2f \n",A)
//in the  book the answer for the gain is 115.47
    which is wrong
//the corect answer is 55.47
```

Chapter 9

Oscillators

Scilab code Exa 9.1 Frequency of oscillations

```
//EX9_1 PG-9.13
clc
R=4.7e3;//each resistance of the RC phase shift
    oscillator
C=0.47e-6;//each capacitance of the RC phase shift
    oscillator
f=1/(2*%pi*sqrt(6)*R*C);
printf("\n Therefore frequency of oscillation is %.3
    f Hz \n",f)
```

Scilab code Exa 9.2 Design of an RC phase shift oscillator

```
//EX9_2 PG-9.13
clc
f=900e3;//frequency of oscillation
C=1e-12;//each capacitance of the RC phase shift
    oscillator
R=1/(2*%pi*sqrt(6)*f*C);
```

```
printf("\n each resistance of the RC phase shift
    oscillator is %.3f kohm \n",R*1e-3)
G=29;//opamp gain Rf/R1=29
R1=1e3;
printf("\n R1=%.0f kohm \n",R1*1e-3)
Rf=G*R1;
printf("\n Rf=%.0f kohm \n",Rf*1e-3)
disp("the design circuit is shown ")
```

This code can be downloaded from the website wwww.scilab.in

Scilab code Exa 9.3 estimation of R and C

```
//EX9_3 PG-9.14
clc
f=1e3;//frequency of oscillation
C=0.1e-6;//We choose the value of each capacitance
    of the RC phase shift oscillator
R=1/(2*%pi*sqrt(6)*f*C);
printf("\n  each resistance of the RC phase shift
    oscillator is %.3f ohm \n",R)
disp(" The standard value of R=680 ohm")
```

Scilab code Exa 9.4 Frequency of oscillations

```
//EX9_4 PG-9.14
clc
R=5e3;//each resistance of the RC phase shift
    oscillator
C=0.1e-6;//each capacitance of the RC phase shift
    oscillator
f=1/(2*%pi*sqrt(6)*R*C);
```

```
printf("\n Therefore frequency of oscillation is %.3
    f Hz \n",f)
```

Scilab code Exa 9.5 Value of capacitor C

```
//EX9_5 PG-9.20
clc
L=100e-6;
f=500e3;
disp("We know that for a colpitts oscillator f=1/(2*
    %pi*sqrt(L*Ceq))")
Ceq=1/(f^2*4*%pi^2*L)
Ceq1=Ceq*1e9;
printf("\n Ceq = %.5f nF \n",Ceq1)
//C1=C2=C
C=Ceq1*2;//Ceq=(C*C)/(C+C)
printf("\n Therefore C = %.5f nF \n",C)
```

Scilab code Exa 9.6 Frequency of oscillation

```
//EX9_6 PG-9.20
clc
L=50e-6;
C1=150e-12;
C2=1.5e-9;
Ceq=(C1*C2)/(C1+C2);
f=1/(2*%pi*sqrt(L*Ceq));
f=f*1e-6;
printf("\n Therefore frequency of oscillation is %.3
    f MHz \n",f)
```

Scilab code Exa 9.7 value of inductance L

```
//EX9_7 PG-9.21
clc
C=1000e-12;
C1=C;
C2=C;
f=500e3;
Ceq=(C1*C2)/(C1+C2);
L=1/(4*%pi^2*f^2*Ceq);//since f=1/(2*%pi*sqrt(L*Ceq)
    );
L=L*1e6;
printf("\n Therefore L=%.3f micro H \n",L)
```

Scilab code Exa 9.8 designing the value of inductor L

```
//EX9_8 PG-9.21
clc
C1=100e-12;
C2=50e-12;
f=10e6;
Ceq=(C1*C2)/(C1+C2);
L=1/(4*%pi^2*f^2*Ceq);//f=1/(2*%pi*sqrt(L*Ceq));
L=L*1e6;
printf("\n Therefore inductor L = %.1f microH \n",L)
```

Scilab code Exa 9.9 Frequency of oscillation

```
//EX9_9 PG-9.24
clc
L1=0.5e-3;
L2=1e-3;
C=0.2e-6;
```

```
Leq=L1+L2;//total inductance  for Hartley oscillator
f=1/(2*%pi*sqrt(Leq*C));
printf("\n Therefore frequency of oscillation is %.f Hz \n",f)
//there is a slight difference between the answer given in the book
//and the and output in the book they have taken the approximate value
```

Scilab code Exa 9.10 Range for the capacitor C

```
//EX9_10 PG-9.24
clc
L1=2e-3;
L2=20e-6;
Leq=L1+L2;//total inductance  for Hartley oscillator
fmax=2050e3;//maximum frequency
fmin=950e3;//minimum frequency
printf("For f=fmax=2050kHz")
f=fmax;
C=1/(4*%pi^2*f^2*Leq);//since f=1/(2*%pi*sqrt(Leq*C));
C=C*1e12
printf("\n C=%.2f pF \n",C)
printf("\n For f=fmin=950kHz")
f=fmin;
C1=1/(4*%pi^2*f^2*Leq);//since f=1/(2*%pi*sqrt(Leq*C));
C1=C1*1e12
printf("\n C=%.2f pF \n",C1)
printf("\n Hence C must be varied between %.2f pF and %.2f pF \n",C,C1)
```

Scilab code Exa 9.11 Range of capacitor C

```
//EX9_11 PG-9.25
clc
L1=20e-6;
L2=2e-3
Leq=L1+L2;//total inductance  for Hartley oscillator
fmax=2.5e6;//maximum frequency
fmin=1e6;//minimum frequency
printf("For f=fmax=2.5MHz")
f=fmax;
C=1/(4*%pi^2*f^2*Leq);//since f=1/(2*%pi*sqrt(Leq*C));
C=C*1e12
printf("\n C=%.3f pF \n",C)
printf("\n For f=fmin=1MHz")
f=fmin;
C1=1/(4*%pi^2*f^2*Leq);//since f=1/(2*%pi*sqrt(Leq*C));
C1=C1*1e12
printf("\n C=%.3f pF \n",C1)
printf("\n Hence C must be varied between %.3f pF and %.2f pF \n",C,C1)
```

Scilab code Exa 9.12 1series resonant frequency 2parallel resonant frequency 3perc

```
//EX9_12 PG-9.32
clc
L=0.4;
C=0.085e-12;
R=5e3;
Cm=1e-12;
f=1/(2*%pi*sqrt(L*C));//series resonant frequency for crystal oscillator
printf("\n series resonant frequency for crystal
```

```
oscillator fs=%.3f MHz \n",f*1e-6)
Ceq=C*Cm/(C+Cm);
fp=1/(2*%pi*sqrt(L*Ceq));//parallel resonant frequency for crystal oscillator
printf("\n parallel resonant frequency for crystal oscillator=%.3f MHz \n",fp*1e-6)
%increase=(fp-f)/f*100;
printf("\n increase in parallel frequency fp=%.3f %% \n",%increase)
w=2*%pi*f;
Q=w*L/R;//Q factor
printf("\n Therefore Q factor=%.3f \n",Q)
//in the book fs=0.856MHz is wrong,correct answer is fs=.863MHz
//in the book %increase=5.023% is wrong the correct answer is %increase=4.163%
//in the Q=430.272 which is wrong the correct answer is Q=433.861
```

Scilab code Exa 9.13 `Series and parallel resonant frequency`

```
//EX9_13 PG-9.32
clc
C=0.01e-12;
Cm=2e-12;
L=2;
R=2e3;
fs=1/(2*%pi*sqrt(L*C));//series resonant frequency for crystal oscillator
printf("\n series resonant frequency for crystal oscillator fs=%.3f MHz \n",fs*1e-6)
Ceq=C*Cm/(C+Cm);
fp=1/(2*%pi*sqrt(L*Ceq));//parallel resonant frequency for crystal oscillator
printf("\n parallel resonant frequency for crystal
```

```
oscillator=%.3f MHz \n",fp*1e-6)
```

Chapter 10

Introduction to Operational Amplifiers

Scilab code Exa 10.1 Output voltage

```
//EX10_1 PG-10.7
clc
disp("Refer to the figure -10.5 shown")
V1=300e-6;
V2=240e-6;
Vd=V1-V2//differential mode voltage
Vc=(V1+V2)/2;//common mode voltage
Ad=5000;//differential gain
printf("\n when CMRR=100")
CMRR=100;//common mode rejection ratio
Ac=Ad/CMRR;
printf("\n common mode gain Ac=%.f \n",Ac)
Vo=Ad*Vd+Ac*Vc;//output voltage
printf(" output voltage is Vo=%.1f mV \n",Vo*1e3)
printf("\n when CMRR=100000")
CMRR=1e5;//common mode rejection ratio
Ac=Ad/CMRR;
printf("\n common mode gain Ac=%.2f \n",Ac)
Vo=Ad*Vd+Ac*Vc;//output voltage
```

```
printf(" output voltage is Vo=%.4f mV \n",Vo*1e3)
```

Scilab code Exa 10.2 Input bias current and input offset current

```
//EX10_2 PG-10.17
clc
Ib1=18;//in microA
Ib2=22;//in microA
Ib=(Ib1+Ib2)/2;//input bias current
Ib=Ib
printf("\n input bias current= %.0f microA \n",Ib)
Iios=(Ib1-Ib2);//input offset current
Iios=abs(Iios);
Iios=Iios
printf("\n input offset current= %.0f microA \n",
   Iios)
```

Scilab code Exa 10.3 Values of two input bias current

```
//EX10_3 PG-10.17
clc
Iios=20;//Input offset current in nA
Ib=60;//Input bias current in nA
//Iios=Ib1-Ib2=20
//Ib=(Ib1+Ib2)/2=60
//ie Ib=(Ib1+Ib2)=120
disp(" Iios=Ib1-Ib2=20")
disp(" ie Ib=(Ib1+Ib2)/2=60 ie Ib=(Ib1+Ib2)=120")
a=[1 -1;1 1];//coefficient of Ib1 and Ib2 for Iios
   and Ib
b=[20 ;120];//value of Iios and Ib
x=inv(a)*b
disp(" values of Ib1 andIb2 are")
```

```
printf("  %.0f nA ",x)
```

Scilab code Exa 10.4 Voltage gain

```
//EX10_4 PG-10.35
clc
disp("Refer to the figure -10.32 shown")
//the circuit is an inverting amplifier
R1=10e3;
Rf=47e3;//feedback resistance
A=-Rf/R1;//gain of an inverting amplifier
printf("\n the gain is %.1f (inverting amplifier) \n
   ",A)
```

Scilab code Exa 10.5 Output voltage and waveforms

```
//EX10_5 PG-10.35
clc
R1=10e3;
Rf=50e3;//feedback resistance
Vcc=12;//supply voltage in volts
A=-Rf/R1;//gain of an inverting amplifier
A=abs(A);//magnitude of the gain
printf("If Vm=0.5V then")
Vm=0.5;//peak value of the input voltage
Vo=A*Vm;//output voltage
printf("\n peak value of the output voltage Vo=%.1f
   V \n",Vo)
printf("\n if Vm=5V then")
Vm=5;//peak value of the input voltage
Vo1=A*Vm;//output voltage
printf("\n peak value of the output voltage Vo=%.0f
   V \n",Vo1)
```

```
disp("but the opamp output saturates at +/- 12V
    hence portion above +12V and")
printf(" below -12V will be clipped off.So 25V \n
    peak output is not practically possible it will
    show upto +/- 12V")
clf()

subplot(2,1,1)          //input voltage Vin=0.5V peak
x=0:%pi/100:9
 y=0.5*sin(x)
 plot(x,y)
xtitle("(Vin)m=0.5V" ,"time","Input voltage Vin= 0.5
    V peak")
 xgrid(color("grey"));

subplot(2,1,2)           //output voltage Vo=2.5V peak
 x=0:%pi/100:9
 y=-Vo*sin(x)          //output is inverted
 plot(x,y)
xtitle("(Vin)m=0.5V" ,"time","Output voltage(
    inverted) Vo=2.5V peak")
 xgrid(color("grey"));

xset('window',1)
clf()

subplot(2,1,1)          //input voltage Vin=5V peak
x=0:%pi/100:9
 y=5*sin(x)
 plot(x,y)
xtitle("(Vin)m=5V" ,"time","Input voltage Vin=5V
    peak")
 xgrid(color("grey"));

subplot(2,1,2)          //output voltage Vo=25V peak
    but clipped at + or-12V
 x=0:%pi/100:9
 y=-Vo1*sin(x)         //output is inverted
```

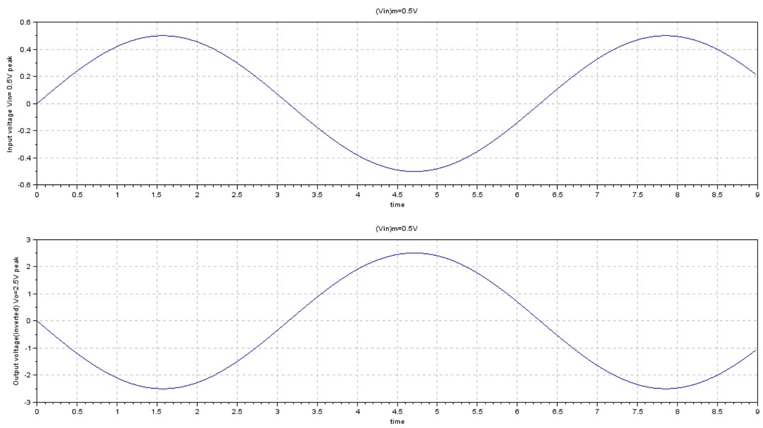

Figure 10.1: Output voltage and waveforms

```
y(find(y > 12)) =  12;
y(find(y < -12))= -12;
 plot(x,y,style=3)
xtitle("(Vin)m=5V" ,"time","Output voltage(inverted)
    Vo=12V clipped")
 xgrid(color("grey"));
```

Scilab code Exa 10.6 `Reading of output voltage on the voltmeter`

```
//EX10_6 PG-10.36
clc
R1=10e3;
Rf=47e3;//feedback resistance
Vcc=12;//supply voltage
A=-Rf/R1;//gain of an inverting amplifier
A=abs(A);//magnitude of the gain of an inverting
    amplifier
```

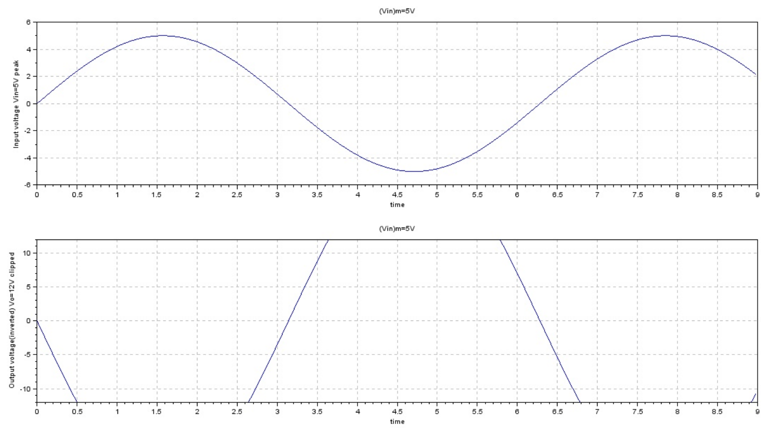

Figure 10.2: Output voltage and waveforms

```
Vin=2;//peak to peak input voltage
Voo=Vin*A;//peak to peak output voltage
//the AC voltmeter measures the rms value
Vo=Voo/2;//peak output voltage
Vrms=Vo/sqrt(2);//rms value of the output voltage
printf("\n Reading on the AC voltmeter is %.4f V \n"
    ,Vrms)
```

Scilab code Exa 10.7 `Voltage gain and output waveforms`

```
//EX10_7 PG-10.37
clc
R1=1e3;
Rf=10e3;//feedback resistance
Vin=100e-3;//input voltage peak to peak
A=-Rf/R1;//gain of an inverting amplifier
Vo=A*Vin;
printf("\n peak to peak value of the output voltage
    Vo = %.0f V \n",Vo)
printf("\n voltage gain of the inverting amplifier
    Af = %.0f \n ",A)
```

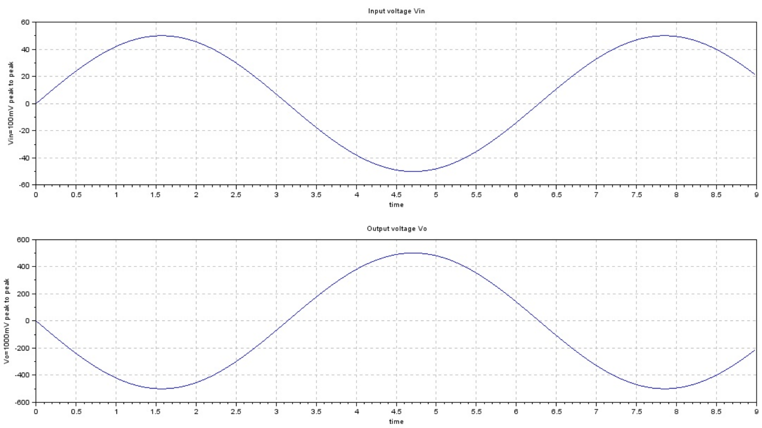

Figure 10.3: Voltage gain and output waveforms

```
//plotting of the waveforms
clf()

 subplot(2,1,1)     //input voltage Vin=0.5V peak
x=0:%pi/100:9
 y=50*sin(x)
 plot(x,y)
xtitle("Input voltage Vin " ,"time","Vin=100mV peak to peak")
 xgrid(color("grey"));
subplot(2,1,2)      //output voltage Vo=2.5V peak
 x=0:%pi/100:9
 y=-500*sin(x)
 plot(x,y)
xtitle("Output voltage Vo" ,"time","Vo=1000mV peak to peak")
 xgrid(color("grey"));
```

Scilab code Exa 10.8 `Feedback resistor Rf`

```
//EX10_8 PG-10.40
clc
A=61;//gain required for the non inverting amplifier
R1=1e3;
printf("Refer to the figure-10.36 shown\n")
printf(" \n The gain of the non inverting amplifier
    is A=1+Rf/R1")
//the gain of the non inverting amplifier is A=1+Rf/
    R1
x=A-1;//x=Rf/R1
Rf=x*R1;
printf("\n\n Therefore feedback resistance Rf=%.0f
    kohm \n",Rf*1e-3)
```

Scilab code Exa 10.9 `Gain and output voltage`

```
//EX10_9 PG-10.40
clc
R1=1e3;
Rf=10e3;//feedback resistance
A=1+Rf/R1;//gain of a non-inverting amplifier
printf("Gain is %.0f\n",A)
disp("For Vin =0.5V ")
Vin=0.5;//input voltage
Vo=A*Vin;
printf(" Output voltage Vo=%.1f V \n",Vo)
disp("For Vin =-3V ")
Vin=-3;//input voltage
Vo=A*Vin;
printf(" Output voltage Vo=%.1f V \n",Vo)
printf("\n but Vo=-33V is not possible. Output will
    saturate at -12V \n")
printf(" And the remaining portion will be clipped
    from output.")
```

Scilab code Exa 10.10 `Closed loop gain and feedback resistance`

```
//EX10_10 PG-10.48
clc
printf("Refer to the figure-10.43 shown\n ")
//A is grounded so B is virtual ground
//Vb=Va=0
Vb=0;
R1=1e3;
R2=5e3;
R3=5e3;
R4=100;
printf("\n Vb=Va=0  .................(1)")
printf("\n Vb=0 \n I1=(Vin-Vb)/R1=Vin/R1 \n I1=(Vb-Vx)/R2=-Vx/R2 \n \n   Vin/Rf=-Vx/R2")
printf("\n =>Vx=-R2/R1*Vin........(2) \n \n now Vx=I2*R4 and (I1-I2)=(Vx-Vo)/R3 \n")
printf(" =>I2=Vx/R4 and I1-Vx/R4=(Vx-Vo)/R3 \n Therefore \n Vin/R1-Vx/R4=(Vx-Vo)/R3 ....")
printf(".......using I1=Vin/R1 \n Vin/R1-Vx(1/R4+1/R3)=-Vo/R3 \n \n ")
printf("Vin/R1-(-R2/R1)*Vin*(1/R4+1/R3)=-Vo/R3 ........using (2) \n")
printf(" Vin*(1/R1+R2/R1*(1/R4+1/R3))=-Vo/R3 \n \n ")
printf("Vin=-(R1*R4/(R3*R4+R2*R3+R2*R4))*Vo \n \n")
printf(" Acl=Vo/Vin=-(R3*R4+R2*R3+R2*R4)/(R1*R4) \n")
Acl=-(R3*R4+R2*R3+R2*R4)/(R1*R4);
printf("\n closed loop gain Acl=%.0f \n",Acl)
Acl=abs(Acl);
Rf=R1*Acl;//equivalent feedback resistance
printf("\n equivalent feedback resistance Rf= %.0f kohm ",Rf*1e-3)
```

This code can be downloaded from the website wwww.scilab.in

Scilab code Exa 10.11 `Output voltage Vo`

```
//EX10_11 PG-10.50
clc
disp("Refer to the figure-10.45 and figure-10.45(a)
   shown")
//the circuit is a non inverting amplifier
Vin=10;//input voltage
//opamp input current is zero...
R1=10e3;//resistance connected to the -ve terminal
   of the amplifier
R2=1e3;//resistance connected to the +ve terminal of
    the amplifier to the input voltage
R3=1e3;////resistance connected to the +ve terminal
   of the amplifier to the gound
Rf=50e3;//feedback resistance
I=Vin/(R2+R3);
Vb=I*R3;
V0=(1+Rf/R1)*Vb;//output voltage
printf("\n output voltage Vo=%.f V \n",V0)
```

Scilab code Exa 10.12 `Otuput voltage Vo`

```
//EX10_12 PG-10.51
clc
disp("Refer to the figure-10.46 and figure-10.46(a)
   shown")
disp("We split the circuit of the figure-10.46 as
   shown in figure-10.46(a)" )
```

```
disp("For first stage the circuit is a non−inverting amplifier")
Rf=10e3;//feedback resistance for the first stage of the circuit
R1=100e3;//value of R1 for the first stage of the circuit
V1=poly(0,'V1');//V1=Vin
Vo1=(1+Rf/R1)*V1;
disp(Vo1,"Therefore Vo1 =")

printf("\n For the second stage we use superposition principle.\n We use each input at one time\n")
disp("First we assume Vo1 is active and V2 is grounded as shown in figure10.46(b)")
Rf1=100e3;//feedback resistance for figure−10.46(b)
R11=10e3;//value of R1 for figure1−10.46(a)
Vo_=-Rf1/R11*Vo1;//when V2=0V as shown in figure−10.46(b)
disp(Vo_,"Therefore Vo_ =")

disp("then we assume V2 is active and Vo1 is grounded as shown in figure10.46(c)")
V2=poly(0,'V2')
Vo__=(1+Rf1/R11)*V2;//when Vo1=0V as shown in figure−10.46(c),it is a non inverting amplifier
disp(Vo__,"Therefore Vo__ =")
printf("\n   Therefore output voltage Vo = Vo_ + Vo__= 11V2−11V1 ")
disp("  =>Vo=11(V2−V1)")
```

This code can be downloaded from the website wwww.scilab.in This code

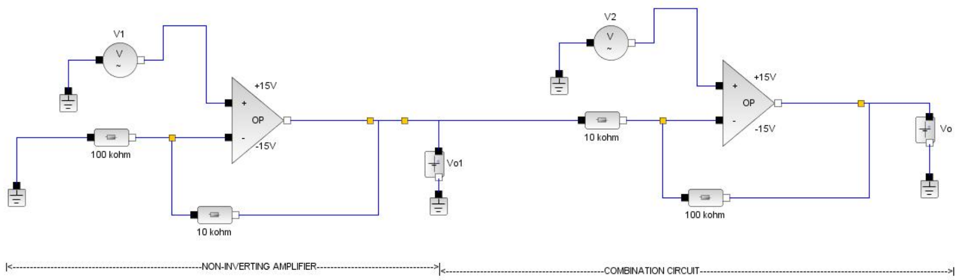

Figure 10.4: Otuput voltage Vo

can be downloaded from the website wwww.scilab.in

Scilab code Exa 10.13 `Output voltage Vo`

```
//EX10_13 PG-10.62
clc
disp("Refer to the figure -10.55 and shown")
R1=1e3;
Rf=1e3;//feedback resistance
Vin1=2;
Vin2=1;
Vin3=4;
Vout=-(Vin1+Vin2+Vin3)*Rf/R1
printf("\n output voltage Vout=%.f V \n",Vout)
```

Scilab code Exa 10.14 `Design of Scaling adder`

```
//EX10_14 PG-10.62
clc
printf("\n    Vo = -(3V1 + 4V2 + 5V3)\n\n")
Rf=120;//we assume feedback resistance to be equal to 120kohm
R1=Rf/3;//Rf/R1=3 given
```

```
R2=Rf/4; //Rf/R2=4 given
R3=Rf/5; //Rf/R3=5 given
printf("    R1= %.0fkohm   R2= %.0fkohm  R3 = %.0fkohm\
    n",R1,R2,R3)
printf("\n    The circuit design is shown")
```

This code can be downloaded from the website wwww.scilab.in

Scilab code Exa 10.15 `Relation between Rf and R`

```
//EX10_15 PG-10.63
clc
printf("\n      Refer to the figure -10.57 shown")
printf("\n      Output in terms of Va is Vo = (1+Rf/R
    )*Va.....(1)")
printf("\n\n      (V1-Va)/R +(V2-Va)/R +(V3-Va)/R = 0"
    )
printf("\n\n      Rearranging for Va we get Va = (V1+
    V2+V3)/R*(R/3)")
printf("\n      Putting the value of Va in (1) we get
    ")
printf("\n\n      Vo = (1+Rf/R)*(V1+V2+V3)/3")
printf("\n      But from the figure Vo = V1+V2+V3 \n\
    n      Therefore")
printf("    (V1+V2+V3) = (1+Rf/R)*(V1+V2+V3)/3")
printf("\n\n      Therefore (1+Rf/R) = 3")
printf("\n    => Rf=2R")
```

Scilab code Exa 10.16 `Output voltage`

```
//EX10_16 PG-10.64
clc
```

```
disp("Refer to the figure -10.58 and figure -10.58(a)
   shown")
R1=1e3;
R2=R1;
Rf=5e3;//feedback resistance
R=1e3;//resistance connected to the inverting
   terminal
V1=1;//first input voltage at the non inverting
   terminal
V2=3;//second input voltage at the non inverting
   terminal
Vb=(V1*R2+V2*R1)/(R1+R2);//voltage at the non
   inverting terminal
Vo=(1+Rf/R)*Vb;//output voltage
printf("\n Therefore output voltage is Vo=%.0f V \n"
   ,Vo)
//alternatively we can find the output voltage by
   the following equation
Vo1=(R2*(R+Rf))/(R*(R1+R2))*V1+(R1*(R+Rf))/(R*(R1+R2
   ))*V2
printf("\n Vo=%.0f \n",Vo1)
```

Scilab code Exa 10.17 Output voltage

```
//EX10_17 PG-10.65
clc
disp("Refer to the figure -10.59 shown")
Rf=10e3;//feedback resistance
R1=10e3;
R2=20e3;
R3=30e3;
R4=40e3;
V1=-1;//first input voltage at the inverting
   terminal
V2=2;//second input voltage at the  inverting
```

```
terminal
V3=3;//third input voltage at the inverting terminal
V4=-2;//fourth input voltage at the inverting terminal
Vo=-(Rf/R1*V1+Rf/R2*V2+Rf/R3*V3+Rf/R4*V4);//output voltage
printf("\n Therefore output voltage is Vo=%.1f V \n",Vo)
```

Scilab code Exa 10.18 `Output voltage`

```
//EX10_18 PG-10.66
clc
Rf=1e6;//feedback resistance
R1=200e3;
R2=250e3;
R3=500e3;
V1=-2;//first input voltage at the inverting terminal
V2=2;//second input voltage at the  inverting terminal
V3=11;//third input voltage at the inverting terminal
Vo=-(Rf/R1*V1+Rf/R2*V2+Rf/R3*V3);//output voltage
printf("\n Therefore output voltage is Vo=%.0f V \n",Vo)
//in the book the output Vo=-20V if the value of V3=11V
//but in the question the value of V3=1V so
//I have taken V3=11V so that the Vo=-20V
```

Scilab code Exa 10.19 `Output voltage`

```
//EX10_19 PG-10.66
clc
Rf=60e3;//feedback resistance
R1=10e3;
R2=20e3;
R3=30e3;
V1=-1;//first input voltage at the inverting
    terminal
V2=-2;//second input voltage at the  inverting
    terminal
V3=3;//third input voltage at the inverting terminal
Vo=-(Rf/R1*V1+Rf/R2*V2+Rf/R3*V3);//output voltage
printf("\n Therefore output voltage is Vo= %.0f V \n
    ",Vo)
```

Scilab code Exa 10.20 Design of adder circuit

```
//EX10_20 PG-10.67
clc
printf("\n    Vo = -(0.1V1 + 0.5V2 + 20V3)\n\n")
Rf=10;//we assume feedback resistance to be equal to
     10kohm
R1=Rf/0.1;//Rf/R1=0.1 given
R2=Rf/0.5;//Rf/R2=0.5 given
R3=Rf/20;//Rf/R3=20 given
printf("    R1= %.0f kohm   R2= %.0f kohm R3 = %.0f
    ohm\n",R1,R2,R3*1e3)
printf("\n    The circuit design is shown")
```

This code can be downloaded from the website wwww.scilab.in

Scilab code Exa 10.21 Design of an Opamp circuit

```
//EX10_21 PG-10.70
clc
printf("\n    Vo = (2V1 - 3V2 + 4V3 - 5V4)\n\n")
printf("    The positive terms and negative terms can
    be added separately\n\n")
printf("\n    Vo1 = (2V1 + 4V3 )\n")
Rf1=100;//we assume feedback resistance to be equal
    to 100kohm
R1=Rf1/2;//Rf/R1=2 given
R3=Rf1/4;//Rf/R3=4 given
printf("    Therefore R1= %.0fkohm    R3 = %.0fkohm\n"
    ,R1,R3)

printf("\n    Vo2 = -(3V2 + 5V4)\n")
Rf2=120;//we assume feedback resistance to be equal
    to 120kohm
R2=Rf2/3;//Rf/R2=3 given
R4=Rf2/5;//Rf/R4=5 given
printf("    Therefore R2= %.0fkohm    R4 = %.0fkohm\n"
    ,R2,R4)
printf("\n    The output voltage is Vo = Vo2-Vo1 = (2
    V1 - 3V2 + 4V3 - 5V4)")
printf("\n    The circuit design is shown")
printf("\n    For the subtractor we use R = 100kohm
    ")
```

This code can be downloaded from the website wwww.scilab.in

Scilab code Exa 10.22 `Output voltage`

```
//EX10_22 PG-10.76
clc
R1=100e3;
Cf=1e-6;
```

```
Vm=6e-3;//peak value of the input voltage
fr=2e3;//frequency  supplied at the input
w=2*%pi*fr;//angular frequency in rad/s
a=-Vm/(R1*Cf)//constant
printf("\n   We integrate Vin ie Vo = %.2f*integrate
   (sin4*pi*100)\n\n",a)
function y=f(x),y=sin(w*x),endfunction
t=input('   Enter the value of t (between 0 and 1) :
     ");
//the limit is from 0 to t
I=intg(0,t,f);//we should enter the value of t
x=I*a;
disp(x);
```

Chapter 11

Cathode Ray Oscilloscope CRO

Scilab code Exa 11.1 Amplitude and rms value of sinusoidal voltage

```
//EX11_1 PG-11.16
clc
disp("Refer to the figure -11.11 showwn")
disp("from the figure we can see that the screen is
   divided such")
disp("that one part of the half wave is sub-divided
   into 5 units")
disp("1subdivision=(1/5)units=0.2 units")
disp("amplitude of the positive peak signal=2
   division+3subdivision ")
disp("ie amplitude of the positive peak signal
   =2+3*0.2=2.6")
Vp=2+3*0.2;//amplitude of the peak voltage in terms
   of dividion
Vpp=2*Vp;//peak to peak voltage in terms of division
VA=2;//Vertical attenuation in (mV/div)
V=Vpp*VA;//required peak to peak output voltage in
   volts
Vm=V/2;//amplitude of the output voltage in volts
printf("\n Therefore amplitude of the output voltage
    in volts=%.1f mV \n",Vm)
```

```
Vrms=Vm/sqrt(2);//rms value of the output voltage
printf("\n Therefore rms value of the output voltage in volts=%.4f mV \n",Vrms)
```

Scilab code Exa 11.2 Rms voltage and frequency

```
//EX11_2 PG-11.19
clc
Vd=2;//voltage per division in(V/div)
Td=2e-3;//time base division in (s/div)
Vdiv=3;//vertical occupancy in division as shown in the screen
Vpp=Vd*Vdiv;//peak to peak voltage
Vm=Vpp/2;//peak voltage
Vrms=Vm/sqrt(2);//rms value of the output voltage
printf("\n Therefore rms value of the output voltage in volts=%.4f V \n",Vrms)
Hdiv=2;//horizontal occupancy in division as shown in the screen
T=Hdiv*Td;//time period of the waveform
f=1/T;//frequency
printf("\n Therefore frequency f=%.0f Hz \n ",f)
```

Chapter 12

Communication Systems

Scilab code Exa 12.1 Modulation index and percentage modulation

```
//EX12_1 Pg-12.14
clc
Vm=40;//peak value of the modulating signal
Vc=50;//peak value of the carrier signal
m=Vm/Vc;//modulation index
%m=m*100;
printf("modulation index m=%.1f \n
    %%m=%.0f%% \n",m,%m)
```

Scilab code Exa 12.2 Lower and upper side band frequency

```
//EX12_2 Pg-12.14
clc
clear
L=40e-6;
C=12e-9;
x=2*%pi*sqrt(L*C);
fc=1/x;//carrier frequency
```

```
f=5e3;//given audio frequency
fusb=fc+f;//upper side band frequency
flsb=fc-f;//lower side band frequency
BW=fusb-flsb;//required bandwidth
printf("Therefore upper side band frequency fusb=%.0
    f Hz \n",fusb)
printf(" Therefore lower side band frequency flsb=%
    .0f Hz \n",flsb)
printf("\n required bandwidth BW=%.0fHz",BW)
//in the book fc is approximated to 230Khz but the
    exact answer is 229.72kHz
```

Scilab code Exa 12.3 AM modulated waveform

```
//EX12_3 Pg-12.17
clc
clear
fm=5e3;//assume modulation frequency f=5kHz
fc=1080e3;//assume carrier frequency f=1080kHz
time=0:2.3148e-7:8e-4;
//Waveform of modulated signal for m=0.75
m1=0.75;//modulation index
VmbyVc=m1
Vm=1;//we assume modulation voltage=1V
Vc=Vm/m1;//carrier voltage
k=VmbyVc;//modulation index = Vm/Vc
printf("\n for modulation index m=0.75 Vc=%.2f V",Vc
    )

xset('window',1)
mt=k*sin(2*%pi*fm*time);
sam=Vc*(1+mt).*sin(2*%pi*fc*time);
plot(time(1:1500),sam(1:1500));
title(' Waveform of modulated signal m=0.75');
xlabel('Time (sec)');
```

```
ylabel('Amplitude (Vc=1.33V)');
xgrid(color("gray"));

//Waveform of modulated signal for m=1
m1=1;
VmbyVc=m1
Vm=1;//we assume modulation voltage=1V
Vc=Vm/m1;//carrier voltage
k=VmbyVc;//modulation index = Vm/Vc
printf("\n for modulation index m=1     Vc=%.2f V",Vc
   )

xset('window',2)
mt=k*sin(2*%pi*fm*time);
sam=Vc*(1+mt).*sin(2*%pi*fc*time);
plot(time(1:1500),sam(1:1500));
title(' Waveform of modulated signal m=1');
xlabel('Time (sec)');
ylabel('Amplitude (Vc=1V)');
xgrid(color("gray"));

//Waveform for modulated signal for m=1.25
m1=1.25;
VmbyVc=m1
Vm=1;//we assume modulation voltage=1V
Vc=Vm/m1;//carrier voltage
k=VmbyVc;//modulation index = Vm/Vc
printf("\n for modulation index m=1.25 Vc=%.2f V",Vc
   )

xset('window',3)
mt=k*sin(2*%pi*fm*time);
sam=Vc*(1+mt).*sin(2*%pi*fc*time);
plot(time(1:1500),sam(1:1500));
title(' Waveform of modulated signal m=1.25');
xlabel('Time (sec)');
ylabel('Amplitude (Vc=0.8V)');
xgrid(color("gray"));
```

Scilab code Exa 12.4 1Modulation index 2sideband frequencies and amplitude 3bandwi

```
//EX12_4 Pg-12.22
clc
clear
//modulation index
Vm=10;//peak value of the audio frequency signal
Vc=50;//peak value of the carrier signal
m=Vm/Vc;//modulation index
%m=m*100;
printf("modulation index m=%.1f \n      ie
    %%m=%.0f %% \n",m,%m)
//sideband frequencies
wm=2*%pi*500;
fm=wm/(2*%pi);
wc=2*%pi*1e5;
fc=wc/(2*%pi);
fusb=fc+fm;//upper side band frequency
flsb=fc-fm;//lower side band frequency
printf("\n Therefore upper side band frequency fusb=
    %.1f kHz \n",fusb*1e-3)
printf(" Therefore lower side band frequency fusb=%
    .1f kHz \n",flsb*1e-3)
//amplitude of each sinusoidal frequency
A=m*Vc/2;
printf("\n amplitude of upper and lower side bands=%
    .f V \n",A)
//bandwidth required
BW=fusb-flsb;//required bandwidth
printf("\n required bandwidth BW=%.0fHz\n",BW)
//power delivered to the load
R=600;//load resistance
P=Vc^2/(2*R)*(1+m^2/2)
printf("\n power delivered to the load %.3f W \n ",P
```

```
)
//transmission efficiency
n=m^2/(2+m^2)*100;
mprintf("\n transmission efficiency n=%.2f %% \n",n)
```

Scilab code Exa 12.5 Total power

```
//EX12_5 Pg-12.23
clc
clear
Pc=400;//power of the carrier signal
m=0.8;//modulation index
P=Pc*(1+m^2/2)
printf("  Therefore total power in the modulated
    wave is %.0f W",P)
```

Scilab code Exa 12.6 Power of each sidebands

```
//EX12_6 Pg-12.23
clc
clear
m=0.75;//modulation index
P=20;//total power in kW
Pc=P/(1+m^2/2)//since P=Pc*(1+m^2/2)
printf("therefore carrier power in the modulated
    wave is %.1f kW",Pc)
Psb=Pc*m^2/4;//side band power
Pusb=Psb;
Plsb=Psb;
printf("\n Pusb=%.1f kW \n Plsb=%.1f kW",Pusb,Plsb)
```

Scilab code Exa 12.7 Antenna current

```
//EX12_7 Pg-12.25
clc
clear
m=0.6;//modulation index
Itotal=5;//total antenna current
//Ic=total antenna current when only the carrier is
    sent
Ic=Itotal/sqrt(1+m^2/2)//since Itotal=Ic*sqrt(1+m
    ^2/2)
printf("Therefore total antenna current when only
    the carrier is sent Ic=%.1f A",Ic)
```

Scilab code Exa 12.8 modulation index

```
//EX12_8 Pg-12.26.
clc
clear
Ic=poly(0,'Ic');//unmodulated carrier signal
Itotal=1.15*Ic;//total rms current when the signal
    is modulated
x=Itotal/Ic;
x=horner(x,1)
y=2*((x)^2-1)
m=sqrt(y)
printf("modulation index m=%.1f ",m)
```

Scilab code Exa 12.9 modulation index

```
//EX12_9 Pg-12.28
clc
clear
```

```
m1=0.6;//first modulation index
m2=0.3;//second modulation index
m3=0.4;//third modulation index
mt=sqrt(m1^2+m2^2+m3^2);
printf("total modulation index m=%.3f ",mt)
```

Scilab code Exa 12.10 `modulation index and total radiated power`

```
//EX12_10 Pg-12.28
clc
clear
Ptotal=11.8;//radiated power in kW when the carrier is modulated
Pc=10;//radiated power in kW when the carrier is unmodulated
m=sqrt(2*((Ptotal/Pc)-1))
//when another sine wave of 30% of the
//initial modulation is transmitted simultaneously then
m1=0.3;//added sine wave signal is 30%
mt=sqrt(m1^2+m^2);
P=Pc*(1+mt^2/2);//total radiated power
printf("total radiated power P=%.2f kW ",P)
```

Scilab code Exa 12.11 `Carrier power`

```
//EX12_11 Pg-12.30
clc
clear
Ptotal=10;//radiated power in kW when the carrier is modulated
m=0.75;//modulation index
Pc=Ptotal/(1+m^2/2)//since Ptotal=Pc*sqrt(1+m^2/2)
```

```
printf("\n carrier power Pc=%.1f kW \n",Pc)
```

Scilab code Exa 12.12 Frequencies

```
//EX12_12 Pg-12.30
clc
clear
fc=1000e3;//carrier  frequency
fm1=300;//first audio frequency
fm2=800;//second audio frequency
fm3=1e3;//third audio frequency
fusb1=fc+fm1;//upper side band frequency
flsb1=fc-fm1;//lower side band frequency
fusb2=fc+fm2;//upper side band frequency
flsb2=fc-fm2;//lower side band frequency
fusb3=fc+fm3;//upper side band frequency
flsb3=fc-fm3;//lower side band frequency
printf("fusb1=%.1f kHz \n flsb1=%.1f kHz\n\n fusb2=%
    .1f kHz\n",fusb1*1e-3,flsb1*1e-3,fusb2*1e-3)
printf(" flsb2=%.1f kHz \n\n fusb3=%.1f kHz \n flsb3
    =%.1f kHz \n",flsb2*1e-3,fusb3*1e-3,flsb3*1e-3)
```

Scilab code Exa 12.13 Total sideband power radiated

```
//EX12_13 Pg-12.30
clc
clear
m1=0.55;//first modulation index
m2=0.65;//second modulation index
mt=sqrt(m1^2+m2^2);
Pc=360;;//power radiated by the carrier signal
Psb=Pc*mt^2/2//total sideband power radiated
```

```
printf("Therefore total sideband power radiated Psb=
    %.3f W",Psb)
//in the question Pc is taken as 360W but in the
    answer it is taken as
//300W I have taken Pc=300W so that Psb=150.5W
```

Scilab code Exa 12.14 `Current I`

```
//EX12_14 Pg-12.30
clc
clear
m=0.5;// modulation index
It=12;//antenna current when AM transmitter is 50%
    modulated
Ic=It/sqrt(1+m^2/2);//carrier current
m=0.9;//when modulation depth is increase to 0.9
It=Ic*sqrt(1+m^2/2)
printf("Therefore total antenna current It=%.2f A",
    It)
```

Scilab code Exa 12.15 `Current and percentage power swing`

```
//EX12_15 Pg-12.31
clc
clear
m1=0.6;// modulation index
It=1.5;//antenna current when AM transmitter is 50%
    modulated
Ic=It/sqrt(1+m1^2/2);//carrier current
m2=0.7//when another modulated signal is added
m=sqrt(m1^2+m2^2);//total modulation index
It=Ic*sqrt(1+m^2/2)

```

```
printf("\n Therefore total antenna current It=%.2f A \n",It)
disp("Ptotal=Pc+Pc*m^2/4+Pc*m^2/4 => Total power radiated")
Pc=poly(0,'Pc');
Ptotal=Pc+Pc*m^2/4+Pc*m^2/4 ;
P=Pc+Pc*m^2/4 //Total power if one of the side band is suppressed
%P=P/Ptotal;//percentage power saving
%P=horner(%P,1)*100
mprintf("\n Therefore percentage power saving %%P=%.0f %%",%P)
```

Scilab code Exa 12.16 1Carrier frequency 2modulating frequency 3carrier power 4tot

```
//EX12_16 Pg-12.32
clc
clear
disp(" Output voltage of the transmitter Vam=400(1+0.4sin6280t)sin3.14*10^7t")
Vc=400;//amplitude of carrier voltage
m=0.4;//modulation index
R=600;//load resistance
wm=6280;
wc=3.14e7;
fc=wc/(2*%pi);
fm=wm/(2*%pi);
Pc=Vc^2/(2*R);
Ptotal=Pc*(1+m^2/2);
printf("\n carrier frequency Fc=%.0f MHz \n modulating frequency =%.0f Hz \n ",fc*1e-6,fm)
printf("\n carrier power Pc=%.2f W \n Total power output Ptotal=%.1f W \n",Pc,Ptotal)
//peak power output results when modulating signal
//is at the peak of the +ve half cycle
```

```
Vm=m*Vc;
V=Vc+Vm;//peak output voltage
P=V^2/(2*R);//peak power output
printf("\n Peak output voltage P=%.2f W ",P)
//The exact value of fc is 4997465 Hz but in the
   book the
//value is taken as 5 MHz  same is the case for fm
```

Scilab code Exa 12.17 Equation of the wave and output wave form

```
//EX12_17 Pg-12.33
clc
clear
disp(" equation of the Am sine wave =Vc(1+m*sinwm*t)
   *sinwc*t")
Vc=12;//amplitude of carrier voltage
m=0.5;//modulation index
fc=10e6;//carrier frequency
fm=1e3;//modulated frequency
wc=2*%pi*fc;
wm=2*%pi*fm;
t=0:2.3148e-7:8e-4;
Vam=Vc*(1+m*sin(wm*t).*sin(wc*t))
fusb=fc+fm;//upper side band frequency
flsb=fc-fm;//lower side band frequency
A=m*Vc/2;//amplitude of side bands

//plotting of the graph
clf();
x=[ flsb flsb  ];//x-coordinate
y=[  0    A  ];//y-coordinate

plot2d(x,y,style=2)
x1=[10e6  10e6];//x-coordinate
y1=[ 0     12];//y-coordinate
```

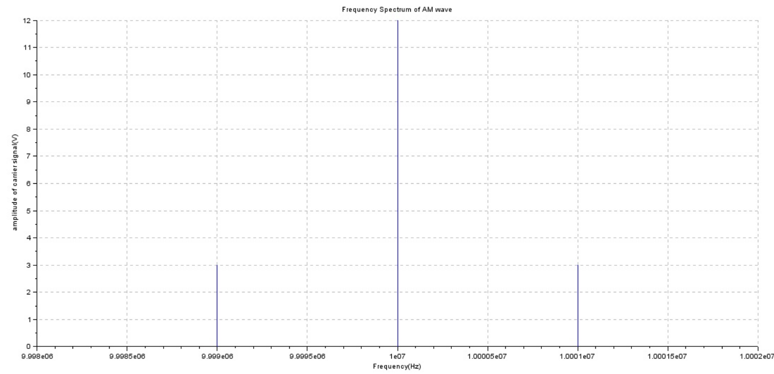

Figure 12.1: Equation of the wave and output wave form

```
plot2d(x1,y1,style=2)
x2=[fusb  fusb];//x-coordinate
y2=[ 0     A];//y-coordinate
plot(x2,y2)
x5=[flsb-1e3 flsb-1e3];//x-coordinate
y5=[    0        4    ];//y-coordinate
plot(x5,y5)
x6=[fusb+1e3 fusb+1e3];//x-coordinate
y6=[  0          4 ];//y-coordinate
plot2d(x6,y6)

xlabel('Frequency(Hz)');
ylabel('amplitude of carrier signal(V)');
title("Frequency Spectrum of AM wave")
xgrid(color("grey"));
```

Scilab code Exa 12.18 Percentage power saving

```
//Ex12_18 PG-12.18
```

```
clc
clear
printf(" When modulation index m=100%% ")
m=1;
disp(" Now Pdsbfc=1.5*Pc or Pdsbfc/Pc=1.5")
Pc=poly(0,'Pc')
Pdsbfc=Pc*(1+m^2/2);//power required for double sideband with full carrier transmission
Pssb=Pc*m^2/4;
%P=(Pdsbfc-Pssb)/Pdsbfc*100
x=horner(%P,1)
printf("  %%Power saving %%P :%f %%\n\n",x)

disp(" When modulation index m=50%% ")
m=0.5;
Pdsbfc=Pc*(1+m^2/2);//power required for double sideband with full carrier transmission
printf("  Now Pdsbfc=Pc*(1+m^2/2) \n  Pdsbfc=1.125*Pc  or Pdsbfc/Pc=1.125\n")
Pssb=Pc*m^2/4;
%P=(Pdsbfc-Pssb)/Pdsbfc*100
x=horner(%P,1)
printf("  %%Power saving %%P :%f %%",x)
```

Scilab code Exa 12.19 frequency deviation and modulation index

```
//EX12_19 Pg-40
clc
clear
R=2;//frequency deviation constant in KHz/V
V=20;//amplitude of the modulation signal
fd=R*V;// frequency deviation
f=4;//frequency applied in kHz
printf("\n Therefore frequency deviation f=%.0f kHz \n",fd)
```

```
m=fd/f;//modulation index
printf("     modulation index m=%.0f ",m)
```

Scilab code Exa 12.20 Frequency deviation and modulation index

```
//EX12_20 Pg-40
clc
clear
disp("when modulating voltage V=2.5 V")
V=2.5;//modulating voltage
fd1=5;// frequency deviation in kHz
R=fd1/V;//frequency deviation constant in KHz/V
printf("\n frequency deviation constant R=%.0f KHz/V
    \n",R)
disp("when modulating voltage V=7.5 V")
V=7.5;//new value amplitude of the modulating
   voltage
fd2=R*V//new frequency deviation in kHz
printf("\n Therefore frequency deviation f=%.0f kHz
   \n",fd2)
disp("when modulating voltage V=10 V")
V=10;//new value amplitude of the modulating voltage
fd3=R*V//new frequency deviation in kHz
printf("\n Therefore frequency deviation f=%.0f kHz
   \n",fd3)
fm=0.5;//modulation frequency in kHz ie 0.5kHz=500Hz
mf1=fd1/fm;
mf2=fd2/fm;
fm1=.25//new modulation frequency in kHz ie 0.25kHz
   =250Hz
mf3=fd3/fm1;
printf(" \n modulation index \n mf1=%.0f \n mf2=%.0f
    \n mf3=%.0f \n",mf1,mf2,mf3)
```

Scilab code Exa 12.21 Maximum deviation and modulation index

```
//EX12_21 Pg-41
clc
clear
mf=60;//modulation index
fm=0.4;//modulation frequency in kHz ie 0.4kHz=400Hz
//fd_max=maximum frequency deviation
fd_max=mf*fm;//since mf=fd_max_/fm
printf("maximum deviation fd_max=%.0f kHz \n",fd_max)
V=2.4;//modulating voltage
R=fd_max/V;//frequency deviation constant in KHz/V
disp("when modulating voltage V=3.2 V")
V=3.2;
fd=R*V;//frequency deviation
fm=0.25//modulation frequency in kHz ie 0.25kHz=250 Hz
mf=fd/fm;
printf(" \n modulation index mf=%.0f",mf)
//in the book the final answer for modulation mf=80
//is wrong the correct answer is mf=128
```

Scilab code Exa 12.22 Peak frequency deviation modulation index

```
//EX12_22 Pg-41
clc
clear
R=5;//frequency deviation constant in KHz/V
fm=10;//modulation frequency in kHz
V=15;//amplitude of the modulating signal
fd=R*V;//frequency deviation
```

```
printf("\n maximum frequency deviation  fd=%.0f KHz/V \n",fd)
mf=fd/fm;
printf(" \n modulation index mf=%.1f",mf)
```

Scilab code Exa 12.23 `1carrier and modulating frequency 2modulation index and maxi`

```
//EX12_23 Pg-41.43
clc
clear
disp("equation of a frequency modulated v=A*sin(wc*t+mf*sin(wm*t))")
disp("or v=A*sin(wc*t-fd/fm*cos(wm*t)) where fd=frequency deviation")
disp("Now v=50*sin(5e8*t-10*cos(1000*t))")
A=50;//peak value of the modulating signal
wc=5e8;
mf=10;
wm=1000;
fc=wc/(2*%pi);//carrier frequency
printf("\n carrier frequency fc=%.2f MHz \n",fc*1e-6)
fm=wm/(2*%pi);//modulating frequency
printf(" modulating frequency fc=%.2f Hz \n",fm)
//fd_max=maximum frequency deviation
fd_max=mf*fm;//since mf=fd_max_/fm
printf(" \n modulation index mf=%.0f",mf)
printf("\n maximum deviation fd_max=%.2f Hz \n",fd_max)
Vrms=A/sqrt(2);//rms value of the modulating signal
R=75;//wave resistance
P=Vrms^2/R;
printf("\n Power dissipated by the wave in resistance P=%.2f W",P)
```

Scilab code Exa 12.24 sketching the spectrum of the modulated FM wave

```
//EX12_24 Pg-41.47
clc
clear
V=5;//amplitude of modulating voltage
R=1;//frequency deviation constant in KHz/V
fd=V*R;// frequency deviation in kHz
fm=15;//modulating frequency in kHz
mf=fd/fm;//modulation index
printf(" \n modulation index mf=%.3f",mf);
disp("Now we refer from the table -12.2 of Bessel function ")
printf(" \n For modulation index mf=%.3f we take the value of J0,J1,and J2 ",mf);
J0=0.96;//for carrier frequency
J1=0.18;//first side frequency
J2=0.02;//second side frequency
A=5//amplitude of the carrier frequency
J0=J0*A;//for carrier frequency
J1=J1*A;//first side frequency
J2=J2*A;//second side frequency
printf("\n J0=%.1fV\n J1=%.1fV\n J2=%.1fV ",J0,J1,J2)
disp("Now we plot the frequency spectrum")
clf()
x=[ 89.97     89.97];//x-coordinate
y=[  0        0.1];//y-coordinate
plot2d(x,y,style=2)
x1=[89.985   89.985];//x-coordinate
y1=[  0       0.9];//y-coordinate
plot2d(x1,y1,style=2)
x2=[90     90];//x-coordinate
y2=[ 0   4.8];//y-coordinate
```

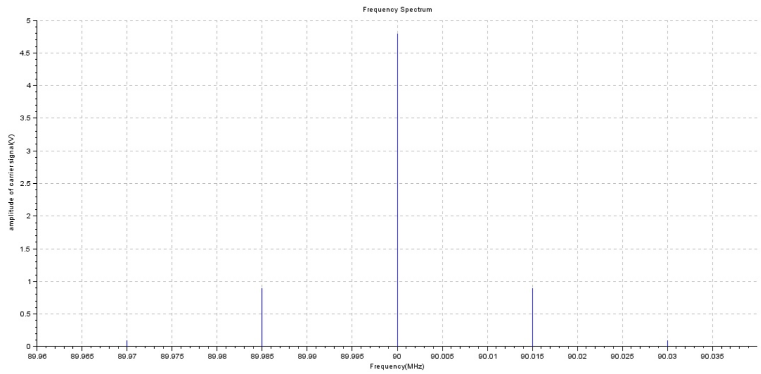

Figure 12.2: sketching the spectrum of the modulated FM wave

```
plot(x2,y2)
x3=[90.015 90.015];//x-coordinate
y3=[  0      0.9];//y-coordinate
plot(x3,y3)
x4=[90.03 90.03];//x-coordinate
y4=[  0      0.1];//y-coordinate
plot(x4,y4)
x5=[90.04 90.04];//x-coordinate
y5=[  0      5];//y-coordinate
plot(x5,y5)
x6=[ 89.96   89.96];//x-coordinate
y6=[  0       5];//y-coordinate
plot2d(x6,y6)
xlabel('Frequency(MHz)');
ylabel('amplitude of carrier signal(V)');
title("Frequency Spectrum")
xgrid(color("grey"));
```

Scilab code Exa 12.25 `Percent modulation`

```
//EX12_25 Pg-41.48
clc
clear
fd=15;//frequency deviation in kHz
f=75;//maximum frequency deviation in kHz for FM broadcast
%M=fd/f*100;
printf("\n For FM broadcast percent modulation is %%M=%.0f%%\n",%M)
f=25;//maximum frequency deviation in kHz for TV broadcast
%M=fd/f*100;
printf("\n For TV broadcast percent modulation is %%M=%.0f%% \n",%M)
```

Scilab code Exa 12.26 `Frequency deviation and carrier swing`

```
//EX12_26 Pg-41.48
clc
clear
%M=80;// percent modulation in %
f=75;//maximum frequency deviation in kHz for FM broadcast
fd=f*%M/100;//frequency deviation in kHz since %M=fd/f*100;
printf("\n Therefore frequency deviation for FM broadcast fd=%.0f Khz \n",fd)
cs=2*fd;//carrier swing
printf(" carrier swing=%.0f kHz\n",cs)
f=25;//maximum frequency deviation in kHz for TV broadcast
fd=f*%M/100;//frequency deviation in kHz since %M=fd/f*100;
printf("\n Therefore frequency deviation for TV broadcast fd=%.0f Khz \n",fd)
```

```
cs=2*fd;//carrier swing
printf(" carrier swing=%.0f kHz\n",cs)
```

Scilab code Exa 12.27 1Carrier swing 2highest and lowest frequencies 3modulation i

```
//EX12_27 Pg-41.51
clc
clear
fd=40;//frequency deviation in kHz
cs=2*fd;//carrier swing
printf("\n carrier swing=%.0f kHz\n",cs)
fc=93.2e3;//;carrier frequency in kHz
fh=fc+fd;//highest frequency reached
fl=fc-fd;//lowest frequency reached
printf("\n Therefore highest frequency reached fh=%.2f MHz\n ",fh*1e-3)
printf("Lowest frequency reached fl=%.2f MHz",fl*1e-3)
fm=5;//modulating frequency in kHz
m=fd/fm;//frequency modulation
printf("\n\n modulation index m=%.0f ",m)
```

Scilab code Exa 12.28 1Frequency deviation 2carrier swing 3lowest frequency reache

```
//EX12_28 Pg-41.51
clc
clear
fc=50.4e3;//;carrier frequency in kHz
fh=50.405e3//highest frequency reached in kHz
fd=fh-fc;//frequency deviation in kHz
printf("\n Therefore frequency deviation produced fd=%.0f Khz \n",fd)
cs=2*fd;//carrier swing
```

```
printf("\n carrier swing=%.0f kHz\n",cs)
fl=fc-fd;//lowest frequency reached
printf("\n Therefore lowest frequency reached fl=%.3f MHz",fl*1e-3)
```

Scilab code Exa 12.29 modulation index

```
//EX12_29 Pg-41.52
clc
clear
cs=70;//carrier swing in kHz
//since cs=2*fd
fd=cs/2;//frequency deviation in kHz
fm=7//modulating frequency in kHz
m=fd/fm;
printf("\n    Modulation index m=%.0f ",m)
```

Scilab code Exa 12.30 1Carrier swing 2frequency deviation 3carrier frequency 4modu

```
//EX12_30 Pg-41.52
clc
clear
fh=99.047e3;//highest frequency reached in kHz
fl=99.023e3;//lowest frequency reached in kHz
fm=7//modulating frequency in kHz
cs=fh-fl;//carrier swing
printf("Carrier swing=%.0f kHz\n",cs)
fd=cs/2;//frequency deviation in kHz
printf("\n Therefore frequency deviation fd=%.0f Khz \n",fd)
fc=fh-fd;//;carrier frequency in kHz
printf("\n Therefore carrier frequency fc=%.3f Mhz \n",fc*1e-3)
```

```
m=fd/fm;
printf("\n Modulation index m=%.3f ",m)
```

Chapter 13

Number Systems

Scilab code Exa 13.1 Represesntation of a number 98point72 in powers of 10

```
//EX13_1 PG-13.2
 clc
clear
disp("representation of the number 98.72 in power of
    10")
disp("N=(9*10^1)+(8*10^0)+(7*10^(-1))+(2*10^(-2))
   =98.72")
disp("The digit 9 has a weight of 10,the digit 8 has
    a weight of 1,the digit 7")
disp("has a weight of (1/10) and the digit 2 has a
   weight of (1/100)")
```

Scilab code Exa 13.2 1101point101 in power of 2 and its decimal equivalent

```
//EX13_2 PG-13.3
clc
clear
disp("Representation of the binary number 1101.101
   in power of 2")
```

```
disp("N=(1*2^3)+(1*2^2)+(0*2^1)+(1*2^0)+(1*2^(-1))
    +(0*2^(-2))+(1*2^(-3))=13.625")
N=(1*2^3)+(1*2^2)+(0*2^1)+(1*2^0)+(1*2^(-1))
    +(0*2^(-2))+(1*2^(-3))
printf("\n The decimal equivalent of binary no
    1101.101 is: %.3f",N)
```

Scilab code Exa 13.3 567 in power of 8 and its decimal equivalent

```
//EX13_3 PG-13.3
clc
clear
disp("representation of the number 567 in power of 8
    ")
disp("N=(5*8^2)+(6*8^1)+(7*8^0)=375")
printf(" Therefore decimal equivalent of 567 is: ")
N=(5*8^2)+(6*8^1)+(7*8^0)
printf("%.0f",N)
```

Scilab code Exa 13.4 3FD in power of 16 and its decimal equivalent

```
//EX13_4 PG-13.4
clc
clear
disp("Representation of the hexadecimalnumber 3FD in
     power of 16")
disp("N=(3*16^2)+(F*16^1)+(D*16^0)=1021")
printf(' The Decimal equivalent of the Binary number
     3FD is:');
N=(3*16^2)+(15*16^1)+(13*16^0)
printf("%.0f",N)
```

Scilab code Exa 13.5 decimal equivalent of 231point23 with base 4

```
//EX13_5 PG-13.5
clc
clear
printf('The Decimal equivalent of the number 231.23
  with base 4 is: ');
x=(2*4^2)+(3*4^1)+(1*4^0)+(2*4^(-1))+(3*4^(-2))
printf("%.4f",x)
```

Scilab code Exa 13.6 counting from 0 to 9 in radix 5

```
//EX13_6 PG-13.5
clc
clear
disp("decimal from 0 to 9 in radix 5 ")
for i=0:1:9;
 a=i/5;
 b=modulo(i,5);
 printf("      %d=%d%d\n",i,a,b);//conversion from
   decimal to radix 5
 end
```

Scilab code Exa 13.7 111101100 with base 2 to octal equivalent

```
//EX13_7 PG-13.7
clc
clear
printf("Conversion of binary no 111101100 to octal
  equivalent is :")
```

```
a=['111101100'];
x=bin2dec(a);
y=dec2oct(x);
printf("%s",y)
```

Scilab code Exa 13.8 634 with base 8 to binary equivalent

```
//EX13_8 PG-13.7
clc
clear
printf("conversion of octal no 634 to binary
   equivalent is :")
a=['634']
x=oct2dec(a)//first we convert to decimal
y=dec2bin(x)
printf("%s",y)
```

Scilab code Exa 13.9 725point63 with base 8 to binary equivalent

```
//EX13_9 PG-13.7
clc
clear
disp("conversion of octal no 725.63 to binary
   equivalent")
a=725.63;
//first we convert the number 725.63(octal)to
   decimal
x=(7*8^2)+(2*8^1)+(5*8^0)+(6*8^(-1))+(3*8^(-2));
//then we convert the decimal to binary
z=modulo(x,1)
x=floor(x);//separating the decimal from the integer
    part
b=0;
```

```
c=0;
d=0;
while(x>0)  //taking integer part into a matrix and
   convert to equivalent binary
y=modulo(x,2);
b=b+(10^c)*y;
x=x/2;
x=floor(x);
c=c+1;
end
for i=1:10;//converting the values after the decimal
    point into binary
    z=z*2;
    q=floor(z);
    d=d+q/(10^i);
    if z>=1 then
        z=z-1;
    end
end
s=b+d;
printf("\n The binary equivalent of the given octal
   number 725.63 is =%.6f",s);
```

Scilab code Exa 13.10 1101100010011011 with base 2 to hexadecimal equivalent

```
//EX13_10 PG-13.7
clc
clear
printf("hexadecimal equivalent of 1101100010011011
   binary is : ")
a=['1101100010011011'];
x=bin2dec(a);
y=dec2hex(x);
printf("%s",y)
```

Scilab code Exa 13.11 `3FD with base 16 to binary equivalent`

```
//EX13_11 PG-13.8
clc
clear
printf("Binary equivalent of 3FD hexadecimal is :")
a=['3FD']
x=hex2dec(a)
y=dec2bin(x)
printf("00%s",y)
```

Scilab code Exa 13.12 `5A9pointB4 with base 16 to binary equivalent`

```
//EX13_12 PG-13.8
clc
clear
printf("Conversion of hexadecimal no 5A9.B4 to
    binary equivalent\n ")
a=['5A9.B4'];
//first we convert the number 5A9.B4(hexadecimalal)
    to decimal
x=(5*16^2)+(10*16^1)+(9*16^0)+(11*16^(-1))
    +(4*16^(-2));
//then we convert the decimal to binary
z=modulo(x,1)
x=floor(x);//separating the decimal from the integer
    part
b=0;
c=0;
d=0;
while(x>0)  //taking integer part into a matrix and
    convert to equivalent binary
```

```
y=modulo(x,2);
b=b+(10^c)*y;
x=x/2;
x=floor(x);
c=c+1;
end
for i=1:10;//converting the values after the decimal point into binary
    z=z*2;
    q=floor(z);
    d=d+q/(10^i);
    if z>=1 then
        z=z-1;
    end
end
s=b+d;
printf("The binary equivalent of the given\n hexadecimal number 5A9.B4 is = 0%.6f00",s);
```

Scilab code Exa 13.13 615 with base 8 to hexadecimal equivalent

```
//EX13_13 PG-13.9
clc
clear
printf("conversion of octal no 615 to its hexadecimal equivalent :")
a=['615'];
x=oct2dec(a)
y=dec2hex(x)
printf("%s",y)
```

Scilab code Exa 13.14 25B with base 16 to octal equivalent

```
//EX13_14 PG-13.9
clc
clear
printf("conversion of hexadecimal no 25B to its
   octal equivalent : ")
a=['25B'];
x=hex2dec(a)
y=dec2oct(x)
printf("%s",y)
```

Scilab code Exa 13.15 1101point1 with base 2 to decimal equivalent

```
//EX13_15 PG-13.10
clc
clear
printf("conversion of binary no 1101.1 to its
   decimal equivalent =")
N=(1*2^3)+(1*2^2)+(0*2^1)+(1*2^0)+(1*2^(-1));
printf(" %.1f",N)
```

Scilab code Exa 13.16 475point25 with base 8 to decimal equivalent

```
//EX13_16 PG-13.10
clc
clear
printf("conversion of octal no 475.25 to its decimal
    equivalent =")
N=(4*8^2)+(7*8^1)+(5*8^0)+(2*8^(-1))+(5*8^(-2));
printf(" %.5f",N)
```

Scilab code Exa 13.17 `9B2point1A with base 16 to decimal equivalent`

```
//EX13_17 PG-13.10
clc
clear
printf("conversion of hexadecimal no 9B2.1A to its
    decimal equivalent =")
N=(9*16^2)+(11*16^1)+(2*16^0)+(1*16^(-1))
    +(10*16^(-2));
printf(" %.1f",N)
```

Scilab code Exa 13.18 `3102point12 with base 4 to decimal equivalent`

```
//EX13_18 PG-13.10
clc
clear
printf("Conversion of the number 3102.12 \n with
    base 4 to its decimal equivalent =")
N=(3*4^3)+(1*4^2)+(0*4^1)+(2*4^0)+(1*4^(-1))
    +(2*4^(-2));
printf(" %.3f",N)
```

Scilab code Exa 13.19 `614point15 with base 7 to decimal equivalent`

```
//EX13_18 PG-13.10
clc
clear
printf("Conversion of the number 614.15 with base 7
    to its decimal equivalent =")
N=(6*7^2)+(1*7^1)+(4*7^0)+(1*7^(-1))+(5*7^(-2));
printf(" %.4f",N)
```

Scilab code Exa 13.20 37 with base 10 to binary equivalent

```
//Ex13_20 PG-13.11
clc
clear
printf("  Conversion of decimal number 37 to its
   binary equivalent =")
a=[37];
x=dec2bin(a);
printf("%s",x)
```

Scilab code Exa 13.21 214 with base 10 to octal equivalent

```
//Ex13_21 PG-13.12
clc
clear
printf("conversion of decimal number 214 to its
   octal equivalent =")
a=[214];
x=dec2oct(a)
printf(" %s",x)
```

Scilab code Exa 13.22 3509 with base 10 to hexadecimal equivalent

```
//Ex13_22 PG-13.12
clc
clear
printf("conversion of decimal number 3509 to its
   hexadecimal equivalent =")
```

```
a=[3509];
x=dec2hex(a)
printf(" %s",x)
```

Scilab code Exa 13.23 54 with base 10 to radix 4

```
//Ex13_23 PG-13.13
clc
clear
printf("conversion of decimal number 54 base  to a
   number with base 4 =")
a=[54]
x=dec2base(a,4);
printf(" %s",x)
```

Scilab code Exa 13.24 point8125 with base 10 to binary equivalent

```
//Ex13_24 PG-13.13
clc
clear
disp("Conversion of decimal number 0.8125 base  to
   its binary equivalent ")
a=[0.8125];
z=modulo(a,1);
d=0;
for i=1:10;//converting the values after the decimal
    point into binary
    z=z*2;
    q=floor(z);
    d=d+q/(10^i);
    if z>=1 then
        z=z-1;
    end
```

```
end
s=d;
printf("\n The binary equivalent of the given
   decimal number 0.8125 is = %.4f",s);
```

Scilab code Exa 13.25 `point95 with base 10 to binary equivalent`

```
//Ex13_25 PG-13.14
clc
clear
disp("Conversion of decimal number 0.95 base  to its
    binary equivalent ")
a=[0.95];
z=modulo(a,1);
d=0;
for i=1:10;//converting the values after the decimal
    point into binary
    z=z*2;
    q=floor(z);
    d=d+q/(10^i);
    if z>=1 then
        z=z-1;
    end
end
s=d;
printf("\n The binary equivalent of the given
   decimal number 0.95 is = %.7f",s);
```

Scilab code Exa 13.26 `point640625 with base 10 to octal equivalent`

```
//Ex13_26 PG-13.14
clc
clear
```

```
disp("Conversion of decimal number 0.640625 base  to
    its octal equivalent =")
a=[0.640625];
z=modulo(a,1);
d=0;
for i=1:10;//converting the values after the decimal
    point into octal
    z=z*8;
    q=floor(z);
    d=d+q/(10^i);
    if z>=1 then
        z=z-q;
    end
end
s=d;
printf("\n The octal equivalent of the given decimal
    number 0.640625 is = %.2f",s);
```

Scilab code Exa 13.27 point1289062 with base 10 to hexadecimal equivalent

```
//Ex13_27 PG-13.14
clc
clear
disp("Conversion of decimal number 0.1289062 base
   to its hexadecimal equivalent ")
a=[0.1289062];
z=modulo(a,1);
d=0;
for i=1:10;//converting the values after the decimal
    point into octal
    z=z*16;
    q=floor(z);
    d=d+q/(10^i);
    if z>=1 then
        z=z-q;
```

```
    end
end
s=d;
printf("\n The hexadecimal equivalent of the given
   decimal number 0.640625 is = %.3f",s);
```

Scilab code Exa 13.28 24point6 with base 10 to binary equivalent

```
 //Ex13_28 PG-13.14
clc
clear
disp("Conversion of decimal number 24.6 base  to its
    binary equivalent ")
a=24.6;
z=modulo(a,1)
x=floor(a);//separating the decimal from the integer
    part
b=0;
c=0;
d=0;
while(x>0)  //taking integer part into a matrix and
   convert to equivalent binary
y=modulo(x,2);
b=b+(10^c)*y;
x=x/2;
x=floor(x);
c=c+1;
end
for i=1:10;//converting the values after the decimal
    point into binary
    z=z*2;
    q=floor(z);
    d=d+q/(10^i);

    if z>=1 then
```

```
        z=z-1;
    end
end
s=b+d;
printf("\n The binary equivalent of the given
   decimal number 24.6 is =%.5f",s);
```

Scilab code Exa 13.29 `35point45 with base 10 to octal equivalent`

```
//Ex13_29 PG-13.15
clc
clear
disp("Conversion of decimal number 35.45 to its
   octal equivalent ")
a=35.45;
z=modulo(a,1)
x=floor(a);//separating the decimal from the integer
    part
b=0;
c=0;
d=0;
while(x>0)  //taking integer part into a matrix and
   convert to equivalent octal
y=modulo(x,8);
b=b+(10^c)*y;
x=x/8;
x=floor(x);
c=c+1;
end
for i=1:10;//converting the values after the decimal
    point into octal
    z=z*8;
    q=floor(z);
    d=d+q/(10^i);
    if z>=1 then
```

```
        z=z-q;
    end
end
s=b+d;
printf("\n The octal equivalent of the given decimal
    number 35.45 is =%.5f",s);
```

Scilab code Exa 13.30 `22point46 with base 10 to hexadecimal equivalent`

```
//Ex13_30 PG-13.16
clc
clear
printf("\n Conversion of decimal number 22.64 to its
    hexadecimal equivalent \n ")
a=22.64;
z=modulo(a,1)
x=floor(a);//separating the decimal from the integer
    part
b=dec2hex(x)

//converting the decimal part of the number into
   hexadecimal
    z=z*16;
    q=floor(z);
       if  (q==10)
        a1=['A']
 else   if (q==11)
        a1=['B']
 else  if (q==12)
        a1=['C']
 else   if (q==13)
        a1=['D']
 else    if (q==14)
        a1=['E']
 else   if (q==15)
```

```
        a1=['F']
       else a1=q
end
end
end
end
end
end
if z>=1 then
        z=z-q;
end

   z=z*16;
    q=floor(z);
       if  (q==10)
        a2=['A']
 else   if (q==11)
        a2=['B']
 else  if (q==12)
        a2=['C']
 else   if (q==13)
        a2=['D']
 else    if (q==14)
        a2=['E']
 else   if (q==15)
        a2=['F']
       else a2=q
end
end
end
end
end
end
if z>=1 then
        z=z-q;
end

   z=z*16;
```

```
    q=floor(z);
       if  (q==10)
        a3=['A']
 else   if (q==11)
        a3=['B']
 else  if (q==12)
        a3=['C']
 else   if (q==13)
        a3=['D']
 else    if (q==14)
        a3=['E']
 else   if (q==15)
        a3=['F']
       else a3=q
end
end
end
end
end
end
if z>=1 then
        z=z-q;
end

   z=z*16;
    q=floor(z);
       if  (q==10)
        a4=['A']
 else   if (q==11)
        a4=['B']
 else  if (q==12)
        a4=['C']
 else   if (q==13)
        a4=['D']
 else    if (q==14)
        a4=['E']
 else   if (q==15)
        a4=['F']
```

```
        else a4=q
end
end
end
end
end
end
if z>=1 then
         z=z-q;
end
printf("The hexadecimal equivalent of the given
   decimal number 22.64 is:")
printf(" %s.%s%.0f%s%.0f",b,a1,a2,a3,a4);
```

Scilab code Exa 13.31 convert the given binary nos to dec hex and oct

```
//Ex13_31 PG-13.17
clc
clear
printf("\n  i)\n Conversion of binary number
   101101.1101 to\n\n")

//conversion into decimal form
N=(1*2^5)+(0*2^4)+(1*2^3)+(1*2^2)+(0*2^1)+(1*2^0)
   +(1*2^(-1))+(1*2^(-2))+(0*2^(-3))+(1*2^(-4));
printf(" Decimal form = %.4f\n",N)

//Conversion into hexadecimal form
x=floor(N);//separating the integer part from the
   decimal part
b=dec2hex(x);
z=modulo(N,1);
d=0;
//converting the values after the decimal point into
    hexadecimal
```

```
//first we convert into decimal form and then into hexadecimal form
for i=2:5;//converting the values after the decimal point into hexadecimal
    z=z*16;
    q=floor(z);
    d=d+q/(10^i);
    if z>=1 then
        z=z-q;
    end
end
   if (d==.10)
           a=['A']
           end
   if (d==.11)
           a=['B']
           end
   if (d==.12)
           a=['C']
           end
   if (d==.13)
           a=['D']
           end
   if (d==.14)
           a=['E']
           end
   if (d==.15)
           a=['F']
           end
printf(" Hexadecimal form = %s.%s\n",b,a)

//conversion into octal form
z=modulo(N,1)
x=floor(N);//separating the decimal from the integer part
b=0;
c=0;
d=0;
```

```
while(x>0)  //taking integer part into a matrix and
   convert to equivalent binary
y=modulo(x,8);
b=b+(10^c)*y;
x=x/8;
x=floor(x);
c=c+1;
end
for i=1:10;//converting the values after the decimal
    point intooctal
    z=z*8;
    q=floor(z);
    d=d+q/(10^i);
    if z>=1 then
        z=z-q;
    end
end
s=b+d;
printf(" octal  form = %.2f\n\n",s);
printf("\n  ii)\n Conversion of binary number
   11011011.100101 to\n\n")

//conversion into decimal form
N=(1*2^7)+(1*2^6)+(0*2^5)+(1*2^4)+(1*2^3)+(0*2^2)
   +(1*2^1)+(1*2^0)+(1*2^(-1))+(0*2^(-2))+(0*2^(-3))
   +(1*2^(-4))+(0*2^(-5))+(1*2^(-6));
printf(" Decimal form =%.6f\n\n",N)

//Conversion into hexadecimal form
x=floor(N);//separating the integer part from the
   decimal part
b=dec2hex(x);
z=modulo(N,1);//first we convert into decimal form
   and then into hexadecimal form
d=0;
//converting the values after the decimal point into
    hexadecimal
for i=1:10;//converting the values after the decimal
```

```
point into hexadecimal
    z=z*16;
    q=floor(z);
    d=d+q/(10^i);
    if z>=1 then
        z=z-q;
    end
end
printf(" Hexadecimal form\n    integer part 11011011 =  %s \n    ",b)
printf("decimal part 100101   =%.2f\n\n",d)

//conversion into octal form
z=modulo(N,1)
x=floor(N);//separating the decimal from the integer part
b=0;
c=0;
d=0;
while(x>0)  //taking integer part into a matrix and convert to equivalent binary
y=modulo(x,8);
b=b+(10^c)*y;
x=x/8;
x=floor(x);
c=c+1;
end
for i=1:10;//converting the values after the decimal point intooctal
    z=z*8;
    q=floor(z);
    d=d+q/(10^i);
    if z>=1 then
        z=z-q;
    end
end
s=b+d;
printf(" octal  form =%.2f",s);
```

Scilab code Exa 13.32 `2AC5pointD with base 16 to dec oct and bin`

```
//EX13_32 Pg-18
clc
clear
printf("\n\n    Conversion of hexadecimal number 2AC5
    .D to \n\n")

//conversion into decimal form
N=(2*16^3)+(10*16^2)+(12*16^1)+(5*16^0)+(13*16^(-1))
printf("    Decimal form = %.4f\n\n",N)

//conversion into octal form
//we take the value of the decimal form and convert
    it to octal form
z=modulo(N,1)
x=floor(N);//separating the decimal from the integer
     part
b=0;
c=0;
d=0;
while(x>0)  //taking integer part into a matrix and
    convert to equivalent binary
y=modulo(x,8);
b=b+(10^c)*y;
x=x/8;
x=floor(x);
c=c+1;
end
for i=1:10;//converting the values after the decimal
     point intooctal
    z=z*8;
    q=floor(z);
    d=d+q/(10^i);
```

```
    if z>=1 then
        z=z-q;
    end
end
s=b+d;
printf("    Octal  form = %.2f\n\n",s);

//conversion into binary form
//we take the value of the decimal form and convert
   it to octal form
z=modulo(N,1)
x=floor(N);//separating the decimal from the integer
    part
b=0;
c=0;
d=0;
while(x>0)  //taking integer part into a matrix and
   convert to equivalent binary
y=modulo(x,2);
b=b+(10^c)*y;
x=x/2;
x=floor(x);
c=c+1;
end
for i=1:10;//converting the values after the decimal
    point into binary
    z=z*2;
    q=floor(z);
    d=d+q/(10^i);

    if z>=1 then
        z=z-1;
    end
end
s=b+d;
printf("    Binary  form : ")
printf("\n       Integer part 2AC5 = 00%.0f",b)
printf("\n       Decimal part 0.D = %.4f00",d)
```

Scilab code Exa 13.33 `determination of the value of base x`

```
//EX13_33 Pg-18
clc
clear
printf("\n    i)\n    Detemination of value of base x \n")
printf("    (193)_x = (623)_8\n\n")
printf("    First we convert (623)_8 octal into decimal\n\n")
N=(6*8^2)+(2*8^1)+(3*8^0)
printf("            (623)_8 = (%.0f)_10\n\n",N)
printf("   Therefore (193)_x = (%.0f)_10\n\n",N)
printf("   (1*x^2)+(9*x^1)+(3*x^0) = %.0f\n\n",N)
printf("    x^2 + 9x + 3 = %.0f\n\n",N)
printf("    x^2 + 9x %.0f = 0\n\n",3-N)
a=1;
b=9;
c=-400
x1=(-b+sqrt(b^2-4*a*c))/(2*a);
x2=(-b-sqrt(b^2-4*a*c))/(2*a);
printf("Therefore x1 = %.0f\n            x2 = %.0f",x1,x2)
printf("\n   Negative is not applicable. Therefore x = %.0f",x1)
printf("\n   Hence (193)_%.0f = (623)_8\n\n",x1)

printf("\n    ii)\n    Detemination of value of base x \n")
printf("    (225)_x = (341)_8\n\n")
printf("    First we convert (341)_8 octal into decimal\n\n")
N=(3*8^2)+(4*8^1)+(1*8^0)
printf("            (341)_8 = (%.0f)_10\n\n",N)
```

```
printf("Therefore (225)_x = (%.0f)_10\n\n",N)
printf("    (2*x^2)+(2*x^1)+(2*x^0) = %.0f\n\n",N)
printf("     2x^2 + 2x +5 = %.0f\n\n",N)
printf("     2x^2 + 2x %.0f= 0\n\n",5-N)
a=2;
b=2;
c=-200
x1=(-b+sqrt(b^2-4*a*c))/(2*a);
x2=(-b-sqrt(b^2-4*a*c))/(2*a);
printf("Therefore x1 = %.0f\n            x2 = %.0f",x1,x2)
printf("\n   Negative is not applicable. Therefore x = %.0f",x1)
printf("\n   Hence (225)_%.0f = (341)_8\n\n",x1)

printf("\n    iii)\n     Detemination of value of base x \n")
printf("     (211)_x = (152)_8\n\n")
printf("     First we convert (152)_8 octal into decimal\n\n")
N=(1*8^2)+(5*8^1)+(2*8^0)
printf("            (152)_8 = (%.0f)_10\n\n",N)
printf("Therefore (211)_x = (%.0f)_10\n\n",N)
printf("    (2*x^2)+(1*x^1)+(1*x^0) = %.0f\n\n",N)
printf("     2x^2 + 1x +1 = %.0f\n\n",N)
printf("     2x^2 + 1x %.0f = 0\n\n",1-N)
a=2;
b=1;
c=-105
x1=(-b+sqrt(b^2-4*a*c))/(2*a);
x2=(-b-sqrt(b^2-4*a*c))/(2*a);
printf("Therefore x1 = %.0f\n            x2 = %.0f",x1,x2)
printf("\n   Negative is not applicable. Therefore x = %.0f",x1)
printf("\n   Hence (211)_%.0f = (152)_8\n\n",x1)
```

Scilab code Exa 13.34 1s complement of 1101

```
//EX13_34 Pg-20
clc
clear
printf("   1''s complement of 1101 = ")
x=['1101'];
y=bin2dec(x)
z=dec2bin(bitcmp(y,4))
printf("00%s",z)
```

Scilab code Exa 13.35 1s complement of 10111001

```
//EX13_34 Pg-20
clc
clear
printf("1''s complement of 10111001 = ")
x=['10111001'];
y=bin2dec(x)
z=dec2bin(bitcmp(y,8))
printf("0%s",z)
```

Scilab code Exa 13.36 2s complement of 1001

```
//EX13_36 Pg-21
clc
clear
printf("   2''s complement 1001 is :")
x=['1001'];
```

```
y=bin2dec(x);//binary to decimal conversion//
z=bitcmp(y,4);//one's complement of the number//
z=z+1;
z2=dec2bin(z)//2's complement of the number//
printf(" 0%s",z2)
```

Scilab code Exa 13.37 `2s complement of 10100011`

```
//EX13_37 Pg-21
clc
clear
printf("2''s complement 10100011 is : ")
x=['10100011'];
y=bin2dec(x);//binary to decimal conversion//
z=bitcmp(y,8);//one's complement of the number//
z=z+1;
z2=dec2bin(z)//2's complement of the number//
printf("0%s",z2)
```

Scilab code Exa 13.38 `7s complement of 612`

```
//EX13_38 Pg-22
clc
clear
printf("7''s complement of(612)_8 is : ")
x=['612'];
y=oct2dec(x)
z=dec2oct(bitcmp(y,8))
printf("%s",z)
```

Scilab code Exa 13.39 8s complement of 346

```
//EX13_39 Pg-21
clc
clear
printf("8''s complement (346)_8 is : ")
x=['346'];
y=oct2dec(x);//octal to decimal conversion//
z=bitcmp(y,9);//one's complement of the number//
z=z+1;
z2=dec2oct(z)//8's complement of the number//
printf("%s",z2)
```

Scilab code Exa 13.40 15s complement of A9B

```
//EX13_40 Pg-22
clc
clear
printf("15''s complement (A9B)_16 is :")
x=['A9B'];
y=hex2dec(x);//hexadecimal to decimal conversion//
z=dec2hex(bitcmp(y,11));//15's complement of the
    number//
printf("%s",z)
```

Scilab code Exa 13.41 16s complement of A8C

```
//EX13_41 Pg-23
clc
clear
printf("16''s complement (A8C)_16 is : ")
x=['A8C'];
y=hex2dec(x);//hexadecimal to decimal conversion//
```

```
z=bitcmp(y,12);//one's complement of the number//
z=z+1;
z2=dec2hex(z)//16's complement of the number//
printf("%s",z2)
```

Scilab code Exa 13.42 binary addition

```
//EX13_42 Pg-23
clc
clear
x=['1010'];
y=['0011'];
//binary to decimal conversion//
x=bin2dec(x)
y=bin2dec(y)
z=x+y;
a=dec2bin(z)//decimal to binary conversion//
printf('the addition of given numbers is: ')
printf("%s",a)
```

Scilab code Exa 13.43 addition of 28 and 15 in binary

```
//EX13_43 Pg-23
clc
clear
x=28;
y=15;
// decimal to binary  conversion//
z=x+y;
a=dec2bin(z)
printf('the addition of given numbers in binary is:
    ')
printf("%s",a)
```

```
printf('\n the addition of given numbers in decimal
   is: ')
printf("%.0f",z)
```

Scilab code Exa 13.44 subtraction of 0101 from 1011

```
//EX13_44 Pg-24
clc
clear
x=['1011'];
y=['0101'];
//binary to decimal conversion//
x=bin2dec(x)
y=bin2dec(y)
z=x-y;
a=dec2bin(z)//decimal to binary conversion//
printf('the subtraction of given numbers is: ')
printf("0%s",a)
```

Scilab code Exa 13.45 subtraction of 101011 from 111001 using 1s complement

```
//EX13_45 Pg-25
clc
clear
printf("subtraction of 101011 from 111001 using 1''s
    complement method")
printf("\n\n we know that 111001>101011\n\n")
printf(" Therefore 111001-101011 =")
x=['111001'];
y=['101011'];
//we should note that in the question the first
   binary number is 101011 and not 1001011

```

```
//binary to decimal conversion//
x=bin2dec(x)
y=bin2dec(y)
y1=bitcmp(y,6)//one's complement of the smaller
    number
z=x+y1;//addition of x with the one's complent of y
//subtraction of smaller number from larger number
w=bitset(z,7,0)//the end round carry should be
    remove and add to z
a=w+1;
a1=dec2bin(a)//final result
printf(" 00%s",a1)
```

Scilab code Exa 13.46 subtraction of 111001 from 101011 using 1s complement

```
//EX13_46 Pg-25
clc
clear
printf("subtraction of 111001 from 101011 using 1''s
    complement method")
printf("\n\n we know that 101011<111001\n\n")
printf(" Therefore 101011-111001 = ")
x=['101011'];
y=['111001'];
//binary to decimal conversion//
x=bin2dec(x)
y=bin2dec(y)
y1=bitcmp(y,6)//one's complement of the larger
    number
z=x+y1;//addition of x with the one's complement of
    y
//subtraction of larger number from smaller number
z=bitcmp(z,6);//one's complement of the result
a=dec2bin(z)//decimal to binary conversion
printf(" -00%s",a)//final answer is -ve
```

Scilab code Exa 13.47 subtraction of 101011 from 111001 using 2s complement

```
//EX13_47 Pg-26
clc
clear
printf("subtraction of 101011 from 111001 using 2''s
    complement method")
printf("\n\n we know that 111001>101011\n\n")
printf(" Therefore 111001-101011 = ")
x=['111001'];
y=['101011'];
//binary to decimal conversion//
x=bin2dec(x)
y=bin2dec(y)
y1=bitcmp(y,6)//one's complement of the smaller
    number
y2=y1+1;//2's complement of the smaller number
//subtraction of smaller number from larger number
a=x+y2;
w=bitset(a,7,0)//we discard the carry
s=dec2bin(w)
printf("00%s",s)
```

Scilab code Exa 13.48 subtraction of 111001 from 101011 using 2s complement

```
//EX13_48 Pg-26
clc
clear
printf("subtraction of 111001 from 101011 using 2''s
    complement method")
printf("\n\n we know that 101011<111001\n\n")
```

```
printf(" Therefore 101011−111001 =")
x=['101011'];
y=['111001'];
//binary to decimal conversion//
x=bin2dec(x)
y=bin2dec(y)
y1=bitcmp(y,6)//one's complement of the larger
   number
y2=y1+1;//2's complement of the larger number
//subtraction of larger number from smaller number
a=x+y2;//result is in two complement
a1=bitcmp(a,6)//one's complement of the result
a2=a1+1;//final answer
s=dec2bin(a2)
printf("  −00%s",s)//final answer is −ve
```

Scilab code Exa 13.49 expression of the given decimal nos in 1s and 2s complement

```
//EX13_49 Pg−26
clc
clear
printf('for decimal number = −56 the binary
   equivalent is ')
x=56;
y=dec2bin(x)
printf("0%s",y)
z=bitcmp(x,7)//1's complement=−56
printf('\n the 1''s complement is ')
z1=dec2bin(z)
printf("%s",z1)
z=z+1;//2's complement
printf('\n the 2''s complement ')
z2=dec2bin(z)
printf("%s",z2)
printf('\n\n for decimal number = −107 the binary
```

```
    equivalent is ')
x=107;
y=dec2bin(x)
printf("0%s",y)
z=bitcmp(x,8)//1's complement=-56
printf('\n the 1''s complement is ')
z1=dec2bin(z)
printf("%s",z1)
z=z+1;//2's complement
printf('\n the 2''s complement ')
z2=dec2bin(z)
printf("%s",z2)
```

Scilab code Exa 13.50 decimal nos subtraction using 2s complement

```
//EX13_50 Pg-27
clc
clear
//subtraction of decimal numbers using 2's
    complement
//given decimal numbers
x=42;
y=68;
//conversion from
x1=dec2bin(x)
y1=dec2bin(y)
printf("The binary equivalent of 42 is :")
printf("  00%s\n",x1)
printf(" The binary equivalent of 68 is :")
printf("  0%s\n",y1)
//finding 2's complement of 68
y2=bitcmp(y,7);
y2=y2+1;
z=x+y2;
//since we have subtracted a larger number from a
```

```
// smaller we sjould take  2's complement of the
   result
z1=bitcmp(z,7)
z2=z1+1;
a=dec2bin(z2)
printf(" Therefore (42)_10-(68)_10 in binary is =")
//final answer
printf("  00%s = ",a)//the final result is negative
printf(' -%d',z2)
```

Scilab code Exa 13.51 `binary nos subtraction using 1s and 2s complement`

```
//EX13_51 Pg-29
clc
clear
//subtraction of 10000 from 11010 using 1''s
   complement method
printf("  i)\n  subtraction of 10000 from 11010
   using 1''s complement method ")
printf("\n  Therefore 11010-10000 =")
x=['11010'];
y=['10000'];
//binary to decimal conversion//
x=bin2dec(x)
y=bin2dec(y)
y1=bitcmp(y,5)//one's complement of the larger
   number
z=x+y1;//addition of x with the one's complement of
   y
//subtraction of smaller number from larger number
w=bitset(z,6,0)//the end round carry should be
   remove and add to z
a=w+1;
a1=dec2bin(a)//final result
printf("   %s",a1)
```

```
x=['1000100'];
y=['1010100'];
//subtraction of 1000100 from 1010100 using 1''s
   complement method
printf("\n\n  Subtraction of 1010100 from 1000100
   using 1''s complement method ")
printf("\n  Therefore 1000100-1010100 =")
//binary to decimal conversion//
x=bin2dec(x)
y=bin2dec(y)
y1=bitcmp(y,6)//one's complement of the larger
   number
z=x+y1;//addition of x with the one's complement of
   y
//subtraction of larger number from smaller number
z=bitcmp(z,6);//one's complement of the result
a=dec2bin(z)//decimal to binary conversion
printf("  -%s\n",a)//the final result is negative

//subtraction of 10000 from 11010 using 2''s
   complement method
printf("\n\n  ii)\n  Subtraction of 10000 from 11010
    using 2''s complement method")
printf("\n  Therefore 11010-10000 =")
x=['11010'];
y=['10000'];
//binary to decimal conversion//
x=bin2dec(x)
y=bin2dec(y)
y1=bitcmp(y,6)//one's complement of the smaller
   number
y2=y1+1;//2's complement of the smaller number
//subtraction of smaller number from larger number
a=x+y2;
w=bitset(a,7,0)//we discard the carry
s=dec2bin(w)
printf("   %s",s)
//subtraction of 1000100 from 1010100 using 2''s
```

```
    complement method
printf("\n\n  Subtraction of 1010100 from 1000100
    using 2''s complement method ")
printf("\n  Therefore 1000100-1010100 =")
x=['1000100'];
y=['1010100'];
//binary to decimal conversion//
x=bin2dec(x)
y=bin2dec(y)
y1=bitcmp(y,6)//one's complement of the larger
    number
y2=y1+1;//2's complement of the larger number
//subtraction of larger number from smaller number
a=x+y2;//result is in two complement
a1=bitcmp(a,6)//one's complement of the result
a2=a1+1;//final answer
s=dec2bin(a2)
printf("  -%s",s)//the final result is negative
```

Scilab code Exa 13.52 addition of octal nos

```
//EX13_52 Pg-30
clc
clear
printf("Addition of two numbers with octal base")
printf("\n\n    4_8+2_8 = ")
x=['4'];
y=['2'];
//octal to decimal conversion
x1=oct2dec(x)
y1=oct2dec(y)
z=x1+y1;//addition
a=dec2oct(z)//final result
printf("%s\n",a)
printf("\n\n    6_8+7_8 = ")
```

```
x=['6'];
y=['7'];
//octal to decimal conversion
x1=oct2dec(x)
y1=oct2dec(y)
z=x1+y1;//addition
//we should note that the decimal sum is greater
   than 8
//hence we should subtract the decimal sum by 8 to
   obtain the octal result
a=z-8;
a=dec2oct(a)
printf("%s\n",a)
printf("\n\n    1_8+7_8 = ")
x=['1'];
y=['7'];
//octal to decimal conversion
x1=oct2dec(x)
y1=oct2dec(y)
z=x1+y1;//addition
//we should note that the decimal sum is greater
   than 8
//hence we should subtract the decimal sum by 8 to
   obtain the octal result
a=z-8;
a=dec2oct(a)
printf("%s\n",a)
```

Scilab code Exa 13.53 `octal nos addition`

```
//EX13_53 Pg-31
clc
clear
printf("Addition of two octal numbers ")
printf("\n\n    167_8+325_8 = ")
```

```
x=['167'];
y=['325'];
//octal to decimal conversion
x1=oct2dec(x)
y1=oct2dec(y)
z=x1+y1;//addition
a=dec2oct(z)//final result
printf("%s\n",a)
```

Scilab code Exa 13.54 octal nos addition

```
//EX13_54 Pg-31
clc
clear
printf("Addition of four octal numbers ")
printf("\n\n     (341)_8+(125)_8+(472)_8+(577)_8 = ")
x=['341'];
y=['125'];
u=['472']
v=['577']
//octal to decimal conversion
x1=oct2dec(x)
y1=oct2dec(y)
u1=oct2dec(u)
v1=oct2dec(v)
z=x1+y1+u1+v1;//addition
a=dec2oct(z)//final result
printf("%s\n",a)
```

Scilab code Exa 13.55 7s complement of 612

```
//EX13_55 Pg-31
clc
```

```
clear
printf("7''s complement of (612)_8 = ")
x=['612']
y=oct2dec(x)
z=bitcmp(y,8)
a=dec2oct(z)
printf("%s\n",a)
```

Scilab code Exa 13.56 subtract 157 from 176 using 7s complement

```
//EX13_56 Pg-32
clc
clear
printf("subtraction of two octal numbers using 7''s
    complement")
printf("\n (176)_8>(157)_8")
printf("\n\n Therefore    (176)_8-(157)_8 = ")
x=['176']
y=['157']
//octal to decimal conversion//
x=oct2dec(x)
y=oct2dec(y)
y1=bitcmp(y,9)//7's complement of the smaller
    number
z=x+y1;//subtraction of smaller number from larger
    number
//if there's any carry we should discard the carry
//and add the carry to the result
z=bitset(z,10,0)
z1=z+1;
a=dec2oct(z1)
printf("0%s\n",a)
```

Scilab code Exa 13.57 subtract 243 from 153 using 7s complement

```
//EX13_57 Pg-32
clc
clear
printf("subtraction of two octal numbers using 7''s
    complement")
printf("\n  (153)_8<(243)_8")
printf("\n\n Therefore (153)_8-(243)_8 =")
x=['153']
y=['243']
//octal to decimal conversion//
x=oct2dec(x)
y=oct2dec(y)
y1=bitcmp(y,9)//7's complement of the larger  number
z=x+y1;
//in the subtraction of larger number from smaller
    number
//no carry is present so we should take 7's
    complement of the result
z1=bitcmp(z,8)
a=dec2oct(z1)
printf("  -%s",a)//the final result is negative
```

Scilab code Exa 13.58 8s complement of 346

```
//EX13_58 Pg-33
clc
clear
printf("8''s complement (346)_8 is : ")
x=['346'];
y=oct2dec(x);//octal to decimal conversion//
z=bitcmp(y,9);//one's complement of the number//
z=z+1;
z2=dec2oct(z)//8's complement of the number//
```

```
printf("%s\n",z2)
```

Scilab code Exa 13.59 `subtract 413 from 516 using 8s complement`

```
//EX13_59 Pg-33
clc
clear
printf("subtraction of two octal numbers using 8''s complement")
printf("\n (516)_8>(413)_8")
printf("\n\n (516)_8-(413)_8 =")
x=['516']
y=['413']
//octal to decimal conversion//
x=oct2dec(x)
y=oct2dec(y)
y1=bitcmp(y,8)//7's complement of the smaller number
y2=y1+1;//8's complement of the smaller number
//subtraction of smaller number from larger number
a=x+y2;
w=bitset(a,10,0)//we discard the carry
s=dec2oct(w)
printf(" %s\n",s)
```

Scilab code Exa 13.60 `subtract 451 from 316 using 8s complement`

```
//EX13_60 Pg-34
clc
clear
printf("subtraction of two octal numbers using 8''s complement")
printf("\n (316)_8<(451)_8")
```

```
printf("\n\n (316)_8-(451)_8 =")
x=['316']
y=['451']
//octal to decimal conversion//
x=oct2dec(x)
y=oct2dec(y)
y1=bitcmp(y,8)//7's complement of the larger  number
y2=y1+1;//8's complement of the larger number
//subtraction of larger number from smaller number
a=x+y2;//the result obtained will have no carry
    since
//the second number is larger than the first number
//hence we should take 8's complement of the result
a1=bitcmp(a,8)//8's complement of the result
a2=a1+1;//final answer
s=dec2oct(a2)
printf("  -%s",s)//the final result is negative
```

Scilab code Exa 13.61 hexadecimal addition

```
//EX13_61 Pg-34
clc
clear
printf("Addition of two numbers with hexadecimal
    base")
printf("\n\n     3_16+9_16 =")
x=['3'];
y=['9'];
//octal to decimal conversion
x1=hex2dec(x)
y1=hex2dec(y)
z=x1+y1;//addition
a=dec2hex(z)//final result
printf(" %s\n",a)
printf("\n     9_16+7_16 =")
```

```
x=['9'];
y=['7'];
//octal to decimal conversion
x1=hex2dec(x)
y1=hex2dec(y)
z=x1+y1;//addition
//we should note that the decimal sum is greater
   than 16
//hence we should subtract the decimal sum by 16 to
   obtain the hexadecimal result
a=z-16;
a=dec2hex(a)
printf(" %s ",a)
printf('with carry of 1')
printf("\n\n     A_16+8_16 =")
x=['A'];
y=['8'];
//octal to decimal conversion
x1=hex2dec(x)
y1=hex2dec(y)
z=x1+y1;//addition
//we should note that the decimal sum is greater
   than 16
//hence we should subtract the decimal sum by 16 to
   obtain the hexadecimal result
a=z-16;
a=dec2hex(a)
printf(" %s ",a)
printf('with carry of 1')
```

Scilab code Exa 13.62 `hexadecimal addition`

```
//EX13_62 Pg-35
clc
clear
```

```
printf("Addition of two hexadecimal numbers ")
printf("\n\n     (3F8)_16+(5B3)_16 =")
x=['3F8'];
y=['5B3'];
//octal to decimal conversion
x1=hex2dec(x)
y1=hex2dec(y)
z=x1+y1;//addition
a=dec2hex(z)//final result
printf(" %s\n",a)
```

Scilab code Exa 13.63 `hexadecimal addition`

```
//EX13_63 Pg-35
clc
clear
printf("    Addition of four hexadecimal numbers")
printf("\n     Therefore (4FB)_16+(75D)_16+(A12)_16+(
   C39)_16 = ")
x=['4FB']
y=['75D']
u=['A12']
v=['C39']
x1=hex2dec(x)
y1=hex2dec(y)
u1=hex2dec(u)
v1=hex2dec(v)
z=x1+y1+u1+v1
a=dec2hex(z)
printf("%s",a)
```

Scilab code Exa 13.64 `15s complement of A9B`

```
//EX13_64 Pg-35
clc
clear
printf(" 15''s complement of (A9B)_16 =")
x=['A9B']
y=hex2dec(x)
z=bitcmp(y,11)
a=dec2hex(z)
printf(" %s\n",a)
```

Scilab code Exa 13.65 `subtraction of 98F from B02 using 15s complement`

```
//EX13_65 Pg-36
clc
clear
printf("subtraction of two hexadecimal numbers using
    15''s complement")
printf("\n        (B02)_16>(98F)_16")
printf("\n\n      (B02)_16-(98F)_16 =")
x=['B02']
y=['98F']
//hexadecimal to decimal conversion
x=hex2dec(x)
y=hex2dec(y)
y1=bitcmp(y,12)//15's complement of the smaller
   number
z=x+y1;//subtraction of smaller number from larger
   number
//if there's any carry we should discard the carry
//and add the carry to the result
z=bitset(z,13,0)
z1=z+1;
a=dec2hex(z1)
printf(" %s\n",a)
```

Scilab code Exa 13.66 subtraction of C14 from 69B using 15s complement

```
//EX13_66 Pg-36
clc
clear
printf("subtraction of two hexadecimal numbers using
    15''s complement")
printf("\n     (69B)_16<(C14)_16")
printf("\n\n     (69B)_16-(C14)_16 =")
x=['69B']
y=['C14']
//hexadecimal to decimal conversion
x=hex2dec(x)
y=hex2dec(y)
y1=bitcmp(y,10)//15's complement of the larger
   number
z=x+y1;
// in subtraction of larger number from smaller
   number
//no carry is present so we should take 15's
   complement of the result
z1=bitcmp(z,12)
a=dec2hex(z1)
printf("  -%s",a)//the final result is negative
```

Scilab code Exa 13.67 16s complement of A8C

```
//EX13_67 Pg-37
clc
clear
printf(" 16''s complement of (A8C)_16 =")
x=['A8C']
```

```
y=hex2dec(x)
z=bitcmp(y,11)
z1=z+1
a=dec2hex(z1)
printf(" %s\n",a)
```

Scilab code Exa 13.68 subtraction of 972 from CB2 using 16s complement

```
//EX13_68 Pg-37
clc
clear
printf(" Subtraction of two hexadecimal numbers
   using 16''s complement")
printf("\n (CB2)_16 >(972)_16")
printf("\n  Therefore (CB2)_16 -(972)_16 =")
x=['CB2']
y=['972']
//hexadecimal to decimal conversion
x=hex2dec(x)
y=hex2dec(y)
y1=bitcmp(y,12)//15's complement of the smaller
   number
y2=y1+1;//16's complement of the smaller number
//subtraction of smaller number from larger number
a=x+y2;
w=bitset(a,13,0)//we discard the carry
s=dec2hex(w)
printf(" %s\n",s)
```

Scilab code Exa 13.69 subtraction of 854 from 3B7 using 16s complement

```
//EX13_68 Pg-37
clc
```

```
clear
printf(" Subtraction of two hexadecimal numbers
   using 16''s complement")
printf("\n (3B7)<16-(854)_16")
printf("\n  Therefore (3B7)_16 -(854)_16 =")
x=['3B7']
y=['854']
//hexadecimal to decimal conversion
x=hex2dec(x)
y=hex2dec(y)
y1=bitcmp(y,12)//15's complement of the larger
   number
y2=y1+1;//16's complement of the larger number
//subtraction of larger number from smaller number
a=x+y2;//the result obtained will have no carry
   since
//the second number is larger than the first number
//hence we should take 16's complement of the result
a1=bitcmp(a,12)//16's complement of the result
a2=a1+1;//final answer
s=dec2hex(a2)
printf(" -%s",s)
```

Scilab code Exa 13.70 decimal addition in 8421BCD

```
//EX13_70 Pg-13.41
clc
clear
printf("Decimal addition in 8-4-2-1 BCD ")
printf("\n\n   24+18 = 42 =")
//given decimal number
x=24;
y=18;
//we separate each of the digit in the decimal
   number
```

```
x1=2;   x2=4;
y1=1;   y2=8;
//then we add each of the digit together
//ie x1+y1 and x2+y2
z1=x1+y1;    z2=x2+y2;

a=9;                  //since 9_(decimal)=1001_(binary)
if (z1>a)
  z1=z1+6;            //since 6_(decimal)=0110_(binary
    )
end

if (z2>a) then
   z2=z2+6;
//if any carry is present in z2
//we should add the carry with z1
m=bitget(z2,5)
if (m==1) then
z1=z1+m
z2=bitset(z2,5,0)
end
end
z1=dec2bin(z1)
z2=dec2bin(z2)
printf(" 0%s 00%s",z1,z2)

printf("\n\n   48+58 = 106 =")
//given decimal number
x=48;
y=58;
//we separate each of the digit in the decimal
   number
x1=4;   x2=8;
y1=5;   y2=8;
//then we add each of the digit together
//ie x1+y1 and x2+y2
z3=0;   z1=x1+y1;    z2=x2+y2;

```

```
//if there is any carry is present in z2
// during addition then it is added to z1
m=bitget(z2,5)
if (m==1) then
z1=z1+m;
//if z2 is greater than 9(=1001 in binary )
//then we should add 6(=0110 in binary )
if (z2>9) then
z2=bitset(z2,5,0)
z2=z2+6;
end
end

//agian if z1 is greater than 9(=1001 in binary )
   then 6(=0110 inbinary) is added
if (z1>9) then
z1=z1+6;
// if any carry is present is present in z1 then we
   should add the carry with z3
m=bitget(z1,5)
if (m==1) then
z3=z3+m
z1=bitset(z1,5,0)
end
end
//conversion into binary
z4=dec2bin(z3)
z5=dec2bin(z1)
z6=dec2bin(z2)
printf(" 000%s 000%s 0%s",z4,z5,z6)

printf("\n\n   175+326 = 501 =")
//given decimal number
x=175;
y=326;
//we separate each of the digit in the decimal
   number
x1=1;   x2=7;   x3=5;
```

```
y1=3;   y2=2;   y3=6
//then we add each of the digit together
//ie x1+y1 and x2+y2  x3+y3
z1=x1+y1;   z2=x2+y2;   z3=x3+y3;
//if z1,z2,z3>9(=1001 in binary ) then 6(=0110
    inbinary) is added
if (z1>9) then
z1=z1+6;
end
if (z2>9) then
z2=z2+6;
end
if (z3>9) then
z3=z3+6;
end
//if there is any carry is present in z3 during
    addition then it is added
//to z2 and in turn is there is any carry in z2 it
    is added to z1
m=bitget(z3,5)
if (m==1) then
z2=z2+m
if (z3>9) then
z3=bitset(z3,5,0)
end
end
//if z2  is greater than 9 then we add 6 to z2
if (z2>9) then
z2=z2+6;
end
//if there is any carry present in z2 during
    addition then it is added to z1
m=bitget(z2,5)
if (m==1) then
z1=z1+m
if (z2>9) then
z2=bitset(z2,5,0)
end
```

```
end
//conversion into binary
z4=dec2bin(z1)
z5=dec2bin(z2)
z6=dec2bin(z3)
printf("    0%s 000%s 000%s",z4,z5,z6)

printf("\n\n    589+199 = 788 =")
//given decimal number
x=589;
y=199;
//we separate each of the digit in the decimal number
x1=5;   x2=8;   x3=9;
y1=1;   y2=9;   y3=9;
//then we add each of the digit together
//ie x1+y1 and x2+y2  x3+y3
z1=x1+y1;    z2=x2+y2;   z3=x3+y3;

//if there is any carry is present in z3 during addition then it is added
//to z2 and in turn is there is any carry in z2 it is added to z1
//for z2
m=bitget(z2,5)
if (m==1) then
z1=z1+m
//if z2 is greater than 9(=1001 in binary )
//then we should add 6(=0110 in binary )
if (z2>9) then
z2=bitset(z2,5,0)
z2=z2+6;
end
end

//for z3
m=bitget(z3,5)
if (m==1) then
```

```
z2=z2+m
//if z3 is greater than 9(=1001 in binary )
//then we should add 6(=0110 in binary )
if (z3>9) then
z3=bitset(z3,5,0)
z3=z3+6;
end
end

//conversion into binary
z4=dec2bin(z1)
z5=dec2bin(z2)
z6=dec2bin(z3)
printf(" 0%s %s %s",z4,z5,z6)
```

Scilab code Exa 13.71 decimal subtraction in 8421BCD using 9s complement method

```
//EX13_71 Pg-13.43
clc
clear
printf(" Decimal subtraction in 8-4-2-1 BCD using
   9''s complement method")
printf("\n\n    79-26 = 53 =")
//given decimal number
x=79;
y=26;
//we separate each of the digit in the decimal
   number
x1=7;   x2=9;
y1=2;   y2=6;
//first we take 9's complement of the second number
// ie 2 and 6
y1=9-y1         //9's complement of y1
y2=9-y2         //9's complement of y2

```

```
//then we add each of the digit of the number together
//ie x1+y1 and x2+y2
z1=x1+y1;    z2=x2+y2;
//if z2>9(=1001 in binary) then we should add
//6(=0110 in binary) to z2 and add the carry to z1
if (z2>9) then
   z2=z2+6;
   m=bitget(z2,5)
if (m==1) then
z1=z1+m
z2=bitset(z2,5,0)
end
end

//again if z1>9(=1001 in binary) then we should add the
//6(=0110 in binary) to z1 and add the carry to z2

if (z1>9) then
   z1=z1+6;
   m=bitget(z1,5)
if (m==1) then
z2=z2+m
z1=bitset(z1,5,0)
end
end
//decimal to binary conversion
z3=dec2bin(z1)
z4=dec2bin(z2)

printf(" 0%s 00%s ",z3,z4)

printf("\n\n   89-54 = 35 =")
//given decimal number
x=89;
y=54;
//we separate each of the digit in the decimal
```

```
    number
x1=8;   x2=9;
y1=5;   y2=4;
//first we take 9's complement of the second number
// ie 2 and 6
y1=9-y1         //9's complement of y1
y2=9-y2         //9's complement of y2

//then we add each of the digit of the number
    together
//ie x1+y1 and x2+y2
z1=x1+y1;    z2=x2+y2;
//if z2>9(=1001 in binary) then we should add
//6(=0110 in binary) to z2 and add the carry to z1
if (z2>9) then
   z2=z2+6;
   m=bitget(z2,5)
if (m==1) then
z1=z1+m
z2=bitset(z2,5,0)
end
end

//again if z1>9(=1001 in binary) then we should add
    the
//6(=0110 in binary) to z1 and add the carry to z2

if (z1>9) then
   z1=z1+6;
   m=bitget(z1,5)
if (m==1) then
z2=z2+m
z1=bitset(z1,5,0)
end
end
//decimal to binary conversion
z3=dec2bin(z1)
z4=dec2bin(z2)
```

```
printf(" 00%s 0%s ",z3,z4)
```

Chapter 14

Digital Logic

Scilab code Exa 14.1 Representation of the given boolean expression

```
//EX14_1 Pg-14.25
clc
clear
printf("    (((A+B)C)'')D")
printf("\n     Given inputs are A,B,C and D")
printf("\n\n     (A+B)C then we take the complement ")
printf("\n     ((A+B)C)''")
printf("\n     Then we AND together with D")
printf("\n     Therefore Y = (((A+B)C)'')D")
```

This code can be downloaded from the website wwww.scilab.in

Scilab code Exa 14.2 Implementation of EXNOR gate using NAND gate

```
//example 14.2 PG-14.27//
clc
```

```
clear
printf("     Implementation of EX-OR gate using NAND
   gate")
printf("  \n      Refer to the figure-14.45(a) shown"
   )
printf("\n     The Boolean expression for EX-OR gate
    is Y=AB''+A''B")
```

This code can be downloaded from the website wwww.scilab.in

Scilab code Exa 14.3 `Implementation of EXNOR gate using NOR gate`

```
//example 14.3 PG-14.27//
clc
clear
printf("     Implementation of EX-NOR gate using NOR
   gate")
printf("  \n      Refer to the figure-14.46(a) shown"
   )
printf("\n     The Boolean expression for EX-NOR
   gate is Y=AB+A''B''")
printf(" \n      Y=(AB''+A''B)''\n      Y= (AB'')''.(A
   ''B)''")
printf("\n      Y=(A''+B).( A+B'')")
```

This code can be downloaded from the website wwww.scilab.in

Scilab code Exa 14.4 `Boolean Solutions`

```
//example 14.4 PG-14.38
clc
```

```
clear
printf("    given=> A.A''C = 0.C.....   Since A.A''=0\n\n")
printf("                A.A''C = 0 ......   Since A.0=0")
```

Scilab code Exa 14.5 Boolean Solutions

```
//example 14.5 PG-14.38
clc
clear
printf("  Given=> ABCD+ABD = ABD(C+1)..... Distributive property\n\n")
printf("              ABCD+ABD = ABD.1........ Since A+1=1\n\n")
printf("              ABCD+ABD = ABD ......... Since A.1=A")
```

Scilab code Exa 14.6 Boolean Solutions

```
//example 14.6 PG-14.38
clc
clear
printf("  Given=> ABCD+AB''CD = ACD(B+B'')   ....... Distributive property\n\n")
printf("              ABCD+AB''CD = ACD.1   ........ .. Since A+A''=1\n\n")
printf("              ABCD+AB''CD = ACD      .......... Since A.1=A")
```

Scilab code Exa 14.7 Boolean Solutions

```
//example 14.7 PG-14.38
clc
clear
printf("   Given=> A(A+B) = AA+AB   ....... Distributive property\n\n")
printf("              A(A+B) = A+AB   ........Since A.A=A\n\n")
printf("              A(A+B) = A(1+B)  ....... Distributive property\n\n")
printf("              A(A+B) = A  ...........  Since A+1=1\n\n")
```

Scilab code Exa 14.8 `Boolean Solutions`

```
//example 14.8 PG-14.38
clc
clear
printf("   Given=> AB+ABC+AB(D+E)  = AB(1+C+D+E) .......Distributive property\n\n")
printf("              AB+ABC+AB(D+E)  = AB ...........  Since A+1=1\n\n")
```

Scilab code Exa 14.9 `Boolean Solutions`

```
//example 14.9 PG-14.38
clc
clear
printf("Given=> XY+XYZ+XYZ''+X''YZ = XY(1+Z)+XYZ''+X''YZ  ......Distributive property\n\n")
printf("             XY+XYZ+XYZ''+X''YZ = XY+XYZ'' +  X''YZ   ........Since A+1=1\n\n")
printf("             XY+XYZ+XYZ''+X''YZ = XY(1+Z'') + X''YZ   \n\n")
```

```
printf("           XY+XYZ+XYZ''+X''YZ = XY + X''YZ ............Since A+1=1\n\n")
printf("           XY+XYZ+XYZ''+X''YZ = Y(X+X''Z)       \n\n")
printf("           XY+XYZ+XYZ''+X''YZ = Y(X+Z) ...............Since A+A''B=A+B")
```

Scilab code Exa 14.10 Boolean Solutions

```
//example 14.10 PG-14.39
clc
clear
printf("Given=> A''B''C''+A''BC''+A''BC = A''C''(B''+B)+A''BC ......Distributive property")
printf("\n\n           A''B''C''+A''BC''+A''BC = A''C''+ A''BC    .........Since A+A''=1\n\n")
printf("           A''B''C''+A''BC''+A''BC = A''(C''+BC)     .........Distributive property")
printf("\n\n           A''B''C''+A''BC''+A''BC = A''(C''+B)   ..............Since A+A''B=A+B")
```

Scilab code Exa 14.11 Boolean Solutions

```
//example 14.11 PG-14.39
clc
clear
printf("      Given=> LHS = ABC+AB''C+ABC''\n\n")
printf("                  = AC(B+B'') + ABC'' .......Distributive property\n\n")
printf("                  = AC.1 + ABC'' .........Since A+A''=1\n\n")
printf("                  = AC + ABC'' ...........Since A.1=1\n\n")
```

```
printf("                              = A(C+BC'')
    ..............Distributive property\n\n")
printf("                              = A(C+B)
    ............Since A+A''B=A+B\n\n")
printf("Therefore \n")
printf("      ABC+AB''C+ABC''  = A(C+B)")
```

Scilab code Exa 14.12 Boolean Solutions

```
//example 14.12 PG-14.39
clc
clear
printf("   Given=> \n")
printf("             A''BCD''+ BCD''+ BC''D''+ BC''D\n\
   n")
printf("                          = BCD''(A''+1)+ BC''D''+ BC''
   D  .......Distributive property\n\n")
printf("                          = BCD''+ BC''D''+ BC''D
    ...........Since A+1=1\n\n")
printf("                          = BD''(C+C'')+ BC''D
    ..........Distributive property\n\n")
printf("                          = BD''+ BC''D
    ...............Since A+A''=1\n\n")
printf("                          = B(D''+C''D)
    .............Distributive property\n\n")
printf("                          = B(D''+C'')
    ..............Since A+A''B=A+B\n\n")
printf("Therefore \n")
printf("             A''BCD''+ BCD''+ BC''D''+ BC''D =
   B(D''+C'')")
```

Scilab code Exa 14.13 Boolean Solutions

```
//example 14.13 PG-14.39
clc
clear
printf("   Given=> \n")
printf("              AC+C(A+A''B) = ")
printf("AC+AC+A''BC       ........Distributive property
   \n\n")
printf("                                  = AC + A''BC
   .........Since A+A=A\n\n")
printf("                                  = C(A+A''B)
   .........Distributive property\n\n")
printf("                                  = C(A+B)
   ..........Since A+A''B=A+B\n\n")
printf(" Therefore \n")
printf("              AC+C(A+A''B) = C(A+B)")
```

Scilab code Exa 14.14 Boolean Solutions

```
//example 14.14 PG-14.39
clc
clear
printf("\n\n Given=> \n\n")
printf("    A''BC''D + A''BCD + ABD = ")
printf("A''BD(C''+C)+ ABD       ........Distributive
   property\n\n")
printf("                                     = A''BD + ABD
   .............Since A+A=A\n\n")
printf("                                     = BD(A''+A)
   ..........Distributive property\n\n")
printf("                                     = BD
   .................Since A+A''=1\n\n")
printf(" Therefore \n\n")
printf("    A''BC''D + A''BCD + ABD = BD")
```

Scilab code Exa 14.15 `Boolean Solutions`

```
//example 14.15 PG-14.39
clc
clear
printf("\n\n Given=> \n")
printf("        LHS = A + A''B + AB''\n\n")
printf("            = (A + A''B) + AB''\n\n")
printf("            = A + B + AB''   ..........Since A+A''B=A+B\n\n")
printf("            = A + A + B      ..........Since A+A''B=A+B\n\n")
printf("            = A + B      .............Since A+A=A\n\n")
printf(" Therefore \n\n")
printf("        A + A''B + AB'' = A + B")
```

Scilab code Exa 14.16 `Boolean Solutions`

```
//example 14.16 PG-14.40
clc
clear
printf("\n\n Given=> \n\n")
printf("             ((AB)''+A''+AB)'' = ")
printf("(A''+ B''+ A''+ AB)''       ....Since (AB)''=A''+B''\n\n")
printf("                              = (A''+ B''+ A''+ B)''    .......Since A+A''B=A+B\n\n")
printf("                              = (A''+ B''+ B)''    ...........Since A+A=A\n\n")
printf("                              = (A''+ 1)''    ...............Since A+A''=1\n\n")
```

```
printf("                              = 1'' .................Since A+1=1\n\n")
printf("                              = 0 \n\n")
printf("   Therefore \n\n")
printf("             (A''B''+A''+AB)'' = 0")
```

Scilab code Exa 14.17 `Boolean Solutions`

```
//example 14.17 PG-14.40
clc
clear
printf("\n\n Given=> \n\n")
printf("   AB+(AC)''+AB''C(AB+C) = ")
printf("AB+ (AC)''+AAB''BC+AB''CC   ......Distributive property\n\n")
printf("                            = AB+ (AC)''+ AB''CC .............Since A.A''=0\n\n")
printf("                            = AB+ (AC)''+ AB''C ...........Since A.A=A\n\n     ")
printf("                      = AB+ A''+ C''+ AB''C ..........Since(AB)''=A''+B''\n\n")
printf("                            = A''+ B+C''+ AB''C ............Since A+A''B=A+B\n\n")
printf("                            = A''+ AB''C + B+C'' ...........Commutative property")
printf("\n\n                            = A''+ B''C + B+C'' ............Since A+A''B=A+B")
printf("\n  Where B=B''C\n\n")
printf("                            = A''+ B+C''+ B''C ...........Commutative property")
printf("\n\n                            = A''+ B+C''+B'' ............Since A+A''B=A+B")
printf("\n\n                            = A''+C''+1
```

```
    .................Since A+A''=1\n\n")
printf("                              = 1
    .....................Since A+1=1\n\n")
printf(" Therefore \n")
printf("             AB+(AC)''+AB''C(AB+C) = 1")
```

Scilab code Exa 14.18 Boolean Solutions

```
//example 14.18 PG-14.40
clc
clear
printf("   Refer to the Figure-14.50 shown\n\n")
printf("    The Boolean expression for the output Y
    is :\n\n")
printf("    Y = (A''+B'')''.BC\n\n")
printf("    Y = ((AB)'')''.BC      ....Since A''+B
    ''=(AB)'' and (A'')''= A\n")
printf("                                  DeMorgan''s
    Therem\n\n")
printf("    Y = A.B.B.C       ........Since A.A=A\n\n"
    )
printf("    Y = ABC\n\n")
printf("    Truth Table\n")
printf("    A      B      C      Y")
a=zeros(1,4)
b=[0 0 1 0;0 1 0 0;0 1 1 0;1 0 0 0;1 0 1 0;1 1 0 0]
c=ones(1,4)
d=[a;b;c]
disp(d)
```

Scilab code Exa 14.19 construction of the given logic circuit using AND OR and INV

```
//example 14.19 PG-41
```

```
clc
clear
printf("a)          Given  =>\n")
printf("              Y  =  (AB(C+D))''\n\n")
printf("b)          Given  =>\n")
printf("              Z  =  (W+PQ'')''")
```

This code can be downloaded from the website wwww.scilab.in This code can be downloaded from the website wwww.scilab.in

Scilab code Exa 14.20 `truth table of the given logic expression`

```
//example 14.20 PG-14.41
clc
clear
printf("    Refer  to  the  Figure-14.52  shown\n\n")
printf("     From  the  Figure  we  can  see  that\n\n")
printf("     Y  =  ((A+B)''.(B+C'')'')''\n\n")
printf("     Y  =  (A+B)''''+(B+C'')''''        .....Since
   (A.B)''=A''+B''\n")
printf("                                              DeMorgan''s
    Therem  \n")
printf("                                              where  A=A+B
    and  B=B+C''\n\n")
printf("     Y  =  A+B+B+C''              ..........Since  (A
   '')''=  A\n\n")
printf("     Y  =  A+B+C''                ..........Since  A+A=
    A\n\n")
printf("     Truth  Table\n")
printf("     A      B      C      Y")
a=[0 0 0 1;0 0 1 0;0 1 0 1;0 1 1 1;1 0 0 1;1 0 1 1;1
    1 0 1]
```

```
b=ones(1,4)
c=[a;b]
disp(c)
```

Scilab code Exa 14.21 simplification of the given logic expression

```
//example 14.21 PG-14.42
clc
clear
printf("\n\n Given=> \n\n")
printf("     Y = (A+B)(A''+C)(B''+C'')\n\n")
printf("     Y = (AA''+ AC +A''B + BC)(B''+C'')\n\n")
printf("     Y = (AC +A''B + BC)(B''+C'')     .......
   Since A.A''= 0\n\n")
printf("     Y = AB''C + ACC''+ A''BB''+ A''BC''+ BB
   ''C+ BCC''\n\n")
printf("     Y = AB''C + A''BC''       Since A.A'' = 0"
   )
```

Scilab code Exa 14.22 simplification of the given logic expression using Demorgans

```
//example 14.22 PG-14.41
clc
clear
printf("\n    a) Given=> \n\n")
printf("    ((A+B'').(A''+B))'' = (A+B'')''+(A''+B)''
          ....Since (AB)''= A''+B''\n")
printf("

   DeMorgan''s Therem\n\n")
printf("                          = A''.B''''+A''''.B''
           ....Since A''+B''=(AB)''\n")
```

```
printf("

   DeMorgan''s Therem\n\n")
printf("                              = A''.B + A.B''
   .......Since (A'')''=A\n")
printf("  Therefore \n")
printf("   ((A+B'').(A''+B))'' = A''.B + A.B''")

printf("\n\n  b) Given=> \n\n")
printf("   (((A.B)'')C)''D)'' = ((AB)'')C)''''+D''
            ....Since (AB)''= A''+B''\n")
printf("

   DeMorgan''s Therem\n\n")
printf("                              = (AB)''.C + D''
   ....Since (A'')''=A\n\n")
printf("                              = (A''+B'')C + D''
               ....Since (AB)''=A''+ B''\n")
printf("

   DeMorgan''s Therem\n\n")
printf(" Therefore \n")
printf("   (((A.B)'')C)''D)'' = (A''+B'')C + D'' ")
```