Scilab Textbook Companion for Electronics Circuits and Systems by Y. N. Bapat[1]

Created by
Komal
B.Tech
Electronics Engineering
UPTU/ABES Engg. College Gzb
College Teacher
None
Cross-Checked by
Chaya Ravindra

December 29, 2014

[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: Electronics Circuits and Systems

Author: Y. N. Bapat

Publisher: Tata McGraw-Hill, New Delhi

Edition: 1

Year: 2000

ISBN: 0-07-460040-0

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

Chapter 1

Vacuum tubes and semiconductors

Scilab code Exa 1.1 small signal parameters

```
//Ex_1.1
//refer to fig 1.2(c) and given d.c operating points
    VGKQ=-2 V,VAKQ=250 V,IAQ=-1.2 mA
clc
VAK2=300
disp("VAK2 = "+string(VAK2)+" volts") // value of
    anode voltage2
VAK1=170
disp("VAK1 = "+string(VAK1)+" volts") // value of
    anode voltage1
IA2=2*10^(-3)
disp("IA2 = "+string(IA2)+" ampere") // value of
    anode current2
IA1=0*10^(-3)
disp("IA1 = "+string(IA1)+" ampere") // value of
    anode current1
rP=(VAK2-VAK1)/(IA2-IA1)//anode resistance at VGK=
    VGKQ
disp("resistance,rP =(VAK2-VAK1)/(IA2-IA1)="+string(
```

```
rP)+" ohm") //calculation
VGK2=-2.5
disp("VGK2 = "+string(VGK2)+" volts") // value of grid voltage2
VGK3=-1.5
disp("VGK1 = "+string(VGK3)+" volts") // value of grid voltage1
VAK3=200
disp("VAK3 = "+string(VAK3)+" volts") // value of anode voltage1
u=(VAK2-VAK3)/(VGK2-VGK3)//amplification factor at IA=IAQ
disp("amplification factor,u =(VAK2-VAK1)/(VGK2-VGK1)="+string(u)+" unitless ") //calculation
IA4=2.2*10^(-3)
disp("IA4 = "+string(IA4)+" ampere") // value of anode current4
IA3=0.5*10^(-3)
disp("IA1 = "+string(IA3)+" ampere") // value of anode current1
gm=(IA4-IA3)/(VGK2-VGK3)// transconductance at VAK=VAKQ
disp("transconductance,gm =(IAK4-IAK3)/(VGK2-VGK3)="+string(gm)+" ampere/volt ") //calculation
//mistake of negative sign for answers for u(amplification factor) and gm(transconductance)in book
```

Scilab code Exa 1.2 CRT parameters

```
//Ex1_2
clc
d=0.5*10^(-2)
disp("d = "+string(d)+"metre") //initializing value of distance b/w plates
```

```
l=2*10^(-2)
disp("l = "+string(l)+"metre") //initializing value of length of plates
L=20*10^(-2)
disp("L = "+string(L)+"metre") //initializing value of distance b/w centre of plates and screen
Va=2000
disp("Va = "+string(Va)+"volts") ////initializing value ofanode voltage
Vd=100
disp("Vd = "+string(Vd)+"volts") //initializing value of deflecting voltage
m=9.11*10^(-31)
disp("m = "+string(m)+"Kg") //mass of electron
q=1.6*10^(-19)
disp("q = "+string(q)+"coulomb") //charge on an electron
disp("horizontal beam velocity,Vx =(2*Va*q/m)^(0.5) metre/second") //formula
Vx =(2*Va*q/m)^(0.5)
disp("horizontal beam velocity,Vx =(2*Va*q/m)^(0.5)= "+string(Vx)+" metre/second")//calculation
disp("transit time,t1 =(l/Vx) second") //formula
t1=(l/Vx)
disp("transit time,t1 =(l/Vx)= "+string(t1)+" second")//calculation
disp("vertical beam velocity,Vy =(q*Vd*l/d*m*Vx) metre/second") //formula
Vy=((q*Vd*l)/(d*m*Vx))
disp("vertical beam velocity,Vy =(q*Vd*l/d*m*Vx)= "+string(Vy)+" metre/second")//calculation
disp("vertical displacement,D =((l*L*Vd)/(2*d*Va) metre")//formula
D =(l*L*Vd)/(2*d*Va)
disp("vertical displacement,D =((l*L*Vd)/(2*d*Va)="+string(D)+" metre")//calculation
disp("sensitivity of CRT,S =(0.5*l*L)/(d*Va) metre/volt")//formula
```

```
S =(0.5*l*L)/(d*Va)
disp("sensitivity of CRT,S =(0.5*l*L)/(d*Va)="+
    string(S)+" metre/volt")//calculation
```

Scilab code Exa 1.3 Find deflection of electron beam

```
//Ex1-3
clc
m=9.11*10^(-31)
disp("m = "+string(m)+" Kg") //mass of electron
q=1.6*10^(-19)
disp("q = "+string(q)+" coulomb") //charge on an
    electron
B=1.5*10^(-3)
disp("B = "+string(B)+ " wb/m^2") //initializing
    value of magnetic field
l=5*10^(-2)
disp("l = "+string(l)+" metre") //initializing axial
     length of magnetic field
L=30*10^(-2)
disp("L = "+string(L)+" metre") //initializing value
     of distance of screen from centre of magnetic
    field
Va=10000
disp("Va = "+string(Va)+" volts") ////initializing
    value of anode voltage
disp("horizontal beam velocity,Vx =(2*Va*q/m)^(0.5)
    metre/second") //formula
Vx =(2*Va*q/m)^(0.5)
disp("horizontal beam velocity,Vx =(2*Va*q/m)^(0.5)=
     "+string(Vx)+" metre/second")//calculation
disp("radius,r =(m*Vx)/(B*q) metre") //formula
r =(m*Vx)/(B*q)
disp("radius,r =(m*Vx)/(B*q)= "+string(r)+" metre")
    //calculation
```

```
disp("deflection ,D =(L*l)/r) metre")//formula
D =(L*l)/r
disp("deflection ,D =(L*l)/r)="+string(D)+" metre")//calculation
```

Scilab code Exa 1.4 calculate drift velocity

```
//Ex-1.4
clc
q=1.6*10^(-19)
disp("q = "+string(q)+"coulomb") //charge on an electron
I=10
disp("I = "+string(I)+"Ampere") //initializing value of current
r=64.25
disp("radius ,r = "+string(r)+" mils")//initializing value of radius of wire
function[metres]=mils2metres(mils)
metres=(mils*2.54)/(1000*100)
endfunction
[r1]=mils2metres(r)
disp("r1 = "+string(r1)+" metre")
n=5*10^(28)
disp("n = "+string(n)+" electrons/m^3") // electrons concentration in copper
A=(%pi*r1^2) //formulae
disp("cross sectional area ,A =(%pi*r1^2)= "+string(A)+" square metre")//calculation
v=(I)/(A*q*n)//formulae(I=A*q*n*v)
disp("drift velocity ,v=(I)/(A*q*n)="+string(v)+" metre/second")//calculation
```

Scilab code Exa 1.5 calculation of resistance

```
//Ex1_5
clc
A=10*10^(-6); p1=10^(-4); p2=10^(3);p3=10^(10);l
    =1*10^(-2); //initializations
disp("cross sectional area,A ="+string(A)+"merer
    square")
disp("resitivity(rho),p1 ="+string(p1)+" ohm-m")
disp("resitivity(rho),p2 ="+string(p2)+" ohm-m")
disp("resitivity(rho),p3 ="+string(p3)+" ohm-m")
disp("conductor length,l ="+string(l)+" metre")
disp(" resistance for copper,R = p1*l/A = "+string(
    p1*l/A)+"ohm") //calculations for copper
disp(" resistance for silicon,R = p2*l/A = "+string(
    p2*l/A)+"ohm") //calculations for silicon
disp(" resistance for glass,R = p3*l/A = "+string(p3
    *l/A)+"ohm") //calculations for glass
```

Scilab code Exa 1.6 parameters for Silicon atoms

```
//Ex1_6
clc
ni = 1.45*10^10 //initializations
nV = 5*10^22 //initializations
un = 1500 //initializations
up = 0475//initializations
T = 300 //initializations
I=10^(-6)
disp("I = "+string(I)+"Ampere") //initializing value
     of current
A=(50*10^(-4))^2; l=0.5 //initializations
q=1.59*10^(-19) //charge on an electron
disp("intrinsic charge concentration,ni = "+string(
    ni)+" /centimetre cube")
```

```
disp("silicon atoms concntration, nV = "+string(nV)+" /centimetre cube ")

disp("electron mobility,un = "+string(un)+" cm.sq/V-s")
disp("hole mobility,up = "+string(up)+"cm.sq/V-s")
disp("temperature,T = "+string(T)+"K")
disp("q = "+string(q)+"coulomb") //charge on an electron
disp("cross sectional area,A ="+string(A)+"cm square ")
disp("conductor length,l ="+string(l)+"cm")
 N=nV/ni
disp("relative concentration,N =nV/ni= "+string(N)+" silicon atoms per electron -hole pair") //calculation
sigma=(1.59*10^(-19)*(1.45*10^10)*(1500+0475))
disp("intrinsic conductivityi,sigma =(1.59*10^(-19)*(1.45*10^10)*(1500+0475))= "+string(sigma)+" (ohm-cm)^-1") //calculation
pi =(1/sigma)//formulae
disp("resitivity(rho),pi =(1/sigma)="+string(pi)+" ohm-cm")//calculation
R=(2.22*10^5*0.5)/0.000025
disp(" resistance for silicon,R =((2.22*10^5*0.5)/0.000025) = "+string(R)+" ohm") //calculations for silicon
V=I*R
disp(" voltage drop,V =I*R = "+string(V)+" V") //calculations
```

Scilab code Exa 1.7 N type Silicon parameters

```
//Ex1_7
clc
```

```
ni = 1.45*10^10 //initializations
nV = 5*10^22 //initializations
un = 1500 //initializations
up = 0475//initializations
T = 300 //initializations
I=10^(-6)
disp("I = "+string(I)+"Ampere") //initializing value of current
A=(50*10^(-4))^2; l=0.5 //initializations
q=1.59*10^(-19) //charge on an electron
disp("intrinsic charge concentration,ni = "+string(ni)+" /centimetre cube")
disp("silicon atoms concntration, nV = "+string(nV)+" /centimetre cube ")

disp("electron mobility,un = "+string(un)+" cm.sq/V-s")
disp("hole mobility,up = "+string(up)+" cm.sq/V-s")
disp("temperature,T = "+string(T)+" K")
disp("q = "+string(q)+"coulomb") //charge on an electron
disp("cross sectional area,A ="+string(A)+" cm square")
disp("conductor length,l ="+string(l)+" cm")
nD=nV/10^6//formulae
disp("donor concentration,nD= nV/10^6="+string(nD)+" /cm.cube")//calculation
nn=nD//formulae
disp("resulting mobile electron concentration,nn= nD ="+string(nn)+" /cm.cube")//calculation
pn= ni^2/nD//formulae
disp("resulting hole concentration,pn= ni^2/nD="+string(pn)+" /cm.cube")//calculation
sigma=q*nD*un//formulae
disp("n-type semiconductor conductivity,sigma=q*nD*un= "+string(sigma)+" (ohm-cm)^-1") //calculation
pn =(1/sigma)
disp("doped silicon resitivity(rho),pn =(1/sigma)="+
```

```
    string(pn)+" ohm-cm")//calculation
R=(0.084*0.5)/A
disp(" resistance for silicon ,R =((0.084*0.5)/A) = "
    +string(R)+" ohm") //calculations for silicon
V=I*R
disp(" voltage drop ,V =I*R = "+string(V)+" V") //
    calculations
```

Scilab code Exa 1.8 diffusion hole current

```
//Ex1_8
clc
q=1.59*10^(-19) //charge on an electron
disp("q = "+string(q)+"coulomb") //charge on an
    electron
d=0.037
disp("dimension of semiconductor ,d="+string(d)+" cm"
    )
A=(d^2) //area formulae for square shaped
    semiconductor
disp("cross sectional area ,A =d^2="+string(A)+" cm
    square")
x=10^(-4)
disp("thickness ,x ="+string(x)+" cm")
x0=0
disp("thickness ,x0 ="+string(x0)+" cm")
p=9.22*10^(15)//
disp("hole concentration at x,p= "+string(p)+" /cm-
    cube")//calculation
p0=0//
disp("hole concentration at x0,p0= "+string(p0)+" /
    cm-cube")//calculation
dp=(p-p0)//formulae
dx=(x-x0)//formulae
disp(" change in concentration at ,dp= "+string(dp)+
```

```
    " /cm-cube")//calculation
disp("change in thickness,dx= "+string(dx)+" cm")//
    calculation
(dp/dx)==(p-p0)/(x-x0)//formulae
disp(" slope,(dp/dx) =(p-p0)/(x-x0)="+string(dp/dx)+
    " holes/cm-cube")//calculation
Dp=12
disp("hole diffusion constant,Dp= "+string(Dp)+" cm-
    sq/s")//calculation
Ip=A*q*Dp*(dp/dx)
disp(" hole diffusion current,Ip =A*q*Dp*(dp/dx)="+
    string(Ip)+" ampere")//calculation
```

Chapter 2

The Semiconductor diode

Scilab code Exa 2.1 calculate diode current

```
//Ex2_1
clc
IR=50*10^(-9)
disp("IR = "+string(IR)+" ampere") // value of Reverse saturation current
VT=26*10^(-3)
disp(" Thermal voltage,VT= "+string(VT)+"volt")
VAK1=(-0.25)// diode junction voltage
disp("Junction voltage,VAK1="+string(VAK1)+"volt")
IA =IR*[exp(VAK1/(2*VT))-1]// formulae for diode current
disp("Diode current,IA =IR*(exp(VAK1/(2*VT))-1)= "+string(IR*(exp(VAK1/(2*VT))-1))+" ampere") // calculation
VAK2=(+0.25)
disp("Junction voltage,VAK2="+string(VAK2)+"volt")
IA =IR*[exp(VAK2/(2*VT))-1]
disp("Diode current,IA =IR*(exp(VAK2/(2*VT))-1)= "+string(IA)+" ampere") // calculation
VAK3=(+0.5)
disp("Junction voltage,VAK3="+string(VAK3)+"volt")
```

```
disp("Diode current,IA =IR*(exp(VAK3/(2*VT))-1)= "+string(IR*(exp(VAK3/(2*VT))-1))+" ampere") //calculation
VAK4=(+0.6)
disp("Junction voltage,VAK4="+string(VAK4)+"volt")
disp("Diode current,IA =IR*(exp(VAK4/(2*VT))-1)= "+string(IR*(exp(VAK4/(2*VT))-1))+" ampere") //calculation
VAK5=(+0.7)
disp("Junction voltage,VAK3="+string(VAK5)+"volt")
disp("Diode current,IA =IR*(exp(VAK5/(2*VT))-1)= "+string(IR*(exp(VAK5/(2*VT))-1))+" ampere") //calculation
VAK6=(+0.8)
disp("Junction voltage,VAK3="+string(VAK6)+"volt")
disp("Diode current,IA =IR*(exp(VAK6/(2*VT))-1)= "+string(IR*(exp(VAK6/(2*VT))-1))+" ampere") //calculation
```

Scilab code Exa 2.2 Diode current and diode junction voltage calculation

```
//Ex2_2
//refer to fig 2.1(c) ,fig.2.3(b) and fig.2.3(c)
clc
VF=5
disp("source voltage,VF = "+string(VF)+ " volts")//initialization
VD=0.7
disp("voltage drop,VD = "+string(VD)+ " volts")//initialization
R=5*10^(3)
disp("resistance,R = "+string(R)+ "ohm")//initialization
RF=100
disp("resistance,R = "+string(RF)+ "ohm")//
```

```
    initialization
VR=0.6
disp("VR = "+string(VR)+ " volts")//initialization
IA=(VF-VD)/R //formulae
disp("Diode current ,IA = "+string(IA)+" ampere") //
     calculation
IA=(VF-VR)/(R+RF)// Formulae
disp(" using large signal model,IA = "+string(IA)+"
    ampere") // calculation
VAK=(VR+IA*RF)// Formulae
disp("Junction voltage ,VAK ="+ string(VAK)+" volts")
    //calculation
```

Scilab code Exa 2.3 small signal forward and reverse resistance

```
//Ex2_3
clc
VT=26*10^(-3)
disp(" Thermal voltage ,VT= "+string(VT)+" volt")//
    initialization
IR=50*10^(-9)
disp("IR = "+string(IR)+" ampere") // value of
    Reverse saturation current
VAK1=(0.7)// diode junction voltage
disp("Junction voltage ,VAK1="+string(VAK1)+" volt")
    //initialization
gf=(IR/(2*VT))*exp(VAK1/(2*VT))  //Formulae
disp("Forward conductance , gf="+string(gf)+" mho")
rf=1/gf  //Formulae
disp("Forward resistance , rf = "+string(rf)+ " ohm")
VAK2=(-0.7)
gr=(IR/(2*VT))*exp(VAK2/(2*VT))  //Formulae
disp("Reverse conductance , gr="+string(gr)+" mho")
rr=1/gr  //Formulae
disp(" Reverse resistance , rr = "+string(rr)+ " ohm")
```

Scilab code Exa 2.4 various parameters for two diode rectifier

```
//Ex2_4
clc
Vi=10
disp("input voltage ,Vi = "+string(Vi)+" volts")  // initialization
Rs=0.2
disp("resistance ,Rs = "+string(Rs)+ "ohm")  // initialization
RL=10
disp("resistance ,RL = "+string(RL)+ "ohm")  // initialization
VD=0.7
disp("input voltage ,VD = "+string(VD)+" volts")  // initialization
Vim=Vi*sqrt(2)  //Formulae
Iim=(Vim-VD)/(RL+Rs)  //Formulae
disp(" Peak load current ,Iim =(Vim-VD)/(RL+Rs) ="+string(Iim)+" ampere") // calculation
Ildc=(2*Iim/(%pi))  //Formulae
disp(" D.C load current ,Ildc =(2*Iim/(%pi)) ="+string(Ildc)+" ampere") // calculation
Iadc=(Ildc/2)  //Formulae
disp(" diode d.c current ,Iadc =(Ildc/2)= "+string(Iadc)+" ampere") // calculation
PIV=2*Vim  //Formulae
disp("peak inverse voltage ,PIV = 2*Vim="+string(PIV)+" volts") // calculation
Vldc=Ildc*RL  //Formulae
disp("D.C output voltage ,Vldc=Ildc*RL= "+string(Vldc)+" volts") // calculation
```

Scilab code Exa 2.5 calculation of resistance

```
//Ex2_5
clc
Idc=1*10^(-3)
disp(" D.C load current ,Idc = "+string(Idc)+"
   ampere") //initialization
Vi=2.5
disp("input voltage,Vi = "+string(Vi)+" volts")//
   initialization
Vim=Vi*sqrt(2)
VD=0.7
disp("voltage drop,VD = "+string(VD)+ " volts") //
   initialization
Rm=50
disp("resistance,Rm = "+string(Rm)+ " ohm") //
   initialization
R=[(2/%pi)*((Vim-2*VD)/Idc)-Rm] //Formulae
disp("resistance,R =[(2/%pi)*((Vim-2*VD)/Idc)-Rm]= "
   +string(R)+ " ohm")

// NOTE: VALUE OF R=1310 ohm as given in book but
   here calculated ans is 1309.5231ohm
```

Scilab code Exa 2.6 full wave rectifier parameters

```
//Ex2_6
clc
Vi=10
disp("input voltage,Vi = "+string(Vi)+" volts") //
   initialization
Vim=Vi*sqrt(2)
```

```
f1=50
disp("frequency ,f1="+string(f1)+" hertz")   //
    initialization
RL=1100
disp("resistance ,RL = "+string(RL)+ " ohm")   //
    initialization
C=50*10^(-6)
r=1/[(4*sqrt(3))*f1*RL*C] // Formulae
disp("Ripple factor ,r = "+string(r)+"")
x=1/(4*f1*RL*C) // Formulae
VLDC=Vim/(1+x) // Formulae
disp("output voltage ,VLDC = VLDC=Vim/(1+x)="+string(
    VLDC)+" volts") //calculation
VR=(Vim-VLDC)/(VLDC) // Formulae
disp(" voltage Regulation ,VR =(Vim-VLDC)/(VLDC)= "+
    string(VR)+" volts")  //calculation
Vr=VLDC*r // Formulae
disp("Ripple output voltage ,Vr = Vr=VLDC*r="+string(
    Vr)+" volts")//calculation
```

Scilab code Exa 2.7 Zener diode parameters

```
//Ex2_7
clc
VI=10
disp("input voltage ,VI = "+string(VI)+" volts") //
    initialization
Vz=5
disp("diode voltage ,Vz = "+string(Vz)+" volts") //
    initialization
Rz=100
disp("resistance ,Rz = "+string(Rz)+ " ohm")   //
    initialization
RD=500
disp("resistance ,RD = "+string(RD)+ " ohm")   //
```

```
initialization
DVI=25
disp("percentage change in VI,DVI= "+string(DVI)+" volts") //initialization
DVL=(DVI)*(Rz/(RD+Rz))  //Formulae
disp("percentage change in VL,DVL=(DVI)*(Rz/(RD+Rz)) = "+string(DVL)+" %")
R0=(RD*Rz)/(RD+Rz) //Formulae
disp("Output resistance,R0 =(RD*Rz)/(RD+Rz)= "+string(R0)+ " ohm")
VImax=12.5
Izmax=(VImax-Vz)/(RD+Rz) //Formulae
disp("resistance,RD = "+string(RD)+ " ohm")
PZmax=(Izmax*Vz)  //Formulae
disp("Power dissipated,PZmax =PZmax=(Izmax*Vz)= "+string(PZmax)+ " watt")
Prd=(Izmax*Izmax*RD)  //Formulae
disp("Power dissipated,Prd=Prd=(Izmax*Izmax*RD)= "+string(Prd)+ " watt")
PD=(PZmax+Prd)  //Formulae
disp("Power dissipated,PD = "+string(PD)+ " watt")
RL=0.5*(10^3)
disp("resistance,RL = "+string(RL)+ " ohm") //initialization
%VR=(R0*100)/RL  //Formulae
disp(" voltage Regulation Percentage,%VR =(R0/RL)*(100)= "+string(%VR)+"% ")
```

Scilab code Exa 2.8 reference voltage and temperature coefficient

```
//Ex2_8
clc
Vz=10 //initialization
disp("diode voltage,Vz = "+string(Vz)+" volts")
TC1=(10*0.02)/(100) //calculation
```

```
disp(" Zener diode TC1 = "+string(TC1)+" V/degree celsius")
VD=0.7
disp(" voltage drop,VD = "+string(VD)+" volts")
TC2=(-2.5*10^(-3))  //calculation
disp("Si diode TC = "+string(TC2)+" V/degree celsius")
Vref1=VD+Vz
disp("Combined voltage ,Vref=VD+Vz= "+string(Vref1)+" volts")
TC3=(TC1+TC2) //calculation
disp(" Combined TC = "+string(TC3)+" V/degree celsius")
TC=(TC1+TC2)*100/(Vref1) //calculation
disp("New Combined TC = (TC1+TC2)*100/(Vref1)="+string(TC)+" percent/degree celsius")
T1=25//temperature
T2=50// new temperature
Vref=Vref1-((-TC3)*(T2-T1))//calculation
disp("New Combined reference voltage ,Vref= Vref1-((-TC3)*(T2-T1))="+string(Vref)+" volts")
```

Scilab code Exa 2.9 load voltage and current

```
//Ex2_9
clc
Vi1=0.2
disp("input voltage,Vi1 = "+string(Vi1)+" volts") //initialization
VD=0.7
disp(" voltage drop,VD = "+string(VD)+" volts") //initialization
RL=5*(10^3)
disp("resistance,RL = "+string(RL)+ " ohm") //initialization
```

```
Vcc=5
disp("Supply voltage ,Vcc = "+string(Vcc)+" volts")
V01=VD+Vi1 //Formulae
disp("output voltage ,V01 ==VD+Vi1 = "+string(V01)+" volts")
IL1=(Vcc-V01)/RL //Formulae
disp(" output current ,IL1=IL1=(Vcc-V01)/RL = "+string(IL1)+" ampere") // calculation
Vi2=5
disp("input voltage ,Vi2 = "+string(Vi2)+" volts") // initialization
V02=3*VD //Formulae
disp("output voltage ,V02 =3*VD= "+string(V02)+" volts")
IL2=(Vcc-V02)/RL //Formulae
disp(" output current ,IL2= IL2=(Vcc-V02)/RL = "+string(IL2)+" ampere") // calculation
VAK=V02-Vi2 //Formulae
disp(" Diode voltage ,VAK = V02-Vi2 = "+string(VAK)+" volts")

//NOTE:correct value of IL2=0.58 mA but in book given as 0.592mA
```

Chapter 3

Bipolar Junction Transistor

Scilab code Exa 3.1 BJT current and voltages

```
//Ex3_1
clc
Vcc=15
disp("Vcc = "+string(Vcc)+" volts")    //initialization
VBB=1
disp("VBB = "+string(VBB)+" volts")   //initialization
VBE=0.7
disp("VBE = "+string(VBE)+" volts")   //initialization
RB=5*(10^3)
disp("resistance ,RB = "+string(RB)+ " ohm")   //initialization
RL=650
disp("resistance ,RL = "+string(RL)+ " ohm")   //initialization
Bf=200
disp("Gain , Bf = "+string(Bf)+ " ")   //initialization
IB=(VBB-VBE)/RB //Formulae
disp("IB =(VBB-VBE)/RB = "+string(IB)+" ampere") //
```

```
calculation
IC=IB*Bf   //Formulae
disp("IC =IB*Bf= "+string(IC)+" ampere")//calculation
IE=IB+IC   //Formulae
disp("IE = IB+IC="+string(IE)+" ampere")//calculation
VCE=Vcc-IC*RL //Formulae
disp("VCE =Vcc-IC*RL= "+string(VCE)+" volts") //calculation
VCB=VCE-VBE  //Formulae
disp("VCB = VCE-VBE="+string(VCB)+" volts")//calculation
RB=(Vcc-VBE)/IB  //Formulae
disp("resistance,RB =(Vcc-VBE)/IB= "+string(RB)+ " ohm") //calculation
```

Scilab code Exa 3.2 find current gain and voltage gain

```
//refer BJT characteristics in fig3.2(a)and fig3.4(a)
//Ex3_2
clc
Vbe1=0.025
disp("Vbe1 = "+string(Vbe1)+" volts") // value of base-emitter voltage1
Vbe2=(-0.025)
disp("Vbe2 = "+string(Vbe2)+" volts") // value of base-emitter voltage2
ib1=20*10^(-6)
disp("ib1 = "+string(ib1)+" ampere") // value of base current1
ib2=(-20*10^(-6))
disp("ib2 = "+string(ib2)+" ampere")// value of base current2
```

```
IBQ=60*10^(-6)
disp("IBQ = "+string(IBQ)+" ampere") // operating
    point
ICP=15.5*10^(-3)
disp("ICP = "+string(ICP)+" ampere") //
    initialization
ICR=8.5*10^(-3)
disp("ICR = "+string(ICR)+" ampere") //
    initialization
VCEP=5
disp("VCEP = "+string(VCEP)+" volts") // value of
    collector-emitter voltage1
VCER=9
disp("VCER = "+string(VCER)+" volts") // value of
    collector-emitter voltage2
change_IC=ICP-ICR  //change in collector current
disp("change_IC = "+string(change_IC)+" ampere")
change_VCE=VCER-VCEP //change in collector voltage
disp("change_VCE = "+string(change_VCE)+" volts")
change_VBE=Vbe1-Vbe2
change_IB=ib1-ib2
AV=(change_VCE/change_VBE) //formulae voltage gain
disp("AV = "+string(AV)+" ")//voltage gain
AI=change_IC/change_IB //formulae current gain
disp("AI = "+string(AI)+" ")
```

Scilab code Exa 3.3 Find transconductance gm and hybrid parameter hie and unity gain frequency

```
//Ex3_3
clc
ICQ=12*10^(-3)
disp("ICQ = "+string(ICQ)+" ampere") // collector
    current
B=200
```

```
disp("B = "+string(B)+" ") //BJT gain
Cbe=100*10^(-12)// capacitance
disp("capacitance,Cbe = "+string(Cbe)+" F ")
VT=26*10^(-3)
disp("VT = "+string(VT)+" volts") // thermal voltage
gm=(ICQ/VT) //transconductance
disp("gm =(ICQ/VT)= "+string(gm)+" A/V")
hie=B/gm //forward resistance hybrid parameter
disp("hie =(B/gm)= "+string(hie)+" ohm")
fT=((1/2)*(gm/Cbe)*(1/%pi)) //unity gain frequency formulae
disp("fT =((1/2)*(gm/Cbe)*(1/%pi))= "+string(fT)+" hertz")
```

Scilab code Exa 3.4 operating point of BJT

```
//Ex3_4
//refer fig3.2(a)
clc
VCC=20
disp("VCC = "+string(VCC)+" volts") // collector supply voltage
VBB=VCC
RL=5*(10^3)
disp("RL= "+string(RL)+ " ohm")  //resistance
RB=965*(10^3)
disp("RB = "+string(RB)+ " ohm")  //initialization base resistance
VBE=(0.7)
disp("VBE = "+string(VBE)+" volts") // value of base-emitter voltage
BF=50
disp("BF = "+string(BF)+" ") //BJT gain
ICO=10*10^(-9)
disp("ICO = "+string(ICO)+" ampere") // collector
```

```
reverse bias current
Vi=0
disp("Vi = "+string(Vi)+" volts") // value of input
IBQ=(VCC-VBE)/RB //base current as operating point
disp("IBQ = "+string(IBQ)+" ampere")
ICQ=BF*IBQ //operating point (collector current)
disp("ICQ =BF*IBQ= "+string(ICQ)+" ampere") // calculation
VCEQ=VCC-ICQ*RL //  collector-emitter voltage as operating point
disp("VCEQ =VCC-ICQ*RL = "+string(VCEQ)+" volts") // calculation
```

Scilab code Exa 3.5 Find operating point of BJT

```
//Ex3_5
clc
BF1=100
disp("BF1 = "+string(BF1)+" ") //BJT gain
VCC=20
disp("VCC = "+string(VCC)+" volts") // collector supply voltage
VBB=VCC
RL=5*(10^3)
disp("resistance ,RL= "+string(RL)+ " ohm")  // initialization
RB=965*(10^3)
disp("resistance ,RB = "+string(RB)+ " ohm")  // initialization
VBE=(0.7)
disp("VBE = "+string(VBE)+" volts") // value of base -emitter voltage
ICO=10*10^(-9)
disp("ICO = "+string(ICO)+" ampere") // collector reverse bias current
```

```
Vi=0
disp("Vi = "+string(Vi)+" volts") // value of input
IBQ=(VCC-VBE)/RB //base current as operating point
disp("IBQ = "+string(IBQ)+" ampere")
ICQ1=BF1*IBQ //operating point (collector current)
disp("ICQ1 =BF1*IBQ= "+string(ICQ1)+" ampere") // calculation
VCEQ1=VCC-ICQ1*RL //  collector-emitter voltage as operating point
disp("VCEQ1 =VCC-ICQ1*RL = "+string(VCEQ1)+" volts") //calculation
```

Scilab code Exa 3.6 Find operating point of BJT

```
//Ex3_6
clc
VBE2=(0.5)
disp("VBE2 = "+string(VBE2)+" volts") // value of base-emitter voltage
VCC=20
disp("VCC = "+string(VCC)+" volts") // collector supply voltage
BF2=150
disp("BF2 = "+string(BF2)+" ") //BJT gain
ICO2=500*10^(-9)
disp("ICO2 = "+string(ICO2)+" ampere") // collector reverse bias current
RB=965*(10^3)
disp("RB = "+string(RB)+ " ohm")  //initialization base resistance
RL=5*(10^3)
disp("RL= "+string(RL)+ " ohm")  // load resistance
IBQ2=(VCC-VBE2)/RB   //base current as operating point
disp("IBQ2 = (VCC-VBE2)/RB="+string(IBQ2)+" ampere")
```

```
ICQ2=(BF2*IBQ2)    //operating point (collector
    current)
disp("ICQ2 =BF2*IBQ2= "+string(ICQ2)+" ampere") //
dICQ2=BF2*ICO2  // increase in reverse bias current
disp("dICQ2 =BF2*ICO2= "+string(dICQ2)+" ampere") //
ICQ3=ICQ2+dICQ2
disp("ICQ3 =ICQ2+dICQ2= "+string(ICQ3)+" ampere") //
    calculation
VCEQ3=VCC-ICQ3*RL //  collector-emitter voltage as
    operating point
disp("VCEQ3 =VCC-ICQ3*RL = "+string(VCEQ3)+" volts")
    //calculation
//NOTE: Calculated ans for VCEQ3=4.4695596 volts but
    in book it is given as 4.625 volts(due to
    approximations done in)
```

Scilab code Exa 3.7 Calculate extreme variation in operating collector current ICQ

```
//Ex3_7
clc
VCC=20
disp("VCC = "+string(VCC)+" volts") // collector
    supply voltage
RL=5*(10^3)
disp("resistance ,RL= "+string(RL)+ " ohm")  //
    initialization
R1=90*(10^3)
disp("resistance ,R1 = "+string(R1)+ " ohm")  //
    initialization
R2=10*(10^3)
disp("resistance ,R2 = "+string(R2)+ " ohm")  //
    initialization
Rc=1*(10^3)
disp("resistance ,Rc = "+string(Rc)+ " ohm")  //
```

```
resistance at collector
VBEmax=(0.7)
disp("VBEmax = "+string(VBEmax)+" volts") // maximum base-emitter voltage
VBEmin=(0.5)
disp("VBEmin = "+string(VBEmin)+" volts") // minimum base-emitter voltage
BFmax=150
disp("BFmax = "+string(BFmax)+" ") //BJT gain maximum
BFmin=50
disp("BFmin = "+string(BFmin)+" ") //BJT gain minimum
ICOmax=500*10^(-9)
disp("ICOmax = "+string(ICOmax)+" ampere") // maximum collector reverse bias current
ICOmin=10*10^(-9)
disp("ICOmin = "+string(ICOmin)+" ampere") // minimum collector reverse bias current
VBB=(VCC*R2)/(R1+R2)
disp("VBB = "+string(VBB)+" volts") // Base supply voltage
RB=(R1*R2)/(R1+R2)
disp("RB = (R1*R2)/(R1+R2)="+string(RB)+ " ohm") // eqivalent base resistance
ICmin=[(BFmin*(VBB-VBEmax)+(RB+Rc)*(1+BFmin)*ICOmin)/(RB+Rc*(1+BFmin))] // minimum collector  current
disp("ICmin = "+string(ICmin)+" ampere")
VCEQmax=VCC-ICmin*RL //  maximum collector-emitter voltage (d.c value)
disp("VCEQmax =VCC-ICmin*RL = "+string(VCEQmax)+" volts") //calculation
ICmax=[(BFmax*(VBB-VBEmin)+(RB+Rc)*(1+BFmax)*ICOmax)/(RB+Rc*(1+BFmax))] // maximum collector  current
disp("ICmax = "+string(ICmax)+" ampere")
VCEQmin=VCC-ICmax*RL // minimum collector-emitter voltage (d.c value)
disp("VCEQmin =VCC-ICmax*RL = "+string(VCEQmin)+"
```

```
   volts") //calculation
change_IC=ICmax-ICmin
disp("change_IC= "+string(change_IC)+" ampere") //
   extreme variation in collector current
// ERROR - NOTE: Extreme variation in collector
   current given in book is 0.397 mA but calculated
   correct ans is 0.3276 mA
```

Scilab code Exa 3.8 find value of collector current Ic and check is BJT in active region

```
//Ex3_8
clc
VCC=20
disp("VCC = "+string(VCC)+" volts") // collector
   supply voltage
RL=2*(10^3)
disp("RL= "+string(RL)+ " ohm")  //resistance
R1=100*(10^3)
R2=R1
disp("R1 =R2= "+string(R1)+ " ohm")  //resistance
VBE=(0.7)
disp("VBE = "+string(VBE)+" volts") //  base-emitter
    voltage
BF=100
disp("BF = "+string(BF)+" ") //BJT gain
ICO=0
VBB=(VCC*R2)/(R1+R2)
disp("VBB = "+string(VBB)+" volts") // Base supply
   voltage
RB=(R1*R2)/(R1+R2)
disp("RB = (R1*R2)/(R1+R2)="+string(RB)+ " ohm") //
   eqivalent base resistance
IC=[(BF*(VBB-VBE))/(RB+RL*(1+BF))] // collector
   current
```

```
disp("IC = "+string(IC)+" ampere")
VE=IC*RL
disp("VE = "+string(VE)+" volts") //  emitter
    voltage
VB=VBE+VE
disp("VB = "+string(VB)+" volts") //  base voltage
VCB=VCC-VB
disp("VCB = "+string(VCB)+" volts") //  collector-
    base voltage
//hence BJT in active region.
```

Scilab code Exa 3.9 find forced vaue of current gain for BJT in saturation region

```
//Ex3_9
clc
VCC=5
disp("VCC = "+string(VCC)+" volts") // collector
    supply voltage
RL=250
disp("RL= "+string(RL)+ " ohm")  //initialization
RB=25*10^(3)
disp("RB ="+string(RB)+ " ohm") //  base resistance
VCS=(0.2)
disp("VCS = "+string(VCS)+" volts") //   voltage
BF=200
disp("BF = "+string(BF)+" ") //BJT gain
VBS=(0.8)
disp("VBS = "+string(VBS)+" volts") //  base-emitter
     voltage for BJT switch
VI=5
disp("VI = "+string(VI)+" volts")// input voltage
VCON=0.3
disp("VCON = "+string(VCON)+" volts")
ICON=(VCC-VCON)/RL
```

```
disp("ICON = (VCC-VCON)/RL="+string(ICON)+" ampere") //collector current for saturated BJT
IBON=(ICON)/BF
disp("IBON = (ICON)/BF="+string(IBON)+" ampere")// Base current for saturated BJT
IBS=(VI-VBS)/RB
disp("IBS = (VI-VBS)/RB="+string(IBS)+" ampere")// Base-emitter current for saturated BJT
ICS=(VCC-VCS)/RL
disp("ICS = (VCC-VCS)/RL="+string(ICS)+" ampere")// Collector-emitter current for saturated BJT
Bforced=ICS/IBS
disp("Bforced = ICS/IBS="+string(Bforced)+" ") //BJT forced gain
//IBS>>IBON hence BJT in saturation
```

Scilab code Exa 3.10 Find maximum allowed power dissipation at 25 and 75 degree celsius

```
//Ex3_10
clc
TJmax=175
disp("TJmax= "+string(TJmax)+ "degree celsius") // maximum allowed junction temperature
theta=0.5
disp("theta= "+string(theta)+ "degree celsius/mW ") //thermal resistances b/w junction to ambient
change_T=TJmax-25//temperature difference
PDmax=change_T/theta
disp("at 25 degree celsius,PDmax=(TJmax-25 )/theta = "+string(PDmax)+ " mW ") //maximum allowed power dissipation at TA=25 degree celsius
change_T=TJmax-75
PDmax2=change_T/theta
disp("at 75 degree celsius,PDmax2= (TJmax-75)/theta
```

```
    = "+string(PDmax2)+ " mW ")   //maximum allowed
    power dissipation at TA=75 degree celsius
```

Scilab code Exa 3.11 Find maximum allowed power dissipation at 25 and 75 degree celsius

```
//Ex3_11
clc
TJmax=175
disp("TJmax= "+string(TJmax)+ " degree celsius")  //
   maximum allowed junction temperature
theta=0.1
disp("theta= "+string(theta)+ " degree celsius/mW ")
     //thermal resistances b/w junction to ambient
change_T=TJmax-25 //temperature difference
PDmax=change_T/theta
disp("at 25 degree celsius,PDmax=(TJmax-25 )/theta =
     "+string(PDmax)+ " mW ")  //maximum allowed
    power dissipation at TA=25 degree celsius
change_T=TJmax-75 //temperature difference
PDmax=change_T/theta
disp("at 75 degree celsius,PDmax= (TJmax-75)/theta =
     "+string(PDmax)+ " mW ")  //maximum allowed
    power dissipation at TA=75 degree celsius
```

Scilab code Exa 3.12 Calculate resistance R for BJT current mirror

```
//Ex3_12
clc
VBE=(0.7)
disp("VBE = "+string(VBE)+" volts") // value of base
   -emitter voltage
VCC=10
```

```
disp("VCC = "+string(VCC)+" volts") // collector
    supply voltage
IREF=5*10^(-3)
disp("IREF =IQ= "+string(IREF)+" ampere") //
    current mirror source current
R=(VCC-VBE)/(IREF)// formulae
disp("R=(VCC-VBE)/(IREF)= "+string(R)+ " ohm")  //
    resistance
```

Chapter 4

Field Effect Transistors

Scilab code Exa 4.1 Find drain to source voltage VDS and drain current ID for JFET

```
//Ex4_1
clc
IDSS=10*10^(-3)
disp("IDSS = "+string(IDSS)+" ampere") //  maximum drain current
VP=(-4)
disp("VP= "+string(VP)+" volts") // pinch off voltage
VGS=(-2)
disp("VGS = "+string(VGS)+" volts") // gate to source voltage
VDSmin=VGS-VP
disp("VDSmin  =VGS-VP="+string(VDSmin)+" volts") // Drain to source voltage
ID=IDSS*(1-VGS/VP)^2
disp("ID =IDSS*(1-VGS/VP)^2= "+string(ID)+" ampere") //   drain current
```

Scilab code Exa 4.2 Find drain to source resistance RDS for JFET

```
//Ex4_2
clc
IDSS=10*10^(-3)
disp("IDSS = "+string(IDSS)+" ampere") //  maximum
    drain current
VP=(-4)
disp("VP= "+string(VP)+" volts") // pinch off
    voltage
VGS=(0)
disp("VGS = "+string(VGS)+" volts") // gate to
    source voltage1
RDS=1/[(2*(IDSS/(-VP)))*(1-VGS/VP)]//formula for
    JFET
disp("RDS = 1/[(2*(IDSS/(-VP)))*(1-VGS/VP)]="+string
    (RDS)+" ohm") //   drain to source resistance for
     VGS=0V
VGS=(-2)
disp("VGS = "+string(VGS)+" volts") // gate to
    source voltage2
RDS=1/[(2*(IDSS/(-VP)))*(1-VGS/VP)]
disp("RDS = 1/[(2*(IDSS/(-VP)))*(1-VGS/VP)]="+string
    (RDS)+" ohm") //   drain to source resistance for
     VGS=(-2)
```

Scilab code Exa 4.3 Find load resistance RL for EMOSFET

```
//Ex4_3
clc
ID=10*10^(-3)
disp("ID = "+string(ID)+" ampere") //  given drain
    current
VDD=(24)
disp("VDD= "+string(VDD)+" volts") // Drain voltage
```

```
VT=(5)
disp("VT= "+string(VT)+" volts") // Threshold voltage
VGS=(10)
disp("VGS= "+string(VGS)+" volts") // gate to source voltage1
KF=ID/(VGS-VT)^2
disp("KF = ID/(VGS-VT)^2 = "+string(KF)+" A/V^2") // To calculate Scale factor for finding ID2
VDS=(7)
disp("VDS =VGS= "+string(VDS)+" volts") // drain to source voltage
VGS=(VDS)
ID=KF*(VGS-VT)^2
disp("ID =KF*(VGS-VT)^2= "+string(ID)+" ampere") // New drain current for VDS=24V
RL=(VDD-VDS)/ID
disp("RL=(VDD-VDS)/ID= "+string(RL)+ " ohm")  // calculation for load resistance at VDS=24V
```

Scilab code Exa 4.4 calculate transconductance gm for FETs at various device parameters

```
//Ex4_4
clc
disp("part(i) ")// part(i) of this question
IDSS=5*10^(-3)
disp("IDSS = "+string(IDSS)+" ampere") //  maximum drain current JFET 1
VP=(2)
disp("VP= "+string(VP)+" volts") // pinch off voltage for JFET 1
IDQ=4.42*10^(-3)
disp("IDQ = "+string(IDQ)+" ampere") //   drain current for JFET 1
```

```
gm=[(2)*sqrt(IDQ*IDSS)]/VP
disp("gm =[(2)*sqrt(IDQ*IDSS)]/VP= "+string(gm)+" A/
   V")// calculating transconductance for JFET with
   IDQ = 4.42 mA

disp("part(ii) ")// part(ii) of this question
IDQ=6.04*10^(-3)
disp("IDQ = "+string(IDQ)+" ampere") //   drain
   current for JFET 1
IDSS=15*10^(-3)
disp("IDSS = "+string(IDSS)+" ampere") //  maximum
   drain current JFET2
VP=(6)
disp("VP= "+string(VP)+" volts") // pinch off
   voltage for JFET2
gm=[(2)*sqrt(IDQ*IDSS)]/VP
disp("gm =[(2)*sqrt(IDQ*IDSS)]/VP= "+string(gm)+" A/
   V")// calculating transconductance for JFET with
   IDQ = 6.04 mA

disp("part(iii) ")// part(iii) of this question
IDQ=1*10^(-3)
disp("IDQ = "+string(IDQ)+" ampere") //   drain
   current for EMOSFET 1
KF=0.25*10^(-3)
disp("KF = "+string(KF)+" A/V^2") //  Scale factor
   for finding EMOSFET1
gm=sqrt(4*IDQ*KF)
disp("gm =sqrt(4*IDQ*KF)= "+string(gm)+" A/V")//
   calculating transconductance for EMOSFET1 with
   IDQ = 1 mA

disp("part(iv) ")// part(iv) of this question
IDQ=0.91*10^(-3)
disp("IDQ = "+string(IDQ)+" ampere") //   drain
   current for EMOSFET 2
KF=0.375*10^(-3)
disp("KF = "+string(KF)+" A/V^2") //  Scale factor
```

```
for finding EMOSFET2
gm=sqrt(4*IDQ*KF)
disp("gm =sqrt(4*IDQ*KF)= "+string(gm)+" A/V")//
    calculating transconductance for EMOSFET2 with
    IDQ = 0.91 mA
```

Scilab code Exa 4.5 Design potential divider biasing for JFET

```
//Ex4_5
//Refer fig.4.8
clc
IDQmax=5*10^(-3)
disp("IDQmax = "+string(IDQmax)+" ampere") // drain
    current for JFET for maximum transfer
    characteristics
IDmax=IDQmax// maximum drain current will be given
    by IDQmax
IDQmin=3*10^(-3)
disp("IDQmin = "+string(IDQmin)+" ampere") // drain
    current for JFET for minimum transfer
    characteristics
VDD=(20)
disp("VDD= "+string(VDD)+" volts") // Drain voltage
    supply
VDSmin=(6)
disp("VDSmin= "+string(VDSmin)+" volts") // minimum
    Drain to source voltage supply
ID=2.4*10^(-3)
disp("ID = "+string(ID)+" ampere") // drain current
    chosen for operation within max and min limits
VGG=3
disp("VGG= "+string(VGG)+" volts") // Gate voltage
    from fig.
Ri=100*10^(3)
disp("Ri= "+string(Ri)+ " ohm") //eqivalent input
```

```
resistance
RF=(VGG-0)/(ID-0)
disp("RF= (VGG-0)/(ID-0)= "+string(RF)+ " ohm")   //calculation for feedback resistance
R1=VDD*Ri/VGG //using formulae VGG=VDD*Ri/R1
disp("R1= VDD*Ri/VGG= "+string(R1)+ " ohm")   //calculation for first resistance R1 at input side
R2=R1*VGG/(VDD-VGG)
disp("R2= R1*VGG/(VDD-VGG)= "+string(R2)+ " ohm")   //calculation for second resistance R2 at input side
RL=[((VDD-VDSmin)/IDmax)-RF] // using formulae VDD=IDmax(RL+RF)+VDSmin
disp("RL=[((VDD-VDSmin)/IDmax)-RF]="+string(RL)+ " ohm")   //Load resistance calculation
```

Scilab code Exa 4.6 calculate feedback resistance RF for n channel JFET

```
//Ex4_6
clc
IDSS=50*10^(-3)
disp("IDSS = "+string(IDSS)+" ampere") //  maximum drain current JFET
VP=(-10)
disp("VP= "+string(VP)+" volts") // pinch off voltage for JFET
VGSQ=(-5)
disp("VGSQ= "+string(VGSQ)+" volts") // Gate operating point voltage
ID=IDSS*(1-VGSQ/VP)^2
disp("ID =IDSS*(1-VGS/VP)^2= "+string(ID)+" ampere")  //   drain current JFET
RF=abs(VGSQ/ID)
disp("RF= (VGSQ)/(ID)= "+string(RF)+ " ohm")   //
```

```
calculation for feedback resistance
```

Scilab code Exa 4.7 Find drain current ID and verify operating region is pinch off

```
//Ex4_7
clc
IDSS=5*10^(-3)
disp("IDSS = "+string(IDSS)+" ampere") // maximum drain current JFET
RL=910
disp("RL= "+string(RL)+ " ohm") //Load resistance
RF=2.29*10^(3)
disp("RF= "+string(RF)+ " ohm") // feedback resistance
R1=12*10^(6)
disp("R1= "+string(R1)+ " ohm") // first resistance R1 at input side
R2=8.57*10^(6)
disp("R2= "+string(R2)+ " ohm") // second resistance R2 at input side
VDD=(24)
disp("VDD= "+string(VDD)+" volts") // Drain voltage supply
VP=(-2)
disp("VP= "+string(VP)+" volts") // pinch off voltage for JFET
VGG=(VDD*R2)/(R1+R2)
disp("VGG= VDD*R2/(R1+R2)="+string(VGG)+" volts") // Gate voltage for JFET
disp("Quadratic equation =5.244*ID^(2)-55.76*ID+144=0")// Forming Quadratic equation using VGS = VGG-ID*RF and ID = IDSS(1-VGS/VP)^2 where ID in mA
p = [5.244 -55.76 144]
```

```
ID=roots(p)*10^(-3)// values of ID converted into Ampere by multiplying by 10^(-3)
disp("ID = "+string(ID)+" ampere") //  drain current JFET
disp("Since ID <=IDSS, hence ID=6.214 mA cannot be chosen, so we chose ID=4.42 mA")
IDQ=4.42*10^(-3)
disp("IDQ ="+string(IDQ)+"  A")//Since ID <=IDSS, hence ID=6.214 mA cannot be chosen, so we chose ID=4.42 mA
VGSQ=VGG-IDQ*RF
disp("VGSQ = VGG-IDQ*RF = "+string(VGSQ)+" volts") //  Gate  operating point voltage
VDSQ=VDD-IDQ*(RL+RF)
disp("VDSQ= VDD-IDQ*(RL+RF)= "+string(VDSQ)+" volts") // Drain voltage for JFET
VDGQ=VDSQ-VGSQ
disp("VDGQ = VDSQ-VGSQ ="+string(VDGQ)+" volts") // Drain-Gate voltage for JFET
disp("VDGQ >magnitude_VP ,Hence FET is in pinch off region")
```

Scilab code Exa 4.8 Find operating values of drain current and drain to source voltage and gate to source voltage

```
//Ex4_8// NOTE : all values of book ans is wrong so give note-INCOMPLETE QUESTION
clc
IDSS=15*10^(-3)
disp("IDSS = "+string(IDSS)+" ampere") //  maximum drain current of JFET
RL=910
disp("RL= "+string(RL)+ " ohm")  //Load resistance
RF=2.29*10^(3)
disp("RF= "+string(RF)+ " ohm")  //  feedback
```

```
resistance
R1=12*10^(6)
disp("R1= "+string(R1)+ " ohm")  // first resistance R1 at input side
R2=8.57*10^(6)
disp("R2= "+string(R2)+ " ohm")  // second resistance R2 at input side
VDD=(24)
disp("VDD= "+string(VDD)+" volts") // Drain voltage supply
VP=(-6)
disp("VP= "+string(VP)+" volts") // pinch off voltage for JFET
VGG=(VDD*R2)/(R1+R2)
disp("VGG= VDD*R2/(R1+R2)="+string(VGG)+" volts") // Gate voltage for JFET
disp("Quadratic equation =5.244*ID^(2)-75.68*ID+256=0")// where ID in mA
p = [5.244  -75.68  256]
ID=roots(p)*10^(-3)//values of ID converted into Ampere by multiplying by 10^(-3)
disp("ID = "+string(ID)+" ampere") //   drain current JFET
VDG=VDD-(ID*RL)-VGG
disp("VDG= "+string(VDG)+" volts") // Drain-gate voltage for JFET
disp("since VDG < magnitude_VP for ID=9.0189 mA which is inappropriate for JFET pinch off region ,hence ID=5.4128 mA is choosen  !")
IDQ=5.41*10^(-3) // since VDG < magnitude_VP for ID=9.0189 mA which is inappropriate for JFET pinch off region ,hence ID=5.4128 mA is choosen  !
disp("IDQ ="+string(IDQ)+"  ampere")
VGSQ=VGG-IDQ*RF
disp("VGSQ = VGG-IDQ*RF = "+string(VGSQ)+" volts") //  Gate  operating point voltage
VDSQ=VDD-IDQ*(RL+RF)
disp("VDSQ= VDD-IDQ*(RL+RF)= "+string(VDSQ)+" volts"
```

```
) // Drain voltage for JFET
VDGQ=VDSQ-VGSQ
disp("VDGQ = VDSQ-VGSQ ="+string(VDGQ)+" volts") // Drain-Gate voltage for JFET
disp("VDGQ > VP, Hence FET is in pinch off region")


//Roots for drain current quadratic equation are wrong in the book thus value for VGSQ,VDSQ and VDGQ are also wrong
```

Scilab code Exa 4.9 Find operating values of drain current ID and drain to source voltage VDS and gate to source voltage VGS for FET

```
//Ex4_9
clc
RL=12*10^(3)
disp("RL= "+string(RL)+ " ohm")  //Load resistance
RF=6*10^(3)
disp("RF= "+string(RF)+ " ohm")  //  feedback resistance
R1=12*10^(6)
disp("R1= "+string(R1)+ " ohm")  // first resistance R1 at input side
R2=8.57*10^(6)
disp("R2= "+string(R2)+ " ohm")  // second resistance R2 at input side
VDD=(24)
disp("VDD= "+string(VDD)+" volts") // Drain voltage supply
VT=(3)
disp("VT= "+string(VT)+" volts") // Threshold voltage for n-channel EMOSFET
KF=0.25*10^(-3)
disp("KF= "+string(KF)+" A/V^2") // Constant for n-
```

```
channel EMOSFET
VGG=(VDD*R2)/(R1+R2)
disp("VGG= VDD*R2/(R1+R2)="+string(VGG)+" volts") // Gate voltage for n-channel EMOSFET
disp("Quadratic equation =9*ID^(2)-25*ID+16=0")// IDS=KF*(VGS-VT)^2 and VGS=VGG-ID*RD ,so Quadratic equation formed is :IDS=KF*(VGG-ID*RD-VT)^2 where ID in mA
p = [9  -25  16]
ID=roots(p)*10^(-3)//values of ID converted into Ampere by multiplying by 10^(-3)
disp("ID = "+string(ID)+" A") //  drain current n-channel EMOSFET in Ampere
VGS=VGG-ID*RF// For ID=1.78 mA and ID=1mA
disp("VGS = VGG-ID*RF = "+string(VGS)+" volts") // Gate  operating point voltage
disp("Since VGS < VT for ID=1.78 mA, hence ID = 1.78 mA cannot be chosen, so we chose ID= 1 mA as operating drain current IDQ")
IDQ=1*10^(-3)
disp("IDQ ="+string(IDQ)+"A")//Since VGS < VT for ID =1.78 mA, hence ID = 1.78 mA cannot be chosen, so we chose ID= 1 mA as operating drain current IDQ
VGSQ=VGG-IDQ*RF
disp("VGSQ = VGG-IDQ*RF = "+string(VGSQ)+" volts") //  Gate  operating point voltage
VDSQ=VDD-IDQ*(RL+RF)
disp("VDSQ= VDD-IDQ*(RL+RF)= "+string(VDSQ)+" volts") // Drain voltage for n-channel EMOSFET
// NOTE:Value of VGS= -0.6676390 volts for ID=1.78 mA but in book given as -0.68 V
```

Scilab code Exa 4.10 Find operating values of drain current IDQ and drain to source voltage VDSQ and gate to source voltage VGSQ for EMOSFET

```
//Ex4_10
clc
RL=12*10^(3)
disp("RL= "+string(RL)+ " ohm")  //Load resistance
RF=6*10^(3)
disp("RF= "+string(RF)+ " ohm")  //  feedback
   resistance
R1=12*10^(6)
disp("R1= "+string(R1)+ " ohm")  // first resistance
    R1 at input side
R2=8.57*10^(6)
disp("R2= "+string(R2)+ " ohm")  // second
   resistance R2 at input side
VDD=(24)
disp("VDD= "+string(VDD)+" volts") // Drain voltage
   supply
VT=(3)
disp("VT= "+string(VT)+" volts") // Threshold
   voltage for n-channel EMOSFET
KF=0.375*10^(-3)
disp("KF= "+string(KF)+" A/V^2") // Constant for  n-
   channel EMOSFET
VGG=(VDD*R2)/(R1+R2)
disp("VGG= VDD*R2/(R1+R2)="+string(VGG)+" volts") //
    Gate voltage for  n-channel EMOSFET
disp("Quadratic equation =36*ID^(2)-86.67*ID+49=0")
   // IDS=KF*(VGS-VT)^2 and VGS=VGG-ID*RD ,so
   Quadratic equation formed is :IDS=KF*(VGG-ID*RD-
   VT)^2 ,where ID in mA
p = [36  -86.67  49]
ID=roots(p)*10^(-3)//values of ID converted into
   Ampere  by multiplying by 10^(-3)
disp("ID = "+string(ID)+" A") //   drain current n-
   channel EMOSFET in Ampere
VGS=VGG-ID*RF//  Gate voltage for ID = 1.5 mA and ID
    = 0.91 mA
disp("VGS = VGG-ID*RF = "+string(VGS)+" volts") //
   Gate voltage
```

```
disp("Since VGS < VT for ID=1.5 mA, hence ID = 1.5
    mA cannot be chosen, so we chose ID= 0.91 mA as
    operating drain current IDQ")
IDQ=0.91*10^(-3)
disp("IDQ ="+string(IDQ)+" A")//Since VGS < VT for
    ID=1.5 mA, hence ID = 1.5 mA cannot be chosen, so
     we chose ID= 0.91 mA as operating drain current
    IDQ
change_IDQ=[(1-0.91)*100]/(1)//
disp("change in IDQ = "+string(change_IDQ)+" percent
    ")// Percent change in IDQ from value 1 mA from
    its actual value IDQ=0.91mA
VGSQ=VGG-IDQ*RF
disp("VGSQ = VGG-IDQ*RF = "+string(VGSQ)+" volts")
    //  Gate  operating point voltage
VDSQ=VDD-IDQ*(RL+RF)
disp("VDSQ= VDD-IDQ*(RL+RF)= "+string(VDSQ)+" volts"
    ) // Drain voltage for n-channel EMOSFET


// note: in the textbook author has given KF = .375
    but I have work with KF = .375*10^-3A/V^2
```

Scilab code Exa 4.11 Calculate load resistance RL and percentage change in drain current ID for JFET

```
//Ex4_11 Refer fig 4.7(b)// ANS is not correct check
     &correct
clc
RF=6*10^(3)
disp("RF= "+string(RF)+ " ohm")  //  feedback
    resistance
VDD=(20)
disp("VDD= "+string(VDD)+" volts") // Drain voltage
    supply
```

```
disp("part(i) ")// part(i) of this question
VT=(2)
disp("VT= "+string(VT)+" volts") // Threshold voltage for EMOSFET
KF=0.25*10^(-3)
disp("KF= "+string(KF)+" A/V^2") // Constant for EMOSFET
ID=1*10^(-3)
disp("ID = "+string(ID)+" A") //   drain current EMOSFET in Ampere
RL=[VDD-VT-sqrt(ID/KF)]/ID // Using formulae ID=KF*(VDD-ID*RL-VT)
disp("RL=[VDD-VT-sqrt(ID/KF)]/ID= "+string(RL)+ " ohm")  //Load resistance
disp("part(ii) ")// part(ii) of this question
VT=(3)
disp("VT= "+string(VT)+" volts") // Threshold voltage for EMOSFET
KF=0.375*10^(-3)
disp("KF= "+string(KF)+" A/V^2") // Constant for EMOSFET
disp("Quadratic equation =(256)*ID^(2)-(546.67)*ID+289=0")//IDS=KF*(VGS-VT)^2 =KF*(VDS-VT)^2 and VDS=VDD-ID*RL,so Quadratic equation  is:IDS=KF*(VDD-ID*RL-VT)^2 ,where ID in mA
p = [256  -546.66  289]
ID=roots(p)*10^(-3)//values of ID converted into Ampere  by multiplying by 10^(-3)
disp("ID = "+string(ID)+" A") //   drain current EMOSFET in Ampere
VDS=VDD-ID*RL// Drain voltage for ID = 1.173 mA and ID = 0.962 mA
disp("VDS =VDD-ID*RL = "+string(VDS)+" volts") // Drain voltage
IDQ=0.962*10^(-3)
disp("IDQ ="+string(IDQ)+" A")//Since VDS < VT for ID=1.173 mA, hence ID = 1.173 mA cannot be chosen , so we chose ID= 0.962 mA as operating drain
```

```
current IDQ
Percentage_change=[(1-0.962)*100]/(1)
disp("Percentage change= "+string(Percentage_change)+" percent")// Percent change in IDQ  value from 1 mA(part(i)) to its  value ( of part(ii))IDQ =0.91mA
// NOTE: part(ii):the values of ID = 1.173 mA or ID = 0.962 mA but in book given as ID= 1.197 mA and ID = 0.939 mA .Hence (correct) Percentage_change in ID= 3.8 % but in book given as 6.1 %
```

Scilab code Exa 4.12 Find ON stage output voltage VDON for EMOSFET

```
//Ex4_12 Refer fig 4.9(a)and fig 4.9(b)
clc
VDD=(5)
disp("VDD= "+string(VDD)+" volts") // Drain voltage supply
RL1=125*10^(3)
disp("RL1= "+string(RL1)+ " ohm")  //Load resistance
RL2=200*10^(3)
disp("RL2= "+string(RL2)+ " ohm")  //Load resistance
IDON1=34.88*10^(-6)
disp("IDON1 ="+string(IDON1)+" A")//Drain current for load line 1 from fig.
IDON2=22.5*10^(-6)
disp("IDON2 ="+string(IDON2)+" A")//Drain current for load line 2 from fig.
VDON1=VDD-IDON1*RL1
disp("VDON1=VDD-IDON1*RL1= "+string(VDON1)+" volts")  //  output voltage  at drain terminal for IDON1
VDON2=VDD-IDON2*RL2
disp("VDON2=VDD-IDON2*RL2= "+string(VDON2)+" volts")  //  output voltage  at drain terminal for IDON2
```

Scilab code Exa 4.13 Verify DEMOSFET act as DC constant current source

```
//Ex4_13
clc
IDSS=10*10^(-3)
disp("IDSS = "+string(IDSS)+" ampere") //  maximum drain current for n-channel DEMOSFET
ID=IDSS // since VGS=0V, so ID=maximum
VP=(-4)
disp("VP= "+string(VP)+" volts") // pinch off voltage
VGS=(0)
disp("VGS= "+string(VGS)+" volts") // Gate to source voltage
VDD=(10)
disp("VDD= "+string(VDD)+" volts") // Drain supply voltage
RL=0.5*10^(3)
disp("RL= "+string(RL)+ " ohm")  //Load resistance
VDS=VDD-ID*RL
disp("VDS=VDD-ID*RL= "+string(VDS)+" volts") // Drain to source  voltage ,since VDS>VP DEMOSFET is in pinch off
disp("VDS>VP,so pinch off region")
RL=0.75*10^(3)
disp("RL= "+string(RL)+ " ohm")  // New Load resistance value
VDS=VDD-ID*RL
disp("VDS=VDD-ID*RL= "+string(VDS)+" volts") // New Drain to source  voltage for RL=750 ohm
disp("VDS<VP,so ohmic region")// since VDS < VP DEMOSFET is in ohmic region for RL=750 ohm and hence will not operate as a current source
```

Scilab code Exa 4.14 Find resistance value to obtain constant current source for MOS current mirror

```
//Ex4_14
clc
KF1=0.25*10^(-3)
disp("KF1 = "+string(KF1)+" A/V^2") //  Scale factor
KF2=KF1
disp("KF2 = "+string(KF2)+" A/V^2") //  Scale factor
IQ=1*10^(-3)
disp("IQ= "+string(IQ)+" ampere") // constant
    current  source value
VT1=2
disp("VT1 = "+string(VT1)+" volts")// Threshold
    voltage
VT2=VT1
disp("VT2 = "+string(VT2)+" volts")// Threshold
    voltage
VDD=(15)
disp("VDD= "+string(VDD)+" volts") // Drain supply
    voltage
IREF=IQ
disp("IREF =IQ= "+string(IREF)+" ampere") //
    Reference current value
VGS=VT1+sqrt(2*IREF/KF1) // Formulae
disp("VGS= VT1+sqrt(2*IREF/KF1)="+string(VGS)+"
    volts") // Gate to source voltage
R=(VDD-VGS)/IREF
disp("R= (VDD-VGS)/IREF="+string(R)+ " ohm")  //
    resistance value to obtain constant current
//  ERROR NOTE: values of VGS and R (correct) are
    4.8284271 volts and 10171.573 ohm  respectively
    but given in book are VGS=4V and R=11 kilo ohms
```

Scilab code Exa 4.15 Determine ON voltage and off isolation errors for RL and analog shunt switch

```
//Ex4_15
clc
RON=100
disp("RON= "+string(RON)+ " ohm")  //ON resistance of  analog series switch
ROFF=10^(10)
disp("ROFF= "+string(ROFF)+ " ohm")  //OFF resistance analog series switch
Vip=1
disp("Vip= "+string(Vip)+" volts")// Peak amplitude of analog voltage
Rs=100
disp("Rs= "+string(Rs)+ " ohm")  //Voltage source resistance
RL=10*10^(3)
disp("RL= "+string(RL)+ " ohm")  //Load resistance
disp("part(i) ")// part(i) of this question
Vo=(Vip*RL)/(RL+RON+Rs)
disp("Vo=(Vip*RL)/(RL+RON+Rs)= "+string(Vo)+" volts")// ON voltage
ErON=[Vip*(RON+Rs)/(RL+RON+Rs)]*100
disp("ErON=[Vip*(RON+Rs)/(RL+RON+Rs)]*100= "+string(ErON)+" percent")// Output voltage error
vOFF=(Vip*RL)/ROFF
disp("vOFF=(Vip*RL)/ROFF= "+string(vOFF)+" volts")// Output voltage in OFF state
OFF_isolation=20*log10(Vip/vOFF)
disp("OFF_isolation=20*log10(Vip/vOFF)= "+string(OFF_isolation)+" dB") // OFF_isolation=20*log10(Vip/vOFF) in dB// Thus ON error and OFF isolation decrease with increasing values of RL.
```

```
disp("part(ii) ")// part(ii) of this question
vOFF=(Vip*RON)/(Rs+RON)
disp("vOFF=(Vip*RON)/(Rs+RON)= "+string(vOFF)+" volts")// Output voltage in OFF state for analog shunt switch
OFF_isolation=20*log10((Rs+RON)/RON)// OFF_isolation of shunt switch
disp("OFF_isolation=20*log10((Rs+RON)/RON)= "+string(OFF_isolation)+" dB")// Thus shunt switch is inferior to series switch in its OFF isolation property

// ERROR NOTE: in question the author has given RL = 10K but in solution he has used RL = 1 k ... I have done programming using RL = 10 K.
```

Chapter 5

Basic Transistor Amplifiers

Scilab code Exa 5.1 Find voltage gain and input resistance and output resistance and current gain for CE and CS amplifiers

```
//Ex5_1
clc
RL=5*10^(3)
disp("RL= "+string(RL)+" ohm")  //Load resistance
R1=100*10^(3)
disp("R1= "+string(R1)+" ohm")  // resistance
R2=10*10^(3)
disp("R2= "+string(R2)+" ohm")  // resistance
rc=50*10^(3)
disp("rc= "+string(rc)+" ohm")  //collector
    resistance
rd=rc // Drain and collector  resistance are equal
rbe=1*10^(3)
disp("rbe= "+string(rbe)+" ohm")  //Load resistance
gm=50*10^(-3)
disp("gm = "+string(gm)+" A/V")//  transconductance
    for BJT
Av=(-gm*RL)
disp("For BJT,Av=(-gm*RL)= "+string(Av)) //Voltage
    gain for BJT
```

```
AI=gm*rbe
disp("AI=(gm*rbe)= "+string(AI)) // current gain for BJT
gm=5*10^(-3)
disp("gm = "+string(gm)+" A/V")//  transconductance for FET
Av=(-gm*RL)
disp("For FET,Av=(-gm*RL)= "+string(Av)+ " ") // gain for FET
R0=rd
disp("R0= "+string(R0)+ " ohm")  //output resistance for FET and BJT
Ri=rbe
disp("Ri= "+string(Ri)+ " ohm")  //BJT input resistance
RB=(R1*R2)/(R1+R2)
disp("RB=(R1*R2)/(R1+R2)= "+string(RB)+ " ohm")  // eqivalent Base resistance for  BJT
Ri=(RB*rbe)/(RB+rbe)
disp("Ri= (RB*rbe)/(RB+rbe)="+string(Ri)+ " ohm") //New value of BJT input resistance
```

Scilab code Exa 5.2 Find voltage gain and overall voltage gain and current gain and input resistance and output resistance for CC amplifier

```
//Ex5_2
clc
RL=5*10^(3)
disp("RL= "+string(RL)+ " ohm")  //Load resistance
R1=100*10^(3)
disp("R1= "+string(R1)+ " ohm")  // resistance
R2=100*10^(3)
disp("R2= "+string(R2)+ " ohm")  // resistance
Rs=5*10^(3)
disp("Rs= "+string(Rs)+ " ohm")  // Source
```

```
resistance
Beta_o=50
disp("Beta_o = "+string(Beta_o)) //BJT gain
rbe=1*10^(3)
disp("rbe= "+string(rbe)+ " ohm")  //Base-emitter resistance
gm=50*10^(-3)
disp("gm = "+string(gm)+" A/V")//  transconductance for BJT
rc=50*10^(3)
disp("rc= "+string(rc)+ " ohm")  //collector resistance
Av=RL/(RL+1/gm) // Gain formulae
disp("Av=RL/(RL+1/gm)= "+string(Av)) // voltage gain for BJT
Avs=RL/[(Rs/Beta_o)+(1/gm)+(RL)]
disp("Avs=RL/[(Rs/Beta_o)+(1/gm)+(RL)]= "+string(Avs)) // Overall voltage gain for BJT
AI=-(Beta_o+1)
disp("AI=-(Beta_o+1)= "+string(AI)) // current gain for BJT
R0=(Rs+rbe)/Beta_o
disp("R0= (Rs+rbe)/Beta_o="+string(R0)+ " ohm")  //output resistance for  BJT
Ri=rbe+Beta_o*RL // formulae
disp("Ri= rbe+Beta_o*RL="+string(Ri)+ " ohm")  //value of BJT input resistance
RB=(R1*R2)/(R1+R2)
disp("RB=(R1*R2)/(R1+R2)= "+string(RB)+ " ohm")  //eqivalent Base resistance for  BJT
Rieff=(Ri*RB)/(RB+Ri)
disp(" Rieff= (Ri*RB)/(RB+Ri)="+string(Rieff)+ " ohm")  //Effective value of BJT input resistance
```

Scilab code Exa 5.3 Find input resistance and voltage gain and current gain and output resistance for CE amplifier

```
//Ex5_3
clc
RL=5*10^(3)
disp("RL= "+string(RL)+ " ohm")  //Load resistance
RF=5*10^(3)
disp("RF= "+string(RF)+ " ohm")  // resistance
Beta_o=50
disp("Beta_o = "+string(Beta_o)) //BJT gain
rbe=1*10^(3)
disp("rbe= "+string(rbe)+ " ohm")  //Base-emitter resistance
gm=50*10^(-3)
disp("gm = "+string(gm)+" A/V")//  transconductance for BJT
rc=50*10^(3)
disp("rc= "+string(rc)+ " ohm")  //collector resistance
Ri=rbe+RF*(1+gm*rbe) // formulae
disp("Ri= rbe+RF*(1+gm*rbe)="+string(Ri)+ " ohm") //  BJT input resistance
Av=(-gm*RL)/(1+gm*RF)// formulae
disp("Av=(-gm*RL)/(1+gm*RF)= "+string(Av)) // voltage gain for BJT
AI=Beta_o
disp("AI=(Beta_o)= "+string(AI)) // current gain for BJT
R0=Beta_o*rc
disp("R0= Beta_o*rc="+string(R0)+ " ohm")  //output resistance for  BJT
```

Scilab code Exa 5.4 Find overall voltage gain Avs and input resistance Ri and output resistance Ro for CB and CG amplifier

```
//Ex5_4
clc
RL=5*10^(3)
disp("RL= "+string(RL)+ " ohm")  //Load resistance
RF=2.5*10^(3)
disp("RF= "+string(RF)+ " ohm")  // resistance
Rs=50
disp("Rs= "+string(Rs)+ " ohm")  // resistance
ro=50*10^(3)
disp("ro= "+string(ro)+ " ohm")  // output
   resistance
rd=ro // drain resistance
rc=ro// Collector resistance
disp("rc= "+string(rc)+ " ohm")  // Collector
   resistance
rbe=1*10^(3)
disp("rbe= "+string(rbe)+ " ohm")  //base -emitter
   resistance
disp("For CG Amplifier")
gm=5*10^(-3)
disp("gm = "+string(gm)+" A/V")//  transconductance
   for FET
Ri=1/gm // formulae
disp("Ri= 1/gm="+string(Ri)+ " ohm")  // value of
   CGA (common gate amplifier)input resistance for
   FET
Avs=gm*RL/(1+gm*Rs)
disp("Avs=gm*RL/(1+gm*Rs)= "+string(Avs)) // Overall
    voltage gain for FET (CGA)
Ro=rd*(1+gm*Rs)
disp("Ro=rd*(1+gm*Rs)="+string(Ro)+ " ohm")  //
   output resistance for  FET (CGA)
disp("For CB Amplifier")
gm=50*10^(-3)
disp("gm = "+string(gm)+" A/V")//  transconductance
   for BJT
Ri=1/gm // formulae
disp("Ri= 1/gm="+string(Ri)+ " ohm")  // value of
```

```
   CBA (common base amplifier)input resistance for
   BJT
Avs=gm*RL/(1+gm*Rs)
disp("Avs=gm*RL/(1+gm*Rs)= "+string(Avs)) // Overall
    voltage gain for BJT (CBA)
Ro=gm*(rbe*rc)
disp("Ro=gm*(rbe*rc)="+string(Ro)+ " ohm")  //output
    resistance for  BJT (CBA)

//NOTE: I have calculated first all the parameters
   for CG amplifier and then for CB amplifier but in
    book parameters have been calculated
   alternatingly for CG and CB amplifiers.
```

Scilab code Exa 5.5 Find high cut off frequency and bandwidth and unity gain bandwidth

```
//Ex5_5
clc
RL=5*10^(3)
disp("RL= "+string(RL)+ " ohm")  //Load resistance
Cc=0.1*10^(-6)
disp("Cc= "+string(Cc)+ " farad")  //capacitance
Ri=100*10^(3)
disp("Ri= "+string(Ri)+ " ohm")  //  input
   resistance for Amplifier
CSH=100*10^(-12)
disp("CSH= "+string(CSH)+ " farad")  //shunt load
   capacitance
Avm=100
disp("Avm="+string(Avm)) // Mid-frequency gain
fL=1/[2*(%pi)*(Ri)*(Cc)]
disp("fL=1/[2*(%pi)*(Ri)*(Cc)]= "+string(fL)+ "Hz ")
    // Lower cutoff-frequency
fH=1/[2*(%pi)*(RL)*(CSH)]
```

```
disp("fH=1/[2*(%pi)*(RL)*(CSH)]= "+string(fH)+" Hz") // Higher cutoff-frequency
BW=fH-fL
disp("BW=fH-fL= "+string(BW)+" Hz") // Bandwidth
fT=Avm*fH
disp("fT=Avm*fH= "+string(fT)+" Hz") // Unity gain bandwidth
// ERROR NOTE: calculated value of lower cutoff frequency , fL= 15.915494 Hz but in book given as 15.0 Hz
```

Scilab code Exa 5.6 Find percentage second harmonic distortion in small signal CS JFET amplifier

```
//Ex5_6
clc
IDSS=16*10^(-3)
disp("IDSS = "+string(IDSS)+" ampere") //  maximum drain current JFET
VP=(-4)
disp("VP= "+string(VP)+" volts") // pinch off voltage for JFET
VGSQ=(-2)
disp("VGSQ= "+string(VGSQ)+" volts") // Gate operating point voltage
Vsm=(0.2)
disp("Vsm= "+string(Vsm)+" volts") //  sinusoidal input voltage for JFET
D=[((0.5)*(Vsm)^2)/(4*Vsm)]*100 // derived from ID=IDSS(1-VGS/VP)^2 and putting value of VGS=VGSQ+Vs , where Vs=Vsm sinwt
disp("D=[((0.5)*(Vsm)^2)/(4*Vsm)]*100  ="+string(D)+"% ") //  Percentage second harmonic distortion calculation
```

Scilab code Exa 5.7 Compare perrformance of CE and CC and CB configuration of BJT

```
//Ex5_7
clc
Ic=1*10^(-3)
disp("Ic = "+string(Ic)+" ampere") //  collector current BJT
rbe=2*10^(3)
disp("rbe= "+string(rbe)+ " ohm")  //base -emitter resistance
gm=50*10^(-3)
disp("gm = "+string(gm)+" A/V")//  transconductance for BJT
Beta_o=100
disp("Beta_o = "+string(Beta_o)+" ") //BJT gain
rc=50*10^(3)
disp("rc= "+string(rc)+ " ohm")  //collector resistance
Cbe=10*10^(-12)
disp("Cbe= "+string(Cbe)+ " farad")  //base -emitter capacitance
Ctc=1*10^(-12)
disp("Ctc= "+string(Ctc)+ " farad")  //input device capacitance
disp("part(i)")// part(i)of question
RL=10*10^(3)
disp("RL= "+string(RL)+ " ohm")  //Load resistance
Rs=500
disp("Rs= "+string(Rs)+ " ohm")  //input source resistance
Rth=(Rs*rbe)/(Rs+rbe)
disp("Rth=(Rs*rbe)/(Rs+rbe)="+string(Rth)+ " ohm")
// eqivalent resistance
```

```
Avm=(-gm*RL)
disp("Avm=(-gm*RL)="+string(Avm)) // Mid-frequency gain for CE amplifier
CM=Ctc*(1-Avm)
disp("CM=Ctc*(1-Avm)= "+string(CM)+" farad") // calculated capacitance
Ci=Cbe
disp("Ci=Cbe= "+string(Ci)+" farad") //calculated input capacitance
fHi=1/[2*(%pi)*(Rth)*(Cbe+CM)]
disp("fHi=1/[2*(%pi)*(Rth)*(Cbe+CM)]= "+string(fHi)+" Hz") // Higher-frequency cutoff for CE amplifier
Ri=rbe
disp("Ri=rbe ="+string(Ri)+" ohm") //input resistance CE amplifier
Ro=rc
disp("R0= rc="+string(Ro)+" ohm") //output resistance for CE amplifier
fB=1/[2*(%pi)*(rbe)*(Cbe)]
disp("fB=1/[2*(%pi)*(rbe)*(Cbe)]= "+string(fB)+" Hz") // base terminal frequency cutoff
fT=Beta_o*fB
disp("fT=Beta_o*fB= "+string(fT)+" Hz") // Unity gain bandwidth for CE amplifier
disp("part(ii)")// part(ii)of question
Rs=50*10^(3)
disp("Rs= "+string(Rs)+" ohm") //input source resistance for CC amplifier
RL=1*10^(3)
disp("RL= "+string(RL)+" ohm") //Load resistance for CC amplifier
fhi=1/[2*(%pi)*(Rs)*(Ctc)]
disp("fhi=1/[2*(%pi)*(Rs)*(Ctc)]= "+string(fhi)+" Hz") // Higher-frequency cutoff for CC amplifier
Avm=(gm*RL)/(1+gm*RL)
disp("Avm=(gm*RL)/(1+gm*RL)="+string(Avm)) // Mid-frequency gain for CC amplifier
```

```
Ro=1/gm
disp("Ro= 1/gm="+string(Ro)+" ohm")  //output resistance for CC amplifier
Ri=Beta_o*RL
disp("Ri=Beta_o*RL ="+string(Ri)+" ohm")  //input resistance CE amplifier
disp("part(iii)")// part(iii)of question
RL=10*10^(3)
disp("RL= "+string(RL)+" ohm") //Load resistance for CB amplifier
Rs=50
disp("Rs= "+string(Rs)+" ohm")  //input source resistance for CB amplifier
fHi=gm/[2*(%pi)*(Cbe)]
disp("fHi=gm/[2*(%pi)*(Cbe)]= "+string(fHi)+" Hz") // Higher-frequency cutoff for CB amplifier
fHo=1/[2*(%pi)*(Ctc)*(RL)]
disp("fHo=gm/[2*(%pi)*(Ctc)*(RL)]= "+string(fHo)+" Hz") // Higher-frequency cutoff for CB amplifier
Avs=(gm*RL)/(1+gm*Rs)
disp("Avs=(gm*RL)/(1+gm*Rs)="+string(Avs)) // Mid-frequency gain for CB amplifier
Ri=1/gm
disp("Ri= 1/gm="+string(Ri)+" ohm")  //output resistance for CB amplifier
Ro=Beta_o*rc
disp("Ro=Beta_o*rc ="+string(Ro)+" ohm")  //input resistance CB amplifier
//ERROR NOTE:some parameters in the book have been calculated using gm=40 mA/V while given value is gm=50 mA/V. So ,for part(ii) CC amplifier correct value of R0=20 ohm,Ri=100000 ohm,and for part(iii)CB amplifier over all voltage gain Avs =142.85714 ,Ri=20 ohm all calculated for gm=50 mA/V.
```

Scilab code Exa 5.8 Find load resistance RL and coupling capacitor

```
//Ex5_8
clc
tp=10*10^(-3)
disp("tp= "+string(tp)+" s") // Time period of pulse
tr=0.05*10^(-6)
disp("tr= "+string(tr)+" s") // Rise-Time of pulse
CSH=50*10^(-12)
disp("CSH= "+string(CSH)+ " farad")  //output capacitor
tilt=5
disp("percentage tilt= "+string(tilt)+ "%")  //Sag or percentage tilt of output
Ri=100*10^(3)
disp("Ri= "+string(Ri)+ " ohm")  // source resistance
RL=tr/(2.2*CSH)
disp("RL=tr/(2.2*CSH)= "+string(RL)+ " ohm")  //Load resistance calculation
Cc=(tp*100)/(tilt*Ri)
disp("Cc= (tp*100)/( tilt*Ri)="+string(Cc)+ " farad")  //capacitance
//ERROR NOTE: calculated value of RL=454.54545 ohm but in book given as 455 ohm
```

Chapter 6

Multitransistor Multistage Amplifiers

Scilab code Exa 6.1 Express voltage gains in decibels

```
//Ex6_1
clc
Av=0.1
disp("Av= "+string(Av)) //Voltage gain
AvdB=20*log10(Av)
disp("Av(dB)=20*log10(Av)= "+string(AvdB)+"dB ") //
    Voltage gain in decibel
Av=0.707
disp("Av= "+string(Av)) //Voltage gain
AvdB=20*log10(Av)
disp("Av(dB)=20*log10(Av)= "+string(AvdB)+"dB ") //
    Voltage gain in decibel
Av=1
disp("Av= "+string(Av)) //Voltage gain
AvdB=20*log10(Av)
disp("Av(dB)=20*log10(Av)= "+string(AvdB)+"dB ") //
    Voltage gain in decibel
Av=10
disp("Av= "+string(Av)) //Voltage gain
```

```
AvdB=20*log10(Av)
disp("Av(dB)=20*log10(Av)= "+string(AvdB)+"dB ") //
    Voltage gain in decibel
Av=100
disp("Av= "+string(Av)) //Voltage gain
AvdB=20*log10(Av)
disp("Av(dB)=20*log10(Av)= "+string(AvdB)+"dB ") //
    Voltage gain in decibel
Av=1000
disp("Av= "+string(Av)) //Voltage gain
AvdB=20*log10(Av)
disp("Av(dB)=20*log10(Av)= "+string(AvdB)+"dB ") //
    Voltage gain in decibel
//NOTE: calculated voltage gain in dB for Av=0.707 is
    -3.0116117dB
```

Scilab code Exa 6.2 Find voltage gain Av and current gain Ai and power gain Ap for amplifier

```
//Ex6_2
clc
Ri=0.5*10^(3)
disp("Ri= "+string(Ri)+ " ohm")  // Amplifier input
    resistance
RL=0.05*10^(3)
disp("RL= "+string(RL)+ " ohm")  // Load resistance
Vom=1
disp("Vom= "+string(Vom)+" volts") // Output voltage
Vo=Vom/sqrt(2)//RMS value of Output voltage
Vim=1*10^(-3)
disp("Vim= "+string(Vim)+" volts") // Peak Input
    voltage
Vi=Vim/sqrt(2)//RMS Input voltage
Av=20*log10(Vo/Vi)
disp("Av(in dB)=20*log10(Vo/Vi)= "+string(Av)+" dB "
```

```
   ) //Voltage gain in decibel
Iim=Vim/Ri
disp("Iim= Vim/Ri= "+string(Iim)+" A") // Input peak
    current
Ii=Iim/sqrt(2) //RMS value of input current
Iom=Vom/RL
disp("Iom= Vom/RL= "+string(Iom)+" A") // Output
   peak current
Io=Iom/sqrt(2) //RMS value of Output current
Ai=20*log10(Io/Ii)
disp("Ai=20*log10(Io/Ii)= "+string(Ai)+" dB ") //
   Current gain in decibel
pi=Vi^2/Ri
disp("pi= Vi^2/Ri= "+string(pi)+" W") // Input power
po=Vo^2/RL
disp("po= Vo^2/RL= "+string(po)+" W") // Output
   power
Ap=10*log10(po/pi)
disp("Ap=10*log10(po/pi)= "+string(Ap)+" dB ") //
   Power gain in decibel
```

Scilab code Exa 6.3 Find common mode and differential gain and CMRR and input resistances and output voltage and error

```
//Ex6_3
clc
RL=1*10^(3)
disp("RL= "+string(RL)+ " ohm")  //Load resistance
RF=500*10^(3)
disp("RF= "+string(RF)+ " ohm")  //Feedback
   resistance
Beta_o=50
disp("Beta_o = "+string(Beta_o)) //BJT gain
rbe=1*10^(3)
disp("rbe= "+string(rbe)+ " ohm")  //Base-emitter
```

```
resistance
gm=50*10^(-3)
disp("gm = "+string(gm)+" A/V")//  transconductance for BJT
rc=50*10^(3)
disp("rc= "+string(rc)+ " ohm")  //collector resistance
disp("part(i)")
Adm1=(-gm*RL)
disp("Adm1=(-gm*RL)= "+string(Adm1)) // Differential mode gain for BJT for DIDO and SIDO modes
Adm2=(0.5*gm*RL)
disp("Adm2=(0.5*gm*RL)= "+string(Adm2)) // Differential mode gain for BJT for DISO and SISO modes
Rid=2*rbe
disp("Rid=2*rbe= "+string(Rid)+ " ohm")  //input differential mode resistance
Acm=(-RL)/(2*RF)
disp("Acm=(-RL)/(2*RF)= "+string(Acm)) // Common mode gain for BJT for DISO and SISO modes
Ric=Beta_o*RF
disp("Ric=Beta_o*RF= "+string(Ric)+ " ohm")  // common mode input resistance
CMRR=2*gm*RF
disp("CMRR=2*gm*RF= "+string(CMRR))  // common mode rejection ratio
disp("part(ii)")
Vi1=(-0.5)*10^(-3)
disp("Vi1= "+string(Vi1)+" volts") // input voltage1
Vi2=(+0.5)*10^(-3)
disp("Vi2= "+string(Vi2)+" volts") // input voltage2
Vcm=(10)*10^(-3)
disp("Vcm= "+string(Vcm)+" volts") // common mode voltage
Vd=Vi1-Vi2
disp("Vd=Vi1-Vi2= "+string(Vd)+" volts") // differential voltage
```

```
Vod=abs(Vd*Adm2)
disp("Vod=abs(Vd*Adm2)= "+string(Vod)+" volts") //output differential voltage for DISO and SISO modes
Voc=abs(Vcm*Acm)
disp("Voc=abs(Vcm*Acm)= "+string(Voc)+" volts") //output common mode voltage
Error=(Voc/Vod)*100
disp("percentage error=(Voc/Vod)*100= "+string(Error)+"%")//percentage error due to CM signal
disp("part(iii)")
RLeff=(RL*Rid)/(RL+Rid)
disp("RLeff=(RL*Rid)/(RL+Rid)= "+string(RLeff)+" ohm") // Effective load resistance
Adm=gm*RLeff
disp("Adm=gm*RLeff= "+string(Adm)) // Modified Differential mode gain for BJT for DIDO and SIDO modes
Acm=(-RLeff)/(2*RF)
disp("Acm=(-RLeff)/(2*RF)= "+string(Acm)) // Modified Common mode gain for BJT for DISO and SISO modes
CMRR=abs(Adm/(Acm))
disp("CMRR=abs(Adm/(Acm))= "+string(CMRR)) // Modified common mode rejection ratio
//NOTE: In Book, Formulae used for Acm in part(iii) is written as Acm=(-RL)/(2*RF)but ans is calculated by using RLeff in place of RL.So i have written formulae as Acm=(-RLeff)/(2*RF) in programming.
// Assigned variable name: in part(i) Adm for DIDO and SIDO modes is represented by Adm1 and Adm for DISO and SISO modes is represented by Adm2 to resist any anamoly in the programming.
```

Scilab code Exa 6.4 Find load resistance and collector to emitter voltage and output power and DC power and dissipated power

```
//Ex6_4
clc
VCC=(10)
disp("VCC= "+string(VCC)+" volts") // Collector voltage supply
VEE=VCC
disp("VEE=VCC= "+string(VEE)+" volts") // Emitter supply voltage
IQ=2*10^(-3)
disp("IQ = "+string(IQ)+" ampere") //  operating current for CC class-Aamplifier
VBE=(0.7)
disp("VBE= "+string(VBE)+" volts") // Base-emitter voltage
disp("part(i)")
RL=VCC/IQ
disp("RL=VCC/IQ= "+string(RL)+ " ohm")  //Load resistance
Pomax=VCC^2/(2*RL)
disp("Pomax=VCC^2/(2*RL)= "+string(Pomax)+" W") // maximum Output power
PDC=2*VCC*IQ
disp("PDC=2*VCC*IQ= "+string(PDC)+" W") // Total D.C power supply
Etta_max=(Pomax/PDC)*100
disp("Efficiency,Etta_max=(Pomax/PDC)*100= "+string(Etta_max)+"%") //maximum power amplifier conversion efficiency
PDmax=VCC*IQ
disp("PDmax=VCC*IQ= "+string(PDmax)+" W") // maximum power dissipation
disp("part(ii)")
Vcm=(5)
disp("Vcm= "+string(Vcm)+" volts") // common mode voltage
```

```
Po=Vcm^2/(2*RL)
disp("Po=Vcm^2/(2*RL)= "+string(Po)+" W") // Output power
Etta=(Po/PDC)*100
disp("Efficiency,Etta=(Po/PDC)*100= "+string(Etta)+" %") // power amplifier conversion efficiency
PDCavg=PDmax-Po//Using law of conservation of energy
disp("PDCavg=PDmax-Po= "+string(PDCavg)+" W") // Average power dissipated in BJT
```

Scilab code Exa 6.5 calculate output power Po and DC power PDC and efficiency at different conditions

```
//Ex6_5
// For  class-AB BJT amplifier
clc
VCC=(10)
disp("VCC= "+string(VCC)+" volts") // Collector voltage supply
VEE=VCC
disp("VEE=VCC= "+string(VEE)+" volts") // Emitter supply voltage
ICQ_0=10*10^(-3)
disp("ICQ_0 = "+string(ICQ_0)+" ampere") //  Zero signal collector current
RL=5
disp("RL= "+string(RL)+ " ohm")  //Load resistance
disp("part(i)")
Po=0// Since Output power at Zero signal condition is Zero
disp("Po="+string(Po)+" W") // Output power at Zero signal condition
PDC=2*VCC*ICQ_0
disp("PDC=2*VCC*ICQ_0= "+string(PDC)+" W") // Total D.C power supply for Zero signal condition
```

```
disp("part(ii)")
Vcm=VCC//For Full output voltage swing Vcm=VCC
disp("Vcm=VCC =" +string(Vcm)+" volts") // common mode voltage for full swing condition
Icm=VCC/RL
disp("Icm = VCC/RL="+string(Icm)+" ampere") // common mode current
Po=(1/2)*(Icm*Vcm)
disp("Po=(1/2)*(Icm*Vcm)="+string(Po)+" W") // Output power at full swing condition
ICavg=(Icm)/(%pi)
disp("ICavg=(Icm)/(%pi)="+string(ICavg)+" ampere") //  Average value of common mode current
PDC=2*(ICavg*VCC)
disp("PDC=2*VCC*ICavg= "+string(PDC)+" W") // Total D.C power supply for full swing condition
Etta=(Po/PDC)*100
disp("Efficiency,Etta=(Po/PDC)*100= "+string(Etta)+" %") // power amplifier conversion efficiency
disp("part(iii)")
Vcm1=(5)//given value
disp("Vcm1= "+string(Vcm1)+" volts") // common mode voltage for output swing Vcm=5 V
ICavg1=(Vcm1)/(%pi*RL)
disp("ICavg1=(Vcm1)/(%pi*RL)="+string(ICavg1)+" ampere") //  Average value of common mode current
Po1=(Vcm1^2)/(2*RL)
disp("Po1=(Vcm1^2)/(2*RL)="+string(Po1)+" W") // Output power for output swing Vcm=5 V
PDC1=2*(ICavg1*VCC)
disp("PDC1=2*VCC*ICavg1= "+string(PDC1)+" W") // Total D.C power supply for  output swing Vcm=5 V
Etta=(Po1/PDC1)*100
disp("Efficiency,Etta=(Po1/PDC1)*100= "+string(Etta)+" %") // power amplifier conversion efficiency for output swing Vcm=5 V
// NOTE: Correct value of Efficiency,Etta=(Po1/PDC1)*100= 39.269908 % for part(iii) but book ans is
```

```
    39.31%(because of approximation used during
    calculation)
```

Scilab code Exa 6.6 Find maximum difference voltage Vdmax for op amp without feedback

```
//Ex6_6
clc
Av=1*10^(5)
disp("Av= "+string(Av)) //Voltage gain
VCC=(10)
disp("VCC= "+string(VCC)+" volts") // Collector
    voltage supply
vo=VCC
disp("vo= VCC="+string(vo)+" volts") // maximum
    output voltage
Vdmax=VCC/Av
disp("Vdmax= VCC/Av="+string(Vdmax)+" volts") //
    Difference input voltage at OP-amp terminals
```

Chapter 7

Feedback Amplifiers and Sinusoidal oscillators

Scilab code Exa 7.1 Calculate amount of negative feedback and feedback factor

```
//Ex7_1
clc
A=60000
disp("A= "+string(A)) //Amplifier gain
Af=10000
disp("Af= "+string(Af)) //Feedback gain
N_dB=20*log10(Af/A)
disp("N_dB=20*log10(Af/A)= "+string(N_dB)+"dB") //
    Negative feedback gain
B=[1/(Af)]-(1/A)// formulae using [Af=A/(1+A*B)]
disp("B=[1/(Af)]-(1/A)= "+string(B)) //Feedback
    factor
```

Scilab code Exa 7.2 Find gain with feedback

```
//Ex7_2
clc
A=10000
disp("A= "+string(A)) //Amplifier gain
B=0.01
disp("B= "+string(B)) //Feedback factor
Af=[A/(1+A*B)]
disp("Af= [A/(1+A*B)]="+string(Af)) //Feedback gain
A1=100000
disp("A1= "+string(A1)) //New amplifier gain value
Af1=[A1/(1+A1*B)]
disp("Af1= [A1/(1+A1*B)]="+string(Af1)) //New feedback gain
```

Scilab code Exa 7.3 calculate amount of feedback and gain Af and new input voltage

```
//Ex7_3
clc
Vo=(50)
disp("Vo= "+string(Vo)+" volts") // output voltage
Vi=(0.5)
disp("Vi= "+string(Vi)+" volts") // input voltage
disp("part(i)")
A=Vo/Vi
disp("A= Vo/Vi="+string(A)) //Amplifier gain
H.D=10
disp("Harmonic_distortion="+string(H.D)+"%")//Percentage second harmonic distortion
D=(10*Vo)/100
disp("D= (10*Vo)/100 = "+string(D)+" volts") //Second Harmonic distortion
Df=(1*Vo)/100
disp("Df= (1*Vo)/100 = "+string(Df)+" volts") //Harmonic distortion with Feedback
```

```
B=[D/(Df*A)]-(1/A) //Using formulae Df=[D/(1+A*B)]
disp("B=[D/(Df*A)]-(1/A) = "+string(B)) //Feedback factor
disp("part(ii)")
Af=[A/(1+A*B)]
disp("Af= [A/(1+A*B)] = "+string(Af)) //Feedback gain
disp("part(iii)")
Vif=Vo/Af
disp("Vif= Vo/Af = "+string(Vif)+" volts") // New input voltage required
```

Scilab code Exa 7.4 find signal bandwidth and frequency below which midband gain will not deviate

```
//Ex7_4
clc
GBW=10^(6)
disp("GBW= "+string(GBW)+" Hz")// Gain-Bandwidth product
AMf=100
disp("AMf="+string(AMf)) // Midband gain with feedback
fHF=GBW/AMf
disp("fHF=GBW/AMf= "+string(fHF)+" Hz")//Signal bandwidth
f_10percent=(10*fHF)/100
disp("f_10percent=(10*fHF)/100= "+string(f_10percent)+" Hz")//Frequency below which AMf will not deviate by more than 10 percent
```

Scilab code Exa 7.5 find midband gain and higher cut off frequency fHf and lower cut off frequency fLf

```
//Ex7_5
clc
AM=50000
disp("AM="+string(AM)) // Midband gain
fH=20*10^(3)
disp("fH= "+string(fH)+" Hz")// Upper cut-off
    frequency
fL=30
disp("fL= "+string(fL)+" Hz")// Lower cut-off
    frequency
B=5*10^(-5)
disp("B= "+string(B)) //Feedback factor
AMf=AM/(1+B*AM)
disp("AMf=AM/(1+B*AM)="+string(AMf)) // Midband gain
     with feedback
fHf=fH*(1+B*AM)
disp("fHf=fH*(1+B*AM)= "+string(fHf)+" Hz")//Upper
    cut-off frequency with feedback
fLf=fL/(1+B*AM)
disp("fLf=fL/(1+B*AM)= "+string(fLf)+" Hz")//Lower
    cut-off frequency with feedback
//NOTE: calculated value of AMf is AMf=14285.714
    and fLF=8.5714286 but in book given as AMf=14286
     and fLF=8.58 Hz
```

Scilab code Exa 7.6 calculate phase margin for different feedback gains

```
//Ex7_6
//Refer fig7.4
clc
AM=100
disp("AM="+string(AM)+"dB") // Midband gain
fc1=1*10^(4)
disp("fc1= "+string(fc1)+" Hz")// First Critical
    frequency
```

```
fc2=10^5
disp("fc2= "+string(fc2)+" Hz")// Second Critical frequency
fc3=10^6
disp("fc3= "+string(fc3)+" Hz")// Third Critical frequency
disp("part(i)")
Af1=85
disp("Af1="+string(Af1)+"dB") //  gain at 50 kHz and -20dB/decade roll-off
f=50*10^(3)
disp("f= "+string(f)+" Hz")// operating frequency
theta_A=- atand(f/fc1)-  atand(f/fc2)-  atand(f/fc3)//phase shift in radians
disp("theta_A="+string(theta_A)+" degree")// Phase shift for feedback gain Af1
theta_pm=180-abs(theta_A)// formulae phase margin
disp("theta_pm=180-abs(theta_A)="+string(theta_pm)+" degree")// Phase Margin for feedback gain Af1
disp("Amplifier stable")// Since phase margin is (+)ive
disp("part(ii)")
Af2=50
disp("Af2="+string(Af2)+" dB") //  gain at 500 kHz and -40dB/decade roll-off
f=500*10^(3)
disp("f= "+string(f)+" Hz")//  frequency
theta_A=- atand(f/fc1)-  atand(f/fc2)-  atand(f/fc3)//phase shift in radians
disp("theta_A= "+string(theta_A)+" degree")//  Phase shift for feedback gain Af2
theta_pm=180-abs(theta_A)// formulae phase margin
disp("theta_pm=180-abs(theta_A)="+string(theta_pm)+" degree")// Phase Margin for feedback gain Af1
disp("Amplifier  unstable")// Since phase margin is (-)ive
disp("part(iii)")
Af3=20
```

```
disp("Af3="+string(Af3)+"dB") //  gain at 1100 kHz
    and -60dB/decade roll-off
f=1100*10^(3)
disp("f= "+string(f)+" Hz")//  frequency
theta_A=- atand(f/fc1)-  atand(f/fc2)-  atand(f/fc3)
    //phase shift in radians
disp("theta_A="+string(theta_A)+" degree")// Phase
    shift for feedback gain Af3
theta_pm=180-abs(theta_A)// formulae phase margin
disp("theta_pm=180-abs(theta_A)="+string(theta_pm)+"
     degree")// Phase Margin for feedback gain Af1
disp("Amplifier unstable")// Since phase margin is
    (-)ive
//NOTE:Correct ans for part(i) phase margin ,
    theta_pm=71.882476 degree but in book given as
    71.86 degree
// correct ans for part(iii) phase shift , theta_A
    =-222.01103 degree but in book given as -220.02
    degree
```

Scilab code Exa 7.7 Find maximum allowed input voltage without causing output clipping

```
//Ex7_7
clc
AV=50000
disp("AV="+string(AV)) // Voltage gain
Ri=50*10^(6)
disp("Ri= "+string(Ri)+ " ohm")  //Input resistance
    of OP-AMP
R0=1*10^(3)
disp("R0= "+string(R0)+ " ohm")  //Output resistance
AVf=10
disp("AVf="+string(AVf)) // Overall Voltage gain
RSf=50*10^(3)
```

```
disp("RSf= "+string(RSf)+ " ohm")  //Source
   resistance
R1=RSf
RF=AVf*(R1)
disp("RF=AVf*(R1)= "+string(RF)+ " ohm")  //Feedback
    resistance
VS=30
disp("VS= "+string(VS)+" volts") // Peak-peak output
    swing voltage
Vomax=0.5*(VS)
disp("Vomax=0.5*(VS)= -"+string(Vomax)+", +"+string(
   Vomax)+" volts") // Maximum output voltage swing
   at negative and positive polarities respectively
Vsmax=Vomax/AVf
disp("Vsmax=Vomax/AVf= -"+string(Vsmax)+", +"+string
   (Vsmax)+" volts") // Maximum output voltage
   without overload clipping at both polarities


//for overall voltage gain author has used two
   notations 'Avf' and 'Af' ... but I am working
   with 'Avf' only
```

Scilab code Exa 7.8 find output voltage and error for gain and value of gain to reduce output error

```
//Ex7_8
//refer fig. 7.6(b)
clc
R1=50*10^(3)
disp("R1= "+string(R1)+ " ohm")  // resistance at
   input terminal of OP-AMP
RF=500*10^(3)
disp("RF= "+string(RF)+ " ohm")  //Feedback
   resistance
```

```
VS=1
disp("VS= "+string(VS)+" volts") // Peak-peak output swing voltage
disp("part(i)")
disp("A = infinite")// voltage gain
Vo1=-(RF/R1) //Output voltage when gain, A=infinite
disp("Vo1=-(RF/R1)= "+string(Vo1)+" volts")
disp("part(ii)")
A=50000
disp("A="+string(A)) //  gain of OP-AMP
B=R1/(R1+RF)
disp("B=R1/(R1+RF)= "+string(B)) //Feedback factor
Vo2=-[(RF)*(B*A)]/(R1*(1+A*B))
disp("Vo2=-[(RF)*(B*A)]/(R1*(1+A*B))= "+string(Vo2)+" volts")// output voltage for A=50000
e=-[(Vo2-Vo1)*100]/Vo1
disp("%Error,e= [(Vo2-Vo1)*100]/Vo1="+string(e)+"% ")//  calculation for percentage error in output voltage
disp("part(iii)")
e=0.01
disp("%Error,e="+string(e)+"% ")//Given percentage error in output voltage
Vo3=-(Vo1-(e*Vo1/100))
disp("Vo3=Vo1-(e*Vo1/100)= "+string(Vo3)+" volts")// output voltage for error 0.01%
x=Vo3*(R1/RF)
A=(x)/[B*(1-x)] //using formulae Vo=-(RF/R1)*[(B*A)/1+A*B)]
disp("A=(Vo*R1)/[B*RF*(1-(Vo*RF/R1))]="+string(A)) // New Required gain for error less than 0.01%

// while solving the problem I have used 'e' for the error as no varriable is given for the same in textbook by author
// in textbook author has used 'Vo' for output voltage in all parts.. but to remove any ambiguity in the programe I have used 'Vo1' 'Vo2'
```

```
'Vo3' for part i, ii, iii, respectively
```

Scilab code Exa 7.9 find feedback resistances RF and R1 and resulting output resistance and voltage gain and power gain with feedback

```
//Ex7_9
clc
AV=100000
disp("AV="+string(AV)) // Voltage gain
Ri=10*10^(3)
disp("Ri= "+string(Ri)+" ohm") //Input resistance of OP-AMP
Ro=10
disp("Ro= "+string(Ro)+" ohm") //Output resistance
Rs=10*10^(6)
disp("Rs= "+string(Rs)+" ohm") //Source resistance
RL=1*10^(3)
disp("RL= "+string(RL)+" ohm") //Load resistance
B=(Rs-Ri)/(AV*Ri)
disp("B=(Rs-Ri)/(AV*Ri)= "+string(B)) //Feedback factor
AVf=AV/(1+B*AV)
disp("AVf=AV/(1+B*AV)="+string(AVf)) // Overall Voltage gain with feedback
Rof=Ro/(1+B*AV)
disp("Rof=Ro/(1+B*AV) ="+string(Rof)+" ohm") // output resistance with feedback
Rif=Ri*(1+B*AV)
disp("Rif=Ri/(1+B*AV) ="+string(Rif)+" ohm") // Input resistance with feedback
Ap=(AVf^2)*(Rif/RL)
disp("Ap=(AVf^2)*(Rif/RL)="+string(Ap)) // Overall Power gain
AP=10*log10(Ap)
disp("AP=10*log10(Ap)="+string(AP)+"dB") // Overall
```

Power gain in dB

Scilab code Exa 7.10 Find different parameters for Pierce oscillator

```
//Ex7_10
clc
gm=10*10^(-3)
disp("gm = "+string(gm)+" A/V")//  transconductance
Cgs=5*10^(-12)
disp("Cgs= "+string(Cgs)+ " farad")  // capacitance
    between gate-source
Cds=1*10^(-12)
disp("Cds= "+string(Cds)+ " farad")  // capacitance
    between drain-source
rd=50*10^(3)
disp("rd= "+string(rd)+ " ohm")  //Drain resistance
RG=10*10^(6)
disp("RG= "+string(RG)+ " ohm")  //Gate resistance
Rse=1*10^(3)
disp("Rse= "+string(Rse)+ " ohm")  //Gate resistance
L=0.5
disp("L= "+string(L)+ " H")  //Inductance
C2=0.05*10^(-12)
disp("C2= "+string(C2)+ " farad")  // Crystal
    parameter
C1=1*10^(-12)
disp("C1= "+string(C1)+ " farad")  // Crystal
    parameter
disp("part(i)")
x=C1+[(Cds*Cgs)/(Cds+Cgs)]
CT=1/[(1/C2)+(1/x)]
disp("CT= "+string(CT)+ " farad")  // Equivalent
    series-resonating capacitance
disp("part(ii)")
fo=sqrt(2)/[2*%pi*sqrt(L*CT)]
```

```
disp("fo= sqrt(2)/[2*%pi*sqrt(L*CT)]="+string(fo)+" Hz")// frequency of oscillations
disp("part(iii)")
z=sqrt((L*C1*C2)/(C1+C2))
fp=1/[2*%pi*z]
disp("fp= "+string(fp)+" Hz")// parallel-resonant frequency
p=sqrt(L*C2)
fs=1/[2*%pi*p]
disp("fs= "+string(fs)+" Hz")// series-resonant frequency
Q=[sqrt(L/C2)]/(Rse)
disp("Q=[sqrt(L/C2)]/(Rse)= "+string(Q))// Quality factor
disp("part(iv)")
AB=gm*rd*(Cds/Cgs)
disp("AB=gm*rd*(Cds/Cgs)= "+string(AB))//Loop gain
T_bias=RG*(Cgs+Cds)
disp("T_bias=RG*(Cgs+Cds)= "+string(T_bias)+"s")// Bias Time-Constant
T_r = 1/(2*%pi*fo)
disp("T_r =1/(2*%pi*fo)= "+string(T_r)+"s")// resonant Time-Constant for 'fo'
disp("for proper operation T_bias >> T_r")



// in part (ii)... value calculated for series resonant frequecy 'fo' is wrong in textbook.
// NOTE: in part(iii)... there is a misprint in the calculated value of Quality factor 'Q' in the textbook.
//I have used T_r instead of 1/wo (given in the book)
```

Chapter 8

Linear Op Amp Applications

Scilab code Exa 8.1 calculate feedback factor and variation in feedback gain Af

```
//Ex8_1
clc
Amin=8000
disp("Amin="+string(Amin)) // Minimum gain of OP-AMP
Amax=64000
disp("Amax="+string(Amax)) // Maximum gain
disp("part (i)")
delta_Af=0.01
disp("delta_Af="+string(delta_Af)) // Change in
    overall feedBack gain
delta_A=(Amax-Amin)/Amin
disp("delta_A= (Amax-Amin)/Amin = "+string(delta_A))
     // Change in open loop gain
Sg = delta_Af/delta_A
B = (1/Sg - 1)/Amax
disp("Sg = delta_Af/delta_A = "+string(Sg))//
    desensitivity factor
disp(" B = (1/Sg - 1)/Amax = "+string(B))//feedBack
    factor
 disp(" part (ii)")
```

```
Af_min = Amin/(1+B*Amin)//minimum change in overall feedBack gain
Af_max = Amax/(1+B*Amax)///maximum change in overall feedBack gain
disp("Af_min = Amin/(1+B*Amin) = "+string(Af_min))
disp("Af_max = Amax/(1+B*Amax) = "+string(Af_max))
disp("variation in Af = "+string(Af_max/Af_min))//variation in Af with feedBack factor 'B'


// for above problem author has divided question in two parts but during solution has written 3 parts.
// part (i) and part (ii) combinedly equivlent to part (i)
// part (iii) is equivalent to part (ii)
```

Scilab code Exa 8.2 Design an inverting op amp

```
//Ex8_2
clc
Avf=-100
disp("Avf="+string(Avf)) // Voltage gain
Rif=1
disp("Rif= "+string(Rif)+" ohm") //Input resistance of OP-AMP
R1=Rif
RF=-R1*Avf // using formulae Vo=(-RF/R1)*Vi
disp("RF= -R1*Avf="+string(RF)+" ohm") //Feedback resistance of OP-AMP
// NOTE: Error in value of RF since they have given value of Rif=1ohm but calculated RF by using Rif=1 Kilo ohm
// So i have calculated using Ri=1ohm and hence RF=100 ohm
```

Scilab code Exa 8.3 Find resistances R11 and R12 and R13 for op amp adder and output voltage

```
//Ex8_3
clc
R11=1*10^(3)
disp("R11= "+string(R11)+ " ohm")  // resistance at
    input terminal of OP-AMP Adder
RF=100*10^(3)
disp("RF= "+string(RF)+ " ohm")  //Feedback
    resistance
R12=10*10^(3)
disp("R12= "+string(R12)+ " ohm")  // resistance at
    input terminal of OP-AMP  Adder
R13=100*10^(3)
disp("R13= "+string(R13)+ " ohm")  // resistance at
    input terminal of OP-AMP  Adder
disp("vo = -("+string(RF/R11)+"vs1 +"+string(RF/R12)
    +"vs2 +"+string(RF/R13)+"vs3)") // output voltage
     of opamp adder in terms of input vs1,vs2 vs3

// for average value of input signal
n = 3// given inputs are '3'
R11 = n*RF
R12 = n*RF
R13 = n*RF
disp("vo = -("+string(RF/R11)+" vs1 +"+string(RF/R12
    )+" vs2 +"+string(RF/R13)+ " vs3)") // output
    voltage of opamp adder


// note : the output voltage of inverting adder is
    negative
// but while calculating weighted output voltage in
```

```
   above question ..author has neglected  or miss
   the negative sign
```

Scilab code Exa 8.4 Find scale factor of photometer

```
//Ex8_4
clc
Ir=10*10^(-3)
disp("Ir = "+string(Ir)+" ampere/lumen of radiant
   energy ") //photodiode Reverse saturation current
    for constant reverse bias VR
RF=10*10^(3)
disp("RF= "+string(RF)+ " ohm")  //Feedback
   resistance
E=1*10^(-2)
disp("E = "+string(E)+" lumens")//  radiant energy
IR=Ir*E
disp("IR =Ir*E= "+string(IR)+" ampere") // Reverse
   saturation current
Vo=IR*RF
disp("Vo=IR*RF= "+string(Vo)+" volts") //  output
   voltage
s=E/Vo
disp("scale factor=E/Vo= "+string(E)+" lumens/V") //
     Scale factor of photometer
```

Scilab code Exa 8.5 Find effective resistance and sensitivity and feedback resistance RF of DC ammeter

```
//Ex8_5
clc
Av=1*10^(5)
disp("Av= "+string(Av)) //Voltage gain
```

```
RF=100*10^(3)
disp("RF= "+string(RF)+ " ohm")  //Feedback
    resistance
RM=10*10^(3)
disp("RM= "+string(RM)+ " ohm")  // D.C Ammeter
    internal resistance
is=10*10^(-6)
disp("is = "+string(is)+" ampere") //  Source
    current
vo=is*RF
disp("vo=is*RF= "+string(vo)+" volts") //  output
    voltage
S=vo/is
disp("S=vo/is= "+string(S)+" V/A") // Sensitivity of
     Ammeter
Rif=RF/(1+Av)
disp("Rif=RF/(1+Av)= "+string(Rif)+ " ohm")  //Input
     resistance of OP-AMP
im=100*10^(-6)
disp("im = "+string(im)+" ampere") // Meter Full-
    Scale deflection current
RF=(im*RM)/is
disp("RF=(im*RM)/is= "+string(RF)+ " ohm")  // New
    required Feedback resistance for im=100 micro
    ampere
```

Scilab code Exa 8.6 Find resistances R1 and RF for non inverting op amp

```
//Ex8_6
clc
Av=36
disp("Av= "+string(Av)+" dB") //Voltage gain
R1=1*10^(3)// Choosing value of R1
disp("R1= "+string(R1)+ " ohm")  // Resistor at
    input side of OP-AMP
```

```
RF=R1*[10^(Av/20)-1] // Using formulae Av=20*log(1+
    RF/R1)
disp("RF=R1*[10^(Av/20)-1]= "+string(RF)+ " ohm")
    // Calculated Feedback resistance
//NOTE: Correct value of RF=62095.734 ohm or 62.095
    kilo ohm but in book given as 62.24 kilo ohm
```

Scilab code Exa 8.7 Find resistances R1 and input feedback resistance Rif for full scale moving coil meter

```
//Ex8_7
clc
if=100*10^(-6)
disp(" if = "+string(if)+" ampere") //Full-Scale
    deflection current
Av=1*10^(5)
disp("Av= "+string(Av)) //Voltage gain
vs=10*10^(-3)
disp(" vs= "+string(vs)+" volts") //  Input voltage
RM=100
disp("RM= "+string(RM)+ " ohm")  // Moving coil
    Ammeter internal resistance
Ri=10*10^(3)
disp(" Ri= "+string(Ri)+ " ohm")  //Input resistance
    of OP-AMP
R1=vs/if
disp("R1=vs/if= "+string(R1)+ " ohm")  // Resistor
    at input side of OP-AMP in Voltage-to-Current
    converter
Avf=1+(RM/R1) // formulae using Avf=1+(RF/R1)=1+(RM/
    R1)// since RF=RM
disp(" Avf=1+(RM/R1)="+string(Avf)) // Overall
    Voltage gain
Rif=Ri*(Av/Avf)
disp(" Rif=Ri*(Av/Avf)="+string(Rif)+ " ohm")  //
```

Equivalent input side resistance of OP-AMP with feedback

Scilab code Exa 8.8 Find worst case change in output voltage Vo

```
//Ex8_8
clc
Ro=0.001
disp("Ro= "+string(Ro)+ " ohm")  //Output resistance
Sv=0.01
disp("Sv= "+string(Sv)+"%") // Input Regulation for
   IC regulator
delta_VI=12-9
disp("change in regulator voltage= "+string(delta_VI
   )+" volts") // Regulator input voltage variation
delta_IL=1.25-1
disp("change in regulator Current= "+string(delta_IL
   )+" A") // Regulator Current variation
delta_Vo=[delta_VI*(Sv/100)+delta_IL*Ro]
disp("change in regulator output voltage= "+string(
   delta_Vo)+" volts") // Regulator output voltage
   variation
```

Scilab code Exa 8.9 Design 2nd order LP Butterworth filter

```
//Ex8_9
clc
alpha=1.414// Damping coefficient for Butterworth LP
    filter
disp("alpha="+string(alpha))
AM=3-alpha
disp("AM="+string(AM)) // Midband gain of filter
f0H=1*10^(3)
```

```
disp("fOH= "+string(fOH)+" Hz")//Cut off frequency
R1=10*10^(3)// Choosing value of R1 same as in book
disp("R1= "+string(R1)+ " ohm")  // Resistor at input side of (OP-AMP) filter
RF=R1*(AM-1)
disp("RF=R1*(AM-1)="+string(RF)+ " ohm")  //Feedback resistance
C=0.1*10^(-6) // Choosing value of capacitor same a in book
disp("C="+string(C)+"farad")
R=1/(2*%pi*fOH*C)// Using formulae wOH=1/C*R and wOH=(2*%pi*fOH)
disp("R=1/(omega_OH*C)=1/(2*%pi*fOH*C)="+string(R)+" ohm")  // resistance value for filter design
```

Scilab code Exa 8.10 Design 2nd order single op amp band pass filter

```
//Ex8_10
clc
fo=150
disp("fo= "+string(fo)+" Hz")//Central frequency of band pass filter
BW=15
disp("BW= "+string(BW)+" Hz")// Upper cut-off frequency or 3-dB bandwidth
Q=fo/BW // Quality factor
disp("Q= "+string(Q))
C=0.05*10^(-6) // Choosing value of capacitor same as in book
disp("C="+string(C)+"farad")
R=sqrt(2)/(2*%pi*fo*C)
disp("R=sqrt(2)/(2*%pi*fo*C)="+string(R)+ " ohm")  // resistance value for filter design
Am=5-(sqrt(2)/Q) // formulae
disp("Am=5-(sqrt(2)/Q)="+string(Am)) // Midband gain
```

```
Abp=Am/(5-Am)
disp("Abp=Am/(5-Am)="+string(Abp)) // Central frequency gain
```

Scilab code Exa 8.11 Determine filter response for state variable filter

```
//Ex8_11
clc
R=10*10^(3)
disp("R= "+string(R)+ " ohm")  // resistance
R1=10*10^(3)
disp("R1= "+string(R1)+ " ohm")  // resistance
C=0.01*10^(-6) //  value of capacitor
disp("C="+string(C)+" farad")
R1_ratio_K=2.5*10^(3)
disp("R1_ratio_K= "+string(R1_ratio_K)+ " ohm")  // resistance
R2=5*10^(3)
disp("R= "+string(R)+ " ohm")  // resistance
alpha_R2=250
disp("alpha_R2= "+string(alpha_R2)+ " ohm")  // resistance
alpha=alpha_R2/R2
disp("alpha=alpha_R2/R2= "+string(alpha))  // Damping factor
Q=1/alpha
disp("Q= 1/alpha="+string(Q))// Quality factor
omega_o=1/(R*C)
disp("omega_o=1/(R*C)= "+string(omega_o)+" radian") // centre angular frequency
BW=omega_o/Q
disp("Bandwidth=omega_o/Q= "+string(BW)+" radian")// Upper cut-off frequency or 3-dB bandwidth
K=R1/(R1_ratio_K)// Pass band gain for lPF and HPF of state variable filter
```

```
disp("K=R1/(R1_ratio_K)= "+string(K))
Gm=K/alpha// Pass band gain of state variable filter
disp("center frequency gain for BPF, K/alpha=K*Q= "+string(Gm)) // Centre frequency gain for BP filter
```

Scilab code Exa 8.12 Find output offset voltage with and without comensation

```
//Ex8_12
clc
IB=0.5*10^(-6)
disp("IB = "+string(IB)+" ampere") //Input bias current
Iio=0.05*10^(-6)
disp("Iio = "+string(Iio)+" ampere") //Input offset current
Vio=1*10^(-3)
disp("Vio= "+string(Vio)+" volts") //Input offset voltage
R1=10*10^(3)
disp("R1= "+string(R1)+ " ohm")  // resistance
RF=500*10^(3)
disp("RF= "+string(RF)+ " ohm")  //Feedback resistance
Vos1=Vio*(1+RF/R1)
disp("Vos1=Vio*(1+RF/R1)="+string(Vos1)+" volts") // output offset voltage due to input offset voltage
Vos2=IB*RF
disp("Vos2=IB*RF="+string(Vos2)+" volts") //output offset voltage due to Input bias current
Vos=Vos1+Vos2
disp("Vos=Vos1+Vos2="+string(Vos)+" volts") //total output offset voltage
R2=(R1*RF)/(R1+RF)
```

```
disp("R2=(R1*RF)/(R1+RF)= "+string(R2)+ " ohm")   //
   resistance to balance IB effect
Vos2=Iio*RF
disp("Vos2=Iio*RF="+string(Vos2)+" volts") //
   Reduced output offset voltage due to Input offset
    current
Vos=Vos1+Vos2
disp("Vos=Vos1+Vos2="+string(Vos)+" volts") //
   output offset voltage with compensation
```

Scilab code Exa 8.13 find resistance R2 to minimize offset and output offset voltage at 25 and 75 degree celsius

```
//Ex8_13
clc
Iio=0.1*10^(-9)
disp("Iio = "+string(Iio)+" ampere/degree _celsius")
    //Input offset current
Vio=10*10^(-6)
disp("Vio= "+string(Vio)+" volt/degree _celsius") //
   Input offset voltage
Vs=10*10^(-3)
disp("Vs= "+string(Vs)+" volts") //Input voltage
R1=10*10^(3)
disp("R1= "+string(R1)+ " ohm")   // resistance
RF=100*10^(3)
disp("RF= "+string(RF)+ " ohm")   //Feedback
   resistance
disp("part(i)")
R2=(R1*RF)/(R1+RF)// R1 in parallel with RF
disp("R2=(R1*RF)/(R1+RF)= "+string(R2)+ " ohm")   //
   resistance to balance IB i.e offset effect
disp("part(ii)")
delta_T=75-25
disp("delta_T=75-25 = "+string(delta_T)+"
```

```
degree_celsius") //Temperature change
delta_Vo=[(Vio*delta_T)*(1+RF/R1)]+(Iio*delta_T*RF)
disp("delta_Vo=[(Vio*delta_T)*(1+RF/R1)]+(Iio*delta_T*RF)= "+string(delta_Vo)+" volts") //Output voltage drift
disp("part(iii)")
Vo=(-RF/R1)*Vs
disp("Vo=(-RF/R1)*Vs= "+string(Vo)+" volts") //Inverting OP-AMP output voltage
e=(delta_Vo/Vo)*100
disp("Percentage error=(delta_Vo/Vo)*100 =(-)"+string(abs(e))+", (+)"+string(abs(e))+"  percent")//percentage error
```

Scilab code Exa 8.14 Find power supply voltage regulation

```
//Ex8_14
clc
Iio=0.1*10^(-9)
disp("Iio = "+string(Iio)+" ampere") //Input offset current
VCC=15
disp("VCC= "+string(VCC)+" volts") //  voltage supply
PSRR=150*10^(-6)
disp("PSRR= "+string(PSRR)+"  volts/V")// Power supply rejection ratio
Vio=10*10^(-6)
disp("Vio= "+string(Vio)+" volts") //Input offset voltage
R1=10*10^(3)
disp("R1= "+string(R1)+ " ohm")  // resistance
RF=100*10^(3)
disp("RF= "+string(RF)+ " ohm")  //Feedback resistance
```

```
delta_T=75-25
disp("delta_T=75-25 = "+string(delta_T)+" celsius")
    //Temperature change
delta_Vo=[(Vio*delta_T)*(1+RF/R1)]+(Iio*delta_T*RF)
disp("delta_Vo=[(Vio*delta_T)*(1+RF/R1)]+(Iio*
    delta_T*RF)= "+string(delta_Vo)+" volts") //
    Output voltage drift
delta_Vio1=(delta_Vo)*(R1/RF)
disp("delta_Vio1=(delta_Vo)*(R1/RF)= "+string(
    delta_Vio1)+" volts") // voltage change at Input
    for voltage drift found
delta_Vio2=(delta_Vio1)*(1/10)
disp("delta_Vio2=(delta_Vio1)*(1/10)= "+string(
    delta_Vio2)+" volts") //  change in Vio due to
    PSRR
p=[(delta_Vio2)/(VCC*PSRR)]*100
disp("power supply regulation=[(delta_Vio2)/(VCC*
    PSRR)]*100 ="+string(p)+ " percent")// power
    supply regulation requirement

//delta_Vio1 corresponds to voltage change at Input
    for voltage drift found
//delta_Vio2 corresponds voltage change at input due
     to PSRR
```

Scilab code Exa 8.15 calculate full power bandwidth

```
//Ex8_15
clc
SR=0.65
disp("SR= "+string(SR)+"  volts/microsecond")// Slew
     rate of OP-AMP
disp("part(i)")
Vm=5
disp("Vm= "+string(Vm)+" volts") //  Output peak
```

```
voltage1
fsm=SR/[10^(-6)*(2*%pi*Vm)] // using formulae SR=2*pi*fsm*Vm
disp("fsm=SR/[10^(-6)*(2*%pi*Vm)] = "+string(fsm)+" Hz")// // Full power bandwidth for Output peak voltage Vm=5V
disp("part(ii)")
Vm=1
disp("Vm= "+string(Vm)+" volts") // Output peak voltage2
fsm=SR/[10^(-6)*(2*%pi*Vm)] // using formulae SR=2*pi*fsm*Vm
disp("fsm=SR/[10^(-6)*(2*%pi*Vm)] = "+string(fsm)+" Hz")// // Full power bandwidth for Output peak voltage Vm=1V
```

Chapter 9

Digital Circuits and Logic Families

Scilab code Exa 9.1 Find minimum gain for proper operation of NOT gate

```
//Ex9_1
clc
VCC=5
disp("VCC= "+string(VCC)+" volts") //  voltage
    supply
RB=10*10^(3)
disp("RB= "+string(RB)+ " ohm")  // Base-resistance
RL=1*10^(3)
disp("RL= "+string(RL)+ " ohm")  // Load resistance
VCS=0.2
disp("VCS= "+string(VCS)+" volts") // collector
    saturated voltage
VBS=0.8
disp("VBS= "+string(VBS)+" volts") // Base voltage
    at saturation
V_gamma=0.6
disp("V_gamma= "+string(V_gamma)+" volts") //
    Threshold or cut-in voltage
ICS=(VCC-VCS)/RL
```

```
disp("ICS = (VCC-VCS)/RL="+string(ICS)+" ampere") // Saturation collector current of transistor T1
vi=5
disp("vi= "+string(vi)+" volts") //  Input voltage
IBS=(vi-VBS)/RB
disp("IBS=(vi-VBS)/RB="+string(IBS)+" ampere") // Forward base drive required to sustain ICS
Beta_Fmin=ICS/IBS
disp("Beta_Fmin=ICS/IBS= "+string(Beta_Fmin)) // Common-emitter current gain

//NOTE:  Correct formulae for ICS=(VCC-VCS)/RL
// but in book it is written wrong as ICS=(VCC-VCS)/RB but had calculated ans (in book) according to correct formulae ICS=(VCC-VCS)/RL
```

Scilab code Exa 9.2 calculate output voltage Vx for OR logic gate

```
//Ex9_2 Refer fig.9.3(e)
clc
VD=0.7
disp("VD= "+string(VD)+" V") //  Diode voltage drop in conduction mode

disp("part(i)")// part(i)of question
vA=0
disp("vA= "+string(vA)+" V") //   Input voltage1 of diode OR logic gate
vB=0
disp("vB= "+string(vB)+" V") //  Input voltage2 of diode OR logic gate
vX=0 // Since both input voltages vA=vB=0V
disp("vX="+string(vX)+" V") //  Output voltage of diode OR logic gate for part(i)

```

```
disp("part(ii)")// part(ii)of question
vA=0
disp("vA= "+string(vA)+" V") //  Input voltage1 of
    diode OR logic gate
vB=5
disp("vB= "+string(vB)+" V") //  Input voltage2 of
    diode OR logic gate for  SECOND CASE: when vA=0V
    and vB=5V
vX=vB-VD
disp("vX=vB-VD= "+string(vX)+" V") //  Output
    voltage of diode OR logic gate for  SECOND CASE

disp("part(iii)")// part(iii)of question
vA=5
disp("vA= "+string(vA)+" V") //  Input voltage1 of
    diode OR logic gate for  THIRD CASE  when vA=5V
    and vB=0V
vB=0
disp("vB= "+string(vB)+" V") //  Input voltage2 of
    diode OR logic gate
vX=vA-VD
disp("vX=vA-VD= "+string(vX)+" V") //  Output
    voltage of diode OR logic gate for  THIRD CASE

disp("part(iv)")// part(iv)of question
vA=(+5)
disp("vA= "+string(vA)+" V") //  Input voltage1 of
    diode OR logic gate
vB=(+5)
disp("vB= "+string(vB)+" V") //  Input voltage2 of
    diode OR logic gate
vX=vA-VD // Since both diodes D1 and D2 are
    conducting
vX=vB-VD
disp("vX=vA-VD=vB-VD= "+string(vX)+" V") //  Output
    voltage of diode OR logic gate for FOURTH CASE:
    when vA=5V and vB=5V
```

Scilab code Exa 9.3 Calculate output voltage Vx for diode AND logic gate

```
//Ex9_3 Refer fig.9.4(e)
// For AND logic gate
clc
VD=0.7
disp("VD= "+string(VD)+" V") //  Diode voltage drop in conduction mode

disp("part(i)")
vA=0
disp("vA= "+string(vA)+" V") //  Input voltage of diode AND logic gate
vB=0
disp("vB= "+string(vB)+" V") //  Input voltage2 of diode  AND logic gate

vX=VD // Since both input voltages vA=vB=0V
disp("vX=VD="+string(vX)+" V") //  Output voltage of diode  AND logic gate for FIRST CASE: when vA=0V and vB=0V

disp("part(ii)")
vA=0
disp("vA= "+string(vA)+" V") //  Input voltage of diode AND logic gate
vB=5
disp("vB= "+string(vB)+" V") //  Input voltage2 of diode AND logic gate for  SECOND CASE: when vA=0V and vB=5V
vX=VD //due to diode A which is  conducting and the Diode B is reverse biased with a voltage VD-VB =0.7-5=-4.3
disp("vX=VD "+string(vX)+" V")
```

```
//due to diode B which is conducting



disp("part(iii)")
vA=5
disp("vA= "+string(vA)+" V") //  Input voltage for THIRD CASE  when vA=5V and vB=0V
vB=0
disp("vB= "+string(vB)+" V") //  Input voltage2 of diode  AND logic gate
vX = VD//due to diode B which is  conducting and the Diode A is reverse biased with a voltage VD-VA =0.7-5=-4.3
disp("vX= "+string(vX)+" V")

disp("part(iv)")
vA=5
disp("vA= "+string(vA)+" V") //  Input voltage forfourth CASE  when vA=5V and vB=5V
vB=5
disp("vB= "+string(vB)+" V") //  Input voltage2 of diode AND logic gate for  CASE: when vA=0V and vB =5V
vX=vA // Since both diodes D1 and D2 are Non-conducting, so no voltage drop across'R'(resistor)
disp("vX = vA = vB= "+string(vX)+" V") //  Output voltage of diode  AND logic gate for FOURTH CASE: when vA=5V and vB=5V
```

Scilab code Exa 9.6 Find low and high level noise immunities

```
//Ex9_6
clc
```

```
VIL=0.6
disp("VIL= "+string(VIL)+" V") //  Minimum input
   voltage level for which output is maximum
VIH=0.75
disp("VIH= "+string(VIH)+" V") //  Maximum input
   voltage level for which output is minimum
VOL=0.2
disp("VOL= "+string(VOL)+" V") //  Minimum output
   voltage level for  maximum input level
VOH=1
disp("VOH= "+string(VOH)+" V") //   Maximum output
   voltage level for   minimum input level
NML=VIL-VOL
disp("NML=VIL-VOL= "+string(NML)+" V") //  Low level
    noise immunities
NMH=VOH-VIH
disp("NMH=VOH-VIH= "+string(NMH)+" V") // High level
    noise immunities
```

Scilab code Exa 9.7 Find fan out of TTL

```
//Ex9_7
clc
IIL=-1.6*10^(-3)
disp("IIL= "+string(IIL)+" A") //  Input sink
   Current of TTL driver
IIH=40*10^(-6)
disp("IIH= "+string(IIH)+" A") //  source (supply)
   reverse Current of TTL driver
IOL=16*10^(-3)
disp("IOL= "+string(IOL)+" A") //  Specified Maximum
    sink Current of TTL driver
IOH=-400*10^(-6)
disp("IOH= "+string(IOH)+" A") //  Specified Maximum
    source Current of TTL driver
```

```
Fan_out=abs((IOH/IIH))
disp("Fan-out=abs((IOH/IIH)=abs((IOL/IIL))= "+string(Fan_out))// Fan-out of TTL
```

Scilab code Exa 9.8 Find low and high level noise immunities

```
//Ex9_8
clc
VIL=0.8
disp("VIL= "+string(VIL)+" V") //  Minimum input voltage level for which output is maximum
VIH=2
disp("VIH= "+string(VIH)+" V") //  Maximum input voltage level for which output is minimum
VOL=0.4
disp("VOL= "+string(VOL)+" V") //  Minimum output voltage level for  maximum input level
VOH=2.4
disp("VOH= "+string(VOH)+" V") //   Maximum output voltage level for   minimum input level
NML=VIL-VOL
disp("NML=VIL-VOL= "+string(NML)+" V") //  Low level noise immunities
NMH=VOH-VIH
disp("NMH=VOH-VIH= "+string(NMH)+" V") // High level noise immunities
```

Scilab code Exa 9.9 Find low and high level noise immunities

```
//Ex9_9
clc
VIL=1
```

```
disp("VIL= "+string(VIL)+" V") //  Minimum input voltage level for which output is maximum
VIH=4
disp("VIH= "+string(VIH)+" V") //  Maximum input voltage level for which output is minimum
VOL=0.5
disp("VOL= "+string(VOL)+" V") //  Minimum output voltage level for  maximum input level
VOH=4.5
disp("VOH= "+string(VOH)+" V") //   Maximum output voltage level for   minimum input level
NML=VIL-VOL
disp("NML=VIL-VOL= "+string(NML)+" V") //  Low level noise immunities
NMH=VOH-VIH
disp("NMH=VOH-VIH= "+string(NMH)+" V") // High level noise immunities
```

Scilab code Exa 9.10 Show ECL gate works properly as OR and NOR logic gates BJT do not go in saturation

```
//Ex9_10 Refer Fig.9.17(a)
clc
V_gamma=0.6
disp("V_gamma= "+string(V_gamma)+" volts") // Threshold voltage
VEE=-5.2
disp("VEE= "+string(VEE)+" volts") //  voltage supply
VBE3=0.7
VBE4=VBE3
VBE5=VBE3
disp("VBE3=VBE4=VBE5 "+string(VBE3)+" volts") //base-emitter voltage
RE=779
```

```
disp("RE= "+string(RE)+ " ohm")  // Emitter-resistance
RL2=220
disp("RL2= "+string(RL2)+ " ohm")  // Load resistance
RL3=245
disp("RL3= "+string(RL3)+ " ohm")  // Load resistance
VREF=-1.29
disp("VREF= "+string(VREF)+" volts") // Reference-voltage
V_1=-0.7
disp("V(1)= "+string(V_1)+" volts") // Acceptable voltage for high logic
V_0=-1.7
disp("V(0)= "+string(V_0)+" volts") // Acceptable voltage for low logic

disp("part(i)")// part(i)of question
VE=VREF-VBE3
disp("VE=VREF-VBE3= "+string(VE)+" volts") // Emitter- voltage
IE=(VEE-VE)/RE
disp("IE=(VEE-VE)/RE= "+string(IE)+" A") //Emitter-Current
IC3=IE// since IC=IE neglecting IB
disp("IC3=IE= "+string(IE)+" A") //Collector-Current
vC3=IC3*RL3
disp("vC3=IC3*RL3= "+string(vC3)+" volts") // Collector- voltage
vY=vC3-VBE5
disp("vY=vC3-VBE5= "+string(vY)+" volts") // Emitter follower output voltage for vB=V(0)
vC2=0
vX=vC2-VBE4
disp("vX=vC2-VBE4= "+string(vX)+" volts") // Emitter follower output voltage  for vB=V(0)
```

```
VBEr=(V_0)-VE
disp("Base -Emitter reverse voltage,VBEr=V(0)-VE= "+string(VBEr)+" volts")//Base- Emitter junction reverse voltage ,this is sufficient to keep T1 and T2 off since threshold =0.6V
disp("Transistor T1 and T2 off since VBEr < V_gamma") // Since VBEr < V_gamma hence T1 and T2 off

disp("part(ii)")// part(ii)of question
IC2=IE
VBE=0.7
vB=V_1
IC3=0
VC3=0
vY=VC3-VBE5
disp("vY=VC3-VBE= "+string(vY)+" volts")// Emitter follower output voltage  for SECOND CASE for vB=V(1)
VE=vB-VBE
disp("VE=vB-VBE= "+string(VE)+" volts") // Emitter-voltage
VBE3=VREF-VE
disp("VBE3=VREF-VE= "+string(VBE3)+" volts")//Base-Emitter junction voltage
disp("VBE3 is smaller than V_gamma,hence T3 is off")
IC2=(VEE-VE)/RE
disp("IC2=(VEE-VE)/RE= "+string(IC2)+" A") //Collector- Current for T2(transistor)
vC2=IC2*RL2
disp("vC2=IC2*RL2= "+string(vC2)+" volts") //Collector- voltage for T2
vX=vC2-VBE4
disp("vX=vC2-VBE4= "+string(vX)+" volts") // Emitter follower output voltage for vB=V(1)

disp("part(iii)")// part(iii)of question
VE3=-1.99
disp("VE3=VE= "+string(VE3)+" volts") //  Transistor
```

```
T3 Emitter- voltage ,when T3 is conducting
VB3=VREF
disp("VB3=VREF= "+string(VB3)+" volts") // Base-voltage when T3 is conducting
IC3=(VEE-VE3)/RE// Collector current for T3 neglecting IB
disp("IC3=(VEE-VE3)/RE= "+string(IC3)+" A") // Collector- Current
VC3=IC3*RL3
disp("VC3=IC3*RL3= "+string(VC3)+" volts") // Collector- voltage when T3 is conducting
VCB3=VC3-VB3
disp("VCB3=VC3-VB3= "+string(VCB3)+" volts") // Base - voltage when T3 is conducting
//All parameters have appropriate signs for npn BJT hence BJT in active region not in saturation in which VCB will have a (-)value
disp("All parameters have appropriate signs for npn BJT hence BJT in active region ")


// NOTE: Author ha not used any symbol for Base-Emitter junction reverse voltage  But I have used 'VBEr' for it.
// ERROR :sign of IE is given wrong in the book in part(i) and sign of IC2 in part(ii)
// In part(i) Correct Formulae of vC3 is vC3 =IC3*RL3 but given in book is vC3 =(-)IC3*RL3 because author has included the (-)ive sign or the polarity of IC3 in the formulae
// IN book in part(ii) mistakenly it  is written as vB=V_0 =-0.7 V but Correct expression is vB=V_1 =-0.7 V because vB is at high at V_1=-0.7 V
// In part(ii) Author has used formulae vC2=-IC2*RL2 because he has included the (-)ive sign of the IC2 in the formulae  but I have used vC2=IC2*RL2 to remove any ambiguity in program
```

Scilab code Exa 9.11 find low and high level noise immunities

```
//Ex9_11
clc
VIL=-1.475
disp("VIL= "+string(VIL)+" V") //  Minimum input
   voltage level for which output is maximum
VIH=-1.105
disp("VIH= "+string(VIH)+" V") //  Maximum input
   voltage level for which output is minimum
VOL=-1.63
disp("VOL= "+string(VOL)+" V") //  Minimum output
   voltage level for  maximum input level
VOH=-0.98
disp("VOH= "+string(VOH)+" V") //   Maximum output
   voltage level for   minimum input level
NML=VIL-VOL
disp("NML=VIL-VOL= "+string(NML)+" V") //  Low level
    noise immunities
NMH=VOH-VIH
disp("NMH=VOH-VIH= "+string(NMH)+" V") // High level
    noise immunities
```

Chapter 10

Combinational Logic Systems

Scilab code Exa 10.1 Convert decimal to binary

```
//Ex10_1
clc
x=72;//given value in Decimal
disp("Decimal number="+string(x))
str=dec2bin(x)
disp("Eqivalent Binary number="+string(str))//Binary
    value
```

Scilab code Exa 10.2 Convert binary to decimal

```
//Ex10_2
clc
x='1001000';//Binary value
disp("Binary number="+string(x))
str=bin2dec(x)
disp("Eqivalent Decimal number="+string(str))//
    decimal value
```

Scilab code Exa 10.3 Convert binary to octal

```
//Ex10_3
clc
b='1001000';
disp("Binary number="+string(b))//Binary value
d=bin2dec(b)// Binary to decimal value
o=dec2oct(d)// Decimal to octal
disp("Eqivalent Octal number="+string(o))
```

Scilab code Exa 10.4 Convert decimal to octal

```
//Ex10_4
clc
x=72;
disp("Decimal number="+string(x))//Decimal value
str=dec2oct(x)//decimal to octal
disp("Eqivalent Octal number="+string(str))
```

Scilab code Exa 10.5 Convert octal to binary

```
//Ex10_5
clc
x='110';
disp("Octal number="+string(x))//octal value
y=oct2dec(x)// octal to decimal
str=dec2bin(y)//decimal to binary
disp("Eqivalent Binary number="+string(str))
```

Scilab code Exa 10.6 Convert octal to decimal

```
//Ex10_6
clc
x='110';
disp("Octal number="+string(x))// octal value
str=oct2dec(x)//octal to decimal
disp("Eqivalent Decimal number="+string(str))
```

Scilab code Exa 10.7 Convert decimal to hexadecimal

```
//Ex10_7
clc
x=72;
disp("Decimal number="+string(x))//decimal value
str=dec2hex(x)// decimal to hexadecimal
disp("Eqivalent Hexadecimal number="+string(str))
```

Scilab code Exa 10.8 Convert hexadecimal to decimal

```
//Ex10_8
clc
h='48';
disp("Hexadecimal number="+string(h))// value in
    hexadecimal
d=hex2dec(h)//hexadecimal to decimal
disp("Eqivalent Decimal number="+string(d))
```

Scilab code Exa 10.9 Convert hexadecimal to binary

```
//Ex10_9
clc
h='48';
disp("Hexadecimal number="+string(h))//hexadecimal
d=hex2dec(h)// converting hexadecimal to decimal
str=dec2bin(d)// converting decimal to binary
disp("Eqivalent Binary number="+string(str))
```

Scilab code Exa 10.10 convert binary to hexadecimal

```
//Ex10_10
clc
x='1001000';
disp("Binary number="+string(x))//binary value
d=bin2dec(x)//binary to decimal
h=dec2hex(d)//decimal to hexa decimal
disp("Eqivalent hexadecimal number="+string(h))
```

Scilab code Exa 10.11 Convert decimal to BCD

```
//Ex10_11
//for two digit decimal value to convert into BCD
clc
x=72;
n=2// length of decimal input
f=10^(n-1)
z =int( x/f)             // taking MSB of input
   string
y=x-(z*10^(n-1))   // taking LSB of input string
disp("Decimal number="+string(x))
```

```
str1=dec2bin(z,4)// for 4 bit binary of MSB digit i.
    e 7 here
str2=dec2bin(y,4)//for 4 bit binary of LSB digit i.e
     2
disp(" Binary number of 7 ="+string(str1))
disp(" Binary number of 2 ="+string(str2))
disp("Eqivalent BCD of 72="+string(str1)+string(str2
    ))// Binary coded decimal for  72
```

Chapter 11

Sequential Logic Systems

Scilab code Exa 11.1 Find maximum operating clock frequency for flip flop

```
//Ex11_1
clc
tsu=20*10^(-9)
disp("tsu= "+string(tsu)+" seconds") // Input set-up time of second flip flop
tpd=30*10^(-9)
disp("tpd= "+string(tpd)+" seconds") // Input set-up time of first flip flop
Tmin=tpd+tsu
disp("Tmin=tpd+tsu= "+string(Tmin)+" seconds") // Minimum allowed time interval b/w threshold levels of two consecutive triggering clock edges activating two flip-flops
fCkmax=1/Tmin //  formulae
disp("fCkmax=1/Tmin = "+string(fCkmax)+" Hz")// Maximum clock frequency at which flip-flop can operate reliably
```

Scilab code Exa 11.4 Find maximum counting rate for 3 bit ripple count

```
//Ex11_4
clc
tphL=40*10^(-9)
disp("tphL= "+string(tphL)+" seconds") // Time taken
    from Clear to output
n=3
disp("n= "+string(n))// Number of bits in counter i.
    e no. of flip-flops used
fmax=1/(n*tphL) // Using formulae fmax<= 1/(n*tphL)
disp("fmax=1/(n*tphL) = "+string(fmax)+" Hz")//
    Maximum counting rate at which flip-flop can
    operate reliably
```

Scilab code Exa 11.6 Find display count for decade counter

```
//Ex11_6
clc
fs=2*10^(3)
disp(" fs= "+string(fs)+" Hz")// sine wave input
    signal frequency
fB=1*10^(6)
disp(" fB= "+string(fB)+" Hz")// input Time-Base
    clock frequency

disp("part(i)")// part(i)of question
fb=fB/(10^5)
disp(" fb= fB/(10^5)="+string(fb)+" Hz")//  Time-Base
    frequency for 5 decade counter
delta_t=1/fb
disp(" delta_t=1/fb= "+string(delta_t)+" seconds") //
    Gate Time interval
DISP1=fs*delta_t
disp(" fs*delta_t= "+string(DISP1))// Display
```

```
    indication for 5 decade counter
disp("Display indication=0200")// Display indication
     as 4-bit

disp("part(ii)")// part(ii)of question
fb=fB/(10^3)
disp("fb=fB/(10^3)= "+string(fb)+" Hz")//  Time-Base
     frequency for 3 decade counter
delta_t=1/fb
disp("delta_t=1/fb= "+string(delta_t)+" seconds") //
     Gate Time interval for 3 decade counter
DISP2=fs*delta_t
disp("fs*delta_t= "+string(DISP2))// Display
    indication for 3 decade counter
disp("Display indication=0002")// Display indication
     as 4-bit
```

Chapter 12

Waveshaping and Waveform Generation

Scilab code Exa 12.1 Find parameters for op amp based schmitt trigger

```
//Ex12_1
clc
VEE=15
disp("VEE= "+string(VEE)+" volts") //  voltage
   supply
VCC=15
disp("VCC= "+string(VCC)+" volts") //  voltage
   supply
VHI=+5
disp("VHI= "+string(VHI)+" volts") //  output
   voltage upper limit
VLO=-5
disp("VLO= "+string(VLO)+" volts") //  output
   voltage Lower limit
Vo=-VLO
IZmin=1*10^(-3)
disp("IZmin= "+string(IZmin)+" A") // Zener diode
   current rating
SR=0.5*10^(6)
```

```
disp("SR= "+string(SR)+" volts/seconds")//Slew rate
RB=100
disp("RB= "+string(RB)+ " ohm")   // resistance
RA=10*10^(3)
disp("RA= "+string(RA)+ " ohm")   // resistance
A = 5000
disp("A = "+string(A))//op-amp gain
VREF=1
disp("VREF= "+string(VREF)+" volts") // Reference-voltage
disp("part(i)")
RD=(VCC-Vo)/IZmin
disp("RD=(VCC-Vo)/IZmin= "+string(RD)+ " ohm")   // Series dropping-resistance

disp("part(ii)")
t=(VHI-VLO)/SR
disp("t=(VHI-VLO)/SR= "+string(t)+" seconds")// Time required to swing the output
tp=10*t
disp("tp=(VHI-VLO)/SR= "+string(tp)+" seconds")// Pulse width
fmax=1/(2*tp)
disp("fmax=1/(2*tp) = "+string(fmax)+" Hz")// Maximum frequency of operation of OP-AMP comparator
disp("part(iii)")
B=RB/(RA+RB)
disp("B=RB/(RA+RB)= "+string(B))//Feedback factor
VLTP=(VLO*B)+[VREF*(RA/(RA+RB))]
disp("VLTP=(VLO*B)+[VREF*(RA/(RA+RB))]= "+string(VLTP)+" volts") //   Lower trigger point
VUTP=(VHI*B)+[VREF*(RA/(RA+RB))]
disp("VUTP=(VHI*B)+[VREF*(RA/(RA+RB))]= "+string(VUTP)+" volts") // Upper trigger point
VH=VUTP-VLTP
disp("VH=VUTP-VLTP= "+string(VH)+" volts") // Hysteresis voltage
```

Scilab code Exa 12.2 Calculate component values for astable multivibrator

```
//Ex12_2
clc
Vo=14
disp("Vo= "+string(Vo)+" volts") //  output voltage
f=500
disp("f = "+string(f)+" Hz")//frequency
IB2=500*10^(-9)
disp("IB2= "+string(IB2)+" A") //base- Current
B=0.5
disp("B="+string(B))//Feedback factor
vf=B*Vo
disp("vf=B*Vo= +"+string(vf)+", -"+string(vf)+" volts") // Feedback voltage
IR=100*IB2// Taking IR as 100 times that of IB2
disp("IR=100*IB2= "+string(IR)+" A") // Current in RB resistor
RB=vf/IR
disp("RB=vf/IR= "+string(RB)+ " ohm")  // resistance
RA=RB*((1/B)-1)// Using formulae B=RA/(RA+RB)
disp("RA=RB*((1/B)-1)= "+string(RA)+ " ohm")  // resistance
RF=100*10^(3)//Choosing RF=100k
disp("RF= "+string(RF)+ " ohm")  //Feedback resistance
C1=1/[2*RF*f*log(1+(2*RB/RA))]
disp("C1=1/[2*RF*f*log(1+(2*RB/RA))]= "+string(C1)+" farad") // calculated capacitance value
```

Scilab code Exa 12.3 Find capacitor C2 for triangular wave generator

```
//Ex12_3
clc
Vo=14
disp("Vo= "+string(Vo)+" volts") //  output voltage
f=500
disp("f = "+string(f)+" Hz")//frequency
R2=10*10^(3)
disp("R2= "+string(R2)+ " ohm")  // resistance
VTW=14
disp("VTW= "+string(VTW)+" volts") // Triangular
   peak-peak output voltage
T=1/f
C2=(Vo*T)/(2*VTW*R2)
disp("C2=(Vo*T)/(2*VTW*R2)= "+string(C2)+" farad")
   // calculated capacitance value for deriving
   triangular wave from square wave astable
   multivibrator
```

Scilab code Exa 12.4 Find sweep rate and amplitude Vsw for sweep generator

```
//Ex12_4
clc
VI=-15
disp("VI= "+string(VI)+" volts") //  Input voltage
TSW=2*10^(-3)
disp("TSW= "+string(TSW)+" seconds")// triangular
   wave Sweep time
R=10*10^(3)
disp("R= "+string(R)+ " ohm")  // resistance as ckt.
    parameter
C=0.5*10^(-6)
disp("C= "+string(C)+" farad") //  capacitance as
   ckt. parameter
S=-VI/(R*C)
```

```
disp("Sweep rate=VI/(R*C)="+string(S)+ " V/s") // Sweep rate for sweep generator
VSW=TSW*S
disp("VSW=TSW*S= "+string(VSW)+" volts") //  Sweep voltage amplitude


// note in book author has not provided any variable for sweep rate ... but here I have used 'S' for it .
```

Scilab code Exa 12.5 Find parameters for feedback op amp soft limiter circuit

```
//Ex12_5
clc
VEE=15
disp("VEE= "+string(VEE)+" volts") //  voltage supply
VCC=15
disp("VCC= "+string(VCC)+" volts") //  voltage supply
R1=10*10^(3)
disp("R1= "+string(R1)+ " ohm")  // resistance
RF=20*10^(3)
disp("RF= "+string(RF)+ " ohm")  // Feedback resistance
RB1=3*10^(3)
disp("R1= "+string(R1)+ " ohm")  // resistance
RB2=RB1
RF1=1*10^(3)
disp("RF1= "+string(RF1)+ " ohm")  // Feedback resistance
RF2=RF1
Av=1*10^(3)
```

```
disp("Av= "+string(Av))
disp("part(i)")
VBR1= (VCC*RF1)/RB1
VBR2 = VBR1
disp("VBR1=VBR2=(VCC*RF1)/RB1= "+string(VBR1)+" volts") //Limit values at the break points and VBR=VBR1=VBR2
So=-RF/R1
disp("So=-RF/R1= "+string(So))// slope of Transfer characteristic at zero crossings
S1=-(RF1/R1)
disp("S1=S2=-RF1/R1= "+string(S1))// slope of Transfer characteristic at the extreme ends
VSL=(-VBR1/So)
disp("VSL=VSU=(-VBR1/So)= "+string(VSL)+" volts") // magnitude of input voltage required to produce vo=VBR
VSU=VSL
disp("part(ii)")
VSU=(VBR2/Av)//Formulae
disp("VSU=VSL=(VBR2/Av)= -"+string(VSU)+", +"+string(VSU)+" volts") //  magnitude of input voltage required to produce vo=VBR in case gain Av is very large
```

Chapter 13

Non Linear Analog Systems

Scilab code Exa 13.1 output voltage for log amplifier

```
//Ex13_1
clc
VT=26*10^(-3)
disp("VT= "+string(VT)+" volts") // Thermal voltage
R1=5*10^(3)
disp("R1= "+string(R1)+ " ohm")  // resistance
Iso=1*10^(-10)
disp(" Iso = "+string(Iso)+" ampere") // Scale
    factor (as current)directly proportional to cross
    -section area of EBJ

disp("part(i)")
vs=1*10^(-3)
disp("vs= "+string(vs)+" volts") // Input voltage1
vo=-VT*[log(vs/(Iso*R1))]
disp("vo=-VT*[log(vs/(Iso*R1))]= "+string(vo)+"
    volts") // Output voltage of Log OP-AMP for
    input1 i.e vs = 1 mV

disp("part(ii)")
vs=10*10^(-3)
```

```
disp(" vs= "+string(vs)+" volts") // Input voltage2
vo=-VT*[log(vs/(Iso*R1))]
disp(" vo=-VT*[log(vs/(Iso*R1))]= "+string(vo)+"
    volts") // Output voltage of Log OP-AMP for
    input1 i.e vs = 10 mV

disp("part(iii)")
vs=100*10^(-3)
disp(" vs= "+string(vs)+" volts") // Input voltage3
vo=-VT*[log(vs/(Iso*R1))]
disp(" vo=-VT*[log(vs/(Iso*R1))]= "+string(vo)+"
    volts") // Output voltage of Log OP-AMP for
    input1 i.e vs = 100 mV

disp("part(iv)")
vs=1
disp(" vs= "+string(vs)+" volts") // Input voltage4
vo=-VT*[log(vs/(Iso*R1))]
disp(" vo=-VT*[log(vs/(Iso*R1))]= "+string(vo)+"
    volts") // Output voltage of Log OP-AMP for
    input1 i.e vs = 1V
```

Scilab code Exa 13.2 output voltage for log amplifier

```
//Ex13_2
clc
VT=26*10^(-3)
disp("VT= "+string(VT)+" volts") // Thermal voltage
R1=100*10^(3)
disp("R1= "+string(R1)+ " ohm") // resistance
Iso=50*10^(-9)
disp(" Iso = "+string(Iso)+" ampere") // Scale
    factor (as current)directly proportional to cross
    -section area of EBJ
vs=2.5
```

```
disp("vs= "+string(vs)+" volts") // Input voltage
vo=-VT*[log(vs/(Iso*R1))]
disp("vo=-VT*[log(vs/(Iso*R1))]= "+string(vo)+" volts") // Output voltage of Log OP-AMP for input1 i.e vs = 2.5 V
```

Scilab code Exa 13.3 output for antilog amplifier

```
//Ex13_3
clc
VT=26*10^(-3)
disp("VT= "+string(VT)+" volts") // Thermal voltage
RF=100*10^(3)
disp("RF= "+string(RF)+ " ohm")  // resistance
Iso=50*10^(-9)
disp(" Iso = "+string(Iso)+" ampere") // Scale factor (as current)directly proportional to cross-section area of EBJ
vs=-0.162
disp("vs= "+string(vs)+" volts") // Input voltage
vo=Iso*RF*(exp(-vs/VT))
disp("vo=Iso*RF*(exp(-vs/VT))= "+string(vo)+" volts") // Output voltage of Antilog OP-AMP for input1 i.e vs = -0.162 V
```

Chapter 14

Digital Analog Systems

Scilab code Exa 14.1 Find quantization error and resolution

```
//Ex14_1 Refer fig 14.1(b) and (c)
clc
n=3
disp("n= "+string(n))// Number of bits
L=2^(n)
disp("L=2^(n)= "+string(L))// Number of quantization levels
VFS=10
disp("VFS= "+string(VFS)+" volts") // Maximum value of analog input voltage
Q.E=VFS/L
disp("Q.E=VFS/L= "+string(Q.E))// Quantization error
disp("Q.E= +0.625,-0.625")// To make Quantization error symmetrical ittaken as (-Q.E/2) negative and positive value(+Q.E/2)
Resolution=(100/2^(n))//Formulae
disp("Resolution=(100/2^(n))= "+string(Resolution)+" percent")//Resolution
disp("Resolution= "+string(+Resolution)+" percent,"+string(-Resolution)+" percent")// Since Resolution is (+)as well as (-)
```

Scilab code Exa 14.2 Find parameters for 3 bit ADC

```
//Ex14_2
clc
n=3
disp("n= "+string(n))// Number of bits
L=2^(n)
disp("L=2^(n)= "+string(L))// Number of quantization levels
VFS=1024*10^(-3)
disp("VFS= "+string(VFS)+" volts") // Maximum value of analog input voltage

disp("part(i)")// Part(i)
LSB=VFS/(2^n)
disp("LSB=VFS/(2^n)= "+string(LSB)+" volts") // Lowest significant bit of 3-bit ADC

disp("part(ii)")// Part(ii)
disp("vh= 64 to 192 mV with offset") // Analog voltage corresponding to binary word 001

disp("part(iii)")// Part(iii)
I.E=(LSB)/2
disp("Inherent error,I.E= (LSB)/2= -"+string(I.E)+" V,+"+string(I.E)+" V")// Inherent error in each binary word

disp("part(iv)")// Part(iv)
Resolution=(1*10^(-3))
disp("Resolution= "+string(Resolution)+" V")// Resolution
VFS=1
disp("VFS= "+string(VFS)+" V") // Maximum value of
```

```
analog input voltage2
k=VFS/(Resolution)
disp("k=VFS/(Resolution)= "+string(k)) // 'k' taken
    only for calculation purpose
disp("number of bits=10")// since k =[VFS/(
    Resolution)]is approximately equal to 2^10,
disp("so 10-bit ADC required")
```

Scilab code Exa 14.3 Find output voltage for weighted resistor DAC

```
//Ex14_3
clc
VREF=-10
disp("VREF= "+string(VREF)+" V") // Reference
    voltage
RF=5*10^(3)
disp("RF= "+string(RF)+ " ohm")  //Feedback
    resistance
R=10*10^(3)
disp("R= "+string(R)+ " ohm")  // resistance
vLSB=(-RF*VREF)/(8*R)// Since IF=I/8,so vLSB=(-RF*IF
    )=(-RF*I/8)=(-RF*VREF/8*R)
disp("vLSB=(-RF*VREF)/(8*R)="+string(vLSB)+" V") //
    Equivalent voltage for binary word 0001
vo=-2*vLSB// Since current IF=I/4
disp("vo = -2*vLSB ="+string(vo)+" V") // Equivalent
     voltage for binary word 0010=2 (in decimal)
vo=-15*vLSB// Since current IF=I+(I/2)+(I/4)+(I/8)
    =(15*I/8),so vo=15*VLSB
disp("vo= -15*vLSB ="+string(vo)+" V") // Equivalent
     voltage for binary word 0010=2 (in decimal)
```

Scilab code Exa 14.4 Output for resistor ladder DAC

```
//Ex14_4
clc
VREF=-10
disp("VREF= "+string(VREF)+" V") // Reference
    voltage
RF=5*10^(3)
disp("RF= "+string(RF)+ " ohm")  //Feedback
    resistance
R=10*10^(3)
disp("R= "+string(R)+ " ohm")  // resistance
vMSB=-(RF*VREF)/(2*R)// Since IF=I/2,so vMSB=(-RF*IF
    )=(-RF*I/2)=(-RF*VREF/2*R)
disp("vMSB=-(RF*VREF)/(2*R)="+string(vMSB)+" V") //
    Equivalent voltage for binary word 1000=8(in
    decimal)
vo2=vMSB/2// Since current IF=I/4
disp("vo2 = vMSB/2 ="+string(vo2)+" V") //
    Equivalent voltage for binary word 0100=4 (in
    decimal)
vo3=(15/8)*vMSB// Since current IF=I+(I/2)+(I/4)+(I
    /8)+(I/16)=(15*I/6),so vo=(15/8)*VMSB
disp("vo3= (15/8)*vMSB ="+string(vo3)+" V") //
    Equivalent voltage for binary word 1111=15 (in
    decimal)
```

Scilab code Exa 14.5 Quantization error and percentage resolution for 12 bit DAC

```
//Ex14_5
clc
n=12
disp("n= "+string(n))// Number of bits
VFS=50
disp("VFS= "+string(VFS)+" volts") // Maximum value
    of analog input voltage
```

```
S=VFS/(2^n)
disp("S=VFS/(2^n)= "+string(S)+" volts") // Maximum
    quantization error
Resolution=(100/2^(n))//Formulae
disp("Resolution=(100/2^(n))= -"+string(Resolution)+
    " percent , +"+string(Resolution)+" percent")//
    Since Resolution is (+)as well as (-)
```

Scilab code Exa 14.7 Find highest analog input frequency for 12 bit ADC

```
//Ex14_7
clc
n=12
disp("n= "+string(n))// Number of bits
t=5*10^(-6)
disp("t= "+string(t)+" A")
Vsp=10
disp("Vsp= "+string(Vsp)+" volts") // value of
    analog input voltage
LSB=Vsp/(2^n)
disp("LSB=Vsp/(2^n)= "+string(LSB)+" volts") //
    Lowest significant bit of 12-bit ADC
disp("LSB/2= -"+string(LSB/2)+" V, -"+string(LSB/2)+
    " V")
SR=(LSB/2)/t
disp("SR=(LSB/2)/t= "+string(SR)+" V/s")
fmax=SR/(2*%pi*Vsp)
disp("f = SR/(2*%pi*Vsp)="+string(fmax)+" Hz")//
    Highest frequency allowed at the input
```