

Scilab Textbook Companion for
Electronics Circuits and Systems
by Y. N. Bapat¹

Created by

Komal

B.Tech

Electronics Engineering

UPTU/ABES Engg. College Gzb

College Teacher

None

Cross-Checked by

Chaya Ravindra

June 2, 2016

¹Funded 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	4
1 Vacuum tubes and semiconductors	5
2 The Semiconductor diode	15
3 Bipolar Junction Transistor	24
4 Field Effect Transistors	37
5 Basic Transistor Amplifiers	56
6 Multitransistor Multistage Amplifiers	67
7 Feedback Amplifiers and Sinusoidal oscillators	76
8 Linear Op Amp Applications	87
9 Digital Circuits and Logic Families	101
10 Combinational Logic Systems	113
11 Sequential Logic Systems	118
12 Waveshaping and Waveform Generation	121
13 Non Linear Analog Systems	127
14 Digital Analog Systems	130

List of Scilab Codes

Exa 1.1	small signal parameters	5
Exa 1.2	CRT parameters	6
Exa 1.3	Find deflection of electron beam	8
Exa 1.4	calculate drift velocity	9
Exa 1.5	calculation of resistance	10
Exa 1.6	parameters for Silicon atoms	10
Exa 1.7	N type Silicon parameters	11
Exa 1.8	diffusion hole current	13
Exa 2.1	calculate diode current	15
Exa 2.2	Diode current and diode junction voltage calculation	16
Exa 2.3	small signal forward and reverse resistance	17
Exa 2.4	various parameters for two diode rectifier	18
Exa 2.5	calculation of resistance	19
Exa 2.6	full wave rectifier parameters	19
Exa 2.7	Zener diode parameters	20
Exa 2.8	reference voltage and temperature coefficient	21
Exa 2.9	load voltage and current	22
Exa 3.1	BJT current and voltages	24
Exa 3.2	find current gain and voltage gain	25
Exa 3.3	Find transconductance gm and hybrid parameter hie and unity gain frequency	26
Exa 3.4	operating point of BJT	27
Exa 3.5	Find operating point of BJT	28
Exa 3.6	Find operating point of BJT	29
Exa 3.7	Calculate extreme variation in operating collector current ICQ	30
Exa 3.8	find value of collector current Ic and check is BJT in active region	32

Exa 3.9	find forced value of current gain for BJT in saturation region	33
Exa 3.10	Find maximum allowed power dissipation at 25 and 75 degree celsius	34
Exa 3.11	Find maximum allowed power dissipation at 25 and 75 degree celsius	35
Exa 3.12	Calculate resistance R for BJT current mirror	35
Exa 4.1	Find drain to source voltage VDS and drain current ID for JFET	37
Exa 4.2	Find drain to source resistance RDS for JFET	38
Exa 4.3	Find load resistance RL for EMOSFET	38
Exa 4.4	calculate transconductance gm for FETs at various device parameters	39
Exa 4.5	Design potential divider biasing for JFET	41
Exa 4.6	calculate feedback resistance RF for n channel JFET	42
Exa 4.7	Find drain current ID and verify operating region is pinch off	43
Exa 4.8	Find operating values of drain current and drain to source voltage and gate to source voltage	44
Exa 4.9	Find operating values of drain current ID and drain to source voltage VDS and gate to source voltage VGS for FET	46
Exa 4.10	Find operating values of drain current IDQ and drain to source voltage VDSQ and gate to source voltage VGSQ for EMOSFET	47
Exa 4.11	Calculate load resistance RL and percentage change in drain current ID for JFET	49
Exa 4.12	Find ON stage output voltage VDON for EMOSFET	51
Exa 4.13	Verify DEMOSFET act as DC constant current source	52
Exa 4.14	Find resistance value to obtain constant current source for MOS current mirror	53
Exa 4.15	Determine ON voltage and off isolation errors for RL and analog shunt switch	54
Exa 5.1	Find voltage gain and input resistance and output resistance and current gain for CE and CS amplifiers	56
Exa 5.2	Find voltage gain and overall voltage gain and current gain and input resistance and output resistance for CC amplifier	57

Exa 5.3	Find input resistance and voltage gain and current gain and output resistance for CE amplifier	58
Exa 5.4	Find overall voltage gain A_{vs} and input resistance R_i and output resistance R_o for CB and CG amplifier . . .	59
Exa 5.5	Find high cut off frequency and bandwidth and unity gain bandwidth	61
Exa 5.6	Find percentage second harmonic distortion in small signal CS JFET amplifier	62
Exa 5.7	Compare performance of CE and CC and CB configuration of BJT	63
Exa 5.8	Find load resistance R_L and coupling capacitor	66
Exa 6.1	Express voltage gains in decibels	67
Exa 6.2	Find voltage gain A_v and current gain A_i and power gain A_p for amplifier	68
Exa 6.3	Find common mode and differential gain and CMRR and input resistances and output voltage and error . . .	69
Exa 6.4	Find load resistance and collector to emitter voltage and output power and DC power and dissipated power . . .	71
Exa 6.5	calculate output power P_o and DC power P_{DC} and efficiency at different conditions	73
Exa 6.6	Find maximum difference voltage V_{dmax} for op amp without feedback	75
Exa 7.1	Calculate amount of negative feedback and feedback factor	76
Exa 7.2	Find gain with feedback	76
Exa 7.3	calculate amount of feedback and gain A_f and new input voltage	77
Exa 7.4	find signal bandwidth and frequency below which midband gain will not deviate	78
Exa 7.5	find midband gain and higher cut off frequency f_{Hf} and lower cut off frequency f_{Lf}	78
Exa 7.6	calculate phase margin for different feedback gains . . .	79
Exa 7.7	Find maximum allowed input voltage without causing output clipping	81
Exa 7.8	find output voltage and error for gain and value of gain to reduce output error	82

Exa 7.9	find feedback resistances R_F and R_1 and resulting output resistance and voltage gain and power gain with feedback	84
Exa 7.10	Find different parameters for Pierce oscillator	85
Exa 8.1	calculate feedback factor and variation in feedback gain A_f	87
Exa 8.2	Design an inverting op amp	88
Exa 8.3	Find resistances R_{11} and R_{12} and R_{13} for op amp adder and output voltage	89
Exa 8.4	Find scale factor of photometer	90
Exa 8.5	Find effective resistance and sensitivity and feedback resistance R_F of DC ammeter	90
Exa 8.6	Find resistances R_1 and R_F for non inverting op amp	91
Exa 8.7	Find resistances R_1 and input feedback resistance R_{if} for full scale moving coil meter	92
Exa 8.8	Find worst case change in output voltage V_o	93
Exa 8.9	Design 2nd order LP Butterworth filter	93
Exa 8.10	Design 2nd order single op amp band pass filter	94
Exa 8.11	Determine filter response for state variable filter	95
Exa 8.12	Find output offset voltage with and without comensation	96
Exa 8.13	find resistance R_2 to minimize offset and output offset voltage at 25 and 75 degree celsius	97
Exa 8.14	Find power supply voltage regulation	98
Exa 8.15	calculate full power bandwidth	99
Exa 9.1	Find minimum gain for proper operation of NOT gate	101
Exa 9.2	calculate output voltage V_x for OR logic gate	102
Exa 9.3	Calculate output voltage V_x for diode AND logic gate	104
Exa 9.6	Find low and high level noise immunities	105
Exa 9.7	Find fan out of TTL	106
Exa 9.8	Find low and high level noise immunities	107
Exa 9.9	Find low and high level noise immunities	107
Exa 9.10	Show ECL gate works properly as OR and NOR logic gates BJT do not go in saturation	108
Exa 9.11	find low and high level noise immunities	112
Exa 10.1	Convert decimal to binary	113
Exa 10.2	Convert binary to decimal	113
Exa 10.3	Convert binary to octal	114
Exa 10.4	Convert decimal to octal	114

Exa 10.5	Convert octal to binary	114
Exa 10.6	Convert octal to decimal	115
Exa 10.7	Convert decimal to hexadecimal	115
Exa 10.8	Convert hexadecimal to decimal	115
Exa 10.9	Convert hexadecimal to binary	116
Exa 10.10	convert binary to hexadecimal	116
Exa 10.11	Convert decimal to BCD	116
Exa 11.1	Find maximum operating clock frequency for flip flop .	118
Exa 11.4	Find maximum counting rate for 3 bit ripple count . .	119
Exa 11.6	Find display count for decade counter	119
Exa 12.1	Find parameters for op amp based schmitt trigger . .	121
Exa 12.2	Calculate component values for astable multivibrator .	123
Exa 12.3	Find capacitor C2 for triangular wave generator	123
Exa 12.4	Find sweep rate and amplitude Vsw for sweep generator	124
Exa 12.5	Find parameters for feedback op amp soft limiter circuit	125
Exa 13.1	output voltage for log amplifier	127
Exa 13.2	output voltage for log amplifier	128
Exa 13.3	output for antilog amplifier	129
Exa 14.1	Find quantization error and resolution	130
Exa 14.2	Find parameters for 3 bit ADC	131
Exa 14.3	Find output voltage for weighted resistor DAC	132
Exa 14.4	Output for resistor ladder DAC	132
Exa 14.5	Quantization error and percentage resolution for 12 bit DAC	133
Exa 14.7	Find highest analog input frequency for 12 bit ADC .	134

Chapter 1

Vacuum tubes and semiconductors

Scilab code Exa 1.1 small signal parameters

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

```

    rP)+" ohm") //calculation
14 VGK2=-2.5
15 disp("VGK2 = "+string(VGK2)+" volts") // value of
    grid voltage2
16 VGK3=-1.5
17 disp("VGK1 = "+string(VGK3)+" volts") // value of
    grid voltage1
18 VAK3=200
19 disp("VAK3 = "+string(VAK3)+" volts") // value of
    anode voltage1
20 u=(VAK2-VAK3)/(VGK2-VGK3)//amplification factor at
    IA=IAQ
21 disp(" amplification factor ,u =(VAK2-VAK1)/(VGK2-VGK1
    )=" +string(u)+" unitless ") //calculation
22 IA4=2.2*10^(-3)
23 disp("IA4 = "+string(IA4)+" ampere") // value of
    anode current4
24 IA3=0.5*10^(-3)
25 disp("IA1 = "+string(IA3)+" ampere") // value of
    anode current1
26 gm=(IA4-IA3)/(VGK2-VGK3)// transconductance at VAK=
    VAKQ
27 disp(" transconductance ,gm =(IAK4-IAK3)/(VGK2-VGK3)="
    +string(gm)+" ampere/volt ") //calculation
28 //mistake of negative sign for answers for u(
    amplification factor) and gm(transconductance)in
    book

```

Scilab code Exa 1.2 CRT parameters

```

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

```

```

5 l=2*10(-2)
6 disp("l = "+string(l)+" metre") //initializing value
  of length of plates
7 L=20*10(-2)
8 disp("L = "+string(L)+" metre") //initializing value
  of distance b/w centre of plates and screen
9 Va=2000
10 disp("Va = "+string(Va)+" volts") ////initializing
  value of anode voltage
11 Vd=100
12 disp("Vd = "+string(Vd)+" volts") //initializing
  value of deflecting voltage
13 m=9.11*10(-31)
14 disp("m = "+string(m)+" Kg") //mass of electron
15 q=1.6*10(-19)
16 disp("q = "+string(q)+" coulomb") //charge on an
  electron
17 disp("horizontal beam velocity ,Vx =(2*Va*q/m)^(0.5)
  metre/second") //formula
18 Vx =(2*Va*q/m)^(0.5)
19 disp("horizontal beam velocity ,Vx =(2*Va*q/m)^(0.5)=
  "+string(Vx)+" metre/second") //calculation
20 disp("transit time ,t1 =(l/Vx) second") //formula
21 t1=(l/Vx)
22 disp("transit time ,t1 =(l/Vx)= "+string(t1)+" second
  ")//calculation
23 disp("vertical beam velocity ,Vy =(q*Vd*l/d*m*Vx)
  metre/second") //formula
24 Vy=((q*Vd*l)/(d*m*Vx))
25 disp("vertical beam velocity ,Vy =(q*Vd*l/d*m*Vx)= "+
  string(Vy)+" metre/second") //calculation
26 disp("vertical displacement ,D =((l*L*Vd)/(2*d*Va)
  metre")//formula
27 D =(l*L*Vd)/(2*d*Va)
28 disp("vertical displacement ,D =((l*L*Vd)/(2*d*Va)= "+
  string(D)+" metre") //calculation
29 disp("sensitivity of CRT,S =(0.5*l*L)/(d*Va) metre/
  volt")//formula

```

```

30 S =(0.5*1*L)/(d*Va)
31 disp("sensitivity of CRT,S =(0.5*1*L)/(d*Va)="+
      string(S)+" metre/volt")//calculation

```

Scilab code Exa 1.3 Find deflection of electron beam

```

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

```

```

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

```

Scilab code Exa 1.4 calculate drift velocity

```

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

```

Scilab code Exa 1.5 calculation of resistance

```
1 //Ex1_5
2 clc
3 A=10*10^(-6); p1=10^(-4); p2=10^(3); p3=10^(10); l
   =1*10^(-2); //initializations
4 disp("cross sectional area ,A =" +string(A)+" merer
   square")
5 disp("resitivity (rho) ,p1 =" +string(p1)+" ohm-m")
6 disp("resitivity (rho) ,p2 =" +string(p2)+" ohm-m")
7 disp("resitivity (rho) ,p3 =" +string(p3)+" ohm-m")
8 disp("conductor length ,l =" +string(l)+" metre")
9 disp(" resistance for copper ,R = p1*l/A = " +string(
   p1*l/A)+"ohm") //calculations for copper
10 disp(" resistance for silicon ,R = p2*l/A = " +string(
   p2*l/A)+"ohm") //calculations for silicon
11 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

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

```

13 disp(" silicon atoms concentration , nV = "+string(nV)+
    " /centimetre cube ")
14
15 disp(" electron mobility ,un = "+string(un)+" cm.sq/V-
    s")
16 disp(" hole mobility ,up = "+string(up)+"cm.sq/V-s")
17 disp(" temperature ,T = "+string(T)+"K")
18 disp(" q = "+string(q)+" coulomb") //charge on an
    electron
19 disp(" cross sectional area ,A =" +string(A)+"cm square
    ")
20 disp(" conductor length ,l =" +string(l)+"cm")
21 N=nV/ni
22 disp(" relative concentration ,N =nV/ni= "+string(N)+"
    silicon atoms per electron -hole pair") //
    calculation
23 sigma=(1.59*10^(-19))*(1.45*10^10)*(1500+0475))
24 disp(" intrinsic conductivityi ,sigma =(1.59*10^(-19)
    *(1.45*10^10)*(1500+0475))= "+string(sigma)+" (
    ohm-cm)^-1") // calculation
25 pi =(1/sigma)//formulae
26 disp(" resitivity (rho) ,pi =(1/sigma)=" +string(pi)+"
    ohm-cm")// calculation
27 R=(2.22*10^5*0.5)/0.000025
28 disp(" resistance for silicon ,R =((2.22*10^5*0.5)
    /0.000025) = "+string(R)+" ohm") //calculations
    for silicon
29 V=I*R
30 disp(" voltage drop ,V =I*R = "+string(V)+" V") //
    calculations

```

Scilab code Exa 1.7 N type Silicon parameters

```

1 //Ex1_7
2 clc

```



```

3 ni = 1.45*10^10 //initializations
4 nV = 5*10^22 //initializations
5 un = 1500 //initializations
6 up = 0475//initializations
7 T = 300 //initializations
8 I=10^(-6)
9 disp(" I = "+string(I)+" Ampere") //initializing value
    of current
10 A=(50*10^(-4))^2; l=0.5 //initializations
11 q=1.59*10^(-19) //charge on an electron
12 disp("intrinsic charge concentration ,ni = "+string(
    ni)+" /centimetre cube")
13 disp("silicon atoms concentration , nV = "+string(nV)+
    " /centimetre cube ")
14
15 disp("electron mobility ,un = "+string(un)+" cm.sq/V-
    s")
16 disp("hole mobility ,up = "+string(up)+" cm.sq/V-s")
17 disp("temperature ,T = "+string(T)+" K")
18 disp("q = "+string(q)+" coulomb") //charge on an
    electron
19 disp("cross sectional area ,A =" +string(A)+" cm
    square")
20 disp("conductor length ,l =" +string(l)+" cm")
21 nD=nV/10^6//formulae
22 disp("donor concentration ,nD= nV/10^6=" +string(nD)+"
    /cm.cube")//calculation
23 nn=nD//formulae
24 disp("resulting mobile electron concentration ,nn= nD
    =" +string(nn)+" /cm.cube")//calculation
25 pn= ni^2/nD//formulae
26 disp("resulting hole concentration ,pn= ni^2/nD=" +
    string(pn)+" /cm.cube")//calculation
27 sigma=q*nD*un//formulae
28 disp("n-type semiconductor conductivity ,sigma=q*nD*
    un= "+string(sigma)+" (ohm-cm)^-1") //calculation
29 pn =(1/sigma)
30 disp("doped silicon resitivity (rho) ,pn =(1/sigma)=" +

```

```

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

```

Scilab code Exa 1.8 diffusion hole current

```

1 //Ex1_8
2 clc
3 q=1.59*10^(-19) //charge on an electron
4 disp("q = "+string(q)+" coulomb") //charge on an
    electron
5 d=0.037
6 disp(" dimension of semiconductor ,d="+string(d)+" cm"
    )
7 A=(d^2) //area formulae for square shaped
    semiconductor
8 disp(" cross sectional area ,A =d^2="+string(A)+" cm
    square")
9 x=10^(-4)
10 disp(" thickness ,x =" +string(x)+" cm")
11 x0=0
12 disp(" thickness ,x0 =" +string(x0)+" cm")
13 p=9.22*10^(15) //
14 disp(" hole concentration at x,p= "+string(p)+" /cm-
    cube")//calculation
15 p0=0//
16 disp(" hole concentration at x0 ,p0= "+string(p0)+" /
    cm-cube")//calculation
17 dp=(p-p0)//formulae
18 dx=(x-x0)//formulae
19 disp(" change in concentration at ,dp= "+string(dp)+

```

```

    " /cm-cube"))// calculation
20 disp(" change in thickness ,dx= "+string(dx)+" cm")//
    calculation
21 (dp/dx)==(p-p0)/(x-x0)//formulae
22 disp(" slope ,( dp/dx) =(p-p0)/(x-x0)="+string(dp/dx)+
    " holes/cm-cube"))// calculation
23 Dp=12
24 disp(" hole diffusion constant ,Dp= "+string(Dp)+" cm-
    sq/s"))// calculation
25 Ip=A*q*Dp*(dp/dx)
26 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

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

```

17 disp(" Diode current ,IA =IR*(exp(VAK3/(2*VT))-1)= "+
      string(IR*(exp(VAK3/(2*VT))-1))+ " ampere") //
      calculation
18 VAK4=(+0.6)
19 disp(" Junction voltage ,VAK4="+string(VAK4)+" volt")
20 disp(" Diode current ,IA =IR*(exp(VAK4/(2*VT))-1)= "+
      string(IR*(exp(VAK4/(2*VT))-1))+ " ampere") //
      calculation
21 VAK5=(+0.7)
22 disp(" Junction voltage ,VAK3="+string(VAK5)+" volt")
23 disp(" Diode current ,IA =IR*(exp(VAK5/(2*VT))-1)= "+
      string(IR*(exp(VAK5/(2*VT))-1))+ " ampere") //
      calculation
24 VAK6=(+0.8)
25 disp(" Junction voltage ,VAK3="+string(VAK6)+" volt")
26 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

```

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

```

```

        initialization
12 VR=0.6
13 disp("VR = "+string(VR)+ " volts")//initialization
14 IA=(VF-VD)/R //formulae
15 disp("Diode current ,IA = "+string(IA)+" ampere") //
    calculation
16 IA=(VF-VR)/(R+RF)// Formulae
17 disp(" using large signal model ,IA = "+string(IA)+"
    ampere") // calculation
18 VAK=(VR+IA*RF)// Formulae
19 disp(" Junction voltage ,VAK =" + string(VAK)+" volts")
    //calculation

```

Scilab code Exa 2.3 small signal forward and reverse resistance

```

1 //Ex2_3
2 clc
3 VT=26*10(-3)
4 disp(" Thermal voltage ,VT= "+string(VT)+" volt")//
    initialization
5 IR=50*10(-9)
6 disp("IR = "+string(IR)+" ampere") // value of
    Reverse saturation current
7 VAK1=(0.7)// diode junction voltage
8 disp(" Junction voltage ,VAK1="+string(VAK1)+" volt")
    //initialization
9 gf=(IR/(2*VT))*exp(VAK1/(2*VT)) //Formulae
10 disp(" Forward conductance ,gf="+string(gf)+" mho")
11 rf=1/gf //Formulae
12 disp(" Forward resistance ,rf = "+string(rf)+ " ohm")
13 VAK2=(-0.7)
14 gr=(IR/(2*VT))*exp(VAK2/(2*VT)) //Formulae
15 disp(" Reverse conductance ,gr="+string(gr)+" mho")
16 rr=1/gr //Formulae
17 disp(" Reverse resistance ,rr = "+string(rr)+ " ohm")

```

Scilab code Exa 2.4 various parameters for two diode rectifier

```
1 //Ex2_4
2 clc
3 Vi=10
4 disp("input voltage ,Vi = "+string(Vi)+" volts") //
  initialization
5 Rs=0.2
6 disp("resistance ,Rs = "+string(Rs)+ "ohm") //
  initialization
7 RL=10
8 disp("resistance ,RL = "+string(RL)+ "ohm") //
  initialization
9 VD=0.7
10 disp("input voltage ,VD = "+string(VD)+" volts") //
  initialization
11 Vim=Vi*sqrt(2) //Formulae
12 Iim=(Vim-VD)/(RL+Rs) //Formulae
13 disp(" Peak load current ,Iim =(Vim-VD)/(RL+Rs) =" +
  string(Iim)+" ampere") // calculation
14 Ildc=(2*Iim/(%pi)) //Formulae
15 disp(" D.C load current ,Ildc =(2*Iim/(%pi)) =" +
  string(Ildc)+" ampere") // calculation
16 Iadc=(Ildc/2) //Formulae
17 disp(" diode d.c current ,Iadc =(Ildc/2)= "+string(
  Iadc)+" ampere") // calculation
18 PIV=2*Vim //Formulae
19 disp("peak inverse voltage ,PIV = 2*Vim="+string(PIV
  )+" volts") // calculation
20 Vldc=Ildc*RL //Formulae
21 disp("D.C output voltage ,Vldc=Ildc*RL= "+string(Vldc
  )+" volts") // calculation
```

Scilab code Exa 2.5 calculation of resistance

```
1 //Ex2_5
2 clc
3 Idc=1*10^(-3)
4 disp(" D.C load current ,Idc = "+string(Idc)+"
      ampere") //initialization
5 Vi=2.5
6 disp("input voltage ,Vi = "+string(Vi)+" volts")//
  initialization
7 Vim=Vi*sqrt(2)
8 VD=0.7
9 disp("voltage drop ,VD = "+string(VD)+ " volts") //
  initialization
10 Rm=50
11 disp("resistance ,Rm = "+string(Rm)+ " ohm") //
  initialization
12 R=[(2/%pi)*((Vim-2*VD)/Idc)-Rm] //Formulae
13 disp("resistance ,R =[(2/%pi)*((Vim-2*VD)/Idc)-Rm]= "
      +string(R)+ " ohm")
14
15 // 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

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



```

6 f1=50
7 disp(" frequency ,f1="+string(f1)+" hertz") //
  initialization
8 RL=1100
9 disp(" resistance ,RL = "+string(RL)+ " ohm") //
  initialization
10 C=50*10^(-6)
11 r=1/[(4*sqrt(3))*f1*RL*C] // Formulae
12 disp(" Ripple factor ,r = "+string(r)+"")
13 x=1/(4*f1*RL*C) // Formulae
14 VLDC=Vim/(1+x) // Formulae
15 disp(" output voltage ,VLDC = VLDC=Vim/(1+x)="+string(
  VLDC)+" volts") // calculation
16 VR=(Vim-VLDC)/(VLDC) // Formulae
17 disp(" voltage Regulation ,VR =(Vim-VLDC)/(VLDC)= "+
  string(VR)+" volts") // calculation
18 Vr=VLDC*r // Formulae
19 disp(" Ripple output voltage ,Vr = Vr=VLDC*r="+string(
  Vr)+" volts") // calculation

```

Scilab code Exa 2.7 Zener diode parameters

```

1 //Ex2_7
2 clc
3 VI=10
4 disp(" input voltage ,VI = "+string(VI)+" volts") //
  initialization
5 Vz=5
6 disp(" diode voltage ,Vz = "+string(Vz)+" volts") //
  initialization
7 Rz=100
8 disp(" resistance ,Rz = "+string(Rz)+ " ohm") //
  initialization
9 RD=500
10 disp(" resistance ,RD = "+string(RD)+ " ohm") //

```

```

        initialization
11 DVI=25
12 disp("percentage change in VI,DVI= "+string(DVI)+"
        volts") //initialization
13 DVL=(DVI)*(Rz/(RD+Rz)) //Formulae
14 disp("percentage change in VL,DVL=(DVI)*(Rz/(RD+Rz))
        = "+string(DVL)+" %")
15 R0=(RD*Rz)/(RD+Rz) //Formulae
16 disp("Output resistance ,R0 =(RD*Rz)/(RD+Rz)= "+
        string(R0)+ " ohm")
17 VImax=12.5
18 Izmax=(VImax-Vz)/(RD+Rz) //Formulae
19 disp("resistance ,RD = "+string(RD)+ " ohm")
20 PZmax=(Izmax*Vz) //Formulae
21 disp("Power dissipated ,PZmax =PZmax=(Izmax*Vz)= "+
        string(PZmax)+ " watt")
22 Prd=(Izmax*Izmax*RD) //Formulae
23 disp("Power dissipated ,Prd=Prd=(Izmax*Izmax*RD)= "+
        string(Prd)+ " watt")
24 PD=(PZmax+Prd) //Formulae
25 disp("Power dissipated ,PD = "+string(PD)+ " watt")
26 RL=0.5*(10^3)
27 disp("resistance ,RL = "+string(RL)+ " ohm") //
        initialization
28 %VR=(R0*100)/RL //Formulae
29 disp("voltage Regulation Percentage ,%VR =(R0/RL)
        *(100)= "+string(%VR)+"% ")

```

Scilab code Exa 2.8 reference voltage and temperature coefficient

```

1 //Ex2_8
2 clc
3 Vz=10 //initialization
4 disp("diode voltage ,Vz = "+string(Vz)+" volts")
5 TC1=(10*0.02)/(100) //calculation

```

```

6 disp(" Zener diode TC1 = "+string(TC1)+" V/degree
      celsius")
7 VD=0.7
8 disp(" voltage drop ,VD = "+string(VD)+" volts")
9 TC2=(-2.5*10^(-3)) //calculation
10 disp(" Si diode TC = "+string(TC2)+" V/degree celsius
      ")
11 Vref1=VD+Vz
12 disp(" Combined voltage ,Vref=VD+Vz= "+string(Vref1)+
      " volts")
13 TC3=(TC1+TC2) //calculation
14 disp(" Combined TC = "+string(TC3)+" V/degree
      celsius")
15 TC=(TC1+TC2)*100/(Vref1) //calculation
16 disp("New Combined TC = (TC1+TC2)*100/(Vref1)="+
      string(TC)+" percent/degree celsius")
17 T1=25 //temperature
18 T2=50 // new temperature
19 Vref=Vref1-((-TC3)*(T2-T1)) //calculation
20 disp("New Combined reference voltage ,Vref= Vref1
      -((-TC3)*(T2-T1))="+string(Vref)+" volts")

```

Scilab code Exa 2.9 load voltage and current

```

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

```

```

9 Vcc=5
10 disp(" Supply voltage ,Vcc = "+string(Vcc)+" volts")
11 V01=VD+Vi1 //Formulae
12 disp(" output voltage ,V01 ==VD+Vi1 = "+string(V01)+"
    volts")
13 IL1=(Vcc-V01)/RL //Formulae
14 disp(" output current ,IL1=IL1=(Vcc-V01)/RL = "+
    string(IL1)+" ampere") // calculation
15 Vi2=5
16 disp("input voltage ,Vi2 = "+string(Vi2)+" volts") //
    initialization
17 V02=3*VD //Formulae
18 disp(" output voltage ,V02 =3*VD= "+string(V02)+"
    volts")
19 IL2=(Vcc-V02)/RL //Formulae
20 disp(" output current ,IL2= IL2=(Vcc-V02)/RL = "+
    string(IL2)+" ampere") // calculation
21 VAK=V02-Vi2 //Formulae
22 disp(" Diode voltage ,VAK = V02-Vi2 = "+string(VAK)+
    " volts")
23
24 //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

```
1 //Ex3_1
2 clc
3 Vcc=15
4 disp("Vcc = "+string(Vcc)+" volts") //
  initialization
5 VBB=1
6 disp("VBB = "+string(VBB)+" volts") //
  initialization
7 VBE=0.7
8 disp("VBE = "+string(VBE)+" volts") //
  initialization
9 RB=5*(10^3)
10 disp("resistance ,RB = "+string(RB)+ " ohm") //
  initialization
11 RL=650
12 disp("resistance ,RL = "+string(RL)+ " ohm") //
  initialization
13 Bf=200
14 disp("Gain ,Bf = "+string(Bf)+ " ") //initialization
15 IB=(VBB-VBE)/RB //Formulae
16 disp("IB =(VBB-VBE)/RB = "+string(IB)+" ampere") //
```

```

    calculation
17 IC=IB*Bf //Formulae
18 disp(" IC =IB*Bf= " +string(IC)+" ampere")//
    calculation
19 IE=IB+IC //Formulae
20 disp(" IE = IB+IC=" +string(IE)+" ampere")//
    calculation
21 VCE=Vcc-IC*RL //Formulae
22 disp("VCE =Vcc-IC*RL= " +string(VCE)+" volts") //
    calculation
23 VCB=VCE-VBE //Formulae
24 disp("VCB = VCE-VBE=" +string(VCB)+" volts")//
    calculation
25 RB=(Vcc-VBE)/IB //Formulae
26 disp("resistance ,RB =(Vcc-VBE)/IB= " +string(RB)+ "
    ohm") //calculation

```

Scilab code Exa 3.2 find current gain and voltage gain

```

1 //refer BJT characteristics in fig3.2(a)and fig3.4(a
  )
2 //Ex3-2
3 clc
4 Vbe1=0.025
5 disp("Vbe1 = " +string(Vbe1)+" volts") // value of
  base-emitter voltage1
6 Vbe2=(-0.025)
7 disp("Vbe2 = " +string(Vbe2)+" volts") // value of
  base-emitter voltage2
8 ib1=20*10(-6)
9 disp("ib1 = " +string(ib1)+" ampere") // value of
  base current1
10 ib2=(-20*10(-6))
11 disp("ib2 = " +string(ib2)+" ampere")// value of base
  current2

```

```

12 IBQ=60*10(-6)
13 disp("IBQ = "+string(IBQ)+" ampere") // operating
    point
14 ICP=15.5*10(-3)
15 disp("ICP = "+string(ICP)+" ampere") //
    initialization
16 ICR=8.5*10(-3)
17 disp("ICR = "+string(ICR)+" ampere") //
    initialization
18 VCEP=5
19 disp("VCEP = "+string(VCEP)+" volts") // value of
    collector-emitter voltage1
20 VCER=9
21 disp("VCER = "+string(VCER)+" volts") // value of
    collector-emitter voltage2
22 change_IC=ICP-ICR //change in collector current
23 disp("change_IC = "+string(change_IC)+" ampere")
24 change_VCE=VCER-VCEP //change in collector voltage
25 disp("change_VCE = "+string(change_VCE)+" volts")
26 change_VBE=Vbe1-Vbe2
27 change_IB=ib1-ib2
28 AV=(change_VCE/change_VBE) //formulae voltage gain
29 disp("AV = "+string(AV)+" ")//voltage gain
30 AI=change_IC/change_IB //formulae current gain
31 disp("AI = "+string(AI)+" ")

```

Scilab code Exa 3.3 Find transconductance g_m and hybrid parameter h_{ie} and unity gain frequency

```

1 //Ex3_3
2 clc
3 ICQ=12*10(-3)
4 disp("ICQ = "+string(ICQ)+" ampere") // collector
    current
5 B=200

```

```

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

```

Scilab code Exa 3.4 operating point of BJT

```

1 //Ex3_4
2 //refer fig3.2(a)
3 clc
4 VCC=20
5 disp("VCC = "+string(VCC)+" volts") // collector
    supply voltage
6 VBB=VCC
7 RL=5*(10^3)
8 disp("RL= "+string(RL)+ " ohm") //resistance
9 RB=965*(10^3)
10 disp("RB = "+string(RB)+ " ohm") //initialization
    base resistance
11 VBE=(0.7)
12 disp("VBE = "+string(VBE)+" volts") // value of base
    -emitter voltage
13 BF=50
14 disp("BF = "+string(BF)+" ") //BJT gain
15 ICO=10*10^(-9)
16 disp("ICO = "+string(ICO)+" ampere") // collector

```



```

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

```

Scilab code Exa 3.5 Find operating point of BJT

```

1 //Ex3_5
2 clc
3 BF1=100
4 disp("BF1 = "+string(BF1)+" ") //BJT gain
5 VCC=20
6 disp("VCC = "+string(VCC)+" volts") // collector
    supply voltage
7 VBB=VCC
8 RL=5*(10^3)
9 disp("resistance ,RL= "+string(RL)+ " ohm") //
    initialization
10 RB=965*(10^3)
11 disp("resistance ,RB = "+string(RB)+ " ohm") //
    initialization
12 VBE=(0.7)
13 disp("VBE = "+string(VBE)+" volts") // value of base
    -emitter voltage
14 ICO=10*10^(-9)
15 disp("ICO = "+string(ICO)+" ampere") // collector
    reverse bias current

```

```

16 Vi=0
17 disp("Vi = "+string(Vi)+" volts") // value of input
18 IBQ=(VCC-VBE)/RB //base current as operating point
19 disp("IBQ = "+string(IBQ)+" ampere")
20 ICQ1=BF1*IBQ //operating point (collector current)
21 disp("ICQ1 =BF1*IBQ= "+string(ICQ1)+" ampere") //
    calculation
22 VCEQ1=VCC-ICQ1*RL // collector-emitter voltage as
    operating point
23 disp("VCEQ1 =VCC-ICQ1*RL = "+string(VCEQ1)+" volts")
    //calculation

```

Scilab code Exa 3.6 Find operating point of BJT

```

1 //Ex3_6
2 clc
3 VBE2=(0.5)
4 disp("VBE2 = "+string(VBE2)+" volts") // value of
    base-emitter voltage
5 VCC=20
6 disp("VCC = "+string(VCC)+" volts") // collector
    supply voltage
7 BF2=150
8 disp("BF2 = "+string(BF2)+" ") //BJT gain
9 ICO2=500*10(-9)
10 disp("ICO2 = "+string(ICO2)+" ampere") // collector
    reverse bias current
11 RB=965*(103)
12 disp("RB = "+string(RB)+ " ohm") //initialization
    base resistance
13 RL=5*(103)
14 disp("RL= "+string(RL)+ " ohm") // load resistance
15 IBQ2=(VCC-VBE2)/RB //base current as operating
    point
16 disp("IBQ2 = (VCC-VBE2)/RB="+string(IBQ2)+" ampere")

```

```

17 ICQ2=(BF2*IBQ2) //operating point (collector
    current)
18 disp("ICQ2 =BF2*IBQ2= "+string(ICQ2)+" ampere") //
19 dICQ2=BF2*ICQ2 // increase in reverse bias current
20 disp("dICQ2 =BF2*ICQ2= "+string(dICQ2)+" ampere") //
21 ICQ3=ICQ2+dICQ2
22 disp("ICQ3 =ICQ2+dICQ2= "+string(ICQ3)+" ampere") //
    calculation
23 VCEQ3=VCC-ICQ3*RL // collector-emitter voltage as
    operating point
24 disp("VCEQ3 =VCC-ICQ3*RL = "+string(VCEQ3)+" volts")
    //calculation
25 //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

```

1 //Ex3_7
2 clc
3 VCC=20
4 disp("VCC = "+string(VCC)+" volts") // collector
    supply voltage
5 RL=5*(10^3)
6 disp("resistance ,RL= "+string(RL)+ " ohm") //
    initialization
7 R1=90*(10^3)
8 disp("resistance ,R1 = "+string(R1)+ " ohm") //
    initialization
9 R2=10*(10^3)
10 disp("resistance ,R2 = "+string(R2)+ " ohm") //
    initialization
11 Rc=1*(10^3)
12 disp("resistance ,Rc = "+string(Rc)+ " ohm") //

```

```

    resistance at collector
13 VBEmax=(0.7)
14 disp("VBEmax = "+string(VBEmax)+ " volts") // maximum
    base-emitter voltage
15 VBEmin=(0.5)
16 disp("VBEmin = "+string(VBEmin)+ " volts") // minimum
    base-emitter voltage
17 BFmax=150
18 disp("BFmax = "+string(BFmax)+ " ") //BJT gain
    maximum
19 BFmin=50
20 disp("BFmin = "+string(BFmin)+ " ") //BJT gain
    minimum
21 IC0max=500*10(-9)
22 disp("IC0max = "+string(IC0max)+ " ampere") //
    maximum collector reverse bias current
23 IC0min=10*10(-9)
24 disp("IC0min = "+string(IC0min)+ " ampere") //
    minimum collector reverse bias current
25 VBB=(VCC*R2)/(R1+R2)
26 disp("VBB = "+string(VBB)+ " volts") // Base supply
    voltage
27 RB=(R1*R2)/(R1+R2)
28 disp("RB = (R1*R2)/(R1+R2)="+string(RB)+ " ohm") //
    equivalent base resistance
29 ICmin=[(BFmin*(VBB-VBEmax)+(RB+Rc)*(1+BFmin)*IC0min)
    /(RB+Rc*(1+BFmin))] // minimum collector current
30 disp("ICmin = "+string(ICmin)+ " ampere")
31 VCEQmax=VCC-ICmin*RL // maximum collector-emitter
    voltage (d.c value)
32 disp("VCEQmax =VCC-ICmin*RL = "+string(VCEQmax)+
    volts") //calculation
33 ICmax=[(BFmax*(VBB-VBEmin)+(RB+Rc)*(1+BFmax)*IC0max)
    /(RB+Rc*(1+BFmax))] // maximum collector current
34 disp("ICmax = "+string(ICmax)+ " ampere")
35 VCEQmin=VCC-ICmax*RL // minimum collector-emitter
    voltage (d.c value)
36 disp("VCEQmin =VCC-ICmax*RL = "+string(VCEQmin)+

```

```

    volts") //calculation
37 change_IC=ICmax-ICmin
38 disp("change_IC= "+string(change_IC)+" ampere") //
    extreme variation in collector current
39 // 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 I_c and check is BJT in active region

```

1 //Ex3_8
2 clc
3 VCC=20
4 disp("VCC = "+string(VCC)+" volts") // collector
    supply voltage
5 RL=2*(10^3)
6 disp("RL= "+string(RL)+ " ohm") //resistance
7 R1=100*(10^3)
8 R2=R1
9 disp("R1 =R2= "+string(R1)+ " ohm") //resistance
10 VBE=(0.7)
11 disp("VBE = "+string(VBE)+" volts") // base-emitter
    voltage
12 BF=100
13 disp("BF = "+string(BF)+" ") //BJT gain
14 ICO=0
15 VBB=(VCC*R2)/(R1+R2)
16 disp("VBB = "+string(VBB)+" volts") // Base supply
    voltage
17 RB=(R1*R2)/(R1+R2)
18 disp("RB = (R1*R2)/(R1+R2)="+string(RB)+ " ohm") //
    equivalent base resistance
19 IC=[(BF*(VBB-VBE))/(RB+RL*(1+BF))] // collector
    current

```

```

20 disp("IC = "+string(IC)+" ampere")
21 VE=IC*RL
22 disp("VE = "+string(VE)+" volts") // emitter
    voltage
23 VB=VBE+VE
24 disp("VB = "+string(VB)+" volts") // base voltage
25 VCB=VCC-VB
26 disp("VCB = "+string(VCB)+" volts") // collector-
    base voltage
27 //hence BJT in active region.

```

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

```

1 //Ex3_9
2 clc
3 VCC=5
4 disp("VCC = "+string(VCC)+" volts") // collector
    supply voltage
5 RL=250
6 disp("RL= "+string(RL)+ " ohm") //initialization
7 RB=25*10^(3)
8 disp("RB ="+string(RB)+ " ohm") // base resistance
9 VCS=(0.2)
10 disp("VCS = "+string(VCS)+" volts") // voltage
11 BF=200
12 disp("BF = "+string(BF)+" ") //BJT gain
13 VBS=(0.8)
14 disp("VBS = "+string(VBS)+" volts") // base-emitter
    voltage for BJT switch
15 VI=5
16 disp("VI = "+string(VI)+" volts")// input voltage
17 VCON=0.3
18 disp("VCON = "+string(VCON)+" volts")
19 ICON=(VCC-VCON)/RL

```

```

20 disp("ICON = (VCC-VCON)/RL=" + string(ICON) + " ampere")
    //collector current for saturated BJT
21 IBON=(ICON)/BF
22 disp("IBON = (ICON)/BF=" + string(IBON) + " ampere") //
    Base current for saturated BJT
23 IBS=(VI-VBS)/RB
24 disp("IBS = (VI-VBS)/RB=" + string(IBS) + " ampere") //
    Base-emitter current for saturated BJT
25 ICS=(VCC-VCS)/RL
26 disp("ICS = (VCC-VCS)/RL=" + string(ICS) + " ampere") //
    Collector-emitter current for saturated BJT
27 Bforced=ICS/IBS
28 disp("Bforced = ICS/IBS=" + string(Bforced) + " ") //BJT
    forced gain
29 //IBS>>IBON hence BJT in saturation

```

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

```

1 //Ex3_10
2 clc
3 TJmax=175
4 disp("TJmax= " + string(TJmax) + " degree celsius") //
    maximum allowed junction temperature
5 theta=0.5
6 disp("theta= " + string(theta) + " degree celsius/mW ")
    //thermal resistances b/w junction to ambient
7 change_T=TJmax-25//temperature difference
8 PDmax=change_T/theta
9 disp("at 25 degree celsius ,PDmax=(TJmax-25 )/theta =
    " + string(PDmax) + " mW ") //maximum allowed
    power dissipation at TA=25 degree celsius
10 change_T=TJmax-75
11 PDmax2=change_T/theta
12 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

```
1 //Ex3_11  
2 clc  
3 TJmax=175  
4 disp("TJmax= "+string(TJmax)+ " degree celsius") //  
    maximum allowed junction temperature  
5 theta=0.1  
6 disp("theta= "+string(theta)+ " degree celsius/mW ")  
    //thermal resistances b/w junction to ambient  
7 change_T=TJmax-25 //temperature difference  
8 PDmax=change_T/theta  
9 disp("at 25 degree celsius ,PDmax=(TJmax-25 )/theta =  
    "+string(PDmax)+ " mW ") //maximum allowed  
    power dissipation at TA=25 degree celsius  
10 change_T=TJmax-75 //temperature difference  
11 PDmax=change_T/theta  
12 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

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



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

```
1 //Ex4_1
2 clc
3 IDSS=10*10^(-3)
4 disp("IDSS = "+string(IDSS)+" ampere") // maximum
   drain current
5 VP=(-4)
6 disp("VP= "+string(VP)+" volts") // pinch off
   voltage
7 VGS=(-2)
8 disp("VGS = "+string(VGS)+" volts") // gate to
   source voltage
9 VDSmin=VGS-VP
10 disp("VDSmin =VGS-VP="+string(VDSmin)+" volts") //
   Drain to source voltage
11 ID=IDSS*(1-VGS/VP)^2
12 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

```
1 //Ex4_2
2 clc
3 IDSS=10*10^(-3)
4 disp("IDSS = "+string(IDSS)+" ampere") // maximum
   drain current
5 VP=(-4)
6 disp("VP= "+string(VP)+" volts") // pinch off
   voltage
7 VGS=(0)
8 disp("VGS = "+string(VGS)+" volts") // gate to
   source voltage1
9 RDS=1/[(2*(IDSS/(-VP)))*(1-VGS/VP)]//formula for
   JFET
10 disp("RDS = 1/[(2*(IDSS/(-VP)))*(1-VGS/VP)]="+string
   (RDS)+" ohm") // drain to source resistance for
   VGS=0V
11 VGS=(-2)
12 disp("VGS = "+string(VGS)+" volts") // gate to
   source voltage2
13 RDS=1/[(2*(IDSS/(-VP)))*(1-VGS/VP)]
14 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

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

```

7 VT=(5)
8 disp("VT= "+string(VT)+" volts") // Threshold
  voltage
9 VGS=(10)
10 disp("VGS= "+string(VGS)+" volts") // gate to source
  voltage1
11 KF=ID/(VGS-VT)^2
12 disp("KF = ID/(VGS-VT)^2 = "+string(KF)+" A/V^2") //
  To calculate Scale factor for finding ID2
13 VDS=(7)
14 disp("VDS =VGS= "+string(VDS)+" volts") // drain to
  source voltage
15 VGS=(VDS)
16 ID=KF*(VGS-VT)^2
17 disp("ID =KF*(VGS-VT)^2= "+string(ID)+" ampere") //
  New drain current for VDS=24V
18 RL=(VDD-VDS)/ID
19 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

```

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

```

```

10 gm=[(2)*sqrt(IDQ*IDSS)]/VP
11 disp("gm =[(2)*sqrt(IDQ*IDSS)]/VP= "+string(gm)+" A/
    V")// calculating transconductance for JFET with
    IDQ = 4.42 mA
12
13 disp("part(ii) ")// part(ii) of this question
14 IDQ=6.04*10^(-3)
15 disp("IDQ = "+string(IDQ)+" ampere") // drain
    current for JFET 1
16 IDSS=15*10^(-3)
17 disp("IDSS = "+string(IDSS)+" ampere") // maximum
    drain current JFET2
18 VP=(6)
19 disp("VP= "+string(VP)+" volts") // pinch off
    voltage for JFET2
20 gm=[(2)*sqrt(IDQ*IDSS)]/VP
21 disp("gm =[(2)*sqrt(IDQ*IDSS)]/VP= "+string(gm)+" A/
    V")// calculating transconductance for JFET with
    IDQ = 6.04 mA
22
23 disp("part(iii) ")// part(iii) of this question
24 IDQ=1*10^(-3)
25 disp("IDQ = "+string(IDQ)+" ampere") // drain
    current for EMOSFET 1
26 KF=0.25*10^(-3)
27 disp("KF = "+string(KF)+" A/V^2") // Scale factor
    for finding EMOSFET1
28 gm=sqrt(4*IDQ*KF)
29 disp("gm =sqrt(4*IDQ*KF)= "+string(gm)+" A/V")//
    calculating transconductance for EMOSFET1 with
    IDQ = 1 mA
30
31 disp("part(iv) ")// part(iv) of this question
32 IDQ=0.91*10^(-3)
33 disp("IDQ = "+string(IDQ)+" ampere") // drain
    current for EMOSFET 2
34 KF=0.375*10^(-3)
35 disp("KF = "+string(KF)+" A/V^2") // Scale factor

```

```

    for finding EMOSFET2
36 gm=sqrt(4*IDQ*KF)
37 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

```

1 //Ex4_5
2 //Refer fig.4.8
3 clc
4 IDQmax=5*10(-3)
5 disp("IDQmax = "+string(IDQmax)+" ampere") // drain
    current for JFET for maximum transfer
    characteristics
6 IDmax=IDQmax// maximum drain current will be given
    by IDQmax
7 IDQmin=3*10(-3)
8 disp("IDQmin = "+string(IDQmin)+" ampere") // drain
    current for JFET for minimum transfer
    characteristics
9 VDD=(20)
10 disp("VDD= "+string(VDD)+" volts") // Drain voltage
    supply
11 VDSmin=(6)
12 disp("VDSmin= "+string(VDSmin)+" volts") // minimum
    Drain to source voltage supply
13 ID=2.4*10(-3)
14 disp("ID = "+string(ID)+" ampere") // drain current
    chosen for operation within max and min limits
15 VGG=3
16 disp("VGG= "+string(VGG)+" volts") // Gate voltage
    from fig.
17 Ri=100*10(3)
18 disp("Ri= "+string(Ri)+ " ohm") //equivalent input

```

```

    resistance
19 RF=(VGG-0)/(ID-0)
20 disp("RF= (VGG-0)/(ID-0)= "+string(RF)+ " ohm") //
    calculation for feedback resistance
21 R1=VDD*Ri/VGG //using formulae VGG=VDD*Ri/R1
22 disp("R1= VDD*Ri/VGG= "+string(R1)+ " ohm") //
    calculation for first resistance R1 at input
    side
23 R2=R1*VGG/(VDD-VGG)
24 disp("R2= R1*VGG/(VDD-VGG)= "+string(R2)+ " ohm")
    //calculation for second resistance R2 at input
    side
25 RL=[((VDD-VDSmin)/IDmax)-RF] // using formulae VDD=
    IDmax(RL+RF)+VDSmin
26 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

```

1 //Ex4_6
2 clc
3 IDSS=50*10^(-3)
4 disp("IDSS = "+string(IDSS)+" ampere") // maximum
    drain current JFET
5 VP=(-10)
6 disp("VP= "+string(VP)+" volts") // pinch off
    voltage for JFET
7 VGSQ=(-5)
8 disp("VGSQ= "+string(VGSQ)+" volts") // Gate
    operating point voltage
9 ID=IDSS*(1-VGSQ/VP)^2
10 disp("ID =IDSS*(1-VGS/VP)^2= "+string(ID)+" ampere")
    // drain current JFET
11 RF=abs(VGSQ/ID)
12 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

```
1 //Ex4_7
2 clc
3 IDSS=5*10^(-3)
4 disp("IDSS = "+string(IDSS)+" ampere") // maximum
   drain current JFET
5 RL=910
6 disp("RL= "+string(RL)+ " ohm") //Load resistance
7 RF=2.29*10^(3)
8 disp("RF= "+string(RF)+ " ohm") // feedback
   resistance
9 R1=12*10^(6)
10 disp("R1= "+string(R1)+ " ohm") // first resistance
   R1 at input side
11 R2=8.57*10^(6)
12 disp("R2= "+string(R2)+ " ohm") // second
   resistance R2 at input side
13 VDD=(24)
14 disp("VDD= "+string(VDD)+" volts") // Drain voltage
   supply
15 VP=(-2)
16 disp("VP= "+string(VP)+" volts") // pinch off
   voltage for JFET
17 VGG=(VDD*R2)/(R1+R2)
18 disp("VGG= VDD*R2/(R1+R2)="+string(VGG)+" volts") //
   Gate voltage for JFET
19 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
20 p = [5.244 -55.76 144]
```



```

21 ID=roots(p)*10^(-3)// values of ID converted into
    Ampere by multiplying by 10^(-3)
22 disp("ID = "+string(ID)+" ampere") // drain
    current JFET
23 disp("Since ID <=IDSS, hence ID=6.214 mA cannot be
    chosen, so we chose ID=4.42 mA")
24 IDQ=4.42*10^(-3)
25 disp("IDQ = "+string(IDQ)+" A")//Since ID <=IDSS,
    hence ID=6.214 mA cannot be chosen, so we chose
    ID=4.42 mA
26 VGSQ=VGG-IDQ*RF
27 disp("VGSQ = VGG-IDQ*RF = "+string(VGSQ)+" volts")
    // Gate operating point voltage
28 VDSQ=VDD-IDQ*(RL+RF)
29 disp("VDSQ= VDD-IDQ*(RL+RF)= "+string(VDSQ)+" volts"
    ) // Drain voltage for JFET
30 VDGQ=VDSQ-VGSQ
31 disp("VDGQ = VDSQ-VGSQ = "+string(VDGQ)+" volts") //
    Drain-Gate voltage for JFET
32 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

```

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

```

```

    resistance
9  R1=12*10^(6)
10 disp("R1= "+string(R1)+ " ohm") // first resistance
    R1 at input side
11 R2=8.57*10^(6)
12 disp("R2= "+string(R2)+ " ohm") // second
    resistance R2 at input side
13 VDD=(24)
14 disp("VDD= "+string(VDD)+" volts") // Drain voltage
    supply
15 VP=(-6)
16 disp("VP= "+string(VP)+" volts") // pinch off
    voltage for JFET
17 VGG=(VDD*R2)/(R1+R2)
18 disp("VGG= VDD*R2/(R1+R2)="+string(VGG)+" volts") //
    Gate voltage for JFET
19 disp("Quadratic equation =5.244*ID^(2) -75.68*ID
    +256=0") // where ID in mA
20 p = [5.244 -75.68 256]
21 ID=roots(p)*10^(-3) // values of ID converted into
    Ampere by multiplying by 10^(-3)
22 disp("ID = "+string(ID)+" ampere") // drain
    current JFET
23 VDG=VDD-(ID*RL)-VGG
24 disp("VDG= "+string(VDG)+" volts") // Drain-gate
    voltage for JFET
25 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 !")
26 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 !
27 disp("IDQ =" +string(IDQ)+ " ampere")
28 VGSQ=VGG-IDQ*RF
29 disp("VGSQ = VGG-IDQ*RF = "+string(VGSQ)+" volts")
    // Gate operating point voltage
30 VDSQ=VDD-IDQ*(RL+RF)
31 disp("VDSQ= VDD-IDQ*(RL+RF)= "+string(VDSQ)+" volts")

```

```

    ) // Drain voltage for JFET
32 VDGQ=VDSQ-VGSQ
33 disp("VDGQ = VDSQ-VGSQ =" +string(VDGQ)+" volts") //
    Drain-Gate voltage for JFET
34 disp("VDGQ > VP,Hence FET is in pinch off region")
35
36
37 //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

```

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

```

```

channel EMOSFET
17 VGG=(VDD*R2)/(R1+R2)
18 disp("VGG= VDD*R2/(R1+R2)=" +string(VGG)+" volts") //
    Gate voltage for n-channel EMOSFET
19 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
20 p = [9 -25 16]
21 ID=roots(p)*10^(-3)//values of ID converted into
    Ampere by multiplying by 10^(-3)
22 disp("ID = " +string(ID)+" A") // drain current n-
    channel EMOSFET in Ampere
23 VGS=VGG-ID*RF// For ID=1.78 mA and ID=1mA
24 disp("VGS = VGG-ID*RF = " +string(VGS)+" volts") //
    Gate operating point voltage
25 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")
26 IDQ=1*10^(-3)
27 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
28 VGSQ=VGG-IDQ*RF
29 disp("VGSQ = VGG-IDQ*RF = " +string(VGSQ)+" volts")
    // Gate operating point voltage
30 VDSQ=VDD-IDQ*(RL+RF)
31 disp("VDSQ= VDD-IDQ*(RL+RF)= " +string(VDSQ)+" volts"
    ) // Drain voltage for n-channel EMOSFET
32 // 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

```

1 //Ex4_10
2 clc
3 RL=12*10^(3)
4 disp("RL= "+string(RL)+ " ohm") //Load resistance
5 RF=6*10^(3)
6 disp("RF= "+string(RF)+ " ohm") // feedback
   resistance
7 R1=12*10^(6)
8 disp("R1= "+string(R1)+ " ohm") // first resistance
   R1 at input side
9 R2=8.57*10^(6)
10 disp("R2= "+string(R2)+ " ohm") // second
   resistance R2 at input side
11 VDD=(24)
12 disp("VDD= "+string(VDD)+" volts") // Drain voltage
   supply
13 VT=(3)
14 disp("VT= "+string(VT)+" volts") // Threshold
   voltage for n-channel EMOSFET
15 KF=0.375*10^(-3)
16 disp("KF= "+string(KF)+" A/V^2") // Constant for n-
   channel EMOSFET
17 VGG=(VDD*R2)/(R1+R2)
18 disp("VGG= VDD*R2/(R1+R2)="+string(VGG)+" volts") //
   Gate voltage for n-channel EMOSFET
19 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
20 p = [36 -86.67 49]
21 ID=roots(p)*10^(-3)//values of ID converted into
   Ampere by multiplying by  $10^{-3}$ 
22 disp("ID = "+string(ID)+" A") // drain current n-
   channel EMOSFET in Ampere
23 VGS=VGG-ID*RF// Gate voltage for ID = 1.5 mA and ID
   = 0.91 mA
24 disp("VGS = VGG-ID*RF = "+string(VGS)+" volts") //
   Gate voltage

```

```

25 disp(" Since  $V_{GS} < V_T$  for  $I_D=1.5$  mA, hence  $I_D = 1.5$ 
      mA cannot be chosen , so we chose  $I_D= 0.91$  mA as
      operating drain current  $I_{DQ}$ ")
26  $I_{DQ}=0.91*10^{(-3)}$ 
27 disp("  $I_{DQ} = "$ +string( $I_{DQ}$ )+" A")// Since  $V_{GS} < V_T$  for
       $I_D=1.5$  mA, hence  $I_D = 1.5$  mA cannot be chosen , so
      we chose  $I_D= 0.91$  mA as operating drain current
       $I_{DQ}$ 
28 change_IDQ=[(1-0.91)*100]/(1) //
29 disp(" change in  $I_{DQ} = "$ +string(change_IDQ)+" percent
      ")// Percent change in  $I_{DQ}$  from value 1 mA from
      its actual value  $I_{DQ}=0.91$ mA
30  $V_{GSQ}=V_{GG}-I_{DQ}*R_F$ 
31 disp("  $V_{GSQ} = V_{GG}-I_{DQ}*R_F = "$ +string( $V_{GSQ}$ )+" volts")
      // Gate operating point voltage
32  $V_{DSQ}=V_{DD}-I_{DQ}*(R_L+R_F)$ 
33 disp("  $V_{DSQ}= V_{DD}-I_{DQ}*(R_L+R_F)= "$ +string( $V_{DSQ}$ )+" volts"
      ) // Drain voltage for n-channel EMOSFET
34
35
36 // note: in the textbook author has given  $K_F = .375$ 
      but I have work with  $K_F = .375*10^{-3}A/V^2$ 

```

Scilab code Exa 4.11 Calculate load resistance R_L and percentage change in drain current I_D for JFET

```

1 //Ex4_11 Refer fig 4.7(b)// ANS is not correct check
  &correct
2 clc
3  $R_F=6*10^{(3)}$ 
4 disp("  $R_F= "$ +string( $R_F$ )+" ohm") // feedback
  resistance
5  $V_{DD}=(20)$ 
6 disp("  $V_{DD}= "$ +string( $V_{DD}$ )+" volts") // Drain voltage
  supply

```

```

7 disp("part(i) ")// part(i) of this question
8 VT=(2)
9 disp("VT= "+string(VT)+" volts") // Threshold
    voltage for EMOSFET
10 KF=0.25*10(-3)
11 disp("KF= "+string(KF)+" A/V2") // Constant for
    EMOSFET
12 ID=1*10(-3)
13 disp("ID = "+string(ID)+" A") // drain current
    EMOSFET in Ampere
14 RL=[VDD-VT-sqrt(ID/KF)]/ID // Using formulae ID=KF*(
    VDD-ID*RL-VT)
15 disp("RL=[VDD-VT-sqrt(ID/KF)]/ID= "+string(RL)+ "
    ohm") //Load resistance
16 disp("part(ii) ")// part(ii) of this question
17 VT=(3)
18 disp("VT= "+string(VT)+" volts") // Threshold
    voltage for EMOSFET
19 KF=0.375*10(-3)
20 disp("KF= "+string(KF)+" A/V2") // Constant for
    EMOSFET
21 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
22 p = [256 -546.66 289]
23 ID=roots(p)*10(-3)//values of ID converted into
    Ampere by multiplying by 10(-3)
24 disp("ID = "+string(ID)+" A") // drain current
    EMOSFET in Ampere
25 VDS=VDD-ID*RL// Drain voltage for ID = 1.173 mA and
    ID = 0.962 mA
26 disp("VDS =VDD-ID*RL = "+string(VDS)+" volts") //
    Drain voltage
27 IDQ=0.962*10(-3)
28 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
29 Percentage_change=[(1-0.962)*100]/(1)
30 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
31 // 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 EMOS-FET

```

1 //Ex4_12 Refer fig 4.9(a)and fig 4.9(b)
2 clc
3 VDD=(5)
4 disp("VDD= "+string(VDD)+" volts") // Drain voltage
    supply
5 RL1=125*10^(3)
6 disp("RL1= "+string(RL1)+ " ohm") //Load resistance
7 RL2=200*10^(3)
8 disp("RL2= "+string(RL2)+ " ohm") //Load resistance
9 IDON1=34.88*10^(-6)
10 disp("IDON1 =" +string(IDON1)+" A")//Drain current
    for load line 1 from fig.
11 IDON2=22.5*10^(-6)
12 disp("IDON2 =" +string(IDON2)+" A")//Drain current
    for load line 2 from fig.
13 VDON1=VDD-IDON1*RL1
14 disp("VDON1=VDD-IDON1*RL1= "+string(VDON1)+" volts")
    // output voltage at drain terminal for IDON1
15 VDON2=VDD-IDON2*RL2
16 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

```
1 //Ex4_13
2 clc
3 IDSS=10*10(-3)
4 disp("IDSS = "+string(IDSS)+" ampere") // maximum
    drain current for n-channel DEMOSFET
5 ID=IDSS // since VGS=0V, so ID=maximum
6 VP=(-4)
7 disp("VP= "+string(VP)+" volts") // pinch off
    voltage
8 VGS=(0)
9 disp("VGS= "+string(VGS)+" volts") // Gate to source
    voltage
10 VDD=(10)
11 disp("VDD= "+string(VDD)+" volts") // Drain supply
    voltage
12 RL=0.5*10(3)
13 disp("RL= "+string(RL)+ " ohm") //Load resistance
14 VDS=VDD-ID*RL
15 disp("VDS=VDD-ID*RL= "+string(VDS)+" volts") //
    Drain to source voltage ,since VDS>VP DEMOSFET
    is in pinch off
16 disp("VDS>VP,so pinch off region")
17 RL=0.75*10(3)
18 disp("RL= "+string(RL)+ " ohm") // New Load
    resistance value
19 VDS=VDD-ID*RL
20 disp("VDS=VDD-ID*RL= "+string(VDS)+" volts") // New
    Drain to source voltage for RL=750 ohm
21 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

```

1 //Ex4_14
2 clc
3 KF1=0.25*10^(-3)
4 disp("KF1 = "+string(KF1)+" A/V^2") // Scale factor
5 KF2=KF1
6 disp("KF2 = "+string(KF2)+" A/V^2") // Scale factor
7 IQ=1*10^(-3)
8 disp("IQ= "+string(IQ)+" ampere") // constant
   current source value
9 VT1=2
10 disp("VT1 = "+string(VT1)+" volts")// Threshold
   voltage
11 VT2=VT1
12 disp("VT2 = "+string(VT2)+" volts")// Threshold
   voltage
13 VDD=(15)
14 disp("VDD= "+string(VDD)+" volts") // Drain supply
   voltage
15 IREF=IQ
16 disp("IREF =IQ= "+string(IREF)+" ampere") //
   Reference current value
17 VGS=VT1+sqrt(2*IREF/KF1) // Formulae
18 disp("VGS= VT1+sqrt(2*IREF/KF1)="+string(VGS)+"
   volts") // Gate to source voltage
19 R=(VDD-VGS)/IREF
20 disp("R= (VDD-VGS)/IREF="+string(R)+ " ohm") //
   resistance value to obtain constant current
21 // 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

```

1 //Ex4_15
2 clc
3 RON=100
4 disp("RON= "+string(RON)+ " ohm") //ON resistance
    of analog series switch
5 ROFF=10^(10)
6 disp("ROFF= "+string(ROFF)+ " ohm") //OFF
    resistance analog series switch
7 Vip=1
8 disp("Vip= "+string(Vip)+" volts")// Peak amplitude
    of analog voltage
9 Rs=100
10 disp("Rs= "+string(Rs)+ " ohm") //Voltage source
    resistance
11 RL=10*10^(3)
12 disp("RL= "+string(RL)+ " ohm") //Load resistance
13 disp("part(i) ")// part(i) of this question
14 Vo=(Vip*RL)/(RL+RON+Rs)
15 disp("Vo=(Vip*RL)/(RL+RON+Rs)= "+string(Vo)+" volts"
    )// ON voltage
16 ErON=[Vip*(RON+Rs)/(RL+RON+Rs)]*100
17 disp("ErON=[Vip*(RON+Rs)/(RL+RON+Rs)]*100= "+string(
    ErON)+" percent")// Output voltage error
18 vOFF=(Vip*RL)/ROFF
19 disp("vOFF=(Vip*RL)/ROFF= "+string(vOFF)+" volts")//
    Output voltage in OFF state
20 OFF_isolation=20*log10(Vip/vOFF)
21 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.

```

```

22 disp("part(ii) ")// part(ii) of this question
23 vOFF=(Vip*RON)/(Rs+RON)
24 disp("vOFF=(Vip*RON)/(Rs+RON)= "+string(vOFF)+"
      volts")// Output voltage in OFF state for analog
      shunt switch
25 OFF_isolation=20*log10((Rs+RON)/RON)// OFF_isolation
      of shunt switch
26 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
27
28 // 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

```
1 //Ex5_1
2 clc
3 RL=5*10^(3)
4 disp("RL= "+string(RL)+ " ohm") //Load resistance
5 R1=100*10^(3)
6 disp("R1= "+string(R1)+ " ohm") // resistance
7 R2=10*10^(3)
8 disp("R2= "+string(R2)+ " ohm") // resistance
9 rc=50*10^(3)
10 disp("rc= "+string(rc)+ " ohm") //collector
    resistance
11 rd=rc // Drain and collector resistance are equal
12 rbe=1*10^(3)
13 disp("rbe= "+string(rbe)+ " ohm") //Load resistance
14 gm=50*10^(-3)
15 disp("gm = "+string(gm)+" A/V")// transconductance
    for BJT
16 Av=(-gm*RL)
17 disp(" For BJT ,Av=(-gm*RL)= "+string(Av)) //Voltage
    gain for BJT
```

```

18 AI=gm*rbe
19 disp("AI=(gm*rbe)= "+string(AI)) // current gain for
    BJT
20 gm=5*10^(-3)
21 disp("gm = "+string(gm)+" A/V")// transconductance
    for FET
22 Av=(-gm*RL)
23 disp("For FET,Av=(-gm*RL)= "+string(Av)+ " ") //
    gain for FET
24 R0=rd
25 disp("R0= "+string(R0)+ " ohm") //output resistance
    for FET and BJT
26 Ri=rbe
27 disp("Ri= "+string(Ri)+ " ohm") //BJT input
    resistance
28 RB=(R1*R2)/(R1+R2)
29 disp("RB=(R1*R2)/(R1+R2)= "+string(RB)+ " ohm") //
    equivalent Base resistance for BJT
30 Ri=(RB*rbe)/(RB+rbe)
31 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

```

1 //Ex5_2
2 clc
3 RL=5*10^(3)
4 disp("RL= "+string(RL)+ " ohm") //Load resistance
5 R1=100*10^(3)
6 disp("R1= "+string(R1)+ " ohm") // resistance
7 R2=100*10^(3)
8 disp("R2= "+string(R2)+ " ohm") // resistance
9 Rs=5*10^(3)
10 disp("Rs= "+string(Rs)+ " ohm") // Source

```

```

    resistance
11 Beta_o=50
12 disp("Beta_o = "+string(Beta_o)) //BJT gain
13 rbe=1*10^(3)
14 disp("rbe= "+string(rbe)+ " ohm") //Base-emitter
    resistance
15 gm=50*10^(-3)
16 disp("gm = "+string(gm)+" A/V")// transconductance
    for BJT
17 rc=50*10^(3)
18 disp("rc= "+string(rc)+ " ohm") //collector
    resistance
19 Av=RL/(RL+1/gm) // Gain formulae
20 disp("Av=RL/(RL+1/gm)= "+string(Av)) // voltage gain
    for BJT
21 Avs=RL/[(Rs/Beta_o)+(1/gm)+(RL)]
22 disp("Avs=RL/[(Rs/Beta_o)+(1/gm)+(RL)]= "+string(Avs
    )) // Overall voltage gain for BJT
23 AI=-(Beta_o+1)
24 disp("AI=-(Beta_o+1)= "+string(AI)) // current gain
    for BJT
25 R0=(Rs+rbe)/Beta_o
26 disp("R0= (Rs+rbe)/Beta_o="+string(R0)+ " ohm") //
    output resistance for BJT
27 Ri=rbe+Beta_o*RL // formulae
28 disp("Ri= rbe+Beta_o*RL="+string(Ri)+ " ohm") //
    value of BJT input resistance
29 RB=(R1*R2)/(R1+R2)
30 disp("RB=(R1*R2)/(R1+R2)= "+string(RB)+ " ohm") //
    equivalent Base resistance for BJT
31 Rieff=(Ri*RB)/(RB+Ri)
32 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

```

1 //Ex5_3
2 clc
3 RL=5*10^(3)
4 disp("RL= "+string(RL)+ " ohm") //Load resistance
5 RF=5*10^(3)
6 disp("RF= "+string(RF)+ " ohm") // resistance
7 Beta_o=50
8 disp("Beta_o = "+string(Beta_o)) //BJT gain
9 rbe=1*10^(3)
10 disp("rbe= "+string(rbe)+ " ohm") //Base-emitter
    resistance
11 gm=50*10^(-3)
12 disp("gm = "+string(gm)+" A/V")// transconductance
    for BJT
13 rc=50*10^(3)
14 disp("rc= "+string(rc)+ " ohm") //collector
    resistance
15 Ri=rbe+RF*(1+gm*rbe) // formulae
16 disp("Ri= rbe+RF*(1+gm*rbe)="+string(Ri)+ " ohm")
    // BJT input resistance
17 Av=(-gm*RL)/(1+gm*RF)// formulae
18 disp("Av=(-gm*RL)/(1+gm*RF)=" +string(Av)) //
    voltage gain for BJT
19 AI=Beta_o
20 disp("AI=(Beta_o)=" +string(AI)) // current gain for
    BJT
21 R0=Beta_o*rc
22 disp("R0= Beta_o*rc="+string(R0)+ " ohm") //output
    resistance for BJT

```

Scilab code Exa 5.4 Find overall voltage gain A_{vs} and input resistance R_i and output resistance R_o for CB and CG amplifier


```

1 //Ex5_4
2 clc
3 RL=5*10^(3)
4 disp("RL= "+string(RL)+ " ohm") //Load resistance
5 RF=2.5*10^(3)
6 disp("RF= "+string(RF)+ " ohm") // resistance
7 Rs=50
8 disp("Rs= "+string(Rs)+ " ohm") // resistance
9 ro=50*10^(3)
10 disp("ro= "+string(ro)+ " ohm") // output
    resistance
11 rd=ro // drain resistance
12 rc=ro// Collector resistance
13 disp("rc= "+string(rc)+ " ohm") // Collector
    resistance
14 rbe=1*10^(3)
15 disp("rbe= "+string(rbe)+ " ohm") //base -emitter
    resistance
16 disp("For CG Amplifier")
17 gm=5*10^(-3)
18 disp("gm = "+string(gm)+" A/V")// transconductance
    for FET
19 Ri=1/gm // formulae
20 disp("Ri= 1/gm="+string(Ri)+ " ohm") // value of
    CGA (common gate amplifier)input resistance for
    FET
21 Avs=gm*RL/(1+gm*Rs)
22 disp("Avs=gm*RL/(1+gm*Rs)= "+string(Avs)) // Overall
    voltage gain for FET (CGA)
23 Ro=rd*(1+gm*Rs)
24 disp("Ro=rd*(1+gm*Rs)="+string(Ro)+ " ohm") //
    output resistance for FET (CGA)
25 disp("For CB Amplifier")
26 gm=50*10^(-3)
27 disp("gm = "+string(gm)+" A/V")// transconductance
    for BJT
28 Ri=1/gm // formulae
29 disp("Ri= 1/gm="+string(Ri)+ " ohm") // value of

```

```

    CBA (common base amplifier) input resistance for
    BJT
30 Avs=gm*RL/(1+gm*Rs)
31 disp("Avs=gm*RL/(1+gm*Rs)= "+string(Avs)) // Overall
    voltage gain for BJT (CBA)
32 Ro=gm*(rbe*rc)
33 disp("Ro=gm*(rbe*rc)="+string(Ro)+ " ohm") //output
    resistance for BJT (CBA)
34
35 //NOTE: I have calculated first all the parameters
    for CG amplifier and then for CB amplifier but in
    book parameters have been calculated
    alternately for CG and CB amplifiers.

```

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

```

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

```

```

16 disp("fH=1/[2*(%pi)*(RL)*(CSH)]= "+string(fH)+ " Hz"
    ) // Higher cutoff-frequency
17 BW=fH-fL
18 disp("BW=fH-fL= "+string(BW)+ " Hz") // Bandwidth
19 fT=Avm*fH
20 disp("fT=Avm*fH= "+string(fT)+ " Hz") // Unity gain
    bandwidth
21 // 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

```

1 //Ex5_6
2 clc
3 IDSS=16*10^(-3)
4 disp("IDSS = "+string(IDSS)+" ampere") // maximum
    drain current JFET
5 VP=(-4)
6 disp("VP= "+string(VP)+" volts") // pinch off
    voltage for JFET
7 VGSQ=(-2)
8 disp("VGSQ= "+string(VGSQ)+" volts") // Gate
    operating point voltage
9 Vsm=(0.2)
10 disp("Vsm= "+string(Vsm)+" volts") // sinusoidal
    input voltage for JFET
11 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
12 disp("D=[((0.5)*(Vsm)^2)/(4*Vsm)]*100 =" +string(D)+
    "% ") // Percentage second harmonic distortion
    calculation

```

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

```

1 //Ex5_7
2 clc
3 Ic=1*10^(-3)
4 disp("Ic = "+string(Ic)+" ampere") // collector
    current BJT
5 rbe=2*10^(3)
6 disp("rbe= "+string(rbe)+ " ohm") //base -emitter
    resistance
7 gm=50*10^(-3)
8 disp("gm = "+string(gm)+" A/V")// transconductance
    for BJT
9 Beta_o=100
10 disp("Beta_o = "+string(Beta_o)+" ") //BJT gain
11 rc=50*10^(3)
12 disp("rc= "+string(rc)+ " ohm") //collector
    resistance
13 Cbe=10*10^(-12)
14 disp("Cbe= "+string(Cbe)+ " farad") //base -emitter
    capacitance
15 Ctc=1*10^(-12)
16 disp("Ctc= "+string(Ctc)+ " farad") //input device
    capacitance
17 disp("part(i)")// part(i) of question
18 RL=10*10^(3)
19 disp("RL= "+string(RL)+ " ohm") //Load resistance
20 Rs=500
21 disp("Rs= "+string(Rs)+ " ohm") //input source
    resistance
22 Rth=(Rs*rbe)/(Rs+rbe)
23 disp("Rth=(Rs*rbe)/(Rs+rbe)="+string(Rth)+ " ohm")
    // equivalent resistance

```

```

24 Avm=(-gm*RL)
25 disp("Avm=(-gm*RL)=" +string(Avm)) // Mid-frequency
    gain for CE amplifier
26 CM=Ctc*(1-Avm)
27 disp("CM=Ctc*(1-Avm)= " +string(CM)+ " farad") //
    calculated capacitance
28 Ci=Cbe
29 disp("Ci=Cbe= " +string(Ci)+ " farad") //calculated
    input capacitance
30 fHi=1/[2*(%pi)*(Rth)*(Cbe+CM)]
31 disp("fHi=1/[2*(%pi)*(Rth)*(Cbe+CM)]= " +string(fHi)+
    " Hz") // Higher-frequency cutoff for CE
    amplifier
32 Ri=rbe
33 disp("Ri=rbe =" +string(Ri)+ " ohm") //input
    resistance CE amplifier
34 Ro=rc
35 disp("Ro= rc=" +string(Ro)+ " ohm") //output
    resistance for CE amplifier
36 fB=1/[2*(%pi)*(rbe)*(Cbe)]
37 disp("fB=1/[2*(%pi)*(rbe)*(Cbe)]= " +string(fB)+ " Hz
    ") // base terminal frequency cutoff
38 fT=Beta_o*fB
39 disp("fT=Beta_o*fB= " +string(fT)+ " Hz") // Unity
    gain bandwidth for CE amplifier
40 disp("part(ii)") // part(ii) of question
41 Rs=50*10^(3)
42 disp("Rs= " +string(Rs)+ " ohm") //input source
    resistance for CC amplifier
43 RL=1*10^(3)
44 disp("RL= " +string(RL)+ " ohm") //Load resistance
    for CC amplifier
45 fhi=1/[2*(%pi)*(Rs)*(Ctc)]
46 disp("fhi=1/[2*(%pi)*(Rs)*(Ctc)]= " +string(fhi)+ "
    Hz") // Higher-frequency cutoff for CC amplifier
47 Avm=(gm*RL)/(1+gm*RL)
48 disp("Avm=(gm*RL)/(1+gm*RL)=" +string(Avm)) // Mid-
    frequency gain for CC amplifier

```

```

49 Ro=1/gm
50 disp("Ro= 1/gm="+string(Ro)+ " ohm") //output
    resistance for CC amplifier
51 Ri=Beta_o*RL
52 disp("Ri=Beta_o*RL =" +string(Ri)+ " ohm") //input
    resistance CE amplifier
53 disp(" part( iii)")// part( iii)of question
54 RL=10*10^(3)
55 disp("RL= " +string(RL)+ " ohm") //Load resistance
    for CB amplifier
56 Rs=50
57 disp("Rs= " +string(Rs)+ " ohm") //input source
    resistance for CB amplifier
58 fHi=gm/[2*(%pi)*(Cbe)]
59 disp("fHi=gm/[2*(%pi)*(Cbe)]= " +string(fHi)+ " Hz")
    // Higher-frequency cutoff for CB amplifier
60 fHo=1/[2*(%pi)*(Ctc)*(RL)]
61 disp("fHo=gm/[2*(%pi)*(Ctc)*(RL)]= " +string(fHo)+ "
    Hz") // Higher-frequency cutoff for CB amplifier
62 Avs=(gm*RL)/(1+gm*Rs)
63 disp("Avs=(gm*RL)/(1+gm*Rs)="+string(Avs)) // Mid-
    frequency gain for CB amplifier
64 Ri=1/gm
65 disp("Ri= 1/gm="+string(Ri)+ " ohm") //output
    resistance for CB amplifier
66 Ro=Beta_o*rc
67 disp("Ro=Beta_o*rc =" +string(Ro)+ " ohm") //input
    resistance CB amplifier
68 //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

```
1 //Ex5_8
2 clc
3 tp=10*10(-3)
4 disp("tp= "+string(tp)+" s") // Time period of pulse
5 tr=0.05*10(-6)
6 disp("tr= "+string(tr)+" s") // Rise-Time of pulse
7 CSH=50*10(-12)
8 disp("CSH= "+string(CSH)+ " farad") //output
   capacitor
9 tilt=5
10 disp("percentage tilt= "+string(tilt)+ "%") //Sag
   or percentage tilt of output
11 Ri=100*10(3)
12 disp("Ri= "+string(Ri)+ " ohm") // source
   resistance
13 RL=tr/(2.2*CSH)
14 disp("RL=tr/(2.2*CSH)= "+string(RL)+ " ohm") //Load
   resistance calculation
15 Cc=(tp*100)/(tilt*Ri)
16 disp("Cc= (tp*100)/( tilt*Ri)="+string(Cc)+ " farad"
   ) //capacitance
17 //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

```
1 //Ex6_1
2 clc
3 Av=0.1
4 disp("Av= "+string(Av)) //Voltage gain
5 AvdB=20*log10(Av)
6 disp("Av(dB)=20*log10(Av)= "+string(AvdB)+"dB ") //
   Voltage gain in decibel
7 Av=0.707
8 disp("Av= "+string(Av)) //Voltage gain
9 AvdB=20*log10(Av)
10 disp("Av(dB)=20*log10(Av)= "+string(AvdB)+"dB ") //
   Voltage gain in decibel
11 Av=1
12 disp("Av= "+string(Av)) //Voltage gain
13 AvdB=20*log10(Av)
14 disp("Av(dB)=20*log10(Av)= "+string(AvdB)+"dB ") //
   Voltage gain in decibel
15 Av=10
16 disp("Av= "+string(Av)) //Voltage gain
```



```

17 AvdB=20*log10(Av)
18 disp("Av(dB)=20*log10(Av)= "+string(AvdB)+" dB ") //
    Voltage gain in decibel
19 Av=100
20 disp("Av= "+string(Av)) //Voltage gain
21 AvdB=20*log10(Av)
22 disp("Av(dB)=20*log10(Av)= "+string(AvdB)+" dB ") //
    Voltage gain in decibel
23 Av=1000
24 disp("Av= "+string(Av)) //Voltage gain
25 AvdB=20*log10(Av)
26 disp("Av(dB)=20*log10(Av)= "+string(AvdB)+" dB ") //
    Voltage gain in decibel
27 //NOTE:calculated voltage gain in dB for Av=0.707 is
    -3.0116117dB

```

Scilab code Exa 6.2 Find voltage gain A_v and current gain A_i and power gain A_p for amplifier

```

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

```

```

    ) //Voltage gain in decibel
15 Iim=Vim/Ri
16 disp("Iim= Vim/Ri= "+string(Iim)+" A") // Input peak
    current
17 Ii=Iim/sqrt(2) //RMS value of input current
18 Iom=Vom/RL
19 disp("Iom= Vom/RL= "+string(Iom)+" A") // Output
    peak current
20 Io=Iom/sqrt(2) //RMS value of Output current
21 Ai=20*log10(Io/Ii)
22 disp("Ai=20*log10(Io/Ii)= "+string(Ai)+" dB ") //
    Current gain in decibel
23 pi=Vi^2/Ri
24 disp("pi= Vi^2/Ri= "+string(pi)+" W") // Input power
25 po=Vo^2/RL
26 disp("po= Vo^2/RL= "+string(po)+" W") // Output
    power
27 Ap=10*log10(po/pi)
28 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

```

1 //Ex6_3
2 clc
3 RL=1*10^(3)
4 disp("RL= "+string(RL)+ " ohm") //Load resistance
5 RF=500*10^(3)
6 disp("RF= "+string(RF)+ " ohm") //Feedback
    resistance
7 Beta_o=50
8 disp("Beta_o = "+string(Beta_o)) //BJT gain
9 rbe=1*10^(3)
10 disp("rbe= "+string(rbe)+ " ohm") //Base-emitter

```

```

    resistance
11 gm=50*10(-3)
12 disp("gm = "+string(gm)+" A/V") // transconductance
    for BJT
13 rc=50*10(3)
14 disp("rc= "+string(rc)+ " ohm") // collector
    resistance
15 disp("part(i)")
16 Adm1=(-gm*RL)
17 disp("Adm1=(-gm*RL)= "+string(Adm1)) // Differential
    mode gain for BJT for DIDO and SIDO modes
18 Adm2=(0.5*gm*RL)
19 disp("Adm2=(0.5*gm*RL)= "+string(Adm2)) //
    Differential mode gain for BJT for DISO and SISO
    modes
20 Rid=2*rbe
21 disp("Rid=2*rbe= "+string(Rid)+ " ohm") //input
    differential mode resistance
22 Acm=(-RL)/(2*RF)
23 disp("Acm=(-RL)/(2*RF)= "+string(Acm)) // Common
    mode gain for BJT for DISO and SISO modes
24 Ric=Beta_o*RF
25 disp("Ric=Beta_o*RF= "+string(Ric)+ " ohm") //
    common mode input resistance
26 CMRR=2*gm*RF
27 disp("CMRR=2*gm*RF= "+string(CMRR)) // common mode
    rejection ratio
28 disp("part(ii)")
29 Vi1=(-0.5)*10(-3)
30 disp("Vi1= "+string(Vi1)+" volts") // input voltage1
31 Vi2=(+0.5)*10(-3)
32 disp("Vi2= "+string(Vi2)+" volts") // input voltage2
33 Vcm=(10)*10(-3)
34 disp("Vcm= "+string(Vcm)+" volts") // common mode
    voltage
35 Vd=Vi1-Vi2
36 disp("Vd=Vi1-Vi2= "+string(Vd)+" volts") //
    differential voltage

```

```

37 Vod=abs(Vd*Adm2)
38 disp("Vod=abs(Vd*Adm2)= "+string(Vod)+" volts") //
    output differential voltage for DISO and SISO
    modes
39 Voc=abs(Vcm*Acm)
40 disp("Voc=abs(Vcm*Acm)= "+string(Voc)+" volts") //
    output common mode voltage
41 Error=(Voc/Vod)*100
42 disp("percentage error=(Voc/Vod)*100= "+string(Error
    )+"%")//percentage error due to CM signal
43 disp("part(iii)")
44 RLeff=(RL*Rid)/(RL+Rid)
45 disp("RLeff=(RL*Rid)/(RL+Rid)= "+string(RLeff)+ "
    ohm") // Effective load resistance
46 Adm=gm*RLeff
47 disp("Adm=gm*RLeff= "+string(Adm)) // Modified
    Differential mode gain for BJT for DIDO and SIDO
    modes
48 Acm=(-RLeff)/(2*RF)
49 disp("Acm=(-RLeff)/(2*RF)= "+string(Acm)) //
    Modified Common mode gain for BJT for DISO and
    SISO modes
50 CMRR=abs(Adm/(Acm))
51 disp("CMRR=abs(Adm/(Acm))= "+string(CMRR)) //
    Modified common mode rejection ratio
52 //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.
53 // 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

```

1 //Ex6_4
2 clc
3 VCC=(10)
4 disp("VCC= "+string(VCC)+" volts") // Collector
    voltage supply
5 VEE=VCC
6 disp("VEE=VCC= "+string(VEE)+" volts") // Emitter
    supply voltage
7 IQ=2*10^(-3)
8 disp("IQ = "+string(IQ)+" ampere") // operating
    current for CC class-A amplifier
9 VBE=(0.7)
10 disp("VBE= "+string(VBE)+" volts") // Base-emitter
    voltage
11 disp(" part (i) ")
12 RL=VCC/IQ
13 disp("RL=VCC/IQ= "+string(RL)+ " ohm") //Load
    resistance
14 Pomax=VCC^2/(2*RL)
15 disp("Pomax=VCC^2/(2*RL)= "+string(Pomax)+" W") //
    maximum Output power
16 PDC=2*VCC*IQ
17 disp("PDC=2*VCC*IQ= "+string(PDC)+" W") // Total D.C
    power supply
18 Etta_max=(Pomax/PDC)*100
19 disp(" Efficiency ,Etta_max=(Pomax/PDC)*100= "+string(
    Etta_max)+"%") //maximum power amplifier
    conversion efficiency
20 PDmax=VCC*IQ
21 disp("PDmax=VCC*IQ= "+string(PDmax)+" W") // maximum
    power dissipation
22 disp(" part (ii) ")
23 Vcm=(5)
24 disp("Vcm= "+string(Vcm)+" volts") // common mode
    voltage

```

```

25 Po=Vcm^2/(2*RL)
26 disp("Po=Vcm^2/(2*RL)= "+string(Po)+" W") // Output
    power
27 Etta=(Po/PDC)*100
28 disp("Efficiency ,Etta=(Po/PDC)*100= "+string(Etta)+"
    %") // power amplifier conversion efficiency
29 PDCavg=PDmax-Po//Using law of conservation of energy
30 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

```

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

```

```

17 disp(" part ( ii )")
18 Vcm=VCC//For Full output voltage swing Vcm=VCC
19 disp("Vcm=VCC =" +string(Vcm)+" volts") // common
    mode voltage for full swing condition
20 Icm=VCC/RL
21 disp("Icm = VCC/RL=" +string(Icm)+" ampere") //
    common mode current
22 Po=(1/2)*(Icm*Vcm)
23 disp("Po=(1/2)*(Icm*Vcm)=" +string(Po)+" W") //
    Output power at full swing condition
24 ICavg=(Icm)/(%pi)
25 disp("ICavg=(Icm)/(%pi)=" +string(ICavg)+" ampere")
    // Average value of common mode current
26 PDC=2*(ICavg*VCC)
27 disp("PDC=2*VCC*ICavg= " +string(PDC)+" W") // Total
    D.C power supply for full swing condition
28 Etta=(Po/PDC)*100
29 disp(" Efficiency ,Etta=(Po/PDC)*100= " +string(Etta)+"
    %") // power amplifier conversion efficiency
30 disp(" part ( iii )")
31 Vcm1=(5)//given value
32 disp("Vcm1= " +string(Vcm1)+" volts") // common mode
    voltage for output swing Vcm=5 V
33 ICavg1=(Vcm1)/(%pi*RL)
34 disp("ICavg1=(Vcm1)/(%pi*RL)=" +string(ICavg1)+"
    ampere") // Average value of common mode current
35 Po1=(Vcm1^2)/(2*RL)
36 disp("Po1=(Vcm1^2)/(2*RL)=" +string(Po1)+" W") //
    Output power for output swing Vcm=5 V
37 PDC1=2*(ICavg1*VCC)
38 disp("PDC1=2*VCC*ICavg1= " +string(PDC1)+" W") //
    Total D.C power supply for output swing Vcm=5 V
39 Etta=(Po1/PDC1)*100
40 disp(" Efficiency ,Etta=(Po1/PDC1)*100= " +string(Etta)
    +" %") // power amplifier conversion efficiency
    for output swing Vcm=5 V
41 // 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 V_{dmax} for op amp without feedback

```
1 //Ex6_6
2 clc
3 Av=1*10^(5)
4 disp("Av= "+string(Av)) //Voltage gain
5 VCC=(10)
6 disp("VCC= "+string(VCC)+" volts") // Collector
   voltage supply
7 vo=VCC
8 disp("vo= VCC="+string(vo)+" volts") // maximum
   output voltage
9 Vdmax=VCC/Av
10 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

```
1 //Ex7_1
2 clc
3 A=60000
4 disp("A= "+string(A)) //Amplifier gain
5 Af=10000
6 disp("Af= "+string(Af)) //Feedback gain
7 N_dB=20*log10(Af/A)
8 disp("N_dB=20*log10(Af/A)= "+string(N_dB)+"dB") //
   Negative feedback gain
9 B=[1/(Af)]-(1/A)// formulae using [Af=A/(1+A*B)]
10 disp("B=[1/(Af)]-(1/A)= "+string(B)) //Feedback
   factor
```

Scilab code Exa 7.2 Find gain with feedback

```

1 //Ex7_2
2 clc
3 A=10000
4 disp("A= "+string(A)) //Amplifier gain
5 B=0.01
6 disp("B= "+string(B)) //Feedback factor
7 Af=[A/(1+A*B)]
8 disp(" Af= [A/(1+A*B)]="+string(Af)) //Feedback gain
9 A1=100000
10 disp("A1= "+string(A1)) //New amplifier gain value
11 Af1=[A1/(1+A1*B)]
12 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

```

1 //Ex7_3
2 clc
3 Vo=(50)
4 disp("Vo= "+string(Vo)+" volts") // output voltage
5 Vi=(0.5)
6 disp("Vi= "+string(Vi)+" volts") // input voltage
7 disp("part(i)")
8 A=Vo/Vi
9 disp("A= Vo/Vi="+string(A)) //Amplifier gain
10 H.D=10
11 disp(" Harmonic_distortion="+string(H.D)+"%") //
    Percentage second harmonic distortion
12 D=(10*Vo)/100
13 disp("D= (10*Vo)/100 = "+string(D)+" volts") //
    Second Harmonic distortion
14 Df=(1*Vo)/100
15 disp("Df= (1*Vo)/100 = "+string(Df)+" volts") //
    Harmonic distortion with Feedback

```

```

16 B=[D/(Df*A)]-(1/A) //Using formulae Df=[D/(1+A*B)]
17 disp("B=[D/(Df*A)]-(1/A) = "+string(B)) //Feedback
    factor
18 disp(" part ( ii )")
19 Af=[A/(1+A*B)]
20 disp(" Af= [A/(1+A*B)] = "+string(Af)) //Feedback
    gain
21 disp(" part ( iii )")
22 Vif=Vo/Af
23 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

```

1 //Ex7_4
2 clc
3 GBW=10^(6)
4 disp("GBW= "+string(GBW)+" Hz") // Gain-Bandwidth
    product
5 AMf=100
6 disp("AMf="+string(AMf)) // Midband gain with
    feedback
7 fHF=GBW/AMf
8 disp("fHF=GBW/AMf= "+string(fHF)+" Hz") //Signal
    bandwidth
9 f_10percent=(10*fHF)/100
10 disp("f_10per cent=(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

```

1 //Ex7_5
2 clc
3 AM=50000
4 disp("AM=" + string(AM)) // Midband gain
5 fH=20*10^(3)
6 disp("fH= " + string(fH) + " Hz") // Upper cut-off
  frequency
7 fL=30
8 disp("fL= " + string(fL) + " Hz") // Lower cut-off
  frequency
9 B=5*10^(-5)
10 disp("B= " + string(B)) // Feedback factor
11 AMf=AM/(1+B*AM)
12 disp("AMf=AM/(1+B*AM)=" + string(AMf)) // Midband gain
  with feedback
13 fHf=fH*(1+B*AM)
14 disp("fHf=fH*(1+B*AM)= " + string(fHf) + " Hz") // Upper
  cut-off frequency with feedback
15 fLf=fL/(1+B*AM)
16 disp("fLf=fL/(1+B*AM)= " + string(fLf) + " Hz") // Lower
  cut-off frequency with feedback
17 //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

```

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

```

```

8 fc2=10^5
9 disp(" fc2= "+string(fc2)+" Hz")// Second Critical
  frequency
10 fc3=10^6
11 disp(" fc3= "+string(fc3)+" Hz")// Third Critical
  frequency
12 disp(" part (i) ")
13 Af1=85
14 disp(" Af1="+string(Af1)+"dB") // gain at 50 kHz and
  -20dB/decade roll-off
15 f=50*10^(3)
16 disp(" f= "+string(f)+" Hz")// operating frequency
17 theta_A=- atand(f/fc1)- atand(f/fc2)- atand(f/fc3)
  //phase shift in radians
18 disp(" theta_A="+string(theta_A)+" degree")// Phase
  shift for feedback gain Af1
19 theta_pm=180-abs(theta_A)// formulae phase margin
20 disp(" theta_pm=180-abs(theta_A)="+string(theta_pm)+"
  degree")// Phase Margin for feedback gain Af1
21 disp(" Amplifier stable")// Since phase margin is (+)
  ive
22 disp(" part (ii) ")
23 Af2=50
24 disp(" Af2="+string(Af2)+" dB") // gain at 500 kHz
  and -40dB/decade roll-off
25 f=500*10^(3)
26 disp(" f= "+string(f)+" Hz")// frequency
27 theta_A=- atand(f/fc1)- atand(f/fc2)- atand(f/fc3)
  //phase shift in radians
28 disp(" theta_A= "+string(theta_A)+" degree")// Phase
  shift for feedback gain Af2
29 theta_pm=180-abs(theta_A)// formulae phase margin
30 disp(" theta_pm=180-abs(theta_A)="+string(theta_pm)+"
  degree")// Phase Margin for feedback gain Af1
31 disp(" Amplifier unstable")// Since phase margin is
  (-)ive
32 disp(" part (iii) ")
33 Af3=20

```

```

34 disp(" Af3=" +string(Af3)+"dB") // gain at 1100 kHz
    and -60dB/decade roll-off
35 f=1100*10^(3)
36 disp(" f= " +string(f)+" Hz") // frequency
37 theta_A=- atand(f/fc1)- atand(f/fc2)- atand(f/fc3)
    //phase shift in radians
38 disp(" theta_A=" +string(theta_A)+" degree") // Phase
    shift for feedback gain Af3
39 theta_pm=180-abs(theta_A) // formulae phase margin
40 disp(" theta_pm=180-abs(theta_A)=" +string(theta_pm)+"
    degree") // Phase Margin for feedback gain Af1
41 disp(" Amplifier unstable") // Since phase margin is
    (-)ive
42 //NOTE: Correct ans for part(i) phase margin ,
    theta_pm=71.882476 degree but in book given as
    71.86 degree
43 // 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

```

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

```

```

12 disp("RSf= "+string(RSf)+ " ohm") //Source
    resistance
13 R1=RSf
14 RF=AVf*(R1)
15 disp("RF=AVf*(R1)= "+string(RF)+ " ohm") //Feedback
    resistance
16 VS=30
17 disp("VS= "+string(VS)+" volts") // Peak-peak output
    swing voltage
18 Vomax=0.5*(VS)
19 disp("Vomax=0.5*(VS)= -"+string(Vomax)+" , +" +string(
    Vomax)+" volts") // Maximum output voltage swing
    at negative and positive polarities respectively
20 Vsmax=Vomax/AVf
21 disp("Vsmax=Vomax/AVf= -"+string(Vsmax)+" , +" +string
    (Vsmax)+" volts") // Maximum output voltage
    without overload clipping at both polarities
22
23
24 //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

```

1 //Ex7-8
2 //refer fig. 7.6(b)
3 clc
4 R1=50*10^(3)
5 disp("R1= "+string(R1)+ " ohm") // resistance at
    input terminal of OP-AMP
6 RF=500*10^(3)
7 disp("RF= "+string(RF)+ " ohm") //Feedback
    resistance

```

```

8 VS=1
9 disp("VS= "+string(VS)+" volts") // Peak-peak output
    swing voltage
10 disp(" part ( i )")
11 disp("A = infinite")// voltage gain
12 Vo1=-(RF/R1) //Output voltage when gain , A=infinite
13 disp(" Vo1=-(RF/R1)= "+string(Vo1)+" volts")
14 disp(" part ( ii )")
15 A=50000
16 disp("A="+string(A)) // gain of OP-AMP
17 B=R1/(R1+RF)
18 disp("B=R1/(R1+RF)= "+string(B)) //Feedback factor
19 Vo2=-[(RF)*(B*A)]/(R1*(1+A*B))
20 disp(" Vo2=-[(RF)*(B*A)]/(R1*(1+A*B))= "+string(Vo2)+
    " volts")// output voltage for A=50000
21 e=-[(Vo2-Vo1)*100]/Vo1
22 disp("%Error , e= [(Vo2-Vo1)*100]/Vo1="+string(e)+"% "
    )// calculation for percentage error in output
    voltage
23 disp(" part ( iii )")
24 e=0.01
25 disp("%Error , e="+string(e)+"% ")//Given percentage
    error in output voltage
26 Vo3=-(Vo1-(e*Vo1/100))
27 disp(" Vo3=Vo1-(e*Vo1/100)= "+string(Vo3)+" volts")//
    output voltage for error 0.01%
28 x=Vo3*(R1/RF)
29 A=(x)/[B*(1-x)] //using formulae  $V_o=-\frac{R_F}{R_1} * \frac{(B*A)}{1+A*B}$ 
30 disp("A=(Vo*R1)/[B*RF*(1-(Vo*RF/R1))]="+string(A))
    // New Required gain for error less than 0.01%
31
32 // while solving the problem I have used 'e' for the
    error as no varriable is given for the same in
    textbook by author
33 // 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

```
1 //Ex7_9
2 clc
3 AV=100000
4 disp("AV="+string(AV)) // Voltage gain
5 Ri=10*10^(3)
6 disp("Ri= "+string(Ri)+ " ohm") //Input resistance
   of OP-AMP
7 Ro=10
8 disp("Ro= "+string(Ro)+ " ohm") //Output resistance
9 Rs=10*10^(6)
10 disp("Rs= "+string(Rs)+ " ohm") //Source resistance
11 RL=1*10^(3)
12 disp("RL= "+string(RL)+ " ohm") //Load resistance
13 B=(Rs-Ri)/(AV*Ri)
14 disp("B=(Rs-Ri)/(AV*Ri)= "+string(B)) //Feedback
   factor
15 AVf=AV/(1+B*AV)
16 disp("AVf=AV/(1+B*AV)="+string(AVf)) // Overall
   Voltage gain with feedback
17 Rof=Ro/(1+B*AV)
18 disp("Rof=Ro/(1+B*AV) =" +string(Rof)+ " ohm") //
   output resistance with feedback
19 Rif=Ri*(1+B*AV)
20 disp("Rif=Ri/(1+B*AV) =" +string(Rif)+ " ohm") //
   Input resistance with feedback
21 Ap=(AVf^2)*(Rif/RL)
22 disp("Ap=(AVf^2)*(Rif/RL)="+string(Ap)) // Overall
   Power gain
23 AP=10*log10(Ap)
24 disp("AP=10*log10(Ap)="+string(AP)+" dB") // Overall
```

Power gain in dB

Scilab code Exa 7.10 Find different parameters for Pierce oscillator

```
1 //Ex7_10
2 clc
3 gm=10*10^(-3)
4 disp("gm = "+string(gm)+" A/V") // transconductance
5 Cgs=5*10^(-12)
6 disp("Cgs= "+string(Cgs)+ " farad") // capacitance
   between gate-source
7 Cds=1*10^(-12)
8 disp("Cds= "+string(Cds)+ " farad") // capacitance
   between drain-source
9 rd=50*10^(3)
10 disp("rd= "+string(rd)+ " ohm") //Drain resistance
11 RG=10*10^(6)
12 disp("RG= "+string(RG)+ " ohm") //Gate resistance
13 Rse=1*10^(3)
14 disp("Rse= "+string(Rse)+ " ohm") //Gate resistance
15 L=0.5
16 disp("L= "+string(L)+ " H") //Inductance
17 C2=0.05*10^(-12)
18 disp("C2= "+string(C2)+ " farad") // Crystal
   parameter
19 C1=1*10^(-12)
20 disp("C1= "+string(C1)+ " farad") // Crystal
   parameter
21 disp("part(i)")
22 x=C1+[(Cds*Cgs)/(Cds+Cgs)]
23 CT=1/[(1/C2)+(1/x)]
24 disp("CT= "+string(CT)+ " farad") // Equivalent
   series-resonating capacitance
25 disp("part(ii)")
26 fo=sqrt(2)/[2*%pi*sqrt(L*CT)]
```

```

27 disp(" fo= sqrt(2)/[2*%pi*sqrt(L*CT)]="+string(fo)+"
      Hz")// frequency of oscillations
28 disp(" part( iii)")
29 z=sqrt((L*C1*C2)/(C1+C2))
30 fp=1/[2*%pi*z]
31 disp(" fp= "+string(fp)+" Hz")// parallel-resonant
      frequency
32 p=sqrt(L*C2)
33 fs=1/[2*%pi*p]
34 disp(" fs= "+string(fs)+" Hz")// series-resonant
      frequency
35 Q=[sqrt(L/C2)]/(Rse)
36 disp("Q=[sqrt(L/C2)]/(Rse)= "+string(Q))// Quality
      factor
37 disp(" part( iv)")
38 AB=gm*rd*(Cds/Cgs)
39 disp("AB=gm*rd*(Cds/Cgs)= "+string(AB))//Loop gain
40 T_bias=RG*(Cgs+Cds)
41 disp(" T_bias=RG*(Cgs+Cds)= "+string(T_bias)+" s")//
      Bias Time-Constant
42 T_r = 1/(2*%pi*fo)
43 disp(" T_r =1/(2*%pi*fo)= "+string(T_r)+" s")//
      resonant Time-Constant for 'fo'
44 disp("for proper operation T_bias >> T_r")
45
46
47 // in part (ii)... value calculated for series
      resonant frequency 'fo' is wrong in textbook.
48 // NOTE: in part(iii)... there is a misprint in the
      calculated value of Quality factor 'Q' in the
      textbook.
49 //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

```
1 //Ex8_1
2 clc
3 Amin=8000
4 disp("Amin="+string(Amin)) // Minimum gain of OP-AMP
5 Amax=64000
6 disp("Amax="+string(Amax)) // Maximum gain
7 disp(" part (i)")
8 delta_Af=0.01
9 disp("delta_Af="+string(delta_Af)) // Change in
  overall feedback gain
10 delta_A=(Amax-Amin)/Amin
11 disp("delta_A= (Amax-Amin)/Amin = "+string(delta_A))
  // Change in open loop gain
12 Sg = delta_Af/delta_A
13 B = (1/Sg - 1)/Amax
14 disp("Sg = delta_Af/delta_A = "+string(Sg))//
  desensitivity factor
15 disp(" B = (1/Sg - 1)/Amax = "+string(B))//feedback
  factor
16 disp(" part (ii)")
```

```

17 Af_min = Amin/(1+B*Amin)//minimum change in overall
    feedBack gain
18 Af_max = Amax/(1+B*Amax)//maximum change in overall
    feedBack gain
19 disp(" Af_min = Amin/(1+B*Amin) = "+string(Af_min))
20 disp(" Af_max = Amax/(1+B*Amax) = "+string(Af_max))
21 disp(" variation in Af = "+string(Af_max/Af_min))//
    variation in Af with feedBack factor 'B'
22
23
24 // for above problem author has divided question in
    two parts but during solution has written 3 parts
    .
25 // part (i) and part (ii) combinedly equivlent to
    part (i)
26 // part (iii) is equivalent to part (ii)

```

Scilab code Exa 8.2 Design an inverting op amp

```

1 //Ex8_2
2 clc
3 Avf=-100
4 disp(" Avf=" +string(Avf)) // Voltage gain
5 Rif=1
6 disp(" Rif= " +string(Rif)+ " ohm") //Input
    resistance of OP-AMP
7 R1=Rif
8 RF=-R1*Avf // using formulae Vo=(-RF/R1)*Vi
9 disp(" RF= -R1*Avf=" +string(RF)+ " ohm") //Feedback
    resistance of OP-AMP
10 // NOTE:Error in value of RF since they have given
    value of Rif=1ohm but calculated RF by using Rif
    =1 Kilo ohm
11 // 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

```
1 //Ex8_3
2 clc
3 R11=1*10^(3)
4 disp("R11= "+string(R11)+ " ohm") // resistance at
   input terminal of OP-AMP Adder
5 RF=100*10^(3)
6 disp("RF= "+string(RF)+ " ohm") //Feedback
   resistance
7 R12=10*10^(3)
8 disp("R12= "+string(R12)+ " ohm") // resistance at
   input terminal of OP-AMP Adder
9 R13=100*10^(3)
10 disp("R13= "+string(R13)+ " ohm") // resistance at
   input terminal of OP-AMP Adder
11 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
12
13 // for average value of input signal
14 n = 3// given inputs are '3'
15 R11 = n*RF
16 R12 = n*RF
17 R13 = n*RF
18 disp("vo = -( "+string(RF/R11)+ " vs1 + "+string(RF/R12)
   + " vs2 + "+string(RF/R13)+ " vs3)") // output
   voltage of opamp adder
19
20
21 // note : the output voltage of inverting adder is
   negative
22 // 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

```
1 //Ex8_4
2 clc
3 Ir=10*10^(-3)
4 disp("Ir = "+string(Ir)+" ampere/lumen of radiant
      energy ") //photodiode Reverse saturation current
      for constant reverse bias VR
5 RF=10*10^(3)
6 disp("RF= "+string(RF)+ " ohm") //Feedback
      resistance
7 E=1*10^(-2)
8 disp("E = "+string(E)+" lumens")// radiant energy
9 IR=Ir*E
10 disp("IR =Ir *E= "+string(IR)+" ampere") // Reverse
      saturation current
11 Vo=IR*RF
12 disp("Vo=IR*RF= "+string(Vo)+" volts") // output
      voltage
13 s=E/Vo
14 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

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

```

5 RF=100*10^(3)
6 disp("RF= "+string(RF)+ " ohm") //Feedback
    resistance
7 RM=10*10^(3)
8 disp("RM= "+string(RM)+ " ohm") // D.C Ammeter
    internal resistance
9 is=10*10^(-6)
10 disp("is = "+string(is)+" ampere") // Source
    current
11 vo=is*RF
12 disp("vo=is*RF= "+string(vo)+" volts") // output
    voltage
13 S=vo/is
14 disp("S=vo/is= "+string(S)+" V/A") // Sensitivity of
    Ammeter
15 Rif=RF/(1+Av)
16 disp("Rif=RF/(1+Av)= "+string(Rif)+ " ohm") //Input
    resistance of OP-AMP
17 im=100*10^(-6)
18 disp("im = "+string(im)+" ampere") // Meter Full-
    Scale deflection current
19 RF=(im*RM)/is
20 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

```

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

```



```

7 RF=R1*[10^(Av/20)-1] // Using formulae Av=20*log(1+
  RF/R1)
8 disp("RF=R1*[10^(Av/20)-1]= "+string(RF)+ " ohm")
  // Calculated Feedback resistance
9 //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

```

1 //Ex8_7
2 clc
3 if=100*10^(-6)
4 disp("if = "+string(if)+" ampere") //Full-Scale
  deflection current
5 Av=1*10^(5)
6 disp("Av= "+string(Av)) //Voltage gain
7 vs=10*10^(-3)
8 disp("vs= "+string(vs)+" volts") // Input voltage
9 RM=100
10 disp("RM= "+string(RM)+ " ohm") // Moving coil
  Ammeter internal resistance
11 Ri=10*10^(3)
12 disp("Ri= "+string(Ri)+ " ohm") //Input resistance
  of OP-AMP
13 R1=vs/if
14 disp("R1=vs/if= "+string(R1)+ " ohm") // Resistor
  at input side of OP-AMP in Voltage-to-Current
  converter
15 Avf=1+(RM/R1) // formulae using Avf=1+(RF/R1)=1+(RM/
  R1)// since RF=RM
16 disp("Avf=1+(RM/R1)="+string(Avf)) // Overall
  Voltage gain
17 Rif=Ri*(Av/Avf)
18 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 V_o

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

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

```

8 disp("fOH= "+string(fOH)+" Hz")//Cut off frequency
9 R1=10*10^(3)// Choosing value of R1 same as in book
10 disp("R1= "+string(R1)+ " ohm") // Resistor at
    input side of (OP-AMP) filter
11 RF=R1*(AM-1)
12 disp("RF=R1*(AM-1)="+string(RF)+ " ohm") //Feedback
    resistance
13 C=0.1*10^(-6) // Choosing value of capacitor same a
    in book
14 disp("C="+string(C)+" farad")
15 R=1/(2*%pi*fOH*C)// Using formulae  $\omega_{OH}=1/C*R$  and  $\omega_{OH}
    =(2*%pi*fOH)$ 
16 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

```

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

```

```

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

```

Scilab code Exa 8.11 Determine filter response for state variable filter

```

1 //Ex8_11
2 clc
3 R=10*10^(3)
4 disp("R= " +string(R)+ " ohm") // resistance
5 R1=10*10^(3)
6 disp("R1= " +string(R1)+ " ohm") // resistance
7 C=0.01*10^(-6) // value of capacitor
8 disp("C=" +string(C)+ " farad")
9 R1_ratio_K=2.5*10^(3)
10 disp("R1_ratio_K= " +string(R1_ratio_K)+ " ohm") //
    resistance
11 R2=5*10^(3)
12 disp("R= " +string(R)+ " ohm") // resistance
13 alpha_R2=250
14 disp("alpha_R2= " +string(alpha_R2)+ " ohm") //
    resistance
15 alpha=alpha_R2/R2
16 disp("alpha=alpha_R2/R2= " +string(alpha)) //
    Damping factor
17 Q=1/alpha
18 disp("Q= 1/alpha=" +string(Q)) // Quality factor
19 omega_o=1/(R*C)
20 disp("omega_o=1/(R*C)= " +string(omega_o)+ " radian")
    // centre angular frequency
21 BW=omega_o/Q
22 disp("Bandwidth=omega_o/Q= " +string(BW)+ " radian") //
    Upper cut-off frequency or 3-dB bandwidth
23 K=R1/(R1_ratio_K) // Pass band gain for LPF and HPF
    of state variable filter

```

```

24 disp("K=R1/(R1_ratio_K)= "+string(K))
25 Gm=K/alpha// Pass band gain of state variable filter
26 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

```

1 //Ex8_12
2 clc
3 IB=0.5*10^(-6)
4 disp("IB = "+string(IB)+" ampere") //Input bias
  current
5 Iio=0.05*10^(-6)
6 disp("Iio = "+string(Iio)+" ampere") //Input offset
  current
7 Vio=1*10^(-3)
8 disp("Vio= "+string(Vio)+" volts") //Input offset
  voltage
9 R1=10*10^(3)
10 disp("R1= "+string(R1)+ " ohm") // resistance
11 RF=500*10^(3)
12 disp("RF= "+string(RF)+ " ohm") //Feedback
  resistance
13 Vos1=Vio*(1+RF/R1)
14 disp("Vos1=Vio*(1+RF/R1)="+string(Vos1)+" volts") //
  output offset voltage due to input offset voltage
15 Vos2=IB*RF
16 disp("Vos2=IB*RF="+string(Vos2)+" volts") //output
  offset voltage due to Input bias current
17 Vos=Vos1+Vos2
18 disp("Vos=Vos1+Vos2="+string(Vos)+" volts") //total
  output offset voltage
19 R2=(R1*RF)/(R1+RF)

```

```

20 disp("R2=(R1*RF)/(R1+RF)= "+string(R2)+ " ohm") //
    resistance to balance IB effect
21 Vos2=Iio*RF
22 disp("Vos2=Iio*RF="+string(Vos2)+" volts") //
    Reduced output offset voltage due to Input offset
    current
23 Vos=Vos1+Vos2
24 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

```

1 //Ex8_13
2 clc
3 Iio=0.1*10^(-9)
4 disp("Iio = "+string(Iio)+" ampere/degree _celsius")
    //Input offset current
5 Vio=10*10^(-6)
6 disp("Vio= "+string(Vio)+" volt/degree _celsius") //
    Input offset voltage
7 Vs=10*10^(-3)
8 disp("Vs= "+string(Vs)+" volts") //Input voltage
9 R1=10*10^(3)
10 disp("R1= "+string(R1)+ " ohm") // resistance
11 RF=100*10^(3)
12 disp("RF= "+string(RF)+ " ohm") //Feedback
    resistance
13 disp("part(i)")
14 R2=(R1*RF)/(R1+RF)// R1 in parallel with RF
15 disp("R2=(R1*RF)/(R1+RF)= "+string(R2)+ " ohm") //
    resistance to balance IB i.e offset effect
16 disp("part(ii)")
17 delta_T=75-25
18 disp("delta_T=75-25 = "+string(delta_T)+")

```

```

    degree_celsius") //Temperature change
19 delta_Vo=[(Vio*delta_T)*(1+RF/R1)]+(Iio*delta_T*RF)
20 disp(" delta_Vo=[(Vio*delta_T)*(1+RF/R1)]+( Iio*
    delta_T*RF)= "+string(delta_Vo)+" volts") //
    Output voltage drift
21 disp(" part( iii)")
22 Vo=(-RF/R1)*Vs
23 disp("Vo=(-RF/R1)*Vs= "+string(Vo)+" volts") //
    Inverting OP-AMP output voltage
24 e=(delta_Vo/Vo)*100
25 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

```

1 //Ex8_14
2 clc
3 Iio=0.1*10^(-9)
4 disp(" Iio = "+string(Iio)+" ampere") //Input offset
    current
5 VCC=15
6 disp("VCC= "+string(VCC)+" volts") // voltage
    supply
7 PSRR=150*10^(-6)
8 disp("PSRR= "+string(PSRR)+" volts/V")// Power
    supply rejection ratio
9 Vio=10*10^(-6)
10 disp(" Vio= "+string(Vio)+" volts") //Input offset
    voltage
11 R1=10*10^(3)
12 disp("R1= "+string(R1)+ " ohm") // resistance
13 RF=100*10^(3)
14 disp("RF= "+string(RF)+ " ohm") //Feedback
    resistance

```

```

15 delta_T=75-25
16 disp(" delta_T=75-25 = "+string(delta_T)+" celsius")
    //Temperature change
17 delta_Vo=[(Vio*delta_T)*(1+RF/R1)]+(Iio*delta_T*RF)
18 disp(" delta_Vo=[(Vio*delta_T)*(1+RF/R1)]+( Iio*
    delta_T*RF)= "+string(delta_Vo)+" volts") //
    Output voltage drift
19 delta_Vio1=(delta_Vo)*(R1/RF)
20 disp(" delta_Vio1=(delta_Vo)*(R1/RF)= "+string(
    delta_Vio1)+" volts") // voltage change at Input
    for voltage drift found
21 delta_Vio2=(delta_Vio1)*(1/10)
22 disp(" delta_Vio2=(delta_Vio1)*(1/10)= "+string(
    delta_Vio2)+" volts") // change in Vio due to
    PSRR
23 p=[(delta_Vio2)/(VCC*PSRR)]*100
24 disp(" power supply regulation=[(delta_Vio2)/(VCC*
    PSRR)]*100 =" +string(p)+ " percent")// power
    supply regulation requirement
25
26 //delta_Vio1 corresponds to voltage change at Input
    for voltage drift found
27 //delta_Vio2 corresponds voltage change at input due
    to PSRR

```

Scilab code Exa 8.15 calculate full power bandwidth

```

1 //Ex8_15
2 clc
3 SR=0.65
4 disp("SR= "+string(SR)+" volts/microsecond")// Slew
    rate of OP-AMP
5 disp(" part (i)")
6 Vm=5
7 disp("Vm= "+string(Vm)+" volts") // Output peak

```



```

    voltage1
8  fsm=SR/[10^(-6)*(2*%pi*Vm)] // using formulae SR=2*
    pi*fsm*Vm
9  disp("fsm=SR/[10^(-6)*(2*%pi*Vm)] = "+string(fsm)+"
    Hz")// // Full power bandwidth for Output peak
    voltage Vm=5V
10 disp("part(ii)")
11 Vm=1
12 disp("Vm= "+string(Vm)+" volts") // Output peak
    voltage2
13 fsm=SR/[10^(-6)*(2*%pi*Vm)] // using formulae SR
    =2*pi*fsm*Vm
14 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

```
1 //Ex9_1
2 clc
3 VCC=5
4 disp("VCC= "+string(VCC)+" volts") // voltage
   supply
5 RB=10*10^(3)
6 disp("RB= "+string(RB)+ " ohm") // Base-resistance
7 RL=1*10^(3)
8 disp("RL= "+string(RL)+ " ohm") // Load resistance
9 VCS=0.2
10 disp("VCS= "+string(VCS)+" volts") // collector
   saturated voltage
11 VBS=0.8
12 disp("VBS= "+string(VBS)+" volts") // Base voltage
   at saturation
13 V_gamma=0.6
14 disp("V_gamma= "+string(V_gamma)+" volts") //
   Threshold or cut-in voltage
15 ICS=(VCC-VCS)/RL
```

```

16 disp("ICS = (VCC-VCS)/RL="+string(ICS)+" ampere") //
    Saturation collector current of transistor T1
17 vi=5
18 disp("vi= "+string(vi)+" volts") // Input voltage
19 IBS=(vi-VBS)/RB
20 disp("IBS=(vi-VBS)/RB="+string(IBS)+" ampere") //
    Forward base drive required to sustain ICS
21 Beta_Fmin=ICS/IBS
22 disp("Beta_Fmin=ICS/IBS= "+string(Beta_Fmin)) //
    Common-emitter current gain
23
24 //NOTE: Correct formulae for ICS=(VCC-VCS)/RL
25 // 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 V_x for OR logic gate

```

1 //Ex9_2 Refer fig .9.3(e)
2 clc
3 VD=0.7
4 disp("VD= "+string(VD)+" V") // Diode voltage drop
    in conduction mode
5
6 disp("part(i)") // part(i) of question
7 vA=0
8 disp("vA= "+string(vA)+" V") // Input voltage1 of
    diode OR logic gate
9 vB=0
10 disp("vB= "+string(vB)+" V") // Input voltage2 of
    diode OR logic gate
11 vX=0 // Since both input voltages vA=vB=0V
12 disp("vX="+string(vX)+" V") // Output voltage of
    diode OR logic gate for part(i)
13

```

```

14 disp(" part ( ii )") // part ( ii ) of question
15 vA=0
16 disp(" vA= "+string(vA)+" V") // Input voltage1 of
    diode OR logic gate
17 vB=5
18 disp(" vB= "+string(vB)+" V") // Input voltage2 of
    diode OR logic gate for SECOND CASE: when vA=0V
    and vB=5V
19 vX=vB-VD
20 disp(" vX=vB-VD= "+string(vX)+" V") // Output
    voltage of diode OR logic gate for SECOND CASE
21
22 disp(" part ( iii )") // part ( iii ) of question
23 vA=5
24 disp(" vA= "+string(vA)+" V") // Input voltage1 of
    diode OR logic gate for THIRD CASE when vA=5V
    and vB=0V
25 vB=0
26 disp(" vB= "+string(vB)+" V") // Input voltage2 of
    diode OR logic gate
27 vX=vA-VD
28 disp(" vX=vA-VD= "+string(vX)+" V") // Output
    voltage of diode OR logic gate for THIRD CASE
29
30 disp(" part ( iv )") // part ( iv ) of question
31 vA=(+5)
32 disp(" vA= "+string(vA)+" V") // Input voltage1 of
    diode OR logic gate
33 vB=(+5)
34 disp(" vB= "+string(vB)+" V") // Input voltage2 of
    diode OR logic gate
35 vX=vA-VD // Since both diodes D1 and D2 are
    conducting
36 vX=vB-VD
37 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 V_x for diode AND logic gate

```
1 //Ex9_3 Refer fig .9.4(e)
2 // For AND logic gate
3 clc
4 VD=0.7
5 disp("VD= "+string(VD)+" V") // Diode voltage drop
   in conduction mode
6
7 disp(" part (i)")
8 vA=0
9 disp("vA= "+string(vA)+" V") // Input voltage of
   diode AND logic gate
10 vB=0
11 disp("vB= "+string(vB)+" V") // Input voltage2 of
   diode AND logic gate
12
13 vX=VD // Since both input voltages vA=vB=0V
14 disp("vX=VD="+string(vX)+" V") // Output voltage of
   diode AND logic gate for FIRST CASE: when vA=0V
   and vB=0V
15
16 disp(" part (ii)")
17 vA=0
18 disp("vA= "+string(vA)+" V") // Input voltage of
   diode AND logic gate
19 vB=5
20 disp("vB= "+string(vB)+" V") // Input voltage2 of
   diode AND logic gate for SECOND CASE: when vA=0V
   and vB=5V
21 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
22 disp("vX=VD "+string(vX)+" V")
```

```

23 //due to diode B which is conducting
24
25
26
27 disp(" part ( iii )")
28 vA=5
29 disp("vA= "+string(vA)+" V") // Input voltage for
    THIRD CASE when vA=5V and vB=0V
30 vB=0
31 disp("vB= "+string(vB)+" V") // Input voltage2 of
    diode AND logic gate
32 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
33 disp("vX= "+string(vX)+" V")
34
35 disp(" part ( iv )")
36 vA=5
37 disp("vA= "+string(vA)+" V") // Input voltage
    forfourth CASE when vA=5V and vB=5V
38 vB=5
39 disp("vB= "+string(vB)+" V") // Input voltage2 of
    diode AND logic gate for CASE: when vA=0V and vB
    =5V
40 vX=vA // Since both diodes D1 and D2 are Non-
    conducting , so no voltage drop across 'R'(resistor
    )
41 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

```

1 //Ex9_6
2 clc

```

```

3 VIL=0.6
4 disp("VIL= "+string(VIL)+" V") // Minimum input
    voltage level for which output is maximum
5 VIH=0.75
6 disp("VIH= "+string(VIH)+" V") // Maximum input
    voltage level for which output is minimum
7 VOL=0.2
8 disp("VOL= "+string(VOL)+" V") // Minimum output
    voltage level for maximum input level
9 VOH=1
10 disp("VOH= "+string(VOH)+" V") // Maximum output
    voltage level for minimum input level
11 NML=VIL-VOL
12 disp("NML=VIL-VOL= "+string(NML)+" V") // Low level
    noise immunities
13 NMH=VOH-VIH
14 disp("NMH=VOH-VIH= "+string(NMH)+" V") // High level
    noise immunities

```

Scilab code Exa 9.7 Find fan out of TTL

```

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

```

```

11 Fan_out=abs((IOH/IIH))
12 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

```

1 //Ex9_8
2 clc
3 VIL=0.8
4 disp("VIL= "+string(VIL)+" V") // Minimum input
      voltage level for which output is maximum
5 VIH=2
6 disp("VIH= "+string(VIH)+" V") // Maximum input
      voltage level for which output is minimum
7 VOL=0.4
8 disp("VOL= "+string(VOL)+" V") // Minimum output
      voltage level for maximum input level
9 VOH=2.4
10 disp("VOH= "+string(VOH)+" V") // Maximum output
      voltage level for minimum input level
11 NML=VIL-VOL
12 disp("NML=VIL-VOL= "+string(NML)+" V") // Low level
      noise immunities
13 NMH=VOH-VIH
14 disp("NMH=VOH-VIH= "+string(NMH)+" V") // High level
      noise immunities

```

Scilab code Exa 9.9 Find low and high level noise immunities

```

1 //Ex9_9
2 clc
3 VIL=1

```



```

4 disp("VIL= "+string(VIL)+" V") // Minimum input
    voltage level for which output is maximum
5 VIH=4
6 disp("VIH= "+string(VIH)+" V") // Maximum input
    voltage level for which output is minimum
7 VOL=0.5
8 disp("VOL= "+string(VOL)+" V") // Minimum output
    voltage level for maximum input level
9 VOH=4.5
10 disp("VOH= "+string(VOH)+" V") // Maximum output
    voltage level for minimum input level
11 NML=VIL-VOL
12 disp("NML=VIL-VOL= "+string(NML)+" V") // Low level
    noise immunities
13 NMH=VOH-VIH
14 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

```

1 //Ex9_10 Refer Fig.9.17(a)
2 clc
3 V_gamma=0.6
4 disp("V_gamma= "+string(V_gamma)+" volts") //
    Threshold voltage
5 VEE=-5.2
6 disp("VEE= "+string(VEE)+" volts") // voltage
    supply
7 VBE3=0.7
8 VBE4=VBE3
9 VBE5=VBE3
10 disp("VBE3=VBE4=VBE5 "+string(VBE3)+" volts") //base
    -emitter voltage
11 RE=779

```

```

12 disp("RE= "+string(RE)+ " ohm") // Emitter-
    resistance
13 RL2=220
14 disp("RL2= "+string(RL2)+ " ohm") // Load
    resistance
15 RL3=245
16 disp("RL3= "+string(RL3)+ " ohm") // Load
    resistance
17 VREF=-1.29
18 disp("VREF= "+string(VREF)+" volts") // Reference-
    voltage
19 V_1=-0.7
20 disp("V(1)= "+string(V_1)+" volts") // Acceptable
    voltage for high logic
21 V_0=-1.7
22 disp("V(0)= "+string(V_0)+" volts") // Acceptable
    voltage for low logic
23
24 disp(" part (i) ") // part(i) of question
25 VE=VREF-VBE3
26 disp("VE=VREF-VBE3= "+string(VE)+" volts") //
    Emitter- voltage
27 IE=(VEE-VE)/RE
28 disp("IE=(VEE-VE)/RE= "+string(IE)+" A") //Emitter-
    Current
29 IC3=IE// since IC=IE neglecting IB
30 disp("IC3=IE= "+string(IE)+" A") //Collector-
    Current
31 vC3=IC3*RL3
32 disp("vC3=IC3*RL3= "+string(vC3)+" volts") //
    Collector- voltage
33 vY=vC3-VBE5
34 disp("vY=vC3-VBE5= "+string(vY)+" volts") // Emitter
    follower output voltage for vB=V(0)
35 vC2=0
36 vX=vC2-VBE4
37 disp("vX=vC2-VBE4= "+string(vX)+" volts") // Emitter
    follower output voltage for vB=V(0)

```

```

38 VBEr=(V_0)-VE
39 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
40 disp(" Transistor T1 and T2 off since VBEr < V_gamma"
      ) // Since VBEr < V_gamma hence T1 and T2 off
41
42 disp(" part(ii)")// part(ii) of question
43 IC2=IE
44 VBE=0.7
45 vB=V_1
46 IC3=0
47 VC3=0
48 vY=VC3-VBE5
49 disp("vY=VC3-VBE= "+string(vY)+" volts")// Emitter
      follower output voltage for SECOND CASE for vB=V
      (1)
50 VE=vB-VBE
51 disp("VE=vB-VBE= "+string(VE)+" volts") // Emitter-
      voltage
52 VBE3=VREF-VE
53 disp("VBE3=VREF-VE= "+string(VBE3)+" volts")//Base-
      Emitter junction voltage
54 disp("VBE3 is smaller than V_gamma,hence T3 is off")
55 IC2=(VEE-VE)/RE
56 disp("IC2=(VEE-VE)/RE= "+string(IC2)+" A") //
      Collector- Current for T2(transistor)
57 vC2=IC2*RL2
58 disp("vC2=IC2*RL2= "+string(vC2)+" volts") //
      Collector- voltage for T2
59 vX=vC2-VBE4
60 disp("vX=vC2-VBE4= "+string(vX)+" volts") // Emitter
      follower output voltage for vB=V(1)
61
62 disp(" part(iii)")// part(iii) of question
63 VE3=-1.99
64 disp("VE3=VE= "+string(VE3)+" volts") // Transistor

```

```

        T3 Emitter- voltage ,when T3 is conducting
65 VB3=VREF
66 disp("VB3=VREF= "+string(VB3)+" volts") // Base-
        voltage when T3 is conducting
67 IC3=(VEE-VE3)/RE// Collector current for T3
        neglecting IB
68 disp("IC3=(VEE-VE3)/RE= "+string(IC3)+" A") //
        Collector- Current
69 VC3=IC3*RL3
70 disp("VC3=IC3*RL3= "+string(VC3)+" volts") //
        Collector- voltage when T3 is conducting
71 VCB3=VC3-VB3
72 disp("VCB3=VC3-VB3= "+string(VCB3)+" volts") // Base
        - voltage when T3 is conducting
73 //All parameters have appropriate signs for npn BJT
        hence BJT in active region not in saturation in
        which VCB will have a (-)value
74 disp("All parameters have appropriate signs for npn
        BJT hence BJT in active region ")
75
76
77 // NOTE: Author ha not used any symbol for Base-
        Emitter junction reverse voltage But I have
        used 'VBEr' for it.
78 // ERROR :sign of IE is given wrong in the book in
        part(i) and sign of IC2 in part(ii)
79 // 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
80 // 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
81 // 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

```
1 //Ex9_11
2 clc
3 VIL=-1.475
4 disp("VIL= "+string(VIL)+" V") // Minimum input
   voltage level for which output is maximum
5 VIH=-1.105
6 disp("VIH= "+string(VIH)+" V") // Maximum input
   voltage level for which output is minimum
7 VOL=-1.63
8 disp("VOL= "+string(VOL)+" V") // Minimum output
   voltage level for maximum input level
9 VOH=-0.98
10 disp("VOH= "+string(VOH)+" V") // Maximum output
   voltage level for minimum input level
11 NML=VIL-VOL
12 disp("NML=VIL-VOL= "+string(NML)+" V") // Low level
   noise immunities
13 NMH=VOH-VIH
14 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

```
1 //Ex10_1
2 clc
3 x=72; //given value in Decimal
4 disp("Decimal number="+string(x))
5 str=dec2bin(x)
6 disp("Equivalent Binary number="+string(str)) //Binary
   value
```

Scilab code Exa 10.2 Convert binary to decimal

```
1 //Ex10_2
2 clc
3 x='1001000'; //Binary value
4 disp("Binary number="+string(x))
5 str=bin2dec(x)
6 disp("Equivalent Decimal number="+string(str)) //
   decimal value
```

Scilab code Exa 10.3 Convert binary to octal

```
1 //Ex10_3
2 clc
3 b= '1001000 ' ;
4 disp(" Binary number="+string(b))//Binary value
5 d=bin2dec(b)// Binary to decimal value
6 o=dec2oct(d)// Decimal to octal
7 disp(" Equivalent Octal number="+string(o))
```

Scilab code Exa 10.4 Convert decimal to octal

```
1 //Ex10_4
2 clc
3 x=72;
4 disp(" Decimal number="+string(x))//Decimal value
5 str=dec2oct(x)//decimal to octal
6 disp(" Equivalent Octal number="+string(str))
```

Scilab code Exa 10.5 Convert octal to binary

```
1 //Ex10_5
2 clc
3 x= '110 ' ;
4 disp(" Octal number="+string(x))//octal value
5 y=oct2dec(x)// octal to decimal
6 str=dec2bin(y)//decimal to binary
7 disp(" Equivalent Binary number="+string(str))
```

Scilab code Exa 10.6 Convert octal to decimal

```
1 //Ex10_6
2 clc
3 x= '110';
4 disp(" Octal number="+string(x))// octal value
5 str=oct2dec(x)//octal to decimal
6 disp(" Equivalent Decimal number="+string(str))
```

Scilab code Exa 10.7 Convert decimal to hexadecimal

```
1 //Ex10_7
2 clc
3 x=72;
4 disp(" Decimal number="+string(x))//decimal value
5 str=dec2hex(x)// decimal to hexadecimal
6 disp(" Equivalent Hexadecimal number="+string(str))
```

Scilab code Exa 10.8 Convert hexadecimal to decimal

```
1 //Ex10_8
2 clc
3 h= '48';
4 disp(" Hexadecimal number="+string(h))// value in
   hexadecimal
5 d=hex2dec(h)//hexadecimal to decimal
6 disp(" Equivalent Decimal number="+string(d))
```

Scilab code Exa 10.9 Convert hexadecimal to binary

```
1 //Ex10_9
2 clc
3 h= '48' ;
4 disp(" Hexadecimal number="+string(h))//hexadecimal
5 d=hex2dec(h)// converting hexadecimal to decimal
6 str=dec2bin(d)// converting decimal to binary
7 disp(" Equivalent Binary number="+string(str))
```

Scilab code Exa 10.10 convert binary to hexadecimal

```
1 //Ex10_10
2 clc
3 x= '1001000' ;
4 disp(" Binary number="+string(x))//binary value
5 d=bin2dec(x)//binary to decimal
6 h=dec2hex(d)//decimal to hexa decimal
7 disp(" Equivalent hexadecimal number="+string(h))
```

Scilab code Exa 10.11 Convert decimal to BCD

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

```
10 str1=dec2bin(z,4)// for 4 bit binary of MSB digit i.
    e 7 here
11 str2=dec2bin(y,4)//for 4 bit binary of LSB digit i.e
    2
12 disp(" Binary number of 7 =" +string(str1))
13 disp(" Binary number of 2 =" +string(str2))
14 disp(" Equivalent 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

```
1 //Ex11_1
2 clc
3 tsu=20*10(-9)
4 disp("tsu= "+string(tsu)+" seconds") // Input set-up
   time of second flip flop
5 tpd=30*10(-9)
6 disp("tpd= "+string(tpd)+" seconds") // Input set-up
   time of first flip flop
7 Tmin=tpd+tsu
8 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
9 fCkmax=1/Tmin // formulae
10 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

```
1 //Ex11_4
2 clc
3 tphL=40*10^(-9)
4 disp("tphL= "+string(tphL)+" seconds") // Time taken
   from Clear to output
5 n=3
6 disp("n= "+string(n))// Number of bits in counter i.
   e no. of flip-flops used
7 fmax=1/(n*tphL) // Using formulae fmax<= 1/(n*tphL)
8 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

```
1 //Ex11_6
2 clc
3 fs=2*10^(3)
4 disp("fs= "+string(fs)+" Hz")// sine wave input
   signal frequency
5 fB=1*10^(6)
6 disp("fB= "+string(fB)+" Hz")// input Time-Base
   clock frequency
7
8 disp("part(i)")// part(i) of question
9 fb=fB/(10^5)
10 disp("fb= fB/(10^5)=" +string(fb)+" Hz")// Time-Base
   frequency for 5 decade counter
11 delta_t=1/fb
12 disp("delta_t=1/fb= "+string(delta_t)+" seconds") //
   Gate Time interval
13 DISP1=fs*delta_t
14 disp("fs*delta_t= "+string(DISP1))// Display
```

```

    indication for 5 decade counter
15 disp(" Display indication=0200")// Display indication
    as 4-bit
16
17 disp(" part(ii)")// part(ii) of question
18 fb=fB/(10^3)
19 disp(" fb=fB/(10^3)= "+string(fb)+" Hz")// Time-Base
    frequency for 3 decade counter
20 delta_t=1/fb
21 disp(" delta_t=1/fb= "+string(delta_t)+" seconds") //
    Gate Time interval for 3 decade counter
22 DISP2=fs*delta_t
23 disp(" fs*delta_t= "+string(DISP2))// Display
    indication for 3 decade counter
24 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

```
1 //Ex12_1
2 clc
3 VEE=15
4 disp("VEE= "+string(VEE)+" volts") // voltage
    supply
5 VCC=15
6 disp("VCC= "+string(VCC)+" volts") // voltage
    supply
7 VHI=+5
8 disp("VHI= "+string(VHI)+" volts") // output
    voltage upper limit
9 VLO=-5
10 disp("VLO= "+string(VLO)+" volts") // output
    voltage Lower limit
11 Vo=-VLO
12 IZmin=1*10(-3)
13 disp("IZmin= "+string(IZmin)+" A") // Zener diode
    current rating
14 SR=0.5*10(6)
```

```

15 disp("SR= "+string(SR)+" volts/seconds")//Slew rate
16 RB=100
17 disp("RB= "+string(RB)+ " ohm") // resistance
18 RA=10*10^(3)
19 disp("RA= "+string(RA)+ " ohm") // resistance
20 A = 5000
21 disp("A = "+string(A))//op-amp gain
22 VREF=1
23 disp("VREF= "+string(VREF)+" volts") // Reference-
    voltage
24 disp(" part ( i )")
25 RD=(VCC-Vo)/IZmin
26 disp("RD=(VCC-Vo)/IZmin= "+string(RD)+ " ohm") //
    Series dropping-resistance
27
28 disp(" part ( ii )")
29 t=(VHI-VLO)/SR
30 disp(" t=(VHI-VLO)/SR= "+string(t)+" seconds")// Time
    required to swing the output
31 tp=10*t
32 disp(" tp=(VHI-VLO)/SR= "+string(tp)+" seconds")//
    Pulse width
33 fmax=1/(2*tp)
34 disp(" fmax=1/(2*tp) = "+string(fmax)+" Hz")//
    Maximum frequency of operation of OP-AMP
    comparator
35 disp(" part ( iii )")
36 B=RB/(RA+RB)
37 disp(" B=RB/(RA+RB)= "+string(B))//Feedback factor
38 VLTP=(VLO*B)+[VREF*(RA/(RA+RB))]
39 disp(" VLTP=(VLO*B)+[VREF*(RA/(RA+RB))]= "+string(
    VLTP)+" volts") // Lower trigger point
40 VUTP=(VHI*B)+[VREF*(RA/(RA+RB))]
41 disp(" VUTP=(VHI*B)+[VREF*(RA/(RA+RB))]= "+string(
    VUTP)+" volts") // Upper trigger point
42 VH=VUTP-VLTP
43 disp(" VH=VUTP-VLTP= "+string(VH)+" volts") //
    Hysteresis voltage

```

Scilab code Exa 12.2 Calculate component values for astable multivibrator

```
1 //Ex12_2
2 clc
3 Vo=14
4 disp("Vo= "+string(Vo)+" volts") // output voltage
5 f=500
6 disp("f = "+string(f)+" Hz")//frequency
7 IB2=500*10^(-9)
8 disp("IB2= "+string(IB2)+" A") //base- Current
9 B=0.5
10 disp("B="+string(B))//Feedback factor
11 vf=B*Vo
12 disp("vf=B*Vo= "+string(vf)+" , -"+string(vf)+"
    volts") // Feedback voltage
13 IR=100*IB2// Taking IR as 100 times that of IB2
14 disp("IR=100*IB2= "+string(IR)+" A") // Current in
    RB resistor
15 RB=vf/IR
16 disp("RB=vf/IR= "+string(RB)+ " ohm") // resistance
17 RA=RB*((1/B)-1)// Using formulae B=RA/(RA+RB)
18 disp("RA=RB*((1/B)-1)= "+string(RA)+ " ohm") //
    resistance
19 RF=100*10^(3)//Choosing RF=100k
20 disp("RF= "+string(RF)+ " ohm") //Feedback
    resistance
21 C1=1/[2*RF*f*log(1+(2*RB/RA))]
22 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


```

1 //Ex12_3
2 clc
3 Vo=14
4 disp("Vo= "+string(Vo)+" volts") // output voltage
5 f=500
6 disp("f = "+string(f)+" Hz")//frequency
7 R2=10*10^(3)
8 disp("R2= "+string(R2)+ " ohm") // resistance
9 VTW=14
10 disp("VTW= "+string(VTW)+" volts") // Triangular
    peak-peak output voltage
11 T=1/f
12 C2=(Vo*T)/(2*VTW*R2)
13 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 V_{sw} for sweep generator

```

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

```

```

12 disp("Sweep rate=VI/(R*C)=" + string(S) + " V/s") //
    Sweep rate for sweep generator
13 VSW=TSW*S
14 disp("VSW=TSW*S= " + string(VSW) + " volts") // Sweep
    voltage amplitude
15
16
17 // 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

```

1 //Ex12_5
2 clc
3 VEE=15
4 disp("VEE= " + string(VEE) + " volts") // voltage
    supply
5 VCC=15
6 disp("VCC= " + string(VCC) + " volts") // voltage
    supply
7 R1=10*10^(3)
8 disp("R1= " + string(R1) + " ohm") // resistance
9 RF=20*10^(3)
10 disp("RF= " + string(RF) + " ohm") // Feedback
    resistance
11 RB1=3*10^(3)
12 disp("R1= " + string(R1) + " ohm") // resistance
13 RB2=RB1
14 RF1=1*10^(3)
15 disp("RF1= " + string(RF1) + " ohm") // Feedback
    resistance
16 RF2=RF1
17 Av=1*10^(3)

```

```

18 disp("Av= "+string(Av))
19 disp(" part ( i ) ")
20 VBR1= (VCC*RF1)/RB1
21 VBR2 = VBR1
22 disp("VBR1=VBR2=(VCC*RF1)/RB1= "+string(VBR1)+"
      volts") //Limit values at the break points and
      VBR=VBR1=VBR2
23 So=-RF/R1
24 disp(" So=-RF/R1= "+string(So))// slope of Transfer
      characteristic at zero crossings
25 S1=-(RF1/R1)
26 disp(" S1=S2=-RF1/R1= "+string(S1))// slope of
      Transfer characteristic at the extreme ends
27 VSL=(-VBR1/So)
28 disp(" VSL=VSU=(-VBR1/So)= "+string(VSL)+" volts") //
      magnitude of input voltage required to produce
      vo=VBR
29 VSU=VSL
30 disp(" part ( ii ) ")
31 VSU=(VBR2/Av)//Formulae
32 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

```
1 //Ex13_1
2 clc
3 VT=26*10^(-3)
4 disp("VT= "+string(VT)+" volts") // Thermal voltage
5 R1=5*10^(3)
6 disp("R1= "+string(R1)+ " ohm") // resistance
7 Iso=1*10^(-10)
8 disp(" Iso = "+string(Iso)+" ampere") // Scale
    factor (as current) directly proportional to cross
    -section area of EBJ
9
10 disp(" part (i) ")
11 vs=1*10^(-3)
12 disp(" vs= "+string(vs)+" volts") // Input voltage1
13 vo=-VT*[log(vs/(Iso*R1))]
14 disp(" vo=-VT*[log(vs/(Iso*R1))]= "+string(vo)+"
    volts") // Output voltage of Log OP-AMP for
    input1 i.e vs = 1 mV
15
16 disp(" part (ii) ")
17 vs=10*10^(-3)
```

```

18 disp(" vs= "+string(vs)+" volts") // Input voltage2
19 vo=-VT*[log(vs/(Iso*R1))]
20 disp(" vo=-VT*[log(vs/(Iso*R1))]= "+string(vo)+"
    volts") // Output voltage of Log OP-AMP for
    input1 i.e vs = 10 mV
21
22 disp(" part(iii)")
23 vs=100*10^(-3)
24 disp(" vs= "+string(vs)+" volts") // Input voltage3
25 vo=-VT*[log(vs/(Iso*R1))]
26 disp(" vo=-VT*[log(vs/(Iso*R1))]= "+string(vo)+"
    volts") // Output voltage of Log OP-AMP for
    input1 i.e vs = 100 mV
27
28 disp(" part(iv)")
29 vs=1
30 disp(" vs= "+string(vs)+" volts") // Input voltage4
31 vo=-VT*[log(vs/(Iso*R1))]
32 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

```

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

```

```

10 disp("vs= "+string(vs)+" volts") // Input voltage
11 vo=-VT*[log(vs/(Iso*R1))]
12 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

```

1 //Ex13_3
2 clc
3 VT=26*10^(-3)
4 disp("VT= "+string(VT)+" volts") // Thermal voltage
5 RF=100*10^(3)
6 disp("RF= "+string(RF)+ " ohm") // resistance
7 Iso=50*10^(-9)
8 disp(" Iso = "+string(Iso)+" ampere") // Scale
    factor (as current) directly proportional to cross
    -section area of EBJ
9 vs=-0.162
10 disp("vs= "+string(vs)+" volts") // Input voltage
11 vo=Iso*RF*(exp(-vs/VT))
12 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

```
1 //Ex14_1 Refer fig 14.1(b) and (c)
2 clc
3 n=3
4 disp("n= "+string(n))// Number of bits
5 L=2^(n)
6 disp("L=2^(n)= "+string(L))// Number of quantization
   levels
7 VFS=10
8 disp("VFS= "+string(VFS)+" volts") // Maximum value
   of analog input voltage
9 Q.E=VFS/L
10 disp("Q.E=VFS/L= "+string(Q.E))// Quantization error
11 disp("Q.E= +0.625,-0.625")// To make Quantization
   error symmetrical it taken as (-Q.E/2) negative
   and positive value(+Q.E/2)
12 Resolution=(100/2^(n))//Formulae
13 disp("Resolution=(100/2^(n))= "+string(Resolution)+"
   percent")//Resolution
14 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

```
1 //Ex14_2
2 clc
3 n=3
4 disp("n= "+string(n))// Number of bits
5 L=2^(n)
6 disp("L=2^(n)= "+string(L))// Number of quantization
   levels
7 VFS=1024*10^(-3)
8 disp("VFS= "+string(VFS)+" volts") // Maximum value
   of analog input voltage
9
10 disp(" part (i) ")// Part (i)
11 LSB=VFS/(2^n)
12 disp("LSB=VFS/(2^n)= "+string(LSB)+" volts") //
   Lowest significant bit of 3-bit ADC
13
14 disp(" part (ii) ")// Part (ii)
15 disp("vh= 64 to 192 mV with offset") // Analog
   voltage corresponding to binary word 001
16
17 disp(" part (iii) ")// Part (iii)
18 I.E=(LSB)/2
19 disp("Inherent error ,I.E= (LSB)/2= -"+string(I.E)+"
   V,+"+string(I.E)+" V")// Inherent error in each
   binary word
20
21 disp(" part (iv) ")// Part (iv)
22 Resolution=(1*10^(-3))
23 disp("Resolution= "+string(Resolution)+" V")//
   Resolution
24 VFS=1
25 disp("VFS= "+string(VFS)+" V") // Maximum value of
```



```

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

```

Scilab code Exa 14.3 Find output voltage for weighted resistor DAC

```

1 //Ex14_3
2 clc
3 VREF=-10
4 disp("VREF= "+string(VREF)+" V") // Reference
    voltage
5 RF=5*10^(3)
6 disp("RF= "+string(RF)+ " ohm") //Feedback
    resistance
7 R=10*10^(3)
8 disp("R= "+string(R)+ " ohm") // resistance
9 vLSB=(-RF*VREF)/(8*R)// Since IF=I/8,so vLSB=(-RF*IF
    )=(-RF*I/8)=(-RF*VREF/8*R)
10 disp("vLSB=(-RF*VREF)/(8*R)=" +string(vLSB)+" V") //
    Equivalent voltage for binary word 0001
11 vo=-2*vLSB// Since current IF=I/4
12 disp("vo = -2*vLSB =" +string(vo)+" V") // Equivalent
    voltage for binary word 0010=2 (in decimal)
13 vo=-15*vLSB// Since current IF=I+(I/2)+(I/4)+(I/8)
    =(15*I/8),so vo=15*VLSB
14 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

```

1 //Ex14_4
2 clc
3 VREF=-10
4 disp("VREF= "+string(VREF)+" V") // Reference
    voltage
5 RF=5*10^(3)
6 disp("RF= "+string(RF)+ " ohm") //Feedback
    resistance
7 R=10*10^(3)
8 disp("R= "+string(R)+ " ohm") // resistance
9 vMSB=- (RF*VREF)/(2*R) // Since IF=I/2,so vMSB=(-RF*IF
    )=(-RF*I/2)=(-RF*VREF/2*R)
10 disp("vMSB=- (RF*VREF)/(2*R)=" +string(vMSB)+" V") //
    Equivalent voltage for binary word 1000=8(in
    decimal)
11 vo2=vMSB/2 // Since current IF=I/4
12 disp("vo2 = vMSB/2 =" +string(vo2)+" V") //
    Equivalent voltage for binary word 0100=4 (in
    decimal)
13 vo3=(15/8)*vMSB // Since current IF=I+(I/2)+(I/4)+(I
    /8)+(I/16)=(15*I/6),so vo=(15/8)*VMSB
14 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

```

1 //Ex14_5
2 clc
3 n=12
4 disp("n= "+string(n)) // Number of bits
5 VFS=50
6 disp("VFS= "+string(VFS)+" volts") // Maximum value
    of analog input voltage

```

```

7 S=VFS/(2^n)
8 disp("S=VFS/(2^n)= "+string(S)+" volts") // Maximum
  quantization error
9 Resolution=(100/2^(n))//Formulae
10 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

```

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

```
