Scilab Textbook Companion for
Principles of Electronics
by V. K. Mehta and R. Mehta[1]

Created by
Kavan Patel
B.Tech
Instrumentation Engineering
Dharmsinh Desai University
College Teacher
None
Cross-Checked by
Chaitanya

July 31, 2019

Book Description

Title: Principles of Electronics

Author: V. K. Mehta and R. Mehta

Publisher: S.Chand, New Delhi

Edition: 9

Year: 2005

ISBN: 81-219-2450-2

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

3

4

List of Scilab Codes

6

7

9

10

11

Chapter 1

Introduction

Scilab code Exa 1.1 `terminal voltage`

```
1  //chapter 1
2  //example 1.1
3  //page8
4
5  Eg=24  // V
6  Ri=.01  // ohm
7  P=100  // W
8
9  I=P/Eg  // we know that P=Eg*I since for ideal source
      , V is equivalent to Eg
10  Vi=I*Ri
11  V=Eg-(I*Ri)
12
13  printf("voltage drop in internal resistance = %.3f V
      \n",Vi)
14  printf("terminal voltage = %.3f V",V)
```

Scilab code Exa 17.1 `Example 1`

```
1  //chapter17
2  //example17.1
3  //page375
4
5  L1=58.6d-6 // H
6  C1=300d-12 // F
7
8  f=1/(2*%pi*(L1*C1)^0.5)
9  printf("frequency of oscillations = %.3f Hz or %.3f
      kHz",f,f/1000)
10
11 // in book the answer is 1199 kHz but the accurate
      answer is 1200.358 kHz
```

Scilab code Exa 1.2 Exmaple 2

```
1  //chapter1
2  //example1.2
3  //page10
4
5  Eg=500 // V
6  Ri=1000 // ohm
7
8  // for Rl=10 ohm
9  Rl1=10 // ohm
10 I1=Eg/(Rl1+Ri)
11 printf("load current for Rl=10ohm is %.3f A \n",I1)
12
13 // for Rl=10 ohm
14 Rl2=50 // ohm
15 I2=Eg/(Rl2+Ri)
16 printf("load current for Rl=50ohm is %.3f A \n",I2)
17
18 // for Rl=10 ohm
19 Rl3=100 // ohm
```

```
20  I3=Eg/(Rl3+Ri)
21  printf("load current for Rl=100ohm is %.3f A",I3)
```

Scilab code Exa 1.3 Exmaple 3

```
1   //chapter1
2   //example1.3
3   //page11
4
5   V=10 // V
6   R=10 // ohm
7
8   I=V/R // calculate short−circuit current by shorting
         AB
9   printf("equivalent current source has magnitude = %
        .3f A",I)
10
11  // no load is connected across AB and 10V source has
          negligible resistance
12  // so resistance across AB is 10 ohm
13
14  // the constant voltage source when converted to
        constant current source will thus have a source
        of 1A in parallel with resistor of 10 ohm
```

Scilab code Exa 1.4 Exmaple 4

```
1   //chapter1
2   //example1.4
3   //page12
4
5   I=6 // mA
6   R=2 // kilo ohm
```

```
7
8  V=I*R  // by ohm law
9  printf("voltage of voltage source = %.3f V",V)
10
11 // this voltage source when connected in series with
       2000 ohm gives equivalent voltage source for the
       given constant current source
```

Scilab code Exa 1.5 Exmaple 5

```
1  //chapter1
2  //example1.5
3  //page13
4
5  E=200  // V
6  Ri=100  // ohm
7
8  Rl=100  // for load=100ohm
9  I=E/(Ri+Rl)
10 Pl=I^2*Rl
11 Pt=I^2*(Rl+Ri)
12 efficiency=(Pl/Pt)*100
13 printf("for load=100 ohm, power delivered to load= %
      .3f W and efficiency=%.3f percentage \n \n",Pl,
      efficiency)
14
15 Rl=300  //for load=300ohm
16 I=E/(Ri+Rl)
17 Pl=I^2*Rl
18 Pt=I^2*(Rl+Ri)
19 efficiency=(Pl/Pt)*100
20 printf("for load=300 ohm, power delivered to load= %
      .3f W and efficiency=%.3f percentage \n \n",Pl,
      efficiency)
21
```

```
22 printf ("comment: \n ")
23 printf ("if load resistance is equal to internal
      resistance ,maximum power is \n transferred but
      efficiency is low \n ")
24 printf ("if load resistance is more than internal
      resistance , power transferred \n is less but
      efficiency is high")
```

Scilab code Exa 1.6 Exmaple 6

```
1 // chapter1
2 // example1 .6
3 // page14
4
5 // for maximum power transfer , resistance of load and
       amplifier should match
6 // so we take load =15 ohm
7
8 Rl =15 // ohm
9 Ri =15 // ohm
10 V =12 // V
11
12 Rt = Rl + Ri
13 I = V / Rt
14 P = I ^2* Rl
15
16 printf (" for maximum power transfer load must equal
      amplifier resistance \nso required load = %d ohm\
      n \n" , Rl)
17 printf (" power delivered to load = %.3 f W" , P)
```

Scilab code Exa 1.7 Exmaple 7

```
1  //chapter1
2  //example1.7
3  //page14
4
5  V=50  //  V
6  Rl=100  //  ohm
7  Zi=100+50*%i
8  //for  maximum  power  transfer  load  impedence  should
        be  conjugate  of  internal  resistance  so
9  Zl=100-50*%i
10
11  Zt=Zi+Zl
12  I=V/Zt
13  P=I^2*Rl
14
15  printf("load  for  maximum  power  (in  ohms)=")
16  disp(Zl)
17
18  printf("maximum  power  transfered  to  load=%.3f W",P)
```

Scilab code Exa 1.8 Exmaple 8

```
1  //chapter1
2  //example1.8
3  //page21
4
5  R=8  //  ohm
6  R1=10  //  ohm
7  R2=20  //  ohm
8  R3=12  //  ohm
9  //removing  100  ohm  resistance,  we  form  linear
        equations  by  assuming  currents  I1  through  loop1
        and  I2  through  loop2
10
11  //100=10*I1+20*(I1-I2)
```

```
12  //0=(12+8)*I2+20*(I2-I1)

13

14  //thus we get the following linear equations

15

16  //30*I1-20*I2=100
17  //-20*I1+40*I2=0
18  //solving these equations

19

20  a=[30 -20;-20 40]
21  b=[100;0]
22  x=inv(a)*b // matrix of I1 and I2

23

24  I2=x(2,1) // current through 8 ohm resistor

25

26  E0=I2*R
27  printf("voltage across AB with 100 ohm resistance
        not connected = %.3f V \n",E0)

28

29  R_equi=(R1*R2/(R1+R2))+R3
30  R0=R_equi*R/(R_equi+R)
31  printf("resistance between AB with 100 ohm removed
        and voltage source shorted = %.3f ohm \n",R0)

32

33  I=E0/(R0+R1)
34  printf("current through 100 ohm resistor = %.3f A",I
        )
```

Scilab code Exa 1.9 Exmaple 9

```
1  //chapter1
2  //exzmple1.8
3  //page16

4

5  R1=1 // kilo ohm
6  R2=1 // kilo ohm
```

```
7  R3=1  // kilo ohm
8  V=20  // V
9
10 E0=(R3/(R1+R2))*V  // thevenin voltage = voltage
      across R3 since A and B are open circuited which
      means no drop across R2
11 R0=R2+(R1*R3/(R1+R3))  // thevenin resistance =
      resistance between A and B with no load and
      voltage source shorted
12
13 printf("thevenin voltage = %.2f V \nthevenin
      resistance = %.2f kilo ohm",E0,R0)
```

Scilab code Exa 1.10 Exmaple 10

```
1  //chapter1
2  //example1.10
3  //page18
4
5  V=120  // V
6  R1=40  // ohm
7  R2=20  // ohm
8  R3=60  // ohm
9
10 //removing load, voltage across AB is
11 E0=R2*V/(R1+R2)
12
13 //replacing voltage source by short and removing
      load, resistance across AB is
14 R0=R3+(R1*R2/(R1+R2))
15
16 //for maximum power transfer, load must be equal to
      resistance across AB so
17 Rl=R0
18
```

```
19  P = E0 ^2/(4* Rl )
20  printf ("load  resistance  for  maximum  power  transfer  =
         %.3 f  ohm  \n", Rl )
21  printf ("maximum  power  to  load  = %.3 f  W ",P )
```

Scilab code Exa 1.11 Exmaple 11

```
1  // chapter1
2  // example1 .11
3  // page20
4
5  R1 =4  //  ohm
6  R2 =6  //  ohm
7  R3 =5  //  ohm
8  R4 =8  //  ohm
9  V =40  //  V
10
11  // load  is  removed  and  A  and  B  are  shorted
12  load_source = R1 +( R2 * R3 /( R2 + R3 ))
13  source_current = V / load_source
14
15  norton_current = source_current *( R2 /( R2 + R3 ))  //  short
        circuit  current  in  AB
16
17  printf ("shortcircuit  current  in  AB  = %.3 f  A  \n" ,
        norton_current )
18
19  // load  is  removed  and  battery  is  replaced  by  a
        short
20  norton_resistance = R3 +( R1 * R2 /( R1 + R2 ))
21  printf ("norton  resistance = %.3 f  ohm  \n" ,
        norton_resistance )
22
23  // equivalent  circuit  is  norton  current  source  in
        parallel  with  norton  resistance
```

21

```
24  I=norton_current*(norton_resistance/(
       norton_resistance+R4)) // current  through  8  ohm
       resistance
25  printf(" current  through  8ohm  resistor  = %.3 f A" ,I)
```

Scilab code Exa 1.12 Exmaple 12

```
1  // chapter 1
2  // example 1.12
3  // page 21
4
5  printf("To find Norton equivalent circuit we need to
        find \nNorton current I_N and Norton resistance
       R_N \n \n")
6  printf("If Thevenin resistnce = Ro and Thevenin
       voltage = Eo then \n \n")
7  printf("To convert Thevenin circuit to Norton
       circuit , \n")
8  printf("I_N=Eo/Ro and R_N=Ro \n \n")
9  printf("To convert Norton circuit to Thevenin
       circuit , \n")
10 printf("Eo=I_N*R_N and Ro=R_N \n")
```

Chapter 2

Electron emission

Scilab code Exa 2.1 Example 1

```
1  //chapter 2
2  //example 2.1
3  //page 29
4
5  A=60.2d4 // ampere per square m per square kelvin
6  T=2500 // kelvin
7  phi=4.517 // eV
8  d=0.01d-2 // m
9  l=5d-2 // m
10
11 b=11600*phi
12 Js=A*T^2*exp(-b/T)
13 a=%pi*d*l
14
15 emission_current=Js*a
16
17 printf("emission current=%f A",emission_current)
```

Scilab code Exa 2.2 Example 2

23

```
1  // Chapter 2
2  // example 2.2
3  // page 29
4  Js =0.1 // ampere per square cm
5  A=60.2 // ampere per square cm per square kelvin
6  T=1900 // kelvin
7
8  // Js=A*T^2*exp(-b/T) so b=-T*log(Js/(A*T^2))
9
10 b=-T*log(Js/(A*T^2))
11
12 // b=11600*phi so making phi as subject
13
14 phi=b/11600
15
16 printf("work function=%f eV \n",phi)
17 // the accurate answer is 3.521466
18 // but in the book it is mistakenly written as 3.56
19
20 if(2.63<phi & phi<4.52)
21       printf("thoriated tungsten has work function
            between 2.63eV to 4.52eV.\nSo sample is
            likely to be thoriated tingsten")
22 elseif(phi<=2.63 | phi>4.52)
23            printf("tungsten is contaminated")  // for
                pure tungsten, phi must be 4.52 exactly
24 else
25      printf("tungsten is pure")  // phi=4.52 implies
            tungsten is pure
26 end
27
28 // please note that there is error in the answer of
       work function phi in the book
29 // The correct answer is 3.521466 eV and not 3.56 eV
```

Chapter 3

Vacuum tubes

Scilab code Exa 3.1 Example 1

```
1  // chapter3
2  // example3.1
3  // page41
4
5  Ib1 =10 // mA
6  Eb1 =100 // V
7  Ib2 =20 // mA
8
9  // Ib is proportional to Eb^(3/2)
10 // so we can say Ib1/Ib2 = Eb1^1.5/Eb2^1.5
11 //thus we can write
12
13 log_Eb2 =(2/3) * log ( Eb1^1.5* Ib2 / Ib1 )
14 Eb2 = exp ( log_Eb2 )
15 printf (" required plate voltage = %.3 f V " ,Eb2 )
```

Scilab code Exa 3.2 Example 2

```
1  //chapter3
2  //example3.2
3  //page49
4
5  mu=20
6  rp=8000 // ohm
7
8  gm=mu/rp // since mu=rp*gm
9  gm_micro=gm*10^6 //micro mho
10 printf("mutual conductance of triode = %f mho or %.3
       f micro mho",gm,gm_micro)
```

Scilab code Exa 3.3 Example 3

```
1  //chapter3
2  //example3.3
3  //page49
4
5  // for constant Ec=-1.5
6  Eb1=100 // V
7  Eb2=150 // V
8  Ib1=7.5d-3 // A
9  Ib2=12d-3 // A
10
11 Eb_diff=Eb2-Eb1
12 Ib_diff=Ib2-Ib1
13
14 rp=Eb_diff/Ib_diff
15 rp_kilo_ohm=rp/10^3 //kilo ohm
16
17 printf("plate resistance = %.3 f ohm or %.3 f kilo ohm
       \n",rp,rp_kilo_ohm)
18
19 // for constant Eb=150
20 Ib1=5d-3 // A
```

```
21  Ib2 =12 d -3  // A
22  Ec1 = -3  // V
23  Ec2 = -1.5  // v
24
25  Ib_diff = Ib2 - Ib1
26  Ec_diff = Ec2 - Ec1
27
28  gm = Ib_diff / Ec_diff
29  gm_micro_mho = gm *10^6  // micro mho
30  printf (" mutual  conductance=%.3f  mho  or  %.3f  micro
        mho  \n" , gm , gm_micro_mho )
31
32  mu = rp * gm
33  printf (" amplification  factor  = %.3f" , mu )
34
35  // in  book  the  answer  of  amplification  factor  i.e.
        51.852  is  rounded  off  to  52
```

Scilab code Exa 3.4 Example 4

```
1  // chapter3
2  // example3 .4
3  // page50
4
5  Eb =250  // V
6  Ec = -3  // V
7
8  //  given  that  Ib =0.003*( Eb+30* Ec )^1.5  mA
9  //  differentiating  w.r.t  Ec  with  Eb=constant ,  we  get
10  gm =0.003*1.5*( Eb+30* Ec )^0.5*30*10^ -3
11  mutual_inductance_micro = gm *10^6
12
13  printf (" mutual  conductance  = %f  mho  or  %.3f  micro
        mho  \n" , gm , mutual_inductance_micro )
14
```

```
15  // differentiating given equation w.r.t Ec with Ib=
        constant, we get
16  // 0=0.003*10^-3*1.5*(Eb+Ec)^1.5*(mu+30) where mu is
          equal to ratio of changes in Eb and Ec i.e.
        amplification factor
17  // thus mu+30=0 hence we get
18  mu=-30
19      printf("here negative sign of amplification
            factor indicates that Eb and Ec are in
            opposite direction. \n \n")
20  // here we need not worry as to if mu may be
        positive because the equation given in problem
        statement will always give mu+30=0 i.e. mu=-30
21
22  printf("amplification factor = %.3f \n",mu)
23
24  rp=mu/gm
25  if rp<0      // rp can not be negative
26      rp=-rp
27  end
28
29  printf("plate resistance = %.3f ohm \n",rp)
30
31  //in book, the answers are less accurate. The
        accurate answers are
32  // gm=1707.630 micro mho
33  // plate resistance=17568.209 ohm
```

Scilab code Exa 3.5 Example 5

```
1  //chapter3
2  //example3.5
3  //page58
4
5  // use of Rsg = to obtain desired potential on
```

```
      screen grid since it is connected between power
      supply and screen grid
 6  // use of Csg = to provide ac grounding for the
      screen
 7
 8  Ebb=300 // V
 9  Ib=10d-3 // A
10  Rl=4.7d3 // ohm
11  Rk=68 // ohm
12  Isg=3d-3 // A
13  Vsg=150 // V
14
15  cathode_voltage=Ebb-(Ib*Rl)
16  grid_cathode_bias=-Rk*(Ib+Isg) // since current
      through cathode resistance is Ib+Isg
17  Rsg=(Ebb-Vsg)/Isg // since plate supply voltage =
      grid voltage + drop across Rsg
18  Rsg_kilo_ohm=Rsg/10^3 // in kilo ohm
19
20  printf("zero signal plate cathode voltage = %.3f V \
      n",cathode_voltage)
21  printf("grid cathode bias = %.3f V \n",
      grid_cathode_bias)
22  printf("Resistor Rsg = %.3f ohm or %.3f kilo ohm \n"
      ,Rsg,Rsg_kilo_ohm)
```

Chapter 4

Vacuum tube rectifiers

Scilab code Exa 4.1 Example 1

```scilab
1  // chapter4
2  // example4.1
3  // page68
4
5  rp=300  // ohm
6  Rl=1200  // ohm
7
8  Vm=200*2^0.5  //V
9  Im=Vm/(rp+Rl)
10 Idc=Im/%pi  // in ampere
11 Idc_mA=Idc*1000  // in mA
12 Irms=Im/2
13 Irms_mA=Irms*1000
14 Pdc=Idc^2*Rl
15 Pac=Irms^2*(rp+Rl)
16 efficiency=(Pdc/Pac)*100
17
18 printf("dc current = %.3f A or %.3f mA \n",Idc,
       Idc_mA)
19 printf("rms current = %.3f A or %.3f mA \n",Irms,
       Irms_mA)
```

```
20  printf ( " r e c t i f i c a t i o n   e f f i c i e n c y   = %.2 f   p e r c e n t a g e " ,
       efficiency )
21
22  //  accurate  answer  of  rms  current  is  94.281  mA  but
       in  book  it  is  given  as  94.5  mA
```

Scilab code Exa 4.2 Example 2

```
1  // chapter4
2  // example4 .2
3  // page68
4
5  rp =200  //  ohm
6  Rl =800  //  ohm
7  Edc =100  //  V
8
9  //  if  maximum  ac  voltage  required=Vm  then
10 //  Edc=Idc∗Rl  i . e .  Edc=Vm∗Rl /(%pi ∗( rp+Rl ) )
11 //  thus
12
13 Vm= Edc *%pi *( rp+Rl )/Rl
14 efficiency =(0.406/(1+( rp/Rl )))*100
15
16 printf ( " required  ac  voltage  = %.3 f  V  \n" ,Vm)
17 printf ( " r e c t i f i c a t i o n   e f f i c i e n c y   = %.3 f   p e r c e n t a g e " ,
       efficiency )
```

Scilab code Exa 4.3 Example 3

```
1  // chapter4
2  // example4 .3
3  // page69
4
```

```
 5  Vm =1000  // V
 6  rp =500  // ohm
 7  Rl =4500  // ohm
 8
 9  Im = Vm /( rp + Rl )  // in A
10  Idc = Im /% pi  // in A
11  Idc_mA = Idc *1000  // in mA
12  Irms = Im /2  // since ac current is equal to rms
       current
13  Irms_mA = Irms *1000  // in mA
14  W = Irms ^2*( rp + Rl )  // in watts
15
16  printf ( "dc ammeter reading = %.3 f A or %.3 f mA \n" ,
       Idc , Idc_mA )
17  printf ( "reading of ac ammeter = %.3 f A or %.3 f mA \n
       " , Irms , Irms_mA )
18  printf ( "reading of wattmeter = %.3 f W" ,W)
```

Scilab code Exa 17.4 Example 4

```
 1  // chapter17
 2  // example17 .4
 3  // page381
 4
 5  R =1 d6  // ohm
 6  C =68 d -12  // F
 7
 8  fo =1/(2*% pi *R*C *(6) ^0.5)
 9  printf ( "frequency of oscillations = %.3 f Hz" ,fo)
10
11  // in book the answer given is 954 Hz but the
       accurate answer is 955.511 Hz
```

Scilab code Exa 4.4 Example 4

```
1  //chapter4
2  //example4.4
3  //page74
4
5  Vs=300 // V
6  rp=500 // ohm
7  Rl=2000 // ohm
8  Vm=Vs*2^0.5 // in V
9  Im=Vm/(rp+Rl) // A
10 Idc=2*Im/%pi // A
11 Pdc=Idc^2*Rl // W
12 Irms=Im/2^0.5 //A
13 Pac=Irms^2*(rp+Rl) // W
14 efficiency=(Pdc/Pac)*100
15
16 printf("dc power output = %.3f W \n",Pdc)
17 printf("ac power input = %.3f W \n",Pac)
18 printf("efficiency = %.2f percentage",efficiency)
```

Scilab code Exa 4.5 Example 5

```
1  //chapter4
2  //example4.5
3  //page74
4
5  Vm=1000 // V
6  rp=500 // ohm
7  Rl=4500 // ohm
8
9  Im=Vm/(rp+Rl) // in ampere
10 Idc=2*Im/%pi // in ampere
11 Idc_mA=Idc*1000 // in mA
12 Iac=Im/2^0.5 // in ampere
```

```
13  Iac_mA=Iac*1000 // in mA
14
15  printf("dc ammeter reading = %.3f A or %.3f mA \n",
        Idc,Idc_mA)
16  printf("ac ammeter reading = %.3f A or %.3f mA",Iac,
        Iac_mA)
```

Chapter 5

Vacuum tube amplifiers

Scilab code Exa 5.1 Example 1

```
1  // chspter5
2  // example5 .1
3  // page85
4
5  mu =20
6  rp =10  //  kilo  ohm
7  Rl =15  //  kilo  ohm
8
9  Av = mu * Rl /( rp + Rl )
10
11  printf (" voltage  gain  = %.3 f " , Av )
```

Scilab code Exa 5.2 Example 2

```
1  // chspter5
2  // example5 .2
3  // page85
4
```

```
5  mu =20
6  rp =10  //  kilo  ohm
7  Rl =15  //  kilo  ohm
8  Eg =3  //  V
9
10  //  the  diagram  in  book  is  for  understanding  only .
        Also  we  do  not  have  a  block  of  " triode "  in  scilab
         xcos .  The  figure  is  not  required  to  solve  the
        problem .
11
12  Av = mu * Rl /( rp + Rl )
13  Ip =( mu * Eg /2^0.5) /( rp + Rl )
14  V_out = Ip * Rl
15
16  printf (" voltage  gain  =  %.3 f  \n" , Av )
17  printf (" load  current  =  %.3 f  mA  \n" , Ip )
18  printf (" output  voltage  =  %.3 f  V" , V_out )
19
20  //  the  accurate  answer  for  output  voltage  is  25.456V
         but  in  book  it  is  given  as  25.35V
```

Scilab code Exa 5.3 Example 3

```
1  // chapter5
2  // example5 .3
3  // page85
4
5  //  for  Rl =50 ,  Av =30
6  // for  Rl =85 ,  Av =34
7
8  //  Av=mu* Rl /( rp+Rl )
9  //  thus
10  //  Av*rp−mu*Rl=−Av* rl
11  //  substituting  for  Rl =50  and  Rl =85  we  get  the
        following  linaer  equations
```

```
12
13  //  30*rp−50*mu=−1500  and
14  //  34*rp−85*mu=−2890
15  //  solving  by  matrix
16
17  a=[30 34 ; -50 -85]
18  b=[-1500  -2890]
19  solution=b/a
20  mu=solution(1,2)
21  rp=solution(1,1)  //  in  kilo  ohms  since  RL  was  in
        kilo  ohm  in  the  equations
22
23  gm_kilo_mho=mu/rp
24  gm=gm_kilo_mho/1000
25  printf("mu = %.3f \n",mu)
26  printf("rp = %.3f kilo ohm \n",rp)
27  printf("gm = %.4f mho \n",gm)
```

Scilab code Exa 5.4 Example 4

```
1  //chapter5
2  //example5.4
3  //page86
4
5  mu=6
6  Eg=9  //  V
7  rp=2400  //  ohm
8  Rl=3000  //  ohm
9
10  Ip=mu*Eg/(rp+Rl)  //  A
11  power=Ip^2*Rl  //  W
12
13  printf("ac power in load = %.3f W",power)
```

Scilab code Exa 5.5 Example 5

```scilab
1  //chapter5
2  //example5.5
3  //page95
4
5  rp=1000 // ohm
6  Rl=10 // ohm
7  Eg=8 // V
8  mu=20
9
10 // the diagram in book is for understanding only.
       Also we do not have a block of "triode" in scilab
        xcos. The figure is not required to solve the
       problem.
11 // however, the equivalent circuit has been drawn in
        xcos for reference.
12
13 // since rp=n^2*Rl for maximum power transfer so
14 n=(rp/Rl)^0.5
15
16 // P_max=Ip^2*RE where Ip=mu*Eg/(rp+RE) and RE=rp
17 // thus
18 P_max=((mu*Eg)^2)/(4*rp)
19
20 printf("transformation ratio n= %.2f \n",n)
21 printf("power supplied to speaker when signal is 8V
       rms is = %.3f W",P_max)
```

38

Chapter 9

Semiconductor diode

Scilab code Exa 9.1 Example 1

```
1  //chapter9
2  //example9.1
3  //page142
4
5  printf("in fig. (i), the conventional current coming
        out of battery flows in the \nbranch circuits.
       In diode D1,the conventional current flows in the
        \ndirection of arrowhead and hence this diode is
        forward biased. \nHowever in diode D2, the
       conventional current flows opposite \nto
       arrowhead and hence this diode is reverse biased
       .\n \n")
6  printf("in fig. (ii), During the positive half cycle
        of input ac voltage, the \nconventional current
       flows in the direction of arrowhead and hence
       diode \nis forward biased. However, during the
       negative half cycle \nof input ac voltage, the
       diode is reverse biased.\n \n")
7  printf("in fig. (iii), During the positive half
       cycle of input ac voltage, the \nconventional
       current flows in the direction of arrowhead in D1
```

but it flows \nopposite to arrowhead in D2. So
during positive half cycle, \ndiode D1 is forward
biased and diode D2 is reverse biased. \nHowever
in the negative half cycle of the input ac
voltage, diode D2 \nis forward biased and diode
D1 is reverse biased.\n \n")
8 printf("in fig. (iv), During the positive half
cycle of input ac voltage, \nboth diodes are
reverse biased. However in the negative half
cycle of the \ninput ac voltage, both diodes are
forward biased.\n \n")

Scilab code Exa 9.2 Example 2

```
1  // chapter9
2  // example9.2
3  // page145
4
5  Vi_p=20 // V
6  rf=10 // ohm
7  Rl=500 // ohm
8  Vo=0.7 // V
9  Vin=20 // V
10
11 // peak current through diode will occur when Vin=Vf
        so
12 Vf=Vin
13 // since Vf=Vo+If_peak(rf+Rl) making If_peak as
      subject we get
14 If_peak1=(Vf-Vo)/(rf+Rl) // in ampere
15 Vout_peak1=If_peak1*Rl
16
17 // for ideal diode, Vo=0 and rf=0 so
18 // Vf=If_peak*Rl so we get
19 If_peak2=Vf/Rl   // in ampere
```

```
20  Vout_peak2=If_peak2*Rl
21
22  printf("peak  current  through  given  diode  = %.3f mA
        and  peak  output  voltage  = %.3f V \n",If_peak1
        *1000,Vout_peak1)
23  printf("peak  current  through  ideal  diode  = %.3f mA
        and  peak  output  voltage  = %.3f V \n",If_peak2
        *1000,Vout_peak2)
```

Scilab code Exa 9.3 Example 3

```
1  //chapter9
2  //example9.3
3  //page146
4
5  R1=50  // ohm
6  R2=5  // ohm
7  V=10  // V
8
9  Eo=V*R2/(R1+R2)  // thevenin  voltage
10 Ro=R1*R2/(R1+R2)  // thevenin  resistance
11 I_D=Eo/Ro  // current  through  diode  in  ampere
12
13 printf("current  through  diode  = %.3f mA \n",I_D
        *1000)
```

Scilab code Exa 9.4 Example 4

```
1  //chapter9
2  //example9.4
3  //page146
4
5  V=10  // V
```

```
6   V_D1 =0.7  // V
7   V_D2 =0.7  // V
8   R =48  // ohm
9   R_D1 =1  // ohm
10  R_D2 =1  // ohm
11
12  // D1 and D3 are forward biased while D2 and D4 are
        reverse biased thus
13  V_net =V - V_D1 - V_D2
14  R_t = R_D1 + R + R_D2
15  I = V_net / R_t
16
17  printf ("circuit current = %.3 f mA \n",I *1000)
```

Scilab code Exa 9.5 Example 5

```
1   // chapter9
2   // example9 .5
3   // page147
4
5   E1 =24  // V
6   E2 =4  // V
7   Vo =0.7  // V
8   R =2  // kilo ohm
9
10  // diode D1 is forward biased and diode D2 is
        reverse biased so
11  I =( E1 - E2 - Vo )/R
12
13  printf ("current in the circuit = %.3 f mA \n",I)
```

Scilab code Exa 9.6 Example 6

```
1  // chapter9
2  // example9 .6
3  // page147
4
5  V=20  // V
6  V_D_Ge =0.3  // V
7
8  // when voltage is applied, Ge diode turns on first
       and 0.3 V is maintained across circuit so Si
       diode never turns on. So
9  V_A=V-V_D_Ge
10
11 printf (" voltage V_A at point A = %.3 f V \n",V_A)
```

Scilab code Exa 9.7 Example 7

```
1  // chapter9
2  // example9 .7
3  // page148
4
5  V=10  // V
6  V_D=0.7  // V
7  R_BC=2  // kilo ohm
8  R=2  // kilo ohm
9
10 // by Kirchoff voltage law we get
11 // -V_D-I_D*R_BC-2*I_D*R+V=0 thus making I_D as
       subject we get
12 I_D=(V-V_D)/(R_BC+2*R)
13 V_Q=2*I_D*R
14
15 printf ("I_D = %.3 f mA \n",I_D)
16 printf ("V_Q = %.3 f V \n",V_Q)
```

Scilab code Exa 9.8 Example 8

```
1  //chapter9
2  //example9.8
3  //page148
4
5  V=15 // V
6  R=0.5 // kilo ohm
7  V_D=0.7 // V
8
9  // both diodes are forward biased
10
11 I1=(V-V_D)/R
12 I_D1=I1/2
13 I_D2=I_D1
14
15 printf("current through diode D1 = %.3f mA and diode
        D2 = %.3f mA \n",I_D1,I_D2)
```

Scilab code Exa 9.9 Example 9

```
1  //chapter9
2  //example9.9
3  //page151
4
5  P_dc=40 // W
6  P_ac=100 // W
7
8  efficiency=100*P_dc/P_ac
9
10 printf("rectification efficiency = %.3f percent \n \
       n",efficiency)
```

```
11  printf(" remaining 60 watts are not lost. Crystal
       diode consumes only a \nlittle power due to its
       small internal resistance. \nActualy 100 W ac
       power is contained as 50 W in positive half \
       ncycle and 50 W in negative half cycle.\nThe 50 W
       of negative half cycle are not supplied at all.
       \nThe 50 W of positive half cycle are converted
       to 40 W \n")
```

Scilab code Exa 9.10 `Example 10`

```
1  //chapter9
2  //example9.10
3  //page152
4
5  n=10
6  Vp=230  // V
7
8  Vpm=2^0.5*Vp
9  Vsm=Vpm/n  // since n=Vpm/Vsm=N1/N2
10
11  // Idc=Im/%pi and Vdc=Idc*Rl so
12  // Vdc=(Im/%pi)*Rl .Also Im*Rl=Vsm so
13  Vdc=Vsm/%pi
14
15  // in negative half cycle diode is reverse biased so
       maximum secondary voltage appears across diode.
16  PIV=Vsm
17
18  printf(" output dc voltage = %.2f V \n",Vdc)
19  printf(" peak inverse voltage = %.2f V \n",PIV)
20
21  // accurate answer for output dc voltage is 10.35 V
       not 10.36 V
```

Scilab code Exa 9.11 Example 11

```
1  //chapter9
2  //example9.11
3  //page152
4
5  rf=20 // ohm
6  Rl=800 // ohm
7  Vm=50 // V
8
9  Im=Vm/(rf+Rl) // in ampere
10 Idc=Im/%pi // in ampere
11 Irms=Im/2 // in ampere
12 Pac=Irms^2*(rf+Rl)
13 Pdc=Idc^2*Rl
14 Vout=Idc*Rl
15 efficiency=100*Pdc/Pac
16
17 printf("Im = %.1f mA \n",Im*1000)
18 printf("Idc = %.1f mA \n",Idc*1000)
19 printf("Irms = %.1f mA \n \n",Irms*1000)
20 printf("ac power input = %.3f W \n",Pac)
21 printf("dc power output = %.3f W \n \n",Pdc)
22 printf("dc output voltage = %.3f V \n \n",Vout)
23 printf("efficiency = %.3f percent \n",efficiency)
```

Scilab code Exa 9.12 Example 12

```
1  //chapter9
2  //example9.12
3  //page153
4
```

```
 5  Vdc=50  // V
 6  rf=25  // ohm
 7  Rl=800  // ohm
 8
 9  // Vdc=Idc*Rl and Idc=Im/%pi so
10  // Vdc=Im*Rl/%pi
11  // but Im=Vm/(rf+Rl) so
12  // Vdc=Vm*Rl/(%pi*(rf+Rl))
13  // making Vm as subject we get
14
15  Vm=Vdc*%pi*(rf+Rl)/Rl
16
17  printf("ac voltage required = %.1f V \n",Vm)
```

Scilab code Exa 9.13 Example 13

```
 1  //chapter9
 2  //example9.13
 3  //page157
 4
 5  rf=20  // ohm
 6  Rl=980  // ohm
 7  Vs=50  // V
 8
 9  Vm=Vs*2^0.5
10  Im=Vm/(rf+Rl)
11  Idc=2*Im/%pi  // in ampere
12  Irms=Im/2^0.5  // in ampere
13
14  printf("mean load current  = %.3f mA \n",Idc*1000)
15  printf("rms load current = %.3f mA \n",Irms*1000)
```

Scilab code Exa 19.14 power

```
1  //chapter19
2  //example19.14
3  //page424
4
5  Vc=5 // V
6  V_lower=2.5 // V
7  V_upper=2.5 // V
8  R=2 // kilo ohm
9
10 // figure given in book is just for understanding
       purpose.It is not a part of solution.
11 // however, the figure has been made in xcos and
       screenshot has been attached for reference
12
13 // since power=(rms voltage)^2/R we get
14
15 Pc=(0.707*Vc)^2/R
16 P_lower=(0.707*V_lower)^2/R
17 P_upper=(0.707*V_upper)^2/R
18 Pt=Pc+P_lower+P_upper
19
20 printf("power delivered by carrier = %.3f mW \n",Pc)
21 printf("power delivered by lower sideband = %.3f mW
       \n",P_lower)
22 printf("power delivered by upper sideband = %.3f mW
       \n",P_upper)
23 printf("total power delivered by AM wave = %.3f mW \
       n",Pt)
```

Scilab code Exa 9.14 Example 14

```
1  //chapter9
2  //example9.14
3  //page157
4
```

```
5   rf=0
6   n=5
7   Vp=230  // V rms
8   Rl=100  //ohm
9
10  Vs=Vp/n  // V rms
11  Vsm=Vs*2^0.5  // maximum voltage across secondary
12  Vm=Vsm/2  // maximum voltage across half secondary
       winding
13
14  Idc=2*Vm/(%pi*Rl)
15  Vdc=Idc*Rl
16  PIV=Vsm
17  efficiency=100*0.812/(1+rf/Rl)
18
19  printf("dc output voltage = %.3f V \n",Vdc)
20  printf("PIV = %.3f V \n",PIV)
21  printf("efficiency = %.3f percent \n",efficiency)
```

Scilab code Exa 9.15 Example 15

```
1   //chapter9
2   //example9.15
3   //page158
4
5   n=4
6   Rl=200  // ohm
7   fin=50  // Hz
8   Vp=230  // V rms
9
10  Vs=Vp/n  // V rms
11  Vsm=Vs*2^0.5  // maximum voltage across secondary
12
13  Idc=2*Vsm/(%pi*Rl)
14  Vdc=Idc*Rl
```

```
15  PIV = Vsm
16
17  // in full wave rectifier , output frequency is twice
        input frequency since there are two ouput pulses
        for each cycle of input
18  fout =2* fin
19
20  printf (" dc output voltage = %.3 f V \n" , Vdc )
21  printf (" peak inverse voltage = %.3 f V \n" , PIV )
22  printf (" output frequency = %.3 f Hz" , fout )
23
24  // the accurate answer for dc output voltage is
        51.768 V but in book it is given as 52 V
```

Scilab code Exa 9.16 Example 16

```
1  // chapter9
2  // example9 .16
3  // page158
4
5  // for dc output
6      // for centre -tap circuit
7          n=5
8          Vp =230 // V rms
9          Rl =100 // ohm
10         Vs = Vp /n // V rms
11         Vsm = Vs *2^0.5 // maximum voltage across
                secondary
12         Vm = Vsm /2 // maximum voltage across half
                secondary winding
13         Vdc =2* Vm /% pi // since Vdc = Idc * Rl and Idc =2*
                Vm /(% pi * Rl )
14
15      // for bridge circuit
16          n_dash =5
```

```
17          Vp_dash =230  // V rms
18          Rl_dash =100  //ohm
19          Vs_dash=Vp_dash/n_dash// V rms
20          Vsm_dash=Vs*2^0.5 // maximum voltage across
                 secondary
21          Vm_dash=Vsm_dash
22          Vdc_dash=2*Vm_dash/%pi // since Vdc=Idc*Rl
                and Idc=2*Vm/(%pi*Rl)
23
24
25 // for same dc output Vm must be same for both
      circuits i.e. n=5 for centre-tap and n=10 for
      bridge
26      // for centre-tap circuit
27          n1=5
28          Vs1=Vp/n1 // V rms
29          Vsm1=Vs1*2^0.5 // maximum voltage across
                 secondary
30          Vm1=Vsm1/2
31          PIV1=2*Vm1
32
33      // for bridge circuit
34          n2=5
35          Vs2=Vp/n2 // V rms
36          Vsm2=Vs2*2^0.5 // maximum voltage across
                 secondary
37          Vm2=Vsm2/2
38          PIV2=Vm2
39
40 printf("dc output voltage for centre-tap circuit = %
      .3f V \n",Vdc)
41 printf("dc output voltage for bridge circuit = %.3f
      V \n \n",Vdc_dash)
42
43 printf("for same output, PIV for centre-tap circuit
      = %.3f V and bridge circuit = %.3f V \n",PIV1,
      PIV2)
44
```

```
45  // the figure of transformer is for reference only.
       Also it cannot be plotted in scilab since scilab
       does not have centre−tap transformer
```

Scilab code Exa 9.17 Example 17

```
1  //chapter9
2  //example9.17
3  //page160
4
5  Vin=240 // V rms
6  Rl=480 // ohm
7  rf=1 // ohm
8
9  Vm=Vin*2^0.5
10 // for bridge rectifier we know that
11 Im=Vm/(2*rf+Rl)
12 Idc=2*Im/%pi
13 Irms=Im/2
14 P=Irms^2*rf
15
16 printf("mean load current = %.3f A \n",Idc)
17 printf("power dissipated in each diode = %.3f W \n",
      P)
18
19 // the accurate answers are mean load current =
      0.448 A and power dissipated in each diode =
      0.124 W
```

Scilab code Exa 9.18 Example 18

```
1  //chapter9
2  //example9.18
```

```
3  //page162
4
5  Vrms_A=0.5  // V
6  Vdc_A=10  // V
7  Vrms_B=1  // V
8  Vdc_B=25  // V
9
10  ripple_A=Vrms_A/Vdc_A
11  ripple_B=Vrms_B/Vdc_B
12
13  if ripple_A>ripple_B
14      printf("power supply B is better \n")
15  elseif ripple_B>ripple_A
16      printf("power supply A is better \n")
17  else
18      printf("both are equal \n")
19  end
```

Scilab code Exa 9.19 Example 19

```
1  //chapter9
2  //example9.19
3  //page165
4
5  // the waveform given in book is for understanding
       only. It is not required to solve the problem.
        Also it cannot be plotted in scilab unless Vm and
        Vdc are given.
6
7  R=25  // ohm
8  Rl=750  // ohm
9  Vm=25.7  // V
10
11  Vdc_dash=2*Vm/%pi
12  Vdc=Vdc_dash*Rl/(R+Rl)
```

```
13
14 printf (" voltage across load is %.3 f V plus a small
       ripple \n",Vdc)
15
16 // the accurate answer is 15.833 V but in book it is
        given as 15.9 V
```

Scilab code Exa 9.20 Example 20

```
1 // chapter9
2 // example9.20
3 // page170
4
5 R=5 // kilo ohm
6 Rl =10 // kilo ohm
7 Ei =120 // V
8 Vz =50 // V
9
10 V=Ei*Rl/(R+Rl) // voltage across open circuit if
      zener diode is removed
11 Vo=Vz // output voltage
12 V_R=Ei-Vz // drop across R
13 Il=Vz/Rl // load current
14 I=V_R/R // current through R
15
16 // by Kirchoff first law I=Iz+Il
17 Iz=I-Il
18
19 printf (" output voltage = %.3 f V \n",Vo)
20 printf (" voltage drop across series resistance = %.3 f
       V \n",V_R)
21 printf (" current through Zener diode = %.3 f mA \n",Iz
      )
```

Scilab code Exa 9.21 Example 21

```
1  //chapter9
2  //example9.21
3  //page171
4
5  Vmax=120 // V
6  Vmin=80 // V
7  Vz=50 // V
8  R_L=10 // kilo ohm
9  R1=5 // kilo ohm
10
11 // zener diode is on for Vmax and Vmin both since
      they are > Vz
12
13 // for max Iz
14     V_R1=Vmax-Vz
15     I=V_R1/R1 // current through R1
16     I_L=Vz/R_L // current through load
17     // by Kirchoff first law I=I_L+Iz so applying it
            we get
18     Iz_max=I-I_L
19
20 // for min Iz
21     V_R1_dash=Vmin-Vz
22     I_dash=V_R1_dash/R1// current through R1
23     I_L_dash=Vz/R_L // current through load
24     // by Kirchoff first law I=I_L+Iz so we get
25     Iz_min=I_dash-I_L_dash
26
27 printf("maximum zener current = %.3f mA \n",Iz_max)
28 printf("minimum zener current = %.3f mA \n",Iz_min)
```

Scilab code Exa 9.22 Example 22

```
1  // chapter9
2  // example9.22
3  // page172
4
5  Ei =12  // V
6  Vz =7.2  // V
7  Eo = Vz
8  Iz_min =10d -3  // A
9  Il_max =100d -3  // A
10
11 // we see that R=(Ei-Eo)/(Iz-Il) and minimum Iz
        occurs  when  Il  is  maximum  so
12 R =( Ei - Eo )/( Iz_min + Il_max )
13
14 printf (" required  series  resistance  = %.3f ohm \n",R)
15
16 // on inserting this series resistance the output
        voltage  will  remain  constant  at  7.2  V
17
18 // the accurate answer is 43.636 ohm but in book it
        is  given  as  43.5  ohm
```

Scilab code Exa 9.23 Example 23

```
1  // chapter9
2  // example9.23
3  // page172
4
5  Ei =22  // V
6  Vz =18  // V
```

```
7  Rl =18  // ohm
8  Eo = Vz
9  Iz_min =200 d -3  // A
10
11 // Zener current will be min when input voltage is
       min
12
13 // load current is
14 Il_max = Vz / Rl
15
16 // we see that R=(Ei−Eo)/(Iz−Il) and minimum Iz
       occurs when Il is maximum so
17 R=( Ei - Eo )/( Iz_min + Il_max )
18
19 printf (" required series resistance = %.3 f ohm \n",R)
20
21 // on inserting this series resistance the output
       voltage will remain constant at 18 V
```

Scilab code Exa 9.24 Example 24

```
1  // chapter9
2  // example9 .24
3  // page172
4
5  Ei =13  // V
6  Vz =10  // V
7  Eo = Vz
8  Iz_min =15 d -3  // A
9  Il_max =85 d -3  // A
10
11 // Zener current will be min when input voltage is
       min
12
13 // we see that R=(Ei−Eo)/(Iz−Il) and minimum Iz
```

```
     occurs  when  Il  is  maximum  so
14  R=(Ei-Eo)/(Iz_min+Il_max)
15
16  printf("required  series  resistance  = %.3f  ohm  \n",R)
```

Scilab code Exa 9.25 Example 25

```
1  //chapter9
2  //example9.25
3  //page173
4
5  Ei=45  // V
6  Vz1=15  // V
7  Vz2=15  // V
8  Iz=200d-3 // current  rating  for  each  zener  in  ampere
9
10 Eo=Vz1+Vz2
11
12 R=(Ei-Eo)/Iz
13
14 printf("regulated  output  voltage  = %.3f  V  \n",Eo)
15 printf("required  series  resistance  = %.3f  ohm  \n",R)
```

Scilab code Exa 9.26 Example 26

```
1  //chapter9
2  //example9.26
3  //page173
4
5  Ei=45  // V
6  Vz1=10  // V
7  Vz2=10  // V
8  Vz3=10  // V
```

```
 9  Iz=1000d-3 // current rating for each zener in
        ampere
10
11  Eo=Vz1+Vz2+Vz3
12
13  R=(Ei-Eo)/Iz
14
15  printf("required series resistance = %.3f ohm \n",R)
16
17  // since zener diode is not available in xcos,
        simple diodes are used to represent zener diode
        in the circuit made in xcos
```

Scilab code Exa 9.27 Example 27

```
 1  //chapter9
 2  //example9.27
 3  //page174
 4
 5  R=200 // ohm
 6  Rl=2000 // ohm
 7  Eo=30 // V
 8
 9  // for minimum input voltage i.e. Iz=0
10  Il=Eo/Rl
11  I=Il // since Iz=0
12  Vin_min=Eo+I*R
13
14  // for maximum input voltage i.e. Iz=25 mA
15  Iz=25d-3 // A
16  Il_dash=Eo/Rl
17  I_dash=Il_dash+Iz
18  Vin_max=Eo+I_dash*R
19
20  printf("minimum input voltage = %.3f V \n",Vin_min)
```

```
21  printf("maximum input voltage = %.3 f V \n",Vin_max)
22  printf("thus range of input = %.3 f to %.3 f V \n",
        Vin_min ,Vin_max)
```

Scilab code Exa 9.28 Example 28

```
1  //chapter9
2  //example9.28
3  //page174
4
5  Ei =16  // V
6  Vz =12  // V      since we want to ragulate at 12 V
7  Eo = Vz
8  Iz_min =0  // A
9  Il_max =200d -3  // A
10
11  // Zener current will be min when input voltage is
        min
12
13  // we see that R=(Ei-Eo)/(Iz-Il) and minimum Iz
        occurs when Il is maximum so
14  R=(Ei -Eo)/(Iz_min+Il_max)
15
16  Izm = Il_max
17  Pzm = Vz*Izm
18
19  printf("Zener voltage = %.3 f V \n",Vz)
20  printf("required series resistance = %.3 f ohm \n",R)
21  printf("maximum power rating of zener diode = %.3 f W
        \n",Pzm)
```

Chapter 10

Special purpose diodes

Scilab code Exa 10.1 required series resistor

```
1  // chapter10
2  // example10.1
3  // page182
4
5  Vs =10  // V
6  Vd =1.6   // V
7  If =20d-3  // A
8
9  Rs =( Vs - Vd )/ If
10
11 printf (" required  series  resistor  = %.3 f  ohm ",Rs )
```

Scilab code Exa 10.2 current through LED

```
1  // chapter10
2  // example10.2
3  // page183
4
```

```
5  Vs =15  // V
6  Vd =2  // V
7  Rs =2.2 d3  // ohm
8
9  If =( Vs - Vd )/ Rs
10
11 printf (" current  through  LED = %.3 f  A  or  %.3 f  mA ", If ,
       If *1000)
```

Scilab code Exa 10.3 dark resistance

```
1  // chapter10
2  // example10 .3
3  // page187
4
5  // from  graph ,  we  see  that  for  zero  illumination ,  the
        reverse  current  i . e .  dark  current  is  50  micro
       ampere
6
7  Ir =50 d -6   // A
8  Vr =10  // V
9
10 Rr = Vr / Ir
11
12 printf (" dark  resistance  = %.3 f  ohm  or  %.3 f  kilo  ohm "
       , Rr , Rr /1000)
```

Scilab code Exa 10.4 reverse current

```
1  // chapter10
2  // example10 .4
3  // page188
4
```

```
5  m=37.4  //  microA/mW/cm^2
6  E=2.5  //  mW/cm^2
7
8  //since  reverse  current = sensitivity*illumination
       we  can  write
9  Ir=m*E
10
11 printf(" reverse  current = %.3f  micro  ampere",Ir)
```

Scilab code Exa 10.5 `resonant frequency`

```
1  //chapter10
2  //example10.5
3  //page192
4
5  L=1d-3  //  H
6  C=100d-12  //  F
7
8  fr=1/(2*%pi*(L*C)^0.5)
9
10 printf(" resonant  frequency = %.3f  Hz  or  %.3f  kHz",fr
      ,fr/1000)
```

Scilab code Exa 10.11 `base current`

```
1  //chapter11
2  //example11.11
3  //page210
4
5  V_Rc=1
6  gain_beta=45
7  Rc=1  //  kilo  ohm
8
```

```
 9  Ic=V_Rc/Rc
10  //since gain_beta=Ic/Ib
11  Ib=Ic/gain_beta
12
13  printf("base current = %.3f mA",Ib)
```

Chapter 11

Transistors

Scilab code Exa 11.1 voltage amplification

```scilab
1  //chapter11
2  //example11.1
3  //page202
4
5  Rin=20  //ohm
6  Rout=100d3  //ohm
7  Rc=1d3  //ohm
8  signal=500d-3  //V
9
10 Ie=signal/Rin  // A
11 Ic=Ie
12 Vout=Ic*Rc
13 Av=Vout/signal
14
15 printf("voltage  amplification  = %.2f  \n",Av)
```

Scilab code Exa 11.2 base current

```
1  //chapter11
2  //example11.2
3  //page205
4
5  Ie=1  //mA
6  Ic=0.95  //mA
7
8  //  since  Ie=Ib+Ic  we  get
9  Ib=Ie-Ic
10
11  printf("base  current  = %.3f  mA  \n",Ib)
```

Scilab code Exa 11.3 base current

```
1  //chapter11
2  //example11.3
3  //page205
4
5  alpha=0.9
6  Ie=1  //mA
7
8  //  since  alpha=Ic/Ie  we  get
9
10  Ic=alpha*Ie
11
12  //  since  Ie=Ic+Ib  we  get
13
14  Ib=Ie-Ic
15
16  printf("base  current  = %.3f  mA  \n",Ib)
```

Scilab code Exa 11.4 amplification factor

```
1  // chapter11
2  // example11.4
3  // page205
4
5  Ic=0.95
6  Ib=0.05
7
8  Ie=Ib+Ic
9  alpha=Ic/Ie
10
11 printf("amplification factor = %.3f \n",alpha)
```

Scilab code Exa 11.5 collector current

```
1  // chapter11
2  // example11.5
3  // page205
4
5  Ie=1  //mA
6  alpha=0.92
7  Icbo=50d-3  //mA
8
9  Ic=alpha*Ie+Icbo
10
11 printf("collector current = %.3f mA \n",Ic)
```

Scilab code Exa 11.6 base current

```
1  // chapter11
2  // example11.6
3  // page205
4
5  alpha=0.95
```

```
6  V_Rc=2  // V
7  Rc=2  //kilo ohm
8
9  Ic=V_Rc/Rc  // mA
10
11 // since alpha=Ic/Ie
12 Ie=Ic/alpha
13
14 // since Ie=Ib+Ic
15 Ib=Ie-Ic
16
17 printf("base current = %.3f mA \n",Ib)
```

Scilab code Exa 11.7 collector curent

```
1  //chapter11
2  //example11.7
3  //page206
4
5  Vbe=0.7  // V
6  Vcc=18  // V
7  Vee=8  // V
8  Rc=1.2  // kilo ohm
9  Re=1.5  //kilo ohm
10
11 // by Kirchoff's voltage law to emitter side loop,
       we get Vee=Ie*Re+Vbe so
12 Ie=(Vee-Vbe)/Re
13 Ic=Ie  // nearly
14
15 // by Kirchoff's voltage law to collector side loop,
       we get Vcc=Ic*Rc=Vcb so
16 Vcb=Vcc-Ic*Rc
17
18 printf("collector curent  = %.3f mA \n",Ic)
```

```
19 printf(" collector base voltage = %3f V \n",Vcb)
```

Scilab code Exa 11.8 Example 8

```
1 //chapter11
2 //example11.8
3 //page209
4
5 alpha1=0.9
6 alpha2=0.98
7 alpha3=0.99
8
9 beta1=alpha1/(1-alpha1)
10 beta2=alpha2/(1-alpha2)
11 beta3=alpha3/(1-alpha3)
12
13 printf(" for alpha=0.9, beta=%.1f \n",beta1)
14 printf(" for alpha=0.98, beta=%.1f \n",beta2)
15 printf(" for alpha=0.99, beta=%.1f \n",beta3)
```

Scilab code Exa 11.9 Example 9

```
1 //chapter11
2 //example11.9
3 //page210
4
5 gain_beta=50
6 Ib=20d-3 // mA
7
8 // since gain_beta = Ic/Ib we get
9 Ic=gain_beta*Ib
10 Ie=Ic+Ib
11
```

```
12 printf(" emitter  current  = %.3 f  mA  \n",Ie)
```

Scilab code Exa 11.10 `collector current`

```
1  //chapter11
2  //example11.10
3  //page210
4
5  gain_beta=49
6  Ib=240d-3  // mA
7  Ie=12  // mA
8
9  alpha=gain_beta/(1+gain_beta)
10 Ic=alpha*Ie  //  or  Ic=gain_beta*Ib
11
12 printf(" collector  current  = %.3 f  mA  \n",Ic)
```

Scilab code Exa 11.11 `base current`

```
1  //chapter11
2  //example11.11
3  //page210
4
5  V_Rc=1
6  gain_beta=45
7  Rc=1  //  kilo  ohm
8
9  Ic=V_Rc/Rc
10 //since  gain_beta=Ic/Ib
11 Ib=Ic/gain_beta
12
13 printf(" base  current  = %.3 f  mA",Ib)
```

Scilab code Exa 11.12 collector emitter voltage

```
1  //chapter11
2  //example11.12
3  //page210
4
5  Rc=800d-3 // kilo ohm
6  V_Rc=0.5 // V
7  Vcc=8 // V
8  alpha=0.96
9
10 Vce=Vcc-V_Rc
11 Ic=V_Rc/Rc // mA
12 gain_beta=alpha/(1-alpha)
13 Ib=Ic/gain_beta
14
15 printf("collector emitter voltage = %.3f V \n",Vce)
16 printf("base current = %.3f mA \n",Ib)
```

Scilab code Exa 11.13 base current

```
1  //chapter11
2  //example11.13
3  //page211
4
5  Ic=1000 // micro ampere
6  // when emitter circuit is open, leakage current =
       Icbo so
7  Icbo=0.2 // micro ampere
8
9  // when base is open, leakage current = Iceo so
10 Iceo=20 // micro ampere
```

71

```
11
12  //since  Iceo=Icbo/(1-alpha)  we  get
13  alpha=1-(Icbo/Iceo)
14
15  //  since  Ic=alpha*Ie+Icbo  we  get
16  Ie=(Ic-Icbo)/alpha
17  Ib=Ie-Ic
18
19  printf("alpha = %.3f \n",alpha)
20  printf("emitter current = %.3f micro ampere \n",Ie)
21  printf("base current = %.3f micro ampere \n",Ib)
```

Scilab code Exa 11.14 `dc load line`

```
1  //chapter11
2  //example11.14
3  //page218
4
5  Vcc=12.5 // V
6  Rc=2.5 // kilo ohm
7
8  // we know that Vce=Vcc-Ic*Rc
9  // when Ic=0, Vce=Vcc i.e. 12.5V
10 // when Vce=0, Ic=Vcc/Rc i.e.5mA
11
12 // so equation of load line becomes Ic=-0.4*Vce+5
13 x=linspace(0,12.5,5)
14 y=-0.4*x+5
15 clf()
16 xtitle("dc load line","Vce(volts)","Ic(mA)")
17 plot2d(x,y,style=3,rect=[0,0,13,6])
```

Scilab code Exa 11.15 `Q point`

```
1  // chapter11
2  // example11.15
3  // page219
4
5  Vcc =12 // V
6  Rc =6 // kilo ohm
7
8  // we know that Vce=Vcc-Ic*Rc
9  // when Ic=0, Vce=Vcc i.e. 12V
10 // when Vce=0, Ic=Vcc/Rc i.e.2mA
11
12 // so equation of load line becomes Ic=-(1/6)*Vce+2
13 x=linspace(0,12,5)
14 y=-(1/6)*x+2
15 clf()
16 xtitle("dc load line","Vce(volts)","Ic(mA)")
17 plot2d(x,y,style=3,rect=[0,0,13,6])
18
19
20 // for Q point
21 Ib=20d-3 // mA
22 gain_beta=50
23
24 Ic=gain_beta*Ib
25 Vce=Vcc-Ic*Rc
26
27 printf("Q point = %.3f V and %.3f mA i.e. (%.3f,%.3f
     ) \n",Vce,Ic,Vce,Ic)
```

Scilab code Exa 11.16 collector load

```
1  // chapter11
2  // example11.16
3  // page219
4
```

```
5  Vcc =10
6  Ic =1  // mA
7  Rc1 =4  // kilo ohm
8  Rc2 =5  // kilo ohm
9
10  Vce1=Vcc-Ic*Rc1
11  Vce2=Vcc-Ic*Rc2
12
13  printf("for collector load = 4 kilo ohm, operating
       point is %.3f V,%.3f mA \n",Vce1,Ic)
14  printf("for collector load = 5 kilo ohm, operating
       point is %.3f V,%.3f mA \n",Vce2,Ic)
```

Scilab code Exa 11.17 input resistance

```
1  //chapter11
2  //example11.17
3  //page222
4
5  del_Vbe =200  //mV
6  del_Ib =100  // micro ampere
7
8  Ri=del_Vbe/del_Ib
9
10  printf("input resistance = %.3f kilo ohm \n",Ri)
```

Scilab code Exa 11.18 output resistance

```
1  //chapter11
2  //example11.18
3  //page222
4
5  Vce2 =10  // V
```

```
 6  Vce1 =2  // V
 7  Ic1 =2  // mA
 8  Ic2 =3  // mA
 9
10  del_Vce = Vce2 - Vce1  // V
11  del_Ic = Ic2 - Ic1  // mA
12
13  Ro = del_Vce / del_Ic
14
15  printf ("output resistance = %.3f kilo ohm \n",Ro)
```

Scilab code Exa 11.19 voltage gain

```
 1  // chapter11
 2  // example11.19
 3  // page223
 4
 5  Rc =2  // kilo ohm
 6  Ri =1  // kilo ohm
 7  gain_beta =50
 8
 9  // for single stage, R_AC=Rc so voltage gain becomes
10  Av = gain_beta * Rc / Ri
11
12  printf (" voltage gain = %.3f \n",Av)
```

Scilab code Exa 11.20 saturation collector current

```
 1  // chapter11
 2  // example11.20
 3  // page224
 4
 5  Vcc =20  // V
```

```
6  Rc=1 // kilo ohm
7
8  // for saturation collector current, knee voltage
       becomes 0V so we get
9  Ic_sat=Vcc/Rc
10
11 // it can be seen from the circuit that cut-off
       voltage (i.e. when Ib=0) equals Vcc itself
12 Vce_cutoff=Vcc
13
14 // the equation of load line becomes Ic=-Vce+20
15
16 clf()
17 x=linspace(0,20,5)
18 y=-x+20
19 plot2d(x,y,style=3,rect=[0,0,21,21])
20 xtitle("dc load line","Vce(volts)","Ic(mA)")
21
22 printf("saturation collector current = %.3f mA \n",
       Ic_sat)
23 printf("cut-off collector emitter voltage = %.3f V \
       n",Vce_cutoff)
```

Scilab code Exa 11.21 maximum allowable collector current

```
1  //chapter11
2  //example11.21
3  //page225
4
5  Vce=20 // V
6  Pd=100 // mW
7
8  // since Pd=Vce*Ic we get
9  Ic=Pd/Vce
10
```

```
11  printf("maximum  allowable  collector  current  = %.3 f
    mA \n ",Ic)
```

Chapter 12

Transistor biasing

Scilab code Exa 12.1 `circuit diagram`

```scilab
1  // chapter 12
2  // example 12.1
3  // page 235
4
5  V_CC=6 // V
6  R_C=2.5 // kilo ohm
7
8  // for faithful amplification V_CE should not be
        less than V_CC for Si transistor so
9  V_max=V_CC-1
10 I_max=V_max/R_C
11
12 // As negative and positive half cyces of input are
        equal, change in collector current will be equal
        and opposite so
13 I_min=I_max/2
14
15 printf("Maximum allowable collector current = %.3f
        mA \n",I_max)
16 printf("Minimum zero signal collector current = %.3f
        mA \n",I_min)
```

```
17
18 // the circuit diagram is constructed on xcos and
      its screenshot has been taken.
19 // the waveform given can not be obtained in xcos
      unless we assume necessary values as data is
      insufficient for plotting graph in scilab.
20 // so waveform is constructed as below
21
22 clf()
23 x=linspace(1,5*%pi,100)
24 [t]=sin(x)+1
25 plot(x,[t])
26 xtitle("max and min allowable collector currents","t
      ","i_c (mA)")
```

Scilab code Exa 12.2 maximum input signal voltage

```
1 //chapter12
2 //example12.2
3 //page236
4
5 Vcc=13 // V
6 V_knee=1 // V
7 Rc=4 // kilo ohm
8 gain_beta=100
9
10 V_Rc=Vcc-V_knee
11 Ic=V_Rc/Rc
12 Ib=Ic/gain_beta
13 Vs=Ic/5 // since Ic/Vs = 5 mA/V given
14
15 printf("maximum input signal voltage = %.3f V or %.3
      f mV \n",Vs,Vs*1000)
```

Scilab code Exa 12.3 Example 3

```
1  //chapter12
2  //example12.3
3  //page240
4
5  Vbb=2  // V
6  Vcc=9  // V
7  Rc=2  // kilo ohm
8  Rb=100  // kilo ohm
9  gain_beta=50
10
11 // by Kirchoff voltage law on base side, we get Ib*
      Rb+Vbe=Vbb so
12 Ib=Vbb/Rb  // Vbe is negligible
13 Ic=gain_beta*Ib
14
15 // by Kirchoff voltage law on collector side, we get
       Ic*Rc+Vce=Vcc so
16 Vce=Vcc-Ic*Rc
17
18 // now for Rb=50 kilo ohm
19 Rb2=50  // kilo ohm
20
21 // since Rb is halved, Ib is doubled so
22 Ib2=2*Ib
23 Ic2=Ib2*gain_beta
24 Vce2=Vcc-Ic2*Rc
25
26 printf("for Rb = 100 kilo ohm, collector current = %
      .3f mA \nand collector emitter voltage = %.3f V \
      n \n",Ic,Vce)
27 printf("for Rb = 50 kilo ohm, collector current = %
      .3f mA \nand collector emitter voltage = %.3f V \
```

```
n" , Ic2 , Vce2 )
```

Scilab code Exa 12.4 Example 4

```
1  //chapter12
2  //example12.4
3  //page241
4
5  Vcc=6  // V
6  Rb=530  // kilo ohm
7  Rc=2  // kilo ohm
8  gain_beta=100
9  Vbe=0.7  // V
10
11 // when Ic=0, Vce=Vcc i.e. Vce=6 and when Vce=0, Ic=
       Vcc/Rc i.e. Ic=6/2
12 // so equation of load line becomes Ic=-0.5*Vce+3
13
14 x=linspace(0,6,5)
15 y=-0.5*x+3
16 plot2d(x,y,style=3,rect=[0,0,7,4])
17 xtitle("dc load line","Vce(volts)","Ic(mA)")
18
19 // since Vcc=Ib*Rb+Vbe we get
20 Ib=(Vcc-Vbe)/Rb
21 Ic=Ib*gain_beta
22 Vce=Vcc-Ic*Rc
23
24 printf("the operating point is %.3f V and %.3f mA \n
       ",Vce,Ic)
25
26 stability_factor=gain_beta+1
27
28 printf("stability factor=%.1f \n",stability_factor)
```

Scilab code Exa 12.5 Example 5

```
1  //chapter12
2  //example12.5
3  //page242
4
5  Vcc=12 // V
6  gain_beta=100
7  Vbe=0.3 // V
8  Ic=1 // mA
9
10 // since gain_beta=Ic/Ib
11 Ib=Ic/gain_beta
12
13 // since Vcc=Ib*Rb+Vbe we get
14 Rb=(Vcc-Vbe)/Ib
15
16 gain_beta2=50
17
18 // since Vcc=Ib*Rb+Vbe we get
19 Ib2=(Vcc-Vbe)/Rb
20 Ic2=Ib2*gain_beta2
21
22 printf("for beta = 100, base resistor = %.3f kilo
      ohm \n",Rb)
23 printf("for beta = 50, zero signal collector current
        for same Rb is = %.3f mA \n",Ic2)
```

Scilab code Exa 12.6 Example 6

```
1  //chapter12
2  //example12.6
```

```
3  //page242
4
5  Vcc =10  // V
6  R_B =1d3  // kilo ohm
7  R_E =1  // kilo ohm
8  Vbe =0  // since it is negligible
9  gain_beta =100
10
11 // by Kirchoff voltage law to base side we get Vcc=
      I_B*R_B+Vbe+I_E*R_E
12 // but I_E=I_B+I_C and I_C=gain_beta*I_B
13 // so we get Vcc=I_B*R_B+Vbe+R_E*I_B*(1+gain_beta)
14 // making I_B as subject we get
15
16 I_B =(Vcc -Vbe )/(R_B+R_E*(1+gain_beta))  // in ampere
17 I_C =gain_beta*I_B  // in ampere
18 I_E =I_C+I_B  // in ampere
19
20 printf ("base current = %.4 f mA \n",I_B)
21 printf ("collector current = %.4 f mA \n",I_C)
22 printf ("emitter current = %.4 f mA \n",I_E)
```

Scilab code Exa 12.7 Example 7

```
1  //chapter12
2  //example12.7
3  //page243
4
5  V_CC =15  // V
6  gain_beta =100
7  V_BE =0.6  // V
8  V_CE =8  // V
9  I_C =2  // mA
10
11 // here V_CC=V_CE+I_C*R_C so we get
```

```
12  R_C =( V_CC - V_CE )/ I_C

13

14  I_B = I_C / gain_beta

15

16  // also V_CC=I_B*R_B+V_BE so we get
17  R_B =( V_CC - V_BE )/ I_B

18

19  printf (" collector  resistance = %.3 f  kilo  ohm  \n" ,R_C
        )
20  printf (" base  resistance = %.3 f  kilo  ohm  \n" ,R_B )
```

Scilab code Exa 12.8 Example 8

```
1  // chapter12
2  // example12.8
3  // page245
4
5  V_CC =20  // V
6  R_B =100  // kilo  ohm
7  R_C =1  // kilo  ohm
8  V_BE =0.7  // V
9  gain_beta =100
10
11  // we know that R_B=(V_CC–V_BE–gain_beta*R_C*I_B)/
         I_B so we get
12  I_B =( V_CC - V_BE )/( R_B + gain_beta * R_C )
13
14  I_C = gain_beta * I_B
15
16  V_CE = V_CC - I_C * R_C
17
18  printf (" operating  point  is %.3 f  V, %.3 f  mA  \n" ,V_CE ,
       I_C )
19
20  // the  accurate  answer  is  10.35V,9.65mA  but  in  book
```

Scilab code Exa 12.9 Example 9

```
1  //chapter12
2  //example12.9
3  //page245
4
5  V_CC=12 // V
6  gain_beta1=100
7  gain_beta2=50
8  V_BE=0.3 // V
9  V_CE=8   // V
10 I_C=1 // mA
11
12 // here V_CC=V_CE+I_C*R_C so we get
13 R_C=(V_CC-V_CE)/I_C
14
15 I_B=I_C/gain_beta1
16
17 // we know that R_B=(V_CC-V_BE-gain_beta1*R_C*I_B)/
      I_B so
18 R_B=(V_CC-V_BE-gain_beta1*R_C*I_B)/I_B
19
20
21 // for gain_beta=50 i.e. gain_beta2
22
23 // we know that R_B=(V_CC-V_BE-gain_beta2*R_C*I_B)/
      I_B so we get
24 I_B2=(V_CC-V_BE)/(R_B+gain_beta2*R_C)
25
26 I_C2=gain_beta2*I_B2
27
28 V_CE2=V_CC-I_C2*R_C
29
```

```
30  printf ("for beta=100,required base resistance = %.3f
        kilo ohm \n",R_B)
31  printf ("for beta=50,new operating point is %.3f V, %
        .3f mA \n",V_CE2,I_C2)
```

Scilab code Exa 12.10 `base resistance`

```
1  //chapter12
2  //example12.10
3  //page246
4
5  V_BE=0.7 // V
6  gain_beta=100
7  I_C=1 // mA
8  V_CE=2 // V
9
10 I_B=I_C/gain_beta
11
12 // since V_CE=V_BE+V_CB we get
13 V_CB=V_CE-V_BE
14
15 R_B=V_CB/I_B
16
17 printf ("base resistance=%.3f kilo ohm \n",R_B)
```

Scilab code Exa 12.11 `voltage across`

```
1  //chapter12
2  //example12.11
3  //page248
4
5  Vcc=15 // V
6  Re=2 // kilo ohm
```

```
 7  Rc =1 // kilo ohm
 8  gain_beta =100
 9  Vbe =0.7 // V
10  R1 =10 // kilo ohm
11  R2 =5 // kilo ohm
12
13  // when Ic=0, Vce=Vcc i.e. Vce=6 and when Vce=0, Ic=
        Vcc/(Rc+Re) i.e. Ic=15/(1+2)
14  // so equation of load line becomes Ic=-(1/3)*Vce+5
15
16  clf()
17  x=linspace(0,15,5)
18  y=-(1/3)*x+5
19  plot2d(x,y,style=3,rect=[0,0,16,6])
20  xtitle("dc load line","Vce(volts)","Ic(mA)")
21
22  V2=Vcc*R2/(R1+R2) // voltage across R2 i.e. 5 kilo
        ohm
23  Ie=(V2-Vbe)/Re
24  Ic=Ie
25  Vce=Vcc-Ic*(Rc+Re)
26
27  printf("the operating point is %.3f V and %.3f mA \n
        ",Vce,Ic)
```

Scilab code Exa 12.12 `thevenin`

```
1  //chapter12
2  //example12.12
3  //page249
4
5  Vcc =15 // V
6  Re =2 // kilo ohm
7  Rc =1 // kilo ohm
8  gain_beta =100
```

```
9  Vbe =0.7 // V
10  R1 =10 // kilo ohm
11  R2 =5 // kilo ohm
12
13  Eo = Vcc * R2 /( R1 + R2 )
14  Ro = R1 * R2 /( R1 + R2 )
15
16  printf (" thevenin voltage = %.3 f V \n",Eo)
17  printf (" thevenin resistance = %.3 f kilo ohm \n",Ro)
18
19  // here Eo=Ib*Ro+Vbe+Ie*Re
20  // now considering Ie=gain_beta*Ib, and making Ib as
        subject we get
21  // Ib=(Eo−Vbe)/(Ro+gain_beta*Re)
22  // Ic=gain_beta*Ib=gain_beta*(Eo−Vbe)/(Ro+gain_beta*
    Re)
23  // dividing numerator and denominator by gain_beta
    we get
24  // Ic=(Eo−Vbe)/(Re+Ro/gain_beta)
25  // Ro/gain_beta is negligible compared to Re so
26  Ic =( Eo - Vbe )/ Re
27  Vce = Vcc - Ic *( Rc + Re )
28
29  printf (" the operating point is %.3 f V and %.3 f mA \n
        ",Vce,Ic)
```

Scilab code Exa 19.12 Example 12

```
1  // chapter19
2  // example19.12
3  // page424
4
5  It =8.93 // A
6  Ic =8 // A
7
```

```
 8  // we know that (It/Ic)^2=1+m^2/2 so making m as
        subject we get
 9  m=(2*((It/Ic)^2-1))^0.5
10
11  printf("modulation factor = %.3f or %.3f percent \n"
        ,m,m*100)
```

Scilab code Exa 12.13 collector current

```
 1  //chapter12
 2  //example12.13
 3  //page250
 4
 5  R1=50 // kilo ohm
 6  R2=10 // kilo ohm
 7  Re=1 // kilo ohm
 8  Vcc=12 // V
 9  Vbe1=0.1 // V
10  Vbe2=0.3 // V
11
12  V2=Vcc*R2/(R1+R2) // voltage across R2
13
14  // for Vbe=0.1 V
15  Ic1=(V2-Vbe1)/Re
16
17  // for Vbe=0.3 V
18  Ic2=(V2-Vbe2)/Re
19
20  printf("for V_BE=0.1 V, collector current = %.3f mA
        \n",Ic1)
21  printf("for V_BE=0.3 V, collector current = %.3f mA
        \n \n",Ic2)
22
23  Vbe_change=100*(Vbe2-Vbe1)/Vbe1
24  Ic_change=-100*(Ic2-Ic1)/Ic1 // negative sign since
```

```
      Ic decreases
25 printf("comment : if V_BE changes by %.5f percent , \
      ncollector current changes by %.3f percent \n",
      Vbe_change , Ic_change )
26 printf("so collector current is independent of
      transistor parameter variations \n")
27
28 // the change in V_BE is 200 percent not 300 percent
      . It is mistake in textbook
```

Scilab code Exa 12.14 collector emitter

```
1  //chapter12
2  //example12.14
3  //page251
4
5  Vcc=20 // V
6  Re=5 // kilo ohm
7  Rc=1 // kilo ohm
8  Vbe=0 // considering it as negligible
9  R1=10 // kilo ohm
10 R2=10 // kilo ohm
11
12 V2=Vcc*R2/(R1+R2)
13
14 // since V2=Vbe+Ie*Re so
15 Ie=(V2-Vbe)/Re
16 Ic=Ie
17
18 Vce=Vcc-Ic*(Rc+Re)
19 Vc=Vcc-Ic*Rc
20
21 printf("emitter current = %.3f mA \n",Ie)
22 printf("collector emitter voltage = %.3f V \n",Vce)
23 printf("collector potential = %.3f V \n",Vc)
```

Scilab code Exa 12.15 `Kirchoff voltage law`

```
1  //chapter12
2  //example12.15
3  //page252
4
5  R_C=2.2 // kilo ohm
6  V_CC=9 // V
7  gain_beta=50
8  V_BE=0.3 // V
9  I_C=2 // mA
10 V_CE=3 // V
11
12 I_B=I_C/gain_beta
13 I1=10*I_B
14
15 // I1=V_CC/(R1+R2) so let Rt=R1+R2 thus we get
16 Rt=V_CC/I1
17
18 // by Kirchoff voltage law to collector side we get
19 // V_CC=I_C*R_C+V_CE+I_E*R_E and also we have I_C=
       I_E so
20 // V_CC=I_C*R_C+V_CE+I_C*R_E so making R_E as
       subject we get
21 R_E=((V_CC-V_CE)/I_C)-R_C // in kilo ohm
22
23 V2=V_BE+I_C*R_E // since V_E=I_C*R_E
24 R2=V2/I1
25 R1=Rt-R2
26
27 printf("emitter resistance = %.3f ohm \n",R_E*1000)
28 printf("R1 = %3f kilo ohm \n",R1)
29 printf("R2 = %3f kilo ohm \n",R2)
```

Scilab code Exa 12.16 collector resistance

```
1  //chapter12
2  //example12.16
3  //page252
4
5  alpha=0.985
6  V_BE=0.3  // V
7  V_CC=16  // V
8  V_CE=6  // V
9  I_C=2  // mA
10 R_E=2  // kilo ohm
11 R2=20  // kilo ohm
12
13 gain_beta=alpha/(1-alpha)
14 I_B=I_C/gain_beta
15
16 V_E=I_C*R_E
17 V2=V_BE+V_E
18 V1=V_CC-V2
19
20 I1=V2/R2
21 R1=V1/I1
22
23 V_RC=V_CC-V_CE-V_E
24 R_C=V_RC/I_C
25
26 printf("R1 = %.3f kilo ohm \n",R1)
27 printf("collector resistance = %.3f kilo ohm \n",R_C
    )
```

Scilab code Exa 12.17 thevenin

```
1  // chapter12
2  // example12.17
3  // page253
4
5  Vcc =15 // V
6  Re =2 // kilo ohm
7  Rc =1 // kilo ohm
8  gain_beta =100
9  Vbe =0.7 // V
10 R1 =10 // kilo ohm
11 R2 =5 // kilo ohm
12
13 Eo = Vcc * R2 /( R1 + R2 )
14 Ro = R1 * R2 /( R1 + R2 )
15
16 printf ("thevenin voltage = %.3 f V \n",Eo)
17 printf ("thevenin resistance = %.3 f kilo ohm \n",Ro)
18
19 // here Eo=Ib*Ro+Vbe+Ie*Re
20 // now considering Ie=gain_beta*Ib, we can replace
       Ib=Ie/gain_beta
21 // Eo=(Ie/gain_beta)*Ro+Vbe+Ie*Re
22 // making Ie as subject we get
23 Ie =( Eo - Vbe )/( Re + Ro / gain_beta )
24
25 printf ("emitter current = %.3 f mA \n",Ie)
```

Scilab code Exa 12.18 `kilo ohm`

```
1  // chapter12
2  // example12.18
3  // page254
4
5  V_CC =10 // V
6  V_BE =0.2 // V
```

```
 7  I_E=2  // mA
 8  I_B=50d-3  // mA
 9  R_E=1  // kilo ohm
10  R2=10  // kilo ohm
11
12  V2=V_BE+I_E*R_E
13  I2=V2/R2
14
15  I1=I2+I_B
16  V1=V_CC-V2
17  R1=V1/I1
18
19  printf("R1 = %.3f kilo ohm \n",R1)
```

Scilab code Exa 12.19 `transistor`

```
1  //chapter12
2  //example12.19
3  //page255
4
5  printf(" i) if R2 is shorted, base will be grounded.
        It will be \n    left without forward bias and
       transistor \n    will be cutoff so output is zero
       .\n \n")
6  printf(" ii) if R2 is open,forward bias will be very
        high. The \n    collector current will be very
       high and collector \n    emitter voltage will be
       very low. \n \n")
7  printf(" iii) if R1 is shorted, transistor will be
       in saturation \n    due to excessive forward bias
       . The base will be at \n    Vcc and emitter will
       be slightly below Vcc.\n \n")
8  printf(" iv) if R1 is open, transistor will be
       without forward bias.\n    Hence transistor will
       be cutoff i.e. output will be zero. \n")
```

Scilab code Exa 12.20 circuit is not mid-point biased

```
1  //chapter12
2  //example12.20
3  //page256
4
5  Vcc=8 // V
6  Rb=360 // kilo ohm
7  Rc=2 // kilo ohm
8  gain_beta=100
9  Vbe=0.7 // V
10
11 // when Ic=0, Vce=Vcc i.e. Vce=8 and when Vce=0, Ic=
       Vcc/Rc i.e. Ic=8/2
12 // so equation of load line becomes Ic=-0.5*Vce+4
13
14 clf()
15 x=linspace(0,8,5)
16 y=-0.5*x+4
17 plot2d(x,y,style=3,rect=[0,0,9,5])
18 xtitle("dc load line","Vce(volts)","Ic(mA)")
19
20 // since Vcc=Ib*Rb+Vbe we get
21 Ib=(Vcc-Vbe)/Rb
22 Ic=Ib*gain_beta
23 Vce=Vcc-Ic*Rc
24
25 printf("the operating point is %.3f V and %.3f mA \n
       ",Vce,Ic)
26 if Vce<Vcc/2+0.1 | Vce>Vcc/2-0.1 // check if V_CE is
        nearly half of V_CC
27     printf("circuit is mid-point biased \n")
28 else
29     printf("circuit is not mid-point biased. \n")
```

```
30  end
```

Scilab code Exa 12.21 circuit is not mid-point biased

```
1  //chapter12
2  //example12.21
3  //page257
4
5  V_CC=10  // V
6  R1=12  // kilo ohm
7  R2=2.7  // kilo ohm
8  V_BE=0.7  // V
9  R_E=180d-3  // kilo ohm
10 R_C=620d-3  // kilo ohm
11
12 V2=V_CC*R2/(R1+R2)
13 I_E=(V2-V_BE)/R_E
14 I_C=I_E
15 V_CE=V_CC-I_C*(R_C+R_E)
16
17 printf("the operating point is %.3f V and %.3f mA \n
       ",V_CE,I_C)
18 if V_CE<V_CC/2+0.1 | V_CE>V_CC/2-0.1   // check if
       V_CE is nearly half of V_CC
19     printf("circuit is mid-point biased \n")
20 else
21     printf("circuit is not mid-point biased. \n")
22 end
23
24 // the accurate answer for collector current is
       6.315 mA but in book it is given as 6.33 mA
```

Scilab code Exa 12.22 base current

```
1  // chapter12
2  // example12.22
3  // page257
4
5  V_CC =10  // V
6  R1 =1.5  // kilo ohm
7  R2 =0.68  // kilo ohm
8  R_E =0.24  // kilo ohm
9  V_BE =0.7  // V
10 beta_min =100
11 beta_max =400
12
13 V2=V_CC*R2/(R1+R2)
14 I_E=(V2-V_BE)/R_E
15 I_C=I_E
16
17 beta_avg =(beta_min*beta_max)^0.5
18 I_B=I_E/(beta_avg+1)
19
20 printf("base current = %f micro ampere \n",I_B*1000)
21
22 // the accurate answer for base current is 50.151
      micro ampere but in book it is given as 49.75
      micro ampere
```

Scilab code Exa 12.23 collector cutoff current

```
1  // chapter12
2  // example12.23
3  // page258
4
5  gain_beta =40
6  I_C1 =2  // mA
7  t1 =25  // degrees
8  t2 =55  // degrees
```

```
 9  I_CBO1=5d-3 // mA
10
11  // for I_CBO=5 micro ampere at 25 degrees
12  I_CEO1=(1+gain_beta)*I_CBO1
13
14  I_CBO2=I_CBO1*2^((t2-t1)/10) // since it doubles
        every 10 degrees. So for t2−t1, it becomes 2^((t2
        −t1)/10) times.
15  I_CEO2=(1+gain_beta)*I_CBO2
16  I_C2=I_CEO2+I_C1
17  I_C_change=100*(I_C2-I_C1)/I_C1
18
19  // for I_CBO=0.1 micro ampere at 25 degrees
20  t1_dash=25 // degrees
21  t2_dash=55 // degrees
22  I_CBO1_dash=0.1d-3 // mA
23  I_C1_dash=2 // mA
24
25  I_CBO2_dash=I_CBO1_dash*2^((t2-t1)/10) // since it
        doubles every 10 degrees. So for t2−t1, it
        becomes 2^((t2−t1)/10) times.
26  I_CEO2_dash=(1+gain_beta)*I_CBO2_dash
27  I_C2_dash=I_CEO2_dash+I_C1_dash
28  I_C_change_dash=100*(I_C2_dash-I_C1_dash)/I_C1_dash
29
30  printf("collector cutoff current = %.3f mA \n \n",
        I_CEO1)
31  printf("percent change in zero signal current given
        that \nI_CBO=5 micro ampere at 25 degree is = %.3
        f percent \n \n",I_C_change)
32  printf("percent change in zero signal current given
        that \nI_CBO=0.01 micro ampere at 25 degree is =
        %.3f percent \n",I_C_change_dash)
```

Scilab code Exa 12.24 base current

```
1  //chapter12
2  //example12.24
3  //page259
4
5  alpha=0.99
6  I_E=1 // mA
7  t1=27 // degrees
8  t2=57 // degrees
9  I_CBO1=0.02d-3 // mA
10
11 I_CBO2=I_CBO1*2^((t2-t1)/6) // since it doubles
      every 6 degrees. So for t2-t1, it becomes 2^((t2-
      t1)/6) times.
12
13 I_C=alpha*I_E+I_CBO2
14 I_B=I_E-I_C
15
16 printf("base current = %.1f micro ampere",I_B*1000)
```

Scilab code Exa 12.25 since base voltage is zero

```
1  //chapter12
2  //example12.25
3  //page261
4
5  printf("since base voltage is zero, it means that
      there is no path \nfor current in the base
      circuit. So the transistor will be off i.e. I_C
      =0,I_E=0. \nSo V_C=10V and V_E=0.\nSo obvious
      fault is R1 is open.\n")
```

Scilab code Exa 12.26 voltage at base

```
1  //chapter12
2  //example12.26
3  //page261
4
5  R1=18 // kilo ohm
6  R2=4.7 // kilo ohm
7  Re=1 // kilo ohm
8  Vcc=10 // V
9
10 V_B=Vcc*R2/(R1+R2)
11
12 printf("voltage at base = %.3f V \n",V_B)
13 printf("The fact that V_C=10V and V_E is nearly
       equal to V_B reveals \nthat I_C=0 and I_E=0.So
       I_B drops to zero.So obvious fault is R_E is open
       . \n")
```

Chapter 13

Single stage transistor amplifiers

Scilab code Exa 1.1 collector current

```
1  // chapter 13
2  // example 13.1
3  // page 272
4
5  Rc=4 // kilo ohm
6  Vcc=10 // V
7  Ib_zero=10d-3 // mA
8  Ib_max=15d-3 // mA
9  Ib_min=5d-3 // mA
10 gain_beta=100
11
12 Ic_zero=Ib_zero*gain_beta
13 Ic_max=Ib_max*gain_beta
14 Ic_min=Ib_min*gain_beta
15
16 Vc_zero=Vcc-Ic_zero*Rc
17 Vc_max=Vcc-Ic_max*Rc
18 Vc_min=Vcc-Ic_min*Rc
19
```

```
20  printf ("As collector current increases from %.3 f mA
        to %.3 f mA \noutput voltage decreases from %.3 f V
         to %.3 f V \n", Ic_zero , Ic_max , Vc_zero , Vc_max )
21  printf ("As collector current decreases from %.3 f mA
        to %.3 f mA \noutput voltage increases from %.3 f V
         to %.3 f V \n", Ic_max , Ic_min , Vc_max , Vc_min )
22  printf ("Thus output voltage is 180 degrees out of
        phase from input voltage \n")
23
24  printf ("Note : \ni) input voltage and input current
        are in phase \nii) input voltage and output
        current are in phase \niii) output voltage is 180
         degrees out of phase with input voltage\n")
25
26
27  // plotting base current and collector current and
        output voltage in same graph using following code
         instead of xcos
28  clf ()
29  x=linspace (0 ,2*%pi ,100)
30  ib =5* sin (x) +10
31  ic =0.5* sin (x) +1
32  vc = -4* sin (x) +6
33  plot2d (x , ib , style =1 , rect =[0 ,0 ,20 ,20])
34  xtitle (" base current ( micro ampere ) − Black
                           collector current ( mA) −
      Blue                      output voltage (V) − Green
                           "," t")
35  plot2d (x , ic , style =2 , rect =[0 ,0 ,20 ,20])
36  plot2d (x , vc , style =3 , rect =[0 ,0 ,20 ,20])
```

Scilab code Exa 1.2 Thevenin circuit

```
1  // chapter 13
2  // example 13.2
```

```
3  // page 274
4
5  printf ("i) Refering to the Thevenin circuit, we see
       that voltage source \nis short and resistances
       except Rc and Re are bypassed.\nThus dc load = Rc
       + Re \n\n")
6  printf ("  Refering to ac equivalent circuit, Rc is
       parallel with Rl.\nThus ac load = Rc*Rl/(Rc+Rl) \
       n \n \n")
7  printf ("ii) Since Vcc=Vce+Ic*(Rc+Re) we get \n max
       Vce = Vcc and max Ic = Vcc/(Rc+Re) \n \n \n")
8  printf ("iii)On applying ac signal, collector current
        and collector emitter \nvoltage change about Q
       point.\nMaximum collector current = Ic.\nMaximum
       positive swing of ac collector emitter voltage =
       Ic*R_AC \n So total maximum collector emitter
       voltage = Vce+Ic*R_AC \n\nMaximum positive swing
       of ac collector current = Vce/R_AC so \nTotal
       maximum collector current = Ic+Vce/R_AC \n")
```

Scilab code Exa 13.3 the operating point

```
1  // chapter 13
2  // example 13.3
3  // page 278
4
5  Vcc=15  // V
6  Re=2  // kilo ohm
7  Rc=1  // kilo ohm
8  Rl=1  // kilo ohm
9  Vbe=0.7  // V
10
11  // dc load line
12
13      // when Ic=0, Vce=Vcc i.e. Vce=15 and when Vce
```

```
       =0,  Ic=Vcc/(Rc+Re) i.e.  Ic=15/3
14     // so  equation  of  load  line  becomes  Ic=-(1/3)*
          Vce+15
15
16         clf()
17     x=linspace(0,15,5)
18     y=-(1/3)*x+5
19     plot2d(x,y,style=3,rect=[0,0,16,6])
20     xtitle("dc  load  line-green            ac  load  line-
          blue","collector  emitter  voltage(volts)","
          collector  current(mA)")
21
22     V2=5  // V
23     // since  voltage  across  R2  is  V2=5 V  and  V2=Vbe+
          Ie*Re  we  get
24     Ie=(V2-Vbe)/Re
25     Ic=Ie
26     Vce=Vcc-Ic*(Rc+Re)
27
28     printf("the  operating  point  is  %.3f V  and  %.3f
          mA \n",Vce,Ic)
29
30
31 // ac  load  line
32
33     R_AC=Rc*Rl/(Rc+Rl) // ac  load
34     V_ce=Vce+Ic*R_AC // maximum  collector  emitter
            voltage
35     I_c=Ic+Vce/R_AC // maximum  collector  current
36     // the  equation  of  ac  load  line  in  terms  of  V_ce
            and  I_c  becomes
37     y=-(I_c/V_ce)*x+I_c
38      plot2d(x,y,style=2,rect=[0,0,10,20])
```

Scilab code Exa 13.4 `maximum collector`

```
1  // chapter 13
2  // example 13.4
3  // page 279
4
5  Vcc=20 // V
6  Re=0 // kilo ohm, since given as negligible
7  Rc=10 // kilo ohm
8  Rl=30 // kilo ohm
9  Vbe=0.7 // V
10
11 Vce=10 // mV
12 Ic=1 // mA
13
14 // dc load line
15
16     // when Ic=0, Vce=Vcc i.e. Vce=15 and when Vce
           =0, Ic=Vcc/(Rc+Re) i.e. Ic=20/10 mA
17     // so equation of load line becomes Ic=-(1/10)*
           Vce+2
18
19      clf()
20     x=linspace(0,20,5)
21     y=-(1/10)*x+2
22     plot2d(x,y,style=3,rect=[0,0,21,6])
23     xtitle("dc load line-green          ac load line-
           blue","collector emitter voltage(volts)","
           collector current(mA)")
24
25 // ac load line
26
27     R_AC=Rc*Rl/(Rc+Rl) // ac load
28     V_ce=Vce+Ic*R_AC // maximum collector emitter
            voltage
29     I_c=Ic+Vce/R_AC // maximum collector current
30     // the equation of ac load line in terms of V_ce
            and I_c becomes
31     x=linspace(0,V_ce,10)
32     y=-(I_c/V_ce)*x+I_c
```

```
33          plot2d(x,y,style=2,rect=[0,0,21,6])
```

Scilab code Exa 13.5 `plot`

```
1  // chapter 13
2  // example 13.5
3  // page 280
4
5  printf("operating point is (8V,1mA). During positive
          half cycle of \nac signal collector current
       swings from 1 mA to 1.5 mA \nand collector
       emitter voltage swings from 8 V to 7 V.\nThis is
       at A.During negative half cycle of \nac signal
       collector current swings from 1 mA to 0.5 mA \
       nand collector emitter voltage swings from 8 V to
        9 V.\nThis is at B. \n \n")
6
7  printf("Note : When ac signal is applied, ac signal
       collector current and \ncollector emitter voltage
        variations take place about Q point. \nAlso,
       operating point moves along load line.\n")
8
9  clf()
10 x=linspace(-3*%pi,-%pi,10)
11 plot(x,-0.5*sin(x)+1)
12
13 x=linspace(7,9,10)
14 plot(x,5-0.5*x)
15
16 x=linspace(-3*%pi,-%pi,10)
17 plot(-sin(x)+8,x)
18 plot(x,xgrid())
19 xtitle("collector current and collector emitter
       voltage swings","collector emitter voltage (volts
       )","collector current (mA)")
```

```
20  a=gca(); // Handle on axes entity
21  a.x_location = "origin";
22  a.y_location = "origin";
23
24  // Some operations on entities created by plot ...
25  a=gca();
26  a.isoview='on';
27  a.children // list the children of the axes : here
        it is an Compound child composed of 2 entities
28  poly1= a.children.children(2); //store polyline
        handle into poly1
29  poly1.foreground = 4; // another way to change the
        style...
30  poly1.thickness = 3;  // ...and the tickness of a
        curve.
31  poly1.clip_state='off' // clipping control
32  a.isoview='off';
```

Scilab code Exa 13.6 voltage gain

```
1   //chapter13
2   //example13.6
3   //page282
4
5   Rc=2 // kilo ohm
6   Rl=0.5 // kilo ohm
7   Rin=1 // kilo ohm
8   gain_beta=60
9
10  R_AC=Rc*Rl/(Rc+Rl)
11  Av=gain_beta*R_AC/Rin
12
13  printf("voltage gain  = %.3f \n",Av)
```

Scilab code Exa 13.7 output voltage

```
1  //chapter13
2  //example13.7
3  //page282
4
5  Rc=10  //  kilo  ohm
6  Rl=10  //  kilo  ohm
7  Rin=2.5  //  kilo  ohm
8  gain_beta=100
9  Vin=1  //  mV
10
11 R_AC=Rc*Rl/(Rc+Rl)
12 Av=gain_beta*R_AC/Rin
13
14 //  since  Av=Vout/Vin  we  get
15 Vout=Av*Vin
16
17 printf("output  voltage  =  %.3f  mV  \n",Vout)
```

Scilab code Exa 13.8 input impedence

```
1  //chapter13
2  //example13.8
3  //page282
4
5  del_Ib=10d-3  //  mA
6  del_Ic=1  //  mA
7  del_Vbe=0.02  //  V
8  Rc=5  //  kilo  ohm
9  Rl=10  //  kilo  ohm
10 gain_beta=60.0;
```

```
11
12  Ai=del_Ic/del_Ib
13  Rin=del_Vbe/del_Ib
14  R_AC=Rc*Rl/(Rc+Rl)
15  Av=gain_beta*R_AC/Rin
16  Ap=Av*Ai
17
18  printf(" current  gain  = %.3 f  \n", Ai)
19  printf(" input  impedence  = %.3 f  kilo  ohm  \n", Rin)
20  printf(" ac  load  = %.3 f  kilo  ohm  \n", R_AC)
21  printf(" voltage  gain  = %.3 f  \n", Av)
22  printf(" power  gain  = %.3 f  \n", Ap)
23
24  // the  accurate  answer  for  voltage  gain  =  166.667
        and  for  power  gain  =  16666.667  but  in  book  they
        are  given  as  165  and  16500  respectively.
```

Scilab code Exa 13.9 output voltage

```
1   //chapter13
2   //example13.9
3   //page283
4
5   Rc=3 // kilo ohm
6   Rl=6 // kilo ohm
7   Rin=0.5 // kilo ohm
8   Vin=1 // mV
9   gain_beta=50
10
11  R_AC=Rc*Rl/(Rc+Rl)
12  Av=gain_beta*R_AC/Rin
13
14  // since  Av=Vout/Vin  we  get
15  Vout=Av*Vin
16
```

```
17  printf("output voltage = %.3f mV \n",Vout)
```

Scilab code Exa 13.10 `circuit`

```
1  //chapter13
2  //example13.10
3  //page283
4
5  R1=1  //  kilo ohm
6  R2=2  //  kilo ohm
7  Vt=6  //  V
8
9  Vb=Vt*R1/(R1+R2)
10
11 if Vb==4
12     printf("circuit is operating properly \n")
13 else
14     printf("circuit is not operating properly
           because voltage at B should be %.1f V instead
           of 4 V \n",Vb)
15 end
```

Scilab code Exa 13.11 `ac emitter resistance`

```
1  //chapter13
2  //example13.11
3  //page284
4
5  R1=40  //  kilo ohm
6  R2=10  //  kilo ohm
7  Re=2  //  kilo ohm
8  Vcc=10  //  V
9  Vbe=0.7  //  V
```

```
10
11  V2=Vcc*R2/(R1+R2) // voltage across R2
12  Ve=V2-Vbe // voltage across Re
13  Ie=Ve/Re
14  re_dash=25/Ie
15
16  printf("ac emitter resistance = %.3f ohm \n",re_dash
        )
```

Scilab code Exa 13.12 voltage gain

```
1   //chapter13
2   //example13.12
3   //page286
4
5   R1=150 // kilo ohm
6   R2=20 // kilo ohm
7   Re=2.2 // kilo ohm
8   Rc=12 // kilo ohm
9   Vcc=20 // V
10  Vbe=0.7 // V
11
12  V2=Vcc*R2/(R1+R2) // voltage across R2
13  Ve=V2-Vbe // voltage across Re
14  Ie=Ve/Re
15  re_dash=1d-3*25/Ie // in kilo ohm
16  Av=Rc/re_dash
17
18  printf("voltage gain = %.3f \n",Av)
19
20  // the accurate answer is 360.642
```

Scilab code Exa 13.13 voltage gain

```
1  // chapter13
2  // example13.13
3  // page287
4
5  Rc =12  //  kilo  ohm
6  Rl =6  //  kilo  ohm
7  re_dash =33.3d-3  //  kilo  ohm
8
9  R_AC=Rc*Rl/(Rc+Rl)
10 Av=R_AC/re_dash
11
12 printf("voltage  gain  =  %.3f  \n",Av)
13
14 // the  accurate  answer  is  120.120
```

Scilab code Exa 13.14 input impedence of amplifier circuit

```
1  // chapter13
2  // example13.14
3  // page288
4
5  R1 =45  //  kilo  ohm
6  R2 =15  //  kilo  ohm
7  Re =7.5  //  kilo  ohm
8  Vcc =30  //  V
9  Vbe =0.7  //  V
10 gain_beta =200
11
12 V2=Vcc*R2/(R1+R2)  //  voltage  across  R2
13 Ve=V2-Vbe  //  voltage  across  Re
14 Ie=Ve/Re
15 re_dash =1d-3*25/Ie  //  in  kilo  ohm
16 Zin_base=gain_beta*re_dash
17 Zin=Zin_base*(R1*R2/(R1+R2))/(Zin_base+R1*R2/(R1+R2)
      )
```

```
18
19 printf(" input impedence of amplifier circuit = %.3 f
       kilo ohm \n",Zin)
20
21 // the accurate answer for input impedence is 3.701
       kilo ohm but in book it is given as 3.45 kilo ohm
```

Scilab code Exa 13.15 Class A

```
1 //chapter13
2 //example13.15
3 //page289
4
5 printf(" i ) Class A amplifier means that it raises
       voltage level of signal and its \nmode of
       operation is such that collector current flows
       for whole input signal. \n \n")
6 printf(" ii ) Audio voltage amplifier means it raises
       voltage level of audio signal \nand its mode of
       operation is class A. \n \n")
7 printf(" iii ) Class B power amplifier means that it
       raises power level of signal and its \nmode of
       operation is such that collector current flows
       for half cycle of input signal only. \n \n")
8 printf(" iv ) Class A transformer coupled power
       amplifier means that power amplification \nis
       being done, coupling is by tranformer and mode of
        operation is class A. \n")
```

Scilab code Exa 13.16 required input signal voltag

```
1 //chapter13
2 //example13.16
```

```
3 //page290
4
5 Ao=1000
6 Rout=1 // ohm
7 Rl=4 // ohm
8 Rin=2d3 // ohm
9 I2=0.5 // A
10
11 // here I2/I1=Ao*Rin/(Rout+Rl) so
12 I1=I2*(Rout+Rl)/(Ao*Rin)
13 V1=I1*Rin // in V
14
15 printf(" required input signal voltage = %.3f mV \n" ,
     V1*1d3)
```

Scilab code Exa 13.17 magnitude of output voltage

```
1 //chapter13
2 //example13.17
3 //page291
4
5 Ao=1000
6 Es=10d-3 // V
7 Rs=3d3 // ohm
8 Rin=7d3 // ohm
9 Rout=15 // ohm
10 Rl=35 // ohm
11
12 I1=Es/(Rs+Rin)
13 V1=I1*Rin
14 Av=Ao*Rl/(Rout+Rl)
15 // since V2/V1=Av, we get
16 V2=V1*Av
17
18 P2=V2^2/Rl
```

```
19  P1=V1^2/Rin
20  Ap=P2/P1
21
22  printf("magnitude of output voltage = %.2f V \n",V2)
23  printf("power gain = %.2f \n",Ap)
```

Scilab code Exa 13.18 required signal voltage

```
1   //chapter13
2   //example13.18
3   //page292
4
5   Av=80
6   Ai=120
7   V2=1 // V
8   Rout=1 // ohm
9   Rl=2 // ohm
10
11  V1=V2/Av // in V
12
13  // Av=Ao*Rl/(Rout+Rl) and Ai=Ao*Rin/(Rout+Rl) so
14  // Av/Ai=Rl/Rin hence
15  Rin=Rl*Ai/Av
16
17  I1=V1/Rin // in mA
18  Ap=Av*Ai
19
20  printf("required signal voltage = %.2f mV and
        current = %.2f micro ampere \n",V1*1d3,I1*1d3)
21  printf("power gain = %.3f \n",Ap)
```

Chapter 14

Multistage transistor amplifiers

Scilab code Exa 14.1 gain

```
1  // chapter  14
2  // example  14.1
3  // page  301
4
5  Av =20* log10 (30)
6
7  Pv =10* log10 (100)
8
9  printf (" voltage  gain  = %.3 f  db  \n" ,Av )
10 printf (" power  gain  = %.3 f  db  \n" ,Pv )
```

Scilab code Exa 14.2 gain

```
1  // chapter  14
2  // example  14.2
3  // page  301
4
5  Ap1 =40  // db
```

```
 6  Ap2 =43  // db
 7
 8  // since Ap = 10∗log10(power_gain), we get
 9  power_gain1 =10^(Ap1/10)
10  power_gain2 =10^(Ap2/10)
11
12  printf("power gain of 40 db = %.3f \n",power_gain1)
13  printf("power gain of 43 db = %.3f \n",power_gain2)
14
15  // the accurate answer for power gain of 43 db is
        19952 but in book it is given as 20000 db
```

Scilab code Exa 14.3 `total voltage gain`

```
 1  //chapter 14
 2  //example 14.3
 3  //page 301
 4
 5  Av1 =20∗log10(100)  // db
 6  Av2 =20∗log10(200)  // db
 7  Av3 =20∗log10(400)  // db
 8
 9  Av_total = Av1 + Av2 + Av3
10
11  printf("total voltage gain = %.3f db \n",Av_total)
```

Scilab code Exa 14.4 `Example 4`

```
 1  //chapter 14
 2  //example 14.4
 3  //page 302
 4
 5  gain_abs =30
```

```
6  n=5
7
8  Ap1=10*log10(gain_abs) // db
9  Ap_tot=Ap1*n
10 Ap_f=Ap_tot-10 // db
11
12 printf("total power gain = %.3f db \n",Ap_tot)
13 printf("power gain with negative feedback = %.3f db
       \n",Ap_f)
```

Scilab code Exa 14.5 Example 5

```
1  //chapter 14
2  //example 14.5
3  //page 302
4
5  P1=1.5 // W
6  P2=0.3 // W
7  Pi=10d-3 // W
8
9  // power gain at 2 kHz
10 Ap1=10*log10(P1/Pi)
11
12 // power gain at 20 Hz
13 Ap2=10*log10(P2/Pi)
14
15 Ap_diff=Ap1-Ap2
16 printf("fall in gain from 2 kHz to 20 Hz = %.3f db \
       n",Ap_diff)
```

Scilab code Exa 14.6 output voltage

```
1  //chapter 14
```

```
2 //example 14.6
3 //page 302
4
5 Av=15 // db
6 V1=0.8 // V
7
8 // since db voltage gain Av=20*log10(V2/V1) making
    V2 as subject we get
9
10 V2=V1*10^(Av/20)
11
12 printf("output voltage = %.2f V \n",V2)
```

Scilab code Exa 14.7 minimum value of load resistance

```
1 //chapter 14
2 //example 14.7
3 //page 302
4
5 Ao_db=70 // db
6 Av_db=67 // db
7 Rout=1.5 // kilo ohm
8
9 // since 20*log(Ao)-20*log(Av)=Ao_db-Av_db we get
10 // 20*log(Ao/Av) = Ao_db-Av_db so
11 // Ao/Av = 10^((Ao_db-Av_db)/20)
12 // and also Ao/Av=1+Rout/Rl since Av/Ao=Rl/(Rl+Rout)
13
14 // so making Rl as subject we get
15 Rl=Rout/(10^((Ao_db-Av_db)/20)-1)
16
17 printf("minimum value of load resistance = %.3f kilo
      ohm \n",Rl)
18
19 // the accurate answer is 3.636 kilo ohm
```

Scilab code Exa 14.8 `output voltage`

```
1  //chapter 14
2  //example 14.8
3  //page 303
4
5  gain_db=40 // db
6  Vin=10d-3 // mV
7  Rl=1 // kilo ohm
8
9  // we know that Vout/Vin=10^(gain_db/20) so making
       Vout as subject we get
10 Vout=Vin*10^(gain_db/20)
11 P_load=Vout^2/Rl
12
13 printf("output voltage = %.3f V \n",Vout)
14 printf("load power = %.3f mW \n",P_load)
```

Scilab code Exa 14.9 `frequency`

```
1  // chapter14
2  // example14.9
3  // page 303
4
5  // figure given in book is for reference only. It is
        not required to solve the example since the
       required details are very clearly specified in
       the problem statement.
6  // moreover more data is needed to plot the graph
       given in book.
7
```

```
8  Av_max1=2000 // for 2 kHz
9  Av_sqrt_2=1414 // for 10 kHz and 50 Hz
10
11 percent_Av_max1=70.7*Av_max1/100
12 printf("70.7 percent of maximum gain 2000 is = %.3f
       \n",percent_Av_max1)
13
14 if Av_sqrt_2==percent_Av_max1
15 printf("we observe that 70.7 percent of max gain
       2000 is 1414 \n")
16 printf("this gain 1414 is at 50 Hz and 10 kHz \n")
17 printf("so bandwidth = 50 Hz to 10 kHz \n \n")
18
19 printf("since frequency on lower side at which gain
       falls to \n70.7 percent is 50 Hz.So lower cutoff
       frequency = 50 Hz \n \n")
20 printf("since frequency on upper side at which gain
       falls to \n70.7 percent is 10 kHz.So upper cutoff
        frequency = 10 kHz \n \n")
21 else printf("data is insuficient for finding
       bandwidth and cutoff frequencies \n")
22 end
```

Scilab code Exa 14.10 gain

```
1 // chapter14
2 // example14.10
3 // page 305
4
5 Rc=500 // ohm
6 Rin=1d3 // ohm
7
8 // gain of second stage is 60 since it has no
       loading effect of any stage so
9 Av2=60
```

```
10  load1 = Rc * Rin /( Rc + Rin )
11  Av1 = Av2 * load1 / Rc
12  Av = Av1 * Av2
13
14  printf (" total  gain  =  %.3 f  \n" , Av )
15  printf (" comment  :  gain  of  one  stage =60. So  total  gain
         should  be  60*60=%d  but  here  it  is  %.3 f .\ nThis  is
          because  of  loading  effect  of  input  impedence  of
         second  stage  on  first  stage .  \n" ,60*60 , Av )
16  printf (" So  gain  of  first  stage  decreases .\ nHowever ,
         second  stage  has  no  loading  effect  of  any  next
         stage . So  its  gain  does  not  decrease .  \n" )
17
18  // the  accurate  answer  for  total  gain  is  2400  but  in
         book  it  is  given  as  2397
```

Scilab code Exa 14.11 gain

```
1  // chapter14
2  // example14.11
3  // page 306
4
5  Rin =1 // kilo ohm
6  Rc = 2 // kilo ohm
7  gain_beta =100
8
9  // since  first  stage  has  loading  effect  of  input
        impedence  of  second  stage ,  we  get  effective  load
        of  first  stage  as
10  R_AC = Rc * Rin /( Rc + Rin )
11  Av1 = gain_beta * R_AC / Rin
12
13  // second  stage  has  no  loading  effect  so  its  gain
14  Av2 = gain_beta * Rc / Rin
15  Av = Av1 * Av2
```

```
16
17 printf (" voltage gain of first stage = %.3 f \n",Av1)
18 printf (" voltage gain of second stage = %.3 f \n",Av2)
19 printf (" total voltage gain = %.3 f \n",Av)
20
21 // the accurate answer for gain of first stage is
       66.667 and total gain is 13333.33 but in book
       they are given as 66 and 13200 respectively
```

Scilab code Exa 14.12 voltage gain

```
1 // chapter14
2 // example14.12
3 // page 307
4
5 Rin =1d3 // ohm
6 Rc= 10d3 // ohm
7 Rl =100 // ohm
8 gain_beta =100
9
10 // effective collector load is
11 R_AC=Rc*Rl/(Rc+Rl)
12 Av=gain_beta*R_AC/Rin
13
14 printf (" voltage gain = %.3 f \n",Av)
15 printf (" comment : load is only 100 ohm so efective
       load of amplifier is too much reduced .\ nThus
       voltage gain is very small .\ n")
16 printf (" In such cases we can use a step down
       transformer to serve the purpose . \n")
```

Scilab code Exa 14.13 biasing potential

```
1  // chapter14
2  // example14.13
3  // page 307
4
5  Vcc=20 // V
6  R3=10 // kilo ohm
7  R4=2.2 // kilo ohm
8  Rc=3.6 // kilo ohm
9
10  V_B=Vcc*R4/(R3+R4)
11
12  // replacing Cc by wire
13  Req=R3*Rc/(R3+Rc)
14  V_B2=Vcc*R4/(Req+R4)
15
16  printf("biasing potential before replacing Cc = %.3f
        V \n",V_B)
17  printf("biasing potential after replacing Cc = %.3f
        V \n \n",V_B2)
18  printf("thus biasing potential of second stage
        changes.\nThis could cause the transistor to
        saturate and it would not work as amplifier.\n")
19  printf("Also, we see the use of coupling capacitor
        to maintain \nindependent biasing potential for
        each stage.\nThis allows ac output from one stage
         to pass to next stage.\n")
```

Scilab code Exa 14.14 voltage gain

```
1  // chapter14
2  // example14.14
3  // page 308
4
5  Vcc=15 // V
6  R1=22 // kilo ohm
```

```
 7  R2=3.3  // kilo ohm
 8  R3=5  // kilo ohm
 9  R4=1  // kilo ohm
10  R5=15  // kilo ohm
11  R6=2.5  // kilo ohm
12  R8=1  // kilo ohm
13  R3=5  // kilo ohm
14  R7=5  // kilo ohm
15  Rl=10  // kilo ohm
16  gain_beta=200
17  Vbe=0.7  // V
18
19  // for second stage
20  V_R6=Vcc*R6/(R6+R5)
21  V_R8=V_R6-Vbe
22  I_E2=V_R8/R8  // emitter current in R8
23  re_dash2=25d-3/I_E2
24  Zin_base=gain_beta*re_dash2
25  Zin=R5*(R6*Zin_base/(R6+Zin_base))/(R5+(R6*Zin_base
        /(R6+Zin_base)))
26  R_AC2=R7*Rl/(R7+Rl)
27  Av2=R_AC2/re_dash2
28
29  // for first stage
30  V_R2=Vcc*R2/(R2+R1)
31  V_R4=V_R2-Vbe
32  I_E1=V_R4/R4  // emitter current in R4
33  re_dash1=25d-3/I_E1
34  R_AC1=R3*Zin/(R3+Zin)
35  Av1=R_AC1/re_dash1
36
37  Av=Av1*Av2
38
39  printf("voltage gain of first stage = %.3f \n",Av1)
40  printf("voltage gain of second stage = %.3f \n",Av2)
41  printf("overall voltage gain= %.3f \n",Av)
42
43  // the accurate answers are voltage gain of first
```

```
      stage = 52.616, voltage gain of second stage =
      192.381, overall voltage gain= 10122.329. In book
       the answers are 53,191.4 and 10144
44 // respectively
```

Scilab code Exa 14.15 `transformer`

```
1  // chapter14
2  // example14.15
3  // page 311
4
5  // for maximum power transfer, primary impedence =
      transistor output impedence and secondary
      impedence = load impedence
6  Rp=1d3 // ohm
7  Rs=10 // ohm
8
9  // since Rp=(Np/Ns)^2*Rs, making Np/Ns i.e. n as
      subject we get
10 n=(Rp/Rs)^(0.5)
11
12 printf("required turn ratio = %d \n",n)
13
14 if n>1
15     printf("transformer required is step down
          tranformer \n")
16 elseif n<1
17     printf("transformer required is step up
          tranformer \n")
18 else // n=1
19     printf("transformer is not required \n")
20 end
```

Scilab code Exa 14.16 `voltage across external load`

```
1  // chapter14
2  // example14.16
3  // page 312
4
5  Vp=10  // V
6  // for maximum power transfer, primary impedence =
       output impedence of aource
7  Rp=10d3  // ohm
8  Rs=16  // ohm
9
10 // since Rp=(Np/Ns)^2*Rs, making Np/Ns i.e. n as
       subject we get
11 n=(Rp/Rs)^(0.5)
12
13 // since Vs/Vp=Ns/Np, making Vs as subject we get
14 Vs=(1/n)*Vp
15 printf("required turn ratio = %d \n",n)
16 printf("voltage across external load = %.3f V \n",Vs
       )
```

Scilab code Exa 14.17 `turn ratio for maximum power transfer`

```
1  // chapter14
2  // example14.17
3  // page 312
4
5  Rp=300  //ohm
6  Rs=3  // ohm
7  Ro=3d3  // ohm
8
9  // since output resistance of transistor Ro=Rp+n^2*
       Rs for maximum power transfer, making n as
       subject we get
```

```
10  n=((Ro-Rp)/Rs)^0.5
11
12  printf("turn ratio for maximum power transfer = %d \
        n",n)
```

Scilab code Exa 14.18 `inductance`

```
1  // chapter14
2  // example14.18
3  // page 313
4
5  f=200 // Hz
6  Ro=10d3 // ohm, transistor output impedence
7  Zi2=2.5d3 // ohm, input impedence of next stage
8
9  // since Ro=2*%pi*f*Lp, making Lp as subject we get
10  Lp=Ro/(2*%pi*f)
11
12  // since Zi2=2*%pi*f*Ls, making Ls as subject we get
13  Ls=Zi2/(2*%pi*f)
14
15  printf("primary inductance = %.1f H \n",Lp)
16  printf("secondary inductance = %.1f H \n",Ls)
```

Scilab code Exa 14.19 `inductance`

```
1  // chapter14
2  // example14.19
3  // page 313
4
5  L=10d-6 // H
6  N=1 // turn
7  Lp=8 // H
```

```
 8  Ls =2  // H
 9
10  // since L is proportional to N^2, L=K*N^2 so making
        K as subject we get
11  K = L / N ^2
12
13  // Lp=K*Np^2 so
14  Np =( Lp / K )^0.5
15
16  // Ls=K*Ns^2 so
17  Ns =( Ls / K )^0.5
18
19  printf (" primary turns = %d \n", Np )
20  printf (" secondary turns = %d \n", Ns )
```

Chapter 15

Transistor audio power amplifiers

Scilab code Exa 15.1 `maximum collector current`

```
1  //chapter15
2  //example15.1
3  //page321
4
5  V =12  // V
6  P =2  // W
7
8  // since P=V*Ic we get
9  Ic_max =P/V  // in ampere
10
11 printf("maximum collector current = %.3f mA \n",
       Ic_max *1000)
```

Scilab code Exa 15.2 `maximum collector current`

```
1  //chapter15
```

```
2  //example15 .2
3  //page321
4
5  V=12  //  V
6  R=4  //  kilo  ohm
7
8  //since  maximum  collector  current  will  flow  when
        whole  battery  voltgage  is  dropped  across  Rc,  we
        get
9  Ic_max=V/R
10
11  printf("maximum  collector  current  = %.3 f  mA  \n",
        Ic_max )
```

Scilab code Exa 15.3 `ac output`

```
1  //chapter15
2  //example15 .3
3  //page321
4
5  P=50  //  W
6  R=8  //  ohm
7
8  //  since  p=V^2/R  we  get
9  V=(P*R)^0.5
10  I=V/R
11
12  printf("ac  output  voltage  = %.3 f  V  \n",V)
13  printf("ac  output  current  = %.3 f  A  \n",I)
```

Scilab code Exa 15.4 `collector efficiency`

```
1  //chapter15
```

```
2 //example15.4
3 //page325
4
5 Vcc=20 // V
6 Vbe=0.7 // V
7 Rb=1d3 // ohm
8 Rc=20 // ohm
9 gain=25
10
11 Ib=(Vcc-Vbe)/Rb
12 Ic=Ib*gain
13 Vce=Vcc-Ic*Rc
14
15 ib_peak=10d-3
16 ic_peak=gain*ib_peak
17 Po_ac=ic_peak^2*Rc/2
18 P_dc=Vcc*Ic
19 eta=(Po_ac/P_dc)*100
20
21 printf("operating point = %.3f V and %.3f mA \n",Vce
        ,Ic*1000)
22 printf("output power = %.3f W \n",Po_ac)
23 printf("input power = %.3f W \n",P_dc)
24 printf("collector efficiency = %.3f percent \n",eta)
```

Scilab code Exa 15.5 collector efficiency

```
1 //chapter15
2 //example15.5
3 //page328
4
5 Pdc=10 // W
6 Po=4 // W
7
8 eta=(Po/Pdc)*100
```

```
 9
10  // maximum power dissipation in a transistor occurs
        under zero signal conditions so
11  P = Pdc
12
13  printf (" collector efficiency = %.3 f percent \n",eta)
14  printf (" power rating of transistor = %.3 f W \n",P)
```

Scilab code Exa 15.6 `maximum ac power output`

```
 1  // chapter15
 2  // example15.6
 3  // page328
 4
 5  Rl =100  // ohm
 6  n =10
 7  Ic =100 d -3  // ampere
 8
 9  Rl_1 = n ^2* Rl
10  Pmax =0.5* Ic ^2* Rl_1
11
12  printf (" maximum ac power output = %.3 f W \n",Pmax)
```

Scilab code Exa 15.7 `power dissipation`

```
 1  // chapter15
 2  // example15.7
 3  // page329
 4
 5  Vcc =5  // V
 6  Ic =50 d -3  // ampere
 7
 8  Pac_max = Vcc * Ic /2
```

```
 9  Pdc=Vcc*Ic
10  Pdis=Pac_max*2
11  eta=(Pac_max/Pdc)*100
12
13  printf("maximum power output= %.3f mW \n",Pac_max
        *1000)
14  printf("power dissipation = %.3f mW \n",Pdis*1000)
15  printf("maximum collector efficiency = %.3f percent
         \n",eta)
```

Scilab code Exa 15.8 `loudspeaker`

```
 1  //chapter15
 2  //example15.8
 3  //page329
 4
 5  del_Ic=100d-3 // ampere
 6  del_Vce=12 // V
 7  Rl=5 // ohm
 8
 9  //case 1 : louspeaker directly connected
10  V=del_Ic*Rl
11  P=V*del_Ic
12
13  //case2 : loudspreaker transformer coupled
14  R_primary=del_Vce/del_Ic // for maximum power
        transfer the primary resistance should be equal
        to R
15  n=(R_primary/Rl)^0.5
16  V_secondary=del_Vce/n
17  Il=V_secondary/Rl
18  P_1=Il^2*Rl
19
20  printf("case1 : loudspeaker connected directly \n
            power transferred to loudspeaker = %.3f mW \
```

```
         n", P*1000)
21  printf("case2 : loudspeaker is transformer coupled \
        n         power transferred to loudspeaker = %.3 f mW
        \n", P_1*1000)
```

Scilab code Exa 15.9 `current`

```
 1  // chapter15
 2  // example15.9
 3  // page331
 4
 5  Vcc =17.5 // V
 6  ic_max =35d-3 // ampere
 7  ic_min =1d-3 // ampere
 8  IC =18 // ampere
 9  gain =100
10  vce_max =30 // V
11  vce_min =5 // V
12  Rl =81.6 // ohm
13
14  IC=ic_min +(( ic_max - ic_min )/2)
15  IB=IC/gain
16  VCE=vce_min +(( vce_max - vce_min )/2)
17
18  Pdc=Vcc*IC
19  Vce =( vce_max - vce_min )/(2*2^0.5)
20  Ic =( ic_max - ic_min )/(2*2^0.5)
21  Pac=Vce*Ic
22
23  eta =( Pac/Pdc )*100
24
25  slope =( ic_max - ic_min )/( vce_min - vce_max )
26  Rl_dash =-1/ slope
27  n=( Rl_dash /Rl )^0.5
28
```

```
29  printf("zero signal collector current = %.3f mA \n",
        IC*1000)
30  printf("zero signal base current = %.3f mA \n",IB
        *1000)
31  printf("dc power = %.3f mW and ac power = %.3f mW \n
        ",Pdc*1000,Pac*1000)
32  printf("collector efficiency = %.3f percent \n",eta)
33  printf("transformer turn ratio = %.1f \n",n)
```

Scilab code Exa 15.10 maximum ambient temperature at which transistor can be opera

```
1  //chapter15
2  //example15.10
3  //page333
4
5  P_total=4 // W
6  T_Jmax=90 // degree celcius
7  theta=10 // degree celcius per watt
8
9  // P_total=(T_Jmax-Tamb)/theta so making Tamb as
       subject we get
10 Tamb=T_Jmax-P_total*theta
11
12 printf("maximum ambient temperature at which
       transistor can be operated = %.3f degree C \n",
       Tamb)
```

Scilab code Exa 15.11 maximum power dissipation

```
1  //chapter15
2  //example15.11
3  //page333
4
```

```
5  T_Jmax=90 // degree celcius
6  T_amb=30 // degree celcius
7
8  //case 1 : without heat sink
9  theta1=300 // degree celcius per watt
10 P_total1=(T_Jmax-T_amb)/theta1
11
12 //case 2 : with heat sink
13 theta2=60 // degree celcius per watt
14 P_total2=(T_Jmax-T_amb)/theta2
15
16 printf("case 1 : without heat sink \n    maximum
       power dissipation = %.3f mW \n",P_total1*1000)
17 printf("case 2 : with heat sink \n    maximum power
       dissipation = %.3f mW \n",P_total2*1000)
```

Scilab code Exa 15.12 allowed collector current

```
1  //chapter15
2  //example15.12
3  //page334
4
5  T_Jmax=200 // degree celcius
6  T_amb1=25 // degree celcius
7  T_amb2=75 // degree celcius
8  theta=20 // degree celcius per watt
9  Vcc=4 // V
10
11 P_total1=(T_Jmax-T_amb1)/theta
12 Ic1=P_total1/Vcc
13
14 P_total2=(T_Jmax-T_amb2)/theta
15 Ic2=P_total2/Vcc
16
17 printf("for ambient = 25 degree C, allowed collector
```

```
        current = %.3 f A \n" ,Ic1 )
18  printf (" for  ambient = 75  degree  C,  allowed  collector
        current = %.3 f A \n" ,Ic2 )
```

Chapter 16

Amplifiers with negative feedback

Scilab code Exa 16.1 voltge gain with negative feedback

```
1  // chapter16
2  // example16 .1
3  // page345
4
5  Av =3000
6  mv =0.01
7
8  Avf = Av /(1+ Av * mv )
9  printf (" voltge  gain  with  negative  feedback  = %.3 f  \n
      " , Avf )
10
11 //  accurate  answer  is  96.774  but  in  book  it  is  given
       as  97
```

Scilab code Exa 12.2 fraction of output fed back to input

```
1  // chapter16
2  // example16 .2
3  // page346
4
5  Av =140
6  Avf =17.5
7
8  // since Avf=Av/(1+Av*mv) , making mv as subject we
        get
9  mv =( Av / Avf -1)/ Av
10 printf ("fraction of output fed back to input = %.3 f
        \n", mv )
```

Scilab code Exa 16.3 overall gain

```
1  // chapter16
2  // example16 .3
3  // page346
4
5  Av1 =100
6  Avf1 =50
7  Avf2 =75
8
9  // since Avf=Av/(1+Av*mv) , we get
10 mv =( Av1 / Avf1 -1)/ Av1
11 Av2 = Avf2 /(1 - mv * Avf2 )
12
13 printf ("fraction of output fed back to input = %.3 f
        \n", mv )
14 printf ("for overall gain = 75 and same fraction ,
        required gain = %.3 f \n", Av2 )
```

Scilab code Exa 16.4 fraction of output fed back to input

```
1  // chapter16
2  // example16.4
3  // page346
4
5  Vo =10
6  Vi =0.25
7  Vif =0.5
8
9  Av = Vo / Vi
10 Avf = Vo / Vif
11
12 // since Avf=Av/(1+Av*mv) , we get
13 mv =( Av / Avf -1)/ Av
14
15 printf (" fraction of output fed back to input = %.3 f
       \n" , mv )
```

Scilab code Exa 16.5 percentage reduction in gain

```
1  // chapter16
2  // example16.5
3  // page347
4
5  Av =50
6  Avf =25
7
8  // since Avf=Av/(1+Av*mv) , we get
9  mv =( Av / Avf -1)/ Av
10
11 // without feedback , gain falls from 50 to 40
12 Av1 =50
13 Av2 =40
14 reduction1 =100*( Av1 - Av2 )/ Av1
15
16 // with feedback
```

```
17  Av3 =25
18  Av4 = Av2 /(1+ mv * Av2 )
19  reduction2 =100*( Av3 - Av4 )/ Av3
20
21  printf ("percentage reduction in gain : \n with
        feedback = %.3 f percent \n ",reduction1)
22  printf ("without feedback = %.3 f perent",reduction2)
```

Scilab code Exa 16.6 `percentage change in gain`

```
1  //chapter16
2  //example16.6
3  //page347
4
5  Av =100
6  mv =0.1
7
8  Avf = Av /(1+ Av * mv )
9  mv =( Av / Avf -1)/ Av
10
11  // fall in gain is 6dB so 20 log (Av/Av1 )=6
12  // making Av1 as subject we get
13  Av1 = Av / exp (6* log (10) /20)
14  Avf _new = Av1 /(1+ Av1 * mv )
15  change =100*( Avf - Avf_new )/ Avf
16
17  printf ("percentage change in gain = %.3 f percent \n"
        ,change)
18
19  // the accurate answer is 8.297 percent but in book
        it is given as 8.36 percent
```

Scilab code Exa 16.7 `voltage gain with negative feedback`

```
1  // chapter16
2  // example16.7
3  // page348
4
5  A0=1000
6  Rout=100  // ohm
7  Rl=900
8  mv=1/50
9
10 Av=A0*Rl/(Rout+Rl)
11 Avf=Av/(1+Av*mv)
12 printf("voltage gain with negative feedback = %.3f \
      n",Avf)
13
14 // the accurate answer is 47.368 but in book it is
      given as 47.4
```

Scilab code Exa 16.8 voltage gain with negative feedback

```
1  // chapter16
2  // example16.8
3  // page351
4
5  Av=10000
6  R1=2  // kilo ohm
7  R2=18  // kilo ohm
8  Vi=1  // mV
9
10 mv=R1/(R1+R2)
11 Avf=Av/(1+Av*mv)
12 Vout=Avf*Vi
13
14 printf("feedback fraction  = %.1f \n",mv)
15
16 printf("voltage gain with negative feedback = %.1f \
```

```
    n" , Avf )
17
18 printf ( "output voltage = %.1 f mV \n" , Vout )
```

Scilab code Exa 16.9 Example 9

```
1 // chapter16
2 // example16 .9
3 // page351
4
5 Av =10000
6 R1 =10  // kilo ohm
7 R2 =90  // kilo ohm
8 Zin =10  // kilo ohm
9 Zout =100 d -3  // kilo ohm
10
11 mv = R1 /( R1 + R2 )
12 Avf = Av /(1+ Av * mv )
13 Zin_dash =(1+ Av * mv )* Zin
14 Zout_dash = Zout /(1+ Av * mv )
15
16 printf ( "feedbackfraction   = %.1 f  \n" , mv )
17
18 printf ( "voltage gain with negative feedback = %.1 f \
      n" , Avf )
19
20 printf ( "input impedence with feedback = %.3 f kilo
      ohm or %.3 f mega ohm \n" , Zin_dash , Zin_dash /1000)
21
22 printf ( "output impedence with feedback = %f kilo ohm
       or %.3 f ohm \n" , Zout_dash , Zout_dash *1000)
```

Scilab code Exa 16.10 distortion of amplifier with negative feedback

```
1  //chapter16
2  //example16.10
3  //page352
4
5  Av=150
6  D=5/100
7  mv=10/100
8
9  Dvf=100*D/(1+Av*mv) // in percent
10
11 printf("distortion of amplifier with negative
       feedback = %.3f percent",Dvf)
```

Scilab code Exa 16.11 cutoff frequency with negative feedback

```
1  //chapter16
2  //example16.11
3  //page352
4
5  Av=1000
6  mv=0.01
7  f1=1.5 // kHz
8  f2=501.5 // kHz
9
10 f_1f=f1/(1+Av*mv)
11 f_2f=f2*(1+mv*Av)
12
13 printf("new lower cutoff frequency with negative
       feedback = %.3f kHz or %.3f Hz \n",f_1f,f_1f
       *1000)
14 printf("new upper cutoff frequency with negative
       feedback = %.3f kHz or %.3f MHz \n",f_2f,f_2f
       /1000)
15
16 // the accurate answers are 136.364 Hz and 5.516 MHz
```

Scilab code Exa 16.12 effective current gain of amplifier

```
1  //chapter16
2  //example16.12
3  //page353
4
5  Ai =200
6  mi =0.012
7
8  Aif = Ai /(1+ mi * Ai )
9
10 printf (" effective current gain of amplifier = %.3 f \
       n",Aif )
```

Scilab code Exa 16.13 input impedence with negative feedback

```
1  //chapter16
2  //example16.13
3  //page354
4
5  Zin =15//  kilo ohm
6  Ai =240
7  mi =0.015
8
9  Zin_dash = Zin /(1+ mi * Ai )
10
11 printf (" input impedence with negative feedback = %.3
       f  kilo ohm \n", Zin_dash )
```

Scilab code Exa 16.14 `output impedence with negative feedback`

```
1  //chapter16
2  //example16.14
3  //page355
4
5  Zout=3 // kilo ohm
6  Ai=200
7  mi=0.01
8
9  Zout_dash=Zout*(1+mi*Ai)
10
11 printf("output impedence with negative feedback = %
      .3f kilo ohm \n",Zout_dash)
```

Scilab code Exa 16.15 `Bandwidth with negative feedback`

```
1  //chapter16
2  //example16.15
3  //page355
4
5  BW=400 // kHz
6  Ai=250
7  mi=0.01
8
9  BW_dash=BW*(1+mi*Ai)
10
11 printf("Bandwidth with negative feedback = %.3f kHz
      \n",BW_dash)
```

Scilab code Exa 16.16 `voltage across`

```
1  // chapter16
2  // example16.16
3  // page356
4
5  Vcc =18 // V
6  R1 =16 // kilo ohm
7  R2 =22 // kilo ohm
8  Vbe =0.7 // V
9  Re =910 d -3 // kilo ohm
10
11 V2 = Vcc * R2 /( R1 + R2 )
12 Ve = V2 - Vbe
13 Ie = Ve / Re
14
15 printf (" voltage across Re = %.3 f V \n" ,Ve )
16 printf (" emitter current = %.3 f mA \n" ,Ie )
17
18 clf ()
19 x = linspace (0 ,18 ,100)
20 y = -(19.78/18) *x +19.78
21 xtitle (" dc load line " ," Vce ( volts )" ," Ic ( mA )")
22 plot2d (x ,y , style =3 , rect =[0 ,0 ,19 ,20])
```

Scilab code Exa 16.17 `voltage gain`

```
1  // chapter16
2  // example16.17
3  // page357
4
5  Vcc =10 // V
6  R1 = 10 // kilo ohm
7  R2 =10 // kilo ohm
8  Vbe =0.7 // V
```

```
 9  Re =5000 // ohm
10
11  V2= Vcc * R2 /( R1 + R2 )
12  Ve = V2 - Vbe
13  Ie = Ve /( Re /1000) // in mA
14  re_dash =25/ Ie
15  Av = Re /( re_dash + Re )
16
17  printf (" voltage gain = %.3 f \n", Av )
```

Scilab code Exa 16.18 voltage gain

```
 1  // chapter16
 2  // example16 .18
 3  // page358
 4
 5  Re =5 d3 // ohm
 6  Rl =5 d3 // ohm
 7  re_dash =29.1 // in ohm , from example 16_17
 8
 9  Re_dash = Re * Rl /( Re + Rl )
10  Av = Re_dash /( re_dash + Re_dash )
11
12  printf (" voltage gain = %.3 f \n", Av )
```

Scilab code Exa 16.19 input impedence

```
 1  // chapter16
 2  // example16 .19
 3  // page359
 4
 5  Vcc =10 // V
 6  R1 = 10 // kilo ohm
```

149

```
 7  R2 =10 // kilo ohm
 8  Vbe =0.7 // V
 9  Re =4.3 // kilo ohm
10  gain_beta =200
11
12  V2 = Vcc * R2 /( R1 + R2 )
13  Ve = V2 - Vbe
14  Ie = Ve / Re
15  re_dash =25/ Ie
16  Re_dash = Re * R1 /( Re + R1 )
17  Zin_base = gain_beta *( re_dash + Re_dash )
18  Zin = Zin_base *( R1 * R2 /( R1 + R2 )) /( Zin_base + R1 * R2 /( R1 + R2 )
        )
19
20  printf ( " input impedence = %.3 f kilo ohm \n " , Zin )
21
22  // the accurate answer is 4.996 kilo ohm but in book
         it is given as 4.96 kilo ohm
```

Scilab code Exa 16.20 output impedence

```
 1  // chapter16
 2  // example16.20
 3  // page361
 4
 5  R1 =3 d3 // ohm
 6  R2 =4.7 d3 // ohm
 7  Rs =600 // ohm
 8  re_dash =20 // ohm
 9  gain_beta =200
10
11  Rin_dash = R1 *( R2 * Rs /( R2 + Rs )) /( R1 +( R2 * Rs /( R2 + Rs )))
12
13  Zout = re_dash + Rin_dash / gain_beta
14
```

```
15  printf("output impedence = %.1f ohm \n",Zout)
```

Chapter 17

Sinusoidal oscillations

Scilab code Exa 17.1 frequency of oscillations

```
1  //chapter17
2  //example17.1
3  //page375
4
5  L1=58.6d-6 // H
6  C1=300d-12 // F
7
8  f=1/(2*%pi*(L1*C1)^0.5)
9  printf("frequency of oscillations = %.3f Hz or %.3f
       kHz",f,f/1000)
10
11 // in book the answer is 1199 kHz but the accurate
       answer is 1200.358 kHz
```

Scilab code Exa 17.2 operating frequency

```
1  //chapter17
2  //example17.2
```

```
 3  //page377
 4
 5  C1=0.001d-6 // F
 6  C2=0.01d-6 // F
 7  L=15d-6 // H
 8
 9  Ct=C1*C2/(C1+C2) // since both are in series
10
11  f=1/(2*%pi*(L*Ct)^0.5)
12  mv=C1/C2
13
14  printf("operating frequency = %.3f Hz or %.3f kHz \n
        ",f,f/1000)
15  printf("feedback function = %.3f",mv)
16
17  //in book the answer given is 1361 kHz but accurate
        answer is 1362.922 kHz
```

Scilab code Exa 17.3 `feedback function`

```
 1  //chapter17
 2  //example17.3
 3  //page379
 4
 5  L1=1000d-6 // H
 6  L2=100d-6 // H
 7  M=20d-6 // H
 8  C=20d-12 // F
 9
10  Lt=L1+L2+2*M
11
12  f=1/(2*%pi*(Lt*C)^0.5)
13  mv=L2/L1
14
15  printf("operating frequency = %.3f Hz or %.3f kHz \n
```

```
    ",f,f/1000)
16 printf("feedback function = %.3f",mv)
17
18 //in book the answer is 1052 kHz but the accurate
      answer is 1054.029 kHz
```

Scilab code Exa 17.4 frequency of oscillations

```
1 //chapter17
2 //example17.4
3 //page381
4
5 R=1d6  // ohm
6 C=68d-12  // F
7
8 fo=1/(2*%pi*R*C*(6)^0.5)
9 printf("frequency of oscillations = %.3f Hz",fo)
10
11 // in book the answer given is 954 Hz but the
      accurate answer is 955.511 Hz
```

Scilab code Exa 17.5 frequency of oscillations

```
1 //chapter17
2 //example17.5
3 //page382
4
5 R=220d3  // ohm
6 C=250d-12  // F
7
8 f=1/(2*%pi*R*C)
9 printf("frequency of oscillations = %.3f Hz",f)
10
```

```
11  //in book the answer given is 2892 Hz but the
        accurate answer is 2893.726 Hz
```

Scilab code Exa 17.6 `thickness of crystal`

```
1  //chapter17
2  //example17.6
3  //page387
4
5  // frequency is inversely proportional to thickness
6  // so if thickness is reduced by 1%, frequency
       increases by 1%
7
8  printf("If thickness of crystal is reduced by 1
       percent, then \nfrequency is increased by 1
       percent \nbecause frequency is inversely
       proportional to thickness \n")
```

Scilab code Exa 17.7 `resonant frequency`

```
1  //chapter17
2  //example17.7
3  //page387
4
5  L=1 // H
6  C=0.01d-12 // F
7  Cm=20d-12 // F
8
9  fs=1/(2*%pi*(L*C)^0.5)
10 Ct=C*Cm/(C+Cm)
11 fp=1/(2*%pi*(L*Ct)^0.5)
12
```

```
13  printf("series resonant frequency = %.3f Hz   or %.3f
        kHz\n",fs,fs/1000)
14  printf("parallel resonant frequency = %.3f Hz or %.3
        f kHz\n",fp,fp/1000)
15
16  // in book the answer given is 1589 kHz for series
        resonant frequency but the accurate answer is
        1591.549 kHz
17  // in book the answer given is 1590 kHz for parallel
        resonant frequency but the accurate answer is
        1591.947 kHz
```

Chapter 18

Transistor tuned amplifiers

Scilab code Exa 18.1 impedence of circuit at resonance

```
1  //chapter18
2  //example18.1
3  //page396
4  L=1.25d-3  // H
5  C=250d-12  // F
6  R=10  // ohm
7
8  fr=(((1/(L*C))-(R^2/L^2))^0.5)/(2*%pi)
9  Zr=L/(C*R)
10 Q=2*%pi*fr*L/R
11
12 printf("resonant frequency of circuit = %.3f Hz or %
       .3f kHz \n",fr,fr/1000)
13 printf("impedence of circuit at resonance = %.3f ohm
        or %.3f kilo ohm \n",Zr,Zr/1000)
14 printf("quality factor of the circuit = %.3f",Q)
```

Scilab code Exa 18.2 circuit

```
 1  //chapter18
 2  //example18.2
 3  //page396
 4
 5  L=100d-6 // H
 6  C=100d-12 // F
 7  R=10 // ohm
 8  V=10 // V
 9
10  fr=(((1/(L*C))-(R^2/L^2))^0.5)/(2*%pi)
11  Zr=L/(C*R)
12  I=V/Zr
13
14  printf("resonant frequency of circuit = %.3f Hz or %
        .3f kHz \n",fr,fr/1000)
15  printf("impedence of circuit at resonance = %.3f ohm
         or %.3f kilo ohm or %.3f mega ohm\n",Zr,Zr/1000,
        Zr/1d6)
16  printf("line current = %.4f ampere or %.3f micro
        ampere",I,I*1d6)
17
18  // the accurate answer for resonant frequency is
        1591.470 kHz
```

Scilab code Exa 18.3 frequency

```
 1  //chapter18
 2  //example17.3
 3  //page398
 4
 5  fr=1200 // kHz
 6  Q=60
 7
 8  BW=fr/Q
 9  f1=fr-(BW/2)
```

```
10  f2=fr+(BW/2)
11
12  printf("bandwidth = %.3f kHz \n",BW)
13  printf("lower cut−off frequency = %.3f kHz \n",f1)
14  printf("upper cut−off frequency = %.3f kHz \n",f2)
```

Scilab code Exa 18.4 `frequency`

```
1  //chapter18
2  //example18.4
3  //page401
4
5  L=33d-3 // H
6  C=0.1d-6 // F
7  R=25 // ohm
8
9  fr=1/(2*%pi*(L*C)^0.5)
10  Xl=2*%pi*fr*L
11  Q=Xl/R
12  BW=fr/Q
13
14  printf("resonant frequency = %.3f Hz or %.3f kHz \n"
        ,fr,fr/1000)
15  printf("quality factor = %.3f \n",Q)
16  printf("bandwidth = %.3f Hz \n",BW)
17
18  // the accurate answer for bandwidth is 120.572 Hz
        but in book it is given as 120 Hz
19  // the accurate answer for quality factor is 22.978
        but in book it is given as 23
```

Scilab code Exa 18.5 `co-efficient of coupling`

```
1  // chapter18
2  // example18.5
3  // page402
4
5  BW=200  // kHz
6  fr=10d3  // kHz
7
8  k=BW/fr
9
10 printf("co-efficient of coupling = %.3f \n",k)
```

Scilab code Exa 18.6 `ac load`

```
1  // chapter18
2  // example18.6
3  // page405
4
5  L=50.7d-6  // H
6  C=500d-12  // F
7  R=10  // ohm
8  Rl=1d6  // ohm
9
10 fr=1/(2*%pi*(L*C)^0.5)
11 R_dc=R
12 Xl=2*%pi*fr*L
13 Q=Xl/R
14 Rp=Q*Xl
15 R_ac=Rp*Rl/(Rp+Rl)
16
17 printf("resonant frequency = %.3f Hz or %.3f kHz \n"
       ,fr,fr/1000) // amswer in book is incorrect
18 printf("dc load = %.3f ohm \n",R_dc)
19 printf("ac load = %.3f ohm or %.3f kilo ohm \n",R_ac
       ,R_ac/1000)
20
```

```
21  //  in  book  the  aswer  for  resonant  frequency  is  106
        Hz  which  is  incorrect
22  //  the  correct  answer  is  999.611  kHz
23
24  //  the  accurate  answer  for  ac  load  is  10.038  kilo
        ohm
```

Scilab code Exa 18.7 `load`

```
1  // chapter18
2  // example18 .7
3  // page406
4
5  Vcc =50  //  V
6  Np =5
7  Ns =1
8  R =50  //  ohm
9  R_ac =( Np / Ns ) ^2* R
10  Po = Vcc ^2/ R_ac
11
12  printf (" ac  load  =  %.3 f  ohm  \n", R_ac )
13  printf (" maximum  load  power  =  %.3 f  W  \n", Po )
```

Chapter 19

Modulation and demodulation

Scilab code Exa 19.1 `Vmax\&Vmin`

```
1 // chapter19
2 // example19.1
3 // page416
4
5 // the figure in book is for reference only as
      equations for Ec and Es are already explained in
      the theory in the book.
6
7 printf ("Ec=(Vmax+Vmin)/2 \n")
8 printf ("Es=(Vmax-Vmin)/2 \n")
9 printf ("But, Es=m*Ec \n")
10 printf ("So (Vmax-Vmin)/2 = m*(Vmax+Vmin)/2 \n")
11 printf ("thus, m = (Vmax-Vmin)/(Vmax+Vmin) \n")
```

Scilab code Exa 19.2 `modulation factor`

```
1 // chapter19
2 // example19.2
```

```
 3  // page416
 4
 5  // figure is given in book for understanding purpose
        only.It is not required for solving the example
       as maximum and minimum peak voltages are given in
        the problem statement itself.
 6
 7  Vmax_pp=16 // mV
 8  Vmin_pp=4 // mV
 9
10  Vmax=Vmax_pp/2
11  Vmin=Vmin_pp/2
12
13  m=(Vmax-Vmin)/(Vmax+Vmin)
14
15  printf("modulation factor = %.3f \n",m)
```

Scilab code Exa 19.3 `modulation factor`

```
 1  // chapter19
 2  // example19.3
 3  // page417
 4
 5  Es=50 // V
 6  Ec=100 // V
 7
 8  m=Es/Ec
 9
10  printf("modulation factor = %.3f \n",m)
```

Scilab code Exa 19.4 `Bandwidth for RF amplifier`

```
 1  // chapter19
```

```
2 //example19.4
3 //page419
4
5 fc=2500 // kHz
6 fs_min=0.05 // kHz
7 fs_max=15 // kHz
8
9 upper_sideband_min=fc+fs_min
10 upper_sideband_max=fc+fs_max
11
12 lower_sideband_min=fc-fs_min
13 lower_sideband_max=fc-fs_max
14
15 BW=upper_sideband_max-lower_sideband_max
16
17 printf("lower sideband is from %.3f to %.3f kHz \n",
      lower_sideband_min,lower_sideband_max)
18 printf("upper sideband is from %.3f to %.3f kHz \n",
      upper_sideband_min,upper_sideband_max)
19 printf("Bandwidth for RF amplifier = %.3f kHz \n",BW
      )
```

Scilab code Exa 19.5 amplitudes of components

```
1 //chapter19
2 //example19.5
3 //page420
4
5 // v=5*(1+0.6*cos(6280*t))*sin(211d4*t) V
6 // compare with v=Ec*(1+m*cos(ws*t))*sin(wc*t) we
      get
7 Ec=5 // V
8 m=0.6
9 fs=6280/(2*%pi) // Hz
10 fc=211d4/(2*%pi) // Hz
```

```
11
12  Vmin=Ec-m*Ec
13  Vmax=Ec+m*Ec
14
15  f1=(fc-fs)/1000  // in kHz
16  f2=fc/1000  // in kHz
17  f3=(fc+fs)/1000  // in kHz
18
19  V1=m*Ec/2
20  V2=Ec
21  V3=m*Ec/2
22
23  printf("minimum amplitude = %.3f V and maximum
          amplitude = %.3f V \n",Vmin,Vmax)
24  printf("frequency components = %.1f kHz, %.1f Hz, %
          .1fkHz \n",f1,f2,f3)
25  printf("amplitudes of components = %.3f V, %.3f V, %
          .3f V \n",V1,V2,V3)
26
27  // in book there is error of 0.2 kHz in every
          frequency component. The accurate answers are
          334.8,335.8,336.8 kHz
```

Scilab code Exa 19.6 amplitude of each sideband term

```
1  //chapter19
2  //example19.6
3  //page420
4
5  fc=1000  // kHz
6  fs=5  // kHz
7  m=0.5
8  Ec=100  // V
9
10  lower_sideband=fc-fs
```

```
11  upper_sideband=fc+fs
12  amplitude=m*Ec/2
13
14  printf("lower  and  upper  sideband  frequencies  =  %.3f
         kHz  and  %.3f  kHz  \n",lower_sideband,
         upper_sideband)
15  printf("amplitude  of  each  sideband  term  =  %.3f  V  \n"
         ,amplitude)
```

Scilab code Exa 19.7 `power of modulated wave`

```
1  //chapter19
2  //example19.7
3  //page422
4
5  Pc=500  // W
6  m=1
7
8  Ps=0.5*m^2*Pc
9  Pt=Pc+Ps
10
11  printf("sideband  power  =  %.3f  W  \n",Ps)
12  printf("power  of  modulated  wave  =  %.3f  W  \n",Pt)
```

Scilab code Exa 19.8 `sideband power`

```
1  //chapter19
2  //example19.8
3  //page422
4
5  m1=0.8
6  m2=0.1
7  Pc=50  // kW
```

```
8
9  Ps1 =0.5* m1^2* Pc
10 Ps2 =0.5* m2^2* Pc
11
12 printf (" for  m=0.8 ,  sideband  power  =  %.3 f  kW  \n",Ps1)
13 printf (" for  m=0.1 ,  sideband  power  =  %.3 f  kW  \n",Ps2)
```

Scilab code Exa 19.9 `modulating signal`

```
1  // chapter19
2  // example19 .9
3  // page422
4
5  // block  diagram  is  for  understanding  purpose  inly .
     It  is  not  required  to  solve  the  example
6  m=1
7  eta =0.72
8  // carrier  is  not  affected  by  modulating  signal  so
     its  power  level  remains  unchanged  before  and
     after  modulation
9  Pc=40  // kW
10 Ps =0.5* m^2* Pc
11 P_audio =Ps/ eta
12
13 printf (" carrier  power  after  modulation  =  %.3 f  kW  \n"
     ,Pc )
14 printf (" required  audio  power  =  %.3 f  kW  \n",P_audio )
```

Scilab code Exa 19.10 `frequencie \& bandwith`

```
1  // chapter19
2  // example19 .10
3  // page423
```

167

```
 4
 5 fc=500 // kHz
 6 fs=1 // kHz
 7
 8 lower_sideband=fc-fs
 9 upper_sideband=fc+fs
10 BW=upper_sideband-lower_sideband
11
12 printf("sideband frequencies = %.3f kHz and %.3f kHz
        \n",lower_sideband,upper_sideband)
13 printf("bandwidth required = %.3f kHz \n",BW)
```

Scilab code Exa 19.11 `antenna current`

```
 1 //chapter19
 2 //example19.11
 3 //page423
 4
 5 m=0.4
 6 Ic=8 // A
 7 // Pt=Pc+Ps and Ps=0.5*m^2*Pc so Pt=Pc*(1+m^2/2)
 8 // so Pt/Pc=1+m^2/2 but P is proportional to I^2 so
 9 // (It/Ic)^2=1+m^2/2 and thus we get
10
11 It=Ic*(1+m^2/2)^0.5
12
13 printf("antenna current for m=0.4 is = %.3f A \n",It
        )
```

Scilab code Exa 19.12 `modulation factor`

```
 1 //chapter19
 2 //example19.12
```

```
3  //page424
4
5  It =8.93  // A
6  Ic =8  // A
7
8  // we know that (It/Ic)^2=1+m^2/2 so making m as
        subject we get
9  m=(2*((It/Ic)^2-1))^0.5
10
11 printf(" modulation factor = %.3 f or %.3 f percent \n"
        ,m, m*100)
```

Scilab code Exa 19.13 `modulation factor`

```
1  // chapter19
2  //example19.13
3  //page424
4
5  Vt =110  // V
6  Vc =100  // V
7
8  // since Pt/Pc=1+m^2/2 and P is proportional to V^2
        we get (Vt/Vc)^2=1+m^2/2
9  // making m as subject we get
10
11 m=(2*((Vt/Vc)^2-1))^0.5
12
13 printf(" modulation factor = %.3 f or %.3 f percent \n"
        ,m, m*100)
```

Scilab code Exa 19.14 `AM wave`

```
1  // chapter19
```

```
 2  //example19.14
 3  //page424
 4
 5  Vc=5 // V
 6  V_lower=2.5 // V
 7  V_upper=2.5 // V
 8  R=2 // kilo ohm
 9
10  // figure given in book is just for understanding
       purpose.It is not a part of solution.
11  // however, the figure has been made in xcos and
       screenshot has been attached for reference
12
13  // since power=(rms voltage)^2/R we get
14
15  Pc=(0.707*Vc)^2/R
16  P_lower=(0.707*V_lower)^2/R
17  P_upper=(0.707*V_upper)^2/R
18  Pt=Pc+P_lower+P_upper
19
20  printf("power delivered by carrier = %.3f mW \n",Pc)
21  printf("power delivered by lower sideband = %.3f mW
       \n",P_lower)
22  printf("power delivered by upper sideband = %.3f mW
       \n",P_upper)
23  printf("total power delivered by AM wave = %.3f mW \
       n",Pt)
```

Chapter 20

Regulated dc power supply

Scilab code Exa 20.1 `regulation`

```
1  // chapter20
2  // example20.1
3  // page437
4
5  V_NL=400  // V
6  V_FL=300  // V
7
8  regulation=((V_NL-V_FL)/V_FL)*100
9
10 printf(" percent  voltage  regulation = %.3f  percent  \n
      ", regulation)
```

Scilab code Exa 20.2 `full load voltage`

```
1  // chapter20
2  // example20.2
3  // page437
4
```

```
5  V_NL=30 // V
6  regulation=1
7
8  // since regulation=((V_NL−V_FL)/V_FL)*100, we get
       V_FL as
9
10 V_FL=100*V_NL/(100+regulation)
11 printf("full load voltage = %.3f V \n",V_FL)
```

Scilab code Exa 20.3 `regulation for power`

```
1  //chapter20
2  //example20.3
3  //page437
4
5  // for power supply A
6  V_NL1=30 // V
7  V_FL1=25 // V
8
9  regulation1=((V_NL1-V_FL1)/V_FL1)*100
10
11 //for power supply B
12 V_NL2=30 // V
13 V_FL2=29 // V
14
15 regulation2=((V_NL2-V_FL2)/V_FL2)*100
16
17 printf("regulation for power supply A =%.3f percent
       \n",regulation1)
18 printf("regulation for power supply B =%.3f percent
       \n",regulation2)
19
20 if regulation1>regulation2 then
21         printf("thus, power supply B is better \n")
22       elseif regulation2>regulation1 then
```

```
23              printf("thus, power supply A is better \n")
24       else     printf("both are equally good \n")
25  end
```

Scilab code Exa 20.4 `minimum load resistance`

```
1  //chapter20
2  //example20.4
3  //page438
4
5  V_NL=500  // V
6  V_FL=300  // V
7  I_FL=120  // mA
8
9  regulation=((V_NL-V_FL)/V_FL)*100
10
11  Rl_min=V_FL/I_FL
12
13  printf("voltage regulation = %.3f percent \n",
       regulation)
14  printf("minimum load resistance = %.3f kilo ohm \n",
       Rl_min)
```

Scilab code Exa 20.5 `maximum zener current`

```
1  //chapter20
2  //example20.5
3  //page441
4
5  Vin=24  // V
6  Vout=12  // V
7  Rs=160  // ohm
8  Rl_min=200  // ohm
```

```
 9
10  Is =( Vin - Vout )/ Rs    // in ampere
11
12  // minimum load occurs when Rl tends to infinity so
13  Il_min =0
14
15  // maximum load occurs when Rl=200 ohm
16  Il_max = Vout / Rl_min    // in ampere
17
18  Iz_min = Is - Il_max    // in ampere
19  Iz_max = Is - Il_min    // in ampere
20
21  printf (" current through series reistance = %.3 f mA \
       n \n", Is *1000)
22  printf (" minimum load current = %.3 f mA \n", Il_min
       *1000)
23  printf (" maximum load current = %.3 f mA \n", Il_max
       *1000)
24  printf (" minimum zener current = %.3 f mA \n", Iz_min
       *1000)
25  printf (" maximum zener current = %.3 f mA \n \n",
       Iz_max *1000)
26
27  printf (" comment : current Is through Rs is constant
       .\ nAs load current increases from 0 to 60 mA,
       zener current decreases from 75 to 15 mA, \
       nmaintaining Is constant .\ nThis is the normal
       operation of zener regulator \ni . e . Is and Vout
       remain constant inspite of changes in load or
       source voltage .")
```

Scilab code Exa 20.6 required series resistance

```
1  // chapter20
2  // example20 .6
```

```
3  //page441
4
5  Vin_min =22  // V
6  Vout =15  // V
7  Il_max =0.1  // A
8
9  // for maximum series resistance , we consider the
       case when input voltage is minimum and load
       current is maximum because then zener current
       drops to minimum.Thus,
10 Rs_max =(Vin_min -Vout )/Il_max
11
12 printf ("required series resistance = %.3f ohm \n",
       Rs_max )
```

Scilab code Exa 20.7 load voltage \& current

```
1  //chapter20
2  //example20.7
3  //page442
4
5  Vz =10  // V
6  Vbe =0.5  // V
7  Rl =1000  // ohm
8
9  Vout =Vz -Vbe
10 Il =Vout/Rl
11
12 printf ("load voltage = %.3f V \n",Vout )
13 printf ("load current = %.3f mA \n",Il *1000)
```

Scilab code Exa 20.8 Zener diode

175

```
1  // chapter20
2  // example20.8
3  // page441
4
5  Ic=1 // A
6  gain=50
7  Vout=6 // V
8  Vbe=0.5 // V
9  Vin=10 // V
10  Iz=10d-3 // A
11
12  Ib=Ic/gain
13  Vz=Vbe+Vout // Vout=Vz-Vbe
14
15  V_Rs=Vin-Vz
16  Rs=V_Rs/(Ib+Iz)
17
18  printf("required breakdown voltage for zener diode =
        %.3f V \n",Vz)
19  printf("required value of Rs = %.3f ohm \n",Rs)
20
21  // in book Rs=117 ohm but accurate answer is 116.667
        ohm
22
23  // note : in xcos, there is no Zener diode so in the
        result (circuit) file a simple diode is used to
        represent a zener diode
```

Scilab code Exa 20.9 Zener diode

```
1  // chapter20
2  // example20.9
3  // page443
4
5  Vz=12 // V
```

```
 6  Vbe =0.7  // V
 7  Vin =20  // V
 8  Rs =220  // ohm
 9  Rl =1 d3  // ohm
10  gain =50
11
12  Vout = Vz - Vbe
13  V_Rs = Vin - Vz
14  I_Rs = V_Rs / Rs
15  Il = Vout / Rl
16  Ic = Il
17  Ib = Ic / gain
18  Iz = I_Rs - Ib
19
20  printf ( " output  voltage  = %.3 f  V  \n " , Vout )
21  printf ( " zener  current  = %.3 f  mA  \n " , Iz *1000)
```

Scilab code Exa 20.10 regulated output voltage

```
 1  // chapter20
 2  // example20.10
 3  // page445
 4
 5  R2 =1  // kilo  ohm
 6  R1 =2  // kilo  ohm
 7  Vz =6  // V
 8  Vbe =0.7  // V
 9
10  m= R2 /( R1 + R2 )
11  A_CL =1/ m
12  Vout = A_CL *( Vz + Vbe )
13
14  printf ( " regulated  output  voltage  = %.3 f  V  \n " , Vout )
```

Scilab code Exa 20.11 closed loop voltage gain

```
1  // chapter20
2  // example20.11
3  // page445
4
5  R2=10  //  kilo  ohm
6  R1=30  //  kilo  ohm
7
8  m=R2/(R1+R2)
9  A_CL=1/m
10
11  printf("closed  loop  voltage  gain  = %.3f \n",A_CL)
```

Scilab code Exa 20.12 collector current

```
1  // chapter20
2  // example20.12
3  // page446
4
5  Vz=8.3  // V
6  Vbe=0.7  // V
7  Rl=100  // ohm
8  Rs=130  // ohm
9  Vin=22  // V
10
11  Vout=Vz+Vbe
12  Il=Vout/Rl
13  Is=(Vin-Vout)/Rs
14  Ic=Is-Il
15
16  printf("regulated  output  voltage  = %.3f V \n",Vout)
```

```
17  printf(" load current = %.3 f mA \n",Il*1000)
18  printf(" current through Rs = %.3 f mA \n",Is*1000)
19  printf(" collector current = %.3 f mA \n",Ic*1000)
```

Chapter 21

Solid state switching circuits

Scilab code Exa 21.1 voltage required to saturate transistor

```
1  //chapter21
2  //example21.1
3  //page456
4
5  Vcc=10  // V
6  Vbe=0.7  // V
7  Rb=47d3  // ohm
8  Rc=1d3  // ohm
9  gain=100
10
11  Ic_sat=Vcc/Rc
12  Ib=Ic_sat/gain
13  V_plus=Ib*Rb+Vbe
14
15  printf(" voltage  required  to  saturate  transistor  = +%
       .3 f  V  \n",V_plus)
```

Scilab code Exa 21.2 square wave

```
1  // chapter21
2  // example21.2
3  // page463
4
5  R =10 d3  // ohm
6  C =0.01 d -6  // F
7
8  T =1.4* R * C
9  f =1/ T
10
11 printf (" time  period  of  square  wave = %.3 f  ms  \n" ,T
       *1000)
12 printf (" frequency  of  square  wave = %.3 f  kHz  \n" ,f
       /1000)
13
14 // the  accurate  answer  for  frequency  is  7.143  kHz
       but  in  book  it  is  given  7  kHz
```

Scilab code Exa 21.3 differentiating circuit

```
1  // chapter  21
2  // example  21.3
3  // page  468
4
5  printf (" In  RC  differentiating  circuit ,  the  output
       votage  is  taken  across  \nR  and  waveform  of  output
        depends  on  time  constant  of  \ncircuit .  For
       proper  functioning ,  product  RC  should  be  many  \
       ntimes  smaller  than  time  period  of  input  wave .  \n
       ")
```

Scilab code Exa 21.4 output voltage

```
1  //chapter21
2  //example21.4
3  //page468
4
5  R=10d3  // ohm
6  C=2.2d-6  // F
7  V1=0  // V
8  V2=10  // V
9  t1=0  // sec
10 t2=0.4  // sec
11
12 Eo=R*C*(V2-V1)/(t2-t1)
13
14 printf("output voltage = %.3f V \n",Eo)
```

Scilab code Exa 21.5 plotting input and output waveforms

```
1  //chapter21
2  //example21.5
3  //page472
4
5  Vin_peak=12  // V
6
7  // for positive half cycle diode conducts so
8  Vout_peak=Vin_peak-0.7  // V
9
10 // for negative half cycle diode does not conduct so
11 Vout_min=0  // V
12
13 printf("peak output voltage = %.3f V in positive
        half cycle and \n                          %.3f V in
         negative half cycle",Vout_peak,Vout_min)
14
15 // plotting input and output waveforms in same graph
        using following code instead of using xcos
```

```
16  clf ()
17  t=linspace (0 ,2*%pi ,100)
18  Vin =12* sin(t)
19  Vout = Vout_peak * sin(t)+Vout_min
20  plot2d (t ,Vin ,style =2 ,rect =[0 ,0 ,10 ,20])
21  xtitle (" input - blue      output - green" ,"t" ," volts "
        )
22  plot2d (t ,Vout ,style =3 ,rect =[0 ,0 ,10 ,20])
```

Scilab code Exa 21.6 plotting input and output waveforms

```
1  // chapter21
2  // example21 .6
3  // page472
4
5  Rl =4 // kilo ohm
6  R =1 // kilo ohm
7  Vin_peak =10 // V
8
9  Vout_peak = Vin_peak * Rl /(Rl+R)
10  Vout_min =0 // because of diode
11  printf (" peak output voltage = %.3 f V \n" ,Vout_peak )
12
13  // plotting input and output waveforms in same graph
        using following code instead of using xcos
14  clf ()
15  t=linspace (0 ,2*%pi ,100)
16  Vin = Vin_peak * sin(t)
17  Vout = Vout_peak * sin(t)+Vout_min
18  plot2d (t ,Vin ,style =2 ,rect =[0 ,0 ,10 ,20])
19  xtitle (" input - blue      output - green" ,"t" ," volts "
        )
20  plot2d (t ,Vout ,style =3 ,rect =[0 ,0 ,10 ,20])
```

183

Scilab code Exa 21.7 plotting input and output waveforms

```
1  //chapter21
2  //example21.7
3  //page473
4
5  V=-10 // V
6  Vout=-0.7 // V
7
8  Vr=V-Vout
9
10 printf("output voltage = %.3f V \n",Vout)
11 printf("voltage across R = %.3f V \n",Vr)
12
13 // plotting input and output waveforms in same graph
       using following code instead of using xcos
14 clf()
15 t=linspace(0,%pi,100)
16 Vin=V*sin(t)
17 Vout=Vr*sin(t)
18 subplot(1,2,2)
19 plot2d(t,Vout,style=3,rect=[0,-0.7,10,11])
20 xtitle("Vout","t","volts")
21
22 subplot(1,2,1)
23 plot2d(t,Vin,style=2,rect=[0,-11,10,1])
24 xtitle("Vin","t","volts")
```

Scilab code Exa 21.8 plotting input and output waveforms

```
1  //chapter21
2  //example21.8
```

```
3  //page473
4
5  Rl =1 d3 // ohm
6  R =200 // ohm
7
8  // for positive half cycle, diode is forward biased
       and since load is in parallel with diode we get
9  V_out_p =0.7 // V
10
11 // for negative half cycle, diode is reverse biased
       so it is open. Hence
12 V_in = -10 // V
13 V_out_n = V_in * Rl /( Rl + R )
14
15 printf ("output voltage for positive cycle = %.3 f V \
       nand for negative cycle = %.3 f V", V_out_p , V_out_n
       )
16
17 // plotting input and output waveforms in same graph
         using following code instead of using xcos
18 clf ()
19 t = linspace (0 , % pi ,100)
20 Vin = V_in * sin ( t )
21 Vout = - V_out_n * sin ( t )
22 subplot (2 ,2 ,1)
23 plot2d (t , - Vin , style =3 , rect =[0 ,0 ,10 ,11])
24 xtitle ("Vin +ve " ," t " ," volts ")
25 subplot (2 ,2 ,2)
26 plot2d (t , Vout , style =2 , rect =[0 , -5 ,10 ,0.7])
27 xtitle ("Vout " ," t " ," volts ")
28 t = linspace (% pi ,2*% pi ,100)
29 Vin = V_in * sin ( t )
30 subplot (2 ,2 ,3)
31 plot2d (t , - Vin , style =3 , rect =[0 , -11 ,10 ,0])
32 xtitle ("Vin −ve " ," t " ," volts ")
33 subplot (2 ,2 ,4)
34 plot2d (t , - Vout , style =2 , rect =[0 , -11 ,10 ,0])
35 xtitle ("Vout " ," t " ," volts ")
```

185

Scilab code Exa 21.9 `empty`

```
1  //chapter21
2  //example21.9
3  //page474
4
5  printf("The purpose of using series resistance R is
       : \n 1) if R is not present, diode will short
       voltage source in positive half cycle \n 2) so
       large current will flow which may damage voltage
       source or diode. \n To prevent this i.e. to
       protect diode and voltage source, R is used.")
```

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

Scilab code Exa 21.10 `plot input and output waveforms`

```
1  // chapter 21
2  // example 21.10
3  // page 478
4
5  V=2 // V
6  Vin=5 // V
7
8  // during positive half cycle
9  Vc_p=Vin-V // since Vin-Vc-V=0
10 // thus capacitor charges to Vc_p
11
12 // during negative half cycle
13 Vout=-Vin-Vc_p // since Vin-Vc_p-Vout=0
```

```
14
15  // we plot input and output waveforms using the
        following code instead of using xcos
16
17  clf()
18  t=0:0.1:5*%pi
19  plot(t,5*squarewave(t,50))
20  plot2d(t,-Vc_p+(-Vout+V)*squarewave(t,50)/2,style=3)
21  xtitle("input − blue                      output −
        green","t","volts")
```

Scilab code Exa 21.11 `plot input and output waveforms`

```
1  // chapter 21
2  // example 21.11
3  // page 479
4
5  V=-2 // V
6  Vin=5 // V
7
8  // during positive half cycle
9  Vc_p=Vin-V // since Vin−Vc−V=0
10 // thus capacitor charges to Vc_p
11
12 // during negative half cycle
13 Vout=-Vin-Vc_p // since Vin−Vc_p−Vout=0
14
15 // we plot input and output waveforms using the
        following code instead of using xcos
16
17 clf()
18 t=0:0.1:5*%pi
19 plot(t,5*squarewave(t,50))
20 plot2d(t,-Vc_p+(-Vout+V)*squarewave(t,50)/2,style=3)
21 xtitle("input − blue                      output −
```

green","t","volts")

Chapter 22

Field Effect Transistors

Scilab code Exa 22.1 `Id`

```
1 // chpater22
2 // example22.1
3 // page491
4
5 I_DSS =12 // mA
6 V_GS_off = -5 // V
7
8 printf("I_D=%d*(1+V_GS/%d)^2 mA \n",I_DSS ,-V_GS_off)
```

Scilab code Exa 22.2 `drain current`

```
1 // chapter22
2 // example22.2
3 // page491
4
5 I_DSS =32 // mA
6 V_GS = -4.5 // V
7 V_GS_off = -8 // V
```

```
 8
 9  I_D = I_DSS *(1 - V_GS / V_GS_off )^2
10
11  printf ("drain  current  = %.3f mA \n", I_D)
```

Scilab code Exa 22.3 pinch off voltage

```
 1  // chapter22
 2  // example22.3
 3  // page491
 4
 5  I_D =5  // mA
 6  I_DSS =10  // mA
 7  V_GS_off =-6  // V
 8
 9  // we know that I_D=I_DSS*(1-V_GS/V_GS_off)^2 so
      making V_GS as subject we get
10
11  V_GS = V_GS_off *(1 -( I_D / I_DSS )^0.5)
12  V_P = - V_GS_off
13
14  printf ("gate source voltage = %.3f V \n", V_GS)
15  printf ("pinch off voltage = %.3f V \n", V_P)
```

Scilab code Exa 22.4 gate source resistance

```
 1  // chapter22
 2  // example22.4
 3  // page493
 4
 5  V_GS =15  // V
 6  I_G =1d -9  // A
 7
```

```
 8  R_GS = V_GS / I_G
 9
10  printf ("gate source resistance = %.3f ohm or %.3f
        mega ohm \n",R_GS,R_GS/1d6)
```

Scilab code Exa 22.5 transconductance

```
 1  //chapter22
 2  //example22.5
 3  //page493
 4
 5  Vgs1 = -3.1  // V
 6  Vgs2 = -3  // V
 7  Id1 =1d -3  // A
 8  Id2 =1.3d -3  // A
 9
10  g_fs =(Id2 - Id1)/(Vgs2 - Vgs1)
11
12  printf ("transconductance = %.3f mho or %.3f micro
        mho \n",g_fs,g_fs*1d6)
```

Scilab code Exa 22.6 amplification factor

```
 1  //chapter22
 2  //example22.6
 3  //page493
 4
 5  // for V_GS = 0V constant
 6  V_DS1 =7  // V
 7  V_DS2 =15  // V
 8  I_D1 =10  // mA
 9  I_D2 =10.25  // mA
10
```

```
11  rd =( V_DS2 - V_DS1 )/( I_D2 - I_D1 )
12
13  // for V_DS = 15V constant
14  V_GS1 =0
15  V_GS2 =0.2
16  I_D1 =9.65
17  I_D2 =10.25
18
19  g_fs =( I_D2 - I_D1 )/( V_GS2 - V_GS1 )
20
21  mu = rd * g_fs
22
23  printf (" ac drain resistance = %.3 f ohm or %.3 f kilo
        ohm \n", rd /1000 , rd )
24  printf (" transconductance = %.3 f mho or %.3 f micro
        mho \n", g_fs , g_fs *1000)
25  printf (" amplification factor = %.3 f \n", mu )
```

Scilab code Exa 22.7 Kirchoff's law

```
1  // chapter22
2  // example22 .7
3  // page496
4
5  I_DSS =5d -3 // A
6  V_DD =20 // V
7  V_DS =10 // V
8  V_P = -2 // V
9  V_G =0 // V
10  I_D =1.5d -3 // A
11
12  V_GS = V_P *(1 -(( I_D / I_DSS )^0.5)) // I_D=I_DSS*(1-V_GS/
        V_P)^2
13  V_S = V_G - V_GS
14  R_S = V_S / I_D
```

```
15
16  // by Kirchoff's law we get V_DD=I_D*R_D+V_DS+I_D*
        R_S so making R_D as subject we get
17  R_D=(V_DD-V_DS-I_D*R_S)/I_D
18
19  printf("Rs = %.3f kilo ohm and Rd = %.3f kilo ohm \n
        ",R_S/1000,R_D/1000)
```

Scilab code Exa 22.8 required value of Rs

```
1  //chapter22
2  //example22.8
3  //page496
4
5  V_P=-5 // V
6  V_DD=30 // V
7  I_DSS=10 // mA
8  I_D=2.5 // mA
9  R1=1000 // kilo ohm
10 R2=500 // kilo ohm
11
12 // since I_D=I_DSS*(1-(V_GS/V_P))^2, making V_GS as
       subject we get
13
14 V_GS=V_P*(1-(I_D/I_DSS)^0.5)
15
16 V2=V_DD*R2/(R1+R2)
17
18 // since V2 = V_GS + I_D*Rs, making Rs as subject we
       get
19
20 Rs=(V2-V_GS)/I_D
21
22 printf("required value of Rs = %.3f kilo ohm \n",Rs)
```

Scilab code Exa 22.9 voltage amplification of the circuit

```
1  //chapter22
2  //example22.9
3  //page497
4
5  R_L=10d3 // ohm
6  g_fs=3000d-6 // mho
7
8  // since rd >> R_L, we can write
9
10 Av=g_fs*R_L
11
12 printf("voltage amplification of the circuit = %.3f
       \n",Av)
```

Scilab code Exa 22.10 Vds

```
1  //chapter22
2  //example22.10
3  //page498
4
5  V_DD=30 // V
6  I_D=2.5d-3 // A
7  R_D=5d3 // ohm
8  R_S=200 // ohm
9
10 V_DS=V_DD-I_D*(R_D+R_S)
11 V_GS=-I_D*R_S
12
13 printf("V_DS = %.3f V \n",V_DS)
14 printf("V_GS = %.3f V \n",V_GS)
```

Scilab code Exa 22.11 `dc voltage of drain`

```
1  //chapter22
2  //example22.11
3  //page498
4
5  V_DD=30  // V
6  I_D1=2.15d-3  // A
7  I_D2=9.15d-3  // A
8  R_D1=8.2d3  // ohm
9  R_D2=2d3  // ohm
10 R_S1=680  // ohm
11 R_S2=220  // ohm
12
13 V_RD1=I_D1*R_D1
14 V_D1=V_DD-V_RD1
15 V_S1=I_D1*R_S1
16
17 V_RD2=I_D2*R_D2
18 V_D2=V_DD-V_RD2
19 V_S2=I_D2*R_S2
20
21 printf("For stage 1 : dc voltage of drain = %.3f V
        and source = %.3f V \n",V_D1,V_S1)
22 printf("For stage 2 : dc voltage of drain = %.3f V
        and source = %.3f V \n",V_D2,V_S2)
```

Chapter 23

Silicon controlled rectifiers

Scilab code Exa 23.1 Breakover voltage

```
1  // chapter23
2  // example23.1
3  // page509
4
5  printf("1) Breakover voltage of 400V : It means that
        if gate is open and the \n")
6  printf("    supply voltage is 400V, then SCR will
       start conducting heavily. \n")
7  printf("    However, as long as the supply voltage <
       400V, SCR stays open. \n \n")
8
9  printf("2) Trigger current of 10mA : It means that
        if the supply voltage is \n")
10 printf("    less than breakover voltage and a minimum
        gate current of 10 mA \n")
11 printf("    is passed, SCR conducts. It wont conduct
       if gate current is less \n")
12 printf("    than 10mA. \n \n")
13
14 printf("3) Holding current of 10mA : When SCR is
       conducting, it will not open \n")
```

```
15  printf (" even if triggering current is removed.
        However, if supply voltage \n")
16  printf (" is reduced, anode current also decreases.
        When anode current drops \n")
17  printf (" to 10mA, the holding current, SCR turns
        off. \n \n")
18
19  printf ("4) If gate current is increased to 15mA, the
        SCR will be turned on \n")
20  printf (" lower supply voltage. \n")
```

Scilab code Exa 23.2 `rating`

```
1  // chapter23
2  // example23.2
3  // page510
4
5  t =12d -3  // sec
6  I =50  // A
7  fuse_rating=I^2*t
8
9  if fuse_rating < 90
10      printf (" rating = %.3 f ampere square second
            which is less than maximum \nrating so
            device will not be destroyed \n",fuse_rating
            )
11  else printf (" rating = %.3 f ampere square second
        which is more than maximum \nrating so device may
         get damaged \n",fuse_rating)
12
13  end
```

Scilab code Exa 23.3 `maximum allowable duration of surge`

```
1  //chapter23
2  //example23.3
3  //page510
4
5  rating=50  // ampere square second
6  Is=100  // A
7
8  t_max=rating/Is^2
9
10 printf("maximum allowable duration of surge = %.3f s
        or %.3f ms\n",t_max,t_max*1000)
```

Scilab code Exa 23.4 average current

```
1  //chapter23
2  //example23.4
3  //page514
4
5  v=100  // V
6  Vm=200  // V
7  R_L=100  // ohm
8
9  // since v=Vm*sin(theta), we get
10
11 theta=asin(v/Vm)*180/%pi  // in terms of degrees
12
13 phi=180-theta
14
15 V_avg=Vm*(1+cos(theta*%pi/180))/(2*%pi)
16
17 I_avg=V_avg/R_L
18
19 printf("firing angle = %.2f degrees \n",theta)
20 printf("conduction angle = %.2f degrees \n",phi)
21 printf("average current = %.4f A \n",I_avg)
```

```
22
23  // the accurate answer for average current is 0.594
       A but in book it is given as 0.5925 A
```

Scilab code Exa 23.5 `power output`

```
1  //chapter23
2  //example23.5
3  //page515
4
5  Vm=400  // V
6  v=150  // V
7  R_L=200  // ohm
8
9  // since v=Vm*sin(theta), we get
10
11  theta=asin(v/Vm)*180/%pi  // in terms of degrees
12
13  V_av=Vm*(1+cos(theta*%pi/180))/(2*%pi)
14  I_av=V_av/R_L
15  P=V_av*I_av
16
17  printf("firing angle = %.2f degrees \n",theta)
18  printf("average output voltage = %.3f V \n",V_av)
19  printf("average current for load of 200 ohm = %.3f A
        \n",I_av)
20  printf("power output = %.3f W \n",P)
21
22  // the accurate answer for power output is 75.250 W
       but in book it is given as 75.15 W
```

Scilab code Exa 23.6 `forward breakdown voltage`

```
1  //chapter23
2  //example23.6
3  //page515
4
5  Vm=240  // V
6  v=180  // V
7
8  // figure given is for understanding purpose only.
       It is not required to solve the example
9
10  // SCR remains off till it reaches 180 V i.e.
       forward breakdown voltage
11
12  // since v=Vm*sin(theta), we get
13
14  theta=asin(v/Vm)  // firing angle in terms of degrees
15
16  // since theta=314*t, we get
17
18  t=theta/314  // seconds
19
20  printf(" off duration of SCR = %.3f ms \n",t*1000)  //
       multiply t by 1000 to display time in
       milliseconds
```

Scilab code Exa 23.7 voltage

```
1  //chapter23
2  //example23.7
3  //page517
4
5  alpha=60  // degrees
6  Vm=200  // V
7  R_L=100  // ohm
8
```

```
 9  V_av=Vm*(1+cos(alpha*%pi/180))/%pi
10
11  I_av=V_av/R_L
12
13  printf("dc output voltage = %.3f V \n",V_av)
14  printf("load current for firing angle of 60 degrees
          = %.3f A \n",I_av)
```

Chapter 24

Power electronics

Scilab code Exa 24.1 `(RB1+RB2) = RB1/Rbb`

```
1  //chapter24
2  //example24.1
3  //page533
4
5  RBB =10 // kilo ohm
6  eta =0.6
7
8  //eta=RB1/(RB1+RB2) = RB1/Rbb so
9  RB1=eta*RBB
10 RB2=RBB-RB1
11 printf("RB1 = %.3f kilo ohm \n",RB1)
12 printf("RB2 = %.3f kilo ohm",RB2)
```

Scilab code Exa 24.2 `voltage`

```
1  //chapter24
2  //example24.2
3  //page533
```

```scilab
 4
 5  VBB =10  // V
 6  eta =0.65
 7  VD =0.7  // V
 8
 9  stand_off_voltage=eta*VBB
10  peak_point_voltage=VD+eta*VBB
11
12  printf("stand  off  voltage = %.3f V \n",
       stand_off_voltage)
13  printf("peak  point  voltage = %.3f V \n",
       peak_point_voltage)
```

Chapter 25

Electronic instruments

Scilab code Exa 25.1 sensitivity

```
1 // chapter25
2 // example25.1
3 // page543
4
5 Ig=1d-3 // A
6
7 S=1/Ig
8
9 printf("sensitivity = %.3f ohm per volt \n",S)
```

Scilab code Exa 25.2 equivalent resistance of meter

```
1 // chapter25
2 // example25.2
3 // page543
4
5 S=1000 // ohm per volt
6 V=50 // V
```

```
7  R=50d3 // ohm
8
9  R_meter=S*V
10
11 R_equi=R*R_meter/(R+R_meter) //equivalent resistance
       of meter and given resistance across which meter
       is connected
12
13 printf("ratio of circuit resistance before and after
       connecting multimeter = %.3f \n",R/R_equi)
14 printf("Thus equivalent resistance is reduced to
       half. So current drawn is double \n")
15 printf("Thus multimeter will give highly incorrect
       reading \n \n")
16 printf("As a rule, multimeter resistance should be
       100 times the resistance across \nwhich voltage
       is to be measured \n")
```

Scilab code Exa 25.3 voltage read by multimeter

```
1  //chapter25
2  //example25.3
3  //page544
4
5  S=4 // kilo ohm per volt
6  V_range=10 // V
7  V=20 // V
8  R=10 // kilo ohm
9
10 R_meter=S*V_range
11 R_equi=R+R*R_meter/(R+R_meter)
12 I=V/R_equi
13 V_reading=I*R*R_meter/(R+R_meter)
14
15 printf("voltage read by multimeter = %.3f V \n",
```

```
        V_reading )
```

Scilab code Exa 25.4 voltage read by multimete

```
1  // chapter25
2  // example25.4
3  // page544
4
5  S =20 //  kilo  ohm  per  volt
6  V_range =10 //  V
7  V =20 //  V
8  R =10 //  kilo  ohm
9
10 R_meter = S * V_range
11 R_equi = R + R * R_meter /( R + R_meter )
12 I = V / R_equi
13 V_reading = I * R * R_meter /( R + R_meter )
14
15 printf (" voltage  read  by  multimeter  = %.3 f  V  \n" ,
       V_reading )
16
17 //  answer  in  book  is  9.88V but  accurate  answer  is
       9.756V
```

Scilab code Exa 25.5 resistance of voltmeter

```
1  // chapter25
2  // example25.5
3  // page545
4
5  R1 =20 d3 //  ohm
6  R2 =20 d3 //  ohm
7  R3 =30 d3 //  ohm
```

```
 8  R4 =30 d3  // ohm
 9  V =100  // V
10  Rm =60 d3  // ohm
11
12  // case 1 : meter is not connected
13  R = R1 + R2 + R3 + R4
14  I = V / R
15  V_A = V
16  V_B = V - I * R2
17  V_C = V - I * ( R1 + R2 )
18  V_D = V - I * ( R1 + R2 + R3 )
19
20  // case2 : meter is connected
21      // At A
22      V_A1 = V
23
24      // At B
25      R_total_B = R1 + ( Rm * ( R2 + R3 + R4 ) / ( Rm + R2 + R3 + R4 ) )
26      I1 = V / R_total_B
27      V_B1 = I1 * ( Rm * ( R2 + R3 + R4 ) / ( Rm + R2 + R3 + R4 ) )
28
29      // At C
30      R_total_C = R1 + R2 + ( Rm * ( R3 + R4 ) / ( Rm + R3 + R4 ) )
31      I2 = V / R_total_C
32      V_C1 = V * ( Rm * ( R3 + R4 ) / ( Rm + R3 + R4 ) ) / R_total_C
33
34      // At D
35      R_total_D = R1 + R2 + R3 + ( Rm * R4 / ( Rm + R4 ) )
36      I2 = V / R_total_D
37      V_D1 = V * ( Rm * R4 / ( Rm + R4 ) ) / R_total_D
38
39  printf ("CASE 1 : meter is not connected \n
         Voltage at A = %.3 f V \n        Volatge at B = %.3 f
         V \n        Volatge at C = %.3 f V \n        Volatge at
         D = %.3 f V \n",V_A ,V_B ,V_C ,V_D )
40  printf ("CASE 2 : meter is connected \n        At A
         then voltage at A = %.3 f V",V_A1 )
41  printf ("\n        At B then voltage at B = %.3 f V",
```

```
      V_B1)
42  printf("\n         At  C  then  voltage  at  C  =  %.3 f  V",
        V_C1)
43  printf("\n         At  D  then  voltage  at  D  =  %.3 f  V  \n  \
        n",V_D1)

44
45  printf("resistance  of  voltmeter  should  be  100  times
        the  resistance  across  \nwhich  voltage  is  to  be
        measured.Since  this  condition  is  not  \nsatisfied
        here,  readings  are  wrong.  \n")
```

Scilab code Exa 25.6 `spot shift`

```
1  //chapter25
2  //example25.6
3  //page552
4
5  S=0.01  //mm per volt
6  V=400  // V
7
8  spot_shift=S*V
9
10  printf("spot  shift  =  %.3 f  mm  \n",spot_shift)
```

Scilab code Exa 25.7 `applied voltage`

```
1  //chapter25
2  //example25.7
3  //page552
4
5  S=0.03  // mm per volt
6  spot_shift=3  // mm
7
```

```
8  V=spot_shift/S // since spot shift = deflection
        sensitivity * applied voltage
9
10 printf("applied voltage = %.3f V \n",V)
```

Scilab code Exa 25.8 unknown voltage

```
1  //chapter25
2  //example25.8
3  //page555
4
5  V1=200 // V
6  d1=2 // cm
7  d2=3 // cm
8
9  // since sensitivity = voltage / deflcetion we get
10 S=V1/d1
11 V2=S*d2
12
13 printf("unknown voltage = %.3f V",V2)
```

Scilab code Exa 25.9 ratio of fv to fh

```
1  //chapter25
2  //example25.9
3  //page556
4
5  fh=1000 // Hz
6
7  // case (i) :- ratio of fv to fh = 1:1
8  fv1=1*fh
9
10 // case (ii) :- ratio = 2:1
```

```
11  fv2=2*fh
12
13  // case (iii) :- ratio = 6:1
14  fv3=6*fh
15
16  printf(" for case1 i.e. fv/fh = 1/1, fv = %.3f Hz \n"
        ,fv1)
17  printf(" for case2 i.e. fv/fh = 2/1, fv = %.3f Hz \n"
        ,fv2)
18  printf(" for case3 i.e. fv/fh = 6/1, fv = %.3f Hz \n"
        ,fv3)
```

Chapter 26

Integrated circuits

Scilab code Exa 26.1 regulated dc output voltage

```
1  // chapter26
2  // example26.1
3  // page570
4
5  R2=2.4d3  // ohm
6  R1=240    // ohm
7
8  V_out=1.25*(1+R2/R1)
9
10 printf("regulated dc output voltage = %.3f V \n",
        V_out)
```

Chapter 27

Hybrid parameters

Scilab code Exa 27.1 `ratio`

```
1  //chapter27
2  //example27.1
3  //page574
4
5  R1=10 // ohm
6  R2=5 // ohm
7
8  // for h11 and h21, imagine that output terminals
       are shorted hence it is clear that input
       impedence is equal to R1.
9       // this is h11 by definition so
10      h11=R1
11
12      // now current will flow of same magnitude but
            in opposite directions through input and
            output terminals so output_current/
            input_current = -1
13      // but this ratio is h21 by definition. Thus
14      h21=-1
15
16  // for h12 and h22 imagine a voltage source on
```

```
      output terminals
17      // this voltage will be avilable on input
          terminals also since current through 10 ohm
          resistor = 0.
18      // hence input_voltage/output_voltage = 1
19      // but this ratio is h12 by definition. Thus
20      h12=1
21
22      // here output impedence looking into output
          terminals with input terminals open is 5 ohm.
23      // its reciprocal is h22 by definition. Thus
24      h22=1/5
25
26  printf("h11 = %.3f ohm \n",h11)
27  printf("h21 = %.3f \n",h21)
28  printf("h12 = %.3f \n",h12)
29  printf("h22 = %.3f ohm \n",h22)
```

Scilab code Exa 27.2 `impedence`

```
 1  //chapter27
 2  //example27.2
 3  //page575
 4
 5  R1=4  // ohm
 6  R2=4  // ohm
 7  R3=4  // ohm
 8
 9  // for h11 and h21, imagine that output terminals
        are shorted hence it is clear that input
        impedence is equal to R1+R2*R3/(R2+R3)
10      // this is h11 by definition so
11      h11=R1+R2*R3/(R2+R3)
12
13      // now current will divide equally at junction
```

213

```
          of 4 ohm resistors so output_current/
          input_current = −0.5
14      // but this ratio is h21 by definition. Thus
15      h21 = -0.5
16
17 // for h12 and h22 imagine a voltage source on
      output terminals
18      // this voltage will be divided by a factor 2
19      // hence input_voltage/output_voltage = 0.5
20      // but this ratio is h12 by definition. Thus
21      h12 = 0.5
22
23      // here output impedence looking into output
            terminals with input terminals open is 8 ohm.
24      // its reciprocal is h22 by definition. Thus
25      h22 = 1/8
26
27 printf("h11 = %.3f ohm \n",h11)
28 printf("h21 = %.3f \n",h21)
29 printf("h12 = %.3f \n",h12)
30 printf("h22 = %.3f ohm \n",h22)
```

Scilab code Exa 27.3 voltage gain of circuit

```
1 //chapter27
2 //example27.3
3 //page578
4
5 h11 = 10
6 h12 = 1
7 h21 = -1
8 h22 = 0.2
9 rL = 5 // ohm
10
11 Zin = h11 - (h12*h21/(h22+1/rL))
```

```
12  Av=-h21/(Zin*(h22+1/rL))
13
14  printf("input impedence = %.3f ohm \n",Zin)
15  printf("voltage gain of circuit = %.3f \n",Av)
```

Scilab code Exa 27.4 `approximate current gain`

```
 1  //chapter27
 2  //example27.4
 3  //page581
 4
 5  hie=2000 // ohm
 6  hoe=1d-4 // mho
 7  hre=1d-3
 8  hfe=50
 9  rL=600 // ohm
10
11  Zin=hie-hre*hfe/(hoe+1/rL)
12  // here second term can be neglected compared to hie
        so
13  Zin_approx=hie
14
15  Ai=hfe/(1+hoe*rL)
16  // if hoe*rL << 1 then
17  Ai_approx=hfe
18
19  Av=-hfe/(Zin*(hoe+1/rL))
20  // negative sign indicates phase shift between input
        and output
21
22  printf("input impedence = %.3f ohm \n",Zin)
23  printf("current gain = %.3f \n",Ai)
24  printf("voltage gain = %.3f. Here negative sign
        indicates phase shift between input and output.\n
        \n",Av)
```

```
25
26  printf("approximate input impedence = %.3f ohm \n",
        Zin_approx)
27  printf("approximate current gain = %.3f \n",
        Ai_approx)
```

Scilab code Exa 27.5 input impedence

```
1   //chapter27
2   //example27.5
3   //page582
4
5   hie=1700  // ohm
6   hre=1.3d-4
7   hoe=6d-6  // mho
8   hfe=38
9   rL=2000  // ohm
10
11  Zin=hie-hre*hfe/(hoe+1/rL)
12
13  Ai=hfe/(1+hoe*rL)
14
15  Av=-hfe/(Zin*(hoe+1/rL))
16
17  printf("input impedence = %.3f ohm \n",Zin)
18  printf("current gain = %.3f \n",Ai)
19  printf("voltage gain = %.3f \n",-Av)  // considering
        magnitude of Av,we neglect its negative sign and
        so we display −Av instead of Av
```

Scilab code Exa 27.6 voltage gain

```
1   //chapter27
```

```
2  //example27.6
3  //page582
4
5  hie=1500  // ohm
6  hre=4d-4
7  hoe=5d-5  // mho
8  hfe=50
9  Rc=10d3  // ohm
10 R_L=30d3  // ohm
11 R1=80d3  // ohm
12 R2=40d3  // ohm
13
14 rL=Rc*R_L/(Rc+R_L)
15 Zin=hie-hre*hfe/(hoe+1/rL)
16 Zin_stage=Zin*(R1*R2/(R1+R2))/(Zin+(R1*R2/(R1+R2)))
17
18 Av=-hfe/(Zin*(hoe+1/rL))
19
20 printf("input impedence = %.3f ohm \n",Zin_stage)
21 printf("voltage gain = %.3f \n",Av)
22
23 // the accurate answers are input impedence =
        1321.957 ohm and voltage gain = −196.078 but in
        book they are given as 1320 ohm and −196
        respectively
```

Scilab code Exa 27.7 `micro mho`

```
1  //chapter27
2  //example27.7
3  //page584
4
5  Vbe=10d-3  // V
6  Vbe2=0.65d-3  // V
7  Vce=1  // V
```

```scilab
 8  Ib=10d-6 // A
 9  Ic=1d-3 // A
10  Ic2=60d-6 // A
11
12  hie=Vbe/Ib // in ohm
13  hfe=Ic/Ib // in ohm
14  hre=Vbe2/Vce
15  hoe=Ic2/Vce // in mho
16
17  printf(" hie = %.3 f  ohm  \n",hie)
18  printf(" hfe = %.3 f  ohm  \n",hfe)
19  printf(" hre = %.5 f  \n",hre)
20  printf(" hoe = %.3 f  micro  mho  \n",hoe*1d6)
```

Chapter 28

Digital electronics

Scilab code Exa 28.1 binary equivalent of decimal number 37

```
1 //chapter28
2 //example28.1
3 //page590
4
5 a= dec2bin (37)
6 disp(a,'binary equivalent of decimal number 37 = ')
```

Scilab code Exa 28.2 binary equivalent of decimal number 23

```
1 //chapter28
2 //example28.2
3 //page590
4
5 a= dec2bin (23)
6 disp(a,'binary equivalent of decimal number 23 = ')
```

Scilab code Exa 28.3 equivalent decimal of binary 110001

```
1  //chapter28
2  //example28.3
3  //page591
4
5  a= bin2dec ( ' 110001 ' )
6  printf("equivalent decimal of binary 110001 is %d \n
      ",a)
```

Scilab code Exa 28.4 page598

```
1  //chapter28
2  //example28.4
3  //page598
4
5  disp(" A        B         Y_dash = A + B     Y = Y_dash .A
      ")
6  disp(" 0        0             0                  0      ")
7  disp(" 1        0             1                  1      ")
8  disp(" 0        1             1                  0      ")
9  disp(" 1        1             1                  1      ")
10
11 printf("\nexplanation : \n")
12 printf("A=0 and B=0 give A'=1 and B'=1 so Y_dash = A
      + B is  0 and Y = Y_dash .A is  0 \n")
13
14 printf("A=1 and B=0 give A'=0 and B'=1 so Y_dash = A
      + B is  1 and Y = Y_dash .A is  1 \n")
15
16 printf("A=0 and B=1 give A'=1 and B'=0 so Y_dash = A
      + B is  1 and Y = Y_dash .A is  0 \n")
17
18 printf("A=1 and B=1 give A'=0 and B'=0 so Y_dash = A
      + B is  1 and Y = Y_dash .A is  1 \n")
```

Scilab code Exa 28.5 page598

```
1  //chapter28
2  //example28.5
3  //page598
4
5  disp(" A        B        A'      Y_dash = A'. B        B'
             Y = Y_dash + B' ")
6  disp(" 0        0        1        0                    1
                  1 ")
7  disp(" 1        0        0        0                    1
                  1 ")
8  disp(" 0        1        1        1                    0
                  1 ")
9  disp(" 1        1        0        0                    0
                  0 ")
10
11 printf("\nexplanation : \n")
12 printf("A=0 and B=0 give A'=1 and B'=1 so Y_dash = A
       '.B is 0 and Y = Y_dash + B' is 1 \n")
13
14 printf("A=1 and B=0 give A'=0 and B'=1 so Y_dash = A
       '.B is 0 and Y = Y_dash + B' is 1 \n")
15
16 printf("A=0 and B=1 give A'=1 and B'=0 so Y_dash = A
       '.B is 1 and Y = Y_dash + B' is 1 \n")
17
18 printf("A=1 and B=1 give A'=0 and B'=0 so Y_dash = A
       '.B is 0 and Y = Y_dash + B' is 0 \n")
```

Scilab code Exa 28.6 factorize

```
1 // chapter28
2 // example28.6
3 //page606
4
5 printf("Y = A . B . C' . D' + A' . B . C' . D' + A'
      . B . C . D' + A . B . C . D' \n")
6 printf("taking out the common factors \n")
7 printf("Y = B . C' . D' . ( A + A' ) + B . C . D' .
      ( A + A') \n")
8 printf("By theorem 3 \n")
9 printf("Y = B . C' . D' + B . C . D' \n")
10 printf("again factorize \n")
11 printf("Y = B . D' ( C + C' ) \n")
12 printf("By theorem 3 \n")
13 printf("Y = B . D' . 1 \n")
14 printf("thus \n")
15 printf("Y = B . D' \n")
```

Scilab code Exa 28.7 Apply cummulative law and theorem

```
1 // chapter28
2 // example28.7
3 //page606
4
5 printf("Y = A . B + A . ( B + C ) + B . ( B + C ) \n
      ")
6 printf("By thoerem 14 \n")
7 printf("Y = A . B + A . B + A . C + B . B + B .C \n"
      )
8 printf("By theorem 6 \n")
9 printf("Y= A . B + A . B + A . C + B + B .C \n")
10 printf("By theorem 5 \n")
11 printf("Y = A . B + A . C + B + B . C \n")
12 printf("Factor B out of last 2 terms \n")
13 printf("Y = A . B + A . C + B . ( 1 + C ) \n")
```

```
14  printf (" Apply cummulative law and theorem 7 \n")
15  printf ("Y = A . B + A . C + B . 1 \n")
16  printf (" Apply theorem 2 \n")
17  printf ("Y = A . B + A . C + B \n")
18  printf (" Factor B out of first and third terms \n")
19  printf ("Y = B . ( A + 1 ) + A . C \n")
20  printf (" Apply theorem 7 \n")
21  printf ("Y = B . 1 + A . C \n")
22  printf (" Apply theorem 2 \n")
23  printf ("Y = B + A . C \n")
```

Scilab code Exa 28.8 `Based on theorem 16 \& 3`

```
1  // chapter28
2  // example28 .8
3  // page607
4
5  printf (" i ) Y = A + A' . B \n")
6  printf ("    By theorem 16 \n")
7  printf ("    Y = A + A . B + A' . B \n")
8  printf ("      = A + B ( A + A') \n")
9  printf ("    By theorem 3 \n")
10 printf ("    Y = A + B \n \n")
11
12 printf (" ii ) Y = A . B + A' . C + B . C \n")
13 printf ("        = A . B + A' . C + B . C ( A + A' )\n"
       )
14 printf ("        = A . B + A' . C + A . B . C + A' . B
       . C \n")
15 printf ("        = A . B ( 1 + C ) + A' . C( 1 + B ) \n
       ")
16 printf ("        = A . B + A' . C \n")
```

223

Scilab code Exa 28.9 `Using De Morgan theore`

```
1  //chapter28
2  //example28.9
3  //page607
4
5  printf("Y = (  ( A + B )' . C . D' )' \n")
6  printf("Using De Morgan theorem \n")
7  printf("Y = ( A + B ) + C' + D \n")
8  printf("Y = A + B + C' + D \n")
```

Scilab code Exa 28.10 `De Morgan theorem`

```
1  //chapter28
2  //example28.10
3  //page607
4
5  printf("1) Y  = A . B . C' + A . ( B . C )' \n")
6  printf("   Y' = ( A . B . C' + A . ( B . C )' )' \n"
       )
7  printf("   By De Morgan theorem \n")
8  printf("   Y' = ( A . B . C')' . ( A . ( B . C)' )'
       \n")
9  printf("   By De Morgan theorem \n")
10 printf("   Y' = ( A' + B' + C ) . ( A' + B + C ) \n
       \n")
11 printf("2) Y  = A' . ( B .C' + B' . C ) \n")
12 printf("   Y'  = ( A' . ( B .C' + B' . C ) )' \n")
13 printf("   By De Morgan theorem \n")
14 printf("   Y' = A + ( B . C' + B' .C )' \n")
15 printf("   By De Morgan theorem \n")
16 printf("   Y' = A + ( B . C')' . ( B' . C )' \n")
17 printf("   By De Morgan theorem \n")
18 printf("   Y' = A + ( B' + C ) . ( B + C') \n")
19 printf("   Y' = A + ( B . C )' + ( B . C ) \n")
```

Scilab code Exa 28.11 De Morgan theorem

```
1  //chapter28
2  //example28.11
3  //page608
4
5  printf(" 1) Y = ( A + B + C ) . ( A + B ) \n")
6  printf("     Y = A . A + A . B + B . A + B . B + C . A
        + C . B \n")
7  printf("     Using A . A = A we get \n")
8  printf("     Y = A + A . B + A . B + B + A . C + B . C
        \n")
9  printf("     Using A . B + A . B = A . B we get \n")
10 printf("     Y = A + A . B + B + A . C + B . C \n")
11 printf("     Using A + A . B = A we get \n")
12 printf("     Y = A + B + A . C + B . C \n")
13 printf("        = A . ( 1 + C ) + B . ( 1 + C ) \n")
14 printf("     Using 1 + C = 1 we get \n")
15 printf("     Y = A . 1 + B . 1 \n")
16 printf("     Y = A + B \n \n")
17
18 printf(" 2) Y = A . B + A . B . C + A . B . C' \n")
19 printf("        = A . B + A . B ( C + C' ) \n")
20 printf("     Since C + C' = 1 we get \n")
21 printf("     Y = A . B + A . B \n")
22 printf("        = A . B \n \n")
23
24
25 printf(" 3) Y = 1 + A . ( B . C' + B . C + B' . C') +
        A . B' . C + A . C \n")
26 printf("     Using 1 + A = 1 and 1 + A . ( B . C' + B
        . C + ( B . C )' ) = 1 we get \n")
27 printf("     Y = 1 + A . B' . C + A . C \n")
28 printf("     Y = 1 + A . C \n")
```

225

```
29  printf(" Y = 1 \n \n")
30
31  printf("4) Y = ( ( A + B' + C ) + ( B + C' ))' \n")
32  printf(" By De Morgan theorem \n")
33  printf(" Y = ( A + B' + C )' . ( B + C' )' \n")
34  printf(" By De Morgan theorem \n")
35  printf(" Y = ( A' . B . C' ) . ( B' . C ) \n")
36  printf(" Since B . B' = 0 and C . C' = 0 we get \n
       ")
37  printf(" Y = 0 \n")
```

Scilab code Exa 28.12 `theorem`

```
1  //chapter28
2  //example28.12
3  //page609
4
5  printf(" Y = A . B' . D + A . B' . D' \n")
6  printf(" Factor out A . B' by theorem 14 \n")
7  printf(" Y = A . B' ( D + D' ) \n")
8  printf(" But by theorem 3 D + D' = 1 \n")
9  printf(" Y = A . B' . 1 \n")
10 printf(" By theorem 2 \n")
11 printf(" Y = A . B' \n")
```

Scilab code Exa 28.13 `theorem`

```
1  //chapter28
2  //example28.13
3  //page609
4
5  printf(" Y = ( A' + B ) . ( A + B ) \n")
6  printf(" By theorem 15 \n")
```

```
 7  printf (" Y = A' . A + A' . B + B . A + B . B \n")
 8  printf (" By theorem 4 and 6 \n")
 9  printf (" Y = 0 + A' . B + B . A + B \n")
10  printf (" Y = A' . B + B . A + B \n")
11
12  printf (" By theorem 14 \n")
13  printf (" Y = B . ( A' + A + 1 ) \n")
14  printf (" By theorem 7 \n")
15  printf (" Y = B . ( A' + 1 ) \n")
16  printf (" By theorem 7 \n")
17  printf (" Y = B . 1 ) \n")
18  printf (" By theorem 2 \n")
19  printf (" Y = B \n")
```