Scilab Textbook Companion for Linear Integrated Circuits by D. R. Choudhury[1]

Created by
Vishal Changlani
Electronics and Telecommunication
Electronics Engineering
VES Institute of Technology
College Teacher
None
Cross-Checked by
None

July 31, 2019

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

Book Description

Title: Linear Integrated Circuits

Author: D. R. Choudhury

Publisher: New Age, New Delhi

Edition: 4

Year: 2010

ISBN: 978-81-224-3098-1

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

Exa Example (Solved example)

Eqn Equation (Particular equation of the above book)

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

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

Contents

List of Scilab Codes 4

2 Operational Amplifier 5

3 Operational Amplifier Characteristics 26

4 Operational Amplifier Circuits 32

5 Comparators and Waveform Generators 45

6 Voltage Regulator 51

7 Active Filters 55

8 555 Timer 67

10 D to A and A to D Converters 69

List of Scilab Codes

Exa 2.1	Design an amplifier for given conditions	5
Exa 2.2	To calculate various parameters for given circuit	6
Exa 2.3	To design an amplifier using single op amp	7
Exa 2.4	To determine the various parameters for Non inverting amplifier	7
Exa 2.6	To determine the values of Rc and Re	8
Exa 2.7	To determine the various parameters for basic differential amplifier	9
Exa 2.8	To determine the output voltages vo1 and vo2	11
Exa 2.9	To find the differential voltage gain	12
Exa 2.10	To find the value of Resistor R1	13
Exa 2.11	To design a Wildar current source	14
Exa 2.12	To determine the values of Ic1 Ic2 and Ic3	15
Exa 2.13	To determine the value of Io	16
Exa 2.14	To determine the values of Ic1 Ic2 and Rc	17
Exa 2.15	To find the emitter current in transistor Q3	18
Exa 2.16	To calculate the difference between V1 and V2 for the level shifter	19
Exa 2.17	To calculate collector current in each transistor and the output voltage under quiescent condition along with open loop voltage gain	20
Exa 2.18	To perform the dc analysis and to compute overall voltage gain	23
Exa 3.1	To design an inverting amplifier using 741 opamp	26
Exa 3.2	To calculate maximum output offset voltage along with value of Rcomp	27

Exa 3.3 To find the change in output voltage due to change in temperature 28
Exa 3.4 To find the slew rate of the opamp whose output is shown in figure 29
Exa 3.5 To determine maximum peak to peak input signal that can be applied without distorting the output 29
Exa 3.6 To determine whether IC 741 can be used or not for given specifications 30
Exa 4.1 To design an adder circuit using an operational amplifier 32
Exa 4.2 To find the output voltage for the adder subtractor circuit 33
Exa 4.3 To design an operational amplifier to differentiate an input signal 34
Exa 4.4 To determine various parameters for a lossy integrator 36
Exa 4.5 To find R1 and Rf values in the lossy integrator 39
Exa 4.6 To calculate output of an operational amplifier to a step input of magnitude V and compare it with output of a low pass RC circuit 40
Exa 4.7 To show that output is given by a differential equation 42
Exa 4.8 To show that output is integral of input . . 43
Exa 4.9 To set up an analog simulation to generate a sinusoidal signal 43
Exa 5.1 To plot the transfer curve for opamp with given conditions 45
Exa 5.2 To determine the upper and lower threshold voltages 47
Exa 5.3 To calculate time period of the negative and positive portion of the output waveform . . 48
Exa 5.4 To design a phase shift oscillator to oscillate at 100 Hz 49
Exa 6.1 To calculate the output current coming from 7805 51

Exa 6.2 To calculate component values to get desired output voltage 53
Exa 7.1 Design a second order Butterworth low pass filter 55
Exa 7.2 Design a fourth order Butterworth low pass filter 57
Exa 7.3 To determine order of a low pass Butterworth filter 58
Exa 7.4 Design a second order Butterworth high pass filter 59
Exa 7.5 Design a wide band pass filter 61
Exa 7.6 Design a 50 Hz active notch filter 62
Exa 7.7 Design a wide band reject filter 63
Exa 7.9 Design a switched capacitor integrator . . . 64
Exa 8.1 To calculate value of Capacitor 67
Exa 8.2 To calculate various parameters of Astable multivibrator 68
Exa 10.1 To calculate output voltage for given 9 bit input 69
Exa 10.2 To calculate LSB MSB and full scale output for an 8 bit DAC 70
Exa 10.3 To calculate output voltage for different input 70
Exa 10.4 To calculate value of the resistor R of the integrator 71
Exa 10.5 To find the equivalent digital number 72

Chapter 2

Operational Amplifier

Scilab code Exa 2.1 Design an amplifier for given conditions

```
// Exa 2.1

clc;
clear;

// Given data

// An amplifier( Refer Fig. 2.5(a) )
Acl = -10; // Closed loop gain
Ri = 10 * 10^3; // Input resistance of amplifier( )

// Solution

// Since it is mentioned to design an amplifier, it means to calculate values for Rf(Feedback resistance) and R1.
disp("Referring Fig. 2.5(a), we choose R1 as 10 k i.e equal to input resistance of amplifier.");
R1 = Ri;
// Acl = -1 * Rf/R1;
// Therefore;
```

```
Rf= - Acl * Ri;
printf(' The calculated value of Rf(Feedback
   resistane) is Rf = %d k  . \n',Rf/1000);
```

Scilab code Exa 2.2 To calculate various parameters for given circuit

```
// Exa 2.2

clc;
clear;

// Given data

// An amplifier as given in Fig. 2.5(b)
R1 = 10*10^3;    //Input resistance of amplifier (  )
Rf = 100*10^3;   // Feedback resistance of amplifier
   (  )
vi = 1;          // Input voltage applied (Volts)
RL = 25*10^3;     // Load resistance (  )

// Solution

i1 = vi/R1; //Input current(A)
printf(' The value of input current = i1 = %.1f mA.
   \n ',i1*1000);
vo = -1*(Rf/R1)*vi; // output voltage(V)
printf(' The value of output voltage = vo = %d V. \n
    ',vo);
iL = abs(vo)/RL;  // Load current(A)
printf(' The value of load current = iL = %.1f mA.',
   iL*1000);
disp(" The direction of iL is as shown in Fig. 2.5(b
   ).");
// iTot = i1 + iL;
iTot = i1+iL;  // Total current(A)
```

```
printf(' The value of total current = io = %.1f mA.',iTot*1000);
disp(" In an inverting amplifier, for a +ive input, output will be -ive, therefore the direction of io is as shown in Fig. 2.5(b).");
```

Scilab code Exa 2.3 To design an amplifier using single op amp

```
// Exa 2.3

clc;
clear;

// Given data

// Single op-amp amplifier.
ACL = 5; // Required gain(positive)

// Solution

disp("Since the gain is positive, we have to make a non-inverting amplifier.");
disp("Referring Fig. 2.7(a), select R1 = 10 k .");
R1 = 10*10^3; //Input resistance in
// Then from eqn. (2.20), we get, ACL = 1+ (Rf/R1);
// Therefore
Rf = (ACL-1)*R1;  //Feedback resistance in
printf(' The calculated feedback resistance of amplifier i.e Rf = %d k .\n',Rf/1000);
```

Scilab code Exa 2.4 To determine the various parameters for Non inverting amplifie

```
// Exa 2.4
```

```

clc;
clear;

// Given data

// Referring circuit given in  Fig. 2.7(a)
R1= 5*10^3;  //
Rf=20*10^3; //
Vi=1; // Input voltage(V)
RL=5*10^3; // Load resistor(  )


// Solution

Vo= (1+(Rf/R1))*Vi;  // Output voltage(V)
printf('The output voltage i.e vo = %d  V. \n',Vo);
AcL=Vo/Vi;  // Closed loop Gain
printf(' The  closed loop gain i.e Acl = %d.   \n',
    AcL);
IL=Vo/RL; // Output current(A)
printf(' The  load current i.e iL = %d mA. \n',IL
    *1000);
I1=Vi/R1; // Input current(A)
Io=IL+I1; // Total current(A)
printf(' The  output current i.e io = %.2f  mA. \n',
    Io*1000);
disp("The op-amp output current Io flows outwards
    from the output junction.");
```

Scilab code Exa 2.6 To determine the values of Rc and Re

```
// Exa 2.6

clc;
```

```
clear;

// Given data

// Referring circuit shown in Fig. 2.11(a)
B=200; // Current gain
Icq = 100*10^-6; // Amperes
ADM = 500; // Voltage gain for differential mode
    signal
CMRR_db = 80; // in dB(Common mode rejection ratio)

// Solution

// Since gm = Icq/Vt therefore ,
gm = Icq/(25*10^-3); // for Vt = 25 mV

printf('Using Eq. 2.50 , we have ADM = -gm*Rc so from
     this we get Rc as ');
Rc =abs(- ADM/gm);
printf(' %d k . \n ',Rc/1000);
printf('Since CMRR = 80 dB converting it into non dB
     value so CMRR = ');
CMRR =  10^(CMRR_db/20);
printf(' %d. \n ',CMRR);
printf('Using Eq. 2.55 , we get value of Re as ');
// CMRR = 1+ 2*gm*Re; therefore
Re = (CMRR-1)/(2*gm);
printf(' %.2f M . \n ',Re/10^6);
```

Scilab code Exa 2.7 To determine the various parameters for basic differential amp

```
// Exa 2.7

clc;
clear;
```

```
// Given data

// Fig. 2.11(a) shows the basic differential
   amplifier
Rc = 2*10^3; //
Re = 4.3*10^3; //
Vcc = 5 ; // Vcc = |VEE|
Bo = 200;
Vbe = 0.7; // Volts
Vt=25*10^-3; // Volts

// Solution

printf(' For V1 = V2 = 0, applying KVL for the base
   emitter loop , we may write ,');
printf('\n  Vbe+2*(1+Bo)*Ibq*Re-Vee = 0.\n  From
   this we get Ibq as ');
Ibq = (Vcc-Vbe)/(2*(1+Bo)*Re);
printf(' %.2f  A . \n ',Ibq*10^6);
Icq = Bo*Ibq;
printf(' The value of Icq = %.3f mA. \n ',Icq*10^3);
Vo1 = Vcc - Rc*Icq;
printf(' The value of Vo1 = Vo2(due to symmetry) = %
   .3f V. \n ',Vo1);
Vceq = Vo1-(-Vbe);
printf(' The value of Vceq = %.3f V. \n ',Vceq);
gm = Icq/Vt;
r_pi = Bo/gm;
// using wq. 2.50 ADM = -gm*Rc;
ADM = -gm*Rc;
// using equation 2.53(a) Acm can be given as
ACM = (-Bo*Rc)/(r_pi+2*(1+Bo)*Re);

CMRR = ADM/ACM;
CMRR_db = 10*log(CMRR);
printf(' The remaining values are as follows: \n
  ADM = %.2f. \n  ACM = %.2f. \n  CMRR = %.1f = %.1
```

```
f dB.\n',ADM,ACM,CMRR,CMRR_db);
```

Scilab code Exa 2.8 `To determine the output voltages vo1 and vo2`

```
// Exa 2.8

clc;
clear;

// Given data

// With reference to differential amplifier designed in example 2.6
// 2 applied inputs are
t = [0 :1:100]; // time in mSec
 v1=  15*sin(2*%pi*60*t) + 5*sin(2*%pi*1000*t); // in mV
 v2 = 15*sin(2*%pi*60*t) - 5*sin(2*%pi*1000*t); // in mV
fi = 60; // frequency of interference signal(Hz)
fo = 1000; // frequency at which signal is to be processed(Hz)

// Solution

// We know from Example 2.6
gm=4; // m
Rc=125 ; // k
Re= 1.25; // k
Bo=200;
r_pi= Bo/gm;  // in k

ADM=-500; // from example 2.6(given)
// From eq. 2.53(a) we get ACM as
ACM = (-Bo*Rc)/(r_pi*1000+2*(1+Bo)*Re);
```

```
printf(' The value of ACM = %.2f \n',ACM);
// from eqns 2.56(a and b)
vDM = (v1-v2)/2;
vCM = (v1+v2)/2;

//from Eq. 2.57(a and b)
vo1 = ADM*vDM+ACM*vCM;
vo2 = -ADM*vDM + ACM*vCM;

printf('  Therefore final equations are- \n');
disp("vo1 = -2500*sin(2*%pi*1000*t)-0.75*sin(2*%pi*60*t) mV ");
disp("vo2 = 2500*sin(2*%pi*1000*t)-0.75*sin(2*%pi*60*t) mV");
```

Scilab code Exa 2.9 To find the differential voltage gain

```
// Exa 2.9

clc;
clear;

// Given data

// A differential amplifier with single ended output
// Referring circuit given in Fig. 2.9
Bo = 100; // current gain
Re = 150; //
Rc = 10*10^3; //
IQ = 0.5*10^-3; // mA
VT = 25*10^-3; // mV

// Solution

ICQ = IQ/2;
```

```
gm = ICQ/VT;  // in
r_pi = Bo/gm;

printf(' The differential mode gain for a single stage is found from the equivalent circuit shown in fig.(ii)\n  on page no. 64 and is equal to ');
ADM = (1/2)*(Bo*Rc/(r_pi+(1+Bo)*Re));
printf(' %d V/V. \n',round(ADM));
printf(' \n  We can see that sign of ADM is positive because the output is taken at the collector of Q2\n  whereas input is applied at the base of Q1.\n');
```

Scilab code Exa 2.10 To find the value of Resistor R1

```
// Exa 2.10

clc;
clear;

// Given data

// A current mirrir as shown in Fig. 2.16
Ic = 1; // mA
Vcc = 10; // Volts
B = 125;
Vbe = 0.7; // Bolts

// Solution

// Case(1)- When Ic = 1mA.
printf(' From equations 2.67 and 2.68 we get R1 as - \n\n');
// Ic = (B/(B+2))*((Vcc-Vbe)/R1);
// Therefore
```

```
R1 = (B/(B+2))*((Vcc-Vbe)/Ic);
printf(' The value of R1 when Ic = 1 mA is R1 = %.2f k . \n',R1);

// Now case(2)- when Ic = 10 A .
Ic1 = 10*10^-3; // in mA
R2 = (B/(B+2))*((Vcc-Vbe)/Ic1);
printf(' The value of R1 when Ic = 10 A is R1 = %d k . \n',R2);
```

Scilab code Exa 2.11 To design a Wildar current source

```
// Exa 2.11

clc;
clear;

// Given data

Io=10*10^-6; // Output current(A)
Vcc= 10; // Volts
B=125; // current gain
Vbe=0.7; // Voltage between base and emitter(V)
Vt=25*10^-3; // volt equivalent of temperature at room temperature(V)

//Solution

disp(" Let Iref = 1 mA then using equation 2.79, we get- ");

Iref=1*10^-3; // we choose

R1=(Vcc-Vbe)/Iref;
printf('\n The value of R1 = %.1f k . \n',R1/1000)
```

```
;

disp(" Using equation 2.74, we get-");
Re=(Vt/((1+1/B)*Io))*log(Iref/Io);
printf('\n The value of Re = %.1f k . \n',Re/1000);

disp(" Thus, it is clearly seen that Wildar circuit allows the generation of small currents using relatively small resistors.");
```

Scilab code Exa 2.12 To determine the values of Ic1 Ic2 and Ic3

```
// Exa 2.12

clc;
clear;

// Given data

//Referring fig No. 2.21, we get

Vcc=9; // Volts
Vbe=0.7; // Volts
R1=30*10^3; //
Re=1.94; //
B=125; // current gain
VT = 25*10^-3; // Volts

// Solution

Iref= (Vcc-Vbe)/R1;
printf(' The value of Iref = %.3f mA. \n ',Iref*1000);
// Also at Node A.- Iref=Ic+3*Ib. i.e Ic = Iref*(B/(
```

```
   B+2))
// Assuming IB3 of widlar source negligible.
// Therefore putting back value of Iref we get
   values of Ic1
Ic=Iref*(B/(B+3));
Ic_mA = Ic*1000;  // in mA

printf('\n  The value of Ic1 = Ic2 = %.3f mA. \n ',
   Ic*10^3);
// Calculating Ic3 using eqn 2.74 ;

// Re = (VT/(Ic3*(1+1/B)))*ln(ic_mA/Ic3);
// Re - (VT/(Ic3*(1+1/B)))*ln(ic_mA/Ic3) = 0;

deff('y =  f(x)', 'y = (Re-(VT*log(Ic_mA/x))/(x
   *(1+1/B)))'); // here x = Ic3
[x,v,info]= fsolve(0.01,f);

printf(' \n  By trial and error method, we get Ic3 =
    %.4f mA.\n',x);
```

Scilab code Exa 2.13 To determine the value of Io

```
// Exa 2.13

clc;
clear;

// Given data

// Refering Fig. 2.24 we get,

B=100;  // Current gain
Vbe=0.7; // Volts
Vcc=5; // Volts
```

```
R1= 10*10^3; //

// Solution

printf(' Referring to circuit shown in Fig. 2.24 and
    using KVL we get Iref as ' );
// KVL for loop 1
// Vcc-Vbe-R1*Iref+Vcc = 0;
// Therefore
Iref= (2*Vcc-Vbe)/R1;
printf(' %.2f mA. \n',Iref*1000);
// At emitter node E., Iref=2*Ie (Assuming identical
    transistors
//Then;
Ic= B*Iref/(2*(1+B));

printf('  Due to mirror effect , Io = Ic1 = Ic = %0.2
   f mA. \n ',Ic*1000);
```

Scilab code Exa 2.14 To determine the values of Ic1 Ic2 and Rc

```
// Exa 2.14

clc;
clear;

// Given data

// Fig. 2.25 shows circuit for current example
Vo = 6; // Volts
B = 200;
R1 =15; // k
R2 = 2.8; // k
Vcc = 12; // volts
Vbe = 0.7; // Volts
```

```
// Solution

Iref = (Vcc-Vbe)/R1;
I1 = Vbe/R2;

// At node A- Iref = Ic1 + 2IB + I1;
 //             Iref = Ic1*(1+2/B)+I1;
  // Therefore; we get Ic1 as-
Ic1 = (Iref-I1)/(1+2/B);
printf(' The value of Ic1 = %.3f mA.\n',Ic1 );
printf('  The value of Ic2 = Ic1 due to mirror
   effect.');
 // by KVL to outer loop we get Rc value
 // 12V = Ic2*Rc +Vo;
Rc = (Vcc-Vo)/Ic1;
printf('\n  The value of Rc = %d k  . \n',Rc);
```

Scilab code Exa 2.15 To find the emitter current in transistor Q3

```
// Exa 2.15

clc;
clear;

// Given data

// Referring Fig. 2.26, we get
Vcc=10; // Volts
R1= 4.7*10^3; // k Ohms
B=100; // Current gain(>>1)
Vbe=0.75; // Volts

// Solution

```

```
disp(" Node    A    is at transistor Q1, Node    B    is at transistor Q2 and Node    C    is at transistor Q3.");

printf('\n  From Fig. 2.26 at node    A    . I = Ic1 + Ib1 +  I 1   ...Eqn(1)');
printf(' \n  Also at node    B    .  I 1   = Ic2 + Ib3 .');
printf('\n  Putting value of  I 1   in eqn(1) we get I = (approximately) 2Ic. \n');

I = (Vcc-Vbe)/R1;  // By  o h m s  law

printf('\n The calculated value of I = %.2f mA. \n', I*1000);

Ic3 = I/2;

printf(' The collector current of Q3 is equal to the  collector current of Q1 and Q2 due to mirror action. \n Therefore, the emitter current IE3 = Ic3 = Ic = I/2 = %.2f mA. \n ',Ic3*1000);
```

Scilab code Exa 2.16 To calculate the difference between V1 and V2 for the level s

```
// Exa 2.16

clc;
clear;

// Given data

// A level shifter as shown in fig. 2.31
// Assuming Ideal silicon transistors
Vbe = 0.7; // Volts
```

```
// B(current gain) has very large values
Vcc = 15; // Volts
Rc = 10*10^3; //
Re = 5000; //

// Solution

printf(' From fig. 2.31 we get that, transistors Q1
    and Q2 form a current mirror.\n');
printf('  so Ic1 = Ic2 = I and that can be found by
    O h m s law as ');
I = (Vcc - Vbe)/Rc; //
printf(' I = Ic2 = %.2f mA. \n', I*1000 );
printf('  Now, the difference V1-V2 can be found
    using KVL as ');
dV = Vbe + I * Re;  // KVL between end points
printf(' %.2f V. \n',dV);
```

Scilab code Exa 2.17 To calculate collector current in each transistor and the out

```
// Exa 2.17

clc;
clear;

// Given data

// Fig. 2.33- Internal circuit of Motorola MC 1530
B = 100;
Vbe = 0.7; Vd = 0.7; Vcc = 6; Vee = 6; Vo = 5; Vt=
    0.026; // Volts
// Vt = volt equivalent of temperasture at room
    temperature
R1 = 2.2; R2 = 7.75; R3 = 7.75; R4 = 1.5; R5 = 3.2;
    R6 = 1.5; R7 = 3;
```

```
R8 = 3.4; R9 = 6; R10 = 30; R11 = 5; // All in k

// Solution

printf(' The d.c. analysis is performed by assuming that both the inverting and non-inverting terminals are at ground potential.');

Vbn1 = (-Vee+Vd+Vd)*R5/(R4+R5);
printf('\n  The voltage Vbn1 at base of transistor Q1 w.r.t ground N  = %.2f V. \n',Vbn1);
I1 = (Vee + Vbn1 - Vbe)/R1;
printf('\n The current through emitter of Q1 i.e I1 = %.3f mA. \n',I1);
printf(' If the base current of Q1 is neglected, then Iq(current through collector of Q1) = I1 = %.3f mA. \n',I1);

printf('\n under dc conditions, half of the Iq flows through each transistors Q2 and Q3.Therefore Ic2 = Ic3 = %.3f mA. \n',(I1/2));
Vc2 = Vcc - R2*(I1/2);
printf(' The voltage at collector of Q2 and Q3 = Vc2 = Vc2 = %.2f V. \n',Vc2);
printf('\n By looking at the Internal circuit, the voltages at the base of Q4 and  Q5 i.e Vb4 = Vb5 = %.2f V. \n',Vc2);
Ve4 = Vc2 - Vbe;
printf(' The dc voltage at the emitter of Q4 = Ve4 = %.2f V.\n',Ve4);
I6 = Ve4/R6;
printf('\n The current through R6 = %.3f mA. \n',I6);
printf(' This current divides equally in transistors Q4 and Q5 i.e Ic4 = Ic5 = %.3f mA. \n',(I6/2));
Vc5 = Vcc-(I6/2)*R7;
Ve6 = Vc5-Vbe;
printf('\n The collector voltage of Q5i.e Vc5 and
```

```
    emitter voltage of Q6 i.e are %.2f V and %.2f V
    respectively. \n',Vc5,Ve6);
I8 = (Vee-Vd)/R8;
printf(' Transistors Q7 along with diode D3 forms a
    current mirror of the type shown in fig. 2.15.
    Hence, Ic7 = I8 = %.2f mA. \n',I8);

Vb8 = Vbe + Vd - Vee;
printf('\n The voltage at the base of Q8 i.e Vb8 = %
    .2f V. \n',Vb8);
I9 = (Ve6-Vb8)/R9;
I10 = I8-I9;
printf(' The currents along resistors R9 and R10 are
     %.2f mA and %.2f mA respectively. \n',I9,I10 );

Vo_dc = I10*R10 + Vb8;
printf(' The voltage at output terminal = %.2f V (
    approximately 0)  as expected. \n',Vo_dc);

printf('\n\n');
printf('/////////////////////////////////FOR over all
     voltage gain/////////////////////\n\n');
h_fe = 100;
printf(' we first calculate the voltage gain of the
    differential amplifier stages.\n');
// since Ic2 = Ic3 = Ic4 = Ic5 = approx 0.50 mA
Ic = 0.5; // in mA
h_ie = h_fe*Vt/Ic;

RL2 = (1/R2 + 1/h_ie)^-1;
RL3 = RL2;

Av1 = h_fe*RL2/h_ie;
printf(' The output of the first stage is double
    ended, its differential gain is given as %d. \n',
    Av1);

Av2 = -(1/2)*(h_fe*R7/h_ie);
```

```
printf(' The output of the second stage is single
    ended, so its differential gain is %.1f. \n',Av2)
    ;

printf(' The third stage is emitter follower, so,
    Av3 = (approximately)1. \n');
Av3 = 1;
Av4 =- R10/R9;
printf(' The last output stage uses voltage shunt
    feedback network R9-R10, \n So, Av4 = (
    approimately) %d. \n',Av4);

Av = Av1 * Av2 * Av3 * Av4;
printf(' \n\n Hence the over all op-amp gain is, Av
    = %d. \n ' ,Av);
```

Scilab code Exa 2.18 To perform the dc analysis and to compute overall voltage gai

```
// Exa 2.18

clc;
clear;

// Given data

// op-amp circuit as shown in Fig. 2.35
h_fe = 100;
Vbe = 0.7; Vcc = 15; Vee = 15; Vt = 0.025; // Volts
// Vt = volt equivalent at room temperature
R1 = 20; R2 = 20; R3 = 28.6; R6 = 3; R8 = 2.3;
R9 = 3; Ra = 15.7; // All in k

// Solution

printf('It can be seen that the circuit has the four
```

```
    stages: \n Dual-input, differential output. \n
    Dual-input, single ended output. \n Level
    translator. \n Emitter follower.\n ');

printf('\n\n For d.c. analysis, assume that the
    input terminals are shorted to ground.\n');
I = (Vee-Vbe)/R3;
printf(' The reference current I of the current
    minor Q3-Q4 is obtained as I = %.1f mA. \n',I);
printf('\n Due to current mirror action, Icq4 = I =
    %.1f mA and Icq1 = Icq2 = Icq/2 = %.3f mA. \n',I,
    I/2);
Vcq1 = Vcc-(I/2)*R1;
printf(' The collector voltages for Q1 and Q2 are
    Vcq1 = Vcq2 = %d V. \n',Vcq1);
Veq5 = Vcq1-Vbe;
printf('\n The voltage at emitter of Q5 and Q6 is
    Veq5 = Veq6 = %.1f V. \n',Veq5);
Icq7=4*I;
printf(' Since the area of Q7 is 4 times that of Q3
    and Q4, the transistor Q7 supplies a current Icq7
     = %d mA. \n',Icq7);
printf('\n Thus, collector currents of Q5 and Q6 are
     Icq5 = Icq6 = %d mA. \n',Icq7/2 );
Vcq6 = Vcc-(Icq7/2)*R6;
printf(' Hence, collector voltage of Q6 is Vcq6 = %d
     V. \n',Vcq6);
Veq8 = Vcq6+Vbe;
printf('\n This causes a voltage at the emitter of
    pnp transistor Qr i.e Veq8 = %.1f V. \n',Veq8);
Ieq8 = (Vcc-Veq8)/R8;
printf(' The emitter current of Q8 i.e Ieq8 = %d mA.
     \n',Ieq8);
Va = -Vee + Ieq8*Ra;
printf('\n The voltage Va at the collector of Q8 i.e
     Vcq8 or the base of Q9 i.e Vbq9 = %.1f V. \n',Va
    );
printf(' Since the emitter of Q9 will be 0.7 V below
```

```
the base terminal, \n the voltage at the output terminal 6 is 0 V as is expected.');

Vo = 0; // output voltage(Volts)
Ieq9 = (Vo-(-Vee))/R9;
printf('\n\n The emitter current of Q9 i.e Ieq9 = %d mA. \n',Ieq9);

printf('\n\n//////////////////////////a.c. analysis////////////////////////////\n\n');

h_ie1 = (h_fe*Vt)/(I/2);
printf(' The ac emitter resistance of the transistor Q1-Q2 is h_ie = %.1f k . \n',h_ie1);
h_ie5 = (h_fe*Vt)/(Icq7/2);
printf(' The ac emitter resistance of the transistor Q5-Q6 is h_ie = %.1f k . \n',h_ie5);
RL1 = (1/R1 + 1/h_ie5)^-1;
printf(' Since, emitter of Q5-Q6 is at ground potential under ac conditions,\n the effective load for Q1-Q2 is RL1 = RL2 = %.1f k .\n',RL1);
ADM1 = (h_fe*RL1)/h_ie1;
printf(' The voltage gain of first differential stage is ADM1 = %d. \n',ADM1);
ADM2 = (-1/2)*(h_fe*R6)/h_ie5;
printf(' The voltage gain of second differential stage is ADM2 = (approximately) %d. \n',ADM2);
A3 =-Ra/R8;
printf(' The gain of the level translator stage is A3 = %.2f. \n',A3);
printf(' The last stage is emitter follower, so its voltage gain Av4 = (approximately) 1.');
Av4 = 1;
Av = ADM1 * ADM2 * A3 * Av4;
printf('\n\n So, the overall voltage gain is Av = %d. \n',Av);
```

Chapter 3

Operational Amplifier Characteristics

Scilab code Exa 3.1 To design an inverting amplifier using 741 opamp

```
// Exa 3.1

clc;
clear;

// Given data

//Fig. 3.1(e) shows an inverting amplifier
ACL = -10; // open loop gain of ap-amp 741
R1 = 10*10^6; // Input impedence in

// Solution

printf(' In fig. 3.1(e), to set input impedence Ri =
    10 M   , pick R1 = 10 M  . ');
// since , ACL = - Rf/R1;
// Therefore ,
Rf = -ACL*R1;
printf('\n  The calculated value of Rf = %d M  . \n
```

```
',Rf/10^6);
printf(' Choose Rt = 47 k . \n ');
Rt = 47*10^3; //
Rs = (Rt^2)/(Rf-2*Rt);
printf(' Calculated Rs = %d . ',Rs);
```

Scilab code Exa 3.2 To calculate maximum output offset voltage along with value of

```
// Exa 3.2

clc;
clear;

// Given data

// Fig. 3.2(b) represents the non-inverting amplifier
R1 = 1000; //
Rf = 10000; //
Vios = 0.01; // Volts
Ib = 300*10^-9; // Amperes
Ios = 50*10^-9; // Amperes

// Solution

printf(' From equation 3.24, we get VoT = ');
VoT = (1+(Rf/R1))*Vios + Rf*Ib ;
printf(' %d mV. \n ',VoT*1000);

Rcomp = 1/((1/R1) + (1/Rf)); // Rf || R1
printf(' The value of Rcomp needed to reduce the effect of Ib is %.1f . \n ',Rcomp);
printf(' With Rcomp in the circuit, VoT = ');
VoT1 = (1+(Rf/R1))*Vios + Rf*Ios;
printf(' %.1f mV. \n ' , VoT1*10^3);
```

```
printf('\n It can be seen from this example that it
   is the input offset voltage which is more
   responsible\n for producing an output offset
   voltage compared to input bias current Ib or the
   input offset current Ios.');
// The answer provided in the textbook is wrong.
```

Scilab code Exa 3.3 To find the change in output voltage due to change in temperat

```
// Exa 3.3

clc;
clear;

// Given data

// A non-inverting amplifier
G=100;// Gain of amplifier at 25 degree celsius
T1 = 25; // degree celsius
T2 = 50; // degree celsius
VoT=0.15; // Offset voltage drift in mV/
   degreecelsius

// Solution

printf(' Input offset voltage due to temperature
   rise = ');
Vos=VoT*(T2-T1);
printf(' %.2f mV. \n ',Vos);
printf(' Due to this input change, the output
   voltage will change by ');
Vo=Vos*G;
printf( '%d mV. \n ',Vo);
printf(' This could represent a very major shift in
   the output voltage.');
```

Scilab code Exa 3.4 To find the slew rate of the opamp whose output is shown in fi

```
// Exa 3.4

clc;
clear;

// Given data

// Fig. 3.13 shows output of an op−amp voltage
   follower
F =2; // frequency in MHz
Vipp= 8; // Input voltage (Peak to peak) in volts

// Solution

printf(' Since the frequency is given we can get
   time period as T = %.1f  sec . \n\n',1/F);

printf('  Since, the slew rate is defined as the
   maximum rate of change of the output, \n  so from
    Fig. 3.13, it can be seen that, maximum change
   in output is 6V in 0.25  sec .');

dVo=6;
dT=0.25 ; //  sec
SR=dVo/dT;

printf('\n\n  Therefore, the slew rate of the op−amp
    = %d V/ sec . \n',SR);
```

Scilab code Exa 3.5 To determine maximum peak to peak input signal that can be app

```
// Exa 3.5

clc;
clear;

// Given data

// A 741C op-amp is used
G=50; // Gain of op-amp
F=20*10^3; // Voltage gain Vs frequency curve is flat upto this frequency(Hz)

// Solution

printf(' The slew rate for 741C is 0.5 V/ sec . \n');
SR=0.5; // V/ sec
printf(' From Eq. 3.51, we can get Vm as : ');
// SR = 2*%pi*f*Vm/10^6; // V/ sec
Vm = SR*10^6/(2*%pi*F);
printf('%.2f V. \n ',Vm);
Vpp=2*Vm;
printf(' The peak to peak output voltage = %.2f V. \n',Vpp);
printf(' Hence, for the output to be undistorted sine wave, the maximum input signal should be less than %d mV peak to peak.\n',(Vpp*10^3)/G);
```

Scilab code Exa 3.6 To determine whether IC 741 can be used or not for given speci

```
// Exa 3.6

clc;
clear all;

```

```
// Given data

Vinpp=500; // Peak to peak input voltage in mV
Vopp= 3; //  Peak to peak output voltage in V
Tr= 4; // Rise time in sec

// Solution

printf('Since the output has a peak amplitude
   greater than 1 volt, the slew rate is the
   limiting factor.\n ');

// The slew rate = dVo/dT;

printf('\n From the definition of rise time, it is
   time the output takes to change from 10 to 90
   percent of the final value. \n \n Therefore,the
   change in the output voltage dVo in 4 microsec is
    equal to :');
dVo = (0.9-0.1)*Vopp;
printf(' %.1f V. \n',dVo);

SR = dVo/Tr;
printf(' The slew rate is %.1f V/ sec . \n ',SR);
printf('\n Since, the slew rate of 741 is 0.5 V sec
   , it is too slow and cannot be used.');
```

Chapter 4

Operational Amplifier Circuits

Scilab code Exa 4.1 `To design an adder circuit using an operational amplifier`

```
// Exa 4.1

clc;
clear;

// Given data

// To design an adder circuit as shown in Fig. 4.2(a)
// Vo = -(0.1*V1+V2+10*V3);
// V1,V2,V3 are the inputs

// Solution

printf(' The output in Fig. 4.2(a) is - \n Vo = -[(Rf/R1)*V1 + (Rf/R2)*V2 + (Rf/R3)*V3].');
printf('\n  The desired output is -\n Vo = [(0.1)*V1 + (1)*V2 + (10)*V3].');
printf('\n\n Comparing above two equations,');
printf('\n We can say, Let Rf = 10 k  , R1 = 100 k  and R2 = 10 k  and R3 = 1 k  .\n');
```

```
printf('\n Thus, the desired output expression is
   obtained.');
```

Scilab code Exa 4.2 To find the output voltage for the adder subtractor circuit

```
// Exa 4.2

clc;
clear;

// Given data

// Added−subtractor as shown in fig. 4.4(a).
R1=40*10^3; //
R2=25*10^3; //
R3=10*10^3; //
R4=20*10^3; //
R5=30*10^3; //
Rf=50*10^3; //
V1=2; // Volts
V2=3; // Volts
V3=4; // Volts
V4=5; // Volts

// Solution

printf('The negative sum is obtained by setting V3=
   V4=0. Thus,\n ');
 Vo1=-(Rf/R1)*V1-(Rf/R2)*V2;
printf(' Vo1 = %.1f Volts. \n ',Vo1);
printf('\n Now set V1=V2=0 to find the output
   voltage due to V3 and V4. \n The voltage Vo2 at
   the positive terminal due to V3 and V4 can be
   found by using superposition theorem as shown in
   Fig. 4.4(b) as \n ');
```

```

Rllel=( 1/R4 + 1/R5)^-1;
Rllel1=(1/R3+1/R5)^-1;
Vo2= (Rllel/(R3+Rllel))*V3+ (Rllel1/(Rllel1+R4))*V4;
printf(' Vo2 = %.3f Volts. \n ',Vo2 );
printf('\n The output voltage Vo3 due to V3 and V4 now can be determined from the equivalent circuit of Fig. 4.4(c) as \n ');
Rllel2=(1/R1+1/R2)^-1;
Vo3=(1+(Rf/Rllel2))*Vo2;
printf(' Vo3 = %.3f Volts. \n ',Vo3);
printf('\n The total output voltage V0 is given as sum of Vo1 + Vo3.\n ');
Vout=Vo1+Vo3;
printf(' The output voltage = %.3f Volts. \n ',Vout);
printf('\n\n The equivalent circuit at various in between steps are shown in Fig. 4.4(b-c).');
```

Scilab code Exa 4.3 `To design an operational amplifier to differentiate an input s`

```
// Exa 4.3

clc;
clear;

// Given data

// An op-amp differentiator
fa = 100; // Hz
Vpp = 1; // Volts

// Solution

printf('Select , Fa = fmax = 100 Hz.\n');
```

```
printf(' Let, C1 = 0.1  F .\n');
C1 = 0.1*10^-6; // Farads
// since Fa = 1/(2*%pi*Rf*C1);
// Therefore,
Rf = 1/(2*%pi*fa*C1);
printf(' Therefore, the calculated value of Rf = %.1f k  . \n',Rf/1000);

printf(' Select, fb = 10*Fa = 1000 Hz.\n');
fb = 1000; // Hz
// Therefore
R1 = 1/(2*%pi*fb*C1);
printf(' The calculated value of R1 = %.2f k  . \n',R1/1000);
// Since, RfCf = R1C1
// Therefore we get,
Cf = R1*C1/Rf;
printf(' The calculated value of Cf = %.2f  F . \n',Cf*10^6);

printf('\n\n For a sinusoidal input - \n\n');
disp("since, vi = sin(2*%pi*100*t),  ");
disp("From Eq. 4.69, vo = -Rf*C1* d/dt(vi), ");
disp("Above equation yield following result once solved-  vo = -cos(2*%pi*100*t).");
printf('\n The input and output waveforms are shown in Graphic window 0 ans 1 respectively. \n\n');
// plotting wave forms

t = [0:%pi:13*%pi];
figure(0);

a=gca(); // Handle on axes entity
a.x_location = "origin";
a.y_location = "origin";
plot2d(t,sin(2*%pi*100*t));
title('Sine-wave-input',"color","Red","fontsize",3);
figure(1);
```

```

a=gca(); // Handle on axes entity
a.x_location = "origin";
a.y_location = "origin";
plot2d(t,-cos(2*%pi*100*t));
title('Cosine-wave-output',"color","blue","fontsize"
   ,3);

printf('\n For a square wave input -\n\n');
printf('\n For a square wave input, say 1 V peak and
    1 KHz,\n The output waveform will consist of
   positive and negative spikes of magnitude Vsat\n
   which is approximately 13 V for   15 V op-amp
   power supply.\n\n');
printf(' During the timeperiods for which input is
   constant at    1V, the differentiated output
   will be zero. \n However, when input transits
   between  1V  levels, \n the slope of the input is
    infinite for an ideal square wave. \n\n The
   output, therefore, gets clipped to about   13V
   for a   15 V op-amp power supply.');

printf('\n\n The output of a square wave input is a
   spike output as shown in Fig. 4.22(b). \n');
```

Scilab code Exa 4.4 To determine various parameters for a lossy integrator

```
// Exa 4.4

clc;
clear;

// Given data

// Refering integrator circuit in Fig. 4.23(c)
```

```
R1 = 10*1000; //
Rf = 100*10^3; //
Cf = 10*10^-9;  // Farads
// Solution
printf('For the given component values, the lower frequency limit of integration fa is ');
fa = 1/(2*%pi*Rf*Cf);
printf('%d Hz \n\n',fa);
printf(' For 99 percent accuracy, the input frequency should be at least one decade above fa. \n However, there is an limit up to which circuit will integrate and is determined by the frequency response of op-amp.\n However, as input frequency is increased, the output amplitude reduces as the gain of the integration falls\n at a rate of 6 dB/octave.\n\n' );
// case(1): Sine wave input
printf(' case(1) : For sine Wave Input \n');
printf('\n\n For a input of 1 V peak sine wave at 5 kHz, the integral of vi(t)=1*sin(2*pi*5000*t) is cosine function.\n');
t1 = 0:%pi:100*%pi;
disp("Input Function - vi = sin(2*%pi*5000*t);");
disp("Output Function - vo = 0.318*cos(2*%pi*5000*t);");
printf(' The input and output waveforms are depicted in Graphic window # 0\n');
vi = sin(2*%pi*5000*t1);  // Input
vo = 0.318*cos(2*%pi*5000*t1);  // Output
a=gca(); // Handle on axes entity
a.x_location = "origin";
a.y_location = "origin";
```

```
plot(t1,vi,'ro-');
plot(t1,vo,'o-b');
legend(["Input Function","Output Function"]);
xlabel("time ");
ylabel("Vi,Vo");
title("Sine wave plot");

// case(2): Step input
printf('\n\n case(2) : Step input\n');
printf('\n\n If input is a step voltage vi = 1V for
   0<t<=0.3msec, then the output voltage at t = 0.3
   ms is ');
vos = (-1/(R1*Cf))*integrate('1','x',0,0.3*10^-3);
printf('%d V \n',vos);
printf('\n The output voltage is a ramp function
   with a slope of 10V/ms and is shown in graphic
   window #1\n');
yi = [1,1,1,1,1,1,1,1,1,1,1];
t2 = 0:0.1:1;  // time in milli sec
yo = -10*t2;
figure(1);
a=gca(); // Handle on axes entity
a.x_location = "origin";
a.y_location = "origin";
plot(t2,yi,'ro-');
plot(t2,yo,'o-b');
legend(["Input Function","Output Function"]);
xlabel("Time in millisec");
ylabel("Vi,Vo");
title("Step plot");

// case(3): square wave input
printf('\n\n case(3): Square wave input \n');
printf('\n\n The output waveform for an input of 5
   kHz, 1 V peak square wave. \n It can be seen that
    input is of constant amplitude of 1 V from 0 to
   0.1 msec and -1 V from 0.1 ms to 0.2 ms.\n Thus,
   the expected output waveform will be a triangular
```

```
    wave.');
vosq = -(1/(R1*Cf))*integrate('1','x',0,0.1*10^-3);
printf('\n The peak value of the output for first
    half cycle is %.1f V \n',vosq);
printf('\n Both input and output waveforms are
    depicted in graphic window #3\n');
t3 = 0:0.1*10^-3:10^-3;
zi = [1,-1,1,-1,1,-1,1,-1,1,-1,1];
zo =
    [0.5,-0.5,0.5,-0.5,0.5,-0.5,0.5,-0.5,0.5,-0.5,0.5];

figure(2);
a=gca(); // Handle on axes entity
a.x_location = "origin";
a.y_location = "origin";
plot2d2(t3,zi,2);
plot2d(t3,zo,4);
legend(["Input Function","Output Function"]);
xlabel("Time in sec");
ylabel("Vi,Vo");
title("Square wave plot");
```

Scilab code Exa 4.5 To find R1 and Rf values in the lossy integrator

```
// Exa 4.5

clc;
clear;

// Given data

App = 20; // peak gain in dB
A = 17; // Actual gain in dB
w = 10000; // Angular Frequancy in rad/sec
C = 0.01*10^-6; // Farads
```

```

// Solution

printf('From Eq.(4.84) , we see  that gain is at its
    peak when w = 0.\n Therefore, 20*log(Rf/R1) =
   20.\n');

// Therefore,
// Rf = 10 R1;     .............Eq. (1)
z = 10;  // z is ratio of Rf/R1
printf(' i.e Rf/R1 = %d. \n',z );
printf(' At w = 10^4 rad/sec, gain in dB is down
   from its peak of 20 dB. \n Therefore, convering
   gain to dB in Eq.(4.84) and sub    situting for w
   ,C, and Rf/R1 we can get value of Rf.\n\n');
//  20*log10              10                = 17 dB
 //           ____________________________
 //             sqrt(1 + [10^4*10^-8*Rf]^2)

deff('y=f(x)','y = 20*log10( 10 / sqrt(1+[10^-4*x
   ]^2))-17');// x is Rf(  )
   [x,v,info] = fsolve(10,f);
   printf(' The calculated value of Rf is %d   .
      Rounding off to nearest possible value i.e 10
       k  . \n',x);
   Rf = 10000; //
   printf(' Since we have ratio of Rf  by R1 so, \n
       The value of R1 can be given as R1 = %d k  .
       \n',0.1*(Rf/1000)); //as R1/Rf = 0.1
```

Scilab code Exa 4.6 To calculate output of an operational amplifier to a step inpu

```
// Exa 4.6

clc;
```

```
clear;

// Given data

//Referring circuit in Fig. 4.26
// An op amp integrator and a low pass Rc circuit)

// Solution

printf(' Figure (4.26) is a simple op-amp integrator where Millers theorem is applied across the feedback capacitor Cf. \n The input time constant T = R1*Cf*(1-Av). \n Therefore, vi = V*(1-e^(-t/T));');
printf(' \n Therefore, vo = Av*Vi = Av* V*(1-e^(-t/R1*Cf*(1-Av))); ');
printf(' \n By expanding e^(-t/..) series by Taylors Expansion method we will reach to following approximation');
printf('\n vo     (-V*t/R1*Cf) * [1- t/(2*R1*Cf*(1-Av))];    if Av>>1 ...eq (1)   ');
printf('\n\n');
printf('  Also, we know that for a low pass RC integrating circuit network(without op-amp) the output vo for a step input of V becomes \n');
printf('  For a large Rc, vo      (V*t)/R*C) * (1 - t/(2*R*C) .. eq(2)');  //Eq(2)
printf('\n\n');
printf(' It can be seen that the output voltages of both circuits varies aproximately linearly with time(for large RC) and \n for either case, derivative(vo) = V/RC. \n However, the second term in both the expression represent deviation from the linearity. \n we see that op-amp integrator is more linear than the simple RC circuit by a factor of  1/(1-Av).\n');
```

Scilab code Exa 4.7 To show that output is given by a differential equation

```
// Exa 4.7

clc;
clear;

// Given data

// Referring Circuit in Fig. 4.27

// Solution

printf(' The transfer gain of the cirucuit is - \n')
    ;
printf(' Vo(s) = -Zf =  (R2+R3)+s*C*R2*R3 \n');
printf(' ----      ----    ------------------\n');
printf(' Vi(s) =    R1 =      R1*(1+s*C*R3)\n');

printf('\n i.e R1(1+s*C*R3)*Vo(s)+[(R2+R3)+s*C*R2*R3
    ]*Vi(s) = 0.\n');
printf('\n\n Writing above equation  in time domain
    ( s d /dt), we get,\n');
printf('\n R1 + C*R3*R1(d/dt Vo(t))+ [(R2+R3)+c*R2*
    R3]*(d/dt Vi(t)) = 0   ...eq(1)\n\n');

printf(' Since, vi(t) = V, \n Therefore, d/dt Vi(t)
    = 0.\n\n');
printf(' Therefore eq(1) becomes-  \n C*(d/dt vo) +
    vo/R3 + V/R1 + (R2/R1*R3)*V = 0.\n');
printf(' \n Thus, output vo(t) is given by a
    differential equation as shown above. \n');
```

Scilab code Exa 4.8 To show that output is integral of input

```
// Exa 4.8

clc;
clear;

// Given data

// Referring Fig. (4.28) -Non- inverting terminal integrator

// Solution

printf(' The voltage at the (+) input terminal of the op-amp due to potential divider is,\n');
printf(' V(+) =    1/ s*C     *   Vi(s)\n');
printf('          -----------\n');
printf('            R+ 1/ s*C \n\n');
printf(' The output voltage Vo(s) fot the non-inverting amplifier is - \n');
printf(' Vo(s) = (1 + 1/(s*C*R))*V(+) = Vi(s) / (s*R*C)).\n\n');
printf(' Hence in time domain, we get, vo = (1/(R*C))       vi dt .\n');
printf(' Hence proved. \n');
```

Scilab code Exa 4.9 To set up an analog simulation to generate a sinusoidal signal

```
// Exa 4.9

clc;
```

```
clear;

// Given data

// To generate a sinusoidal signal 10 sin 3t.

// Solution

printf('Let us first obtain a differential equation
   whose solution is 10 sin 3t.\n');
printf(' Let x(t) = 10 sin 3t −−−−−−eq(1)\n');
printf(' The first derivative of this i.e. dx(t) =
   30 cos 3t −−−−eq(2)\n');
printf(' The second derivative of this i.e. d2x(t) =
    −90 sin 3t = −9*x(t) \n');
printf('\n Therefore, required differential equation
    is d2x(t)+9*x(t)=0. \n\n');

printf(' The initial condition is obtained by
   putting t=0 in eq(1&2), \n x(0)=0 and dx(0) = 30.
    \n' );
printf(' Assuming that d2x(t) is available, x(t) can
    be obtained by integrating x twice.\n The
   complete setup is shown in Fig. 4.31−Simulation
   of 10 sin 3t.\n');
```

Chapter 5

Comparators and Waveform Generators

Scilab code Exa 5.1 To plot the transfer curve for opamp with given conditions

```
// Exa 5.1

clc;
clear;

// Given data

// A comparator as shown in FIg. 5.7(a)
Aol=50000; // open loop gain of op-amp
Vz=9; // Volts
Vd=0.7; // cutoff voltage

// Solution

// case 1
printf(' Since AOL =      , even a small positive or
   negative voltage at the input drives the output
   to +- Vsat. \n  This causes Vz1 or Vz2 to break
   down, giving output voltage vo = +-(Vz+Vd)= ');
```

```
Vsat = Vz+Vd;
printf(' %.1f V. \n  The same is shown in Graphic Window No. 0 \n', Vsat);
Vi= [-1:0.1:1];
for i=1:21
    if(Vi(i)<0)
         Vo(i)=-Vsat;
       elseif(Vi(i)==0)
            Vo(i)=Vsat;
        else
            Vo(i)=Vsat;

    end
    end
set(gca(),"grid",[1,1]);
a=gca(); // Handle on axes entity
a.x_location = "origin";
a.y_location = "origin";
plot2d2(Vi,Vo);
title('Transfer curve for ideal op-amp condition',"color","blue","fontsize",3);


// case 2

DellVi = Vsat/Aol; // Zener breaks down after +-Dell_Vi
scf(1);
Vi= [-1:0.1:1];
for i=1:21
    if(Vi(i)<0)
         Vo(i)=-Vsat;
       elseif(Vi(i)==0)
            Vo(i)=DellVi;
        else
            Vo(i)=Vsat;

    end
```

```
end
set(gca(),"grid",[1,1]);
a=gca(); // Handle on axes entity
a.x_location = "origin";
a.y_location = "origin";
plot(Vi,Vo,'ro-');
title('Transfer curve for practical op-amp condition
    ',"color","blue","fontsize",3);

printf(' \n\n  Now since,   Vi   = %.3f mV. The
    zeners break down after +- %.3f mV \n  as shown
    in the transfer curve depicted in Graphic Windows
     No. 1',DellVi*1000,DellVi*1000);
```

Scilab code Exa 5.2 To determine the upper and lower threshold voltages

```
// Exa 5.2

clc;
clear;

// Given data

// Circuit of Schmitt trigger

R2=100; // Ohms
R1=50*10^3; // Ohms
Vref=0; //Volts
Vi=1; // peak to peak(Volts)
Vsat=14;// sSaturation voltage (Volts)

// Solution

printf('Using Equations (5.1) and (5.2), we get
    calculated values as follows -\n ');
```

```
Vut=(R2*Vsat)/(R1+R2);
printf(' Upper threshold voltage (VUT) = %d mV. \n ',round(Vut*1000));
Vlt=(R2*-Vsat)/(R1+R2);
printf(' Lower threshold voltage (VLT) = %d mV. \n ',round(Vlt*1000));
```

Scilab code Exa 5.3 To calculate time period of the negative and positive portion

```
// Exa 5.3

clc;
clear;

// Given data

// A Schmitt trigger as shown in fig. 5.9- circuit for example 5.3

VUT=0; // Upper threshold(V)
VH=0.2; // Hysteresis width(V)
F=1000; // Hz
Vpp=4; // peak to peak voltage(V)

// Solution

// Since VH=VUT-VLT
VLT=VUT-VH;

// From fig. 5.9, the angle   can be calculated as
// VLT = (Vpp/2)* sin(%pi+   );
// Rearranging above equation
Theta = asin(VLT/-(Vpp/2)); // in radians
T= 1/F; // Time period(sec)

```

```
// wT  = 2*%pi*F*T  = 0.1
// Rearranging
Ttheta= Theta/(2*%pi*F);

T1=T/2 + Ttheta;
T2=T/2 - Ttheta;

printf('The duration of positive pulse(T1) = %.3f
   msec and duration of negative pulse(T2) = %.3f
   msec. \n ',T1*1000,T2*1000);
```

Scilab code Exa 5.4 To design a phase shift oscillator to oscillate at 100 Hz

```
// Exa 5.4

clc;
clear;

// Given data

// Phase shift oscillator as given in Fig. 5.15

F=100; // Oscillation frequency(Hz)


// Solution

printf(' Let C=0.1 microFarads, then from Eq. (5.25)
    we can get value of R. \n ');
R = 1/(sqrt(6)*2*%pi*10^-7*F);

printf(' The value of R as calculated = %.2f k . \n
    ',R/1000);

printf(' To prevent overloading of the amplifier by
```

```
    RC network, R1 <= 10*R. \n');
R1=10*R;
printf('  Therefore R1 = %d k  . \n ',round(R1/1000)
    );
// Since Rf = 29*R1;
Rf= 29*round(R1/1000);  // k
printf(' Since Rf = 29*R1, therefore value of Rf =
    %d k  . \n ',Rf);
```

Chapter 6

Voltage Regulator

Scilab code Exa 6.1 To calculate the output current coming from 7805

```
// Exa 6.1

clc;
clear;

// Given data

// IC 7805 is specified
Veb_on=1; // Volts
B=15; // Current gain
R1=100; // Load 1(  )
R2=5; // Load 2(  )
R3=1; // Load 3(  )

// Solution

// Case(1)
printf(' Load = 100   \n\n');
printf('For IC 7805, the output voltage across the load will be 5 V.\n ');
V1=5; // Voltage across load
```

```
IL1=V1/R1;
VR1= 7 * IL1; // Voltage across R1
printf('The output current coming from 7805 = IL1 = Io = Ii = %d mA. \n ',IL1*1000);
printf('The voltage across R1 = %.2f V which is less than 0.7 V. Hence Q1 is off. \n ',VR1);
printf('So Ic1 = 0.');
printf('\n\n');



// Case(2)
printf('  Load = 5     \n');
printf('\n For IC 7805, the output voltage across the load will be 5 V.\n ');
V2=5; // Voltage across load
IL2=V2/R2;
VR2= 7 * IL2; // Voltage across R2
printf('The output current coming from 7805 = IL2 = Io = Ii = %d A. \n ',IL2);
printf('Assume that the entire current comes through regulator and that Q1 is OFF. Now the voltage drop across R1 is equal to %d V.\n Thus,our assumption is wrong and Q1 is ON.\n ',VR2);

// From equation 6.10- Il2 = 1A = (B+1)*Io-B*Veb_on/R2;
// Therefore
Io2 = (IL2+(B*Veb_on)/7)/(B+1);
// From equation 6.6- IL2 = 1A = Ic2+Io2;
// Therefore
Ic2= IL2-Io2;
printf('Using equations 6.6 and 6.10 we got values as Io2 = %d mA and Ic2 = %d mA. \n ',Io2*1000,Ic2*1000);
printf('\n\n');


```

```
// Case(3)
printf('  Load = 1     \n');
printf('\n For IC 7805, the output voltage across the load will be 5 V.\n ');
V3=5; // Voltage across load
IL3=V2/R3;
VR3= 7 * IL3; // Voltage across R3
printf('The output current coming from 7805 = IL3 = Io = Ii = %d A. \n ',IL3);
printf('Assume that the entire current comes through  regulator and that Q1 is OFF. Now the voltage drop across R1 is equal to %d V.\n Thus,our assumption is wrong and Q1 is ON.\n ',VR3);

// From equation 6.10- IL3 = 5A = (B+1)*Io-B*Veb_on/R3;
// Therefore
Io3 = (IL3+(B*Veb_on)/7)/(B+1);
// From equation 6.6- IL3 = 5A = Ic3+Io3;
// Therefore
Ic3= IL3-Io3;
printf('Using equations 6.6 and 6.10 we got values as Io3 = %d mA and Ic3 = %.3f Amp. \n ',Io3*1000,Ic3);
```

Scilab code Exa 6.2 `To calculate component values to get desired output voltage`

```
// Exa 6.2

clc;
clear;

// Given data

```

```
/// Referring Fig. 6.5- Adjustable regulator
Vo= 7.5; // Volts

// Solution

printf(' From the data sheeet of 7805, IQ=4.2 mA. Say, we choose IR1 = 25 mA.\n ');
IQ = 0.0042; // Amperes
IR1 = 0.025; //Amperes
printf(' The voltage across load for 7805 is 5 Volts.\n ');
VR=5; // Volts
R1 = VR/IR1;
printf(' Thus, calculated value of R1 = %d  . \n ',R1);

printf(' We have to choose R2 as to develop a voltage of 2.5 V across it. So, R2 comes out to be,\n ');
R2= 2.5/(IR1+IQ);
printf(' The value of R2 = %d  . \n ',int(R2));
```

Chapter 7

Active Filters

Scilab code Exa 7.1 Design a second order Butterworth low pass filter

```
// Exa 7.1

clc;
clear;

// Given data

n=2;  // Second order Butterworth filter
fL=1000; // Higher cut off frequency(Hz)

// Solution

printf('Let C = 0.1  F . \n');
C=0.1*10^-6; // Farads

//     Since fL = 1/(2 * %pi * R*C);
//     Therefore;
R = 1/(2*%pi*fL*C);
printf(' The calculated value of R = %.1f k  .  \n',
   R/1000);

```

```
printf(' From Table 7.1, for n=2, the damping factor alpha = 1.414.');
alpha=1.414;
A0 = 3-alpha;
printf('\n Then the pass band gain A0 = %.3f. \n',A0);
printf('\n');
printf(' The transfer function of the normalized second order Butterworth filter is 1.586 ');
printf('\n ________________________');
printf('\n Sn^2+1.414*Sn+1');

// Since Af= 1 + Rf/Ri = 1 + 0.586;
printf('\n Since A0= 1.586 so Let Rf = 5.86 k and Ri = 10 k to make A0 = 1.586.' );

printf(' \n The circiuit realized is as shown in Fig. 7.4 with component value as mentioned above.');

printf('\n\n\n Frequency, f in Hz Gain magnitude in dB 20 log(vo/vi)\n');
// Frequency Response
x=[0.1*fL,0.2*fL,0.5*fL,1*fL,5*fL,10*fL]
for i = 1:1:6
    response(i) = 20*log10(A0/(sqrt(1+(fL/x(i))^4)));
    printf(' %d %.2f \n',x(i),response(i));
end
```

Scilab code Exa 7.2 Design a fourth order Butterworth low pass filter

```
// Exa 7.2

clc;
clear;

// Given data

n=4; // Fourth order Butterworth low-pass filter
fH=1000; // Hz

// Solution

printf('Let C = 0.1  F . \n');
C=0.1*10^-6; // Farads
//     Since fH = 1/(2 * %pi * R*C);
//     Therefore;
R = 1/(2*%pi*fH*C);
printf(' The calculated value of R = %.1f k  . \n',R/1000);

printf(' From Table 7.1, for n=4, we get two damping factors namely,\n alpha1 = 0.765 and alpha2 = 1.848.');
alpha1=0.765;
alpha2=1.848;
A01 = 3-alpha1;
A02 = 3-alpha2;
printf('\n');
printf('\n Then the pass band gain A01 = %.3f and A02 = %.3f. \n',A01,A02);
printf('\n');
printf(' The transfer function of the normalized
```

```
    second order low-pass Butterworth filter is
    2.235                    1.152          ');
printf('\n

    -----------------  *  -------------------');
printf('\n

    Sn^2+0.765*Sn+1         Sn^2+1.848*Sn+1 ');

// Since A01= 1 + Rf/Ri = 1 + 1.235;
printf('\n  Since A01= 2.235 so Let Rf1 = 12.35 k
    and Ri1 = 10 k   to make A01 = 2.235.' );
printf('\n  Since A02= 1.152 so Let Rf2 = 15.20 k
    and Ri1 = 100 k   to make A01 = 1.152.' );

printf(' \n The circiuit  realized is as shown in
    Fig. 7.7 with component value as mentioned above.
    ');
```

Scilab code Exa 7.3 `To determine order of a low pass Butterworth filter`

```
// Exa 7.3

clc;
clear;

// Given data

Attn=40; // Attenuation in dB
x=2; // x= ratio of W to Wh

// Solution

printf(' Using equation 7.26,\n');

```

```
// 20*log(H(jw)/A0)=-40; // -ve since it is attenuation
// gives
// H(jw)/A0 = 10^-2 = 0.01
// so
// (0.01)^2 = 1/(1+2^(2*n));
// or 2^2n = 10^4 - 1;
// solving for n, we get

n=log(10^(4) -1)/(2*log(2));
printf(' The calculated value of n = %.2f. \n',n);
printf(' Since order of filter must be an integer so, n = %d. \n',round(n));
```

Scilab code Exa 7.4 Design a second order Butterworth high pass filter

```
// Exa 7.4

clc;
clear;

// Given data

n=2; // Second order Butterworth filter
fH=1000; // Lower cut off frequency(Hz)

// Solution

printf('Let C = 0.1 F . \n');
C=0.1*10^-6; // Farads

// Since fH = 1/(2 * %pi * R*C);
// Therefore;
R = 1/(2*%pi*fH*C);
printf(' The calculated value of R = %.1f k . \n',
```

```
    R/1000);

printf(' From Table 7.1, for n=2, the damping factor
     alpha = 1.414.');
alpha=1.414;
A0 = 3-alpha;
printf('\n Then the pass band gain A0 = %.3f. \n',A0
    );
printf('\n');
printf(' The transfer function of the normalized
    second order low-pass Butterworth filter is
         1.586        ');
printf('\n

    ——————————————');
printf('\n

    Sn^2+1.414*Sn+1');

// Since Af= 1 + Rf/Ri = 1 + 0.586;
printf('\n Since A0= 1.586 so Let Rf = 5.86 k    and
     Ri = 10 k   to make A0 = 1.586.' );

printf(' \n The circiuit  realized is as shown in
    Fig. 7.4 with component value as mentioned above.
    ');

printf('\n By considering minimum DC offset
    condition, the modified value of R and C comes
    out to be R = 1.85 k    and C=0.086  F .');
printf('\n\n\n Frequency, f in Hz           Gain
    magnitude in dB 20 log(vo/vi)\n');
// Frequency Response
x=[0.1*fH,0.2*fH,0.5*fH,1*fH,5*fH,10*fH]
for i = 1:1:6
     response(i) = 20*log10(A0/(sqrt(1+(x(i)/fH)^4)
        ));
     printf(' %d                                  %.2
```

```
        f         \n',x(i),response(i));
end
```

Scilab code Exa 7.5 Design a wide band pass filter

```
// Exa 7.5

clc;
clear;

// Given data

// A wide-band pass filter
fL=400; // Lower cutoff frequency(Hz)
fH=2000; // Higher cutoff frequency(Hz)
A0=4; // passband gain

// Solution

printf('Since, the pass band gain is 4. so each of
   LPF and HPF section may be designed to give gain
   of 2,\n that is Ao=1+ (Rf/Ri) = 2.\n So, Rf and
   Ri should be equal. \n Let Rf=Ri=10 k   for each
   of LPF and HPF sections.');

disp("");
disp("");
disp("For HPF, fL=400 Hz.");
printf(' Assume C2=0.01   F . ');
C2=0.01*10^-6; // Farads
// Since fL= 1/(2*%pi*R2*C2);
// Therefore
R2= 1/(2*%pi*C2*fL);
printf(' \n The calculated value of R = %.1f  k  .',
   int(R2)/1000);
```

```
disp("");
disp("");
disp("For LPF, fH=2000 Hz.");
printf(' Assume C1=0.01  F .');
C1=0.01*10^-6; // Farads
// Since fH= 1/(2*%pi*R1*C1);
// Therefore
R1= 1/(2*%pi*C1*fH);
printf(' \n The calculated value of R = %.2f k  .',R1/1000);

disp("");
disp("");

fo=sqrt(fL*fH);
Q=fo/(fH-fL);

printf(' The value of cutoff frequency = %.1f Hz.\n',fo);
printf('\n The quality factor = %.2f (<10) since wide passband filter.',Q);
```

Scilab code Exa 7.6 Design a 50 Hz active notch filter

```
// Exa 7.6

clc;
clear;

// Given data

// A notch filter
fo=50; // cutoff frequency for notch filter(Hz)

```

```
//Solution

printf('As Given fo=50 Hz. Let C=0.1  F .');
C=0.1*10^-6; // Farads
// since fo=1/(2*%pi*R*C);
// Therefore R -
R=1/(2*%pi*fo*C);
printf(' \n For R/2, take two resistors of 31.8 k
   Ohms in parallel and for 2C,\n take two 0.1
   mocroFarads capacitors in parallel to make the
   twin-T notch filter\n as shown in Fig. 7.15(a) on
    page no. 279 where resistors R1 and R2 are for
   adjustment of gain.\n ')
```

Scilab code Exa 7.7 Design a wide band reject filter

```
// Exa 7.7

clc;
clear;

// Given data
fH= 400; // Higher cutoff frequency(Hz)
fL=2000; // lower cutoff frequency(Hz)
Ao=2;   // Pass band gain

// Solution

disp("For HPF, fL=2 kHz.");
disp("Assume C2=0.1  F . ");
C2=0.1*10^-6; // Farads
// Since fL= 1/(2*%pi*R*C2);
// Therefore
RL= 1/(2*%pi*C2*fL);
printf(' The calculated value of R = %d  .',int(RL)
```

```
   );
printf('\n Let R = 800   .');
// Since Ao=Ao2 = 1+ (Rf/Ri);
disp("Let Rf = Ri =10 k  (say) to give A02 of 2.");
disp("");
disp("");
disp("For LPF, fL=400 Hz.");
disp("Assume C1=0.1  F . ");
C1=0.1*10^-6; // Farads
// Since fH= 1/(2*%pi*R*C1);
// Therefore
RF= 1/(2*%pi*C1*fH);
printf(' The calculated value of R = %d   .',int(RF)
   );
printf('\n Let R = 4 k  .');..
// Since Ao=Ao1 = 1+ (Rf/Ri);
disp("Let Rf = Ri =10 k  (say) to give A01 of 2.");

disp("");
disp("");

disp("The schematic arrangement and the frequency
   response is shown in figs. 7.16(a,b) on page no.
   280.")
```

Scilab code Exa 7.9 Design a switched capacitor integrator

```
// Exa 7.9

clc;
clear;

// Given data

fo=10; // Hz
```

```
// Solution

disp(" For a switched capacitor integrator, assume fCK=1000 Hz.");
fCK=1000; // Hz
 disp(" From Eq. (7.129) on page no. 293, we get, ");
 disp(" Cf/C1 =x= fCK/(2*%pi*fo). "); // x = ratio of Cf by C1
x=fCK/(2*%pi*fo);
disp(" Lets choose cF=15.9 pF.");
cF=15.9*10^-12; // Farads
C1=cF/x;
printf('  By calculation C1 = %d pF.\n ',round(C1*10^12));
disp(" For RC integrator, select R1=1.6*10^6   .") ;
R1=1.6*10^6; //
cF1=1/(2*%pi*R1*fo);
printf('  By calculation cF = %d nF. \n',round(cF1*10^9));
disp("");
printf('  The values of R1 = 1.6 mHz and cF = 10nF are not quite practical for a monolothic circuit.\n      From this, it is obvious that switched capacitor circuits are more practical so far as IC fabrication is concerned.\n  So it can be seen that an SC integrator requires very low values of capacitance compared to lossy integrator.');
disp("");
printf('  If a resistor R2 is placed in parallel with the feedback capacitor cF of Fig. 7.26(a), a lossy or practical integrator is obtained. \n The transfer function for this circuit is given in Eq. (7.130) and (7.131) on page no. 294.' );

printf('\n \n  The switched capacitor implementation of Fig. 7.26(a) is shown in Fig. 7.26(b)\n
```

```
where resistors R1 and R2 have been replaced by
switched capacitors C1 and C2 and its MOS version
 is in Fig. 7.26(c).');
```

Chapter 8

555 Timer

Scilab code Exa 8.1 To calculate value of Capacitor

```
// Exa 8.1

clc;
clear;

// Given data

// Monostable multivibrator
R=100*10^3; //
T=100*10^-3; // Time delay (sec)

// Solution
printf(' Using Eqn.(8.2) on page no.313, we get,');
//T= 1.1*R*C;
C=T/(1.11*R);
printf(' C = %.1f  F .\n  From the graph of Fig.8.6
   on page no. 314, the value of C is found to be
   0.9  F  also.\n',C*10^6);
```

Scilab code Exa 8.2 To calculate various parameters of Astable multivibrator

```
// Exa 8.2

clc;
clear;

// Given data

// Astable multivibrator
Ra=6.8*10^3; //
Rb=3.3*10^3;  //
C=0.1*10^-6; //  F

// Solution

disp("By using Eq. (8.11) on page no. 320 we get,
   tHigh as");

tHigh=0.69*(Ra+Rb)*C; // Time required to charge
   from 1/3 Vcc to 2/3 Vcc
printf(' tHIGH = %.1f mSec. \n',tHigh*1000);
disp("By using Eq. (8.12) on page no. 320 we get,
   tLow as");

tLow=0.69*(Rb)*C;  // TIme required to discharge
   from 2/3 Vcc to 1/3 Vcc
printf(' tLOw = %.2f mSec. \n',tLow*1000);

disp("By using Eq. (8.13) on page no. 320 we get,
   free running frequency as");
f= 1.45/((Ra+2*Rb)*C);
printf(' f = %.2f kHz. \n\n',f/1000);

D= Rb/(Ra+2*Rb);
printf(' The duty cycle D = %.2f (%d percent).\n ' ,
   D,round(D*100));
```

Chapter 10

D to A and A to D Converters

Scilab code Exa 10.1 To calculate output voltage for given 9 bit input

```
// Exa 10.1

clc;
clear;

// Given data

//9-bit DAC
step = 10.3; // mV
y=[1 0 1 1 0 1 1 1 1];
n = 9; // since 9 bit DAC

// Solution

i = n;
add = 0;
while(i>0)
    op = step*2^(i-1)*y((n+1)-i);
    i = i-1;
    add = add + op
    end
```

```
printf('The output voltage for input 101101111 is %
    .2f V. \n',add*10^-3);
```

Scilab code Exa 10.2 To calculate LSB MSB and full scale output for an 8 bit DAC

```
// Exa 10.2

clc;
clear;

// Given data

// 8 bit DAC
n = 8;
Vmin = 0; // Volts
Vmax = 10; // Volts

// Solution

printf(' For a 8-bit DAC :-\n\n');
LSB = 1/2^n;
LSB10 = LSB*Vmax;
printf('  LSB = %.3f V (i.e. 1/256).\n',LSB10);
MSB10 = (1/2)*Vmax;
printf('  MSB = %d V.\n',MSB10);
fso = (Vmax-LSB10);
printf('  Full scale output = %.3f V. \n',fso);
```

Scilab code Exa 10.3 To calculate output voltage for different input

```
// Exa 10.3

clc;
```

```
clear;

// Given data

// D/A converter is used
Vmin = 0; // Voltage
Vmax = 10; // Voltage
ip1 = [1 0]; // for a 2-bit D/A converter(input 1)
ip2 = [0 1 1 0]; // for a 4-bit DAC(input 2)
ip3 = [1 0 1 1 1 1 0 0 ]; // foe a 8-bit DAC(input 3)

// Solution
V01 = Vmax*(ip1(1)*2^(-1)+ip1(2)*2^(-2));  // output 1
printf(' The output voltage for input1 = [10] is %d V. \n ',V01);
V02 = Vmax*(ip2(1)*2^(-1)+ip2(2)*2^(-2)+ip2(3)*2^(-3)+ip2(4)*2^(-4)); // Output 2
printf(' The output voltage for input2 = [0110] is %.2f V. \n ',V02);
V03 = Vmax*(ip3(1)*2^(-1)+ip3(2)*2^(-2)+ip3(3)*2^(-3)+ip3(4)*2^(-4)+ip3(5)*2^(-5)+ip3(6)*2^(-6)+ip3(7)*2^(-7)+ip3(8)*2^(-8)); // Output 3
printf(' The output voltage for input3 = [10111100] is %.2f V. \n',V03);
```

Scilab code Exa 10.4 To calculate value of the resistor R of the integrator

```
// Exa 10.4

clc;
clear;

// Given data
```

```
// A 16 bit dual slope ADC is specified
n = 16; // 16 bit counter
CR = 4*10^6; // clock rate in Hz
Vimax = 10; // Maximum input  voltage
Vomax= -8; // Maximum integrator output voltage
C = 0.1*10^-6; // Capacitor(Farads)

// Solution

//Referring Eqn 10.4, 10.5, 10.6, 10.7 given on page
    no 364 and 365;

T1 = 2^n/CR; // Time Period
// For the integrator
// dell Vo= (-1/RC)*Vmax*T1;
// Therefore
R = -(Vimax*T1)/(Vomax*C); // Resistor value
printf('The value of resistor R of the integrator is
    %d k  . \n ',round(R/1000));
```

Scilab code Exa 10.5 To find the equivalent digital number

```
// Exa 10.5

clc;
clear;

// Given data

// A 16bit dual slope ADC is specified
Va = 4.129; // Input analog Voltage
Vr= 8; // Maximum integrator output voltage(
   Reference Voltage)
n=16; // 16 bit counter
```

```

// Solution

disp("Referring to Eqn 10.7 on page no. 365 we get,"
    )
// Va = Vr*(N/2^n);
N = round(Va * 2^n / Vr); // Digital count
printf(' The digital count N = %d for which the
    binary equivalent = \n',N);

// code to convert decimal to binary weuivalent
Nbin = [0000000000000000];
while (N > 0 & n > 0)
    if (modulo(N,2)== 0)
        Nbin(n)=0;
    else
        Nbin(n)=1;
end
n=n-1;
N=int(N/2);
end
disp((Nbin)');
```