

Scilab Textbook Companion for
Digital Electronics: Circuits And Systems
by V. K. Puri¹

Created by
Rishav Dutta
B.Tech
Electronics Engineering
NIT Durgapur
College Teacher
Prof. Sabyasachi Sengupta,iit Kharagpur
Cross-Checked by
Mukul R. Kulkarni

July 31, 2019

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

Book Description

Title: Digital Electronics: Circuits And Systems

Author: V. K. Puri

Publisher: Tata McGraw Hill, New Delhi

Edition: 1

Year: 1997

ISBN: 0-07-463317-1

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
1 Number systems and binary codes	5
2 Boolean algebra logic gates karnaugh map and minimization	94
3 Wave shaping clipping clamping and sweep circuits	107
5 Sequential circuits flip flops and multivibrators	112
8 D to A and A to D converters	115

List of Scilab Codes

Exa 1.1.a	decimal to binary conversion	5
Exa 1.1.b	binary to decimal conversion	5
Exa 1.2	binary to decimal conversion	6
Exa 1.3	binary to decimal conversion	8
Exa 1.4	decimal to binary conversion	9
Exa 1.5	decimal to octal conversion	10
Exa 1.6	octal to decimal conversion	11
Exa 1.7	binary to octal conversion	11
Exa 1.8.a	decimal to binary conversion	12
Exa 1.8.b	binary to decimal conversion	12
Exa 1.9.a	decimal to binary conversion	13
Exa 1.9.b	decimal to binary conversion	14
Exa 1.9.c	decimal to binary conversion	15
Exa 1.9.d	binary to decimal conversion	15
Exa 1.10.a	hex to decimal conversion	16
Exa 1.10.b	decimal to hex conversion	16
Exa 1.11	hex to binary and decimal conversion	17
Exa 1.12	hex to binary and octal and decimal conversion	17
Exa 1.13.a	hex to decimal conversion	18
Exa 1.13.b	hex to decimal conversion	18
Exa 1.14.a	addition of two binary numbers	19
Exa 1.14.b	addition of two binary numbers	19
Exa 1.14.c	addition of two binary numbers	23
Exa 1.15.a	subtraction of two binary numbers	27
Exa 1.15.b	subtraction of two binary numbers	31
Exa 1.16.a	multiplication of two binary numbers	35
Exa 1.16.b	multiplication of two binary numbers	39
Exa 1.17.a	division of two binary numbers	42

Exa 1.17.b	division of two binary numbers	43
Exa 1.18.a	binary to hex conversion	47
Exa 1.18.b	decimal to octal conversion	48
Exa 1.18.c	binary to decimal conversion	48
Exa 1.19.a	decimal to octal conversion	50
Exa 1.19.b	decimal to octal conversion	51
Exa 1.19.c	decimal to octal conversion	52
Exa 1.20.a	decimal to signed binary number conversion	53
Exa 1.20.b	decimal to signed binary system conversion .	54
Exa 1.20.c	decimal to signed binary system conversion .	54
Exa 1.21.a	subtraction using ones complement	55
Exa 1.21.b	subtraction using ones complement	55
Exa 1.21.c	subtraction using ones complement	56
Exa 1.21.d	subtraction using ones complement	56
Exa 1.21.e	subtraction using ones complement	59
Exa 1.22.a	subtraction using twos complement	62
Exa 1.22.b	subtraction using twos complement	63
Exa 1.22.c	subtraction using twos complement	64
Exa 1.22.d	subtraction using twos complement	66
Exa 1.22.e	subtraction using twos complement	69
Exa 1.22.f	subtraction using twos complement	72
Exa 1.23.a	BCD to decimal equivalent conversion	75
Exa 1.23.b	BCD to decimal equivalent conversion	76
Exa 1.24.a	binary to gray code conversion	78
Exa 1.24.b	binary to gray code conversion	79
Exa 1.24.c	binary to gray code conversion	80
Exa 1.25.a	subtraction using ones complement	81
Exa 1.25.b	gray code to binary conversion	81
Exa 1.25.c	decimal to BCD conversion	82
Exa 1.25.d	subtraction using ones complement	83
Exa 1.26	finding equivalent in decimal for excess 3 code	84
Exa 1.27.a	gray code to binary conversion	85
Exa 1.27.b	gray code to binary conversion	86
Exa 1.27.c	gray code to binary conversion	87
Exa 1.28.a	constructing an even parity 7 bit hamming code	88
Exa 1.28.b	constructing an even parity 7 bit hamming code	89

Exa 1.29	locating error in hamming code and correcting it	91
Exa 2.8.a	simplification of boolean expression	94
Exa 2.8.b	simplification of boolean expression	95
Exa 2.9.a	simplification of boolean expression	95
Exa 2.9.b	simplification of boolean expression	96
Exa 2.9.c	simplification of boolean expression	96
Exa 2.10.a	simplification of boolean expression	97
Exa 2.10.b	simplification of boolean expression	97
Exa 2.10.c	simplification of boolean expression	98
Exa 2.11	simplification of boolean expression	98
Exa 2.13.a	simplification of boolean expression	99
Exa 2.13.b	simplification of boolean expression	99
Exa 2.29	finding SOP	100
Exa 2.30	finding POS	100
Exa 2.31.a	conversion to canonical SOP	101
Exa 2.31.b	conversion to canonical POS	102
Exa 2.32	finding SOP and POS	102
Exa 2.33	finding SOP and POS	103
Exa 2.34	finding maxterms and minterms	104
Exa 2.35	finding maxterms	104
Exa 2.36	finding required data	105
Exa 2.37	finding SOP	105
Exa 2.38	finding sum of minterms and product of maxterms	106
Exa 3.1	finding required data	107
Exa 3.2	finding maximum amplification	108
Exa 3.3.a	finding clamping level voltage	108
Exa 3.3.b	finding clamping level voltage	109
Exa 3.4	finding output voltage	109
Exa 3.6	finding resistance and capacitance	110
Exa 3.7	finding peak voltage	111
Exa 5.1	finding required time	112
Exa 5.2	finding frequency and duty cycle	112
Exa 5.3	finding frequency and duty cycle	113
Exa 8.1	finding binary equivalent weights	115
Exa 8.2	calculating required data	116
Exa 8.3	calculating required data	117

Exa 8.4	finding required data	117
Exa 8.5	calculating required data	118
Exa 8.6	calculating required data	119
Exa 8.7	calculating value of resistors	119
Exa 8.8	finding value of resistor	120
Exa 8.9	finding required data	120
Exa 8.10	finding required data	121
Exa 8.11	calculating output voltages	122
Exa 8.12	finding output voltages	122
Exa 8.13	calculating maximum scale voltage	123
Exa 8.14	finding required data	123
Exa 8.15	calculating number of bits	124
Exa 8.16	finding resolution	125
Exa 8.17	calculating required data	125
Exa 8.18	calculating conversion time	126
Exa 8.19	finding resolution	127
Exa 8.20	finding number of bits	127
Exa 8.21	calculating maximum conversion time	128
Exa 8.22	finding error in voltage	128
Exa 8.23	finding aperture time	129
Exa 8.24	finding quantisation error	129
Exa 8.25	finding digital output in binary form	130

Chapter 1

Number systems and binary codes

Scilab code Exa 1.1.a decimal to binary conversion

```
1 //decimal to binary conversion//
2 //example 1.a//
3 clc
4 //clears the command window//
5 clear
6 //clears//
7 //decimal to binary conversion//
8 a=dec2bin(1993)
9 disp('binary equivalent of decimal number:')
10 disp(a)
11 //answer in binary form//
```

Scilab code Exa 1.1.b binary to decimal conversion

```
1 //binary to decimal conversion//
2 //example 1.b//
```

```

3  clc
4  //clears the command window//
5  clear
6  //clears//
7  //binary to decimal conversion//
8  a=bin2dec('11111001001')
9  disp('decimal equivalent of binary number:')
10 disp(a)
11 //answer in decimal form//

```

Scilab code Exa 1.2 binary to decimal conversion

```

1  //binary to decimal conversion//
2  //example 2//
3  clc
4  //clears the command window//
5  clear
6  // clears //
7  p =1;
8  // initialising //
9  q =1;
10 z =0;
11 b =0;
12 w =0;
13 f =0;
14 //bin= input ( Enter the binary no to be converted
    to its decimal equivalent : )
15 //accepting the binary input from user//
16 bin =11.0111;
17 d =modulo(bin ,1);
18 //separating the decimal part and the integer part//
19 d=d *1010;
20 a = floor (bin) ;
21 //removing the decimal part//
22 while (a >0)

```

```

23 //Loop to take the binary bits of integer into a
    matrix //
24 r = modulo (a ,10) ;
25 b(1,q) = r ;
26 a=a /10;
27 a= floor ( a ) ;
28 q=q +1;
29 end
30 for m =1: q -1
31 // multiplying the bits of integer position values
    and adding//
32 c=m -1;
33 f=f+b(1,m) *(2^ c);
34 end
35 while (d >0)
36 // Loop to take the binary bits of decimal into a
    matrix//
37 e = modulo (d ,2)
38 w(1 ,p)=e
39 d = d /10;
40 d= floor (d)
41 p=p +1;
42 end
43 for n =1: p -1
44 // multiplying the bits of decimal with their
    position values and adding//
45 z=z+w(1 ,n)*(0.5) ^(11 -n);
46 end
47 z = z *10000;
48 //rounding of to 4 decimal values//
49 z= round (z);
50 z = z /10000;
51 x=f+z;
52 disp ( 'The Decimal equivalent of the binary number
    given is ');
53 disp (x); //result is displayed//

```

Scilab code Exa 1.3 binary to decimal conversion

```
1 //binary to decimal conversion//
2 //example 3//
3 clc
4 //clears the command window//
5 clear
6 // clears //
7 p =1;
8 // initialising //
9 q =1;
10 z =0;
11 b =0;
12 w =0;
13 f =0;
14 //bin= input ( Enter the binary no to be converted
    to its decimal equivalent : )
15 //accepting the binary input from user//
16 bin =1100.11;
17 d =modulo(bin ,1);
18 //separating the decimal part and the integer part//
19 d=d *10^10;
20 a = floor (bin) ;
21 //removing the decimal part//
22 while (a >0)
23 //Loop to take the binary bits of integer into a
    matrix //
24 r = modulo (a ,10) ;
25 b(1,q) = r ;
26 a=a /10;
27 a= floor ( a ) ;
28 q=q +1;
29 end
30 for m =1: q -1
```

```

31 // multiplying the bits of integer position values
    and adding//
32 c=m -1;
33 f=f+b(1,m) *(2^ c);
34 end
35 while (d >0)
36 // Loop to take the binary bits of decimal into a
    matrix//
37 e = modulo (d ,2)
38 w(1 ,p)=e
39 d = d /10;
40 d= floor (d)
41 p=p +1;
42 end
43 for n =1: p -1
44 // multiplying the bits of decimal with their
    position values and adding//
45 z=z+w(1 ,n)*(0.5) ^(11 -n);
46 end
47 z = z *10000;
48 //rounding of to 4 decimal values//
49 z= round (z);
50 z = z /10000;
51 x=f+z;
52 disp (x)
53 //result is displayed//

```

Scilab code Exa 1.4 decimal to binary conversion

```

1 //decimal to binary conversion//
2 //example 4//
3 clc
4 //clears the command window//
5 clear
6 //clears//

```

```

7 //decimal to binary conversion//
8 q=0;
9 b=0;
10 s=0;
11 //enter the decimal number to be converted//
12 a=27/32;
13 d=modulo(a,1); //separating decimal and integer part
    //
14 a=floor(a); //removing decimal part//
15 while (a>0)
16 //taking integer part into a matrix and converting
    to equivalent binary//
17 x=modulo(a,2)
18 b=b+(10^q)*x
19 a=a/2
20 a=floor(a)
21 q=q+1
22 end
23 for i=1:10
24 //for values after decimal that should be converted
    to binary//
25 d=d*2
26 q=floor(d)
27 s=s+q/(10^i);
28 if d>=1 then
29     d=d-1
30 end
31 end
32 k=b+s;
33 disp('the binary equivalent of the given decimal
    number is ')
34 disp(k); //displaying the result//

```

Scilab code Exa 1.5 decimal to octal conversion

```
1 //decimal to octal conversion//
2 //example 5//
3 clc
4 //clears the command window//
5 clear
6 //clears//
7 //decimal to octal conversion//
8 x=1993
9 a=dec2oct(x)
10 disp('the number in octal form is:')
11 disp(a)
```

Scilab code Exa 1.6 octal to decimal conversion

```
1 //Octal to decimal conversion//
2 //example 6//
3 clc
4 //clears the command window//
5 clear
6 //clears//
7 //octal to decimal conversion//
8 x='3711'
9 a=oct2dec(x)
10 disp('the number in decimal form :')
11 disp(a)
```

Scilab code Exa 1.7 binary to octal conversion

```
1 //binary to octal conversion//
2 //example 7//
3 clc
4 //clears the control window//
5 clear
```

```
6 //clears//
7 //binary to octal conversion//
8 x='11111001001'
9 a=bin2dec('11111001001')
10 //binary to decimal conversion//
11 z=dec2oct(a)
12 //decimal to octal conversion//
13 disp('the number in octal form:')
14 disp(z)
```

Scilab code Exa 1.8.a decimal to binary conversion

```
1 //decimal to binary conversion//
2 //example 8.a//
3 clc
4 //clears command window //
5 clear
6 //clears//
7 //decimal to binary conversion//
8 x=1996
9 a=dec2bin(1996)
10 disp('the result in binary form is')
11 disp(a)
```

Scilab code Exa 1.8.b binary to decimal conversion

```
1 //binary to decimal conversion//
2 //example 8.b//
3 clc
4 //clears command window //
5 clear
6 //clears//
7 x='11111001100'
```



```

8 a=bin2dec('11111001100'); //binary to decimal
   conversion//
9 disp('the number in decimal form is:')
10 disp(a); //result is displayed//

```

Scilab code Exa 1.9.a decimal to binary conversion

```

1 //decimal to binary conversion//
2 //example 9.a//
3 clc
4 //clears the command window//
5 clear
6 //clears//
7 //decimal to binary conversion//
8 q=0;
9 b=0;
10 s=0;
11 //enter the decimal number to be converted//
12 a=274.1875;
13 d=modulo(a,1); //separating decimal and integer part
   //
14 a=floor(a); //removing decimal part//
15 while (a>0)
16 //taking integer part into a matrix and converting
   to equivalent binary//
17 x=modulo(a,2)
18 b=b+(10^q)*x
19 a=a/2
20 a=floor(a)
21 q=q+1
22   end
23   for i=1:10
24 //for values after decimal that should be converted
   to binary//
25   d=d*2

```

```

26  q=floor(d)
27  s=s+q/(10^i);
28  if d>=1 then
29      d=d-1
30  end
31  end
32  k=b+s;
33  disp('the binary equivalent of the given decimal
        number is ')
34  disp(k); //displaying the result//

```

Scilab code Exa 1.9.b decimal to binary conversion

```

1  //decimal to binary conversion//
2  //example 9.b//
3  clc
4  //clears the command window//
5  clear
6  //clears//
7  //decimal to binary conversion//
8  q=0;
9  b=0;
10 s=0;
11 //enter the decimal number to be converted//
12 a=.65625;
13 d=modulo(a,1); //separating decimal and integer part
    //
14 a=floor(a); //removing decimal part//
15 while (a>0)
16 //taking integer part into a matrix and converting
    to equivalent binary//
17  x=modulo(a,2)
18  b=b+(10^q)*x
19  a=a/2
20  a=floor(a)

```

```

21  q=q+1
22      end
23  for i=1:10
24  //for values after decimal that should be converted
      to binary//
25  d=d*2
26  q=floor(d)
27  s=s+q/(10^i);
28  if d>=1 then
29      d=d-1
30  end
31  end
32  k=b+s;
33  disp('the binary equivalent of the given decimal
      number is:')
34  disp(k); //displaying the result//

```

Scilab code Exa 1.9.c decimal to binary conversion

```

1  //decimal to binary conversion//
2  //example 9.c//
3  clc
4  //clears the command window//
5  clear
6  //clears//
7  //decimal to binary conversion//
8  a=dec2bin(624)
9  disp('binary equivalent of decimal number:')
10 disp(a)
11 //answer in binary form//

```

Scilab code Exa 1.9.d binary to decimal conversion

```
1 //binary to decimal conversion//
2 //example 9.d//
3 clc
4 //clears command window //
5 clear
6 //clears//
7 a=bin2dec('11010'); //binary to decimal conversion//
8 printf('actual age is=%d years',a); //person's actual
    age is displayed//
```

Scilab code Exa 1.10.a hex to decimal conversion

```
1 //hex to decimal conversion//
2 //example 10.a//
3 clc
4 //clears the command window//
5 clear
6 //clears//
7 //hex to decimal conversion//
8 x='7C9'
9 a=hex2dec(x)
10 disp('the number in decimal form is:')
11 disp(a)
```

Scilab code Exa 1.10.b decimal to hex conversion

```
1 //decimal to hex conversion//
2 //example 10.b//
3 clc
4 //clears the command window//
5 clear
6 //clears//
7 //decimal to hex conversion//
```

```
8 x=2047
9 a=dec2hex(2047)
10 disp('the result in hex form is:')
11 disp(a)
```

Scilab code Exa 1.11 hex to binary and decimal conversion

```
1 //hex to binary and decimal conversion//
2 //example 11//
3 clc
4 //clears the command window//
5 clear
6 //clears//
7 //decimal conversion//
8 x='CD42'
9 a=hex2dec('CD42');//hex to decimal conversion//
10 disp(a);//answer displayed in decimal form//
11 //binary conversion//
12 b=dec2bin(a);//dedcimal to binary conversion//
13 disp(b);//answer displayed in binary form//
```

Scilab code Exa 1.12 hex to binary and octal and decimal conversion

```
1 //hex to binary ,octal and decimal conversion//
2 //example 12//
3 clc
4 //clears the command window//
5 clear
6 //clears//
7 //decimal conversion//
8 x='100'
9 d=hex2dec(x);//hex to decimal conversion//
10 b=dec2bin(d);//decimal to binary conversion//
```

```
11 o=dec2oct(d); //decimal to octal conversion//
12 disp(d); //answer in decimal form//
13 disp(b); //answer in binary form//
14 disp(o); //answer in octal form//
```

Scilab code Exa 1.13.a hex to decimal conversion

```
1 //hex to decimal conversion//
2 //example 13.a//
3 clc
4 //clears the command window//
5 clear
6 //clears//
7 x='D2763'
8 a=hex2dec('D2763')
9 disp('the result in decimal form:')
10 disp(a)
```

Scilab code Exa 1.13.b hex to decimal conversion

```
1 //hex to decimal conversion//
2 //example 13.b//
3 clc
4 //clears the command window//
5 clear
6 //clears//
7 x='B3D8'
8 a=hex2dec('B3D8')
9 disp('the result in decimal form:')
10 disp(a)
```

Scilab code Exa 1.14.a addition of two binary numbers

```
1 //addition of two numbers//
2 //example 14.a//
3 clc
4 //clears the command window//
5 clear
6 //clears//
7 //addition of two binary numbers//
8 //binary to decimal conversion//
9 x=bin2dec('100101')
10 y=bin2dec('100101')
11 z=x+y
12 //addition//
13 a=dec2bin(z)
14 //decimal to binary conversion//
15 disp('the addition of two binary numbers is:')
16 disp(a)
```

Scilab code Exa 1.14.b addition of two binary numbers

```
1 //addition of two binary numbers//
2 //example 14.b//
3 clc
4 //clears the command window//
5 clear
6 // clears //; //decimal equivalent of the first
  binary number//
7 //addition of two binary numbers//
8 p =1;
9 // initialising //
10 q =1;
11 z =0;
12 b =0;
13 w =0;
```

```

14 f =0;
15 //bin=1st input ( Enter the binary no to be
    converted to its decimal equivalent : )
16 //accepting the binary input from user//
17 bin =1011.01;
18 d =modulo(bin ,1);
19 //separating the decimal part and the integer part//
20 d=d *10^10;
21 a = floor (bin) ;
22 //removing the decimal part//
23 while (a >0)
24 //Loop to take the binary bits of integer into a
    matrix //
25 r = modulo (a ,10) ;
26 b(1,q) = r ;
27 a=a /10;
28 a= floor ( a ) ;
29 q=q +1;
30 end
31 for m =1: q -1
32 // multiplying the bits of integer position values
    and adding//
33 c=m -1;
34 f=f+b(1,m) *(2^ c);
35 end
36 while (d >0)
37 // Loop to take the binary bits of decimal into a
    matrix//
38 e = modulo (d ,2)
39 w(1 ,p)=e
40 d = d /10;
41 d= floor (d)
42 p=p +1;
43 end
44 for n =1: p -1
45 // multiplying the bits of decimal with their
    position values and adding//
46 z=z+w(1 ,n)*(0.5) ^(11 -n);

```



```

47  end
48  z = z *10000;
49  //rounding of to 4 decimal values//
50  z= round (z);
51  z = z /10000;
52  x1=f+z;
53  disp ( 'The Decimal equivalent of the first binary
         number given is ' );
54  disp (x1);
55  //to find decimal equivalent of second binary number
     //
56  //bin=2nd input ( Enter the binary no to be
         converted to its decimal equivalent : )
57  p =1;
58  // initialising //
59  q =1;
60  z =0;
61  b =0;
62  w =0;
63  f =0;
64  //accepting the binary input from user//
65  bin =1001.11;
66  d =modulo(bin ,1);
67  //separating the decimal part and the integer part//
68  d=d *10^10;
69  a = floor (bin) ;
70  //removing the decimal part//
71  while (a >0)
72  //Loop to take the binary bits of integer into a
         matrix //
73  r = modulo (a ,10) ;
74  b(1,q) = r ;
75  a=a /10;
76  a= floor ( a ) ;
77  q=q +1;
78  end
79  for m =1: q -1
80  // multiplying the bits of integer position values

```

```

        and adding//
81  c=m -1;
82  f=f+b(1,m) *(2^ c);
83  end
84  while (d >0)
85  // Loop to take the binary bits of decimal into a
        matrix//
86  e = modulo (d ,2)
87  w(1 ,p)=e
88  d = d /10;
89  d= floor (d)
90  p=p +1;
91  end
92  for n =1: p -1
93  // multiplying the bits of decimal with their
        position values and adding//
94  z=z+w(1 ,n)*(0.5) ^(11 -n);
95  end
96  z = z *10000;
97  //rounding of to 4 decimal values//
98  z= round (z);
99  z = z /10000;
100 x2=f+z;
101 disp ( 'The Decimal equivalent of the second binary
        number given is ' );
102 disp (x2);
103 res=x1+x2;//addition//
104 //decimal to binary conversion//
105 q=0;
106 b=0;
107 s=0;
108 //enter the decimal number to be converted//
109 a=res;
110 d=modulo(a,1);//separating decimal and integer part
        //
111 a=floor(a);//removing decimal part//
112 while (a>0)
113 //taking integer part into a matrix and converting

```

```

        to equivalent binary//
114 x=modulo(a,2)
115 b=b+(10^q)*x
116 a=a/2
117 a=floor(a)
118 q=q+1
119 end
120 for i=1:10
121 //for values after decimal that should be converted
        to binary//
122 d=d*2
123 q=floor(d)
124 s=s+q/(10^i);
125 if d>=1 then
126     d=d-1
127 end
128 end
129 k=b+s;
130 disp('the addition of two binary numbers is')
131 disp(k); //displaying the result//

```

Scilab code Exa 1.14.c addition of two binary numbers

```

1 //addition of two binary numbers//
2 //example 14.c//
3 clc
4 //clears the command window//
5 clear
6 // clears //;
7 //addition of two binary numbers//
8 //decimal equivalent of the first binary number//
9 p =1;
10 // initialising //
11 q =1;
12 z =0;

```

```

13 b =0;
14 w =0;
15 f =0;
16 //bin=1st input ( Enter the binary no to be
    converted to its decimal equivalent : )
17 //accepting the binary input from user//
18 bin =1100.011;
19 d =modulo(bin ,1);
20 //separating the decimal part and the integer part//
21 d=d *10^10;
22 a = floor (bin) ;
23 //removing the decimal part//
24 while (a >0)
25 //Loop to take the binary bits of integer into a
    matrix //
26 r = modulo (a ,10) ;
27 b(1,q) = r ;
28 a=a /10;
29 a= floor ( a ) ;
30 q=q +1;
31 end
32 for m =1: q -1
33 // multiplying the bits of integer position values
    and adding//
34 c=m -1;
35 f=f+b(1,m) *(2^ c);
36 end
37 while (d >0)
38 // Loop to take the binary bits of decimal into a
    matrix//
39 e = modulo (d ,2)
40 w(1 ,p)=e
41 d = d /10;
42 d= floor (d)
43 p=p +1;
44 end
45 for n =1: p -1
46 // multiplying the bits of decimal with their

```

```

        position values and adding//
47  z=z+w(1 ,n)*(0.5) ^((11 -n));
48  end
49  z = z *10000;
50  //rounding of to 4 decimal values//
51  z= round (z);
52  z = z /10000;
53  x1=f+z;
54  disp ( 'The Decimal equivalent of the first binary
        number given is ');
55  disp (x1);
56  //to find decimal equivalent of second binary number
        //
57  //bin=2nd input ( Enter the binary no to be
        converted to its decimal equivalent : )
58  p =1;
59  // initialising //
60  q =1;
61  z =0;
62  b =0;
63  w =0;
64  f =0;
65  //accepting the binary input from user//
66  bin =1011.011;
67  d =modulo(bin ,1);
68  //separating the decimal part and the integer part//
69  d=d *10^10;
70  a = floor (bin) ;
71  //removing the decimal part//
72  while (a >0)
73  //Loop to take the binary bits of integer into a
        matrix //
74  r = modulo (a ,10) ;
75  b(1,q) = r ;
76  a=a /10;
77  a= floor ( a ) ;
78  q=q +1;
79  end

```

```

80  for m =1: q -1
81  // multiplying the bits of integer position values
      and adding//
82  c=m -1;
83  f=f+b(1,m) *(2^ c);
84  end
85  while (d >0)
86  // Loop to take the binary bits of decimal into a
      matrix//
87  e = modulo (d ,2)
88  w(1 ,p)=e
89  d = d /10;
90  d= floor (d)
91  p=p +1;
92  end
93  for n =1: p -1
94  // multiplying the bits of decimal with their
      position values and adding//
95  z=z+w(1 ,n)*(0.5) ^((11 -n));
96  end
97  z = z *10000;
98  //rounding of to 4 decimal values//
99  z= round (z);
100 z = z /10000;
101 x2=f+z;
102 disp ( 'The Decimal equivalent of the second binary
      number given is ' );
103 disp (x2);
104 res=x1+x2;//addition//
105 q=0;//binary conversion of the decimal answer begins
      here//
106 b=0;
107 s=0;
108 //enter the decimal number to be converted//
109 a=res;
110 d=modulo(a,1);//separating decimal and integer part
      //
111 a=floor(a);//removing decimal part//

```

```

112 while (a>0)
113 //taking integer part into a matrix and converting
    to equivalent binary//
114 x=modulo(a,2)
115 b=b+(10^q)*x
116 a=a/2
117 a=floor(a)
118 q=q+1
119     end
120 for i=1:10
121 //for values after decimal that should be converted
    to binary//
122 d=d*2
123 q=floor(d)
124 s=s+q/(10^i);
125 if d>=1 then
126     d=d-1
127     end
128 end
129 k=b+s;
130 disp('the addition of two numbers give:')
131 disp(k); //displaying the result//

```

Scilab code Exa 1.15.a subtraction of two binary numbers

```

1 //subtraction of two binary numbers//
2 //example 15.a//
3 clc
4 //clears the command window//
5 clear
6 // clears //;
7 //decimal equivalent of the first binary number//
8 p =1;
9 // initialising //
10 q =1;

```

```

11 z =0;
12 b =0;
13 w =0;
14 f =0;
15 //bin=1st input ( Enter the binary no to be
    converted to its decimal equivalent : )
16 //accepting the binary input from user//
17 bin1=110.01;
18 d =modulo(bin1 ,1);
19 //separating the decimal part and the integer part//
20 d=d *10^10;
21 a = floor (bin1) ;
22 //removing the decimal part//
23 while (a >0)
24 //Loop to take the binary bits of integer into a
    matrix //
25 r = modulo (a ,10) ;
26 b(1,q) = r ;
27 a=a /10;
28 a= floor ( a ) ;
29 q=q +1;
30 end
31 for m =1: q -1
32 // multiplying the bits of integer position values
    and adding//
33 c=m -1;
34 f=f+b(1,m) *(2^ c);
35 end
36 while (d >0)
37 // Loop to take the binary bits of decimal into a
    matrix//
38 e = modulo (d ,2)
39 w(1 ,p)=e
40 d = d /10;
41 d= floor (d)
42 p=p +1;
43 end
44 for n =1: p -1

```



```

45 // multiplying the bits of decimal with their
    position values and adding//
46 z=z+w(1 ,n)*(0.5) ^(11 -n);
47 end
48 z = z *10000;
49 //rounding of to 4 decimal values//
50 z= round (z);
51 z = z /10000;
52 x1=f+z;
53 disp ( 'The Decimal equivalent of the first binary
    number given is ' );
54 disp (x1);
55 //to find decimal equivalent of second binary number
    //
56 //bin=2nd input ( Enter the binary no to be
    converted to its decimal equivalent : )
57 p =1;
58 // initialising //
59 q =1;
60 z =0;
61 b =0;
62 w =0;
63 f =0;
64 //accepting the binary input from user//
65 bin2=100.10;
66 d =modulo(bin2,1);
67 //separating the decimal part and the integer part//
68 d=d *10^10;
69 a = floor(bin2) ;
70 //removing the decimal part//
71 while (a >0)
72 //Loop to take the binary bits of integer into a
    matrix //
73 r = modulo (a ,10) ;
74 b(1,q) = r ;
75 a=a /10;
76 a= floor ( a ) ;
77 q=q +1;

```

```

78  end
79  for m =1: q -1
80  // multiplying the bits of integer position values
    and adding//
81  c=m -1;
82  f=f+b(1,m) *(2^ c);
83  end
84  while (d >0)
85  // Loop to take the binary bits of decimal into a
    matrix//
86  e = modulo (d ,2)
87  w(1 ,p)=e
88  d = d /10;
89  d= floor (d)
90  p=p +1;
91  end
92  for n =1: p -1
93  // multiplying the bits of decimal with their
    position values and adding//
94  z=z+w(1 ,n)*(0.5) ^(11 -n);
95  end
96  z = z *10000;
97  //rounding of to 4 decimal values//
98  z= round (z);
99  z = z /10000;
100 x2=f+z;
101 disp ( 'The Decimal equivalent of the second binary
    number given is ' );
102 disp (x2);
103 res=x1-x2;//subtraction//
104 q=0;//binary conversion of the decimal answer begins
    here//
105 b=0;
106 s=0;
107 //enter the decimal number to be converted//
108 a=res;
109 d=modulo(a,1);//separating decimal and integer part
    //

```

```

110 a=floor(a); //removing decimal part//
111 while (a>0)
112 //taking integer part into a matrix and converting
    to equivalent binary//
113 x=modulo(a,2)
114 b=b+(10^q)*x
115 a=a/2
116 a=floor(a)
117 q=q+1
118     end
119 for i=1:10
120 //for values after decimal that should be converted
    to binary//
121 d=d*2
122 q=floor(d)
123 s=s+q/(10^i);
124 if d>=1 then
125     d=d-1
126 end
127 end
128 k=b+s;
129 disp('the subtraction of two numbers give:')
130 disp(k); //displaying the result//

```

Scilab code Exa 1.15.b subtraction of two binary numbers

```

1 //subtraction of two binary numbers//
2 //example 15.b//
3 clc
4 //clears the command window//
5 clear
6 // clears //; //decimal equivalent of the first
    binary number//
7 p =1;
8 // initialising //

```

```

 9 q =1;
10 z =0;
11 b =0;
12 w =0;
13 f =0;
14 //bin=1st input ( Enter the binary no to be
    converted to its decimal equivalent : )
15 //accepting the binary input from user//
16 bin =11.011111;
17 d =modulo(bin ,1);
18 //separating the decimal part and the integer part//
19 d=d *10^10;
20 a = floor (bin) ;
21 //removing the decimal part//
22 while (a >0)
23 //Loop to take the binary bits of integer into a
    matrix //
24 r = modulo (a ,10) ;
25 b(1,q) = r ;
26 a=a /10;
27 a= floor ( a ) ;
28 q=q +1;
29 end
30 for m =1: q -1
31 // multiplying the bits of integer position values
    and adding//
32 c=m -1;
33 f=f+b(1,m) *(2^ c);
34 end
35 while (d >0)
36 // Loop to take the binary bits of decimal into a
    matrix//
37 e = modulo (d ,2)
38 w(1 ,p)=e
39 d = d /10;
40 d= floor (d)
41 p=p +1;
42 end

```

```

43  for n =1: p -1
44  // multiplying the bits of decimal with their
    position values and adding//
45  z=z+w(1 ,n)*(0.5) ^(11 -n);
46  end
47  z = z *10000;
48  //rounding of to 4 decimal values//
49  z= round (z);
50  z = z /10000;
51  x1=f+z;
52  disp ( 'The Decimal equivalent of the first binary
    number given is ' );
53  disp (x1);
54  //to find decimal equivalent of second binary number
    //
55  //bin=2nd input ( Enter the binary no to be
    converted to its decimal equivalent : )
56  p =1;
57  // initialising //
58  q =1;
59  z =0;
60  b =0;
61  w =0;
62  f =0;
63  //accepting the binary input from user//
64  bin =10.01001;
65  d =modulo(bin ,1);
66  //separating the decimal part and the integer part//
67  d=d *10^10;
68  a = floor (bin) ;
69  //removing the decimal part//
70  while (a >0)
71  //Loop to take the binary bits of integer into a
    matrix //
72  r = modulo (a ,10) ;
73  b(1,q) = r ;
74  a=a /10;
75  a= floor ( a ) ;

```

```

76  q=q +1;
77  end
78  for m =1: q -1
79  // multiplying the bits of integer position values
    and adding//
80  c=m -1;
81  f=f+b(1,m) *(2^ c);
82  end
83  while (d >0)
84  // Loop to take the binary bits of decimal into a
    matrix//
85  e = modulo (d ,2)
86  w(1 ,p)=e
87  d = d /10;
88  d= floor (d)
89  p=p +1;
90  end
91  for n =1: p -1
92  // multiplying the bits of decimal with their
    position values and adding//
93  z=z+w(1 ,n)*(0.5) ^(11 -n);
94  end
95  z = z *10000;
96  //rounding of to 4 decimal values//
97  z= round (z);
98  z = z /10000;
99  x2=f+z;
100 disp ( 'The Decimal equivalent of the second binary
    number given is ' );
101 disp (x2);
102 res=x1-x2;//subtraction//
103 q=0;//binary conversion of the decimal answer begins
    here//
104 b=0;
105 s=0;
106 //enter the decimal number to be converted//
107 a=res;
108 d=modulo(a,1);//separating decimal and integer part

```

```

    //
109 a=floor(a); //removing decimal part//
110 while (a>0)
111 //taking integer part into a matrix and converting
    to equivalent binary//
112 x=modulo(a,2)
113 b=b+(10^q)*x
114 a=a/2
115 a=floor(a)
116 q=q+1
117     end
118 for i=1:10
119 //for values after decimal that should be converted
    to binary//
120 d=d*2
121 q=floor(d)
122 s=s+q/(10^i);
123 if d>=1 then
124     d=d-1
125     end
126 end
127 k=b+s;
128 disp('the subtraction of two numbers give:')
129 disp(k); //displaying the result//

```

Scilab code Exa 1.16.a multiplication of two binary numbers

```

1 //multiplication of two binary numbers//
2 //example 16.a//
3 clc
4 //clears the command window//
5 clear
6 // clears //; //decimal equivalent of the first
    binary number//
7 p =1;

```

```

8 // initialising //
9 q =1;
10 z =0;
11 b =0;
12 w =0;
13 f =0;
14 //bin=1st input ( Enter the binary no to be
    converted to its decimal equivalent : )
15 //accepting the binary input from user//
16 bin =1.01;
17 d =modulo(bin ,1);
18 //separating the decimal part and the integer part//
19 d=d *10^10;
20 a = floor (bin) ;
21 //removing the decimal part//
22 while (a >0)
23 //Loop to take the binary bits of integer into a
    matrix //
24 r = modulo (a ,10) ;
25 b(1,q) = r ;
26 a=a /10;
27 a= floor ( a ) ;
28 q=q +1;
29 end
30 for m =1: q -1
31 // multiplying the bits of integer position values
    and adding//
32 c=m -1;
33 f=f+b(1,m) *(2^ c);
34 end
35 while (d >0)
36 // Loop to take the binary bits of decimal into a
    matrix//
37 e = modulo (d ,2)
38 w(1 ,p)=e
39 d = d /10;
40 d= floor (d)
41 p=p +1;

```



```

42 end
43 for n =1: p -1
44 // multiplying the bits of decimal with their
    position values and adding//
45 z=z+w(1 ,n)*(0.5) ^(11 -n);
46 end
47 z = z *10000;
48 //rounding of to 4 decimal values//
49 z= round (z);
50 z = z /10000;
51 x1=f+z;
52 disp ( 'The Decimal equivalent of the first binary
    number given is ' );
53 disp (x1);
54 //to find decimal equivalent of second binary number
    //
55 //bin=2nd input ( Enter the binary no to be
    converted to its decimal equivalent : )
56 p =1;
57 // initialising //
58 q =1;
59 z =0;
60 b =0;
61 w =0;
62 f =0;
63 //accepting the binary input from user//
64 bin =10.1;
65 d =modulo(bin ,1);
66 //separating the decimal part and the integer part//
67 d=d *10^10;
68 a = floor (bin) ;
69 //removing the decimal part//
70 while (a >0)
71 //Loop to take the binary bits of integer into a
    matrix //
72 r = modulo (a ,10) ;
73 b(1,q) = r ;
74 a=a /10;

```

```

75 a= floor ( a ) ;
76 q=q +1;
77 end
78 for m =1: q -1
79 // multiplying the bits of integer position values
    and adding//
80 c=m -1;
81 f=f+b(1,m) *(2^ c);
82 end
83 while (d >0)
84 // Loop to take the binary bits of decimal into a
    matrix//
85 e = modulo (d ,2)
86 w(1 ,p)=e
87 d = d /10;
88 d= floor (d)
89 p=p +1;
90 end
91 for n =1: p -1
92 // multiplying the bits of decimal with their
    position values and adding//
93 z=z+w(1 ,n)*(0.5) ^(11 -n);
94 end
95 z = z *10000;
96 //rounding of to 4 decimal values//
97 z= round (z);
98 z = z /10000;
99 x2=f+z;
100 disp ( 'The Decimal equivalent of the second binary
    number given is ' );
101 disp (x2);
102 res=x1*x2;//multiplication//
103 q=0;//binary conversion of the decimal answer begins
    here//
104 b=0;
105 s=0;
106 //enter the decimal number to be converted//
107 a=res;

```

```

108 d=modulo(a,1); //separating decimal and integer part
    //
109 a=floor(a); //removing decimal part//
110 while (a>0)
111 //taking integer part into a matrix and converting
    to equivalent binary//
112 x=modulo(a,2)
113 b=b+(10^q)*x
114 a=a/2
115 a=floor(a)
116 q=q+1
117     end
118 for i=1:10
119 //for values after decimal that should be converted
    to binary//
120 d=d*2
121 q=floor(d)
122 s=s+q/(10^i);
123 if d>=1 then
124     d=d-1
125 end
126 end
127 k=b+s;
128 disp('the multiplication of two numbers give:')
129 disp(k); //displaying the result//

```

Scilab code Exa 1.16.b multiplication of two binary numbers

```

1 //multiplication of two binary numbers//
2 //example 16.b//
3 clc
4 //clears the command window//
5 clear
6 // clears //;
7 //decimal equivalent of the first binary number//

```

```

8 p =1;
9 // initialising //
10 q =1;
11 z =0;
12 b =0;
13 w =0;
14 f =0;
15 //bin=1st input ( Enter the binary no to be
    converted to its decimal equivalent : )
16 //accepting the binary input from user//
17 bin =101.01;
18 d =modulo(bin ,1);
19 //separating the decimal part and the integer part//
20 d=d *10^10;
21 a = floor (bin) ;
22 //removing the decimal part//
23 while (a >0)
24 //Loop to take the binary bits of integer into a
    matrix //
25 r = modulo (a ,10) ;
26 b(1,q) = r ;
27 a=a /10;
28 a= floor ( a ) ;
29 q=q +1;
30 end
31 for m =1: q -1
32 // multiplying the bits of integer position values
    and adding//
33 c=m -1;
34 f=f+b(1,m) *(2^ c);
35 end
36 while (d >0)
37 // Loop to take the binary bits of decimal into a
    matrix//
38 e = modulo (d ,2)
39 w(1 ,p)=e
40 d = d /10;
41 d= floor (d)

```

```

42  p=p +1;
43  end
44  for n =1: p -1
45  // multiplying the bits of decimal with their
      position values and adding//
46  z=z+w(1 ,n)*(0.5) ^(11 -n);
47  end
48  z = z *10000;
49  //rounding of to 4 decimal values//
50  z= round (z);
51  z = z /10000;
52  x1=f+z;
53  disp ( 'The Decimal equivalent of the first binary
      number given is ' );
54  disp (x1);
55  //to find decimal equivalent of second binary number
      //
56  //bin=2nd input ( Enter the binary no to be
      converted to its decimal equivalent : )
57  //accepting the binary input from user//
58  bin ='11';
59  x2=bin2dec(bin)
60  disp ( 'The Decimal equivalent of the second binary
      number given is ' );
61  disp (x2);
62  res=x1*x2;//multiplication//
63  q=0;//binary conversion of the decimal answer begins
      here//
64  b=0;
65  s=0;
66  //enter the decimal number to be converted//
67  a=res;
68  d=modulo(a,1);//separating decimal and integer part
      //
69  a=floor(a);//removing decimal part//
70  while (a>0)
71  //taking integer part into a matrix and converting
      to equivalent binary//

```

```

72 x=modulo(a,2)
73 b=b+(10^q)*x
74 a=a/2
75 a=floor(a)
76 q=q+1
77     end
78 for i=1:10
79 //for values after decimal that should be converted
    to binary//
80 d=d*2
81 q=floor(d)
82 s=s+q/(10^i);
83 if d>=1 then
84     d=d-1
85     end
86 end
87 k=b+s;
88 disp('the multiplication of two numbers give:')
89 disp(k);//displaying the result//

```

Scilab code Exa 1.17.a division of two binary numbers

```

1 //division of two binary numbers//
2 //example 17.a//
3 clc
4 //clears the command window//
5 clear
6 // clears //; //decimal equivalent of the first
    binary number//
7 //bin=1st input ( Enter the binary no to be
    converted to its decimal equivalent : )
8 //accepting the binary input from user//
9 bin = '1111001'; //1st input//
10 x1=bin2dec(bin)
11 bin='1001'; //2nd input//

```

```

12 x2=bin2dec(bin)
13 res=x1/x2;//division//
14 q=0;
15 b=0;
16 s=0;
17 //enter the decimal number to be converted//
18 a=res;
19 d=modulo(a,1);//separating decimal and integer part
    //
20 a=floor(a);//removing decimal part//
21 while (a>0)
22 //taking integer part into a matrix and converting
    to equivalent binary//
23 x=modulo(a,2)
24 b=b+(10^q)*x
25 a=a/2
26 a=floor(a)
27 q=q+1
28 end
29 for i=1:10
30 //for values after decimal that should be converted
    to binary//
31 d=d*2
32 q=floor(d)
33 s=s+q/(10^i);
34 if d>=1 then
35     d=d-1
36 end
37 end
38 k=b+s;
39 disp('the division of two numbers give:')
40 disp(k);//displaying the result//

```

Scilab code Exa 1.17.b division of two binary numbers

```

1 //division of two binary numbers//
2 //example 17.b//
3 clc
4 //clears the command window//
5 clear
6 // clears //; //decimal equivalent of the first
   binary number//
7 p =1;
8 // initialising //
9 q =1;
10 z =0;
11 b =0;
12 w =0;
13 f =0;
14 //bin= input ( Enter the binary no to be converted
   to its decimal equivalent : )
15 //accepting the binary input from user//
16 bin =11.11;
17 d =modulo(bin ,1);
18 //separating the decimal part and the integer part//
19 d=d *10^10;
20 a = floor (bin) ;
21 //removing the decimal part//
22 while (a >0)
23 //Loop to take the binary bits of integer into a
   matrix //
24 r = modulo (a ,10) ;
25 b(1,q) = r ;
26 a=a /10;
27 a= floor ( a ) ;
28 q=q +1;
29 end
30 for m =1: q -1
31 // multiplying the bits of integer position values
   and adding//
32 c=m -1;
33 f=f+b(1,m) *(2^ c);
34 end

```



```

35  while (d >0)
36  // Loop to take the binary bits of decimal into a
    matrix//
37  e = modulo (d ,2)
38  w(1 ,p)=e
39  d = d /10;
40  d= floor (d)
41  p=p +1;
42  end
43  for n =1: p -1
44  // multiplying the bits of decimal with their
    position values and adding//
45  z=z+w(1 ,n)*(0.5) ^(11 -n);
46  end
47  z = z *10000;
48  //rounding of to 4 decimal values//
49  z= round (z);
50  z = z /10000;
51  x1=f+z;
52  p =1;
53  // initialising //
54  q =1;
55  z =0;
56  b =0;
57  w =0;
58  f =0;
59  //bin=2nd input ( Enter the binary no to be
    converted to its decimal equivalent : )
60  //accepting the binary input from user//
61  bin =.101;
62  d =modulo(bin ,1);
63  //separating the decimal part and the integer part//
64  d=d *10^10;
65  a = floor (bin) ;
66  //removing the decimal part//
67  while (a >0)
68  //Loop to take the binary bits of integer into a
    matrix //

```

```

69  r = modulo (a ,10) ;
70  b(1,q) = r ;
71  a=a /10;
72  a= floor ( a ) ;
73  q=q +1;
74  end
75  for m =1: q -1
76  // multiplying the bits of integer position values
    and adding//
77  c=m -1;
78  f=f+b(1,m) *(2^ c);
79  end
80  while (d >0)
81  // Loop to take the binary bits of decimal into a
    matrix//
82  e = modulo (d ,2)
83  w(1 ,p)=e
84  d = d /10;
85  d= floor (d)
86  p=p +1;
87  end
88  for n =1: p -1
89  // multiplying the bits of decimal with their
    position values and adding//
90  z=z+w(1 ,n)*(0.5) ^(11 -n);
91  end
92  z = z *10000;
93  //rounding of to 4 decimal values//
94  z= round (z);
95  z = z /10000;
96  x2=f+z;
97  res=x1/x2;//division//
98  q=0;
99  b=0;
100 s=0;
101 //enter the decimal number to be converted//
102 a=res;
103 d=modulo(a,1);//separating decimal and integer part

```

```

    //
104 a=floor(a); //removing decimal part//
105 while (a>0)
106 //taking integer part into a matrix and converting
    to equivalent binary//
107 x=modulo(a,2)
108 b=b+(10^q)*x
109 a=a/2
110 a=floor(a)
111 q=q+1
112     end
113 for i=1:10
114 //for values after decimal that should be converted
    to binary//
115 d=d*2
116 q=floor(d)
117 s=s+q/(10^i);
118 if d>=1 then
119     d=d-1
120 end
121 end
122 k=b+s;
123 disp('the division of two numbers give:')
124 disp(k); //displaying the result//

```

Scilab code Exa 1.18.a binary to hex conversion

```

1 //binary to hex conversion//
2 //example 18.a//
3 clc
4 //clears the command window//
5 clear
6 //clears//
7 x='10100110'
8 a=bin2dec(x); //binary to decimal conversion//

```

```
9 res=dec2hex(a); //decimal to hex conversion//
10 disp('the number in hexadecimal form:')
11 disp(res); //answer displayed//
```

Scilab code Exa 1.18.b decimal to octal conversion

```
1 //decimal to octal conversion//
2 //example 18.b//
3 clc
4 //clears the command window//
5 clear
6 //clears//
7 x=6800
8 res=dec2oct(x); //decimal to octal conversion//
9 disp(res); //answer in octal form//
```

Scilab code Exa 1.18.c binary to decimal conversion

```
1 //decimal to binary conversion//
2 //example 18.c//
3 clc
4 //clears the command window//
5 clear
6 // clears //
7 //converting first binary input to decimal//
8 p =1;
9 // initialising //
10 q =1;
11 z =0;
12 b =0;
13 w =0;
14 f =0;
```

```

15 //bin= input ( Enter the binary no to be converted
    to its decimal equivalent : )
16 bin =.0111;
17 d =modulo(bin ,1);
18 //separating the decimal part and the integer part//
19 d=d *10^10;
20 a = floor (bin) ;
21 //removing the decimal part//
22 while (a >0)
23 //Loop to take the binary bits of integer into a
    matrix //
24 r = modulo (a ,10) ;
25 b(1,q) = r ;
26 a=a /10;
27 a= floor ( a ) ;
28 q=q +1;
29 end
30 for m =1: q -1
31 // multiplying the bits of integer position values
    and adding//
32 c=m -1;
33 f=f+b(1,m) *(2^ c);
34 end
35 while (d >0)
36 // Loop to take the binary bits of decimal into a
    matrix//
37 e = modulo (d ,2)
38 w(1 ,p)=e
39 d = d /10;
40 d= floor (d)
41 p=p +1;
42 end
43 for n =1: p -1
44 // multiplying the bits of decimal with their
    position values and adding//
45 z=z+w(1 ,n)*(0.5) ^(11 -n);
46 end
47 z = z *10000;

```

```

48 //rounding of to 4 decimal values//
49 z= round (z);
50 z = z /10000;
51 d1=f+z;
52 disp('the number in binary form:')
53 disp(d1);//displaying the result//

```

Scilab code Exa 1.19.a decimal to octal conversion

```

1 //decimal to octal conversion//
2 //example 19.a//
3 clc
4 //clears the command window//
5 clear
6 //clears//
7 //decimal to octal conversion//
8 q=0;
9 b=0;
10 s=0;
11 //enter the decimal number to be converted//
12 a=4429.625;
13 d=modulo(a,1);//separating decimal and integer part
    //
14 a=floor(a);//removing decimal part//
15 while (a>0)
16 //taking integer part into a matrix and converting
    to equivalent octal//
17 x=modulo(a,8)
18 b=b+(10^q)*x
19 a=a/8
20 a=floor(a)
21 q=q+1
22 end
23 for i=1:10
24 //for values after decimal that should be

```

```

                converted to octal//
25     d=d*8
26     q=floor(d)
27     s=s+q/(10^i);
28     d=d-q
29     end
30     k=b+s;
31     disp('the octal equivalent of the given decimal
           number is ')
32     disp(k); //displaying the result//

```

Scilab code Exa 1.19.b decimal to octal conversion

```

1 //decimal to octal conversion//
2 //example 19.b//
3 clc
4 //clears the command window//
5 clear
6 //clears//
7 //decimal to octal conversion//
8 q=0;
9 b=0;
10 s=0;
11 //enter the decimal number to be converted//
12 a=791.125;
13 d=modulo(a,1); //separating decimal and integer part
    //
14 a=floor(a); //removing decimal part//
15 while (a>0)
16 //taking integer part into a matrix and converting
    to equivalent octal//
17 x=modulo(a,8)
18 b=b+(10^q)*x
19 a=a/8
20 a=floor(a)

```

```

21  q=q+1
22      end
23  for i=1:10
24      //for values after decimal that should be
          converted to octal//
25      d=d*8
26      q=floor(d)
27      s=s+q/(10^i);
28      d=d-q
29  end
30  k=b+s;
31  disp('the octal equivalent of the given decimal
          number is ')
32  disp(k); //displaying the result//

```

Scilab code Exa 1.19.c decimal to octal conversion

```

1  //decimal to octal conversion//
2  //example 19.c//
3  clc
4  //clears the command window//
5  clear
6  //clears//
7  //decimal to octal conversion//
8  q=0;
9  b=0;
10 s=0;
11 //enter the decimal number to be converted//
12 a=11.9375;
13 d=modulo(a,1); //separating decimal and integer part
          //
14 a=floor(a); //removing decimal part//
15 while (a>0)
16     //taking integer part into a matrix and converting
          to equivalent octal//

```



```

17  x=modulo(a,8)
18  b=b+(10^q)*x
19  a=a/8
20  a=floor(a)
21  q=q+1
22  end
23  for i=1:10
24      //for values after decimal that should be
          converted to octal//
25      d=d*8
26      q=floor(d)
27      s=s+q/(10^i);
28      d=d-q
29  end
30  k=b+s;
31  disp('the octal equivalent of the given decimal
          number is ')
32  disp(k); //displaying the result//

```

Scilab code Exa 1.20.a decimal to signed binary number conversion

```

1  //decimal to signed binary number//
2  //example 20.a//
3  //decimal to signed binary number system//
4  clc
5  //clears the command window//
6  clear
7  //clears//
8  x=+29
9  a=dec2bin(x,6)
10 //decimal to binary conversion//
11 disp('the answer is ')
12 disp(a)
13 //since the number is positive it starts with a 0//

```

Scilab code Exa 1.20.b decimal to signed binary system conversion

```
1 //decimal to signed binary number//
2 //example 20.b//
3 clc
4 //clears the command window//
5 clear
6 //clears//
7 //decimal to signed binary number system//
8 x=-29
9 a=dec2bin(-x)
10 //decimal to binary conversion//
11 a=dec2bin(-x+bin2dec('100000'))
12 disp('the answer is:')
13 disp(a)
14 //since the number is negative it starts with a 1//
```

Scilab code Exa 1.20.c decimal to signed binary system conversion

```
1 //decimal to signed binary number//
2 //example 20.c//
3 clc
4 //clears the command window//
5 clear
6 //clears//
7 //decimal to signed binary number system//
8 x=-19
9 a=dec2bin(-x)
10 //decimal to binary conversion//
11 a=dec2bin(-x+bin2dec('100000'))
12 disp('the answer is:')
13 disp(a)
```

```
14 //since the number is negative it starts with a 1,  
    msb indicates the sign//
```

Scilab code Exa 1.21.a subtraction using ones complement

```
1 //subtraction using 1's complement//  
2 //Example 21.a//  
3 //subtraction in one's complement using 8 bits//  
4 clc  
5 //clears the window//  
6 clear  
7 //clears all the existing variables//  
8 x='11001'  
9 y=bin2dec('11001')  
10 z=bin2dec('10110')  
11 c=(bitcmp(z,5));//finding 1's complement//  
12 a=y+c+1  
13 a=a-bin2dec('100000')  
14 a=dec2bin(a,5);//converting the result to binary  
    equivalent//  
15 disp('the subtraction of two numbers give:')  
16 disp(a);//displaying the result//
```

Scilab code Exa 1.21.b subtraction using ones complement

```
1 //subtraction using 1's complement//  
2 //Example 21.b//  
3 //subtraction in one's complement using 8 bits//  
4 clc  
5 //clears the window//  
6 clear  
7 //clears all the existing variables//  
8 x='1011'
```

```

9 y=bin2dec('1011')
10 z=bin2dec('0101')
11 c=(bitcmp(z,4));//finding 1's complement//
12 a=y+c+1
13 a=a-bin2dec('10000');
14 a=dec2bin(a,4);//converting the result to binary
    equivalent//
15 disp('the subtraction of two binary numbers give:')
16 disp(a);//displaying the result//

```

Scilab code Exa 1.21.c subtraction using ones complement

```

1 //subtraction using 1's complement//
2 //Example 21.c//
3 //subtraction in one's complement using 8 bits//
4 clc
5 //clears the window//
6 clear
7 //clears all the existing variables//
8 x='11011'
9 y=bin2dec('11011')
10 z=bin2dec('11001')
11 c=(bitcmp(z,5));//finding 1's complement//
12 a=y+c+1
13 a=a-bin2dec('100000');
14 a=dec2bin(a);//converting the result to binary
    equivalent//
15 disp('the subtraction of two binary numbers give:')
16 disp(a);//displaying the result//

```

Scilab code Exa 1.21.d subtraction using ones complement

```

1 //subtraction using 1's complement//

```

```

2 //example 21.d//
3 clc
4 //clears the command window//
5 clear
6 // clears //
7 //converting first binary input to decimal//
8 p =1;
9 // initialising //
10 q =1;
11 z =0;
12 b =0;
13 w =0;
14 f =0;
15 //bin= input ( Enter the binary no to be converted
    to its decimal equivalent : )
16 //accepting the first binary input from user//
17 bin =10111.1;
18 d =modulo(bin ,1);
19 //separating the decimal part and the integer part//
20 d=d *10^10;
21 a =floor(bin) ;
22 //removing the decimal part//
23 while (a >0)
24 //Loop to take the binary bits of integer into a
    matrix //
25 r = modulo (a ,10) ;
26 b(1,q) = r ;
27 a=a /10;
28 a= floor ( a ) ;
29 q=q +1;
30 end
31 for m =1: q -1
32 // multiplying the bits of integer position values
    and adding//
33 c=m -1;
34 f=f+b(1,m) *(2^ c);
35 end
36 while (d >0)

```

```

37 // Loop to take the binary bits of decimal into a
    matrix//
38 e = modulo (d ,2)
39 w(1 ,p)=e
40 d = d /10;
41 d= floor (d)
42 p=p +1;
43 end
44 for n =1: p -1
45 // multiplying the bits of decimal with their
    position values and adding//
46 z=z+w(1 ,n)*(0.5) ^(11 -n);
47 end
48 z = z *10000;
49 //rounding of to 4 decimal values//
50 z= round (z);
51 z = z /10000;
52 d1=f+z;
53 //converting second binary input to its complement
    decimal number//
54 p =1;
55 // initialising //
56 q =1;
57 z =0;
58 b =0;
59 w =0;
60 f =0;
61 //accepting 2nd binary input//
62 a=10011.1
63 s=0;
64 d=modulo(a,1); //separating integer and decimal part
    //
65 //since complement of decimal part is 0,we do not
    consider it further//
66 a=floor(a); //getting the integer part//
67 for i=1:5
68     x(i)=modulo(a,10)
69     if x(i)==1 then

```

```

70         x(i)=0
71     else
72         x(i)=1
73     end
74     s=s+(x(i)*(2^(i-1)))
75     a=a/10
76     a=floor(a)
77 end
78 d2=s; //assigning the equivalent decimal value to d2
      //
79 r=d1+d2+.5; //adding 1 to the decimal equivalents to
      find the result//
80 r=r-bin2dec('100000');
81 r=dec2bin(r); //converting the result to binary
      equivalent//
82 disp('the subtraction of two binary numbers give:')
83 disp(r); //displaying the result//

```

Scilab code Exa 1.21.e subtraction using ones complement

```

1 //subtraction using 1's complement//
2 //example 21.e//
3 clc
4 //clears the command window//
5 clear
6 // clears //
7 //converting first binary input to decimal//
8 p =1;
9 // initialising //
10 q =1;
11 z =0;
12 b =0;
13 w =0;
14 f =0;
15 //bin= input ( Enter the binary no to be converted

```

```

        to its decimal equivalent : )
16 //accepting the first binary input from user//
17 bin =11011.00;
18 d =modulo(bin ,1);
19 //separating the decimal part and the integer part//
20 d=d *10^10;
21 a = floor (bin) ;
22 //removing the decimal part//
23 while (a >0)
24 //Loop to take the binary bits of integer into a
    matrix //
25 r = modulo (a ,10) ;
26 b(1,q) = r ;
27 a=a /10;
28 a= floor ( a ) ;
29 q=q +1;
30 end
31 for m =1: q -1
32 // multiplying the bits of integer position values
    and adding//
33 c=m -1;
34 f=f+b(1,m) *(2^ c);
35 end
36 while (d >0)
37 // Loop to take the binary bits of decimal into a
    matrix//
38 e = modulo (d ,2)
39 w(1 ,p)=e
40 d = d /10;
41 d= floor (d)
42 p=p +1;
43 end
44 for n =1: p -1
45 // multiplying the bits of decimal with their
    position values and adding//
46 z=z+w(1 ,n)*(0.5) ^(11 -n);
47 end
48 z = z *10000;

```



```

49 //rounding of to 4 decimal values//
50 z= round (z);
51 z = z /10000;
52 d1=f+z;
53 //converting second binary input to its complement
    decimal number//
54 p =1;
55 // initialising //
56 q =1;
57 z =0;
58 b =0;
59 w =0;
60 f =0;
61 a=10011.1
62 s=0;
63 d=modulo(a,1); //separating integer and decimal part
    //
64 //since complement of decimal part is 0,we do not
    consider it further//
65 a=floor(a); //getting the integer part//
66 for i=1:5
67     x(i)=modulo(a,10)
68     if x(i)==1 then
69         x(i)=0
70     else
71         x(i)=1
72     end
73     s=s+(x(i)*(2^(i-1)))
74     a=a/10
75     a=floor(a)
76 end
77 d2=s; //assigning the equivalent decimal value to d2
    //
78 r=d1+d2+.25; //adding 1 to the decimal equivalents to
    find the result//
79 disp(r)
80 r=r-bin2dec('100000')
81 q=0;

```

```

82 b=0;
83 s=0;
84 //enter the decimal number to be converted//
85 a=r;
86 d=modulo(a,1); //separating decimal and integer part
    //
87 a=floor(a); //removing decimal part//
88 while (a>0)
89     //taking integer part into a matrix and converting
        to equivalent binary//
90     x=modulo(a,2)
91     b=b+(10^q)*x
92     a=a/2
93     a=floor(a)
94     q=q+1
95     end
96 for i=1:10
97     //for values after decimal that should be converted
        to binary//
98     d=d*2
99     q=floor(d)
100    s=s+q/(10^i);
101    if d>=1 then
102        d=d-1
103    end
104 end
105 k=b+s;
106 disp('the subtraction of two numbers give :')
107 disp(k); //displaying the result//

```

Scilab code Exa 1.22.a subtraction using twos complement

```

1 //subtraction using 2's complement//
2 //Example 22.a//
3 //subtraction using 2's complement//

```

```

4  clc
5  //clears the console//
6  clear
7  //clears all existing variables//
8  a=bin2dec('11011')
9  b=bin2dec('11001')
10 x=bitcmp(b,5)
11 //complement of the numbers//
12 z=1
13 u=x+z
14 //1 is added to the complements//
15 w=u+a
16 w=w-bin2dec('100000')
17 w=dec2bin(w);//converting to binary form//
18 disp('the subtraction of two binary numbers give:')
19 disp(w);//displaying the result//

```

Scilab code Exa 1.22.b subtraction using twos complement

```

1  //subtraction using 2's complement//
2  //Example 22.b//
3  //subtraction using 2's complement//
4  clc
5  //clears the console//
6  clear
7  //clears all existing variables//
8  a=bin2dec('11011')
9  b=bin2dec('01100')
10 x=bitcmp(b,5)
11 //complement of the numbers//
12 z=1
13 u=x+z
14 //1 is added to the complements//
15 w=u+a
16 w=w-bin2dec('100000')

```

```

17 w=dec2bin(w);//coverting to binary form//
18 disp('the subtraction of two binary numbers give:')
19 disp(w);//displaying the result//

```

Scilab code Exa 1.22.c subtraction using twos complement

```

1 //subtraction using 2's complement//
2 //example 22.c//
3 clc
4 //clears the command window//
5 clear
6 // clears //
7 //converting first binary input to decimal//
8 p =1;
9 // initialising //
10 q =1;
11 z =0;
12 b =0;
13 w =0;
14 f =0;
15 //bin= input ( Enter the binary no to be converted
    to its decimal equivalent : )
16 //accepting the first binary input from user//
17 bin =0.01111;
18 bin=bin*100000
19 y=0;
20 for i=1:4
21     x(i)=modulo(bin,10)
22     y=y+(x(i)/(2^(6-i)))
23     bin=bin/10
24     bin=floor(bin)
25 end
26 d1=y
27 //converting second binary input to its complement
    decimal number//

```

```

28 a=.01001
29 a=a*100000
30 s=0;
31 for i=1:4
32     x(i)=modulo(a,10)
33     if x(i)==1 then
34         x(i)=0
35     else
36         x(i)=1
37     end
38     s=s+(x(i)/(2^(6-i)))
39     a=a/10
40     a=floor(a)
41 end
42 s=s+.5
43 d2=s; //assigning the equivalent decimal value to d2
    //
44 d2=d2+.03125;
45 r=d1+d2; //result in decimal form//
46 r=r-bin2dec('1')
47 //converting to binary form//
48 q=0;
49 b=0;
50 s=0;
51 //enter the decimal number to be converted//
52 a=r;
53 d=modulo(a,1); //separating decimal and integer part
    //
54 a=floor(a); //removing decimal part//
55 while (a>0)
56 //taking integer part into a matrix and converting
    to equivalent binary//
57     x=modulo(a,2)
58     b=b+(10^q)*x
59     a=a/2
60     a=floor(a)
61     q=q+1
62     end

```

```

63  for i=1:10
64  //for values after decimal that should be converted
        to binary//
65  d=d*2
66  q=floor(d)
67  s=s+q/(10^i);
68  if d>=1 then
69      d=d-1
70  end
71  end
72  k=b+s;
73  disp('the subtraction of two binary numbers give:')
74  disp(k);//the result in binary form//

```

Scilab code Exa 1.22.d subtraction using twos complement

```

1  //subtraction using 2's complement//
2  //example 22.d//
3  clc
4  //clears the command window//
5  clear
6  // clears //
7  //converting first binary input to decimal//
8  p =1;
9  // initialising //
10 q =1;
11 z =0;
12 b =0;
13 w =0;
14 f =0;
15 //bin= input ( Enter the binary no to be converted
        to its decimal equivalent : )
16 //accepting the first binary input from user//
17 bin =111.010;
18 d =modulo(bin ,1);

```

```

19 //separating the decimal part and the integer part//
20 d=d *10^10;
21 a = floor (bin) ;
22 //removing the decimal part//
23 while (a >0)
24 //Loop to take the binary bits of integer into a
    matrix //
25 r = modulo (a ,10) ;
26 b(1,q) = r ;
27 a=a /10;
28 a= floor ( a ) ;
29 q=q +1;
30 end
31 for m =1: q -1
32 // multiplying the bits of integer position values
    and adding//
33 c=m -1;
34 f=f+b(1,m) *(2^ c);
35 end
36 while (d >0)
37 // Loop to take the binary bits of decimal into a
    matrix//
38 e = modulo (d ,2)
39 w(1 ,p)=e
40 d = d /10;
41 d= floor (d)
42 p=p +1;
43 end
44 for n =1: p -1
45 // multiplying the bits of decimal with their
    position values and adding//
46 z=z+w(1 ,n)*(0.5) ^(11 -n);
47 end
48 z = z *10000;
49 //rounding of to 4 decimal values//
50 z= round (z);
51 z = z /10000;
52 d1=f+z;

```

```

53 //converting second binary input to its complement
    decimal number//
54 a=010.111;
55 s=0;
56 d=modulo(a,1); //separating integer and decimal part
    //
57 //since complement of decimal part is 0,we do not
    consider it further//
58 a=floor(a); //getting the integer part//
59 for i=1:3
60     x(i)=modulo(a,10)
61     if x(i)==1 then
62         x(i)=0
63     else
64         x(i)=1
65     end
66     s=s+(x(i)*(2^(i-1)))
67     a=a/10
68     a=floor(a)
69 end
70 d2=s; //assigning the equivalent decimal value to d2
    //
71 d2=d2+.125; //since decimal is upto three places//
72 r=d1+d2;
73 r=r-bin2dec('1000'); //step to ignore the msb//
74 //converting the result to its binary equivalent//
75 q=0;
76 b=0;
77 s=0;
78 //enter the decimal number to be converted//
79 a=r;
80 d=modulo(a,1); //separating decimal and integer part
    //
81 a=floor(a); //removing decimal part//
82 while (a>0)
83 //taking integer part into a matrix and converting
    to equivalent binary//
84 x=modulo(a,2)

```



```

85 b=b+(10^q)*x
86 a=a/2
87 a=floor(a)
88 q=q+1
89     end
90     for i=1:10
91         //for values after decimal that should be converted
           to binary//
92         d=d*2
93         q=floor(d)
94         s=s+q/(10^i);
95         if d>=1 then
96             d=d-1
97         end
98     end
99     k=b+s;
100 disp('the subtraction of two binary numbers give:')
101 disp(k); //required result in binary form//

```

Scilab code Exa 1.22.e subtraction using twos complement

```

1 //subtraction using 2's complement//
2 //example 22.e//
3 clc
4 //clears the command window//
5 clear
6 // clears //
7 //converting first binary input to decimal//
8 p =1;
9 // initialising //
10 q =1;
11 z =0;
12 b =0;
13 w =0;
14 f =0;

```

```

15 //bin= input ( Enter the binary no to be converted
    to its decimal equivalent : )
16 //accepting the first binary input from user//
17 bin =111.01;
18 d =modulo(bin ,1);
19 //separating the decimal part and the integer part//
20 d=d *10^10;
21 a = floor (bin) ;
22 //removing the decimal part//
23 while (a >0)
24 //Loop to take the binary bits of integer into a
    matrix //
25 r = modulo (a ,10) ;
26 b(1,q) = r ;
27 a=a /10;
28 a= floor ( a ) ;
29 q=q +1;
30 end
31 for m =1: q -1
32 // multiplying the bits of integer position values
    and adding//
33 c=m -1;
34 f=f+b(1,m) *(2^ c);
35 end
36 while (d >0)
37 // Loop to take the binary bits of decimal into a
    matrix//
38 e = modulo (d ,2)
39 w(1 ,p)=e
40 d = d /10;
41 d= floor (d)
42 p=p +1;
43 end
44 for n =1: p -1
45 // multiplying the bits of decimal with their
    position values and adding//
46 z=z+w(1 ,n)*(0.5) ^(11 -n);
47 end

```

```

48 z = z *10000;
49 //rounding of to 4 decimal values//
50 z= round (z);
51 z = z /10000;
52 d1=f+z;
53 //converting second binary input to its complement
    decimal number//
54 a=110.11;
55 s=0;
56 d=modulo(a,1); //separating integer and decimal part
    //
57 //since complement of decimal part is 0,we do not
    consider it further//
58 a=floor(a); //getting the integer part//
59 for i=1:3
60     x(i)=modulo(a,10)
61     if x(i)==1 then
62         x(i)=0
63     else
64         x(i)=1
65     end
66     s=s+(x(i)*(2^(i-1)))
67     a=a/10
68     a=floor(a)
69 end
70 d2=s; //assigning the equivalent decimal value to d2
    //
71 d2=d2+.25; //since decimal is upto three places//
72 r=d1+d2;
73 r=r-bin2dec('1000'); //step to ignore the msb//
74 //converting the result to its binary equivalent//
75 q=0;
76 b=0;
77 s=0;
78 //enter the decimal number to be converted//
79 a=r;
80 d=modulo(a,1); //separating decimal and integer part
    //

```

```

81 a=floor(a); //removing decimal part//
82 while (a>0)
83 //taking integer part into a matrix and converting
    to equivalent binary//
84 x=modulo(a,2)
85 b=b+(10^q)*x
86 a=a/2
87 a=floor(a)
88 q=q+1
89     end
90 for i=1:10
91 //for values after decimal that should be converted
    to binary//
92 d=d*2
93 q=floor(d)
94 s=s+q/(10^i);
95 if d>=1 then
96     d=d-1
97 end
98 end
99 k=b+s;
100 disp('the subtraction of two binary numbers give:')
101 disp(k); //required result in binary form//

```

Scilab code Exa 1.22.f subtraction using twos complement

```

1 //subtraction using 2's complement//
2 //example 22.f//
3 clc
4 //clears the command window//
5 clear
6 // clears //
7 //converting first binary input to decimal//
8 p =1;
9 // initialising //

```

```

10 q =1;
11 z =0;
12 b =0;
13 w =0;
14 f =0;
15 //bin= input ( Enter the binary no to be converted
    to its decimal equivalent : )
16 //accepting the first binary input from user//
17 bin =10111.1;
18 d =modulo(bin ,1);
19 //separating the decimal part and the integer part//
20 d=d *10^10;
21 a = floor (bin) ;
22 //removing the decimal part//
23 while (a >0)
24 //Loop to take the binary bits of integer into a
    matrix //
25 r = modulo (a ,10) ;
26 b(1,q) = r ;
27 a=a /10;
28 a= floor ( a ) ;
29 q=q +1;
30 end
31 for m =1: q -1
32 // multiplying the bits of integer position values
    and adding//
33 c=m -1;
34 f=f+b(1,m) *(2^ c);
35 end
36 while (d >0)
37 // Loop to take the binary bits of decimal into a
    matrix//
38 e = modulo (d ,2)
39 w(1 ,p)=e
40 d = d /10;
41 d= floor (d)
42 p=p +1;
43 end

```

```

44  for n =1: p -1
45  // multiplying the bits of decimal with their
    position values and adding//
46  z=z+w(1 ,n)*(0.5) ^(11 -n);
47  end
48  z = z *10000;
49  //rounding of to 4 decimal values//
50  z= round (z);
51  z = z /10000;
52  d1=f+z;
53  //converting second binary input to its complement
    decimal number//
54  a=10011.1;
55  s=0;
56  d=modulo(a,1); //separating integer and decimal part
    //
57  //since complement of decimal part is 0,we do not
    consider it further//
58  a=floor(a); //getting the integer part//
59  for i=1:5
60      x(i)=modulo(a,10)
61      if x(i)==1 then
62          x(i)=0
63      else
64          x(i)=1
65      end
66      s=s+(x(i)*(2^(i-1)))
67      a=a/10
68      a=floor(a)
69  end
70  d2=s; //assigning the equivalent decimal value to d2
    //
71  d2=d2+.5; //since decimal is upto three places//
72  r=d1+d2;
73  r=r-bin2dec('100000'); //step to ignore the msb//
74  //converting the result to its binary equivalent//
75  q=0;
76  b=0;

```

```

77 s=0;
78 //enter the decimal number to be converted//
79 a=r;
80 d=modulo(a,1); //separating decimal and integer part
    //
81 a=floor(a); //removing decimal part//
82 while (a>0)
83 //taking integer part into a matrix and converting
    to equivalent binary//
84 x=modulo(a,2)
85 b=b+(10^q)*x
86 a=a/2
87 a=floor(a)
88 q=q+1
89     end
90 for i=1:10
91 //for values after decimal that should be converted
    to binary//
92 d=d*2
93 q=floor(d)
94 s=s+q/(10^i);
95 if d>=1 then
96     d=d-1
97     end
98 end
99 k=b+s;
100 disp('the subtraction of two binary numbers give:')
101 disp(k); //required result in binary form//

```

Scilab code Exa 1.23.a BCD to decimal equivalent conversion

```

1 //BCD to decimal conversion//
2 //Example 23.a//
3 //subtraction using 2's complement//
4 clc

```

```

5 //clears the console//
6 clear
7 //clears all existing variables//
8 a=010000111001;//taking the BCD input//
9 z=0;
10 d=modulo(a,10000)
11     for j=1:4
12         y(j)=modulo(d,10)
13         z=z+(y(j)*(2^(j-1)))
14         d=d/10
15         d=floor(d)
16     end
17 b=a/10000
18 b=floor(b)
19 c=modulo(b,10000)
20 z1=0
21 for j=1:2
22     y(j)=modulo(c,10)
23     z1=z1+(y(j)*(2^(j-1)))
24     c=c/10
25     c=floor(c)
26 end
27 e=b/10000
28 e=floor(e)
29 z2=0
30 for j=1:3
31     y(j)=modulo(e,10)
32     z2=z2+(y(j)*(2^(j-1)))
33     e=e/10
34     e=floor(e)
35 end
36 r=z2*100+z1*10+z
37 printf('the decimal equivalent of BCD code is=%d',r)

```

Scilab code Exa 1.23.b BCD to decimal equivalent conversion


```

1 //BCD to decimal conversion//
2 //Example 23.b//
3 //subtraction using 2's complement//
4 clc
5 //clears the console//
6 clear
7 //clears all existing variables//
8 a=100001001111000; //taking the BCD input//
9 z=0;
10 d=modulo(a,10000)
11     for j=1:4
12         y(j)=modulo(d,10)
13         z=z+(y(j)*(2^(j-1)))
14         d=d/10
15         d=floor(d)
16     end
17 b=a/10000
18 b=floor(b)
19 z1=0
20 c=modulo(b,10000)
21 for i=1:3
22     y(i)=modulo(c,10)
23     z1=z1+(y(i)*(2^(i-1)))
24     c=c/10
25     c=floor(c)
26 end
27 z2=0
28 e=b/10000
29 e=floor(e)
30 f=modulo(e,10000)
31 for i=1:2
32     y(i)=modulo(f,10)
33     z2=z2+(y(i)*(2^(i-1)))
34     f=f/10
35     f=floor(f)
36 end
37 g=e/10000
38 g=floor(g)

```

```

39 z3=0
40 for i=1:3
41     y(i)=modulo(g,10)
42     z3=z3+(y(i)*(2^(i-1)))
43     g=g/10
44     g=floor(g)
45 end
46 r=z3*1000+z2*100+z1*10+z
47 printf('the equivalent decimal number is=%d',r)

```

Scilab code Exa 1.24.a binary to gray code conversion

```

1 //binary to gray code//
2 //Example 24.a//
3 //binary to gray code//
4 clc
5 //clears the console//
6 clear
7 //clears all existing variables//
8 a=110100
9 for i=1:6
10     x(i)=modulo(a,10)
11     a=a/10
12     a=floor(a)
13 end
14 y(6)=x(6)
15 k=5
16 while(k>0)
17     if (bitand(x(k+1),x(k))==1) then
18         a=bitand(x(k+1),x(k))
19         y(k)=bitcmp(a,1)
20     else
21         y(k)=bitor(x(k+1),x(k))
22     end
23     k=k-1

```

```

24 end
25 //display//
26 z=0
27 for i=1:6
28     z=z+y(i)*(10^(i-1))
29 end
30 disp(z)
31 disp('equivalent gray code is displayed')

```

Scilab code Exa 1.24.b binary to gray code conversion

```

1 //binary to gray code//
2 //Example 24.b//
3 //binary to gray code//
4 clc
5 //clears the console//
6 clear
7 //clears all existing variables//
8 a=101101
9 for i=1:6
10     x(i)=modulo(a,10)
11     a=a/10
12     a=floor(a)
13 end
14 y(6)=x(6)
15 k=5
16 while(k>0)
17     if (bitand(x(k+1),x(k))=1) then
18         a=bitand(x(k+1),x(k))
19         y(k)=bitcmp(a,1)
20     else
21         y(k)=bitor(x(k+1),x(k))
22     end
23     k=k-1
24 end

```

```

25 //display//
26 z=0
27 for i=1:6
28     z=z+y(i)*(10^(i-1))
29 end
30 disp(z)
31 disp('equivalent gray code is displayed')

```

Scilab code Exa 1.24.c binary to gray code conversion

```

1 //binary to gray code//
2 //Example 24.c//
3 //binary to gray code//
4 clc
5 //clears the console//
6 clear
7 //clears all existing variables//
8 a=1110010
9 for i=1:7
10     x(i)=modulo(a,10)
11     a=a/10
12     a=floor(a)
13 end
14 y(7)=x(7)
15 k=6
16 while(k>0)
17     if (bitand(x(k+1),x(k))=1) then
18         a=bitand(x(k+1),x(k))
19         y(k)=bitcmp(a,1)
20     else
21         y(k)=bitor(x(k+1),x(k))
22     end
23     k=k-1
24 end
25 //display//

```

```

26 z=0
27 for i=1:7
28     z=z+y(i)*(10^(i-1))
29 end
30 disp(z)
31 disp('equivalent gray code is displayed')

```

Scilab code Exa 1.25.a subtraction using ones complement

```

1 //subtraction using 1's complement//
2 //Example 25.a//
3 //subtraction in one's complement //
4 clc
5 //clears the window//
6 clear
7 //clears all the existing variables//
8 x='1011101'
9 y=bin2dec('1011101')
10 z=bin2dec('1101100')
11 c=bitcmp(z,7); //finding 1's complement//
12 a=y+c+1
13 a=dec2bin(a-(bin2dec('1000000'))))
14 //binary conversion//
15 disp('binary form of the number obtained by adding
16     1011101 to -1101100')
16 disp(a)
17 //result is displayed//

```

Scilab code Exa 1.25.b gray code to binary conversion

```

1 //gray code to binary conversion//
2 //Example 25.b//
3 //gray code to binary//

```

```

4  clc
5  //clears the console//
6  clear
7  //clears all existing variables//
8  a=1011010111001
9  for i=1:13
10     x(i)=modulo(a,10)
11     a=a/10
12     a=floor(a)
13 end
14 y(13)=x(13)
15 k=12
16 while (k>0)
17     if (bitand(y(k+1),x(k))==1) then
18         a=bitand(y(k+1),x(k))
19         y(k)=bitcmp(a,1)
20     else
21         y(k)=bitor(y(k+1),x(k))
22     end
23     k=k-1
24 end
25 z=0
26 for i=1:13
27     z=z+y(i)*(10^(i-1))
28 end
29 disp(z)
30 disp('equivalent binary code is displayed')

```

Scilab code Exa 1.25.c decimal to BCD conversion

```

1  //decimal to BCD conversion//
2  //example 25.c//
3  clc
4  //clears the command window//
5  clear

```

```

6 //clears//
7 a=63547;//input in decimal//
8 for i=1:5
9     x(i)=modulo(a,10)
10    a=a/10
11    a=floor(a)
12 end
13 x=strcat(dec2bin(x(5),4)+dec2bin(x(4),4)+dec2bin(x(3),4)+dec2bin(x(2),4)+dec2bin(x(1),4))
14 disp('required BCD code is displayed')
15 disp(x)

```

Scilab code Exa 1.25.d subtraction using ones complement

```

1 //subtraction using 1's complement//
2 //Example 25.d//
3 //subtraction in one's complement//
4 clc
5 //clears the window//
6 clear
7 //clears all the existing variables//
8 x=85
9 y=32
10 c=bitcmp(y,7);//finding 1's complement//
11 a=x+c+1
12 a=a-bin2dec('10000000')
13 res=dec2bin(a,7)
14 //binary conversion//
15 disp('binary form of the number obtained by adding
16     85 to -32')
16 disp(res)
17 //result is displayed//

```

Scilab code Exa 1.26 finding equivalent in decimal for excess 3 code

```
1 //finding equivalent in decimal for five excess 3
   code numbers//
2 //example 26//
3 clc
4 //clears the command window//
5 clear
6 //clears//
7 //first number in excess -3 code//
8 x='1000 '
9 a=bin2dec('1000');//conversion to binary//
10 y=a-3;//conversion to excess 3 code//
11 disp('equivalent decimal number for first excess 3
   code:')
12 disp(y);//displaying result//
13 //second number in excess -3 code//
14 x='0011 '
15 a=bin2dec('0011');//conversion to binary//
16 y=a-3;//conversion to excess 3 code//
17 disp('equivalent decimal number for second excess 3
   code:')
18 disp(y);//displaying result//
19 //third number in excess 3-code//
20 x='0100 '
21 a=bin2dec('0100');//conversion to binary//
22 y=a-3;//conversion to excess 3 code//
23 disp('equivalent decimal number for third excess 3
   code:')
24 disp(y);//displaying result//
25 //fourth number in excess 3-code//
26 x='0101 '
27 a=bin2dec('0101');//conversion to binary//
28 y=a-3;//conversion to excess 3 code//
29 disp('equivalent decimal number for fourth excess 3
   code:')
30 disp(y);//displaying result//
31 //fifth number in excess 3-code//
```



```

32 x='0111'
33 a=bin2dec('0111');//conversion to binary//
34 y=a-3;//conversion to excess 3 code//
35 disp('equivalent decimal number for fifth excess 3
      code:')
36 disp(y);//displaying result//

```

Scilab code Exa 1.27.a gray code to binary conversion

```

1 //gray code to binary conversion//
2 //Example 27.a//
3 //gray code to binary//
4 clc
5 //clears the console//
6 clear
7 //clears all existing variables//
8 a=101110
9 for i=1:6
10     x(i)=modulo(a,10)
11     a=a/10
12     a=floor(a)
13 end
14 y(6)=x(6)
15 k=5
16 while(k>0)
17     if (bitand(y(k+1),x(k))==1) then
18         a=bitand(y(k+1),x(k))
19         y(k)=bitcmp(a,1)
20     else
21         y(k)=bitor(y(k+1),x(k))
22     end
23     k=k-1
24 end
25 z=0
26 for i=1:6

```

```

27     z=z+y(i)*(10^(i-1))
28 end
29 disp(z)
30 disp('equivalent binary code is displayed')

```

Scilab code Exa 1.27.b gray code to binary conversion

```

1 //gray code to binary conversion//
2 //Example 27.b//
3 //gray code to binary//
4 clc
5 //clears the console//
6 clear
7 //clears all existing variables//
8 a=111011
9 for i=1:6
10     x(i)=modulo(a,10)
11     a=a/10
12     a=floor(a)
13 end
14 y(6)=x(6)
15 k=5
16 while(k>0)
17     if (bitand(y(k+1),x(k))==1) then
18         a=bitand(y(k+1),x(k))
19         y(k)=bitcmp(a,1)
20     else
21         y(k)=bitor(y(k+1),x(k))
22     end
23     k=k-1
24 end
25 z=0
26 for i=1:6
27     z=z+y(i)*(10^(i-1))
28 end

```

```
29 disp(z)
30 disp('equivalent binary code is displayed')
```

Scilab code Exa 1.27.c gray code to binary conversion

```
1 //gray code to binary conversion//
2 //Example 27.c//
3 //gray code to binary//
4 clc
5 //clears the console//
6 clear
7 //clears all existing variables//
8 a=1101110
9 for i=1:7
10     x(i)=modulo(a,10)
11     a=a/10
12     a=floor(a)
13 end
14 y(7)=x(7)
15 k=6
16 while(k>0)
17     if (bitand(y(k+1),x(k))==1) then
18         a=bitand(y(k+1),x(k))
19         y(k)=bitcmp(a,1)
20     else
21         y(k)=bitor(y(k+1),x(k))
22     end
23     k=k-1
24 end
25 z=0
26 for i=1:7
27     z=z+y(i)*(10^(i-1))
28 end
29 disp(z)
30 disp('equivalent binary code is displayed')
```

Scilab code Exa 1.28.a constructing an even parity 7 bit hamming code

```
1 //constructing an even parity 7 bit hamming code//
2 //example 28.a//
3 clc
4 //clears the command window//
5 clear
6 //clears//
7 z=0100; //input//
8 a=0;b=0;c=0;d=0;
9 //taking the input//
10 for i=1:7
11     x(i)=0
12     if (i==5) then
13         x(i)=1
14     end
15 end
16 //establishing even parity at positions 1,3,5,7//
17 for i=3:7
18     if (x(i)==1) then
19         a=a+1
20     end
21 end
22 d=modulo(a,2)
23 if (d==1) then
24     x(1)=1
25 end
26 //establishing even parity at positions 2,3,6,7//
27 for i=3:7
28     if (i==5) then
29         continue
30     end
31     if (x(i)==1) then
32         b=b+1
```

```

33     end
34 end
35 d=modulo(b,2)
36 if (d==1) then
37     x(2)=1
38 end
39 //establishing even parity at positions 4,5,6,7//
40 for i=5:7
41     if (x(i)==1) then
42         c=c+1
43     end
44 end
45 d=modulo(c,2)
46 if (d==1) then
47     x(4)=1
48 end
49 //displaying the result//
50 disp('the required hamming code is:')
51 for i=1:7
52     printf('%d',x(i))
53 end

```

Scilab code Exa 1.28.b constructing an even parity 7 bit hamming code

```

1 //constructing an even parity 7 bit hamming code//
2 //example 28.b//
3 clc
4 //clears the command window//
5 clear
6 //clears//
7 z=1110; //input//
8 a=0;b=0;c=0;d=0;
9 //taking the input//
10 for i=1:7
11     x(i)=0

```

```

12     if (i==3) then
13         x(i)=1
14     end
15     if (i==5) then
16         x(i)=1
17     end
18     if (i==6) then
19         x(i)=1
20     end
21 end
22 //establishing even parity at positions 1,3,5,7//
23 for i=3:7
24     if (i==6) then
25         continue
26     end
27     if (x(i)==1) then
28         a=a+1
29     end
30 end
31 d=modulo(a,2)
32 if (d==1) then
33     x(1)=1
34 end
35 //establishing even parity at positions 2,3,6,7//
36 for i=3:7
37     if (i==5) then
38         continue
39     end
40     if (x(i)==1) then
41         b=b+1
42     end
43 end
44 d=modulo(b,2)
45 if (d==1) then
46     x(2)=1
47 end
48 //establishing even parity at positions 4,5,6,7//
49 for i=5:7

```

```

50     if (x(i)==1) then
51         c=c+1
52     end
53 end
54 d=modulo(c,2)
55 if (d==1) then
56     x(4)=1
57 end
58 //displaying the result//
59 disp('the required hamming code is:')
60 for i=1:7
61     printf('%d',x(i))
62 end

```

Scilab code Exa 1.29 locating error in hamming code and correcting it

```

1 //locating error in hamming code and correcting it//
2 //example 29//
3 clc
4 //clears the command window//
5 clear
6 //clears//
7 a=1111001
8   b=0;c=0;d=0;e=0;
9   for i=1:7
10      x(8-i)=modulo(a,10)
11      a=a/10
12      a=floor(a)
13 end
14 //checking even parity at positions 4,5,6,7//
15 for i=4:7
16     if (x(i)==1) then
17         b=b+1
18     end
19 end

```

```

20 d=modulo(b,2)
21 if (d==0) then
22     r1=0
23 else
24     r1=1
25 end
26 //checking even parity at positions 2,3,6,7//
27 for i=2:7
28     if (i==4) then
29         continue
30     end
31     if (i==5) then
32         continue
33     end
34     if (x(i)==1) then
35         c=c+1
36     end
37 end
38 d=modulo(c,2)
39 if (d==0) then
40     r2=0
41 else
42     r2=1
43 end
44 //checking even parity at positions 1,3,5,7//
45 for i=1:7
46     if (i==2) then
47         continue
48     end
49     if (i==4) then
50         continue
51     end
52     if (i==6) then
53         continue
54     end
55     if (x(i)==1) then
56         e=e+1
57     end

```



```

58 end
59 d=modulo(e,2)
60 if (d==0) then
61     r3=0
62 else
63     r3=1
64 end
65 r=r1*(2^2)+r2*(2^1)+r3;//converting to equivalent
    decimal//
66 printf('error occurs at position=%d\n',r)
67 //bit position at which error occurs has to be
    changed//
68 if (x(r)==1) then
69     x(r)=0
70 else
71     x(r)=1
72 end
73 //displaying the correct hamming code//
74 disp('the correct hamming code is:')
75 for i=1:7
76     printf('%d',x(i))
77 end

```

Chapter 2

Boolean algebra logic gates karnaugh map and minimization

Scilab code Exa 2.8.a simplification of boolean expression

```
1 //simplification of boolean expression//
2 //example 8.a//
3 clc
4 //clears the command window//
5 clear
6 //clears//
7 disp('given Y=AB+AB''C(B''C''+C)+A''C''')
8 disp('Y=AB+AB''B''CC''+AB''CC+A''C'''); //ON
    MULTIPLICATION//
9 disp('Y=AB+AB''C+A'''+C'''); //CC''=0//
10 disp('Y=C'''+AB+(A+A''')(A'''+B''C)')
11 disp('Y=C'''+AB+1.(A'''+B''C)')
12 disp('Y=A'''+A.B+C'''+B''C')
13 disp('Y=(A'''+A).(A'''+B)+(C'''+B''').(C'''+C)')
14 disp('Y=1.(A'''+B)+(C'''+B''').1')
15 disp('the simplified expression is:')
16 disp('Y=A'''+B+C'''+B'''); //required simplified
```

expression//

Scilab code Exa 2.8.b simplification of boolean expression

```
1 //simplification of boolean expression//
2 //example 8.b//
3 clc
4 //clears the command window//
5 clear
6 //clears//
7 disp('given Y=A'BC'+A'BC+AB'C'+ABC')
8 disp('Y=A'BC'+A'BC+A'BC+AB'C'+ABC')
9 disp('Y=A'B(C+C')+A'BC+AB'C+AB'C')
10 disp('Y=A'B.1+BC(A+A')+AB'C'); //C+C'=1//
11 disp('Y=A'B+BC+AB'C'); //A+A'=1//
12 disp('the simplified expression is:')
13 disp('Y=B.(A'+C)+AB'C'); //required simplified
    expression//
```

Scilab code Exa 2.9.a simplification of boolean expression

```
1 //simplification of boolean expression//
2 //example 9.a//
3 clc
4 //clears the command window//
5 clear
6 //clears//
7 //simplification of boolean expression//
8 disp('given Y=ABC'D'+ABC'D+ABCD'+ABCD')
9 disp('since D+D'=1,we get')
10 disp('Y=ABC'(D+D')+ABC(D'+D)')
11 disp('Y=ABC'+ABC')
12 disp('Y=AB(C+C)')
```

```
13 disp('the simplified expression is:')
14 disp('Y=AB'); //C+C'=1//
15 //answer after simplification//
```

Scilab code Exa 2.9.b simplification of boolean expression

```
1 //simplification of boolean expression//
2 //example 9.b//
3 clc
4 //clears the command window//
5 clear
6 //clears//
7 //simplification of boolean expression//
8 disp('given Y=AB+ABC''+A''BC+ABC')
9 disp('on simplification we get')
10 disp('Y=AB+ABC''+A''BC+ABC+A''BC')
11 disp('Y=AB+AB(C+C''')+BC(A+A''')')
12 disp('since x+x''=1,we get')
13 disp('Y=AB+AB+BC')
14 disp('Y=AB+BC')
15 disp('the simplified expression is:')
16 disp('Y=B(A+C)')
17 //answer after simplification//
```

Scilab code Exa 2.9.c simplification of boolean expression

```
1 //simplification of boolean expressions//
2 //example 9.c//
3 clc
4 //clears the command window//
5 clear
6 //clears//
7 //simplification of boolean expression//
```

```

8 disp('given Y=AB(A''BC'''+AB''C'''+A''BC)')
9 disp('on multiplication')
10 disp('Y=ABA''BC'''+ABAB''C'''+ABA''BC')
11 disp('Y=AA'''.BC'''+AB'''.B''C'''+AA'''.BC')
12 disp('since x.x''=0')
13 disp('Y=0.BC'''+A.0.C'''+0.BC')
14 disp('Y=0')
15 //answer after simplification//

```

Scilab code Exa 2.10.a simplification of boolean expression

```

1 //simplification of boolean expressions//
2 //example 10.a//
3 clc
4 //clears the command window//
5 clear
6 //clears//
7 //simplification of boolean expression//
8 disp('given Y=(A+B)'''.(A'''+B''')''')
9 disp('Y=(A'''.B''').(A.B)''); //applying De Morgan's law
   //
10 disp('Y=AA'''.BB''')
11 disp('the simplified expression is:')
12 disp('Y=0'); //since x.x''=0//
13 //simplified boolean expression//

```

Scilab code Exa 2.10.b simplification of boolean expression

```

1 //simplification of boolean expressions//
2 //example 10.b//
3 clc
4 //clears the command window//
5 clear

```

```

6 //clears//
7 //simplification of boolean expression//
8 disp('Y=ABC+A''B+ABC''')
9 disp('Y=AB(C+C''')+A''B')
10 disp('Y=B(A+A''')'); //C+C''=1//
11 disp('the simplified expression is:')
12 disp('Y=B'); //simplified expression//

```

Scilab code Exa 2.10.c simplification of boolean expression

```

1 //simplification of boolean expressions//
2 //example 10.c//
3 clc
4 //clears the command window//
5 clear
6 //clears//
7 //simplification of boolean expression//
8 disp('given Y=C(AB''+AB)+BC')
9 disp('Y=CA(B''+B)+BC')
10 disp('Y=CA+BC'); //B+B''=1//
11 disp('the simplified expression is:')
12 disp('Y=C(A+B)')
13 //simplified boolean expression//

```

Scilab code Exa 2.11 simplification of boolean expression

```

1 //simplification of boolean expressions//
2 //example 11//
3 clc
4 //clears the command window//
5 clear
6 //clears//
7 //simplification of boolean expression//

```

```

8 disp('given Y=ABCD+ABCD''+A''BCD+A''BCD''')
9 disp('Y=BC(AD+AD''+A''D+A''D''')')
10 disp('Y=BC{A(D+D''')+A''(D+D''')}')
11 disp('Y=BC(A.1+A'''.1)')
12 disp('BC(A+A''')')
13 disp('the simplified expression is:')
14 disp('Y=BC'); //simplified boolean expression//

```

Scilab code Exa 2.13.a simplification of boolean expression

```

1 //simplification of boolean expressions//
2 //example 13.a//
3 clc
4 //clears the command window//
5 clear
6 //clears//
7 disp('given Z=(AB''C''+AB''C+ABC+ABC''')(A+B)')
8 disp('Z={}AB''(C+C''')+AB(C+C''')}(A+B)')
9 disp('Z=(AB''+AB)(A+B)')
10 disp('Z=A(B''+B)(A+B)')
11 disp('Z=A.1.(A+B)')
12 disp('Z=AA+AB')
13 disp('Z=A+AB')
14 disp('Z=A(1+B)')
15 disp('the simplified expression is:')
16 disp('A'); //simplified boolean expression//

```

Scilab code Exa 2.13.b simplification of boolean expression

```

1 //simplification of boolean expressions//
2 //example 13.b//
3 clc
4 //clears the command window//

```

```

5 clear
6 //clears//
7 disp('given A=P+P'QR'+(Q+R)')
8 disp('A=P+P'QR'+Q'.R'); //DE MORGAN'S LAW//
9 disp('A=P+R'(P'Q+Q'))
10 disp('A=P+R'[P'Q+Q'(P+P)])
11 disp('A=P+R'[P'(Q+Q')+PQ'])
12 disp('A=P+R'[P'+PQ])
13 disp('A=P+R'{(P'+P).(P'+Q)})
14 disp('A=P+R'(P'+Q))
15 disp('A=P+P'R'+Q'R')
16 disp('A=(P+P).(P'+R')+Q'R')
17 disp('A=1.(P+R')+Q'R')
18 disp('A=P+R'+Q'R')
19 disp('A=P+R'+Q'R')
20 disp('A=P+R'(1+Q)')
21 disp('the simplified expression is:')
22 disp('A=P+R')

```

Scilab code Exa 2.29 finding SOP

```

1 //finding SOP//
2 //example 29//
3 clc
4 //clears the command window//
5 clear
6 //clears//
7 disp('f=summation(1,2,4,7)')
8 disp('f=001+0101+100+111')
9 disp('required SOP form:')
10 disp('f=A'B'C+A'BC'+AB'C'+ABC');

```

Scilab code Exa 2.30 finding POS


```

1 //finding POS//
2 //example 30//
3 clc
4 //clears the command window//
5 clear
6 //clears//
7 disp('given f=product(0,3,5,6)')
8 disp('this indicates (000)(011)(101)(110)')
9 disp('f=(A+B+C)(A+B'+C')(A'+B+C')(A'+B'+C)');
    //required POS expression//

```

Scilab code Exa 2.31.a conversion to canonical SOP

```

1 //conversion to canonical SOP//
2 //example 31.a//
3 clc
4 //clears the command window//
5 clear
6 //clears//
7 //conversion to canonical SOP//
8 disp('given Y=AB+A'B'+AC+A'C')
9 disp('since x+x'=1,we get')
10 disp('Y=AB(C+C')+A'B'(C+C')+AC(B+B')+A'C'(B+B')')
11 disp('Y=ABC+ABC'+A'B'C+A'B'C'+ACB'+ACB'+A'C'B+A'C'B')')
12 disp('since ABC and A'B'C' appear twice,we write them once')
13 disp('Y=ABC+ABC'+A'B'C+A'B'C'+AB'C+A'BC')')
14 disp('the required SOP expression')
15 disp('Y=111+110+001+000+101+010')
16 disp('hence the required entries will be at the positions 0,1,2,5,6,7')
17 disp('f=summation(0,1,2,5,6,7)')

```

Scilab code Exa 2.31.b conversion to canonical POS

```
1 //conversion to canonical POS//
2 //example 31.b//
3 clc
4 //clears the command window//
5 clear
6 //clears//
7 //conversion to canonical POS//
8 disp('given  $Y=AB+A'B'+AC+A'C'$ ')
9 disp('we use the complement of the previous example'
    )
10 disp('the entries will be at 3,4 i.e. 011,100
    positions')
11 disp('Y= $A'BC+AB'C'$ ')
12 disp('Y= $(A'BC.AB'C')$ ')
13 //de morgan's theorem//
14 disp('Y= $(A+B'+C')$ .(A'+B+C)')
15 disp('the required POS expression')
```

Scilab code Exa 2.32 finding SOP and POS

```
1 //finding SOP and POS//
2 //example 32//
3 clc
4 //clears the command window//
5 clear
6 //clears//
7 disp('given  $f=A'.(B+C)'$ ')
8 disp('f= $(A'+BB'+CC')(B+C'+AA)'$ ')
9 disp('f= $[(A'+B)(A'+B')+CC'][(B+C'+A)(B+C'+A)'$ 
    ]')
```

```

10 disp('f=[C+(A''+B)(A''+B'')][C''+(A''+B)(A''+B)][(A+
    B+C'')(A''+B+C'')')
11 disp('f=(C+A''+B)(C+A''+B'')(C''+A''+B)(C''+A''+B'')
    (A+B+C'')(A''+B+C'')')
12 disp('f=(A''+B+C)(A''+B''+C'')(A''+B+C'')(A''+B''+C'
    ')(A+B+C'')')
13 disp('f=(100)(110)(101)(111)(001)')
14 disp('required POS form is:')
15 disp('f=product(1,4,5,6,7)'); //required POS form//
16 disp('f=A''(B+C'')')
17 disp('f=A''B+A''C'')')
18 disp('f=A''B(C+C'')+A''C''(B+B'')')
19 disp('f=A''BC+A''BC''+A''BC''+A''B''C''')
20 disp('f=011+010+000')
21 disp('required SOP form is:')
22 disp('f=summation(0,2,3)'); //required SOP form//

```

Scilab code Exa 2.33 finding SOP and POS

```

1 //finding SOP and POS//
2 //example 33//
3 clc
4 //clears the command window//
5 clear
6 //clears//
7 disp('given f=B''C')
8 disp('f=(B''+AA''+CC'')(C+AA''+BB'')')
9 disp('f=((B''+A)(B''+A'')+CC'')((C+A)(C+A'')+BB'')')
10 disp('f=[C+(B''+A)(B''+A'')][C''+(B''+A)(B''+A'')][B
    +(C+A)(C+A'')][B''+(C+A)+(C+A'')]'); //using
    distributive property//
11 disp('f=(A+B''+C)(A''+B''+C)(A+B''+C'')(A''+B''+C'')
    (A+B+C)(A''+B+C)'); //using distributive property
    and retaining repeated factors only once//
12 disp('f=(010)(110)(011)(111)(000)(100)')

```

```

13 disp('required POS form:')
14 disp('f=product(0,2,3,4,6,7)')
15 //finding SOP//
16 disp('f=(A+A')'.B'C')
17 disp('f=AB'C+A'BC')
18 disp('f=101+001')
19 disp('required SOP form:')
20 disp('f=summation(5,1)')

```

Scilab code Exa 2.34 finding maxterms and minterms

```

1 //finding maxterms and minterms//
2 //example 34//
3 clc
4 //clears the command window//
5 clear
6 //clears//
7 //finding minterms and maxterms//
8 disp('given Y=AC'+A'C')
9 disp('Y=A'C(B+B')+AC'(B+B)')
10 disp('Y=A'CB+A'CB'+AC'B+AC'B')
11 disp('these are the required minterms')
12 disp('the minterms indicate the positions 1,3,4,6')
13 disp('the maxterms indicate the positions 0,2,5,7')
14 //complementary of minterms//
15 disp('Y=(A+B+C)(A+B'+C)+(A'+B+C')(A'+B'+C)')
16 disp('these are the required maxterms')

```

Scilab code Exa 2.35 finding maxterms

```

1 //finding maxterms//
2 //example 35//
3 clc

```

```

4 //clears the command window//
5 clear
6 //clears//
7 disp('given f=product(0,3,5,6)')
8 disp('f=(000)(011)(101)(110)')
9 disp('required maxterms:')
10 disp('f=(A+B+C)(A+B'+C')(A'+B+C')(A'+B'+C)')

```

Scilab code Exa 2.36 finding required data

```

1 //finding required data//
2 //example 36//
3 clc
4 //clears the command window//
5 clear
6 //clears//
7 disp('f=A'B'C+A'BC'+AB'C+ABC'); //since f is 1
    at positions 1,2,5,6;this is the required SOP
    expression//
8 disp('f=(A+B+C)(A+B'+C')(A'+B+C)(A'+B'+C)');
    //since f is 0 at 0,3,4,7;this is the required
    POS expression//
9 disp('sum of minterms=summation(1,2,5,6)')
10 disp('product of maxterms=product(0,3,4,7)')

```

Scilab code Exa 2.37 finding SOP

```

1 //finding SOP//
2 //example 37//
3 clc
4 //clears the command window//
5 clear
6 //clears//

```

```

7 disp('given f=AC')
8 disp('f=AC(B+B''')(D+D''')')
9 disp('f=ACBD+ACBD'''+ACB''D+ACB''D'''); //required sum
    of minterms//
10 disp('f=1111+1110+1011+1010')
11 disp('required SOP form:')
12 disp('f=summation(10,11,14,15)')

```

Scilab code Exa 2.38 finding sum of minterms and product of maxterms

```

1 //sum of minterms and product of maxterms//
2 //example 38//
3 clc
4 //clears the command window//
5 clear
6 //clears//
7 disp('given f=AB+A''BC+C''D')
8 disp('f=AB(C+C''')(D+D''')+AB''C(D+D''')+C''D(A+A''')(B+
    B''')')
9 disp('required sum of minterms:')
10 disp('f=ABCD+ABCD'''+ABC''D+ABC''D'''+AB''CD+AB''CD'''+
    AB''C''D+A''BC''D+A''B''C''D')
11 disp('f=summation(1,5,9,10,11,12,13,14,15)')
12 disp('therefore f=product(0,2,3,4,6,7,8)')
13 disp('required product of maxterms:')
14 disp('f=(A+B+C+D)(A+B+C'''+D)(A+B+C'''+D''')(A+B+C'''+D'
    ''')(A+B'''+C+D)(A+B'''+C'''+D)(A+B'''+C'''+D''')(A'''+B+C
    +D)')

```

Chapter 3

Wave shaping clipping clamping and sweep circuits

Scilab code Exa 3.1 finding required data

```
1 //finding required data//
2 //example 1//
3 clc
4 //clears the command window//
5 clear
6 //clears//
7 //finding time constant and time for capacitor to
  charge 90% of supplied voltage//
8 R=106 ;//resistance in ohms//
9 C=10-5 ;//capacitance in farads//
10 T=R*C
11 printf('the time constant=%f seconds\n',T)
12 //time constant is found out//
13 v=90/100*10;//v=voltage at time t//
14 V=10;//voltage in volts//
15 //t=required time//
16 disp('from the formula  $v=V*(1-\exp(-t/(R*C)))$ , we get
  the required time as:')
17 t=-((R*C)*log(1-(v/V)))
```

```
18 printf('required time to charge to 90 percent of the
    supplied voltage=%f seconds',t)
19 //the result is t seconds//
```

Scilab code Exa 3.2 finding maximum amplification

```
1 //finding minimum amplification//
2 //example 2//
3 clc
4 //clears the command window//
5 clear
6 //clears//
7 r=.04*10^-6
8 //r is the rise time//
9 disp('rise time=(2*v)/(2*pi*f*V), hence ')
10 //f=frequency in hetrz;v=output voltage in volts;V=
    required maximum output voltage in volts//
11 f=400*10^3
12 v=6.28
13 V=(2*v)/(r*2*pi*f)
14 Vin=12.5;//input voltage//
15 a=V/Vin
16 printf('minimum amplification required=%f ',a)
```

Scilab code Exa 3.3.a finding clamping level voltage

```
1 //finding clamping level voltage//
2 //example 3.a//
3 clc
4 //clears the command window//
5 clear
6 //clears//
7 //charge=voltage/time//
```



```

8 disp('from equation  $((V*4000*10^{-6})/20)=((5-V)$ 
       $*4*10^{-6})/(20*10^{-3})$ ,we get ')
9 //resistance=20kohm;V=clamping level voltage;charge=
  voltage*capacitance//
10 V=5*(10-6)
11 printf('clamping level voltage=%f volt ',V)
12 //the answer is displayed//

```

Scilab code Exa 3.3.b finding clamping level voltage

```

1 //finding clamping level voltage//
2 //example 3.b//
3 clc
4 //clears the command window//
5 clear
6 //clears//
7 //charge=voltage/time//
8 disp('from equation  $((V*4*10^{-6})/20)=((5-V)$ 
       $*4000*10^{-6})/(20*10^{-3})$ ,we get ')
9 //resistance=20kohm;V=clamping level voltage;charge=
  voltage*capacitance//
10 V=20/8
11 printf('clamping level voltage=%f volt ',V)
12 //the answer is displayed//

```

Scilab code Exa 3.4 finding output voltage

```

1 //finding output voltage//
2 //example 4//
3 clc
4 //clears the command window//
5 clear
6 //clears//

```

```

7 V1=10; //voltage source amplitude//
8 Rf=50; //forward rsistance//
9 r1=50; //internal resistance//
10 i=V1*Rf/(Rf+r1)
11 //current through diode during positive half cycle//
12 C=10^-6
13 T=(Rf+r1)*C; //time constant during conduction//
14 f=10*10^3
15 T1=1/(2*f); //time duration during which input
    voltage is positive//
16 t=50*10^-6; //given time//
17 v=V1*(1-exp(-(t/T))); //voltage across capacitor
    after 50micro sec//
18 Vo=((V1-v)*Rf)/(Rf+r1); //output voltage across diode
    //
19 R=20*10^3; //resistance in series//
20 T2=(r1+R)*C; //time constant of discharging circuit//
21 V2=v*exp(-(t/T2)); //voltage at the end of 50 micro
    sec//
22 printf('output voltage=%f volt ',V2)
23 //result is displayed//

```

Scilab code Exa 3.6 finding resistance and capacitance

```

1 //finding load resistance and value of capacitor//
2 //example 6//
3 clc
4 //clears the command window//
5 clear
6 //clears//
7 Vo=15; //output voltage//
8 t=10*10^-3; //time duration//
9 tr=.05*10^-6; //rise time//
10 Rs=100*10^3; //source resistance//
11 C=50*10^-12; //output capacitance//

```

```

12 S=.05; //sag allowed//
13 //tr=.35/f2 where f2=upper cut off frequency//
14 //also tr=2.2*R*C where R=load resistance//
15 R=tr/(2.2*C)
16 printf('resistance=%f ohms\n',R)
17 //result is displayed//
18 Cs=t/(Rs*S); //Cs=source capacitor//
19 printf('capacitance=%f farad ',Cs); //result is
    displayed//

```

Scilab code Exa 3.7 finding peak voltage

```

1 //finding peak voltage//
2 //example 7//
3 clc
4 //clears the command window//
5 clear
6 //clears//
7 T=250*10^-12; //time constant//
8 t=.05*10^-6;
9 Vo=100; //peak output voltage//
10 Vi=(Vo*t)/T; //peak voltage of generator//
11 printf('peak voltage=%f volt ',Vi)
12 //result is displayed//

```

Chapter 5

Sequential circuits flip flops and multivibrators

Scilab code Exa 5.1 finding required time

```
1 //finding required time//
2 //example 1//
3 clc
4 //clears the command window//
5 clear
6 //clears//
7 R=10*103; //external resistance//
8 C=0.1*10-6; //timing capacitor//
9 T=1.1*R*C; //time for which output remains high in
    monostable multivibrator using IC 555 timer//
10 printf('required time=%f seconds',T); //result is
    displayed
```

Scilab code Exa 5.2 finding frequency and duty cycle

```
1 //calculating frequency and duty cycle//
```

```

2 //example 2//
3 clc
4 //clears the command window//
5 clear
6 //clears//
7 Ra=10*10^3
8 Rb=10*10^3
9 C=.001*10^-6
10 T1=.693*(Ra+Rb)*C;//charging time constant//
11 T2=.693*Rb*C;//discharging time constant//
12 T=T1+T2;//total time//
13 f=1/T;//frequency//
14 printf('frequency=%f hertz\n',f);//result is
    displayed//
15 D=T2/T;//duty cycle//
16 D=D*100
17 printf('duty cycle=%f percent',D)
18 //result is displayed//

```

Scilab code Exa 5.3 finding frequency and duty cycle

```

1 //calculating frequency and duty cycle//
2 //example 3//
3 clc
4 //clears the command window//
5 clear
6 //clears//
7 Ra=6.8*10^3
8 Rb=3.3*10^3
9 C=.1*10^-6
10 T1=.693*(Ra+Rb)*C;//charging time constant//
11 T2=.693*Rb*C;//discharging time constant//
12 T=T1+T2;//total time//
13 f=1/T//frequency//
14 printf('frequency=%f hertz\n',f);//result is

```

```
    displayed//  
15 D=T2/T; //duty cycle//  
16 D=D*100  
17 printf('duty cycle=%f percent ',D); //result is  
    displayed//
```

Chapter 8

D to A and A to D converters

Scilab code Exa 8.1 finding binary equivalent weights

```
1 //finding binary equivalent weights//
2 //example 1//
3 clc
4 //clears the command window//
5 clear
6 //clears//
7 n=6; //no. of bits//
8 LSB1=1/((2^6)-1)
9 printf('weight of 1st lsb=%f\n',LSB1)
10 LSB2=2*LSB1
11 printf('weight of 2nd lsb=%f\n',LSB2)
12 LSB3=4*LSB1
13 printf('weight of 3rd lsb=%f\n',LSB3)
14 LSB4=8*LSB1
15 printf('weight of 4th lsb=%f\n',LSB4)
16 LSB5=16*LSB1
17 printf('weight of 5th lsb=%f\n',LSB5)
18 MSB=32*LSB1
19 printf('weight of msb=%f',MSB)
20 //results are displayed//
```

Scilab code Exa 8.2 calculating required data

```
1 //calculating the required data//
2 //example 2//
3 clc
4 //clears the command window//
5 clear
6 //clears//
7 n=4;
8 WLSB1=1/(2^4-1); //weight assigned to LSB//
9 printf('weight assigned to LSB=%f\n',WLSB1)
10 WLSB2=2*WLSB1; //weight assigned to 2nd LSB//
11 printf('weight assigned to second LSB=%f\n',WLSB2)
12 WLSB3=4*WLSB1; //weight assigned to 3rd LSB//
13 printf('weight assigned to 3rd LSB=%f\n',WLSB3)
14 //high level 1=10V//
15 V1=10*WLSB1; //change in output voltage due to change
    in LSB//
16 V2=10*8/15; //change in output voltage due to change
    in MSB//
17 printf('change in output voltage due to change in
    LSB=%f volt\n',V1)
18 printf('change in output voltage due to change in
    MSB=%f volt\n',V2)
19 //results are displayed//
20 disp('Vo1=(20/R+40/R+80/R)/(1/R+2/R+4/R+8/R)')
21 Vo1=140/15;
22 printf('output voltage for input 1110=%f volt\n',Vo1
    )
23 disp('Vo2=(10/R+40/R+80/R)/(1/R+2/R+4/R+8/R)')
24 Vo2=130/15;
25 printf('output voltage for input 1101=%f volt',Vo2)
26 //results are displayed//
```

Scilab code Exa 8.3 calculating required data

```
1 //calculating the required data//
2 //example 2//
3 clc
4 //clears the command window//
5 clear
6 //clears//
7 disp('the full scale maximum voltage occurs when all
      input are at high level 1')
8 Vm=10;//full scale maximum output voltage//
9 printf('full scale maximum output voltage=%d volt\n',
      ,Vm);//result is displayed//
10 n=5
11 LSB=1/((2^n)-1);//weight of LSB//
12 V1=10*LSB;//change in output voltage due to LSB//
13 printf('change in output voltage due to change in
      lsb=%f volt\n',V1)
14 disp('analog output voltage for 11001 is given by Vo
      =(10/R+80/R+160/R)/(1/R+2/R+4/R+8/R+16/R)')
15 Vo=250/31;
16 printf('analog output voltage for input 11001=%f
      volt\n',Vo)
```

Scilab code Exa 8.4 finding required data

```
1 //calculating required data//
2 //example 4//
3 clc
4 //clears the command window//
5 clear
6 //clears//
```

```

7 a=bin2dec('10010')
8 Vo=9*10^-3;//output voltage//
9 k=Vo/a;//proportionality factor//
10 b=bin2dec('11011')
11 Vo1=k*b;//analog output voltage//
12 printf('analog output voltage=%f volt\n',Vo1);
13 //maximum 5 bit digital input is 11111 i.e. 31 in
    decimal//
14 Vm=k*31;//maximum full scale output voltage//
15 printf('maximum full scale output voltage=%f volt',
    Vm);

```

Scilab code Exa 8.5 calculating required data

```

1 //calculating required data//
2 //example 5//
3 clc
4 //clears the command window//;
5 clear
6 //clears//
7 LSB1=10;//change in output voltage due to LSB//
8 LSB2=2*LSB1;//change in output voltage due to second
    LSB//
9 LSB3=4*LSB1;//change in output voltage due to third
    LSB//
10 LSB4=8*LSB1;//change in output voltage due to fourth
    LSB//
11 LSB5=16*LSB1;//change in output voltage due to fifth
    LSB//
12 Vmax=LSB1+LSB2+LSB3+LSB4+LSB5;//maximum full scale
    output which occurs at input 11111//
13 //all the voltage units are in mV//
14 printf('maximum full scale output in mV at 1111=%d
    volt\n',Vmax);//displaying the result//
15 R=10;//resolution in mV since smallest increment due

```

```

    to change in LSB is 10mV//
16 printf('resolution=%d volt\n',R); //diaplaying the
    result//
17 S=10; //step size=resolution//
18 R1=S/Vmax*100; //percentage resolution//
19 printf('percentage resolution=%f percent ',R1); //
    displaying the result//

```

Scilab code Exa 8.6 calculating required data

```

1 //calculating required data//
2 //example 6//
3 clc
4 //clears the command window//;
5 clear
6 //clears//
7 N=(2^8)-1; //no. of steps//
8 M=1000/N; //no. of steps in which motor speed can
    varied; motor speed varies from 0 to 1000rpm//
9 N1=450/M; //no. of steps required to reach 450rpm//
10 N2=round(N1); //rounding the no. of steps//
11 disp('For 115th step, speed of motor will be
    approximately 451rpm which gives an accuracy of
    .22%')
12 //displaying the answer//

```

Scilab code Exa 8.7 calculating value of resistors

```

1 //calculating values of resistors//
2 //example 7//
3 clc
4 //clears the command window//;
5 clear

```

```

6 //clears//
7 R1=10*103; //resistor in ohm for msb//
8 R=16*R1; //resistor for LSB//
9 R2=R/2; //resistor for 2nd LSB//
10 printf('resistor for lsb=%d ohm\n',R)
11 printf('resistor for 2nd lsb=%d ohm',R2)

```

Scilab code Exa 8.8 finding value of resistor

```

1 //calculating value of resistor//
2 //example 8//
3 clc
4 //clears the command window//;
5 clear
6 //clears//
7 R=640*(103); //resistor for LSB//
8 R1=R/32; //resistor for MSB; since 6 bit, hence LSB is
   divided by 32//
9 printf('resistor for msb=%d ohm',R1); //displaying
   result//

```

Scilab code Exa 8.9 finding required data

```

1 //calculating required data//
2 //example 9//
3 clc
4 //clears the command window//;
5 clear
6 //clears//
7 N=210-1; //no. of steps//
8 Vmax=10*10-3*N; //maximum full scale output//
9 R=(10*10-3)/Vmax*100; //percentage resolution//
10 printf('maximum full scale output=%f ampere\n',Vmax)

```

```
11 printf('percentage resolution=%f percent',R)
```

Scilab code Exa 8.10 finding required data

```
1 //calculating required data//
2 //example 10//
3 clc
4 //clears the command window//;
5 clear
6 //clears//
7 S=10/10^3; //step size=weight of Ao in V//
8 printf('step size=%f volt\n',S); //displaying step
   size//
9 n=99; //no. of steps//
10 Vmax=S*n; //maximum full scale output//
11 printf('maximum full scale output=%f volt\n',Vmax);
   //displaying full scale output//
12 R=S/Vmax*100; //percentage resolution//
13 printf('percentage resolution=%f percent\n',R); //
   percentage resolution displayed//
14 Ao=10/10^3
15 Bo=20/10^3
16 Co=40/10^3
17 Do=80/10^3
18 A1=100/10^3
19 B1=200/10^3
20 C1=400/10^3
21 D1=800/10^3
22 //weight in V of different inputs//
23 //input for D1,C1,B1,A1 is 1101 and for Do,Co,Bo,Ao
   is 1011//
24 V=D1+C1+A1+Do+Bo+Ao; //analog output voltage//
25 printf('analog output voltage=%f volt',V); //result
   displayed//
```

Scilab code Exa 8.11 calculating output voltages

```
1 //calculating output voltages//
2 //example 11//
3 clc
4 //clears the command window//
5 clear
6 //clears//
7 V=10;//high input voltage//
8 VMSB=V/2;//output caused by MSB//
9 VMSB2=V/4;//output caused by second MSB//
10 VMSB3=V/8;//output caused by third MSB//
11 VMSB4=V/16;//output caused by fourth MSB//
12 VMSB5=V/32;//output caused by fifth MSB//
13 VMSB6=V/64;//output caused by sixth MSB//
14 printf('output caused by MSB=%f volt\n',VMSB)
15 printf('output caused by second MSB=%f volt\n',VMSB2
    )
16 printf('output caused by third MSB=%f volt\n',VMSB3)
17 printf('output caused by fourth MSB=%f volt\n',VMSB4
    )
18 printf('output caused by fifth MSB=%f volt\n',VMSB5)
19 printf('output caused by sixth MSB=%f volt\n',VMSB6)
20 //displaying the results//
```

Scilab code Exa 8.12 finding output voltages

```
1 //calculating required data//
2 //example 12//
3 clc
4 //clears the command window//
5 clear
```

```

6 //clears//
7 V=10;//high input voltage//
8 //first part//
9 //digital input is 101001//
10 Vo=(10*2^0+0+0+10*2^3+0+10*2^5)/2^6;//output voltage
    //
11 printf('output voltage for digital input 101001=%f
    volt\n',Vo);//displaying the result//
12 //second part//
13 //digital input is 110001//
14 Vo1=(10*1+0+0+0+10*16+10*32)/64;//output voltage//
15 printf('output voltage for input 110001=%f volt',Vo1
    );//displaying the result//

```

Scilab code Exa 8.13 calculating maximum scale voltage

```

1 //calculating maximum full scale voltage//
2 //example 13//
3 clc
4 //clears the command window//;
5 clear
6 //clears//
7 //digital input is 11111111//
8 //high input level=10V//
9 Vo=10*(1+2+4+8+16+32+64+128)/256;//output voltage//
10 printf('maximum full scale voltage=%f volt',Vo);//
    result displayed//

```

Scilab code Exa 8.14 finding required data

```

1 //calculating required data//
2 //example 14//
3 clc

```

```

4 //clears the command window//;
5 clear
6 //clears//
7 //part i//
8 //output high voltage=10V//
9 VMSB1=1/2*10;//output caused by MSB//
10 VMSB2=1/4*10;//output caused by second MSB//
11 VMSB3=1/8*10;//output caused by third MSB//
12 VLSB=1/16*10;//output caused by LSB//
13 printf('output caused by msb=%f volt\n',VMSB1)
14 printf('output caused by second msb=%f volt\n',VMSB2
    )
15 printf('output caused by third msb=%f\n',VMSB3)
16 printf('output caused by lsb=%f volt\n',VLSB)
17 //displaying the results//
18 //part ii//
19 //input is 1011//
20 V=1*VMSB1+0+1*VMSB3+1*VLSB;//output voltage//
21 printf('output voltage at input 1011=%f volt\n',V);
    //displaying the result//
22 //3rd part//
23 Vo=VMSB1+VMSB2+VMSB3+VLSB;//full scale output
    voltage which occurs at 1111//
24 printf('full scale output voltage=%f volt',Vo);//
    displaying the result//

```

Scilab code Exa 8.15 calculating number of bits

```

1 //calculating no of bits//
2 //example 15//
3 clc
4 //clears the command window//;
5 clear
6 //clears//
7 disp('resolution=voltage corresponding to LSB')

```



```

8 //weight of LSB=1/2^n,where n=no. of bits//
9 disp('Voltage corresponding to LSB=1/2^n*10;which
      should be equal to resolution which is 10mv')
10 disp('therefore 2^n=1000')
11 disp('n=10'); //nearest value of n which satisfies
      the equation is 10//
12 //result is displayed//

```

Scilab code Exa 8.16 finding resolution

```

1 //calculating required data//
2 //example 16//
3 clc
4 //clears the command window//;
5 clear
6 //clears//
7 V=10; //full scale voltage
8 W=1/2^12; //weight of LSB//
9 R=V*W; //resolution//
10 printf('resolution=%f volt\n',R); //result is
      displayed//
11 S=R; //step size//
12 R1=S/V*100; //percentage resolution//
13 printf('percentage resolution=%f percent',R1)
14 //result displayed//

```

Scilab code Exa 8.17 calculating required data

```

1 //calculating required data//
2 //example 17//
3 clc
4 //clears the command window//;
5 clear

```

```

6 //clears//
7 //n=no. of bits//
8 //resolution=1/2^n*5//
9 R=5*10^-3;//resolution//
10 V=5;//full scale output//
11 n=log((5*10^-3)/5)/log(.5)
12 //no. of bits//
13 n=round(n)
14 disp('no of bits:')
15 disp(n);//result is displayed//
16 S=1/2^n*5;//step size//
17 R1=S/V*100;//percentage resolution//
18 disp('percentage resolution:')
19 disp(R1)
20 //result is displayed//

```

Scilab code Exa 8.18 calculating conversion time

```

1 //calculating conversion time//
2 //example 18//
3 clc
4 //clears the command window//;
5 clear
6 //clears//
7 T=1/(1000*10^3);//time for one clock cycle//
8 disp(T)
9 n=10;//no. of bits//
10 t=(2^n-1)*T;//conversion time for 10 bit A/D
    converter//
11 disp('conversion time for digital ramp A/D converter
    :')
12 disp(t);//displaying result//
13 t1=T*10;
14 disp('conversion time for 10 bit successive
    approximation A/D converter:')

```

15 `disp(t1)`

Scilab code Exa 8.19 finding resolution

```
1 //calculating required data//
2 //example 19//
3 clc
4 //clears the command window//;
5 clear
6 //clears//
7 E=10;//change in input voltage//
8 n=8;//no. of bits//
9 R=E/((2^n)-1);//resolution//
10 disp('resolution:')
11 disp(R);//displaying result//
12 R1=(1/((2^n)-1))*100;//percentage resolution//
13 disp('percentage resolution:')
14 disp(R1);//displaying result//
```

Scilab code Exa 8.20 finding number of bits

```
1 //finding no. of bits//
2 //example 20//
3 clc
4 //clears the command window//;
5 clear
6 //clears//
7 E=10;//change in input voltage//
8 R=10^-3;//resolution//
9 n=log(E/R+1)/log(2);//no of bits//
10 n=n+1;
11 n=round(n)
12 disp('no of bits:')
```

```
13 disp(n); //result displayed //
```

Scilab code Exa 8.21 calculating maximum conversion time

```
1 //finding maximum conversion time//
2 //example 21//
3 clc
4 //clears the command window//;
5 clear
6 //clears//
7 c=2^10; //maximum no of counts//
8 f=2*10^-6; //counter advance rate of 1 count per
   second//
9 T=c*f; //conversion time//
10 disp('maximum conversion time:')
11 disp(T); //displaying result //
```

Scilab code Exa 8.22 finding error in voltage

```
1 //finding required data//
2 //example 22//
3 clc
4 //clears the command window//;
5 clear
6 //clears//
7 f=500; //frequency//
8 F=2*f; //aperture rate//
9 t=5*10^-6; //aperture time//
10 v=2*%pi*F*t; //error in voltage//
11 disp('error in voltage:')
12 disp(v); //displaying result//
13 V=v*100; //percentage error//
14 disp('percentage error:')
```

```
15 disp(V); //displaying result //
```

Scilab code Exa 8.23 finding aperture time

```
1 //finding aperture time//
2 //example 23//
3 clc
4 //clears the command window//;
5 clear
6 //clears//
7 a=.5; //percentage accuracy//
8 f=1000; //aperture rate//
9 t=a/(2*%pi)*10^-5; //aperture time//
10 disp('aperture time:')
11 disp(t); //displaying result //
```

Scilab code Exa 8.24 finding quantisation error

```
1 //calculating required data//
2 //example 24//
3 clc
4 //clears the command window//;
5 clear
6 //clears//
7 n=10; //no. of bits//
8 q=1/(2^n)*100; //quantization error expressed as
   percentage//
9 disp('quantization percentage error:')
10 disp(q); //displaying result //
```

Scilab code Exa 8.25 finding digital output in binary form

```
1 //finding digital output in binary form//
2 //example 25//
3 clc
4 //clears the command window//;
5 clear
6 //clears//
7 V=1.085; //analog input voltage//
8 R=20*10-3; //resolution//
9 n=V/R; //no of steps//
10 n=round(n); //so that output voltage produced is less
    than analog input voltage//
11 num=dec2bin(n,8); //converting to binary//
12 disp('digital output:')
13 disp(num); //displaying result in binary form//
```
