

Scilab Textbook Companion for
Digital Electronics: An Introduction To
Theory And Practice
by W. H. Gothmann¹

Created by
Aritra Ray
B.Tech
Electronics Engineering
NIT-DURGAPUR
College Teacher
Prof. Sabyasachi Sengupta,iit-kharagpur
Cross-Checked by
Lavitha M. Pereira

May 23, 2016

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

Book Description

Title: Digital Electronics: An Introduction To Theory And Practice

Author: W. H. Gothmann

Publisher: Prentice Hall Of India Pvt. Ltd., New Delhi

Edition: 2

Year: 1994

ISBN: 81-203-0348-2

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

Exa Example (Solved example)

Eqn Equation (Particular equation of the above book)

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

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

Contents

List of Scilab Codes	4
2 Number Systems	5
3 Binary Codes	40
4 Boolean Algebra	56
6 Combinational Logic	60
11 Analog Digital Conversion	95

List of Scilab Codes

Exa 2.1	Convert 10111 to decimal	5
Exa 2.2	Convert 1011101001 to decimal	5
Exa 2.3	Convert 110111 to decimal	6
Exa 2.4	Convert 43 to binary	6
Exa 2.5	Convert 200 to binary	6
Exa 2.6	Convert 43 to binary	6
Exa 2.7	Convert 200 to binary	7
Exa 2.8	Add 1011 and 110	7
Exa 2.9	Add 11110 and 11	8
Exa 2.10	Subtract one from 100	8
Exa 2.11	Multiply 1011 by 101	9
Exa 2.12	Multiply 11010 by 11011	9
Exa 2.13	Divide 110110 by 101	10
Exa 2.14	Convert 1010111010 to hexadecimal	10
Exa 2.15	Convert 11011110101110 to hexadecimal	11
Exa 2.16	Convert 4A8C to binary	11
Exa 2.17	Convert FACE to binary	12
Exa 2.18	Convert 2C9 to decimal	12
Exa 2.19	Convert EB4A to decimal	12
Exa 2.20	Convert 423 to hexadecimal	12
Exa 2.21	Convert 72905 to hexadecimal	13
Exa 2.22	Add 1A8 and 67B	13
Exa 2.23	Add ACEF1 and 16B7D	14
Exa 2.24	Subtract 3A8 from 1273	14
Exa 2.25	Multiply 1A3 by 89	15
Exa 2.26	Divide 1EC87 by A5	15
Exa 2.27	Convert 11111011110101 to octal	16
Exa 2.28	Convert 1011110100011000111 to octal	16

Exa 2.29	Convert octal number 3674 to binary	17
Exa 2.30	Minus 5 in twos complement form	17
Exa 2.31	12 bit twos complement	18
Exa 2.32	16 bit twos complement	18
Exa 2.33	Twos complement of minus4 second method	19
Exa 2.34	Twos complement of minus17 second method	20
Exa 2.35	Twos complement of minus4 third method	20
Exa 2.36	Twos complement of minus17 third method	21
Exa 2.37	Negative decimal number represented by 10011011	22
Exa 2.38	Add minus 17 to minus 30	23
Exa 2.39	Add minus 20 to 26	24
Exa 2.40	Add minus 29 to 14	24
Exa 2.41	Ones complement of minus 13	25
Exa 2.42	Ones complement of minus 13 second method	26
Exa 2.43	Ones complement addition using 4 bits	26
Exa 2.44	Ones complement addition using 4 bits	27
Exa 2.45	Ones complement addition using 8 bits	27
Exa 2.46	Convert 1101 point 000101 to decimal	28
Exa 2.47	Convert point 375 to binary	30
Exa 2.48	Convert point 54545 to binary	31
Exa 2.49	Convert 38 point 21 to binary	32
Exa 2.50	Binary addition	34
Exa 2.51	Binary mutiplication	36
Exa 3.1	Add 647 to 492 in BCD	40
Exa 3.2	Add 4318 and 7678 in BCD	42
Exa 3.3	Add 3 and 2 in XS3	44
Exa 3.4	Add 6 and 8 in XS3	45
Exa 3.5	Convert binary 1011 to Gray code	46
Exa 3.6	Convert binary 1001011 to Gray code	47
Exa 3.7	Convert Gray code 1011 to binary	48
Exa 3.8	Convert Gray code 1001011 to binary	49
Exa 3.9	Hamming code for 1011	50
Exa 3.10	Hamming code for 0101	52
Exa 3.11	Correction of Hamming code 1111101	53
Exa 4.1	Demorganize a function	56
Exa 4.2	Reduce an expression	56
Exa 4.3	Reduce an expression	57
Exa 4.4	Reduce a boolean expression	58

Exa 4.5	Reduce a Boolean expression	59
Exa 6.1	Convert A plus B to minterms	60
Exa 6.2	Find minterms for A plus BC	60
Exa 6.3	Find minterms for AB plus ACD	61
Exa 6.4	Minterm designation	62
Exa 6.5	Minterm designation	62
Exa 6.6	2 variable mapping	63
Exa 6.7	3 variable mapping	64
Exa 6.8	Mapping an expression	65
Exa 6.9	Mapping an expression	66
Exa 6.10	4 variable mapping	67
Exa 6.11	Reduce an expression by Kmap	68
Exa 6.12	Reduce an expression using Kmap	70
Exa 6.13	Reduce expression by kmap	72
Exa 6.14.a	Inputs Required	73
Exa 6.14.b	Inputs required	73
Exa 6.14.c	Inputs required	74
Exa 6.14.d	Inputs required	74
Exa 6.15	Minimise expression using mapping	75
Exa 6.16	Reduce using mapping	77
Exa 6.17	Kmap POS SOP	81
Exa 6.18	Minimise expression	84
Exa 6.19	Reduce kmap by POS	85
Exa 6.20	Find minimum of expression	86
Exa 6.24	Multiple output equation using mapping	87
Exa 6.25	Multiple output equation using mapping	89
Exa 6.26	Reduce by variable mapping	91
Exa 6.27	Reduce by variable mapping	91
Exa 6.28	Solve using 3 variable mapping	92
Exa 6.29	Reduce by mapping	92
Exa 6.30	Reducing expression by Kmap	93
Exa 11.1	Percentage resolution of 5bit DA converter	95
Exa 11.2	6 bit analog to digital converter	95
Exa 11.3	Compute the gain of an Opamp	96
Exa 11.4	Compute the output voltage	97
Exa 11.5	Compute output voltage	98
Exa 11.6	Resolution and Percent resolution 12bit DAconverter	99
Exa 11.7	Resolution and Percent resolution 10bit ADconverter	100

Chapter 2

Number Systems

Scilab code Exa 2.1 Convert 10111 to decimal

```
1 //Example 2-1//
2 //Binary to Decimal Conversion//
3 a=bin2dec('10111')
4 //Decimal equivalent of the binary number//
5 disp(a)
6 //answer in decimal form//
```

Scilab code Exa 2.2 Convert 1011101001 to decimal

```
1 //Example 2-2//
2 //Binary to Decimal Conversion//
3 a=bin2dec('1011101001')
4 //Decimal equivalent of the binary number//
5 disp(a)
6 //answer in decimal form//
```

Scilab code Exa 2.3 Convert 110111 to decimal

```
1 //Example 2-3//
2 //Binary to Decimal Conversion//
3 a=bin2dec('110111')
4 //Decimal equivalent of the binary number//
5 disp(a)
6 //answer in decimal form//
```

Scilab code Exa 2.4 Convert 43 to binary

```
1 //Example 2-4//
2 //Decimal to binary conversion//
3 a=dec2bin(43)
4 //Binary equivalent of the decimal number//
5 disp(a)
6 //answer in binary form//
```

Scilab code Exa 2.5 Convert 200 to binary

```
1 //Example 2-5//
2 //Decimal to binary conversion//
3 a=dec2bin(200)
4 //Binary equivalent of the decimal number//
5 disp(a)
6 //answer in binary form//
```

Scilab code Exa 2.6 Convert 43 to binary

```
1 //Example 2-6//
```

```
2 //Decimal to binary conversion//
3 a=dec2bin(43)
4 //Binary equivalent of the decimal number//
5 disp(a)
6 //answer in binary form//
```

Scilab code Exa 2.7 Convert 200 to binary

```
1 //Example 2-7//
2 //Decimal to binary conversion//
3 a=dec2bin(200)
4 //Binary equivalent of the decimal number//
5 disp(a)
6 //answer in binary form//
```

Scilab code Exa 2.8 Add 1011 and 110

```
1 //Example 2-8//
2 //Binary addition//
3 clc
4 //clears the console//
5 clear
6 //clears the already existing variables//
7 x=bin2dec('1011')
8 y=bin2dec('110')
9 //binary to decimal conversion//
10 z=x+y
11 //addition//
12 a=dec2bin(z)
13 //decimal to binary conversion//
14 disp(' addition of the 2 binary numbers is: ')
15 disp(a)
16 //answer in binary form//
```

Scilab code Exa 2.9 Add 11110 and 11

```
1 //Example 2-9//
2 //Binary addition//
3 clc
4 //clears the console//
5 clear
6 //clears the already existing variables//
7 x=bin2dec('11110')
8 y=bin2dec('11')
9 //binary to decimal conversion//
10 z=x+y
11 //addition//
12 a=dec2bin(z)
13 //decimal to binary conversion//
14 disp(' addition of the 2 binary numbers is: ')
15 disp(a)
16 //answer in binary form//
```

Scilab code Exa 2.10 Subtract one from 100

```
1 //Example 2-10//
2 //Binary Subtraction//
3 clc
4 //clears the console//
5 clear
6 //clears all existing variables//
7 x=bin2dec('100')
8 y=bin2dec('1')
9 //binary to decimal conversion//
10 z=x-y
```

```
11 //subtraction//
12 a=dec2bin(z)
13 //decimal to binary conversion//
14 disp('subtraction of two binary numbers is:')
15 disp(a)
16 //answer in binary form//
```

Scilab code Exa 2.11 Multiply 1011 by 101

```
1 //Example 2-11//
2 //Binary multiplication//
3 clc
4 //clears the console//
5 clear
6 //clears all existing variables//
7 x=bin2dec('1011')
8 y=bin2dec('101')
9 //binary to decimal conversion//
10 z=x*y;
11 //multiplication//
12 a=dec2bin(z)
13 //decimal to binary conversion//
14 disp(' multiplication of two binary numbers is: ')
15 disp(a)
16 //answer in binary form//
```

Scilab code Exa 2.12 Multiply 11010 by 11011

```
1 //Example 2-12//
2 //Binary multiplication//
3 clc
4 //clears the console//
5 clear
```

```

6 //clears all existing variables//
7 x=bin2dec('11010')
8 y=bin2dec('11011')
9 //binary to decimal conversion//
10 z=x*y;
11 //multiplication//
12 a=dec2bin(z)
13 //decimal to binary conversion//
14 disp(' multiplication of two binary numbers is: ')
15 disp(a)
16 //answer in binary form//

```

Scilab code Exa 2.13 Divide 110110 by 101

```

1 //Example 2-13//
2 //Binary Division//
3 x=bin2dec('110110')
4 y=bin2dec('101')
5 r=modulo(x,y)
6 //finding the remainder//
7 z=x/y
8 q=floor(z)
9 //finding the quotient//
10 quo=dec2bin(q)
11 rem=dec2bin(r)
12 //decimal to binary conversions//
13 disp('the quotient is :')
14 disp(quo)
15 disp('the remainder is : ')
16 disp(rem)
17 //answers in binary form//

```

Scilab code Exa 2.14 Convert 1010111010 to hexadecimal

```
1 //Example 2-14//
2 //binary to hexadecimal conversion//
3 x=bin2dec('1010111010')
4 //decimal equivalent of the binary number//
5 a=dec2hex(x)
6 //Hex equivalent of the decimal number//
7 disp(a)
8 //answer in hexadecimal form//
```

Scilab code Exa 2.15 Convert 11011110101110 to hexadecimal

```
1 //Example 2-15//
2 //binary to hexadecimal conversion//
3 x=bin2dec('11011110101110')
4 //decimal equivalent of the binary number//
5 a=dec2hex(x)
6 //Hex equivalent of the decimal number//
7 disp(a)
8 //answer in hexadecimal form//
```

Scilab code Exa 2.16 Convert 4A8C to binary

```
1 //Example 2-16//
2 //Hexadecimal to binary conversion//
3 x=hex2dec('4A8C')
4 //decimal conversion of the hexadecimal number//
5 a=dec2bin(x)
6 //Binary equivalent of the decimal number//
7 disp(a)
8 //answer in binary form//
```

Scilab code Exa 2.17 Convert FACE to binary

```
1 //Example 2-17//
2 //Hexadecimal to binary conversion//
3 x=hex2dec('FACE')
4 //decimal conversion of the hexadecimal number//
5 a=dec2bin(x)
6 //Binary equivalent of the decimal number//
7 disp(a)
8 //answer in binary form//
```

Scilab code Exa 2.18 Convert 2C9 to decimal

```
1 //Example 2-18//
2 //Hexadecimal to decimal conversion//
3 a=hex2dec('2C9')
4 //decimal equivalent of the hexadecimal number//
5 disp(a)
6 //answer in decimal form//
```

Scilab code Exa 2.19 Convert EB4A to decimal

```
1 //Example 2-19//
2 //Hexadecimal to decimal conversion//
3 a=hex2dec('EB4A')
4 //decimal equivalent of the hexadecimal number//
5 disp(a)
6 //answer in decimal form//
```

Scilab code Exa 2.20 Convert 423 to hexadecimal

```
1 //Example 2-20//
2 //Decimal to hexadecimal conversion//
3 a=dec2hex(423)
4 //hexadecimal equivalent of the decimal number//
5 disp(a)
6 //answer in hexadecimal form//
```

Scilab code Exa 2.21 Convert 72905 to hexadecimal

```
1 //Example 2-21//
2 //Decimal to hexadecimal conversion//
3 a=dec2hex(72905)
4 //hexadecimal equivalent of the decimal number//
5 disp(a)
6 //answer in hexadecimal form//
```

Scilab code Exa 2.22 Add 1A8 and 67B

```
1 //Example 2-22//
2 //Hexadecimal addition//
3 clc
4 //clears the console//
5 clear
6 //clears all existing variables//
7 x=hex2dec('1A8')
8 y=hex2dec('67B')
9 //Decimal conversion of the hexadecimal numbers//
10 z=x+y
11 //addition//
12 a=dec2hex(z)
13 //decimal to hexadecimal conversion//
14 disp(' addition of the 2 hexadecimal numbers is ')
15 disp(a)
```



```
16 //answer in hexadecimal form//
```

Scilab code Exa 2.23 Add ACEF1 and 16B7D

```
1 //Example 2-23//
2 //Hexadecimal addition//
3 clc
4 //clears the console//
5 clear
6 //clears all existing variables//
7 x=hex2dec('ACEF1')
8 y=hex2dec('16B7D')
9 //Decimal conversion of the hexadecimal numbers//
10 z=x+y
11 //addition//
12 a=dec2hex(z)
13 //decimal to hexadecimal conversion//
14 disp(' addition of the 2 hexadecimal numbers is ')
15 disp(a)
16 //answer in hexadecimal form//
```

Scilab code Exa 2.24 Subtract 3A8 from 1273

```
1 //Example 2-24//
2 //Hexadecimal Subtraction//
3 clc
4 //clears the console//
5 clear
6 //clears all existing variables//
7 x=hex2dec('1273')
8 y=hex2dec('3A8')
9 //Decimal conversion of the hexadecimal numbers//
10 z=x-y
```

```

11 //addition//
12 a=dec2hex(z)
13 //decimal to hexadecimal conversion//
14 disp(' addition of the 2 hexadecimal numbers is ')
15 disp(a)
16 //answer in hexadecimal form//

```

Scilab code Exa 2.25 Multiply 1A3 by 89

```

1 //Example 6-25//
2 //Solve multiple output equation using mapping//
3 clc
4 //clears the window//
5 clear
6 //clears all existing variables//
7 x=hex2dec('1A3')
8 y=hex2dec('89')
9 //Decimal conversion of the hexadecimal numbers//
10 z=x*y
11 //multiplication//
12 a=dec2hex(z)
13 //decimal to hexadecimal conversion//
14 disp(' multiplication of the 2 hexadecimal numbers
      is ')
15 disp(a)
16 //answer in hexadecimal form//

```

Scilab code Exa 2.26 Divide 1EC87 by A5

```

1 //Example 2-26//
2 //Hexadecimal Division//
3 x=hex2dec('1EC87')
4 y=hex2dec('A5')

```

```

5 r=modulo(x,y)
6 //finding the remainder//
7 z=x/y
8 q=floor(z)
9 //finding the quotient//
10 quo=dec2hex(q)
11 rem=dec2hex(r)
12 //decimal to binary conversions//
13 disp('the quotient is :')
14 disp(quo)
15 disp('the remainder is : ')
16 disp(rem)
17 //answers in binary form//

```

Scilab code Exa 2.27 Convert 11111011110101 to octal

```

1 //Example 2-27//
2 //Binary to octal conversion//
3 x=bin2dec('11111011110101')
4 //decimal equivalent of the binary number//
5 a=dec2oct(x)
6 //octal equivalent of the decimal number//
7 disp(a)
8 //answer in octal form//

```

Scilab code Exa 2.28 Convert 1011110100011000111 to octal

```

1 //Example 2-28//
2 //Binary to octal conversion//
3 x=bin2dec('1011110100011000111')
4 //decimal equivalent of the binary number//
5 a=dec2oct(x)
6 //octal equivalent of the decimal number//

```

```
7 disp(a)
8 //answer in octal form//
```

Scilab code Exa 2.29 Convert octal number 3674 to binary

```
1 //Example 2-29//
2 //octal to binary conversion//
3 x=oct2dec('3674')
4 //decimal equivalent of the octal number//
5 a=dec2bin(x)
6 //binary equivalent of the decimal number//
7 disp(a)
8 //answer in binary form//
```

Scilab code Exa 2.30 Minus 5 in twos complement form

```
1 //Example 2-30//
2 // -5 in two s complement form//
3 clc
4 //clears the console//
5 clear
6 //clears all existing variables//
7 x=2^8
8 //Smallest nine bit number//
9 y=5
10 z=x-y
11 //subtraction//
12 a=dec2bin(z)
13 //binary conversion of the decimal number//
14 disp(' -5 in two s complement form is ')
15 disp(a)
16 //answer in binary form//
```

Scilab code Exa 2.31 12 bit twos complement

```
1 //Example 2-31//
2 // -4,-15,-17 in two s complement form//
3 clc
4 //clears the console//
5 clear
6 //clears all existing variables//
7 x=2^12
8 //Smallest nine bit number//
9 p=4
10 q=15
11 r=17
12 u=x-p
13 v=x-q
14 w=x-r
15 //subtraction//
16 a=dec2bin(u)
17 b=dec2bin(v)
18 c=dec2bin(w)
19 //binary conversion of the decimal number//
20 disp(' -4 in two s complement form is ')
21 disp(a)
22 disp(' -15 in two s complement form is ')
23 disp(b)
24 disp(' -17 in two s complement form is ')
25 disp(c)
26 //answers in binary form//
```

Scilab code Exa 2.32 16 bit twos complement

```
1 //Example 2-32//
```

```

2 // -16000 in two s complement form//
3 clc
4 //clears the console//
5 clear
6 //clears all existing variables//
7 x=2^16
8 //Smallest nine bit number//
9 y=16000
10 z=x-y
11 //subtraction//
12 a=dec2bin(z)
13 //binary conversion of the decimal number//
14 disp(' -16000 in two s complement form is ')
15 disp(a)
16 //answer in binary form//

```

Scilab code Exa 2.33 Twos complement of minus4 second method

```

1 //Example 2-33//
2 // -4 in two s complement form by second method//
3 clc
4 //clears the window//
5 clear
6 //clears all existing variables//
7 x=bitcmp(4,8)
8 //complement of the decimal number 4(8 bit
  representation)//
9 y=1
10 z=x+y
11 //1 is added to the complement//
12 a=dec2bin(z)
13 //binary conversion of the decimal number//
14 disp(' -4 in two s complement form is: ')
15 disp(a)
16 //result is displayed//

```

Scilab code Exa 2.34 Twos complement of minus17 second method

```
1 //Example 2-34//
2 // -17 in two's complement form by second method//
3 clc
4 //clears the window//
5 clear
6 //clears all existing variables//
7 x=bitcmp(17,8)
8 //complement of the decimal number 17(8 bit
   representation)//
9 y=1
10 z=x+y
11 //1 is added to the complement//
12 a=dec2bin(z)
13 //binary conversion of the decimal number//
14 disp(' -17 in two's complement form is: ')
15 disp(a)
16 //result is displayed//
```

Scilab code Exa 2.35 Twos complement of minus4 third method

```
1 //Example 2-35//
2 //-4 in two's complement by third method//
3 clc
4 //clears the console//
5 clear
6 //clears all existing variables//
7 a=00000100
8 c=0
9 z=0
```

```

10 for i=1: 8
11     x(i)=modulo(a,10)
12     a=a/10
13     a=floor(a)
14 end
15 for i=1: 8
16     if c>1 then
17         break
18     end
19     if x(i)==1 then
20         for k=i+1: 8
21             x(k)=bitcmp(x(k),1)
22             c=c+1
23         end
24     end
25 end
26 for i=1: 8
27     z=z+x(i)*10^(i-1)
28 end
29 disp('-4 in twos complement is :')
30 disp(z)
31 //answer is displayed//

```

Scilab code Exa 2.36 Twos complement of minus17 third method

```

1 //Example 2-36//
2 //-17 in two's complement by third method//
3 clc
4 //clears the console//
5 clear
6 //clears all existing variables//
7 a=00010001
8 c=0
9 z=0
10 for i=1: 8

```



```

11     x(i)=modulo(a,10)
12     a=a/10
13     a=floor(a)
14 end
15 for i=1: 8
16     if c>1 then
17         break
18     end
19     if x(i)==1 then
20         for k=i+1: 8
21             x(k)=bitcmp(x(k),1)
22             c=c+1
23         end
24     end
25 end
26 for i=1: 8
27     z=z+x(i)*10^(i-1)
28 end
29 disp('−17 in twos complement is :')
30 disp(z)
31 //answer is displayed//

```

Scilab code Exa 2.37 Negative decimal number represented by 10011011

```

1 //Example 2−37//
2 // negative decimal representation of 10011011//
3 clc
4 //clears the window//
5 clear
6 //clears all existing variables//
7 b=bin2dec('10011011')
8 x=bitcmp(b,8)
9 //complement of the decimal number 17(8 bit
   representation)//
10 y=1

```

```

11 z=x+y
12 //1 is added to the complement//
13 a=dec2bin(z)
14 //binary conversion of the decimal number//
15 z=z*(-1)
16 disp(' the negative value that 10011011 represents
      is: ')
17 disp(z)
18 //result is displayed//

```

Scilab code Exa 2.38 Add minus 17 to minus 30

```

1 //Example 2-38//
2 //add -17 to -30//
3 clc
4 //clears the console//
5 clear
6 //clears all existing variables//
7 x=bitcmp(17,8)
8 y=bitcmp(30,8)
9 //complement of the decimal numbers 17 and 30//
10 z=1
11 u=x+z
12 v=y+z
13 //1 is added to the complements//
14 w=u+v
15 a=dec2bin(w)
16 //binary conversion of the decimal number//
17 disp('binary form of number obtained by adding -17
      to -30')
18 disp(a)
19 //result is displayed//
20 disp(' the msb is discarded ,so eight bit
      representation is the answer in binary form ')
21 a=dec2bin(w-(2^8))

```

```
22 disp(a)
23 //final result is displayed//
```

Scilab code Exa 2.39 Add minus 20 to 26

```
1 //Example 2-39//
2 //add -20 to 26//
3 clc
4 //clears the console//
5 clear
6 //clears all existing variables//
7 x=bitcmp(20,8)
8 //finds complement of 29//
9 y=1
10 u=x+y
11 //1 is added to the complement//
12 v=26
13 w=u+v
14 a=dec2bin(w)
15 //binary conversion of the decimal number//
16 disp(' binary form of the number obtained by adding
      -20 to 26 ')
17 disp(a)
18 //result is displayed//
19 disp(' the msb is discarded ,so eight bit
      representation is the answer in binary form ')
20 a=dec2bin(w-(2^8))
21 disp(a)
22 //final result is displayed//
```

Scilab code Exa 2.40 Add minus 29 to 14

```
1 //Example 2-40//
```

```

2 //add -29 to 14//
3 clc
4 //clears the console//
5 clear
6 //clears all existing variables//
7 x=bitcmp(29,8)
8 //finds complement of 29//
9 y=1
10 u=x+y
11 //1 is added to the complement//
12 v=14
13 w=u+v
14 a=dec2bin(w)
15 //binary conversion of the decimal number//
16 disp(' binary form of the number obtained by adding
        -29 to 14 ')
17 disp(a)
18 //result is displayed//

```

Scilab code Exa 2.41 Ones complement of minus 13

```

1 //Example 2-41//
2 //one's complement form of -13//
3 clc
4 //clears the window//
5 clear
6 //clears all existing variables//
7 x=bitcmp(13,8)
8 //complement of 13//
9 a=dec2bin(x)
10 //binary conversion of the decimal number//
11 disp(' ones complement form of -13 ')
12 disp(a)
13 //result is displayed//

```

Scilab code Exa 2.42 Ones complement of minus 13 second method

```
1 //Example 2-42//
2 //ones complement of -13 by 2nd method//
3 clc
4 //clears the window//
5 clear
6 //clears all existing variables//
7 a=bitcmp(0,8)
8 //decimal equivalent of 11111111//
9 b=13
10 c=a-b
11 //subtracting 13 from decimal equivalent of
    11111111//
12 z=dec2bin(c)
13 disp('ones complement of -13 is:')
14 disp(z)
15 //result is displayed//
```

Scilab code Exa 2.43 Ones complement addition using 4 bits

```
1 //Example 2-43//
2 //add -3 to -2 in one's complement using 4 bits//
3 clc
4 //clears the window//
5 clear
6 //clears all the existing variables//
7 x=bitcmp(3,4)
8 y=bitcmp(2,4)
9 //complement of the decimal number 2//
10 z=x+y+1
11 //carry is added//
```

```

12 a=dec2bin(z)
13 //binary conversion//
14 disp('binary form of the number obtained by adding
      -3 to -2')
15 disp(a)
16 //result is displayed//
17 disp('msb is discarded ,4 bit representation is the
      answer in binary form')
18 a=dec2bin(z-(2^4))
19 disp(a)
20 //Final result is displayed//

```

Scilab code Exa 2.44 Ones complement addition using 4 bits

```

1 //Example 2-44//
2 //add -3 to 2 in one's complement using 4 bits//
3 clc
4 //clears the window//
5 clear
6 //clears all the existing variables//
7 x=2
8 y=bitcmp(3,4)
9 //complement of the decimal number 2//
10 z=x+y
11 a=dec2bin(z)
12 //binary conversion//
13 disp('binary form of the number obtained by adding
      -3 to 2')
14 disp(a)
15 //result is displayed//

```

Scilab code Exa 2.45 Ones complement addition using 8 bits

```

1 //Example 2-45//
2 //add 3 to -2 in one's complement using 8 bits//
3 clc
4 //clears the window//
5 clear
6 //clears all the existing variables//
7 x=3
8 y=bitcmp(2,8)
9 //complement of the decimal number 2//
10 z=x+y+1
11 //carry is added//
12 a=dec2bin(z)
13 //binary conversion//
14 disp('binary form of the number obtained by adding 3
      to -2')
15 disp(a)
16 //result is displayed//
17 disp('msb is discarded,8 bit representation is the
      answer in binary form')
18 a=dec2bin(z-(2^8))
19 disp(a)
20 //Final result is displayed//

```

Scilab code Exa 2.46 Convert 1101 point 000101 to decimal

```

1 //Example 2-46//
2 //Conversion to decimal//
3 clc
4 //clears the console//
5 clear
6 //clears all existing variables//
7 p=1
8 q=1
9 z=0
10 b=0

```

```

11 w=0
12 f=0
13 //initialising//
14 //bin=input (enter the binary number to be converted
    to its decimal form)//
15 //accepting the input from the user//
16 bin=1101.000101
17 d=modulo(bin,1)
18 //separating the decimal part from the integer part
    //
19 d=d*10^10
20 a=floor(bin)
21 //removing the decimal part//
22 while(a>0)
23 //loop to enter the binary bits of the integer part
    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 each bit of the integer part with its
    corresponding positional value 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 bits of the decimal part 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 each bit with its corresponding
    positional value and adding//
45 z=z+w(1,n)*(0.5)^(11-n)
46 end
47 z=z*10000
48 z=round(z)
49 //rounding off to 4 decimal places//
50 z=z/10000
51 x=f+z
52 disp('the decimal equivalent of the binary number is
    :')
53 disp(x)
54 //result is displayed//

```

Scilab code Exa 2.47 Convert point 375 to binary

```

1 //Example 2-47//
2 //Conversion of decimal to binary//
3 clc
4 //clears the console//
5 clear
6 //clears all existing variables//
7 q=0
8 b=0
9 s=0
10 //initialising//
11 //a=input(enter the decimal number to be converted
    to its binary form)
12 //taking input from the user//
13 a=0.375
14 d=modulo(a,1)
15 //separating the decimal part from the integer//
16 a=floor(a)
17 //removing the decimal part//
18 while(a>0)

```

```

19 //integer part converted to equivalent binary form//
20 x=modulo(a,2)
21 b=b+(10^q)*x
22 a=a/2
23 a=floor(a)
24 q=q+1
25 end
26 for i=1: 10
27 //taking values after the decimal part and
    converting to equivalent binary form//
28 d=d*2
29 q=floor(d)
30 s=s+q/(10^i)
31 if d>=1 then
32     d=d-1
33 end
34 end
35 k=b+s
36 disp('the decimal number in binary form is :')
37 disp(k)
38 //result is displayed//

```

Scilab code Exa 2.48 Convert point 54545 to binary

```

1 //Example 2-48//
2 //Conversion of decimal to binary//
3 clc
4 //clears the console//
5 clear
6 //clears all existing variables//
7 q=0
8 b=0
9 s=0
10 //initialising//
11 //a=input(enter the decimal number to be converted

```

```

        to its binary form)
12 //taking input from the user//
13 a=0.54545
14 d=modulo(a,1)
15 //separating the decimal part from the integer//
16 a=floor(a)
17 //removing the decimal part//
18 while(a>0)
19 //integer part converted to equivalent binary form//
20 x=modulo(a,2)
21 b=b+(10^q)*x
22 a=a/2
23 a=floor(a)
24 q=q+1
25 end
26 for i=1: 10
27 //taking values after the decimal part and
        converting to equivalent binary form//
28 d=d*2
29 q=floor(d)
30 s=s+q/(10^i)
31 if d>=1 then
32     d=d-1
33 end
34 end
35 k=b+s
36 disp('the decimal number in binary form is :')
37 disp(k)
38 //result is displayed//

```

Scilab code Exa 2.49 Convert 38 point 21 to binary

```

1 //Example 2-49//
2 //Conversion of decimal to binary//
3 clc

```

```

4 //clears the console//
5 clear
6 //clears all existing variables//
7 q=0
8 b=0
9 s=0
10 //initialising//
11 //a=input(enter the decimal number to be converted
    to its binary form)
12 //taking input from the user//
13 a=38.21
14 d=modulo(a,1)
15 //separating the decimal part from the integer//
16 a=floor(a)
17 //removing the decimal part//
18 while(a>0)
19 //integer part converted to equivalent binary form//
20 x=modulo(a,2)
21 b=b+(10^q)*x
22 a=a/2
23 a=floor(a)
24 q=q+1
25 end
26 for i=1: 10
27 //taking values after the decimal part and
    converting to equivalent binary form//
28 d=d*2
29 q=floor(d)
30 s=s+q/(10^i)
31 if d>=1 then
32     d=d-1
33 end
34 end
35 k=b+s
36 disp('the integer part of the binary form is :')
37 disp(b)
38 disp('the fractional part of the binary form is:')
39 disp(s)

```

40 //result is displayed//

Scilab code Exa 2.50 Binary addition

```
1 //Example 2-50//
2 //addition of binary numbers//
3 //this program requires functions binary2decimal.sci
  and decimal2binary.sci//
4 clc
5 //clears the window//
6 clear
7 //clears all existing variables//
8 function x=binary2decimal(bin)
9 p=1
10 q=1
11 z=0
12 b=0
13 w=0
14 f=0
15 //initialising//
16 d=modulo(bin,1)
17 //separating the decimal part from the integer part
  //
18 d=d*1010
19 a=floor(bin)
20 //removing the decimal part//
21 while(a>0)
22 //loop to enter the binary bits of the integer part
  into a matrix//
23 r=modulo(a,10)
24 b(1,q)=r
25 a=a/10
26 a=floor(a)
27 q=q+1
28 end
```

```

29 for m=1: q-1
30 //multiplying each bit of the integer part with its
    corresponding positional value and adding//
31 c=m-1
32 f=f+b(1,m)*(2^c)
33 end
34 while(d>0)
35 //loop to take the bits of the decimal part into a
    matrix//
36 e=modulo(d,2)
37 w(1,p)=e
38 d=d/10
39 d=floor(d)
40 p=p+1
41 end
42 for n=1: p-1
43 //multiplying each bit with its corresponding
    positional value and adding//
44 z=z+w(1,n)*(0.5)^(11-n)
45 end
46 z=z*10000
47 z=round(z)
48 //rounding off to 4 decimal places//
49 z=z/10000
50 x=f+z
51 endfunction
52 function y=decimal2binary(a)
53 q=0
54 b=0
55 s=0
56 //initialising//
57 d=modulo(a,1)
58 //separating the decimal part from the integer//
59 a=floor(a)
60 //removing the decimal part//
61 while(a>0)
62 //integer part converted to equivalent binary form//
63 x=modulo(a,2)

```

```

64 b=b+(10^q)*x
65 a=a/2
66 a=floor(a)
67 q=q+1
68 end
69 for i=1: 10
70 //taking values after the decimal part and
    converting to equivalent binary form//
71 d=d*2
72 q=floor(d)
73 s=s+q/(10^i)
74 if d>=1 then
75     d=d-1
76 end
77 end
78 y=b+s
79 endfunction
80 x=binary2decimal(11.011)
81 y=binary2decimal(10.001)
82 z=x+y
83 a=decimal2binary(z)
84 disp('the addition of the binary numbers is :')
85 disp(a)
86 //result is displayed//

```

Scilab code Exa 2.51 Binary mutiplication

```

1 //Example 2-51//
2 //multiplication of binary numbers//
3 //this program requires functions binary2decimal.sci
    and decimal2binary.sci//
4 clc
5 //clears the window//
6 clear
7 //clears all existing variables//

```

```

8 function x=binary2decimal(bin)
9 p=1
10 q=1
11 z=0
12 b=0
13 w=0
14 f=0
15 //initialising//
16 d=modulo(bin,1)
17 //separating the decimal part from the integer part
    //
18 d=d*10^10
19 a=floor(bin)
20 //removing the decimal part//
21 while(a>0)
22 //loop to enter the binary bits of the integer part
    into a matrix//
23 r=modulo(a,10)
24 b(1,q)=r
25 a=a/10
26 a=floor(a)
27 q=q+1
28 end
29 for m=1: q-1
30 //multiplying each bit of the integer part with its
    corresponding positional value and adding//
31 c=m-1
32 f=f+b(1,m)*(2^c)
33 end
34 while(d>0)
35 //loop to take the bits of the decimal part into a
    matrix//
36 e=modulo(d,2)
37 w(1,p)=e
38 d=d/10
39 d=floor(d)
40 p=p+1
41 end

```



```

42 for n=1: p-1
43 //multiplying each bit with its corresponding
    positional value and adding//
44 z=z+w(1,n)*(0.5)^(11-n)
45 end
46 z=z*10000
47 z=round(z)
48 //rounding off to 4 decimal places//
49 z=z/10000
50 x=f+z
51 endfunction
52 function y=decimal2binary(a)
53 q=0
54 b=0
55 s=0
56 //initialising//
57 d=modulo(a,1)
58 //separating the decimal part from the integer//
59 a=floor(a)
60 //removing the decimal part//
61 while(a>0)
62 //integer part converted to equivalent binary form//
63 x=modulo(a,2)
64 b=b+(10^q)*x
65 a=a/2
66 a=floor(a)
67 q=q+1
68 end
69 for i=1: 10
70 //taking values after the decimal part and
    converting to equivalent binary form//
71 d=d*2
72 q=floor(d)
73 s=s+q/(10^i)
74 if d>=1 then
75     d=d-1
76 end
77 end

```

```
78 y=b+s
79 endfunction
80 x=binary2decimal(10.001)
81 y=binary2decimal(0.11)
82 z=x*y
83 a=decimal2binary(z)
84 disp('the multiplication of the binary numbers is :'
      )
85 disp(a)
86 //result is displayed//
```

Chapter 3

Binary Codes

Scilab code Exa 3.1 Add 647 to 492 in BCD

```
1 //Example 3-1//
2 //BCD addition//
3 clc
4 //clears the console//
5 clear
6 //clears all existing variables//
7 a=647
8 b=492
9 for i=1: 3
10     x(i)=modulo(a,10)
11     a=a/10
12     a=floor(a)
13     y(i)=modulo(b,10)
14     b=b/10
15     b=floor(b)
16 end
17 d=x(1)+y(1)
18 db=dec2bin(d)
19 if d>9 then
20     db=dec2bin(d+6)
21     db=dec2bin(bin2dec(db)-bin2dec('10000'))
```

```

22 end
23 e=x(2)+y(2)
24 eb=dec2bin(e)
25 if d>9 then
26     eb=dec2bin(e+1)
27     e=e+1
28 end
29 if e>9
30     eb=dec2bin(e+6)
31     eb=dec2bin(bin2dec(eb)-bin2dec('10000'))
32 end
33 f=x(3)+y(3)
34 fb=dec2bin(f)
35 if e>9 then
36     fb=dec2bin(f+1)
37     f=f+1
38 end
39 if f>9
40     fb=dec2bin(f+6)
41     fb=dec2bin(bin2dec(fb)-bin2dec('10000'))
42     dc(4)=1
43 end
44 dc(1)=bin2dec(db)
45 dc(2)=bin2dec(eb)
46 dc(3)=bin2dec(fb)
47 z=0
48 for i=1: 4
49     z=z+dc(i)*(10^(i-1))
50 end
51 disp(z)
52 disp('equivalent binary form')
53 p=strcat(dec2bin(dc(4),1)+dec2bin(dc(3),4)+dec2bin(
    dc(2),4)+dec2bin(dc(1),4))
54 disp(p)
55 //answer is displayed//

```

Scilab code Exa 3.2 Add 4318 and 7678 in BCD

```
1 //Example 3-2//
2 //BCD addition//
3 clc
4 //clears the console//
5 clear
6 //clears all existing variables//
7 a=4318
8 b=7678
9 for i=1: 4
10     x(i)=modulo(a,10)
11     a=a/10
12     a=floor(a)
13     y(i)=modulo(b,10)
14     b=b/10
15     b=floor(b)
16 end
17 d=x(1)+y(1)
18 db=dec2bin(d)
19 if d>9 then
20     db=dec2bin(d+6)
21     db=dec2bin(bin2dec(db)-bin2dec('10000'))
22 end
23 e=x(2)+y(2)
24 eb=dec2bin(e)
25 if d>9 then
26     eb=dec2bin(e+1)
27     e=e+1
28     end
29 if e>9
30     eb=dec2bin(e+6)
31     eb=dec2bin(bin2dec(eb)-bin2dec('10000'))
32 end
```

```

33 f=x(3)+y(3)
34 fb=dec2bin(f)
35 if e>9 then
36     fb=dec2bin(f+1)
37     f=f+1
38     end
39 if f>9
40     fb=dec2bin(f+6)
41     fb=dec2bin(bin2dec(fb)-bin2dec('10000'))
42 end
43 g=x(4)+y(4)
44 gb=dec2bin(g)
45 if f>9 then
46     gb=dec2bin(g+1)
47     g=g+1
48     end
49 if g>9
50     gb=dec2bin(g+6)
51     gb=dec2bin(bin2dec(gb)-bin2dec('10000'))
52     dc(5)=1
53 end
54 dc(1)=bin2dec(db)
55 dc(2)=bin2dec(eb)
56 dc(3)=bin2dec(fb)
57 dc(4)=bin2dec(gb)
58 z=0
59 for i=1: 5
60     z=z+dc(i)*(10^(i-1))
61 end
62 disp(z)
63 disp('equivalent binary form')
64 p=strcat(dec2bin(dc(5),1)+dec2bin(dc(4),4)+dec2bin(
        dc(3),4)+dec2bin(dc(2),4)+dec2bin(dc(1),4))
65 disp(p)
66 //answer is displayed//

```

Scilab code Exa 3.3 Add 3 and 2 in XS3

```
1 //Example 3-3//
2 //add 3 and 2 in Excess 3 code//
3 clc
4 //clears the console//
5 clear
6 //clears all existing variables//
7 n=3
8 m=2
9 z=0
10 a=0110
11 b=0101
12 ea=dec2bin(n+3)
13 eb=dec2bin(m+3)
14 for i=1: 4
15     x(i)=modulo(a,10)
16     a=a/10
17     a=floor(a)
18     y(i)=modulo(b,10)
19     b=b/10
20     b=floor(b)
21 end
22 for i=1: 4
23     g(i)=bitand(x(i),y(i))
24     p(i)=bitor(x(i),y(i))
25 end
26 c(1)=0
27 for i=1: 4
28     c(i+1)=bitor(g(i),bitand(p(i),c(i)))
29 end
30 if c(5)==1 then
31     z=dec2bin(bin2dec(ea)+bin2dec(eb)+3)
32 end
```

```

33 if c(5)==0 then
34         z=dec2bin(bin2dec(ea)+bin2dec(eb)-3)
35 end
36 disp('equivalent binary number after excess 3
      addition' )
37 disp(z)
38 disp('equivalent decimal number')
39 disp(m+n)
40 //result is displayed//

```

Scilab code Exa 3.4 Add 6 and 8 in XS3

```

1 //Example 3-4//
2 //add 6 and 8 in Excess 3 code//
3 clc
4 //clears the console//
5 clear
6 //clears all existing variables//
7 n=6
8 m=8
9 a=1001
10 b=1011
11 ea=dec2bin(n+3)
12 eb=dec2bin(m+3)
13 for i=1: 4
14     x(i)=modulo(a,10)
15     a=a/10
16     a=floor(a)
17     y(i)=modulo(b,10)
18     b=b/10
19     b=floor(b)
20 end
21 for i=1: 4
22     g(i)=bitand(x(i),y(i))
23     p(i)=bitor(x(i),y(i))

```



```

24 end
25 c(1)=0
26 for i=1: 4
27     c(i+1)=bitor(g(i),bitand(p(i),c(i)))
28 end
29 if c(5)==1 then
30     z=dec2bin(bin2dec(ea)+bin2dec(eb)+3)
31 end
32 if c(5)==0 then
33     z=dec2bin(bin2dec(ea)+bin2dec(eb)-3)
34 end
35 disp('equivalent binary number after excess 3
      addition ' )
36 disp(z)
37 disp('equivalent decimal number')
38 disp(m+n)
39 //result is displayed//

```

Scilab code Exa 3.5 Convert binary 1011 to Gray code

```

1 //Example 3-5//
2 //binary to gray code//
3 clc
4 //clears the console//
5 clear
6 //clears all existing variables//
7 a=1011
8 for i=1: 4
9     x(i)=modulo(a,10)
10    a=a/10
11    a=floor(a)
12 end
13 y(4)=x(4)
14 k=3
15 while(k>0)

```

```

16     if bitand(x(k+1),x(k))==1 then
17         y(k)=bitcmp(1,1)
18     else
19         y(k)=bitor(x(k+1),x(k))
20     end
21     k=k-1
22 end
23 //display//
24 z=0
25 for i=1: 4
26     z=z+y(i)*(10^(i-1))
27 end
28 disp('equivalent gray code')
29 disp(z)
30 //equivalent gray code is displayed//

```

Scilab code Exa 3.6 Convert binary 1001011 to Gray code

```

1 //Example 3-6//
2 //binary to gray code//
3 clc
4 //clears the console//
5 clear
6 //clears all existing variables//
7 a=1001011
8 for i=1: 7
9     x(i)=modulo(a,10)
10    a=a/10
11    a=floor(a)
12 end
13 y(7)=x(7)
14 k=6
15 while(k>0)
16     if bitand(x(k+1),x(k))==1 then
17         y(k)=bitcmp(1,1)

```

```

18     else
19         y(k)=bitor(x(k+1),x(k))
20     end
21     k=k-1
22 end
23 //display//
24 z=0
25 for i=1: 7
26     z=z+y(i)*(10^(i-1))
27 end
28 disp('equivalent gray code')
29 disp(z)
30 //equivalent gray code is displayed//

```

Scilab code Exa 3.7 Convert Gray code 1011 to binary

```

1 //Example 3-7//
2 //gray code to binary//
3 clc
4 //clears the console//
5 clear
6 //clears all existing variables//
7 a=1011
8 for i=1: 4
9     x(i)=modulo(a,10)
10    a=a/10
11    a=floor(a)
12 end
13 y(4)=x(4)
14 k=3
15 while(k>0)
16     if bitand(y(k+1),x(k))==1 then
17         y(k)=bitcmp(1,1)
18     else
19         y(k)=bitor(y(k+1),x(k))

```

```

20     end
21     k=k-1
22 end
23 z=0
24 for i=1: 4
25     z=z+y(i)*(10^(i-1))
26 end
27 disp('equivalent binary code')
28 disp(z)
29 //equivalent binary code is displayed//

```

Scilab code Exa 3.8 Convert Gray code 1001011 to binary

```

1 //Example 3-8//
2 //gray code to binary//
3 clc
4 //clears the console//
5 clear
6 //clears all existing variables//
7 a=1001011
8 for i=1: 7
9     x(i)=modulo(a,10)
10    a=a/10
11    a=floor(a)
12 end
13 y(7)=x(7)
14 k=6
15 while(k>0)
16     if bitand(y(k+1),x(k))==1 then
17         y(k)=bitcmp(1,1)
18     else
19         y(k)=bitor(y(k+1),x(k))
20     end
21     k=k-1
22 end

```

```

23 z=0
24 for i=1: 7
25     z=z+y(i)*(10^(i-1))
26 end
27 disp('equivalent binary code:')
28 disp(z)
29 //equivalent binary code is displayed//

```

Scilab code Exa 3.9 Hamming code for 1011

```

1 //Example 3-9//
2 //Hamming code//
3 clc
4 //clears the command window//
5 clear
6 //clears all existing variables//
7 z=1011
8 //input//
9 a=0;b=0;c=0;d=0;
10 //taking the input//
11 for i=1:7
12     x(i)=0
13     if i==3 then
14         x(i)=1
15     end
16     if i==5 then
17         x(i)=1
18     end
19     if i==7 then
20         x(i)=1
21     end
22 end
23 //establishing even parity at positions 1,3,5,7//
24 for i=1: 7
25     if x(i)==1 then

```

```

26         a=a+1
27     end
28 end
29 d=modulo(a,2)
30 if (d==1) then
31     x(1)=1
32 end
33 //establishing even parity at positions 2,3,6,7//
34 for i=2: 7
35     if (i==5) then
36         continue
37     end
38     if (x(i)==1) then
39         b=b+1
40     end
41 end
42 d=modulo(b,2)
43 if (d==1) then
44     x(2)=1
45 end
46 //establishing even parity at positions 4,5,6,7//
47 for i=5:7
48     if (x(i)==1) then
49         c=c+1
50     end
51 end
52 d=modulo(c,2)
53 if (d==1) then
54     x(4)=1
55 end
56 //displaying the result//
57 disp('the hamming code for this data is given as')
58 for i=1:7
59     disp(x(i))
60 end

```

Scilab code Exa 3.10 Hamming code for 0101

```
1 //Example 3-10//
2 //Hamming code//
3 clc
4 //clears the command window//
5 clear
6 //clears all existing variables//
7 z=0101
8 //input//
9 a=0;b=0;c=0;d=0;
10 //taking the input//
11 for i=1:7
12     x(i)=0
13     if i==3 then
14         x(i)=1
15     end
16     if i==6 then
17         x(i)=1
18     end
19 end
20 //establishing even parity at positions 1,3,5,7//
21 for i=1: 7
22     if i==6 then
23         continue
24     end
25     if x(i)==1 then
26         a=a+1
27     end
28 end
29 d=modulo(a,2)
30 if (d==1) then
31     x(1)=1
32 end
```

```

33 //establishing even parity at positions 2,3,6,7//
34 for i=2: 7
35     if x(i)==1 then
36         b=b+1
37     end
38 end
39 d=modulo(b,2)
40 if (d==1) then
41     x(2)=1
42 end
43 //establishing even parity at positions 4,5,6,7//
44 for i=5:7
45     if (x(i)==1) then
46         c=c+1
47     end
48 end
49 d=modulo(c,2)
50 if (d==1) then
51     x(4)=1
52 end
53 //displaying the result//
54 disp('the hamming code for this data is given as')
55 for i=1:7
56     disp(x(i))
57 end

```

Scilab code Exa 3.11 Correction of Hamming code 1111101

```

1 //Example 3-11//
2 //Correction of received Hamming code//
3 clc
4 //clears the console//
5 clear
6 //clears all existing variables//
7 z=1111101

```



```

8 //incorrect code taken as input//
9 a=0;b=0;c=0;d=0;
10 //taking the input in an array//
11 for i=1: 7
12     x(i)=1
13     if i==2 then
14         x(i)=0
15     end
16 end
17 //checking for 4,5,6,7 even parity//
18 for i=4: 7
19     if x(i)==1 then
20         a=a+1
21     end
22 end
23 d=modulo(a,2)
24 if d==1 then
25     disp('wrong entry for 4,5,6,7')
26     x(4)=1
27 else
28     disp('correct entry for 4,5,6,7')
29 end
30 //checking for 2,3,6,7 even parity//
31 for i=2: 7
32     if i==4 then
33         continue
34     end
35     if i==5 then
36         continue
37     end
38     if x(i)==1 then
39         a=a+1
40     end
41 end
42 d=modulo(a,2)
43 if d==1 then
44     disp('wrong entry for 2,3,6,7')
45     x(2)=1

```

```

46 else
47     disp('correct entry for 2,3,6,7')
48 end
49 //checking for 1,3,5,7 even parity//
50 for i=1: 7
51     if i==2 then
52         continue
53     end
54     if i==4 then
55         continue
56     end
57     if i==6 then
58         continue
59     end
60     if x(i)==1 then
61         a=a+1
62     end
63 end
64 d=modulo(a,2)
65 if d==1 then
66     disp('correct entry for 1,3,5,7')
67     x(1)=1
68 else
69     disp('wrong entry for 1,3,5,7')
70 end
71 disp('Therefore ,bit 2 is in error and the corrected
       code is :')
72 for i=1: 7
73     disp(x(i))
74 end
75 //correct code is displayed//

```

Chapter 4

Boolean Algebra

Scilab code Exa 4.1 Demorganize a function

```
1 //Example 4-1//
2 //demorganize function//
3 clc
4 //clears the console//
5 clear
6 //clears all existing variables//
7 //the given function is as follows//
8 disp('Given function - (AB'+C)'' ')
9 disp('now demorganizing the function')
10 disp('complement function')
11 disp(' AB'+C ')
12 disp('changing operators')
13 disp(' (A+B''')(C) ')
14 disp('complement variables')
15 disp(' (A'''+B)(C''') ')
16 //final answer displayed after simplification//
```

Scilab code Exa 4.2 Reduce an expression

```

1 //Example 4-2//
2 //Reducing an expression//
3 clc
4 //clears the console//
5 clear
6 //clears all existing variables//
7 //the given expression is as follows//
8 disp(' Given Expression – A+B(C+(DE) )')
9 disp('applying DeMorgan theorem')
10 disp(' reduced expression : ')
11 disp(' A+B(C+D'+E')')
12 disp('Applying DeMorgan theorem again')
13 disp(' A+B(C' DE) ')
14 disp(' Therefore final reduced expression is : ')
15 disp(' A+BC' DE ')
16 //final expression displayed//

```

Scilab code Exa 4.3 Reduce an expression

```

1 //Example 4-3//
2 //Reducing a given expression//
3 clc
4 //clears the console//
5 clear
6 //clears all existing variables//
7 //the given expression is as follows//
8 disp(' Given Expression – ((AB)'+A'+AB)')
9 disp('applying DeMorgan Theorem')
10 disp(' reduced expression ')
11 disp(' (A'+B'+A'+AB)')
12 disp(' as A+A=A')
13 disp(' (A'+B'+AB)')
14 //By law 20//
15 disp(' (A'+B'+A)')
16 disp('rearranging')

```

```

17 disp(' (A+A''+B'')'' ')
18 //by law 13//
19 disp(' (1+B'')'' ')
20 //by law 11//
21 disp(' 1'' ')
22 disp(' 0 ')
23 //final reduced expression is displayed/

```

Scilab code Exa 4.4 Reduce a boolean expression

```

1 //Example 4-4//
2 //Reducing a given expression//
3 clc
4 //clears the window//
5 clear
6 //clears all existing variables//
7 //the given expression is as follows//
8 disp(' Given Expression - AB+(AC)''+AB''C(AB+C) ')
9 disp(' Multiplying ')
10 disp(' AB+(AC)''+AABB''C+AB''CC ')
11 //using laws 18,6,8,9//
12 disp(' AB+(AC)''+AB''C ')
13 disp(' applying DeMorgan theorem ')
14 disp(' AB+A''+C''+AB''C ')
15 disp(' rearrange ')
16 disp(' AB+C''+A''+AB''C ')
17 //reduce using law 20//
18 disp(' AB+C''+A''+B''C ')
19 disp(' rearranging again ')
20 disp(' A''+AB+C''+B''C ')
21 //reduce using law 20//
22 disp(' A''+B+C''+B'' ')
23 //using laws 11 and 13//
24 disp(' 1 ')
25 //final reduced expression is displayed//

```

Scilab code Exa 4.5 Reduce a Boolean expression

```
1 //Example 4-5//
2 //Reduce a given expression//
3 clc
4 //clears the window//
5 clear
6 //clears all existing variables//
7 //the given expression is as follows//
8 disp(' Given Expression - ((AB''+ABC)''+A(B+AB''))''
      ')
9 disp('factorise')
10 disp(' ((A(B''+BC))''+A(B+AB''))'' ')
11 //reduce using laws 18 and 20//
12 disp(' ((A(B''+C))''+A(B+A))'' ')
13 disp('multiply')
14 disp(' ((AB''+AC)''+AA+AB)'' ')
15 //reduce using laws 7,8,11,18//
16 disp(' ((AB''+AC)''+A)'' ')
17 disp('demorganize')
18 disp(' ((A''+B)(A''+C'')+A)'' ')
19 disp('multiply')
20 disp(' (A''A''+A''C''+A''B+BC''+A)'' ')
21 //Reduce using laws 18,8//
22 disp(' (A''(1+C''+B)+BC''+A)'' ')
23 //reduce using law 18,7,11//
24 disp(' 1'' ')
25 disp(' 0 ')
26 //final reduced expression is displayed//
```

Chapter 6

Combinational Logic

Scilab code Exa 6.1 Convert A plus B to minterms

```
1 //Example 6-1//
2 //convert A+B to minterms//
3 clc
4 //clears the console//
5 clear
6 //clears all existing variables//
7 //conversion to minterms//
8 disp(' Given expression - A+B ')
9 disp(' on solving ')
10 disp(' A(B+B')'+B(A+A') ')
11 disp(' multiplying ')
12 disp(' AB+AB'+AB+A'B ')
13 disp(' we know A+A=A ')
14 disp(' AB+AB'+A'B ')
15 //final result is displayed//
```

Scilab code Exa 6.2 Find minterms for A plus BC

```

1 //Example 6-2//
2 //find minterms for A+BC//
3 clc
4 //clears the console//
5 clear
6 //clears all existing variables//
7 //conversion to minterms//
8 disp('Given expression - A+BC ')
9 disp(' on solving ')
10 disp(' A(B+B''')(C+C''')+BC(A+A''') ')
11 disp(' (AB+AB''')(C+C''')+BCA+BCA''') ')
12 disp('multiplying')
13 disp(' C''AB+AB''C+AB''C+AB''C'''+BCA+BCA''') ')
14 //final result is displayed//

```

Scilab code Exa 6.3 Find minterms for AB plus ACD

```

1 //Example 6-3//
2 //find minterms for AB+ACD//
3 clc
4 //clears the window//
5 clear
6 //clears all existing variables//
7 //conversion to minterms//
8 disp(' Given expression - AB+ACD ')
9 disp(' on solving ')
10 disp(' AB(C+C''')(D+D''')+ACD(B+B''') ')
11 disp('multiplying')
12 disp(' ABCD+ABC''D'''+ABC''D+ABCD'''+AB''CD+ABCD ')
13 disp('we know A+A=A')
14 disp(' ABC''D'''+ABC''D+ABCD'''+AB''CD+ABCD ')
15 //result is displayed//

```

Scilab code Exa 6.4 Minterm designation

```
1 //Example 6-4//
2 //minterm designation of AB' 'C' 'D''//
3 clc
4 //clears the window//
5 clear
6 //clears all existing variables//
7 disp('copy original term')
8 disp(' AB' 'C' 'D'' ')
9 disp(' substitute ones for nonbarred letters and
      zeroes for barred letters ')
10 disp('after substitution')
11 disp(' 1000 ')
12 a=bin2dec('1000')
13 disp('the decimal equivalent of 1000 is:')
14 disp(a)
15 disp(' Therefore decimal subscript of m is 8 ')
16 disp(' AB' 'C' 'D'' = m8 ')
17 //result is displayed//
```

Scilab code Exa 6.5 Minterm designation

```
1 //Example 6-5//
2 //minterm designation of W' 'X' 'YZ''//
3 clc
4 //clears the window//
5 clear
6 //clears all existing variables//
7 disp('copy original term')
8 disp(' W' 'X' 'YZ'' ')
9 disp(' substitute ones for nonbarred letters and
      zeroes for barred letters ')
10 disp('after substitution')
11 disp(' 0010 ')

```

```

12 a=bin2dec('0010')
13 disp('therefore the decimal equivalent of 0010 is:')
14 disp(a)
15 disp(' Therefore decimal subscript of m is 2 ')
16 disp(' W' 'X' 'YZ' ' = m2 ')
17 //result is displayed//

```

Scilab code Exa 6.6 2 variable mapping

```

1 //Example 6-6//
2 //map C=A' 'B' '+AB' '//
3 clc
4 //clears the console//
5 clear
6 //clears all existing variables//
7 //Mapping the expression//
8 disp('      B' ' B ')
9 disp('A' ' 1   0 ')
10 disp('A   1   0 ')
11 disp(' From the map, high outputs for 0 and 2 ')
12 a=[0 0 ; 1 0]
13 disp(a)
14 for i=1: 2
15     if a(i,1)==1 then
16         b(i,1)='A'
17     else
18         b(i,1)='A' ' '
19     end
20     if a(i,2)==1 then
21         b(i,2)='B'
22     else
23         b(i,2)='B' ' '
24     end
25 end
26 m=strcat([b(1,1),b(1,2)])

```

```

27 n=strcat([b(2,1),b(2,2)])
28 disp(' evaluating expression from truth table and
      map ')
29 x=strcat([m"+" ,n]);
30 disp(x)
31 //Expression is displayed//

```

Scilab code Exa 6.7 3 variable mapping

```

1 //Example 6-7//
2 //Map X=ABC+AB' 'C+AB' 'C' '//
3 clc
4 //clears the console//
5 clear
6 //clears all existing variables//
7 //Mapping the expression//
8 disp('      B' 'C' ' B' 'C BC BC' ' ')
9 disp('A' ' 0      0      0 0 ')
10 disp('A      1      1      1 0 ')
11 disp(' From the map, high outputs for 4,5 and 7 ')
12 a=[1 0 0 ; 1 0 1 ; 1 1 1]
13 disp(a)
14 for i=1: 3
15     if a(i,1)==1 then
16         b(i,1)='A'
17     else
18         b(i,1)='A' ' '
19     end
20     if a(i,2)==1 then
21         b(i,2)='B'
22     else
23         b(i,2)='B' ' '
24     end
25     if a(i,3)==1 then
26         b(i,3)='C'

```

```

27     else
28         b(i,3)='C'
29     end
30 end
31 disp(' evaluating expression from truth table and
      map ')
32 l=strcat([ b(1,1),b(1,2),b(1,3)])
33 m=strcat([ b(2,1),b(2,2),b(2,3)])
34 n=strcat([ b(3,1),b(3,2),b(3,3)])
35 x=strcat([l"+" ,m"+" ,n ])
36 disp(x)
37 //Expression is displayed//

```

Scilab code Exa 6.8 Mapping an expression

```

1 //Example 6-8//
2 //Mapping an equation//
3 clc
4 //clears the console//
5 clear
6 //clears all existing variables//
7 //Mapping the expression//
8 disp('Z=m(9,20,21,29,30,31)')
9 disp('          A')
10 disp('          D'E' D'E DE DE'          D'E' D
      'E DE DE')
11 disp('B'C'  0    0    0    0          0
      0  0  0 ')
12 disp('BC'  0    0    0    0          1
      1  0  0 ')
13 disp('BC   0    0    0    0          0
      1  1  1 ')
14 disp('BC'  0    1    0    0          0
      0  0  0 ')

```

```

15 disp(' From the map, high outputs for
      9,20,21,29,30,31 ')
16 //Therefore the kmap is displayed//

```

Scilab code Exa 6.9 Mapping an expression

```

1 //Example 6-8//
2 //Mapping an equation//
3 clc
4 //clears the console//
5 clear
6 //clears all existing variables//
7 //Mapping the expression//
8 disp('Z=m
      (0,1,3,5,7,10,11,21,22,23,24,26,30,32,34,35,40,41,46,47,50,51,52,
      ')
9 disp('
      B''
      ')
10 disp('
      E''F'' E''F EF EF'' E''F
      '' E''F EF EF''')
11 disp('
      C''D'' 1 1 1 0 0
      0 0 0 ')
12 disp('A'' CD'' 0 1 1 0 0
      1 1 1 ')
13 disp('
      CD 0 0 0 0 0
      0 0 1 ')
14 disp('
      CD'' 0 0 1 1 1
      0 0 1 ')
15 disp('
      ')
16 disp('
      C''D'' 1 0 1 1 0
      0 1 1 ')
17 disp('A
      CD'' 0 0 0 0 1
      0 0 0 ')

```

```

18 disp('      CD      0      0      1      1      1
      1      0      0      ')
19 disp('      CD''      1      1      0      0      0
      0      0      0      ')
20 disp(' From the map, high outputs for
      0,1,3,5,7,10,11,21,22,23,24,26,30,32,34,35,40,41,46,47,50,51,52,60
      ')
21 //Therefore the kmap is displayed//

```

Scilab code Exa 6.10 4 variable mapping

```

1 //Example 6-10//
2 //Map L=W 'X' 'YZ+WX' 'YZ''+WX' 'Y' 'Z''+W' 'XYZ//
3 clc
4 //clears the console//
5 clear
6 //clears all existing variables//
7 //Mapping the expression//
8 disp('      C''D'' C''D CD CD'' ')
9 disp('A''B''      0      0      1      0 ')
10 disp('AB''      0      0      1      0 ')
11 disp('AB      0      0      0      0 ')
12 disp('AB''      1      0      0      1 ')
13 disp(' From the map, high outputs for 3,7,8,10 ')
14 a=[0 0 1 1 ; 0 1 1 1 ; 1 0 0 0 ; 1 0 1 0]
15 disp(a)
16 for i=1: 4
17     if a(i,1)==1 then
18         b(i,1)='W'
19     else
20         b(i,1)='W'' '
21     end
22     if a(i,2)==1 then
23         b(i,2)='X'
24     else

```

```

25         b(i,2)='X' ''
26     end
27     if a(i,3)==1 then
28         b(i,3)='Y'
29     else
30         b(i,3)='Y' ''
31     end
32     if a(i,4)==1 then
33         b(i,4)='Z'
34     else
35         b(i,4)=' Z' ''
36     end
37 end
38 disp(' evaluating expression from truth table and
      map ')
39 l=strcat([ b(1,1),b(1,2),b(1,3),b(1,4)])
40 m=strcat([ b(2,1),b(2,2),b(2,3),b(2,4)])
41 n=strcat([ b(3,1),b(3,2),b(3,3),b(3,4)])
42 o=strcat([ b(4,1),b(4,2),b(4,3),b(4,4)])
43 x=strcat([l"+" ,m"+" ,n"+" ,o ])
44 disp(x)
45 //Expression is displayed//

```

Scilab code Exa 6.11 Reduce an expression by Kmap

```

1 //Example 6-11//
2 //reduce expression using k-map//
3 clc
4 //clears the window//
5 clear
6 //clears all existing variables//
7 //Mapping the expression//
8 disp('          C'D'' C'D CD CD'' ')
9 disp('A'B''      1      1      1  1 ')
10 disp('AB''       0      0      1  1 ')

```

```

11 disp('AB      0      1      1  0  ')
12 disp('AB''      0      0      0  0  ')
13 disp(' From the map, high outputs for
      0,1,2,3,6,7,13,15  ')
14 //given logic equation//
15 a=[0 0 0 0;0 0 0 1;0 0 1 0;0 0 1 1 ;0 1 1 0;0 1 1
      1;1 1 0 1;1 1 1 1]
16 disp(a)
17 for i=1: 8
18     if a(i,1)==1 then
19         b(i,1)='A'
20     else
21         b(i,1)='A'' '
22     end
23     if a(i,2)==1 then
24         b(i,2)='B'
25     else
26         b(i,2)='B'' '
27     end
28     if a(i,3)==1 then
29         b(i,3)='C'
30     else
31         b(i,3)='C'' '
32     end
33     if a(i,4)==1 then
34         b(i,4)='D'
35     else
36         b(i,4)=' D'' '
37     end
38 end
39 disp(' evaluating expression from truth table and
      map  ')
40 x1=strcat([ b(1,1),b(1,2),b(1,3),b(1,4) ])
41 x2=strcat([ b(2,1),b(2,2),b(2,3),b(2,4) ])
42 x3=strcat([ b(3,1),b(3,2),b(3,3),b(3,4) ])
43 x4=strcat([ b(4,1),b(4,2),b(4,3),b(4,4) ])
44 x5=strcat([ b(5,1),b(5,2),b(5,3),b(5,4) ])
45 x6=strcat([ b(6,1),b(6,2),b(6,3),b(6,4) ])

```



```

46 x7=strcat([ b(7,1),b(7,2),b(7,3),b(7,4) ])
47 x8=strcat([ b(8,1),b(8,2),b(8,3),b(8,4) ])
48 x=( [x1"+" ,x2"+" ,x3"+" ,x4"+" ,x5"+" ,x6"+" ,x7"+" ,x8 ] )
49 disp(x)
50 //Expression is displayed//
51 disp('now reducing expression using boolean algebra'
      )
52 disp('A''B''+A''C+ABD')

```

Scilab code Exa 6.12 Reduce an expression using Kmap

```

1 //Example 6-12//
2 //reduce expression using k-map//
3 clc
4 //clears the window//
5 clear
6 //clears all existing variables//
7 //Mapping the expression//
8 disp('          C''D'' C''D CD CD'' ')
9 disp('A''B''      0      0      1  1 ')
10 disp('AB''       0      1      1  0 ')
11 disp('AB        1      1      1  1 ')
12 disp('AB''       0      1      1  0 ')
13 disp(' From the map, high outputs for
      2,3,5,7,9,11,12,13,14,15 ')
14 //given logic equation//
15 a=[0 0 1 0;0 0 1 1;0 1 0 1;0 1 1 1 ;1 0 0 1;1 0 1
      1;1 1 0 0;1 1 0 1;1 1 1 0;1 1 1 1]
16 disp(a)
17 for i=1: 10
18     if a(i,1)==1 then
19         b(i,1)='A'
20     else
21         b(i,1)='A'' '
22     end

```

```

23     if a(i,2)==1 then
24         b(i,2)='B'
25     else
26         b(i,2)='B''
27     end
28     if a(i,3)==1 then
29         b(i,3)='C'
30     else
31         b(i,3)='C''
32     end
33     if a(i,4)==1 then
34         b(i,4)='D'
35     else
36         b(i,4)=' D''
37     end
38 end
39 disp(' evaluating expression from truth table and
      map ')
40 x1=strcat([ b(1,1),b(1,2),b(1,3),b(1,4) ])
41 x2=strcat([ b(2,1),b(2,2),b(2,3),b(2,4) ])
42 x3=strcat([ b(3,1),b(3,2),b(3,3),b(3,4) ])
43 x4=strcat([ b(4,1),b(4,2),b(4,3),b(4,4) ])
44 x5=strcat([ b(5,1),b(5,2),b(5,3),b(5,4) ])
45 x6=strcat([ b(6,1),b(6,2),b(6,3),b(6,4) ])
46 x7=strcat([ b(7,1),b(7,2),b(7,3),b(7,4) ])
47 x8=strcat([ b(8,1),b(8,2),b(8,3),b(8,4) ])
48 x9=strcat([ b(9,1),b(9,2),b(9,3),b(9,4) ])
49 x10=strcat([ b(10,1),b(10,2),b(10,3),b(10,4) ])
50 x=( [x1"+" ,x2"+" ,x3"+" ,x4"+" ,x5"+" ,x6"+" ,x7"+" ,x8"+" ,
      x9"+" ,x10 ] )
51 disp(x)
52 //Expression is displayed//
53 disp('now reducing expression using boolean algebra'
      )
54 disp('A''B''C+BD+AB+AD')

```

Scilab code Exa 6.13 Reduce expression by kmap

```
1 //Example 6-13//
2 //reduce expression using k-map//
3 clc
4 //clears the console//
5 clear
6 //clears all existing variables//
7 //Mapping the expression//
8 disp('      B''C'' B''C BC BC'' ')
9 disp('A''  1      0      1  1 ')
10 disp('A      1      1      1  0 ')
11 disp(' From the map, high outputs for 0,2,3,4,5 and
      7 ')
12 a=[0 0 0 ; 0 1 0 ; 0 1 1 ; 1 0 0 ; 1 0 1 ;1 1 1]
13 disp(a)
14 for i=1: 6
15     if a(i,1)==1 then
16         b(i,1)='A'
17     else
18         b(i,1)='A'' '
19     end
20     if a(i,2)==1 then
21         b(i,2)='B'
22     else
23         b(i,2)='B'' '
24     end
25     if a(i,3)==1 then
26         b(i,3)='C'
27     else
28         b(i,3)='C'' '
29     end
30 end
31 disp(' evaluating expression from truth table and
```

```

    map ')
32 x1=strcat([ b(1,1),b(1,2),b(1,3) ])
33 x2=strcat([ b(2,1),b(2,2),b(2,3) ])
34 x3=strcat([ b(3,1),b(3,2),b(3,3) ])
35 x4=strcat([ b(4,1),b(4,2),b(4,3) ])
36 x5=strcat([ b(5,1),b(5,2),b(5,3) ])
37 x6=strcat([ b(6,1),b(6,2),b(6,3) ])
38 x=[x1"+" ,x2"+" ,x3"+" ,x4"+" ,x5"+" ,x6]
39 disp(x)
40 //Expression is displayed//
41 disp('applying laws of boolean algebra')
42 disp('AC+A' 'B+B' 'C' ' ')

```

Scilab code Exa 6.14.a Inputs Required

```

1 //example 6-14a//
2 //Number of inputs required//
3 clc
4 //clears the window//
5 clear
6 //clears all existing variables//
7 disp(' W=AB' 'D+ACD' '+EF ')
8 disp('count AND inputs')
9 disp('3+3+2=8')
10 disp('count OR inputs')
11 disp('1+1+1=3')
12 disp('therefore total inputs=11')
13 //result is displayed//

```

Scilab code Exa 6.14.b Inputs required

```

1 //example 6-14b//
2 //Number of inputs required//

```

```

3  clc
4  //clears the window//
5  clear
6  //clears all existing variables//
7  disp(' X=LM+N' 'PQ+LM' 'PQ' ' ' ')
8  disp('count AND inputs')
9  disp('2+3+4=9')
10 disp('count OR inputs')
11 disp('1+1+1=3')
12 disp('therefore total inputs=12')
13 //result is displayed//

```

Scilab code Exa 6.14.c Inputs required

```

1  //example 6-14c//
2  //Number of inputs required//
3  clc
4  //clears the window//
5  clear
6  //clears all existing variables//
7  disp(' Y=ST' 'U' 'V+STO' 'V+UV' '+SUV+TU' 'V' ' ' ')
8  disp('count AND inputs')
9  disp('4+4+2+3+3=16')
10 disp('count OR inputs')
11 disp('1+1+1+1+1=5')
12 disp('therefore total inputs=21')
13 //result is displayed//

```

Scilab code Exa 6.14.d Inputs required

```

1  //example 6-14d//
2  //Number of inputs required//
3  clc

```

```

4 //clears the window//
5 clear
6 //clears all existing variables//
7 disp(' (A+B''+C)(A''+D)(B+D'' ) ')
8 disp('count OR inputs')
9 disp('3+2+2=7')
10 disp('count AND inputs')
11 disp('1+1+1=3')
12 disp('therefore total inputs=10')
13 //result is displayed//

```

Scilab code Exa 6.15 Minimise expression using mapping

```

1 //Example 6-15//
2 //Mapping an equation//
3 clc
4 //clears the console//
5 clear
6 //clears all existing variables//
7 //Mapping the expression//
8 disp('Z=m
      (2,3,6,7,9,10,12,13,14,16,18,22,26,28,29,30)')
9 disp('
      A''
      A
      ')
10 disp('
      D''E'' D''E DE DE''
      D''E'' D
      ''E DE DE''')
11 disp('B''C'' 0 0 1 1 1
      0 0 1 ')
12 disp('BC'' 0 0 1 1 0
      0 0 1 ')
13 disp('BC 1 1 0 1 1
      1 0 1 ')
14 disp('BC'' 0 1 0 1 0
      0 0 1 ')
15 disp(' From the map, high outputs for

```

```

    2,3,6,7,9,10,12,13,14,16,18,22,26,28,29,30 ')
16 //Therefore the kmap is displayed//
17 a=[0 0 0 1 0;0 0 0 1 1;0 0 1 1 0;0 0 1 1 1;0 1 0 0
    1;0 1 0 1 0;0 1 1 0 0;0 1 1 0 1;0 1 1 1 0;1 0 0 0
    0;1 0 0 1 0;1 0 1 1 0;1 1 0 1 0;1 1 1 0 0;1 1 1
    0 1;1 1 1 1 0]
18 disp(a)
19 for i=1: 16
20     if a(i,1)==1 then
21         b(i,1)='A'
22     else
23         b(i,1)='A''''
24     end
25     if a(i,2)==1 then
26         b(i,2)='B'
27     else
28         b(i,2)='B''''
29     end
30     if a(i,3)==1 then
31         b(i,3)='C'
32     else
33         b(i,3)='C''''
34     end
35     if a(i,4)==1 then
36         b(i,4)='D'
37     else
38         b(i,4)=' D''''
39     end
40     if a(i,5)==1 then
41         b(i,5)='E'
42     else
43         b(i,5)=' E''''
44     end
45 end
46 disp(' evaluating expression from truth table and
    map ')
47 x1=strcat([ b(1,1),b(1,2),b(1,3),b(1,4),b(1,5) ])
48 x2=strcat([ b(2,1),b(2,2),b(2,3),b(2,4),b(2,5) ])

```

```

49 x3=strcat([ b(3,1),b(3,2),b(3,3),b(3,4),b(3,5) ])
50 x4=strcat([ b(4,1),b(4,2),b(4,3),b(4,4),b(4,5) ])
51 x5=strcat([ b(5,1),b(5,2),b(5,3),b(5,4),b(5,5) ])
52 x6=strcat([ b(6,1),b(6,2),b(6,3),b(6,4),b(6,5) ])
53 x7=strcat([ b(7,1),b(7,2),b(7,3),b(7,4),b(7,5) ])
54 x8=strcat([ b(8,1),b(8,2),b(8,3),b(8,4),b(8,5) ])
55 x9=strcat([ b(9,1),b(9,2),b(9,3),b(9,4),b(9,5) ])
56 x10=strcat([ b(10,1),b(10,2),b(10,3),b(10,4),b(10,5)
    ])
57 x11=strcat([ b(11,1),b(11,2),b(11,3),b(11,4),b(11,5)
    ])
58 x12=strcat([ b(12,1),b(12,2),b(12,3),b(12,4),b(12,5)
    ])
59 x13=strcat([ b(13,1),b(13,2),b(13,3),b(13,4),b(13,5)
    ])
60 x14=strcat([ b(14,1),b(14,2),b(14,3),b(14,4),b(14,5)
    ])
61 x15=strcat([ b(15,1),b(15,2),b(15,3),b(15,4),b(15,5)
    ])
62 x16=strcat([ b(16,1),b(16,2),b(16,3),b(16,4),b(16,5)
    ])
63 x=( [x1"+" ,x2"+" ,x3"+" ,x4"+" ,x5"+" ,x6"+" ,x7"+" ,x8"+" ,
    x9"+" ,x10"+" ,x11"+" ,x12"+" ,x13"+" ,x14"+" ,x15"+" ,
    x16 ])
64 disp(x)
65 disp('Reduced expression ')
66 disp('A' 'BD' 'E+AB' 'C' 'E' '+A' 'B' 'D+BCD' '+DE' ' ')
67 //Expression is displayed//

```

Scilab code Exa 6.16 Reduce using mapping

```

1 //Example 6-16//
2 //Reduce using mapping//
3 clc
4 //clears the console//

```



```

5 clear
6 //clears all existing variables//
7 //Mapping the expression//
8 disp('Y=m
      (0,2,4,8,10,13,15,16,18,20,23,24,26,32,34,40,42,45,47,48,50,56,57
      ')
9 disp('
      B''
      B
      ')
10 disp('
      E''F'' E''F EF EF'' E''F
      '' E''F EF EF''')
11 disp('
      C''D'' 1 0 0 1 1
      0 0 1 ')
12 disp('A'' CD'' 1 0 0 0 1
      0 1 0 ')
13 disp('
      CD 0 1 1 0 0
      0 0 0 ')
14 disp('
      CD'' 1 0 0 1 1
      0 0 1 ')
15 disp('
      ')
16 disp('
      C''D'' 1 0 0 1 1
      0 0 1 ')
17 disp('A CD'' 0 0 0 0 0
      0 0 0 ')
18 disp('
      CD 0 1 1 0 1
      1 0 0 ')
19 disp('
      CD'' 1 0 0 1 1
      1 0 1 ')
20 disp(' From the map, high outputs for
      0,2,4,8,10,13,15,16,18,20,23,24,26,32,34,40,42,45,47,48,50,56,57,5
      ')
21 //Therefore the kmap is displayed//
22 a=[0 0 0 0 0 0;0 0 0 0 1 0;0 0 0 1 0 0;0 0 1 0 0 0;0
      0 1 0 1 0;0 0 1 1 0 1;0 0 1 1 1 1;0 1 0 0 0 0;0
      1 0 0 1 0;0 1 0 1 0 0;0 1 0 1 1 1;0 1 1 0 0 0;0 1
      1 0 1 0;1 0 0 0 0 0;1 0 0 0 1 0;1 0 1 0 0 0;1 0
      1 0 1 0;1 0 1 1 0 1;1 0 1 1 1 1;1 1 0 0 0 0;1 1 0

```

```

        0 1 0;1 1 1 0 0 0;1 1 1 0 0 1;1 1 1 0 1 0;1 1 1
        1 0 0;1 1 1 1 0 1 ]
23 for i=1: 26
24     if a(i,1)==1 then
25         b(i,1)='A'
26     else
27         b(i,1)='A''''
28     end
29     if a(i,2)==1 then
30         b(i,2)='B'
31     else
32         b(i,2)='B''''
33     end
34     if a(i,3)==1 then
35         b(i,3)='C'
36     else
37         b(i,3)='C''''
38     end
39     if a(i,4)==1 then
40         b(i,4)='D'
41     else
42         b(i,4)=' D''''
43     end
44     if a(i,5)==1 then
45         b(i,5)='E'
46     else
47         b(i,5)=' E''''
48     end
49     if a(i,6)==1 then
50         b(i,6)='F'
51     else
52         b(i,6)=' F''''
53     end
54 end
55 disp(' evaluating expression from truth table and
        map ')
56 x1=strcat([ b(1,1),b(1,2),b(1,3),b(1,4),b(1,5),b
        (1,6) ])

```

```

57 x2= strcat([ b(2,1),b(2,2),b(2,3),b(2,4),b(2,5),b
    (2,6) ])
58 x3= strcat([ b(3,1),b(3,2),b(3,3),b(3,4),b(3,5),b
    (3,6) ])
59 x4= strcat([ b(4,1),b(4,2),b(4,3),b(4,4),b(4,5),b
    (4,6) ])
60 x5= strcat([ b(5,1),b(5,2),b(5,3),b(5,4),b(5,5),b
    (5,6) ])
61 x6= strcat([ b(6,1),b(6,2),b(6,3),b(6,4),b(6,5),b
    (6,6) ])
62 x7= strcat([ b(7,1),b(7,2),b(7,3),b(7,4),b(7,5),b
    (7,6) ])
63 x8= strcat([ b(8,1),b(8,2),b(8,3),b(8,4),b(8,5),b
    (8,6) ])
64 x9= strcat([ b(9,1),b(9,2),b(9,3),b(9,4),b(9,5),b
    (9,6) ])
65 x10= strcat([ b(10,1),b(10,2),b(10,3),b(10,4),b(10,5)
    ,b(10,6) ])
66 x11= strcat([ b(11,1),b(11,2),b(11,3),b(11,4),b(11,5)
    ,b(11,6) ])
67 x12= strcat([ b(12,1),b(12,2),b(12,3),b(12,4),b(12,5)
    ,b(12,6) ])
68 x13= strcat([ b(13,1),b(13,2),b(13,3),b(13,4),b(13,5)
    ,b(13,6) ])
69 x14= strcat([ b(14,1),b(14,2),b(14,3),b(14,4),b(14,5)
    ,b(14,6) ])
70 x15= strcat([ b(15,1),b(15,2),b(15,3),b(15,4),b(15,5)
    ,b(15,6) ])
71 x16= strcat([ b(16,1),b(16,2),b(16,3),b(16,4),b(16,5)
    ,b(16,6) ])
72 x17= strcat([ b(17,1),b(17,2),b(17,3),b(17,4),b(17,5)
    ,b(17,6) ])
73 x18= strcat([ b(18,1),b(18,2),b(18,3),b(18,4),b(18,5)
    ,b(18,6) ])
74 x19= strcat([ b(19,1),b(19,2),b(19,3),b(19,4),b(19,5)
    ,b(19,6) ])
75 x20= strcat([ b(20,1),b(20,2),b(20,3),b(20,4),b(20,5)
    ,b(20,6) ])

```

```

76 x21=strcat([ b(21,1),b(21,2),b(21,3),b(21,4),b(21,5)
              ,b(21,6) ])
77 x22=strcat([ b(22,1),b(22,2),b(22,3),b(22,4),b(22,5)
              ,b(22,6) ])
78 x23=strcat([ b(23,1),b(23,2),b(23,3),b(23,4),b(23,5)
              ,b(23,6) ])
79 x24=strcat([ b(24,1),b(24,2),b(24,3),b(24,4),b(24,5)
              ,b(24,6) ])
80 x25=strcat([ b(25,1),b(25,2),b(25,3),b(25,4),b(25,5)
              ,b(25,6) ])
81 x26=strcat([ b(26,1),b(26,2),b(26,3),b(26,4),b(26,5)
              ,b(26,6) ])
82 x=( [x1"+" ,x2"+" ,x3"+" ,x4"+" ,x5"+" ,x6"+" ,x7"+" ,x8"+" ,
        x9"+" ,x10"+" ,x11"+" ,x12"+" ,x13"+" ,x14"+" ,x15"+" ,
        x16"+" ,x17"+" ,x18"+" ,x19"+" ,x20"+" ,x21"+" ,x22"+" ,
        x23"+" ,x24"+" ,x25"+" ,x26 ] )
83 disp(x)
84 disp('Reduced expression')
85 disp('A' 'BC' 'DEF+A' 'C' 'E' 'F' '+B' 'CDF+ABCE' '+D' 'F' ' ')
86 //Expression is displayed//

```

Scilab code Exa 6.17 Kmap POS SOP

```

1
2 //Example 6-17//
3 //reduce expression using k-map by both POS and SOP
  //
4 clc
5 //clears the window//
6 clear
7 //clears all existing variables//
8 //Mapping the expression//
9 disp('      C' 'D' ' C' 'D CD CD' ' ')
10 disp('A' 'B' '      1      1      1  0 ')
11 disp('AB' '      0      0      1  1 ')

```

```

12 disp('AB      0      1      1  0 ')
13 disp('AB''      1      1      0  0 ')
14 disp(' From the map, high outputs for
      0,1,3,6,7,8,9,13,15 ')
15 disp(' From the map, low outputs for
      2,4,5,10,11,12,14 ')
16 //given logic equation//
17 a=[0 0 0 0;0 0 0 1;0 0 1 1 ;0 1 1 0;0 1 1 1;1 0 0 0
      ;1 0 0 1;1 1 0 1;1 1 1 1]
18 disp(a)
19 c=[0 0 1 0;0 1 0 0;0 1 0 1;1 0 1 0;1 0 1 1;1 1 0 0;1
      1 1 0]
20 disp(c)
21 for i=1: 9
22     if a(i,1)==1 then
23         b(i,1)='A'
24     else
25         b(i,1)='A'' '
26     end
27     if a(i,2)==1 then
28         b(i,2)='B'
29     else
30         b(i,2)='B'' '
31     end
32     if a(i,3)==1 then
33         b(i,3)='C'
34     else
35         b(i,3)='C'' '
36     end
37     if a(i,4)==1 then
38         b(i,4)='D'
39     else
40         b(i,4)=' D'' '
41     end
42 end
43 for i=1: 7
44     if c(i,1)==1 then
45         d(i,1)='A'' '

```

```

46     else
47         d(i,1)='A'
48     end
49     if c(i,2)==1 then
50         d(i,2)='B'' '
51     else
52         d(i,2)='B'
53     end
54     if c(i,3)==1 then
55         d(i,3)='C'' '
56     else
57         d(i,3)='C'
58     end
59     if c(i,4)==1 then
60         d(i,4)='D'' '
61     else
62         d(i,4)=' D '
63     end
64 end
65 disp(' evaluating expression (minterms) from truth
        table and map ')
66 x1=strcat([ b(1,1),b(1,2),b(1,3),b(1,4) ])
67 x2=strcat([ b(2,1),b(2,2),b(2,3),b(2,4) ])
68 x3=strcat([ b(3,1),b(3,2),b(3,3),b(3,4) ])
69 x4=strcat([ b(4,1),b(4,2),b(4,3),b(4,4) ])
70 x5=strcat([ b(5,1),b(5,2),b(5,3),b(5,4) ])
71 x6=strcat([ b(6,1),b(6,2),b(6,3),b(6,4) ])
72 x7=strcat([ b(7,1),b(7,2),b(7,3),b(7,4) ])
73 x8=strcat([ b(8,1),b(8,2),b(8,3),b(8,4) ])
74 x9=strcat([ b(9,1),b(9,2),b(9,3),b(9,4) ])
75 x=( [x1"+" ,x2"+" ,x3"+" ,x4"+" ,x5"+" ,x6"+" ,x7"+" ,x8"+" ,
        x9 ])
76 disp(x)
77 disp(' evaluating expression (maxterms) from truth
        table and map ')
78 disp(' (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) ''')

```

```

79 //Expression is displayed//
80 disp('now reducing expression using boolean algebra'
      )
81 disp('SOP- B''C''+A''B''D+A''BC+ABD          ie 15
      inputs')
82 disp('POS- (B+C''+D)(A''+B''+D)(A+B''+C)(A''+B+C''')
      ie 16 inputs')
83 disp('Therefore SOP form is less expensive')

```

Scilab code Exa 6.18 Minimise expression

```

1 //Example 6-18//
2 //Minimise an expression//
3 clc
4 //clears the console//
5 clear
6 //clears all existing variables//
7 //Mapping the expression//
8 disp('Y=m
      (2,3,4,5,6,7,12,13,14,15,18,19,20,21,22,23,28,29,30,31)
      ')
9 disp('
      A''
      A
      ')
10 disp('
      D''E'' D''E DE DE''
      D''E'' D
      ''E DE DE''')
11 disp('B''C''
      0      0      1      1
      0      1      1      ')
12 disp('BC''
      1      1      1      1
      1      1      1      ')
13 disp('BC
      1      1      1      1
      1      1      1      ')
14 disp('BC''
      0      0      0      0
      0      0      0      ')
15 disp(' From the map, high outputs for
      2,3,4,5,6,7,12,13,14,15,18,19,20,21,22,23,28,29,30,31

```

```

    ')
16 //Therefore the kmap is displayed//
17 disp('Therefore the minimum expression is:')
18 disp('Y=C+B''D')
19 //result is displayed//

```

Scilab code Exa 6.19 Reduce kmap by POS

```

1
2 //Example 6-19//
3 //reduce expression using k-map by finding POS//
4 clc
5 //clears the window//
6 clear
7 //clears all existing variables//
8 //Mapping the expression//
9 disp('      C''D'' C''D CD CD'' ')
10 disp('A''B''  0      0      0 1 ')
11 disp('AB''    1      0      0 0 ')
12 disp('AB      1      1      0 0 ')
13 disp('AB''    1      1      1 0 ')
14 disp(' From the map, low outputs for
      0,1,3,5,6,7,10,14,15 ')
15 //given logic equation//
16 c=[0 0 0 0;0 0 0 1;0 0 1 1;0 1 0 1;0 1 1 0;0 1 1 1;1
      0 1 0;1 1 1 0;1 1 1 1]
17 disp(c)
18 for i=1: 9
19     if c(i,1)==1 then
20         d(i,1)='A''''
21     else
22         d(i,1)='A'
23     end
24     if c(i,2)==1 then
25         d(i,2)='B''''

```



```

26     else
27         d(i,2)='B'
28     end
29     if c(i,3)==1 then
30         d(i,3)='C' ''
31     else
32         d(i,3)='C'
33     end
34     if c(i,4)==1 then
35         d(i,4)='D' ''
36     else
37         d(i,4)=' D '
38     end
39 end
40 disp(' evaluating expression from truth table and
      map')
41 l=strcat([ d(1,1),d(1,2),d(1,3),d(1,4)])
42 m=strcat([ d(2,1),d(2,2),d(2,3),d(2,4)])
43 n=strcat([ d(3,1),d(3,2),d(3,3),d(3,4)])
44 o=strcat([ d(4,1),d(4,2),d(4,3),d(4,4)])
45 x=strcat(['1'+',',m'+',',n'+',',o ])
46 disp('the sum of product expression is:')
47 disp(x)
48 disp('Reading the SOP form')
49 disp(' A ' 'B ' 'CD' '+BC' 'D' '+AB' 'D+AC' ' ' ')
50 //Expression is displayed//
51 disp('now reducing expression using boolean algebra
      from POS')
52 disp(' (A+B+C)(A ' '+C' '+D)(A+D' ')(B ' '+C' ' ' ') ')
53 disp('POS has 14 inputs,SOP has 16 inputs')
54 disp('Therefore,POS form is less expensive')

```

Scilab code Exa 6.20 Find minimum of expression

```
1 //Example 6-20//
```

```

2 //Find minimum of expression//
3 clc
4 //clears the console//
5 clear
6 //clears all existing variables//
7 //Mapping the expression//
8 disp('Y=M(0,1,9,10,11,13,14,15,16,17,22,23,26,27)')
9 disp('
      A')
10 disp('
      D'E' D'E DE DE' D'E' D
      'E DE DE'')
11 disp('B'C' 0 0 1 1 0
      0 1 1 ')
12 disp('BC' 1 1 1 1 1
      1 0 0 ')
13 disp('BC 1 0 0 0 1
      1 1 1 ')
14 disp('BC' 1 0 0 0 1
      1 0 0 ')
15 disp(' From the map, high outputs for
      0,1,9,10,11,13,14,15,16,17,22,23,26,27 ')
16 //Therefore the kmap is displayed//
17 disp('The SOP of expression is:')
18 disp('Y=B'C'D+A'B'C+ABC+BD'E'+ACD'+ABD'')
19 disp('24 inputs')
20 disp('The POS of expression is:')
21 disp('Y=(B+C+D)(B'+C+D')(A'+B+C+D)(A+B'+E')(A+B'+D'')')
22 disp('21 inputs')
23 disp('Therefore POS form is the minimum expression')
24 //result is displayed//

```

Scilab code Exa 6.24 Multiple output equation using mapping

```
1 //Example 6-24//
```

```

2 //Solve multiple output equation using mapping//
3 clc
4 //clears the window//
5 clear
6 //clears all existing variables//
7 disp('f1=Sigma m(0,1,2,4,6,7,10,14,15)')
8 //First function is displayed//
9 disp('f2=Sigma m(3,4,5,9,10,11,14) ')
10 //Second function is displayed//
11 disp('f1.f2=Sigma m(4,10,14)')
12 //Taking the common entries//
13 disp('Mapping for f1.f2')
14 disp('      C'D' C'D CD CD' ')
15 disp('A'B'  0   0   0 0 ')
16 disp('AB'   1   0   0 0 ')
17 disp('AB    0   0   0 1 ')
18 disp('AB'   0   0   0 1 ')
19 disp(' From the map, high outputs for 4,10,14')
20 //given logic equation//
21 a=[0 1 0 0;1 0 1 0;1 1 1 0]
22 disp(a)
23 for i=1: 3
24     if a(i,1)==1 then
25         b(i,1)='A'
26     else
27         b(i,1)='A''
28     end
29     if a(i,2)==1 then
30         b(i,2)='B'
31     else
32         b(i,2)='B''
33     end
34     if a(i,3)==1 then
35         b(i,3)='C'
36     else
37         b(i,3)='C''
38     end
39     if a(i,4)==1 then

```

```

40         b(i,4)='D'
41     else
42         b(i,4)=' D' ' '
43     end
44 end
45 disp(' evaluating expression from truth table and
      map ')
46 l=strcat([ b(1,1),b(1,2),b(1,3),b(1,4)])
47 m=strcat([ b(2,1),b(2,2),b(2,3),b(2,4)])
48 n=strcat([ b(3,1),b(3,2),b(3,3),b(3,4)])
49 x=strcat([l"+" ,m"+" ,n])
50 disp(x)
51 //Expression is displayed//
52 disp('now reducing expression using boolean algebra'
      )
53 disp('ACD' '+A' 'BC' 'D' ' ' )

```

Scilab code Exa 6.25 Multiple output equation using mapping

```

1 //Example 6-25//
2 //Solve multiple output equation using mapping//
3 clc
4 //clears the window//
5 clear
6 //clears all existing variables//
7 disp('f1=Sigma m(2,3,7,10,11,14)+d(1,5,15)')
8 //First function is displayed//
9 disp('f2=Sigma m(0,1,4,7,13,15)+d(5,8,15) ')
10 //Second function is displayed//
11 disp('f1.f2=Sigma m(1,7,14)+d(5,15)')
12 //Taking the common entries//
13 disp('Mapping for f1.f2')
14 disp('      C'D' ' C'D CD CD' ' ')
15 disp('A'B' '   0   1   0  0 ')
16 disp('AB' '   0   X   1  0 ')

```

```

17 disp('AB      0      0      X  1  ')
18 disp('AB''      0      0      0  0  ')
19 disp(' From the map, high outputs for 1,7,14 and
      dont cares for 5,15')
20 //given logic equation//
21 a=[0 0 0 1;0 1 0 1;0 1 1 1;1 1 1 0;1 1 1 1]
22 disp(a)
23 for i=1: 5
24     if a(i,1)==1 then
25         b(i,1)='A'
26     else
27         b(i,1)='A'' '
28     end
29     if a(i,2)==1 then
30         b(i,2)='B'
31     else
32         b(i,2)='B'' '
33     end
34     if a(i,3)==1 then
35         b(i,3)='C'
36     else
37         b(i,3)='C'' '
38     end
39     if a(i,4)==1 then
40         b(i,4)='D'
41     else
42         b(i,4)=' D'' '
43     end
44 end
45 disp(' evaluating expression from truth table and
      map ')
46 l=strcat([ b(1,1),b(1,2),b(1,3),b(1,4)])
47 m=strcat([ b(2,1),b(2,2),b(2,3),b(2,4)])
48 n=strcat([ b(3,1),b(3,2),b(3,3),b(3,4)])
49 o=strcat([ b(4,1),b(4,2),b(4,3),b(4,4)])
50 p=strcat([ b(5,1),b(5,2),b(5,3),b(5,4)])
51 x=strcat([l"+" ,m"+" ,n"+" ,o"+" ,p ])
52 disp(x)

```

```

53 //Expression is displayed//
54 disp('now reducing expression using boolean algebra'
      )
55 disp('ABC+A''C''D+A''BD')

```

Scilab code Exa 6.26 Reduce by variable mapping

```

1 //Example 6-26//
2 //Reduce by Variable Mapping//
3 clc
4 //clears the console//
5 clear
6 //clears all existing variables//
7 disp('M=A''B''CD''+A''B''CD+AB''CD''+AB''CD+A''BCD+A
      ''BC''D''+ABC''D+ABC''D''')
8 disp('Converting the 4 variable minterms to 3
      variable minterms')
9 disp(' M=m1D''+m1D+m5D''+m5D+m3D+m3D''+m6D+m6D''+m2D
      ''')
10 disp('using D+D''=1')
11 disp('Result M= B''C+A''CD+BC''D''+ABC''')
12 //final expression is displayed//

```

Scilab code Exa 6.27 Reduce by variable mapping

```

1 //Example 6-27//
2 //Reduce by Variable Mapping//
3 clc
4 //clears the console//
5 clear
6 //clears all existing variables//
7 disp('Z=A''B''CDE+A''BC''DE''+A''BCD+AB''CD''F+ABC''
      D''+ABCD''')

```

```

8 disp('Converting the 6 variable minterms to 4
      variable minterms')
9 disp('m7-E+E'';m14-F+F''')
10 disp('Result Z= A''CDE+A''BDE''+ABD''+ACD''F')
11 //final expression is displayed//

```

Scilab code Exa 6.28 Solve using 3 variable mapping

```

1 //Example 6-28//
2 //Solve using 3 Variable Mapping//
3 clc
4 //clears the console//
5 clear
6 //clears all existing variables//
7 disp('W=Dm2+m6+Dm5+d(m1+D''m7)')
8 disp('Dm5 combined with XDm1')
9 disp('Dm2 combined with Dm6')
10 disp('D''m6 combined either with Dm6 or XD''m7')
11 disp('Result W=B''CD+ABD''+BC''D')
12 //final expression is displayed//

```

Scilab code Exa 6.29 Reduce by mapping

```

1 //Example 6-29//
2 //Reduce by Mapping//
3 clc
4 //clears the console//
5 clear
6 //clears all existing variables//
7 disp('R=m0+(E+G)m2+E''m5+Gm10+Fm13+m14+m15+d(m1+Em4+
      F''m8+F''m9)')
8 disp('Converting the 7 variable minterms to 4
      variable minterms')

```

```

9 disp('Em2 covers Em0')
10 disp('E''m0 combines with m1')
11 disp('Result R=A''B''D''E+A''B''C''+B''CD''G+ABDF+
      ABC ')
12 //final expression is displayed//

```

Scilab code Exa 6.30 Reducing expression by Kmap

```

1 //Example 6-30//
2 //reduce expression using k-map//
3 clc
4 //clears the window//
5 clear
6 //clears all existing variables//
7 //Mapping the expression//
8 disp('      C''D'' C''D CD CD'' ')
9 disp('A''B''  1      0      1  1 ')
10 disp('AB''    0      0      1  1 ')
11 disp('AB      0      1      0  0 ')
12 disp('AB''    1      1      0  1 ')
13 disp(' From the map, high outputs for
      0,2,3,6,7,8,9,10,13 ')
14 //given logic equation//
15 a=[0 0 0 0;0 0 1 0;0 0 1 1 ;0 1 1 0;0 1 1 1;1 0 0 0
      ;1 0 0 1;1 0 1 0;1 1 0 1]
16 disp(a)
17 for i=1: 9
18     if a(i,1)==1 then
19         b(i,1)='A'
20     else
21         b(i,1)='A'' '
22     end
23     if a(i,2)==1 then
24         b(i,2)='B'
25     else

```



```

26         b(i,2)='B' ''
27     end
28     if a(i,3)==1 then
29         b(i,3)='C'
30     else
31         b(i,3)='C' ''
32     end
33     if a(i,4)==1 then
34         b(i,4)='D'
35     else
36         b(i,4)=' D' ''
37     end
38 end
39 disp(' evaluating expression from truth table and
map ')
40 x1=strcat([ b(1,1),b(1,2),b(1,3),b(1,4) ])
41 x2=strcat([ b(2,1),b(2,2),b(2,3),b(2,4) ])
42 x3=strcat([ b(3,1),b(3,2),b(3,3),b(3,4) ])
43 x4=strcat([ b(4,1),b(4,2),b(4,3),b(4,4) ])
44 x5=strcat([ b(5,1),b(5,2),b(5,3),b(5,4) ])
45 x6=strcat([ b(6,1),b(6,2),b(6,3),b(6,4) ])
46 x7=strcat([ b(7,1),b(7,2),b(7,3),b(7,4) ])
47 x8=strcat([ b(8,1),b(8,2),b(8,3),b(8,4) ])
48 x9=strcat([ b(9,1),b(9,2),b(9,3),b(9,4) ])
49 x=( [x1"+" ,x2"+" ,x3"+" ,x4"+" ,x5"+" ,x6"+" ,x7"+" ,x8"+" ,
x9 ] )
50 disp(x)
51 //Expression is displayed//
52 disp('now reducing expression using boolean algebra'
)
53 disp('B'D'+A'C+AC'D')

```

Chapter 11

Analog Digital Conversion

Scilab code Exa 11.1 Percentage resolution of 5bit DA converter

```
1 //Example 11-1//
2 // % Resolution of a five bit D/A converter//
3 clc
4 //clears the console//
5 clear
6 //clears all existing variables//
7 n=5
8 //here n is the number of bits//
9 disp('Max number that can be represented using 5
      bits is the binary number 11111 ie 31 in decimal
      form ')
10 pres=(1/((2n)-1))*100
11 //pres denotes the percent resolution//
12 disp(' Therefore the percent resolution of the 5 bit
      D/A converter is : ')
13 disp(pres)
14 //answer is displayed//
```

Scilab code Exa 11.2 6 bit analog to digital converter

```

1 //Example 11-2//
2 clc
3 //clears the console//
4 clear
5 //clears all existing variables//
6 n=6
7 mpsv=20
8 //n denotes the number of bits ,mpsv denotes the max.
   precision supply voltage//
9 disp('Each bit represents 1/(2^6)-1 of the total 20V
   ')
10 E1=(1/63)*20
11 //given a binary number 100110 whose voltage value
   is to be found//
12 a=100110
13 b=bin2dec('100110')
14 E2=(b/((2^6)-1))*20
15 disp('The voltage change that each LSB represents is
   : ')
16 disp(E1)
17 disp('The voltage that the binary number 100110
   represents is: ')
18 disp(E2)
19 //answers are displayed//

```

Scilab code Exa 11.3 Compute the gain of an Opamp

```

1 //Example 11-3//
2 //compute gain of an op-amp//
3 clc
4 //clears the console//
5 clear
6 //clears all existing variables//
7 Rf=40
8 Rs=5

```

```

9 Ein=3.2
10 //Given the resistance values in kilo ohms and the
    input voltage in volts//
11 Av=Rf/Rs
12 //Av denotes the gain of the op-amp//
13 Eo=Av*Ein
14 //Eo denotes the output voltage//
15 disp('The voltage gain of the given op-amp is: ')
16 disp(Av)
17 disp('The output voltage of the opamp in volts is: '
    )
18 disp(Eo)
19 //results are displayed//

```

Scilab code Exa 11.4 Compute the output voltage

```

1 //Example 11-4//
2 //compute output voltage for circuit in fig 11-12//
3 clc
4 //clears the window//
5 clear
6 //clears all existing variables//
7 R1=10
8 R2=10
9 R3=10
10 Rf=10
11 E1=3
12 E2=-2
13 E3=-4
14 //Given all the resistance values in kilo ohms and
    the voltages in volts at the inputs//
15 disp('the current through resistor R1 in milli amps
    is: ')
16 I1=E1/R1
17 disp(I1)

```

```

18 disp('the current throught resistor R2 in milli amps
      is: ')
19 I2=E2/R2
20 disp(I2)
21 disp('the current throught resistor R3 in milli amps
      is: ')
22 I3=E3/R3
23 disp(I3)
24 Eo=-(I1+I2+I3)*Rf
25 //Eo denotes the output voltage//
26 disp('The output voltage in volts of the circuit is:
      ')
27 disp(Eo)
28 //answer is displayed//

```

Scilab code Exa 11.5 Compute output voltage

```

1 //Example 11-5//
2 //compute output voltage for the circuit in fig.
  11-13//
3 clc
4 //clears the console//
5 clear
6 //clears all existing variables//
7 disp('switch configuration ABCD is 0110 ')
8 E=8
9 RA=1
10 RB=2
11 RC=4
12 RD=8
13 Rf=1
14 //Given all the resistance values in kilo ohms and
  the common voltage E in volts//
15 disp('the current throught resistor RA in milli amps
      is: ')

```

```

16 IA=0
17 //switch A is open//
18 disp(IA)
19 disp('the current through resistor RB in milli amps
      is: ')
20 IB=E/RB
21 disp(IB)
22 disp('the current through resistor RC in milli amps
      is: ')
23 IC=E/RC
24 disp(IC)
25 disp('the current through resistor RD in milli amps
      is: ')
26 ID=0
27 //switch D is open//
28 disp(ID)
29 Itot=IA+IB+IC+ID
30 //total current is denoted by Itot//
31 If=Itot
32 Eo=If*Rf
33 Eo=Eo*(-1)
34 //Eo denotes output voltage//
35 disp('Output voltage of the circuit in volts is: ')
36 disp(Eo)
37 //result is displayed//

```

Scilab code Exa 11.6 Resolution and Percent resolution 12bit DAconverter

```

1 //Example 11-6//
2 //resolution and percent resolution of a 12 bit D/A
  converter, output varies from -50 to 50//
3 clc
4 //clears the window//
5 clear

```

```

6 //clears all existing variables//
7 E=50-(-50)
8 n=12
9 //given the voltage range E in volts and the number
  of bits n//
10 disp('12 bit converter can have (2^12)-1 non zero
      states ')
11 Res=E/((2^n)-1)
12 Pres=1/((2^n)-1)*(100)
13 disp('the resolution in volts is : ')
14 disp(Res)
15 disp('the percent resolution is : ')
16 disp(Pres)
17 //answers are displayed//

```

Scilab code Exa 11.7 Resolution and Percent resolution 10bit ADconverter

```

1 //Example 11-7//
2 //resolution and percent resolution of a 10 bit D/A
  converter, input voltage varies from -10 to 10//
3 clc
4 //clears the window//
5 clear
6 //clears all existing variables//
7 E=10-(-10)
8 n=10
9 //given the voltage range E in volts and the number
  of bits n//
10 disp('10 bit converter can have (2^10)-1 non zero
      states ')
11 Res=E/((2^n)-1)
12 Pres=1/((2^n)-1)*(100)
13 disp('the resolution in millivolts is : ')
14 disp(Res*1000)

```

```
15 disp('the percent resolution is : ')
16 disp(Pres)
17 //answers are displayed//
```
