Scilab Textbook Companion for Digital Electronics: An Introduction To Theory And Practice by W. H. Gothmann[1]

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

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

Book Description

Title: 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

Chapter 2

Number Systems

Scilab code Exa 2.1 Convert 10111 to decimal

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

Scilab code Exa 2.2 Convert 1011101001 to decimal

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

Scilab code Exa 2.3 Convert 110111 to decimal

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

Scilab code Exa 2.4 Convert 43 to binary

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

Scilab code Exa 2.5 Convert 200 to binary

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

Scilab code Exa 2.6 Convert 43 to binary

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

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

Scilab code Exa 2.7 Convert 200 to binary

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

Scilab code Exa 2.8 Add 1011 and 110

```
//Example 2-8//
//Binary addition//
clc
//clears the console//
clear
//clears the already existing variables//
x=bin2dec('1011')
y=bin2dec('110')
//binary to decimal conversion//
z=x+y
//addition//
a=dec2bin(z)
//decimal to binary conversion//
disp(' addition of the 2 binary numbers is: ')
disp(a)
//answer in binary form//
```

Scilab code Exa 2.9 Add 11110 and 11

```
//Example 2-9//
//Binary addition//
clc
//clears the console//
clear
//clears the already existing variables//
x=bin2dec('11110')
y=bin2dec('11')
//binary to decimal conversion//
z=x+y
//addition//
a=dec2bin(z)
//decimal to binary conversion//
disp(' addition of the 2 binary numbers is: ')
disp(a)
//answer in binary form//
```

Scilab code Exa 2.10 Subtract one from 100

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

```
//subtraction//
a=dec2bin(z)
//decimal to binary conversion//
disp('subtraction of two binary numbers is:')
disp(a)
//answer in binary form//
```

Scilab code Exa 2.11 Multiply 1011 by 101

```
//Example 2-11//
//Binary multiplication//
clc
//clears the console//
clear
//clears all existing variables//
x=bin2dec('1011')
y=bin2dec('101')
//binary to decimal conversion//
z=x*y;
//multiplication//
a=dec2bin(z)
//decimal to binary conversion//
disp(' multiplication of two binary numbers is: ')
disp(a)
//answer in binary form//
```

Scilab code Exa 2.12 Multiply 11010 by 11011

```
//Example 2-12//
//Binary multiplication//
clc
//clears the console//
clear
```

```
//clears all existing variables//
x=bin2dec('11010')
y=bin2dec('11011')
//binary to decimal conversion//
z=x*y;
//multiplication//
a=dec2bin(z)
//decimal to binary conversion//
disp(' multiplication of two binary numbers is: ')
disp(a)
//answer in binary form//
```

Scilab code Exa 2.13 Divide 110110 by 101

```
//Example 2-13//
//Binary Division//
x=bin2dec('110110')
y=bin2dec('101')
r=modulo(x,y)
//finding the remainder//
z=x/y
q=floor(z)
//finding the quotient//
quo=dec2bin(q)
rem=dec2bin(r)
//decimal to binary conversions//
disp('the quotient is :')
disp(quo)
disp('the remainder is : ')
disp(rem)
//answers in binary form//
```

Scilab code Exa 2.14 Convert 1010111010 to hexadecimal

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

Scilab code Exa 2.15 Convert 11011110101110 to hexadecimal

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

Scilab code Exa 2.16 Convert 4A8C to binary

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

Scilab code Exa 2.17 Convert FACE to binary

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

Scilab code Exa 2.18 Convert 2C9 to decimal

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

Scilab code Exa 2.19 Convert EB4A to decimal

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

Scilab code Exa 2.20 Convert 423 to hexadecimal

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

Scilab code Exa 2.21 Convert 72905 to hexadecimal

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

Scilab code Exa 2.22 Add 1A8 and 67B

```
//Example 2-22//
//Hexadecimal addition//
clc
//clears the console//
clear
//clears all existing variables//
x=hex2dec('1A8')
y=hex2dec('67B')
//Decimal conversion of the hexadecimal numbers//
z=x+y
//addition//
a=dec2hex(z)
//decimal to hexadecimal conversion//
disp(' addition of the 2 hexadecimal numbers is ')
disp(a)
```

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

Scilab code Exa 2.23 Add ACEF1 and 16B7D

```
//Example 2-23//
//Hexadecimal addition//
clc
//clears the console//
clear
//clears all existing variables//
x=hex2dec('ACEF1')
y=hex2dec('16B7D')
//Decimal conversion of the hexadecimal numbers//
z=x+y
//addition//
a=dec2hex(z)
//decimal to hexadecimal conversion//
disp(' addition of the 2 hexadecimal numbers is ')
disp(a)
//answer in hexadecimal form//
```

Scilab code Exa 2.24 Subtract 3A8 from 1273

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

```
//addition//
a=dec2hex(z)
//decimal to hexadecimal conversion//
disp(' addition of the 2 hexadecimal numbers is ')
disp(a)
//answer in hexadecimal form//
```

Scilab code Exa 2.25 Multiply 1A3 by 89

```
//Example 6-25//
//Solve multiple output equation using mapping//
clc
//clears the window//
clear
//clears all existing variables//
x=hex2dec('1A3')
y=hex2dec('89')
//Decimal conversion of the hexadecimal numbers//
z=x*y
//multiplication//
a=dec2hex(z)
//decimal to hexadecimal conversion//
disp(' multiplication of the 2 hexadecimal numbers
   is ')
disp(a)
//answer in hexadecimal form//
```

Scilab code Exa 2.26 Divide 1EC87 by A5

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

```
r=modulo(x,y)
//finding the remainder//
z=x/y
q=floor(z)
//finding the quotient//
quo=dec2hex(q)
rem=dec2hex(r)
//decimal to binary conversions//
disp('the quotient is :')
disp(quo)
disp('the remainder is : ')
disp(rem)
//answers in binary form//
```

Scilab code Exa 2.27 Convert 11111011110101 to octal

```
//Example 2-27//
//Binary to octal conversion//
x=bin2dec('11111011110101')
//decimal equivalent of the binary number//
a=dec2oct(x)
//octal equivalent of the decimal number//
disp(a)
//answer in octal form//
```

Scilab code Exa 2.28 Convert 1011110100011000111 to octal

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

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

Scilab code Exa 2.29 Convert octal number 3674 to binary

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

Scilab code Exa 2.30 Minus 5 in twos complement form

```
//Example 2-30//
// -5 in two s complement form//
clc
//clears the console//
clear
//clears all existing variables//
x=2^8
//Smallest nine bit number//
y=5
z=x-y
//subtraction//
a=dec2bin(z)
//binary conversion of the decimal number//
disp(' -5 in two s complement form is ')
disp(a)
//answer in binary form//
```

Scilab code Exa 2.31 12 bit twos complement

```
//Example 2-31//
// -4,-15,-17 in two s complement form//
clc
//clears the console//
clear
//clears all existing variables//
x=2^12
//Smallest nine bit number//
p=4
q=15
r=17
u=x-p
v=x-q
w=x-r
//subtraction//
a=dec2bin(u)
b=dec2bin(v)
c=dec2bin(w)
//binary conversion of the decimal number//
disp(' -4 in two s complement form is ')
disp(a)
disp(' -15 in two s complement form is ')
disp(b)
disp(' -17 in two s complement form is ')
disp(c)
//answers in binary form//
```

Scilab code Exa 2.32 16 bit twos complement

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

```
// −16000 in two s complement form//
clc
//clears the console//
clear
//clears all existing variables//
x=2^16
//Smallest nine bit number//
y=16000
z=x-y
//subtraction//
a=dec2bin(z)
//binary conversion of the decimal number//
disp(' −16000 in two s complement form is ')
disp(a)
//answer in binary form//
```

Scilab code Exa 2.33 Twos complement of minus4 second method

```
//Example 2−33//
// −4 in two s complement form by second method//
clc
//clears the window//
clear
//clears all existing variables//
x=bitcmp(4,8)
//complement of the decimal number 4(8 bit
    representation)//
y=1
z=x+y
//1 is added to the complement//
a=dec2bin(z)
//binary conversion of the decimal number//
disp(' −4 in two s complement form is: ')
disp(a)
//result is displayed//
```

Scilab code Exa 2.34 Twos complement of minus17 second method

```
//Example 2-34//
// -17 in two s complement form by second method//
clc
//clears the window//
clear
//clears all existing variables//
x=bitcmp(17,8)
//complement of the decimal number 17(8 bit
    representation)//
y=1
z=x+y
//1 is added to the complement//
a=dec2bin(z)
//binary conversion of the decimal number//
disp(' -17 in two s complement form is: ')
disp(a)
//result is displayed//
```

Scilab code Exa 2.35 Twos complement of minus4 third method

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

```
for i=1: 8
    x(i)=modulo(a,10)
    a=a/10
    a=floor(a)
end
for i=1: 8
    if c>1 then
       break
    end
    if x(i)==1 then
        for k=i+1: 8
        x(k)=bitcmp(x(k),1)
        c=c+1
        end
    end
end
for i=1: 8
    z=z+x(i)*10^(i-1)
end
disp('−4 in twos complement is :')
disp(z)
//answer is displayed//
```

Scilab code Exa 2.36 Twos complement of minus17 third method

```
//Example 2−36//
//−17 in two's complement by third method//
clc
//clears the console//
clear
//clears all existing variables//
a=00010001
c=0
z=0
for i=1: 8
```

```
    x(i)=modulo(a,10)
    a=a/10
    a=floor(a)
end
for i=1: 8
    if c>1 then
       break
    end
    if x(i)==1 then
        for k=i+1: 8
        x(k)=bitcmp(x(k),1)
        c=c+1
        end
    end
end
for i=1: 8
    z=z+x(i)*10^(i-1)
end
disp('-17 in twos complement is :')
disp(z)
//answer is displayed//
```

Scilab code Exa 2.37 Negative decimal number represented by 10011011

```
//Example 2-37//
// negative decimal representation of 10011011//
clc
//clears the window//
clear
//clears all existing variables//
b=bin2dec('10011011')
x=bitcmp(b,8)
//complement of the decimal number 17(8 bit representation)//
y=1
```

```
z=x+y
//1 is added to the complement//
a=dec2bin(z)
//binary conversion of the decimal number//
z=z*(-1)
disp(' the negative value that 10011011 represents
    is: ')
disp(z)
//result is displayed//
```

Scilab code Exa 2.38 Add minus 17 to minus 30

```
//Example 2-38//
//add -17 to -30//
clc
//clears the console//
clear
//clears all exisiting variables//
x=bitcmp(17,8)
y=bitcmp(30,8)
//complement of the decimal numbers 17 and 30//
z=1
u=x+z
v=y+z
//1 is added to the complements//
w=u+v
a=dec2bin(w)
//binary conversion of the decimal number//
disp('binary form of number obtained by adding -17
    to -30')
disp(a)
//result is displayed//
disp(' the msb is discarded,so eight bit
    representation is the answer in binary form ')
a=dec2bin(w-(2^8))
```

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

Scilab code Exa 2.39 Add minus 20 to 26

```
//Example 2-39//
//add -20 to 26//
clc
//clears the console//
clear
//clears all existing variables//
x=bitcmp(20,8)
//finds complement of 29//
y=1
u=x+y
//1 is added to the complement//
v=26
w=u+v
a=dec2bin(w)
//binary conversion of the decimal number//
disp('binary form of the number obtained by adding
    -20 to 26')
disp(a)
//result is displayed//
disp('the msb is discarded,so eight bit
    representation is the answer in binary form')
a=dec2bin(w-(2^8))
disp(a)
//final result is displayed//
```

Scilab code Exa 2.40 Add minus 29 to 14

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

```
//add -29 to 14//
clc
//clears the console//
clear
//clears all existing variables//
x=bitcmp(29,8)
//finds complement of 29//
y=1
u=x+y
//1 is added to the complement//
v=14
w=u+v
a=dec2bin(w)
//binary conversion of the decimal number//
disp(' binary form of the number obtained by adding
    -29 to 14 ')
disp(a)
//result is displayed//
```

Scilab code Exa 2.41 Ones complement of minus 13

```
//Example 2-41//
//one's complement form of -13//
clc
//clears the window//
clear
//clears all existing variables//
x=bitcmp(13,8)
//complement of 13//
a=dec2bin(x)
//binary conversion of the decimal number//
disp(' ones complement form of -13 ')
disp(a)
//result is displayed//
```

Scilab code Exa 2.42 Ones complement of minus 13 second method

```
//Example 2-42//
//ones complement of -13 by 2nd method//
clc
//clears the window//
clear
//clears all existing variables//
a=bitcmp(0,8)
//decimal equivalent of 11111111//
b=13
c=a-b
//subtracting 13 from decimal equivalent of 11111111//
z=dec2bin(c)
disp('ones complement of -13 is:')
disp(z)
//result is displayed//
```

Scilab code Exa 2.43 Ones complement addition using 4 bits

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

```
a=dec2bin(z)
//binary conversion//
disp('binary form of the number obtained by adding
   −3 to −2')
disp(a)
//result is displayed//
disp('msb is discarded ,4 bit representation is the
   answer in binary form')
a=dec2bin(z-(2^4))
disp(a)
//Final result is displayed//
```

Scilab code Exa 2.44 Ones complement addition using 4 bits

```
//Example 2−44//
//add −3 to 2 in one's complement using 4 bits//
clc
//clears the window//
clear
//clears all the existing variables//
x=2
y=bitcmp(3,4)
//complement of the decimal number 2//
z=x+y
a=dec2bin(z)
//binary conversion//
disp('binary form of the number obtained by adding
   −3 to 2')
disp(a)
//result is displayed//
```

Scilab code Exa 2.45 Ones complement addition using 8 bits

```
//Example 2-45//
//add 3 to -2 in one's complement using 8 bits//
clc
//clears the window//
clear
//clears all the existing variables//
x=3
y=bitcmp(2,8)
//complement of the decimal number 2//
z=x+y+1
//carry is added//
a=dec2bin(z)
//binary conversion//
disp('binary form of the number obtained by adding 3
    to -2')
disp(a)
//result is displayed//
disp('msb is discarded,8 bit representation is the
   answer in binary form')
a=dec2bin(z-(2^8))
disp(a)
//Final result is displayed//
```

Scilab code Exa 2.46 Convert 1101 point 000101 to decimal

```
//Example 2-46//
//Conversion to decimal//
clc
//clears the console//
clear
//clears all existing variables//
p=1
q=1
z=0
b=0
```

```
w=0
f=0
//initialising//
//bin=input (enter the binary number to be converted
    to its decimal form)//
//accepting the input from the user//
bin=1101.000101
d=modulo(bin,1)
//separating the decimal part from the integer part
   //
d=d*10^10
a=floor(bin)
//removing the decimal part//
while(a>0)
//loop to enter the binary bits of the integer part
   into a matrix//
r=modulo(a,10)
b(1,q)=r
a=a/10
a=floor(a)
q=q+1
end
for m=1: q-1
//multiplying each bit of the integer part with its
   corresponding positional value and adding//
c=m-1
f=f+b(1,m)*(2^c)
end
while(d>0)
//loop to take the bits of the decimal part into a
   matrix//
e=modulo(d,2)
w(1,p)=e
d=d/10
d=floor(d)
p=p+1
end
for n=1: p-1
```

```
//multiplying each bit with its corresponding
   positional value and adding//
z=z+w(1,n)*(0.5)^(11-n)
end
z=z*10000
z=round(z)
//rounding off to 4 decimal places//
z=z/10000
x=f+z
disp('the decimal equivalent of the binary number is
    :')
disp(x)
//result is displayed//
```

Scilab code Exa 2.47 Convert point 375 to binary

```
//Example 2-47//
//Conversion of decimal to binary//
clc
//clears the console//
clear
//clears all existing variables//
q=0
b=0
s=0
//initialising//
//a=input(enter the decimal number to be converted
   to its binary form)
//taking input from the user//
a=0.375
d=modulo(a,1)
//separating the decimal part from the integer//
a=floor(a)
//removing the decimal part//
while(a>0)
```

```
//integer part converted to equivalent binary form//
x=modulo(a,2)
b=b+(10^q)*x
a=a/2
a=floor(a)
q=q+1
end
for i=1: 10
//taking values after the decimal part and
    converting to equivalent binary form//
d=d*2
q=floor(d)
s=s+q/(10^i)
if d>=1 then
    d=d-1
end
end
k=b+s
disp('the decimal number in binary form is :')
disp(k)
//result is displayed//
```

Scilab code Exa 2.48 Convert point 54545 to binary

```
//Example 2-48//
//Conversion of decimal to binary//
clc
//clears the console//
clear
//clears all existing variables//
q=0
b=0
s=0
//initialising//
//a=input(enter the decimal number to be converted
```

```
    to its binary form)
//taking input from the user//
a=0.54545
d=modulo(a,1)
//separating the decimal part from the integer//
a=floor(a)
//removing the decimal part//
while(a>0)
//integer part converted to equivalent binary form//
x=modulo(a,2)
b=b+(10^q)*x
a=a/2
a=floor(a)
q=q+1
end
for i=1: 10
//taking values after the decimal part and
    converting to equivalent binary form//
d=d*2
q=floor(d)
s=s+q/(10^i)
if d>=1 then
    d=d-1
end
end
k=b+s
disp('the decimal number in binary form is :')
disp(k)
//result is displayed//
```

Scilab code Exa 2.49 Convert 38 point 21 to binary

```
//Example 2-49//
//Conversion of decimal to binary//
clc
```

```
//clears the console//
clear
//clears all existing variables//
q=0
b=0
s=0
//initialising//
//a=input(enter the decimal number to be converted
   to its binary form)
//taking input from the user//
a=38.21
d=modulo(a,1)
//separating the decimal part from the integer//
a=floor(a)
//removing the decimal part//
while(a>0)
//integer part converted to equivalent binary form//
x=modulo(a,2)
b=b+(10^q)*x
a=a/2
a=floor(a)
q=q+1
end
for i=1: 10
//taking values after the decimal part and
   converting to equivalent binary form//
d=d*2
q=floor(d)
s=s+q/(10^i)
if d>=1 then
    d=d-1
end
end
k=b+s
disp('the integer part of the binary form is :')
disp(b)
disp('the fractional part of the binary form is:')
disp(s)
```

```
//result is displayed//
```

Scilab code Exa 2.50 Binary addition

```
//Example 2-50//
//addition of binary numbers//
//this program requires functions binary2decimal.sci
    and decimal2binary.sci//
clc
//clears the window//
clear
//clears all existing variables//
function x=binary2decimal(bin)
p=1
q=1
z=0
b=0
w=0
f=0
//initialising//
d=modulo(bin,1)
//separating the decimal part from the integer part
   //
d=d*10^10
a=floor(bin)
//removing the decimal part//
while(a>0)
//loop to enter the binary bits of the integer part
   into a matrix//
r=modulo(a,10)
b(1,q)=r
a=a/10
a=floor(a)
q=q+1
end
```

```
for m=1: q-1
//multiplying each bit of the integer part with its
   corresponding positional value and adding//
c=m-1
f=f+b(1,m)*(2^c)
end
while(d>0)
//loop to take the bits of the decimal part into a
   matrix//
e=modulo(d,2)
w(1,p)=e
d=d/10
d=floor(d)
p=p+1
end
for n=1: p-1
//multiplying each bit with its corresponding
   positional value and adding//
z=z+w(1,n)*(0.5)^(11-n)
end
z=z*10000
z=round(z)
//rounding off to 4 decimal places//
z=z/10000
x=f+z
endfunction
function y=decimal2binary(a)
q=0
b=0
s=0
//initialising//
d=modulo(a,1)
//separating the decimal part from the integer//
a=floor(a)
//removing the decimal part//
while(a>0)
//integer part converted to equivalent binary form//
x=modulo(a,2)
```

```
b=b+(10^q)*x
a=a/2
a=floor(a)
q=q+1
end
for i=1: 10
//taking values after the decimal part and
    converting to equivalent binary form//
d=d*2
q=floor(d)
s=s+q/(10^i)
if d>=1 then
    d=d-1
end
end
y=b+s
endfunction
x=binary2decimal(11.011)
y=binary2decimal(10.001)
z=x+y
a=decimal2binary(z)
disp('the addition of the binary numbers is :')
disp(a)
//result is displayed//
```

Scilab code Exa 2.51 Binary mutiplication

```
//Example 2-51//
//multiplication of binary numbers//
//this program requires functions binary2decimal.sci
    and decimal2binary.sci//
clc
//clears the window//
clear
//clears all existing variables//
```

```
function x=binary2decimal(bin)
p=1
q=1
z=0
b=0
w=0
f=0
//initialising//
d=modulo(bin,1)
//separating the decimal part from the integer part//
d=d*10^10
a=floor(bin)
//removing the decimal part//
while(a>0)
//loop to enter the binary bits of the integer part into a matrix//
r=modulo(a,10)
b(1,q)=r
a=a/10
a=floor(a)
q=q+1
end
for m=1: q-1
//multiplying each bit of the integer part with its corresponding positional value and adding//
c=m-1
f=f+b(1,m)*(2^c)
end
while(d>0)
//loop to take the bits of the decimal part into a matrix//
e=modulo(d,2)
w(1,p)=e
d=d/10
d=floor(d)
p=p+1
end
```

```
for n=1: p-1
//multiplying each bit with its corresponding positional value and adding//
z=z+w(1,n)*(0.5)^(11-n)
end
z=z*10000
z=round(z)
//rounding off to 4 decimal places//
z=z/10000
x=f+z
endfunction
function y=decimal2binary(a)
q=0
b=0
s=0
//initialising//
d=modulo(a,1)
//separating the decimal part from the integer//
a=floor(a)
//removing the decimal part//
while(a>0)
//integer part converted to equivalent binary form//
x=modulo(a,2)
b=b+(10^q)*x
a=a/2
a=floor(a)
q=q+1
end
for i=1: 10
//taking values after the decimal part and converting to equivalent binary form//
d=d*2
q=floor(d)
s=s+q/(10^i)
if d>=1 then
    d=d-1
end
end
```

```
y=b+s
endfunction
x=binary2decimal(10.001)
y=binary2decimal(0.11)
z=x*y
a=decimal2binary(z)
disp('the multiplication of the binary numbers is :'
    )
disp(a)
//result is displayed//
```

Chapter 3

Binary Codes

Scilab code Exa 3.1 Add 647 to 492 in BCD

```
//Example 3-1//
//BCD addition//
clc
//clears the console//
clear
//clears all existing variables//
a=647
b=492
for i=1: 3
    x(i)=modulo(a,10)
    a=a/10
    a=floor(a)
    y(i)=modulo(b,10)
    b=b/10
    b=floor(b)
end
d=x(1)+y(1)
db=dec2bin(d)
if d>9 then
    db=dec2bin(d+6)
    db=dec2bin(bin2dec(db)-bin2dec('10000'))
```

```
end
e=x(2)+y(2)
eb=dec2bin(e)
if d>9 then
    eb=dec2bin(e+1)
    e=e+1
    end
if e>9
    eb=dec2bin(e+6)
    eb=dec2bin(bin2dec(eb)-bin2dec('10000'))
end
f=x(3)+y(3)
fb=dec2bin(f)
if e>9 then
    fb=dec2bin(f+1)
    f=f+1
    end
if f>9
    fb=dec2bin(f+6)
    fb=dec2bin(bin2dec(fb)-bin2dec('10000'))
    dc(4)=1
    end
dc(1)=bin2dec(db)
dc(2)=bin2dec(eb)
dc(3)=bin2dec(fb)
z=0
for i=1: 4
    z=z+dc(i)*(10^(i-1))
end
disp(z)
disp('equivalent binary form')
p=strcat(dec2bin(dc(4),1)+dec2bin(dc(3),4)+dec2bin(
   dc(2),4)+dec2bin(dc(1),4))
disp(p)
//answer is displayed//
```

Scilab code Exa 3.2 Add 4318 and 7678 in BCD

```
//Example 3-2//
//BCD addition//
clc
//clears the console//
clear
//clears all existing variables//
a=4318
b=7678
for i=1: 4
    x(i)=modulo(a,10)
    a=a/10
    a=floor(a)
    y(i)=modulo(b,10)
    b=b/10
    b=floor(b)
end
d=x(1)+y(1)
db=dec2bin(d)
if d>9 then
    db=dec2bin(d+6)
    db=dec2bin(bin2dec(db)-bin2dec('10000'))
end
e=x(2)+y(2)
eb=dec2bin(e)
if d>9 then
    eb=dec2bin(e+1)
    e=e+1
    end
if e>9
    eb=dec2bin(e+6)
    eb=dec2bin(bin2dec(eb)-bin2dec('10000'))
end
```

```
f=x(3)+y(3)
fb=dec2bin(f)
if e>9 then
    fb=dec2bin(f+1)
    f=f+1
    end
if f>9
    fb=dec2bin(f+6)
    fb=dec2bin(bin2dec(fb)-bin2dec('10000'))
end
g=x(4)+y(4)
gb=dec2bin(g)
if f>9 then
    gb=dec2bin(g+1)
    g=g+1
    end
if g>9
    gb=dec2bin(g+6)
    gb=dec2bin(bin2dec(gb)-bin2dec('10000'))
    dc(5)=1
end
dc(1)=bin2dec(db)
dc(2)=bin2dec(eb)
dc(3)=bin2dec(fb)
dc(4)=bin2dec(gb)
z=0
for i=1: 5
    z=z+dc(i)*(10^(i-1))
end
disp(z)
disp('equivalent binary form')
p=strcat(dec2bin(dc(5),1)+dec2bin(dc(4),4)+dec2bin(
   dc(3),4)+dec2bin(dc(2),4)+dec2bin(dc(1),4))
disp(p)
//answer is displayed//
```

Scilab code Exa 3.3 Add 3 and 2 in XS3

```
//Example 3-3//
//add 3 and 2 in Excess 3 code//
clc
//clears the console//
clear
//clears all existing variables//
n=3
m=2
z=0
a=0110
b=0101
ea=dec2bin(n+3)
eb=dec2bin(m+3)
for i=1: 4
    x(i)=modulo(a,10)
    a=a/10
    a=floor(a)
    y(i)=modulo(b,10)
    b=b/10
    b=floor(b)
end
for i=1: 4
    g(i)=bitand(x(i),y(i))
    p(i)=bitor(x(i),y(i))
end
c(1)=0
for i=1: 4
    c(i+1)=bitor(g(i),bitand(p(i),c(i)))
end
if c(5)==1 then
          z=dec2bin(bin2dec(ea)+bin2dec(eb)+3)
end
```

```
if c(5)==0 then
          z=dec2bin(bin2dec(ea)+bin2dec(eb)-3)
end
disp('equivalent binary number after excess 3
   addition')
disp(z)
disp('equivalent decimal number')
disp(m+n)
//result is displayed//
```

Scilab code Exa 3.4 Add 6 and 8 in XS3

```
//Example 3-4//
//add 6 and 8 in Excess 3 code//
clc
//clears the console//
clear
//clears all existing variables//
n=6
m=8
a=1001
b=1011
ea=dec2bin(n+3)
eb=dec2bin(m+3)
for i=1: 4
    x(i)=modulo(a,10)
    a=a/10
    a=floor(a)
    y(i)=modulo(b,10)
    b=b/10
    b=floor(b)
end
for i=1: 4
    g(i)=bitand(x(i),y(i))
    p(i)=bitor(x(i),y(i))
```

```
end
c(1)=0
for i=1: 4
    c(i+1)=bitor(g(i),bitand(p(i),c(i)))
end
if c(5)==1 then
       z=dec2bin(bin2dec(ea)+bin2dec(eb)+3)
end
if c(5)==0 then
       z=dec2bin(bin2dec(ea)+bin2dec(eb)-3)
end
disp('equivalent binary number after excess 3
   addition' )
disp(z)
disp('equivalent decimal number')
disp(m+n)
//result is displayed//
```

Scilab code Exa 3.5 Convert binary 1011 to Gray code

```
//Example 3-5//
//binary to gray code//
clc
//clears the console//
clear
//clears all existing variables//
a=1011
for i=1: 4
    x(i)=modulo(a,10)
    a=a/10
    a=floor(a)
end
y(4)=x(4)
k=3
while(k>0)
```

```
    if bitand(x(k+1),x(k))==1 then
        y(k)=bitcmp(1,1)
    else
        y(k)=bitor(x(k+1),x(k))
    end
    k=k-1
end
//display//
z=0
for i=1: 4
    z=z+y(i)*(10^(i-1))
end
disp('equivalent gray code')
disp(z)
//equivalent gray code is displayed//
```

Scilab code Exa 3.6 Convert binary 1001011 to Gray code

```
//Example 3-6//
//binary to gray code//
clc
//clears the console//
clear
//clears all existing variables//
a=1001011
for i=1: 7
    x(i)=modulo(a,10)
    a=a/10
    a=floor(a)
end
y(7)=x(7)
k=6
while(k>0)
    if bitand(x(k+1),x(k))==1 then
        y(k)=bitcmp(1,1)
```

```
    else
        y(k)=bitor(x(k+1),x(k))
    end
    k=k-1
end
//display//
z=0
for i=1: 7
    z=z+y(i)*(10^(i-1))
end
disp('equivalent gray code')
disp(z)
//equivalent gray code is displayed//
```

Scilab code Exa 3.7 Convert Gray code 1011 to binary

```
//Example 3-7//
//gray code to binary//
clc
//clears the console//
clear
//clears all existing variables//
a=1011
for i=1: 4
    x(i)=modulo(a,10)
    a=a/10
    a=floor(a)
end
y(4)=x(4)
k=3
while(k>0)
    if bitand(y(k+1),x(k))==1 then
        y(k)=bitcmp(1,1)
    else
        y(k)=bitor(y(k+1),x(k))
```

```
    end
    k=k-1
end
z=0
for i=1: 4
    z=z+y(i)*(10^(i-1))
end
disp('equivalent binary code')
disp(z)
//equivalent binary code is displayed//
```

Scilab code Exa 3.8 Convert Gray code 1001011 to binary

```
//Example 3-8//
//gray code to binary//
clc
//clears the console//
clear
//clears all existing variables//
a=1001011
for i=1: 7
    x(i)=modulo(a,10)
    a=a/10
    a=floor(a)
end
y(7)=x(7)
k=6
while(k>0)
    if bitand(y(k+1),x(k))==1 then
        y(k)=bitcmp(1,1)
    else
        y(k)=bitor(y(k+1),x(k))
    end
    k=k-1
end
```

```
z=0
for i=1: 7
    z=z+y(i)*(10^(i-1))
end
disp('equivalent binary code:')
disp(z)
//equivalent binary code is displayed//
```

Scilab code Exa 3.9 Hamming code for 1011

```
//Example 3-9//
//Hamming code//
clc
//clears the command window//
clear
//clears all existing variables//
z=1011
//input//
a=0;b=0;c=0;d=0;
//taking the input//
for i=1:7
    x(i)=0
    if i==3 then
       x(i)=1
    end
    if i==5 then
       x(i)=1
    end
    if i==7 then
       x(i)=1
    end
end
//establishing even parity at positions 1,3,5,7//
for i=1: 7
    if x(i)==1 then
```

```
        a=a+1
    end
end
d=modulo(a,2)
if (d==1) then
    x(1)=1
end
//establishing even parity at positions 2,3,6,7//
for i=2: 7
    if (i==5) then
        continue
    end
    if (x(i)==1) then
        b=b+1
    end
end
d=modulo(b,2)
if (d==1) then
    x(2)=1
end
//establishing even parity at positions 4,5,6,7//
for i=5:7
    if (x(i)==1) then
        c=c+1
    end
end
d=modulo(c,2)
if (d==1) then
    x(4)=1
end
//displaying the result//
disp('the hamming code for this data is given as')
for i=1:7
    disp(x(i))
end
```

Scilab code Exa 3.10 Hamming code for 0101

```
//Example 3-10//
//Hamming code//
clc
//clears the command window//
clear
//clears all existing variables//
z=0101
//input//
a=0;b=0;c=0;d=0;
//taking the input//
for i=1:7
    x(i)=0
    if i==3 then
       x(i)=1
    end
    if i==6 then
       x(i)=1
    end
end
//establishing even parity at positions 1,3,5,7//
for i=1: 7
    if i==6 then
       continue
    end
    if x(i)==1 then
       a=a+1
    end
end
d=modulo(a,2)
if (d==1) then
    x(1)=1
end
```

```
//establishing even parity at positions 2,3,6,7//
for i=2: 7
    if x(i)==1 then
        b=b+1
    end
end
d=modulo(b,2)
if (d==1) then
    x(2)=1
end
//establishing even parity at positions 4,5,6,7//
for i=5:7
    if (x(i)==1) then
        c=c+1
    end
end
d=modulo(c,2)
if (d==1) then
    x(4)=1
end
//displaying the result//
disp('the hamming code for this data is given as')
for i=1:7
    disp(x(i))
end
```

Scilab code Exa 3.11 Correction of Hamming code 1111101

```
//Example 3-11//
//Correction of received Hamming code//
clc
//clears the console//
clear
//clears all existing variables//
z=1111101
```

```
//incorrect code taken as input//
a=0;b=0;c=0;d=0;
//taking the input in an array//
for i=1: 7
    x(i)=1
    if i==2 then
       x(i)=0
    end
end
//checking for 4,5,6,7 even parity//
for i=4: 7
    if x(i)==1 then
       a=a+1
    end
end
d=modulo(a,2)
if d==1 then
   disp('wrong entry for 4,5,6,7')
   x(4)=1
else
    disp('correct entry for 4,5,6,7')
end
//checking for 2,3,6,7 even parity//
for i=2: 7
    if i==4 then
       continue
    end
    if i==5 then
       continue
    end
    if x(i)==1 then
       a=a+1
    end
end
d=modulo(a,2)
if d==1 then
   disp('wrong entry for 2,3,6,7')
   x(2)=1
```

```
else
    disp('correct entry for 2,3,6,7')
end
//checking for 1,3,5,7 even parity//
for i=1: 7
    if i==2 then
        continue
    end
    if i==4 then
        continue
    end
    if i==6 then
        continue
    end
    if x(i)==1 then
        a=a+1
    end
end
d=modulo(a,2)
if d==1 then
   disp('correct entry for 1,3,5,7')
   x(1)=1
else
    disp('wrong entry for 1,3,5,7')
end
disp('Therefore,bit 2 is in error and the corrected code is :')
for i=1: 7
    disp(x(i))
end
//correct code is displayed//
```

Chapter 4

Boolean Algebra

Scilab code Exa 4.1 Demorganize a function

```
//Example 4-1//
//demorganize function//
clc
//clears the console//
clear
//clears all existing variables//
//the given function is as follows//
disp('Given function- (AB''+C)'' ')
disp('now demorganizing the function')
disp('complement function')
disp(' AB''+C ')
disp('changing operators')
disp(' (A+B'')(C) ')
disp('complement variables')
disp(' (A''+B)(C'') ')
//final answer displayed after simplification//
```

Scilab code Exa 4.2 Reduce an expression

```
//Example 4-2//
//Reducing an expression//
clc
//clears the console//
clear
//clears all existing variables//
//the given expression is as follows//
disp(' Given Expression- A+B(C+(DE)'')'' ')
disp('applying DeMorgan theorem')
disp(' reduced expression: ')
disp(' A+B(C+D''+E'')'' ')
disp('Applying DeMorgan theorem again')
disp(' A+B(C''DE) ')
disp(' Therefore final reduced expression is : ')
disp(' A+BC''DE ')
//final expression displayed//
```

Scilab code Exa 4.3 Reduce an expression

```
//Example 4-3//
//Reducing a given expression//
clc
//clears the console//
clear
//clears all existing variables//
//the given expression is as follows//
disp(' Given Expression- ((AB)''+A''+AB)'' ')
disp('applying DeMorgan Theorem')
disp(' reduced expression ')
disp(' (A''+B''+A''+AB)'' ')
disp('as A+A=A')
disp(' (A''+B''+AB)'' ')
//By law 20//
disp(' (A''+B''+A)'' ')
disp('rearranging')
```

```
disp(' (A+A''+B'')'' ')
//by law 13//
disp(' (1+B'')'' ')
//by law 11//
disp(' 1'' ')
disp(' 0 ')
//final reduced expression is displayed/
```

Scilab code Exa 4.4 Reduce a boolean expression

```
//Example 4-4//
//Reducing a given expression//
clc
//clears the window//
clear
//clears all existing variables//
//the given expression is as follows//
disp(' Given Expression- AB+(AC)''+AB''C(AB+C) ')
disp('Multiplying')
disp(' AB+(AC)''+AABB''C+AB''CC ')
//using laws 18,6,8,9//
disp(' AB+(AC)''+AB''C ')
disp('applying DeMorgan theorem')
disp(' AB+A''+C''+AB''C ')
disp('rearrange')
disp(' AB+C''+A''+AB''C ')
//reduce using law 20//
disp(' AB+C''+A''+B''C ')
disp('rearranging again')
disp(' A''+AB+C''+B''C ')
//reduce using law 20//
disp(' A''+B+C''+B'' ')
//using laws 11 and 13//
disp(' 1 ')
//final reduced expression is displayed//
```

Scilab code Exa 4.5 Reduce a Boolean expression

```
//Example 4-5//
//Reduce a given expression//
clc
//clears the window//
clear
//clears all existing variables//
//the given expression is as follows//
disp(' Given Expression- ((AB''+ABC)''+A(B+AB''))'' ')
disp('factorise')
disp(' ((A(B''+BC))''+A(B+AB''))'' ')
//reduce using laws 18 and 20//
disp(' ((A(B''+C))''+A(B+A))'' ')
disp('multiply')
disp(' ((AB''+AC)''+AA+AB)'' ')
//reduce using laws 7,8,11,18//
disp(' ((AB''+AC)''+A)'' ')
disp('demorganize')
disp(' ((A''+B)(A''+C'')+A)'' ')
disp('multiply')
disp(' (A''A''+A''C''+A''B+BC''+A)'' ')
//Reduce using laws 18,8//
disp(' (A''(1+C''+B)+BC''+A)'' ')
//reduce using law 18,7,11//
disp(' 1'' ')
disp(' 0 ')
//final reduced expression is displayed//
```

Chapter 6

Combinational Logic

Scilab code Exa 6.1 Convert A plus B to minterms

```
//Example 6-1//
//convert A+B to minterms//
clc
//clears the console//
clear
//clears all existing variables//
//conversion to minterms//
disp(' Given expression- A+B  ')
disp(' on solving ')
disp(' A(B+B'')+B(A+A'') ')
disp('multiplying')
disp(' AB+AB''+AB+A''B  ')
disp('we know A+A=A')
disp(' AB+AB''+A''B  ')
//final result is displayed//
```

Scilab code Exa 6.2 Find minterms for A plus BC

```
//Example 6-2//
//find minterms for A+BC//
clc
//clears the console//
clear
//clears all existing variables//
//conversion to minterms//
disp('Given expression- A+BC ')
disp(' on solving ')
disp(' A(B+B'')(C+C'')+BC(A+A'')   ')
disp(' (AB+AB'')(C+C'')+BCA+BCA''   ')
disp('multiplying')
disp(' C''AB+AB''C+AB''C+AB''C''+BCA+BCA''  ')
//final result is displayed//
```

Scilab code Exa 6.3 Find minterms for AB plus ACD

```
//Example 6-3//
//find minterms for AB+ACD//
clc
//clears the window//
clear
//clears all existing variables//
//conversion to minterms//
disp(' Given expression- AB+ACD ')
disp(' on solving ')
disp(' AB(C+C'')(D+D'')+ACD(B+B'') ')
disp('multiplying')
disp(' ABCD+ABC''D''+ABC''D+ABCD''+AB''CD+ABCD   ')
disp('we know A+A=A')
disp(' ABC''D''+ABC''D+ABCD''+AB''CD+ABCD   ')
//result is displayed//
```

Scilab code Exa 6.4 Minterm designation

```
//Example 6-4//
//minterm designation of AB''C''D''//
clc
//clears the window//
clear
//clears all existing variables//
disp('copy original term')
disp(' AB''C''D''  ')
disp(' substitute ones for nonbarred letters and
   zeroes for barred letters ')
disp('after substitution')
disp(' 1000 ')
a=bin2dec('1000')
disp('the decimal equivalent of 1000 is:')
disp(a)
disp(' Therefore decimal subscript of m is 8 ')
disp(' AB''C''D'' = m8  ')
//result is displayed//
```

Scilab code Exa 6.5 Minterm designation

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

```
a=bin2dec('0010')
disp('therefore the decimal equivalent of 0010 is:')
disp(a)
disp(' Therefore decimal subscript of m is 2 ')
disp(' W''X''YZ'' = m2  ')
//result is displayed//
```

Scilab code Exa 6.6 2 variable mapping

```
//Example 6-6//
//map C=A''B''+AB''//
clc
//clears the console//
clear
//clears all existing variables//
//Mapping the expression//
disp('    B'' B ')
disp('A'' 1   0 ')
disp('A   1   0  ')
disp(' From the map, high outputs for 0 and 2 ')
a=[0 0 ; 1 0]
disp(a)
for i=1: 2
    if a(i,1)==1 then
        b(i,1)='A'
   else
        b(i,1)='A'''
    end
    if a(i,2)==1 then
        b(i,2)='B'
   else
        b(i,2)='B'''
    end
end
m=strcat([b(1,1),b(1,2)])
```

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

Scilab code Exa 6.7 3 variable mapping

```
//Example 6-7//
//Map X=ABC+AB''C+AB''C''//
clc
//clears the console//
clear
//clears all existing variables//
//Mapping the expression//
disp('     B''C''  B''C  BC  BC''  ')
disp('A''    0      0     0   0  ')
disp('A     1      1     1   0  ')
disp(' From the map, high outputs for 4,5 and 7 ')
a=[1 0 0 ; 1 0 1 ; 1 1 1]
disp(a)
for i=1: 3
    if a(i,1)==1 then
       b(i,1)='A'
   else
       b(i,1)='A'''
    end
    if a(i,2)==1 then
       b(i,2)='B'
   else
       b(i,2)='B'''
    end
     if a(i,3)==1 then
       b(i,3)='C'
```

```
    else
        b(i,3)='C'''
     end
end
disp(' evaluating expression from truth table and map ')
l=strcat([ b(1,1),b(1,2),b(1,3)])
m=strcat([ b(2,1),b(2,2),b(2,3)])
n=strcat([ b(3,1),b(3,2),b(3,3)])
x=strcat([l"+",m"+",n ])
disp(x)
//Expression is displayed//
```

Scilab code Exa 6.8 Mapping an expression

```
//Example 6-8//
//Mapping an equation//
clc
//clears the console//
clear
//clears all existing variables//
//Mapping the expression//
disp('Z=m(9,20,21,29,30,31)')
disp('                    A''                    A         ')
disp('        D''E'' D''E DE DE''                D''E'' D''E DE DE''')
disp('B''C''    0       0      0    0                       0      0     0    0    ')
disp('BC''      0       0      0    0                       1      1     0    0    ')
disp('BC        0       0      0    0                       0      1     1    1    ')
disp('BC''      0       1      0    0                       0      0     0    0    ')
```

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

Scilab code Exa 6.9 Mapping an expression

```
//Example 6−8//
//Mapping an equation//
clc
//clears the console//
clear
//clears all existing variables//
//Mapping the expression//
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,(
    ')
disp('                         B''
                                    B               ')
disp('              E''F''  E''F  EF  EF''              E''F
    ''  E''F  EF  EF''')
disp('    C''D''     1      1     1   0                     0
        0     0   0     ')
disp('A''  CD''      0      1     1   0                     0
        1     1   1     ')
disp('    CD         0      0     0   0                     0
        0     0   1     ')
disp('    CD''       0      0     1   1                     1
        0     0   1     ')
disp('

    ')
disp('    C''D''     1      0     1   1                     0
        0     1   1     ')
disp('A   CD''       0      0     0   0                     1
        0     0   0     ')
```

```
disp('     CD          0        0      1    1                         1
        1     0    0     ')
disp('     CD''        1        1      0    0                         0
        0     0    0     ')
disp(' From the map, high outputs for
    0,1,3,5,7,10,11,21,22,23,24,26,30,32,34,35,40,41,46,47,50,51,52,6
    ')
//Therefore the kmap is displayed//
```

Scilab code Exa 6.10 4 variable mapping

```
//Example 6-10//
//Map L=W''X''YZ+WX''YZ''+WX''Y''Z''+W''XYZ//
clc
//clears the console//
clear
//clears all existing variables//
//Mapping the expression//
disp('        C''''D'' C''''D CD CD'' ')
disp('A''''B''    0       0      1    0  ')
disp('AB''       0       0      1    0  ')
disp('AB        0       0      0    0  ')
disp('AB''       1       0      0    1  ')
disp(' From the map, high outputs for 3,7,8,10 ')
a=[0 0 1 1 ; 0 1 1 1 ; 1 0 0 0 ; 1 0 1 0]
disp(a)
for i=1: 4
    if a(i,1)==1 then
        b(i,1)='W'
   else
        b(i,1)='W'''
    end
    if a(i,2)==1 then
        b(i,2)='X'
   else
```

```
        b(i,2)='X'''
     end
      if a(i,3)==1 then
        b(i,3)='Y'
    else
        b(i,3)='Y'''
     end
     if a(i,4)==1 then
        b(i,4)='Z'
    else
        b(i,4)=' Z'''
     end
end
disp(' evaluating expression from truth table and map ')
l=strcat([ b(1,1),b(1,2),b(1,3),b(1,4)])
m=strcat([ b(2,1),b(2,2),b(2,3),b(2,4)])
n=strcat([ b(3,1),b(3,2),b(3,3),b(3,4)])
o=strcat([ b(4,1),b(4,2),b(4,3),b(4,4)])
x=strcat([l"+",m"+",n"+",o ])
disp(x)
//Expression is displayed//
```

Scilab code Exa 6.11 Reduce an expression by Kmap

```
//Example 6-11//
//reduce expression using k-map//
clc
//clears the window//
clear
//clears all existing variables//
//Mapping the expression//
disp('          C''D''  C''D  CD  CD''  ')
disp('A''B''     1        1       1   1  ')
disp('AB''       0        0       1   1  ')
```

```
disp('AB        0      1     1   0  ')
disp('AB''      0      0     0   0  ')
disp(' From the map, high outputs for
   0,1,2,3,6,7,13,15  ')
//given logic equation//
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]
disp(a)
for i=1: 8
    if a(i,1)==1 then
       b(i,1)='A'
   else
       b(i,1)='A'''
    end
    if a(i,2)==1 then
       b(i,2)='B'
   else
       b(i,2)='B'''
    end
     if a(i,3)==1 then
       b(i,3)='C'
   else
       b(i,3)='C'''
    end
    if a(i,4)==1 then
       b(i,4)='D'
   else
       b(i,4)=' D'''
    end
end
disp(' evaluating expression from truth table and
   map ')
x1=strcat([ b(1,1),b(1,2),b(1,3),b(1,4) ])
x2=strcat([ b(2,1),b(2,2),b(2,3),b(2,4) ])
x3=strcat([ b(3,1),b(3,2),b(3,3),b(3,4) ])
x4=strcat([ b(4,1),b(4,2),b(4,3),b(4,4) ])
x5=strcat([ b(5,1),b(5,2),b(5,3),b(5,4) ])
x6=strcat([ b(6,1),b(6,2),b(6,3),b(6,4) ])
```

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

Scilab code Exa 6.12 Reduce an expression using Kmap

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

```
      if a(i,2)==1 then
          b(i,2)='B'
    else
          b(i,2)='B'''
      end
       if a(i,3)==1 then
          b(i,3)='C'
    else
          b(i,3)='C'''
      end
      if a(i,4)==1 then
          b(i,4)='D'
    else
          b(i,4)=' D'''
      end
end
disp(' evaluating expression from truth table and
    map ')
x1=strcat([ b(1,1),b(1,2),b(1,3),b(1,4) ])
x2=strcat([ b(2,1),b(2,2),b(2,3),b(2,4) ])
x3=strcat([ b(3,1),b(3,2),b(3,3),b(3,4) ])
x4=strcat([ b(4,1),b(4,2),b(4,3),b(4,4) ])
x5=strcat([ b(5,1),b(5,2),b(5,3),b(5,4) ])
x6=strcat([ b(6,1),b(6,2),b(6,3),b(6,4) ])
x7=strcat([ b(7,1),b(7,2),b(7,3),b(7,4) ])
x8=strcat([ b(8,1),b(8,2),b(8,3),b(8,4) ])
x9=strcat([ b(9,1),b(9,2),b(9,3),b(9,4) ])
x10=strcat([ b(10,1),b(10,2),b(10,3),b(10,4) ])
x=([x1"+",x2"+",x3"+",x4"+",x5"+",x6"+",x7"+",x8"+",
    x9"+",x10 ])
disp(x)
//Expression is displayed//
disp('now reducing expression using boolean algebra'
    )
disp('A''B''C+BD+AB+AD')
```

Scilab code Exa 6.13 Reduce expression by kmap

```
//Example 6-13//
//reduce expression using k-map//
clc
//clears the console//
clear
//clears all existing variables//
//Mapping the expression//
disp('     B''C'' B''C BC BC'' ')
disp('A''    1     0    1  1 ')
disp('A      1     1    1  0 ')
disp(' From the map, high outputs for 0,2,3,4,5 and
   7 ')
a=[0 0 0 ; 0 1 0 ; 0 1 1 ; 1 0 0 ; 1 0 1 ;1 1 1]
disp(a)
for i=1: 6
    if a(i,1)==1 then
       b(i,1)='A'
   else
       b(i,1)='A'''
    end
    if a(i,2)==1 then
       b(i,2)='B'
   else
       b(i,2)='B'''
    end
     if a(i,3)==1 then
       b(i,3)='C'
   else
       b(i,3)='C'''
    end
end
disp(' evaluating expression from truth table and
```

```
map ')
x1=strcat([ b(1,1),b(1,2),b(1,3) ])
x2=strcat([ b(2,1),b(2,2),b(2,3) ])
x3=strcat([ b(3,1),b(3,2),b(3,3) ])
x4=strcat([ b(4,1),b(4,2),b(4,3) ])
x5=strcat([ b(5,1),b(5,2),b(5,3) ])
x6=strcat([ b(6,1),b(6,2),b(6,3) ])
x=([x1"+",x2"+",x3"+",x4"+",x5"+",x6])
disp(x)
//Expression is displayed//
disp('applying laws of boolean algebra')
disp('AC+A''B+B''C''')
```

Scilab code Exa 6.14.a Inputs Required

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

Scilab code Exa 6.14.b Inputs required

```
//example 6-14b//
//Number of inputs required//
```

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

Scilab code Exa 6.14.c Inputs required

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

Scilab code Exa 6.14.d Inputs required

```
//example 6-14d//
//Number of inputs required//
clc
```

```
//clears the window//
clear
//clears all existing variables//
disp(' (A+B''+C)(A''+D)(B+D'') ')
disp('count OR inputs')
disp('3+2+2=7')
disp('count AND inputs')
disp('1+1+1=3')
disp('therefore total inputs=10')
//result is displayed//
```

Scilab code Exa 6.15 Minimise expression using mapping

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

```
2,3,6,7,9,10,12,13,14,16,18,22,26,28,29,30 ')
//Therefore the kmap is displayed//
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]
disp(a)
for i=1: 16
     if a(i,1)==1 then
        b(i,1)='A'
    else
        b(i,1)='A'''
     end
     if a(i,2)==1 then
        b(i,2)='B'
    else
        b(i,2)='B'''
     end
      if a(i,3)==1 then
        b(i,3)='C'
    else
        b(i,3)='C'''
     end
     if a(i,4)==1 then
        b(i,4)='D'
    else
        b(i,4)=' D'''
     end
      if a(i,5)==1 then
        b(i,5)='E'
    else
        b(i,5)=' E'''
     end
end
disp(' evaluating expression from truth table and
    map ')
x1=strcat([ b(1,1),b(1,2),b(1,3),b(1,4),b(1,5) ])
x2=strcat([ b(2,1),b(2,2),b(2,3),b(2,4),b(2,5) ])
```

```
x3=strcat([ b(3,1),b(3,2),b(3,3),b(3,4),b(3,5) ])
x4=strcat([ b(4,1),b(4,2),b(4,3),b(4,4),b(4,5) ])
x5=strcat([ b(5,1),b(5,2),b(5,3),b(5,4),b(5,5) ])
x6=strcat([ b(6,1),b(6,2),b(6,3),b(6,4),b(6,5) ])
x7=strcat([ b(7,1),b(7,2),b(7,3),b(7,4),b(7,5) ])
x8=strcat([ b(8,1),b(8,2),b(8,3),b(8,4),b(8,5) ])
x9=strcat([ b(9,1),b(9,2),b(9,3),b(9,4),b(9,5) ])
x10=strcat([ b(10,1),b(10,2),b(10,3),b(10,4),b(10,5) ])
x11=strcat([ b(11,1),b(11,2),b(11,3),b(11,4),b(11,5) ])
x12=strcat([ b(12,1),b(12,2),b(12,3),b(12,4),b(12,5) ])
x13=strcat([ b(13,1),b(13,2),b(13,3),b(13,4),b(13,5) ])
x14=strcat([ b(14,1),b(14,2),b(14,3),b(14,4),b(14,5) ])
x15=strcat([ b(15,1),b(15,2),b(15,3),b(15,4),b(15,5) ])
x16=strcat([ b(16,1),b(16,2),b(16,3),b(16,4),b(16,5) ])
x=([x1"+",x2"+",x3"+",x4"+",x5"+",x6"+",x7"+",x8"+",
   x9"+",x10"+",x11"+",x12"+",x13"+",x14"+",x15"+",
   x16 ])
disp(x)
disp('Reduced expression')
disp('A''BD''E+AB''C''E''+A''B''D+BCD''+DE''')
//Expression is displayed//
```

Scilab code Exa 6.16 Reduce using mapping

```
//Example 6-16//
//Reduce using mapping//
clc
//clears the console//
```

```
clear
//clears all existing variables//
//Mapping the expression//
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
    ')
disp('                          B''
                                                 B              ')
disp('               E''F'' E''F EF EF''                 E''F
    '' E''F EF EF''')
disp('     C''D''    1       0      0   1                       1
        0    0   1     ')
disp('A'' CD''       1       0      0   0                       1
        0    1   0     ')
disp('     CD         0       1      1   0                       0
        0    0   0     ')
disp('     CD''       1       0      0   1                       1
        0    0   1     ')
disp('


    ')
disp('     C''D''    1       0      0   1                       1
        0    0   1     ')
disp('A   CD''       0       0      0   0                       0
        0    0   0     ')
disp('     CD         0       1      1   0                       1
        1    0   0     ')
disp('     CD''       1       0      0   1                       1
        1    0   1     ')
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
     ')
//Therefore the kmap is displayed//
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  ]
for i=1: 26
    if a(i,1)==1 then
       b(i,1)='A'
   else
       b(i,1)='A'''
    end
    if a(i,2)==1 then
       b(i,2)='B'
   else
       b(i,2)='B'''
    end
     if a(i,3)==1 then
       b(i,3)='C'
   else
       b(i,3)='C'''
    end
    if a(i,4)==1 then
       b(i,4)='D'
   else
       b(i,4)=' D'''
    end
     if a(i,5)==1 then
       b(i,5)='E'
   else
       b(i,5)=' E'''
    end
    if a(i,6)==1 then
       b(i,6)='F'
   else
       b(i,6)=' F'''
    end
end
disp(' evaluating expression from truth table and
   map ')
x1=strcat([ b(1,1),b(1,2),b(1,3),b(1,4),b(1,5),b
   (1,6) ])
```

```
x2=strcat([ b(2,1),b(2,2),b(2,3),b(2,4),b(2,5),b(2,6) ])
x3=strcat([ b(3,1),b(3,2),b(3,3),b(3,4),b(3,5),b(3,6) ])
x4=strcat([ b(4,1),b(4,2),b(4,3),b(4,4),b(4,5),b(4,6) ])
x5=strcat([ b(5,1),b(5,2),b(5,3),b(5,4),b(5,5),b(5,6) ])
x6=strcat([ b(6,1),b(6,2),b(6,3),b(6,4),b(6,5),b(6,6) ])
x7=strcat([ b(7,1),b(7,2),b(7,3),b(7,4),b(7,5),b(7,6) ])
x8=strcat([ b(8,1),b(8,2),b(8,3),b(8,4),b(8,5),b(8,6) ])
x9=strcat([ b(9,1),b(9,2),b(9,3),b(9,4),b(9,5),b(9,6) ])
x10=strcat([ b(10,1),b(10,2),b(10,3),b(10,4),b(10,5),b(10,6) ])
x11=strcat([ b(11,1),b(11,2),b(11,3),b(11,4),b(11,5),b(11,6) ])
x12=strcat([ b(12,1),b(12,2),b(12,3),b(12,4),b(12,5),b(12,6) ])
x13=strcat([ b(13,1),b(13,2),b(13,3),b(13,4),b(13,5),b(13,6) ])
x14=strcat([ b(14,1),b(14,2),b(14,3),b(14,4),b(14,5),b(14,6) ])
x15=strcat([ b(15,1),b(15,2),b(15,3),b(15,4),b(15,5),b(15,6) ])
x16=strcat([ b(16,1),b(16,2),b(16,3),b(16,4),b(16,5),b(16,6) ])
x17=strcat([ b(17,1),b(17,2),b(17,3),b(17,4),b(17,5),b(17,6) ])
x18=strcat([ b(18,1),b(18,2),b(18,3),b(18,4),b(18,5),b(18,6) ])
x19=strcat([ b(19,1),b(19,2),b(19,3),b(19,4),b(19,5),b(19,6) ])
x20=strcat([ b(20,1),b(20,2),b(20,3),b(20,4),b(20,5),b(20,6) ])
```

```
x21=strcat([ b(21,1),b(21,2),b(21,3),b(21,4),b(21,5)
    ,b(21,6) ])
x22=strcat([ b(22,1),b(22,2),b(22,3),b(22,4),b(22,5)
    ,b(22,6) ])
x23=strcat([ b(23,1),b(23,2),b(23,3),b(23,4),b(23,5)
    ,b(23,6) ])
x24=strcat([ b(24,1),b(24,2),b(24,3),b(24,4),b(24,5)
    ,b(24,6) ])
x25=strcat([ b(25,1),b(25,2),b(25,3),b(25,4),b(25,5)
    ,b(25,6) ])
x26=strcat([ b(26,1),b(26,2),b(26,3),b(26,4),b(26,5)
    ,b(26,6) ])
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 ])
disp(x)
disp('Reduced expression')
disp('A''BC''DEF+A''C''E''F''+B''CDF+ABCE''+D''F''')
//Expression is displayed//
```

Scilab code Exa 6.17 Kmap POS SOP

```

//Example 6-17//
//reduce expression using k-map by both POS and SOP
    //
clc
//clears the window//
clear
//clears all existing variables//
//Mapping the expression//
disp('        C''D''  C''D  CD  CD''  ')
disp('A''B''    1       1     1   0  ')
disp('AB''      0       0     1   1  ')
```

```
disp('AB        0       1      1   0  ')
disp('AB''      1       1      0   0  ')
disp(' From the map, high outputs for
   0,1,3,6,7,8,9,13,15 ')
disp(' From the map, low outputs for
   2,4,5,10,11,12,14 ')
//given logic equation//
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]
disp(a)
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]
disp(c)
for i=1: 9
    if a(i,1)==1 then
       b(i,1)='A'
   else
       b(i,1)='A'''
    end
    if a(i,2)==1 then
       b(i,2)='B'
   else
       b(i,2)='B'''
    end
     if a(i,3)==1 then
       b(i,3)='C'
   else
       b(i,3)='C'''
    end
    if a(i,4)==1 then
       b(i,4)='D'
   else
       b(i,4)=' D'''
    end
end
for i=1: 7
    if c(i,1)==1 then
       d(i,1)='A'''
```

```
    else
        d(i,1)='A'
     end
     if c(i,2)==1 then
        d(i,2)='B'''
    else
        d(i,2)='B'
     end
      if c(i,3)==1 then
        d(i,3)='C'''
    else
        d(i,3)='C'
     end
     if c(i,4)==1 then
        d(i,4)='D'''
    else
        d(i,4)=' D '
     end
end
disp(' evaluating expression(minterms)from truth table and map ')
x1=strcat([ b(1,1),b(1,2),b(1,3),b(1,4) ])
x2=strcat([ b(2,1),b(2,2),b(2,3),b(2,4) ])
x3=strcat([ b(3,1),b(3,2),b(3,3),b(3,4) ])
x4=strcat([ b(4,1),b(4,2),b(4,3),b(4,4) ])
x5=strcat([ b(5,1),b(5,2),b(5,3),b(5,4) ])
x6=strcat([ b(6,1),b(6,2),b(6,3),b(6,4) ])
x7=strcat([ b(7,1),b(7,2),b(7,3),b(7,4) ])
x8=strcat([ b(8,1),b(8,2),b(8,3),b(8,4) ])
x9=strcat([ b(9,1),b(9,2),b(9,3),b(9,4) ])
x=([x1"+",x2"+",x3"+",x4"+",x5"+",x6"+",x7"+",x8"+",x9 ])
disp(x)
disp(' evaluating expression(maxterms)from truth table and map ')
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) ')
```

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

Scilab code Exa 6.18 Minimise expression

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

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

Scilab code Exa 6.19 Reduce kmap by POS

```

//Example 6-19//
//reduce expression using k-map by finding POS//
clc
//clears the window//
clear
//clears all existing variables//
//Mapping the expression//
disp('        C''D''  C''D CD CD''  ')
disp('A''B''    0      0    0   1  ')
disp('AB''      1      0    0   0  ')
disp('AB        1      1    0   0  ')
disp('AB''      1      1    1   0  ')
disp(' From the map, low outputs for
    0,1,3,5,6,7,10,14,15')
//given logic equation//
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]
disp(c)
for i=1: 9
    if c(i,1)==1 then
        d(i,1)='A'''
   else
        d(i,1)='A'
    end
    if c(i,2)==1 then
        d(i,2)='B'''
```

```
    else
        d(i,2)='B'
     end
      if c(i,3)==1 then
        d(i,3)='C'''
    else
        d(i,3)='C'
     end
     if c(i,4)==1 then
        d(i,4)='D'''
    else
        d(i,4)='D'
     end
end
disp(' evaluating expression from truth table and
    map')
l=strcat([ d(1,1),d(1,2),d(1,3),d(1,4)])
m=strcat([ d(2,1),d(2,2),d(2,3),d(2,4)])
n=strcat([ d(3,1),d(3,2),d(3,3),d(3,4)])
o=strcat([ d(4,1),d(4,2),d(4,3),d(4,4)])
x=strcat([l"+",m"+",n"+",o ])
disp('the sum of product expression is:')
disp(x)
disp('Reading the SOP form')
disp(' A''B''CD''+BC''D''+AB''D+AC'' ')
//Expression is displayed//
disp('now reducing expression using boolean algebra
    from POS')
disp(' (A+B+C)(A''+C''+D)(A+D'')(B''+C'') ')
disp('POS has 14 inputs,SOP has 16 inputs')
disp('Therefore,POS form is less expensive')
```

Scilab code Exa 6.20 Find minimum of expression

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

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

Scilab code Exa 6.24 Multiple output equation using mapping

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

```
//Solve multiple output equation using mapping//
clc
//clears the window//
clear
//clears all existing variables//
disp('f1=Sigma m(0,1,2,4,6,7,10,14,15)')
//First function is displayed//
disp('f2=Sigma m(3,4,5,9,10,11,14) ')
//Second function is displayed//
disp('f1.f2=Sigma m(4,10,14)')
//Taking the common entries//
disp('Mapping for f1.f2')
disp('        C''D''  C''D  CD  CD''  ')
disp('A''B''    0      0      0   0  ')
disp('AB''      1      0      0   0  ')
disp('AB        0      0      0   1  ')
disp('AB''      0      0      0   1  ')
disp(' From the map, high outputs for 4,10,14')
//given logic equation//
a=[0 1 0 0;1 0 1 0;1 1 1 0]
disp(a)
for i=1: 3
    if a(i,1)==1 then
       b(i,1)='A'
   else
       b(i,1)='A'''
    end
    if a(i,2)==1 then
       b(i,2)='B'
   else
       b(i,2)='B'''
    end
     if a(i,3)==1 then
       b(i,3)='C'
   else
       b(i,3)='C'''
    end
    if a(i,4)==1 then
```

```
        b(i,4)='D'
    else
        b(i,4)=' D'' '
     end
end
disp(' evaluating expression from truth table and
   map ')
l=strcat([ b(1,1),b(1,2),b(1,3),b(1,4)])
m=strcat([ b(2,1),b(2,2),b(2,3),b(2,4)])
n=strcat([ b(3,1),b(3,2),b(3,3),b(3,4)])
x=strcat([l"+",m"+",n])
disp(x)
//Expression is displayed//
disp('now reducing expression using boolean algebra'
   )
disp('ACD''+A''BC''D''')
```

Scilab code Exa 6.25 Multiple output equation using mapping

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

```
disp('AB        0       0      X   1  ')
disp('AB''      0       0      0   0  ')
disp(' From the map, high outputs for 1,7,14 and
   dont cares for 5,15')
//given logic equation//
a=[0 0 0 1;0 1 0 1;0 1 1 1;1 1 1 0;1 1 1 1]
disp(a)
for i=1: 5
    if a(i,1)==1 then
       b(i,1)='A'
   else
       b(i,1)='A'''
    end
    if a(i,2)==1 then
       b(i,2)='B'
   else
       b(i,2)='B'''
    end
     if a(i,3)==1 then
       b(i,3)='C'
   else
       b(i,3)='C'''
    end
    if a(i,4)==1 then
       b(i,4)='D'
   else
       b(i,4)=' D''  '
    end
end
disp(' evaluating expression from truth table and
   map ')
l=strcat([ b(1,1),b(1,2),b(1,3),b(1,4)])
m=strcat([ b(2,1),b(2,2),b(2,3),b(2,4)])
n=strcat([ b(3,1),b(3,2),b(3,3),b(3,4)])
o=strcat([ b(4,1),b(4,2),b(4,3),b(4,4)])
p=strcat([ b(5,1),b(5,2),b(5,3),b(5,4)])
x=strcat([l"+",m"+",n"+",o"+",p ])
disp(x)
```

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

Scilab code Exa 6.26 Reduce by variable mapping

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

Scilab code Exa 6.27 Reduce by variable mapping

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

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

Scilab code Exa 6.28 Solve using 3 variable mapping

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

Scilab code Exa 6.29 Reduce by mapping

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

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

Scilab code Exa 6.30 Reducing expression by Kmap

```
//Example 6-30//
//reduce expression using k-map//
clc
//clears the window//
clear
//clears all existing variables//
//Mapping the expression//
disp('         C''D''  C''D CD  CD''  ')
disp('A''B''     1       0      1   1  ')
disp('AB''       0       0      1   1  ')
disp('AB         0       1      0   0  ')
disp('AB''       1       1      0   1  ')
disp(' From the map, high outputs for
   0,2,3,6,7,8,9,10,13 ')
//given logic equation//
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]
disp(a)
for i=1: 9
    if a(i,1)==1 then
       b(i,1)='A'
   else
       b(i,1)='A'''
    end
    if a(i,2)==1 then
       b(i,2)='B'
   else
```

```
        b(i,2)='B'''
     end
      if a(i,3)==1 then
        b(i,3)='C'
    else
        b(i,3)='C'''
     end
     if a(i,4)==1 then
        b(i,4)='D'
    else
        b(i,4)=' D'''
     end
end
disp(' evaluating expression from truth table and map ')
x1=strcat([ b(1,1),b(1,2),b(1,3),b(1,4) ])
x2=strcat([ b(2,1),b(2,2),b(2,3),b(2,4) ])
x3=strcat([ b(3,1),b(3,2),b(3,3),b(3,4) ])
x4=strcat([ b(4,1),b(4,2),b(4,3),b(4,4) ])
x5=strcat([ b(5,1),b(5,2),b(5,3),b(5,4) ])
x6=strcat([ b(6,1),b(6,2),b(6,3),b(6,4) ])
x7=strcat([ b(7,1),b(7,2),b(7,3),b(7,4) ])
x8=strcat([ b(8,1),b(8,2),b(8,3),b(8,4) ])
x9=strcat([ b(9,1),b(9,2),b(9,3),b(9,4) ])
x=([x1"+",x2"+",x3"+",x4"+",x5"+",x6"+",x7"+",x8"+",x9 ])
disp(x)
//Expression is displayed//
disp('now reducing expression using boolean algebra')
disp('B''D''+A''C+AC''D')
```

Chapter 11

Analog Digital Conversion

Scilab code Exa 11.1 Percentage resolution of 5bit DA converter

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

Scilab code Exa 11.2 6 bit analog to digital converter

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

Scilab code Exa 11.3 Compute the gain of an Opamp

```
//Example 11-3//
//compute gain of an op-amp//
clc
//clears the console//
clear
//clears all existing variables//
Rf=40
Rs=5
```

```
Ein=3.2
//Given the resistance values in kilo ohms and the
    input voltage in volts//
Av=Rf/Rs
//Av denotes the gain of the op-amp//
Eo=Av*Ein
//Eo denotes the output voltage//
disp('The voltage gain of the given op-amp is: ')
disp(Av)
disp('The output voltage of the opamp in volts is: '
    )
disp(Eo)
//results are displayed//
```

Scilab code Exa 11.4 Compute the output voltage

```
//Example 11-4//
//compute output voltage for circuit in fig 11-12//
clc
//clears the window//
clear
//clears all existing varibles//
R1=10
R2=10
R3=10
Rf=10
E1=3
E2=-2
E3=-4
//Given all the resistance values in kilo ohms and
    the voltages in volts at the inputs//
disp('the current throught resistor R1 in milli amps
     is: ')
I1=E1/R1
disp(I1)
```

```
disp('the current throught resistor R2 in milli amps
    is: ')
I2=E2/R2
disp(I2)
disp('the current throught resistor R3 in milli amps
    is: ')
I3=E3/R3
disp(I3)
Eo=-(I1+I2+I3)*Rf
//Eo denotes the output voltage//
disp('The output voltage in volts of the circuit is:
    ')
disp(Eo)
//answer is displayed//
```

Scilab code Exa 11.5 Compute output voltage

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

```
IA=0
//switch A is open//
disp(IA)
disp('the current throught resistor RB in milli amps
    is: ')
IB=E/RB
disp(IB)
disp('the current throught resistor RC in milli amps
    is: ')
IC=E/RC
disp(IC)
disp('the current throught resistor RD in milli amps
    is: ')
ID=0
//switch D is open//
disp(ID)
Itot=IA+IB+IC+ID
//total current is denoted by Itot//
If=Itot
Eo=If*Rf
Eo=Eo*(-1)
//Eo denotes output voltage//
disp('Output voltage of the circuit in volts is: ')
disp(Eo)
//result is displayed//
```

Scilab code Exa 11.6 Resolution and Percent resolution 12bit DAconverter

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

```
//clears all existing variables//
E=50-(-50)
n=12
//given the voltage range E in volts and the number
   of bits n//
disp('12 bit converter can have (2^12)-1 non zero
   states ')
Res=E/((2^n)-1)
Pres=1/((2^n)-1)*(100)
disp('the resolution in volts is : ')
disp(Res)
disp('the percent resolution is : ')
disp(Pres)
//answers are displayed//
```

Scilab code Exa 11.7 Resolution and Percent resolution 10bit ADconverter

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

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