

Scilab Textbook Companion for  
Digital Principals And Applications  
by D. P. Leach And A. P. Malvino<sup>1</sup>

Created by  
Kapu Venkat Sayeesh  
B.Tech (pursuing)  
Electronics Engineering  
NIT, Warangal  
College Teacher  
S.K.L.V Sai Prakash  
Cross-Checked by  
Giridharan, IITB

July 30, 2019

<sup>1</sup>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 Principals And Applications

**Author:** D. P. Leach And A. P. Malvino

**Publisher:** Tata McGraw - Hill, New Delhi

**Edition:** 6

**Year:** 2006

**ISBN:** 0-07-060175-5

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

**Exa** Example (Solved example)

**Eqn** Equation (Particular equation of the above book)

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

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

# Contents

List of Scilab Codes	4
1 Digital Principles	6
2 Digital Logic	7
3 Combinational Logic Circuits	25
4 Data processing circuits	43
5 Number Systems and Codes	52
6 Arithmetic Circuits	66
7 Clocks and Timing Circuits	90
8 Flip Flops	105
9 Registers	114
10 Counters	122
11 Design of Sequential Circuit	138
12 D to A Conversion and A to D conversion	145

<b>13 Memory</b>	<b>151</b>
<b>14 Digital Integrated circuits</b>	<b>153</b>
<b>15 Applications</b>	<b>156</b>
<b>16 A Simple Computer Design</b>	<b>161</b>

# List of Scilab Codes

Exa 1.1	Finding duty cycle . . . . .	6
Exa 1.2	Maximum decimal count for a counter . . . . .	6
Exa 2.1	7404 waveform . . . . .	7
Exa 2.2	7404 waveform . . . . .	9
Exa 2.3	truth table for given figure . . . . .	12
Exa 2.4	truth table for given figure . . . . .	12
Exa 2.9	proving two circuits are logically equal . . . . .	13
Exa 2.10	truth table for NOR NOR circuit . . . . .	14
Exa 2.11	timing diagram for NOR NOR . . . . .	14
Exa 2.12	proving two circuits are logically equal . . . . .	18
Exa 2.13	truth table for NAND NAND circuit . . . . .	19
Exa 2.14	timing diagram for NAND NAND circuit . . . . .	19
Exa 2.15	detecting all bits low in a register . . . . .	23
Exa 3.1	Boolean Algebra . . . . .	25
Exa 3.2	Boolean Algebra . . . . .	25
Exa 3.3	Testing a circuit using logic clip . . . . .	26
Exa 3.4	Sum of Products . . . . .	27
Exa 3.5	Boolean Algebra . . . . .	28
Exa 3.6	Gives a simplified Boolean equation . . . . .	29
Exa 3.7	simplest logic for given Truth table . . . . .	29
Exa 3.8	simplest logic for given logic equation . . . . .	30
Exa 3.9	Product of sums . . . . .	30
Exa 3.10	sop for the karnaugh map . . . . .	32
Exa 3.11	POS form of karnaugh map . . . . .	32
Exa 3.12	POS form of karnaugh map . . . . .	33
Exa 3.13	Quine Mc clusky method . . . . .	33
Exa 3.14	Dynamic hard . . . . .	37
Exa 4.1	4 to 1 mux using 2 to 1 mux . . . . .	43

Exa 4.2	Realizing boolean equation using 8 to 1 mux	43
Exa 4.3	32 to 1 mux using 16 to 1 and 2 to 1 muxes	46
Exa 4.4	74154 IC y12 . . . . .	46
Exa 4.7	realizing boolean equation using 3 to 8 decoder	47
Exa 4.8	current in LED . . . . .	48
Exa 4.9	which LED lights up for given input conditions	48
Exa 4.10	output of 74147 when button 6 is pressed . .	49
Exa 4.11	priority encoder . . . . .	50
Exa 5.1	Binary to decimal conversion . . . . .	52
Exa 5.2	Binary to decimal conversion . . . . .	54
Exa 5.3	decimal equivalent of 2 Mb . . . . .	56
Exa 5.4	Decimal to binary conversion . . . . .	57
Exa 5.5	Binary number having all ones . . . . .	58
Exa 5.6	Decimal to binary conversion . . . . .	59
Exa 5.7	binary to hexadecimal . . . . .	60
Exa 5.8	hexadecimal to decimal . . . . .	61
Exa 5.9	decimal to hexadecimal and binary . . . . .	62
Exa 5.10	decimal to hexadecimal and binary . . . . .	63
Exa 5.11	decimal to hexadecimal and binary . . . . .	64
Exa 6.1	8bit binary adder . . . . .	66
Exa 6.2	16 bit binary adder . . . . .	68
Exa 6.3	first generation microcomputers addition . .	69
Exa 6.4	binary subtraction . . . . .	71
Exa 6.5	adding 8 bit unsigned numbers . . . . .	73
Exa 6.6	subtraction of unsigned numbers . . . . .	76
Exa 6.7	overflow case . . . . .	78
Exa 6.8	2s compliment . . . . .	80
Exa 6.9	2s compliment . . . . .	83
Exa 6.10	2s compliment subtraction . . . . .	85
Exa 6.12	final carry in a CLA . . . . .	88
Exa 7.1	clock cycle time . . . . .	90
Exa 7.2	maximum clock frequency . . . . .	90
Exa 7.3	frequency limits of the clock . . . . .	91
Exa 7.4	Schmitt trigger . . . . .	93
Exa 7.5	frequency of oscillation for 555 timer . . . .	94
Exa 7.6	finding Ra and C in 555 timer circuit . . . .	94
Exa 7.7	output pulse width for the timer . . . . .	95

Exa 7.8	value of C necessary to change pulse width to given values . . . . .	96
Exa 7.9	monostable multivibrator . . . . .	96
Exa 7.10	74123 . . . . .	100
Exa 7.11	finding timing capacitor values . . . . .	103
Exa 8.4	RS flipflop . . . . .	105
Exa 8.5	positive edge triggered RS flip flop . . . . .	106
Exa 8.6	negative edge triggered RS flip flop . . . . .	106
Exa 8.7	T flip flop . . . . .	107
Exa 8.9	JK master slave . . . . .	107
Exa 8.10	fictitious flip flop excitation table . . . . .	110
Exa 8.12	state transition diagram for given circuit . . . . .	112
Exa 8.13	D flip flop to RS flip flop . . . . .	113
Exa 9.1	shift register serial input . . . . .	114
Exa 9.2	shift register serial input and output graph . . . . .	115
Exa 9.4	54164 shift register . . . . .	120
Exa 9.5	54164 shift register . . . . .	120
Exa 9.8	74ls174 . . . . .	121
Exa 9.9	7495A . . . . .	121
Exa 10.1	ripple counter clock frequency . . . . .	122
Exa 10.2	number of flip flops required to construct a counter . . . . .	122
Exa 10.3	Output waveforms for a 7493A connected as a mod 16 counter . . . . .	124
Exa 10.5	Expression for AND gate connected to the leg of OR gate that drives clock input to flip flop Qd in 74193 . . . . .	126
Exa 10.6	Expression for 4 input AND gate connected to the leg of OR gate that conditions the J and K inputs to the Qd flip flop in a 74191 . . . . .	127
Exa 10.7	number of flip flops required to construct a counter . . . . .	127
Exa 10.8	what modulus counters can be constructed with given number of flip flops e . . . . .	128
Exa 10.9	mod 6 counter . . . . .	128
Exa 10.10	Expression for a gate to decode count 8 in a 7492A . . . . .	132
Exa 10.12	mod 12 counter . . . . .	132



Exa 10.13	4 bit binary counter presettable . . . . .	133
Exa 10.14	self correcting modulo 6 counter . . . . .	135
Exa 10.15	sequence generator . . . . .	137
Exa 11.1	synchronous sequential logic circuit . . . . .	138
Exa 11.2	vending machine . . . . .	140
Exa 11.5	Reducing state transition diagrams . . . . .	141
Exa 11.6	asynchronous sequential circuit . . . . .	142
Exa 11.7	asynchronous sequential circuit problem in op- eration . . . . .	143
Exa 11.8	asynchronous sequential circuit . . . . .	144
Exa 12.1	binary equivalent weight of each bit in a 4bit system . . . . .	145
Exa 12.2	5 bit resistive divider . . . . .	146
Exa 12.3	5 bit ladder . . . . .	146
Exa 12.4	5 bit ladder . . . . .	147
Exa 12.5	5 bit ladder . . . . .	147
Exa 12.6	5 bit ladder . . . . .	148
Exa 12.8	DAC0808 . . . . .	148
Exa 12.9	resolution of 9 bit D to A . . . . .	149
Exa 12.10	resolution . . . . .	149
Exa 12.11	counter type A to D converter . . . . .	149
Exa 12.13	10 bit A to D converter . . . . .	150
Exa 13.2	structure of binary address . . . . .	151
Exa 13.3	decimal and hexadecimal address for the given binary address . . . . .	151
Exa 14.1	diode forward or reverse . . . . .	153
Exa 14.2	Diode current . . . . .	154
Exa 14.3	current in the given circuit . . . . .	154
Exa 14.4	n channel MOSFET inverter . . . . .	155
Exa 15.1	Timing of a six digit display . . . . .	156
Exa 15.4	Basic frequency counter . . . . .	157
Exa 15.5	4 decimal digit frequency counter . . . . .	157
Exa 15.6	instrument to measure time period . . . . .	158
Exa 15.9	ADC0804 . . . . .	158
Exa 15.10	ADC3511 . . . . .	159
Exa 15.11	ADC3511 . . . . .	160
Exa 15.12	ADD3501 . . . . .	160
Exa 16.1	size of PC IR ACC MAR MDR . . . . .	161

Exa 16.6	Number of clock cycles needed to execute a program . . . . .	162
----------	--	-----

# List of Figures

2.1	7404 waveform . . . . .	8
2.2	7404 waveform . . . . .	10
2.3	timing diagram for NOR NOR . . . . .	15
2.4	timing diagram for NAND NAND circuit . . . . .	20
3.1	Dynamic hard . . . . .	38
7.1	Schmitt trigger . . . . .	92
7.2	monostable multivibrator . . . . .	97
7.3	monostable multivibrator . . . . .	98
7.4	74123 . . . . .	100
7.5	74123 . . . . .	101
8.1	JK master slave . . . . .	108
9.1	shift register serial input and output graph . . . . .	116
10.1	Output waveforms for a 7493A connected as a mod 16 counter	123
10.2	mod 6 counter . . . . .	129

# Chapter 1

## Digital Principles

Scilab code Exa 1.1 Finding duty cycle

```
1 //Example 1.1
2 clc;
3 clear;
4 f= 5 * 10^6 ; //given
5 T=1/f; // calculating the time period .
6 H = 0.05 * 10^-6 / T ;
7 printf('Time period of the waveform is T = %f us\n',
      T); //displaying results
8 printf(" Duty cycle H = %f %%",H*100);
```

---

Scilab code Exa 1.2 Maximum decimal count for a counter

```
1 //Example 1.2
2 clc;
3 clear;
4 n=8; // given no of flip flops
5 max_count = 2^n -1 ;
6 printf("Maximum count = %d",max_count);
```

---

# Chapter 2

## Digital Logic

Scilab code Exa 2.1 7404 waveform

```
1 //exmaple 2.1
2 //7404
3 clc
4 close
5 clear
6 //frq= input('Enter the square wave frequency in KHz
   :');
7 frq=1 ;// frequency in KHz
8 t=(1/frq)*100;
9 t=round(t)
10 for r=1:t*10
11     inputc(r)=0;
12     outputc(r)=0;
13 end
14 p=1;
15 while p<t*10 // making arrays
16     to plot the curve
17     if p==1 | modulo(p,t)==0 then
18         for k=1:t/2
```

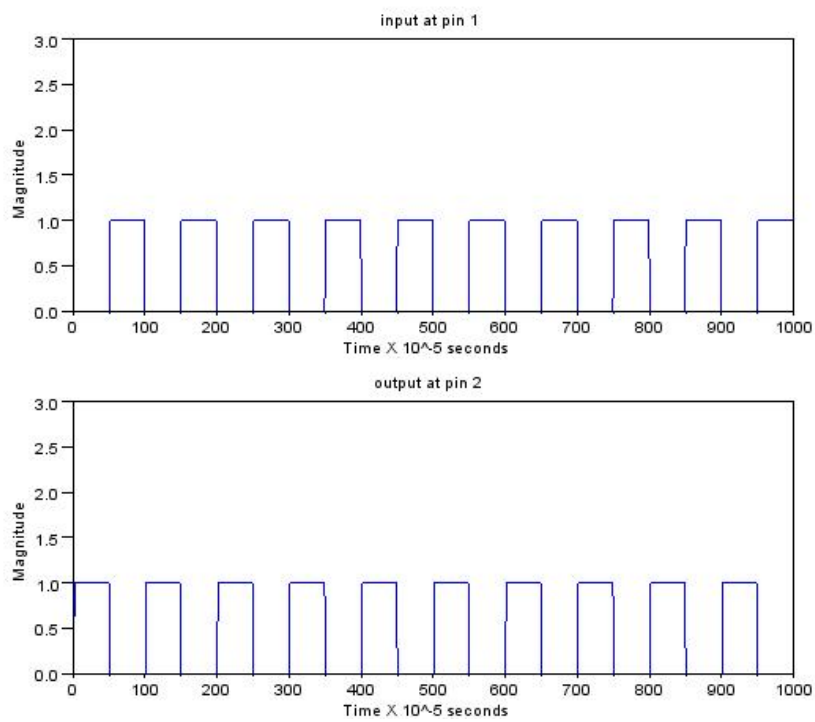


Figure 2.1: 7404 waveform

```

18         inputc(p+k)=0;
19         outputc(p+k)=1;
20     end
21     p=p+t/2;
22     else
23         inputc(p)=1;
24         outputc(p)=0;
25         p=p+1;
26     end
27 end
28 y=[3 3];
29 subplot(2,1,1) //ploting the curves
30 title('input at pin 1')
31 xlabel('Time X 10-5 seconds');
32 ylabel('Magnitude')
33 plot(inputc)
34 plot(y)
35 subplot(2,1,2)
36 title('output at pin 2')
37 xlabel('Time X 10-5 seconds');
38 ylabel('Magnitude')
39 plot(outputc)
40 plot(y)

```

---

### Scilab code Exa 2.2 7404 waveform

```

1 //exmaple 2.2
2 //7404
3 clc
4 close
5 clear
6 //frq= input('Enter the square wave frequency in KHz
:');

```

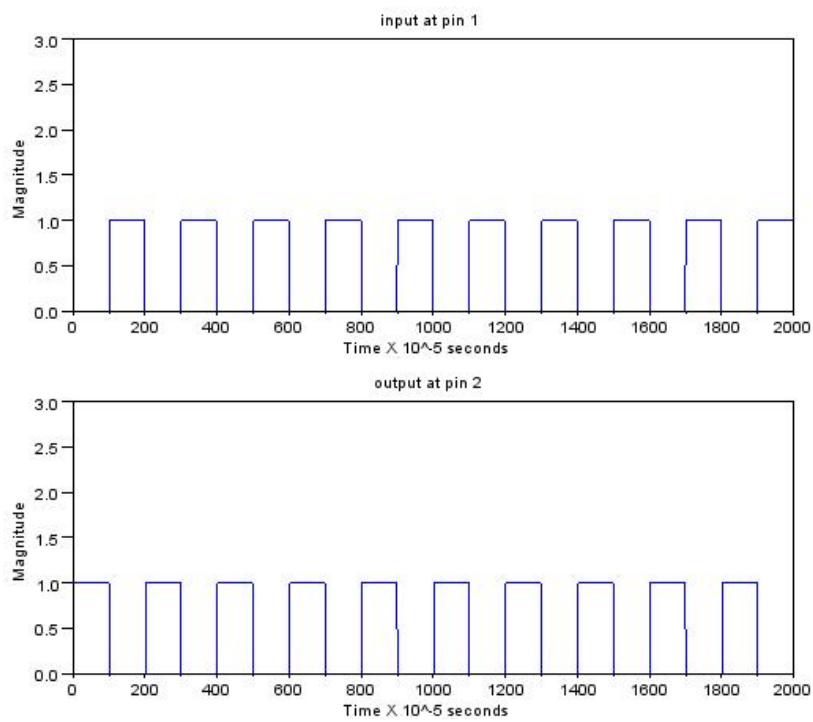


Figure 2.2: 7404 waveform



```

7 frq=0.5 ;//KHz
8 t=(1/frq)*100;
9 t=round(t)
10 for r=1:t*10
11     inputc(r)=0;
12     outputc(r)=0;
13 end
14 p=1;
15 while p<t*10 //
    making arrays t plot the curve
16     if p==1 | modulo(p,t)==0 then
17         for k=1:t/2
18             inputc(p+k)=0;
19             outputc(p+k)=1;
20         end
21         p=p+t/2;
22     else
23         inputc(p)=1;
24         outputc(p)=0;
25         p=p+1;
26     end
27 end
28 y=[3 3];
29 subplot(2,1,1) //ploting the graphs
30 title('input at pin 1')
31 xlabel('Time X 10-5 seconds');
32 ylabel('Magnitude')
33 plot(inputc)
34 plot(y)
35 subplot(2,1,2)
36 title('output at pin 2')
37 xlabel('Time X 10-5 seconds');
38 ylabel('Magnitude')
39 plot(outputc)
40 plot(y)

```

---

Scilab code Exa 2.3 truth table for given figure

```
1 // exemple 2.3
2 clc
3 clear
4 close
5 a=[0 0 1 1];
6 b=[0 1 0 1];
7 for i=1:4
8     r(i)= bitor(bitcmp(a(i),1), bitcmp(b(i),1))
           // given expression
9 end
10 disp('  A      B      Y')
11 for i = 1 : 4
12     Y(i,1)=a(i);
13     Y(i,2)=b(i);
14     Y(i,3)=r(i);
15     end
16 disp(Y); //displaying truth table
17 disp(''1'' represents a HIGH(H) and ''0''
       represents a LOW(L)')
```

---

Scilab code Exa 2.4 truth table for given figure

```
1 // exemple 2.4
2 clear
3 clc
4 a=[0 0 1 1];
5 b=[0 1 0 1];
6 for i=1:4
7     r(i)= bitand(bitcmp(a(i),1), bitcmp(b(i),1)) //
           given expression
```

```

8 end
9 disp('   A       B       Y')
10 for i = 1 : 4
11     Y(i,1)=a(i);
12     Y(i,2)=b(i);
13     Y(i,3)=r(i);
14     end
15 disp(Y); //displaying truth table
16 disp('''1'' represents a HIGH(H) and ''0''
        represents a LOW(L)')

```

---

Scilab code Exa 2.9 proving two circuits are logically equal

```

1 //Example 2.9
2 clc
3 clear
4 close
5 a=[0 0 0 0 0 0 0 0 1 1 1 1 1 1 1];
6 b=[0 0 0 0 1 1 1 1 0 0 0 0 1 1 1];
7 c=[0 0 1 1 0 0 1 1 0 0 1 1 0 0 1];
8 d=[0 1 0 1 0 1 0 1 0 1 0 1 0 1 0];
9 for i=1:16 // finding Y for all 16 cases
10     x=bitor(a(i),b(i));
11     y=bitor(c(i),d(i));
12     r(i)=bitand(x,y);
13     x1=bitcmp(x,1);
14     y1=bitcmp(y,1);
15     z=bitor(x1,y1);
16     r1(i)=bitcmp(z,1);
17 end
18 disp('   Y       Y1');
19 for i = 1 : 16
20     Y(i,1)=r1(i);
21     Y(i,2)=r(i);
22     end

```

```
23 disp(Y); //displaying result
24 disp('Both are logically equivalent');
```

---

Scilab code Exa 2.10 truth table for NOR NOR circuit

```
1 // exemple 2.10
2 clc
3 clear
4 a=[0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1];
5 b=[0 0 0 0 1 1 1 1 0 0 0 0 1 1 1 1];
6 c=[0 0 1 1 0 0 1 1 0 0 1 1 0 0 1 1];
7 d=[0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1];
8 for i=1:16
9     x=bitor(a(i),b(i));
10    y=bitor(c(i),d(i));
11    r(i)=bitand(x,y);
12 end
13 disp('Truth table :');
14 disp('  A      B      C      D      Y')
15 for i = 1 : 16 //displaying truth table
16     Y(i,1)=a(i);
17     Y(i,2)=b(i);
18     Y(i,3)=c(i);
19     Y(i,4)=d(i);
20     Y(i,5)=r(i);
21     end
22 disp(Y);
```

---

Scilab code Exa 2.11 timing diagram for NOR NOR

```
1 // exemple 2.11
```

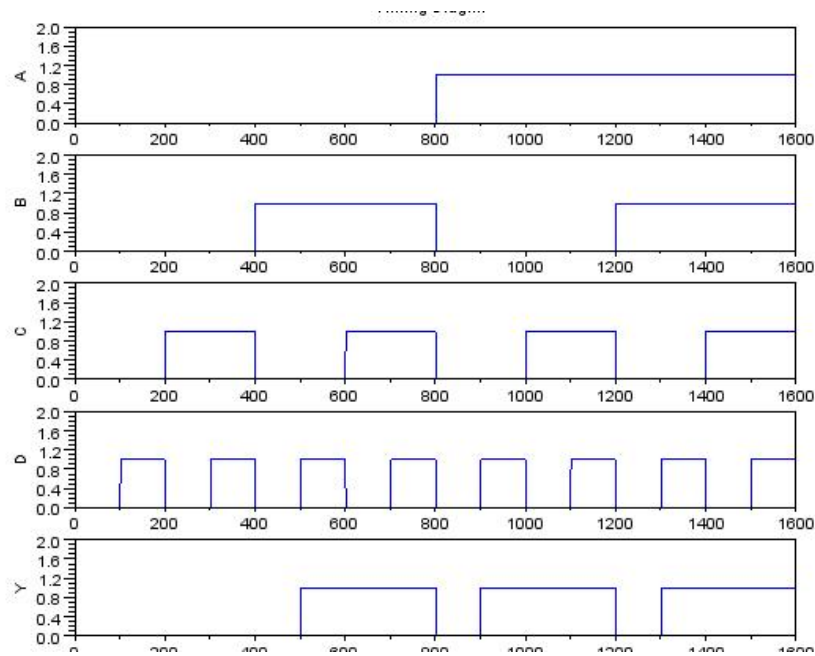


Figure 2.3: timing diagram for NOR NOR

```

2  clc
3  clear
4  close
5  a=[0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1];
6  b=[0 0 0 0 1 1 1 1 0 0 0 0 1 1 1 1];
7  c=[0 0 1 1 0 0 1 1 0 0 1 1 0 0 1 1];
8  d=[0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1];
9  for i=1:16
10     x=bitor(a(i),b(i));
11     y=bitor(c(i),d(i));
12     r(i)=bitand(x,y);
13 end
14 Y=r
15 ap=1;
16 bp=1;
17 cp=1;
18 dp=1; Yp=1;
19 for i=1:16      //Making array to plot the timing

```

```

    diagram
20     if a(i)==1 then
21         for o=1:100
22             a1(ap)=1;
23             ap=ap+1;
24         end
25     else
26         for o=1:100
27             a1(ap)=0;
28             ap=ap+1;
29         end
30 end
31 if b(i)==1 then
32     for o=1:100
33         b1(bp)=1;
34         bp=bp+1;
35     end
36 else
37     for o=1:100
38         b1(bp)=0;
39         bp=bp+1;
40     end
41
42 end
43 if c(i)==1 then
44     for o=1:100
45         c1(cp)=1;
46         cp=cp+1;
47     end
48 else
49     for o=1:100
50         c1(cp)=0;
51         cp=cp+1;
52     end
53 end
54 if d(i)==1 then
55     for o=1:100
56         d1(dp)=1;

```

```

57         dp=dp+1;
58         end
59     else
60         for o=1:100
61             d1(dp)=0;
62             dp=dp+1;
63             end
64         end
65 if Y(i)==1 then
66     for o=1:100
67         Y1(Yp)=1;
68         Yp=Yp+1;
69     end
70 else
71     for o=1:100
72         Y1(Yp)=0;
73         Yp=Yp+1;
74     end
75 end
76 end
77 z=[2 2];
78 subplot(5,1,1); //plotting timing diagram
79 title('Timing Diagram');
80 plot(z);
81 plot(a1);
82 ylabel('A');
83 subplot(5,1,2);
84 plot(z);
85 ylabel('B');
86 plot(b1);
87 subplot(5,1,3);
88 plot(z);
89 ylabel('C');
90 plot(c1);
91 subplot(5,1,4);
92 plot(z);
93 ylabel('D');
94 plot(d1);

```

```

95 subplot(5,1,5);
96 plot(z);
97 ylabel('Y');
98 xlabel('Time in milli seconds');
99 plot(Y1);

```

---

Scilab code Exa 2.12 proving two circuits are logically equal

```

1 //Example 2.12
2 clc
3 clear
4 a=[0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1];
5 b=[0 0 0 0 1 1 1 1 0 0 0 0 1 1 1 1];
6 c=[0 0 1 1 0 0 1 1 0 0 1 1 0 0 1 1];
7 d=[0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1];
8 for i=1:16 // finding Y and y1 for all possible inpt
           cases
9     x=bitand(a(i),b(i));
10    y=bitand(c(i),d(i));
11    r(i)=bitor(x,y);
12    x1=bitcmp(x,1);
13    y1=bitcmp(y,1);
14    z=bitand(x1,y1);
15    r1(i)=bitcmp(z,1);
16 end
17 disp('  Y    Y1');
18 for i = 1 : 16 // displaying result
19     Y(i,1)=r(i);
20     Y(i,2)=r1(i);
21     end
22 disp(Y);
23 disp('Both are logically equivalent');

```

---



Scilab code Exa 2.13 truth table for NAND NAND circuit

```
1 // exemple 2.13
2 //NAND - NAND
3 clc
4 clear
5 close
6 a=[0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1];
7 b=[0 0 0 0 1 1 1 1 0 0 0 0 1 1 1 1];
8 c=[0 0 1 1 0 0 1 1 0 0 1 1 0 0 1 1];
9 d=[0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1];
10 for i=1:16
11     x=bitand(a(i),b(i));
12     y=bitand(c(i),d(i));
13     r(i)=bitor(x,y);
14 end
15 disp('Truth table :');
16 disp('  A      B      C      D      Y')
17 for i = 1 : 16 // displaying the truth table
18     Y(i,1)=a(i);
19     Y(i,2)=b(i);
20     Y(i,3)=c(i);
21     Y(i,4)=d(i);
22     Y(i,5)=r(i);
23 end
24 disp(Y);
```

---

Scilab code Exa 2.14 timing diagram for NAND NAND circuit

```
1 // exemple 2.14
2 clc
3 clear
4 close
```

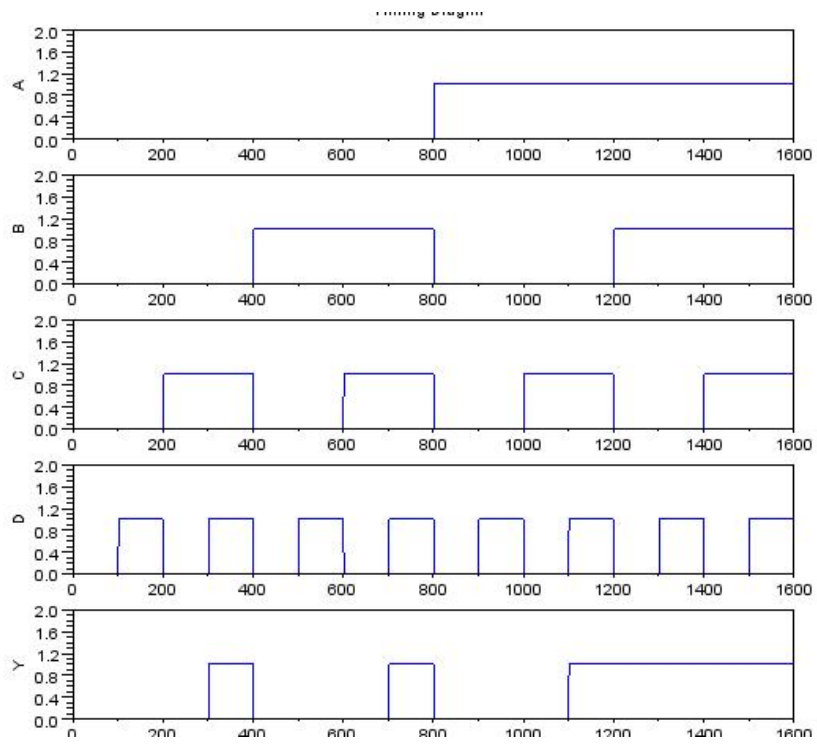


Figure 2.4: timing diagram for NAND NAND circuit

```

5 a=[0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1];
6 b=[0 0 0 0 1 1 1 1 0 0 0 0 1 1 1 1];
7 c=[0 0 1 1 0 0 1 1 0 0 1 1 0 0 1 1];
8 d=[0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1];
9 for i=1:16
10     x=bitand(a(i),b(i));
11     y=bitand(c(i),d(i));
12     r(i)=bitor(x,y);
13 end
14 Y=r
15 ap=1;
16 bp=1;
17 cp=1;
18 dp=1;Yp=1;
19 for i=1:16 //Making arrays to plot the timing
    diagram
20     if a(i)==1 then
21         for o=1:100
22             a1(ap)=1;
23             ap=ap+1;
24         end
25     else
26         for o=1:100
27             a1(ap)=0;
28             ap=ap+1;
29         end
30
31 end
32 if b(i)==1 then
33     for o=1:100
34         b1(bp)=1;
35         bp=bp+1;
36         //z(bp)=3
37     end
38 else
39     for o=1:100
40         b1(bp)=0;
41         bp=bp+1;

```

```

42         //z(bp)=3
43         end
44
45     end
46     if c(i)==1 then
47         for o=1:100
48             c1(cp)=1;
49             cp=cp+1;
50         end
51     else
52         for o=1:100
53             c1(cp)=0;
54             cp=cp+1;
55         end
56
57     end
58     if d(i)==1 then
59         for o=1:100
60             d1(dp)=1;
61             dp=dp+1;
62         end
63     else
64         for o=1:100
65             d1(dp)=0;
66             dp=dp+1;
67         end
68
69     end
70     if Y(i)==1 then
71         for o=1:100
72             Y1(Yp)=1;
73             Yp=Yp+1;
74         end
75     else
76         for o=1:100
77             Y1(Yp)=0;
78             Yp=Yp+1;
79         end

```

```

80
81     end
82
83 end
84 z=[2 2];
85 subplot(5,1,1); //plotting timing diagram
86 title('Timing Diagr');
87 plot(z);
88 plot(a1);
89 ylabel('A');
90 subplot(5,1,2);
91 plot(z);
92 ylabel('B');
93 plot(b1);
94 subplot(5,1,3);
95 plot(z);
96 ylabel('C');
97 plot(c1);
98 subplot(5,1,4);
99 plot(z);
100 ylabel('D');
101 plot(d1);
102 subplot(5,1,5);
103 plot(z);
104 ylabel('Y');
105 xlabel('Time in milli seconds');
106 plot(Y1);

```

---

**Scilab code Exa 2.15** detecting all bits low in a register

```

1 //example 2.15
2 clc
3 clear
4 s=0; // s from the register
5 s(1)=input('Enter the value at S0 :')

```

```
6 s(2)=input('Enter the value at S1 :')
7 s(3)=input('Enter the value at S2 :')
8 s(4)=input('Enter the value at S3 :')
9 s(5)=input('Enter the value at S4 :')
10 s(6)=input('Enter the value at S5 :')
11 s(7)=input('Enter the value at S6 :')
12 s(8)=input('Enter the value at S7 :')
13 count =0;
14 for i =1 :8 //loop to detect a '1'
15     if s(i)==1 then
16         disp('ZERO is LOW');
17         break;
18     end
19     count =count+1;
20 end
21 if count==8 then
22     disp('ZERO is HIGH');
23 end;
```

---

# Chapter 3

## Combinational Logic Circuits

Scilab code Exa 3.1 Boolean Algebra

```
1 //example 3.1
2 clc;
3 clear;
4 disp("we can minimize the given equation as:");
5 disp('Y = AB'' + AB = A(B'' + B)');
6 disp('Therefore, Y = A(1) = A');
7 disp('this says that output Y equals to A, so all we
      hve to do is connect a wire between input A and
      output Y.');
```

---

Scilab code Exa 3.2 Boolean Algebra

```
1 //example 3.2
2
3 clc;
4 clear;
5 disp("Multipl the factors of the foregoing equation
      to get");
```

```

6 disp('Y = A''A + A''B + BA +BB');
7 disp('it becomes,      Y = A''B + AB + B ');
8 disp('We can factor the foregooing equation as
      follows :');
9 disp('Y = B(A'' + A) + B = B + B = B ');
10 disp('this says that output Y equals to B, so all we
      have to do is connect a wire between input B and
      output Y. ');

```

---

### Scilab code Exa 3.3 Testing a circuit using logic clip

```

1 //example 3.3
2 clc;
3 clear;
4 //disp(' Enter the inputs for AND gate ');
5 //for i=1:7
6 //printf("Enter wether the led %d is on or off (1 or
      0) :",i);
7 //a(i)=input(' ');
8 //end;
9 //disp('Enter the inputs for OR gate' )
10 //for i=1:7
11 //printf("Enter wether the led %d is on or off (1 or
      0) :",i);
12 //b(i)=input(' ');
13 //end;
14 a= [0 1 0 1 1 0 0]; //lights on a logic clip
15 b= [0 0 0 0 1 1 0];
16 k=bitand(a(1),a(2));
17 k2 = bitand(a(4),a(5));
18 k3=bitor(b(1),b(2));
19 k4 = bitor(b(4),b(5));
20 if a(3) ~= k then // checking which gate is faulty
21     disp("The first AND gate is diffective ");
22 elseif a(6) ~= k2 then

```



```

23         disp("The second AND gate is diffeective ");
24     elseif b(3) ~= k3 then
25         disp("The first OR gate is diffeective ");
26 elseif b(6) ~= k4 then
27         disp("The second OR gate is diffeective ");
28     else
29         disp('All the gates are working correctly');
30 end

```

---

#### Scilab code Exa 3.4 Sum of Products

```

1 //example 3.4
2 clc;
3 clear;
4 disp('Given the truth table has high output for
    following conditons :');
5 a=[0 0 0 ; 0 1 0 ; 1 0 0 ; 1 1 0 ] //given iput
    conditions for which output is high
6 disp(a)
7 for (i=1:4)
8     if a(i,1)==1 then
9         b(i,1)= 'A'
10    else
11        b(i,1)= 'A^'
12    end
13    if a(i,2)==1 then
14        b(i,2)= 'B'
15    else
16        b(i,2)= 'B^'
17    end
18    if a(i,3)==1 then
19        b(i,3)= 'C'
20    else
21        b(i,3)= 'C^'
22    end

```

```

23 end
24 disp('When you OR these products you get :') //
    displaying sum of products
25 x=strcat([b(1,1) b(1,2) b(1,3) " + " b(2,1) b(2,2) b
    (2,3) " + " b(3,1) b(3,2) b(3,3) " + " b(4,1) b
    (4,2) b(4,3)]);
26 disp(x)

```

---

### Scilab code Exa 3.5 Boolean Algebra

```

1 //example 3.5
2 clc;
3 clear;
4 disp('The boolean equation is :');
5 disp('Y = A''B''C'' + A''BC'' + AB''C'' + ABC''');
6 disp('Since C'' is common to each term, factor as
    follows :');
7 disp('Y = (A''B + A''B + AB'' AB)C''');
8 disp('Again, factor to get :');
9 disp('Y = [A''(B'' + B) + A(B'' + B)]C''');
10 disp('Now, simplify the foregoing as follows :');
11 disp('Y=[A''(1) + A(1)]C'' = (A'' + A)C''');
12 disp(' or Y= C''');
13 disp('This final equation means that you don''t even
    need a logic circuit. All you need is a wire
    connecting input C'' to output Y.'');

```

---

check Appendix [AP 10](#) for dependency:

kmap.sci

check Appendix [AP 2](#) for dependency:

noof.sci

Scilab code Exa 3.6 Gives a simplified Boolean equation

```
1 //example 3.6
2 // this program needs kmap.sci and noof.sci
3 clc
4 Y=[7 9 10 11 12 13 14 15]; //given logic equation
5 k=[0 0 0 0;0 0 1 0 ; 1 1 1 1; 0 1 1 1]; //
    minimizing it using 4-variable kmap
6 disp("The minimal expression of Y from the
    following Kmap is :');
7 kmap(k); //calling the Kmap function
```

---

check Appendix [AP 10](#) for dependency:

kmap.sci

check Appendix [AP 2](#) for dependency:

noof.sci

Scilab code Exa 3.7 simplest logic for given Truth table

```
1 //example 3.7
2 // this program needs kmap.sci and noof.sci
3 clc;
4 disp('The kanaurgh map for given truth table will be
    :');
5 disp('      C'D''  C'D  CD  CD'''); //displaying
    the given kmap
6 disp('A'B''  1   0   0   0');
7 disp('A'B   0   0   0   0');
8 disp('AB   x   x   x   x');
9 disp('AB''  0   0   x   x');
10 disp('The truth table has output one only for the
    input condition 0000. The corresponding
    fundamental product is A'B'C'D''');
11 k=[1 0 0 0 ; 0 0 0 0; 0 0 0 0;0 0 0 0];
12 kmap(k); //calling the Kmap function
```

---

check Appendix [AP 10](#) for dependency:

kmap.sci

check Appendix [AP 2](#) for dependency:

noof.sci

**Scilab code Exa 3.8** simplest logic for given logic equation

```
1 //example 3.8
2 // this program needs kmap.sci and noof.sci
3 clc;
4 disp('          C''D''  C''D  CD  CD'''); //displaying
    the given kmap
5 disp('A''B''  0    0    0    0');
6 disp('A''B    0    0    1    0');
7 disp('AB      x    x    x    x');
8 disp('AB''  0    0    x    x');
9 k=[0 0 0 0;0 0 1 0;0 0 1 0;0 0 0 0];
10 disp('In a Karnaugh map if don''t care condition
    exits , we may consider them as ones if that gives
    a larger group size. ');
11 disp('The minimal expression from the given kmap is
    ');
12 kmap(k); //calling the kmap function
```

---

**Scilab code Exa 3.9** Product of sums

```
1 //example 3.4
2 clc;
3 clear;
```

```

4 disp('Given the truth table has high output for
      following conditons :');
5 a=[0 0 0 ; 0 0 1 ; 0 1 0 ] //given truth table
6 disp(a)
7
8 for (i=1:3) //finding the terms in pos
9     if a(i,1)==0 then
10        b(i,1)= 'A'
11     else
12        b(i,1)= 'A^'
13     end
14     if a(i,2)==0 then
15        b(i,2)= 'B'
16     else
17        b(i,2)= 'B^'
18     end
19     if a(i,3)==0 then
20        b(i,3)= 'C'
21     else
22        b(i,3)= 'C^'
23     end
24 end
25 disp(b)
26 disp('The product-of-sums equation is :') //
      displaying the POS
27 x=strcat(["(" " b(1,1) " + " b(1,2) " + " b(1,3) ")") " "
          (" b(2,1) " + " b(2,2) " + " b(2,3) ")") " (" b
          (3,1) " + " b(3,2) " + " b(3,3) ")") ]);
28 disp(x)

```

---

check Appendix [AP 10](#) for dependency:

kmap.sci

check Appendix [AP 2](#) for dependency:

noof.sci

### Scilab code Exa 3.10 sop for the karnaugh map

```
1 //example 3.10
2 // this program needs kmappos.sci and noof.sci
3
4 k=[0 0 0 0;0 0 0 1; 1 1 1 1;1 1 1 1];
5 disp("The minimal expression of Y from the following
      Kmap is :');
6 kmap(k);
7 disp('After complimenting and simplifying the
      Krarnugh map we get Y =:');
8 k=[1 1 1 1;1 1 1 0; 0 0 0 0;0 0 0 0 ];
9 kmap(k); //calling the Kmap function
```

---

check Appendix [AP 11](#) for dependency:

kmappos.sci

check Appendix [AP 2](#) for dependency:

noof.sci

### Scilab code Exa 3.11 POS form of karnaugh map

```
1 //example 3.11
2 // this program needs kmappos.sci and noof.sci
3
4 clc
5 disp('The given kmap is '); //displaying the given
      kmap
6 disp('      C''D''  C''D  CD  CD''');
7 disp('A''B''  0    0    0    0');
8 disp('A''B    0    0    0    1');
9 disp('AB      1    1    1    1');
10 disp('AB''    1    1    1    1');
11 disp("The simplest POS form of following Kmap is :")
      ;
```

```

12
13 k= [0 0 0 0;0 0 0 1 ;1 1 1 1; 1 1 1 1];
14 kmappos(k); // calling the Kmappos function

```

---

check Appendix [AP 11](#) for dependency:

kmappos.sci

check Appendix [AP 2](#) for dependency:

noof.sci

### Scilab code Exa 3.12 POS form of karnaugh map

```

1 //example 3.12
2 clc
3 disp('The given kmap is '); //displaying the given
   kmap
4 disp('      C'D'' C'D CD CD''');
5 disp('A'B''  0   0   1   0');
6 disp('A'B   0   0   1   1');
7 disp('AB     x   x   x   1');
8 disp('AB''   x   x   x   0');
9 disp('In a Karnaugh map if don''t care condition
   exits , we may consider them as zeros if that
   gives a larger group size. ');
10 disp('The simplest POS form of following Kmap is :')
   ;
11
12 k= [0 0 1 0;0 0 1 1 ;0 0 1 1; 0 0 1 0];
13 kmappos(k); //calling the Kmappos function

```

---

### Scilab code Exa 3.13 Quine Mc clusky method

```

1 //example 3.1
2 clc;
3 clear;
4 a=[0 0 0 0 1 1 1 1;0 0 1 1 0 0 1 1;0 1 0 1 0 1 0 1];
    //from the truth table given
5 y=[0 0 1 0 0 0 1 1];
6 j=1;
7 for i=1:8 // finding for which input conditions the
    output is high
8     if y(i) == 1 then
9         x(j,:) = [a(1,i) a(2,i) a(3,i)];
10        j=j+1;
11    end
12 end
13 for i=1:j-1; // finding the first stage
14     f(i)=0;
15     c=0;
16     for m=3:-1:1
17         f(i) = f(i) + x(i,m)*(2^c);
18         c=c+1;
19     end
20 end
21 disp('stage 1'); //displaying first stage
22 x(:,4)=f;
23 disp('    A    B    C');
24 disp(x)
25 count=zeros(j-2,j-2)
26 pos=count;
27 for i=1:j-2 // for second stage comparing with each
    other
28     for k=1:j-i-1
29         for m=1:3
30             if x(i,m)==x(i+k,m) then
31                 count(i,k)=count(i,k)+1;
32             else
33                 pos(i,k)=m
34             end
35         end

```



```

36 end
37 end
38 r=1;
39 for i=1:j-2 //making a list of second stage
    elements
40     for m=1:j-2
41         if count(i,m)==2 then
42             posi(r) = pos(i,m);
43             sest(r,1) = x(i,4);
44             sest(r,2) = x(i+m,4);
45             r=r+1;
46         end
47     end
48 end
49 disp('stage 2'); //displaying second stage
50 disp(sest);
51 o=size(sest);
52 fin(1)=sest(1,1);
53 fin(2)=sest(1,2);
54 p=3;
55 for i=2:o(1,1) //removing redundancy in second stage
56     t=0;
57     ts=0;
58     for w=1:p-1
59         if fin(w)== sest(i,1) then
60             t=30;
61         end;
62         if fin(w) == sest(i,2) then
63             ts=40;
64         end
65     end
66     if t==0 then
67         fin(p)=sest(i,1);
68         finn(p-2)=i;
69         p=p+1;
70     end
71     if ts==0 then
72         fin(p)=sest(i,2)

```

```

73     finn(p-2)=i;
74     p=p+1;
75     end
76 end
77 ppp=size(finn) //selecting the prime implicants
78 l=1
79 fina(l)= finn(l);
80 for i=2:ppp(1,1)
81     q=0;
82     for b=1:l
83         if fina(b) == finn(i) then
84             q=89 ;
85         end
86     end
87     if q==0 then
88         fina(l+1)=finn(i);
89         l=l+1;
90         q=0;
91     end
92 end
93 kkk=size(fina);
94 i=1;
95 jj=0;
96 bi(1)= ' ';
97 x(i)
98 po=1;
99 for k=1:kkk(1,1)+1
100 for p=1:3 //appending a string to make the
    expression
101     if p ~= posi(i) then
102         if p == 1 & x(i,p)==1 then
103             bi(po)=strcat([bi(po) 'A']);
104         elseif p== 1 &x(i,p)==0 then
105             bi(po)=strcat([bi(po) 'A''']);
106         end
107         if p == 2 & x(i,p)==1 then
108             bi(po)=strcat([bi(po) 'B']);
109         elseif p== 2 &x(i,p)==0 then

```

```

110     bi(po)=strcat([bi(po) 'B''']);
111     end
112     if p == 3 & x(i,p)==1    then
113     bi(po)=strcat([bi(po) 'C']);
114     elseif p== 3 &x(i,p)==0 then
115     bi(po)=strcat([bi(po) 'C''']);
116     end
117     end
118 end
119 jj=jj+1;
120 if jj<=kkk(1,1) then
121 i=fina(jj);
122 bi(po)=strcat([bi(po) ' + ']);
123 end
124 end;
125 disp('The minimised expression is ');
126 disp(bi);

```

---

### Scilab code Exa 3.14 Dynamic hard

```

1 //example 3.9
2 clc;
3 clear;
4 close;
5 c = [1 1 0 0 0 0 0]; //given values
6 a= [1 1 1 1 1 1 1];
7 b= [1 1 1 1 1 1 1] ;
8 for i=1:7
9     y1(i)=0
10    y2(i)=1
11    y3(i)=0
12    y4(i)=1
13    y(i)=0

```

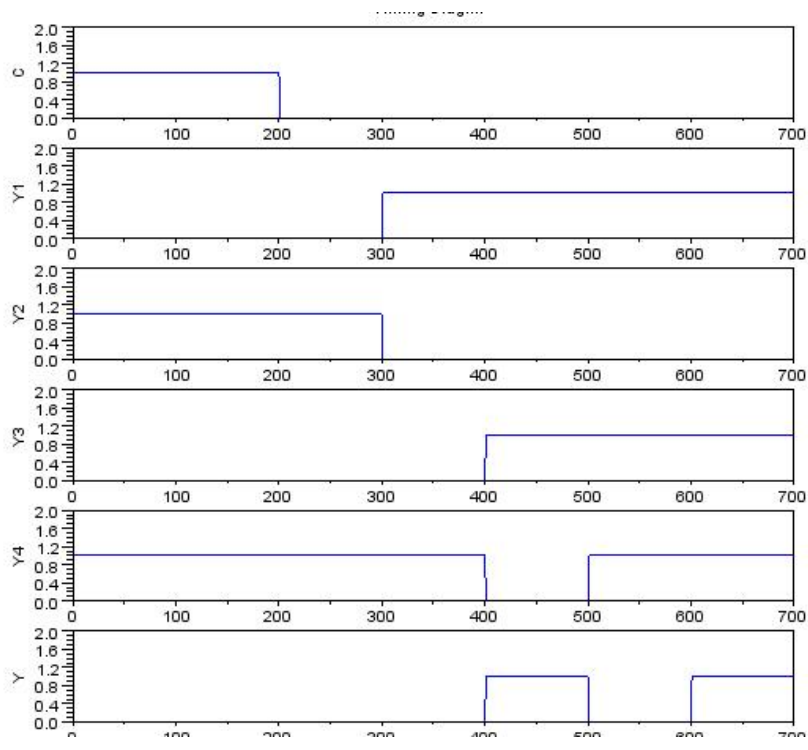


Figure 3.1: Dynamic hard

```

14 end
15 for(i=1: 7) // finding the Y values for next clock
    periods
16 y1(i+1) = bitcmp(c(i),1);
17 y2(i+1) = bitand(a(i),c(i));
18 end;
19 for i=1: 5
20 y3(i+2) = bitand(y1(i+1),b(i));
21 end;
22 for i=1:4
23 y4(i+3) = bitor(y3(i+2),y2(i+2));
24 end;
25 for i=1:3
26 y(i+4)   =   bitand(y4(i+3),y1(i+3));
27 end;
28 y11p=1;
29 y22p=1;
30 y33p=1;
31 y44p=1;
32 cp=1;
33 yf1p=1;
34 for i=1:7 // plotting all of them in to graph
35     if y1(i)==1 then
36         for o=1:100
37             y11(y11p)=1;
38             y11p=y11p+1;
39         end
40     else
41         for o=1:100
42             y11(y11p)=0;
43             y11p=y11p+1;
44         end
45     end
46     if y2(i)==1 then
47         for o=1:100
48             y21(y22p)=1;
49             y22p=y22p+1;
50             //z(bp)=3

```

```

51         end
52     else
53         for o=1:100
54             y21(y22p)=0;
55             y22p=y22p+1;
56             //z(bp)=3
57         end
58     end
59     if y3(i)==1 then
60         for o=1:100
61             y31(y33p)=1;
62             y33p=y33p+1;
63             //z(bp)=3
64         end
65     else
66         for o=1:100
67             y31(y33p)=0;
68             y33p=y33p+1;
69             //z(bp)=3
70         end
71     end
72     if y4(i)==1 then
73         for o=1:100
74             y41(y44p)=1;
75             y44p=y44p+1;
76             //z(bp)=3
77         end
78     end
79     else
80         for o=1:100
81             y41(y44p)=0;
82             y44p=y44p+1;
83             //z(bp)=3
84         end
85     end
86     if c(i)==1 then
87         for o=1:100
88             c1(cp)=1;

```

```

89         cp=cp+1;
90         end
91     else
92         for o=1:100
93             c1(cp)=0;
94             cp=cp+1;
95             end
96     end
97     if y(i)==1 then
98         for o=1:100
99             yf1(yf1p)=1;
100            yf1p=yf1p+1;
101            end
102        else
103            for o=1:100
104                yf1(yf1p)=0;
105                yf1p=yf1p+1;
106                end
107            end
108    end
109    z=[2 2]; //ploting the results
110    subplot(6,1,1);
111    title('Timing Diagram');
112    plot(c1);
113    plot(z);
114    ylabel('C');
115    subplot(6,1,2);
116    plot(y11);
117    ylabel('Y1');
118    plot(z);
119    subplot(6,1,3);
120    plot(y21);
121    ylabel('Y2');
122    plot(z);
123    subplot(6,1,4);
124    plot(z);
125    ylabel('Y3');
126    plot(y31);

```

```
127 subplot(6,1,5);
128 plot(z);
129 ylabel('Y4');
130 xlabel('Time in milli seconds');
131 plot(y41);
132 subplot(6,1,6);
133 plot(z);
134 ylabel('Y');
135 xlabel('Time in milli seconds');
136 plot(yf1);
```

---



# Chapter 4

## Data processing circuits

Scilab code Exa 4.1 4 to 1 mux using 2 to 1 mux

```
1 //example 4.1
2 clc;
3 clear;
4 disp('Logic equation for 2-to-1 Multiplexer :')
5 printf(' Y = A''D0 + AD1\n');
6 disp('Logic equation for 4-to-1 Multiplexer :')
7 printf(' Y = A''B''D0 + A''BD1 + AB''D2 + ABD3\n');
8 disp('This can be rewritten as,')
9 printf(' Y= A''(B''D0 + BD1) + A(B''D2 + BD3)\n');
10 disp('Compare this with equation of 2-to-1 mux. We
    need two 2-to-1 multiplexer to realize the
    bracketed terms where B serves as select input.
    The output of these two multiplexers can be sent
    to a third multiplexer as data inputs where A
    serves as select input and we get the 4-to-1
    multiplexer.');
```

---

Scilab code Exa 4.2 Realizing boolean equation using 8 to 1 mux

```

1 //example 4.2
2 clc;
3 clear
4 a(1,1)=0 // taking input in this form 1 if A, 0 if A
   ' and 2 if no A in the term
5 a(1,2)=1
6 a(1,3)=2
7 a(2,1)=2
8 a(2,2)=0
9 a(2,3)=0
10 a(3,1)=1
11 a(3,2)=1
12 a(3,3)=1
13 p=3;
14 for i=1:3 // finding the minterms here
15     coun =0;
16     for j=1:3
17         if a(i,j)==2 then
18             coun = coun+1
19         end
20     end
21     if coun == 2 then
22         p=p+3
23     else if coun==1 then
24         p=p+1
25     end
26 end
27 end
28 n=4;
29 for m=4:p
30     for l=1:3
31         a(m,l)=0;
32     end
33 end
34 for i= 1:p
35     for j=1:3
36         if a(i,j) ==2 then
37             for k=1:3

```

```

38             a(n,k)=a(i,k)
39             end
40             a(i,j) = 0;
41             a(n,j)=1;
42             n=n+1;
43         end
44     end
45 end
46 for h=1:p
47     f(h)=0
48     c=2;
49     for m=1:3 //finding equivlent decimal values for
the minterms
50         f(h)= f(h) + a(h,m)*(2^c);
51         c=c-1;
52     end
53 end
54 disp('The min terms are :') //displaying the min
terms
55 disp('   A       B       C')
56 disp(a)
57 l=1
58 o(1,l)= f(l); // removing the repetations in
minterms
59 for i=2:p
60     q=0;
61     for b=1:l
62         if o(1,b) == f(i) then
63             q=89 ;
64         end
65     end
66     if q==0 then
67         o(1,l+1)=f(i);
68         l=l+1;
69         q=0;
70     end
71 end
72 disp('The following data lines are to be given ''1''')

```

```

    and remaining should be given '0'); //
    displaying the decimal equivalent of minterms
73 disp(o);
74 disp('For a 4-1 mux, we should give D0 =C'',D1 = ''1
    '',D2 = C'' and D3 = C with A and B as data
    selector inputs ');

```

---

**Scilab code Exa 4.3** 32 to 1 mux using 16 to 1 and 2 to 1 muxes

```

1 //example 4.3
2 clc;
3 clear;
4 m(1)=32;//taking the given values
5 m(2)=log2(m(1)) // making necessary calculations
6 m(3)=m(2)-1;
7 m(4)=m(1)/2;
8 printf('A %d-to-1 multiplexer requires ',m(1));printf
    (' %d select lines , The lower ',m(2));printf(' %d
    select lines choose ',m(3));printf(' %d-to-1
    multiplexer outputs. The 2-to-1 multiplexers
    chooses one of the output of two ',m(4));printf('
    %d-to-1 multiplexers depending on what appears in
    the ',m(4));printf(' %dth select line. ', m(2));
    //displaying the result

```

---

**Scilab code Exa 4.4** 74154 IC y12

```

1 //example 4.4
2 clc;
3 clear;
4 r= input('Enter the value of R (0 or 1) :'); //
    accepting the inputs from the user
5 t= input('Enter the value of T (0 or 1) :');

```

```

6 sel = input('Enter the values of ABCD :');
7 strb = bitcmp(bitand(r,t),1);
8 if strb==0 then // checking whether strobe is high
    or low
9     if sel ==1100 then
10         y='The two pulses are steered to the Y12
            output';
11     else
12         y='The output Y12 remains in the High state'
            ;
13     end
14     else
15         y='The output Y12 remains in the High state';
16 end
17 disp(y) //displaying result

```

---

Scilab code Exa 4.7 realizing boolean equation using 3 to 8 decoder

```

1 //example 4.7
2 clc;
3 clear;
4 n=input('Enter the no.of terms in ur expression :');
    //accepting input from user
5 for i=1:n
6 a(1,i)=input('Enter the term (0-9) :');
7 end;
8 disp ('Since at the decoder output we get all the
        minterms we use them to get the required boolean
        functions by giving the output lines numbered ')
    ;
9 disp(a); //displying the result
10 disp('to a mlti-input OR gate.');
```

---

#### Scilab code Exa 4.8 current in LED

```
1 //example 4.8
2 clc;
3 clear;
4 //r=input('Enter the values of resistance in Kohms
   :');
5 //v=input('Enter the forward voltage drop of LED(in
   volts) :' );
6 r=1//taking the given values for r and v
7 v=2
8 i=5-v/r; //calculating I
9 printf('The current through a LED is : %f mA', i);
   //displaying I
```

---

#### Scilab code Exa 4.9 which LED lights up for given input conditions

```
1 //example 4.9
2 clc;
3 clear;
4
5 sel = input(' Enter the values of ABCD :');
6 a=sel;
7 q=1;
8 while(a>0) // finding the decimal equivalent
9     r=modulo(a,10);
10    b(1,q)=r;
11    a=a/10;
12    a=floor(a);
13    q=q+1;
14 end
15 f=0;
16 for m=1:q-1
17     c=m-1
18     f = f + b(1,m)*(2^c);
```

```

19 end
20 if f >9 then //checking the invalid condition
21     disp('Its a invalid input. Therefore ,none of the
           LEDs is on because all outputs lines are
           high' );
22 else //displaying the LED no if the inputs are
       valid
23     printf('\n LED %d lights up all. All other LEDs
           remain off because the other outputs are high
           . ',f);
24 end

```

---

Scilab code Exa 4.10 output of 74147 when button 6 is pressed

```

1 //example 4.10
2 clc;
3 clear;
4 //sel= input('Enter which is pressed(1 - 9) : ');
5 sel=6;
6 aa=sel;
7 for i=4:-1:1 //converting the sel input to binary
       notation
8     a(1,i)=modulo(aa,2);
9     b(1,i)=bitcmp(a(1,i),1);
10    aa=aa/2;
11    aa=floor(aa);
12 end
13 printf('When switch %d is pressed the ABCD output is
       : ',sel);
14 disp(b); //displaying the result
15 printf(' Which is equivalent to %d when the output
       is complimented.In the output above a ''0''
       represents a LOW and a ''1'' represents a HIGH. ',
       ,sel );

```

---

check Appendix [AP 9](#) for dependency:

kmapsx.sci

check Appendix [AP 2](#) for dependency:

noof.sci

### Scilab code Exa 4.11 priority encoder

```
1 //example 4.11
2 //uses functions kmap.sci and noof.sci so run them
  before running this program ..
3 clc;
4 s=[0 1 1 1 1 ];
5 x1=[0 1 0 0 0 ];
6 x2=[0 0 1 0 0 ];
7 x3=[0 0 0 1 0 ];
8 for i=1:5
9     if s(i)==1 then // finding output A and B
10        if x1(i) == 1 then
11            a(i)=0;
12            b(i)=1;
13        elseif x2(i) == 1 then
14            a(i)=1;
15            b(i)=0;
16        elseif x3(i) == 1 then
17            a(i)=1;
18            b(i)=1;
19        else
20            a(i)=0;
21            b(i)=0;
22        end
23 else
24            a(i)=0;
25            b(i)=0;
```



```

26 end
27 end
28 for i = 1 : 5 //printin the state table
29     Y(i,1)=s(i);
30     Y(i,2)=x1(i);
31     Y(i,3)=x2(i);
32     Y(i,4)=x3(i);
33     Y(i,5)=a(i);
34     Y(i,6)=b(i);
35 end
36 disp('          Input          Output ');
37 disp('  S      X1      X2      X3      A      B ');
38     disp(Y);
39 kmp =[0 0 0 0;0 0 0 1 ;0 0 0 1;0 0 0 1]; //finding
    minimized expressin using 4-variable kmap
40     disp("The minimal expression of A from the
        following Kmap is :');
41     kmapsx(kmp);
42     kmp=[0 0 1 0;0 0 1 1 ;0 0 1 0;0 0 1 0]; //
        finding minimized expressin using 4-variable
        kmap
43     printf('\n');
44     disp('The minimal expression of B from the
        following Kmap is :')
45     kmapsx(kmp);

```

---

# Chapter 5

## Number Systems and Codes

Scilab code Exa 5.1 Binary to decimal conversion

```
1 //Example 5.1
2 clc
3
4 //clears the command window
5 clear
6
7 //clears all the variables
8 p=1;
9
10 //initialising variables
11 q=1;
12 z=0;
13 b=0;
14 w=0;
15 f=0;
16 format('v',18);
17
18 //increasing the precision to 18 .
19 //bin= input ("Enter the binary no to be converted
20 to its decimal equivalent : ") // accepting
21 the binary input from user
```

```

12 bin=110.001;
13 d= modulo(bin,1);

    //separating the decimal part and the integer
    part
14 d=d*10^10;
15 a=floor(bin);

    //removing the decimal part
16 while(a>0)

    //Loop to take the binary bits of integer in to a
    matrix
17     r=modulo(a,10);
18     b(1,q)=r;
19     a=a/10;
20     a=floor(a);
21     q=q+1;
22 end
23 for m=1:q-1

    //multiplying the bits of integer with their
    position values and adding
24     c=m-1;
25     f = f + b(1,m)*(2^c);
26 end
27 while(d>0)

    //Loop to take the binary bits of decimal in to a
    matrix
28     e=modulo(d,2)
29     w(1,p)=e
30     d=d/10;
31     d=floor(d)
32     p=p+1;
33     end
34 for n=1:p-1

```

```

        //multiplying the bits of decimal with their
        position values and adding
35     z=z+w(1,n)*(0.5)^(11-n);
36 end
37 z=z*10000;

        //rounding of to 4 decimal values
38 z=round(z);
39 z=z/10000;
40 printf("The Decimal equivalent of the Binary number
        given is = %f",f+z);        //Displaying the final
        result

```

---

#### Scilab code Exa 5.2 Binary to decimal conversion

```

1 //Example 5.2
2 clc

        //clears the command window
3 clear

        //clears all the variables
4 p=1;

        //initialising variables
5 q=1;
6 z=0;
7 b=0;
8 w=0;
9 f=0;
10 format('v',18);

        //increasing the precision to 18 .
11 //bin= input ("Enter the binary no to be converted
        to its decimal equivalent : ") // accepting

```

```

        the binary input from user
12 bin=1011.11;
13 d= modulo(bin,1);

        //separating the decimal part and the integer
        part
14 d=d*10^10;
15 a=floor(bin);

        //removing the decimal part
16 while(a>0)

        //Loop to take the binary bits of integer in to a
        matrix
17     r=modulo(a,10);
18     b(1,q)=r;
19     a=a/10;
20     a=floor(a);
21     q=q+1;
22 end
23 for m=1:q-1

        //multiplying the bits of integer with their
        position values and adding
24     c=m-1;
25     f = f + b(1,m)*(2^c);
26 end
27 while(d>0)

        //Loop to take the binary bits of decimal in to a
        matrix
28     e=modulo(d,2)
29     w(1,p)=e
30     d=d/10;
31     d=floor(d)
32     p=p+1;
33     end
34 for n=1:p-1

```

```

    //multiplying the bits of decimal with their
    position values and adding
35     z=z+w(1,n)*(0.5)^(11-n);
36 end
37 z=z*10000;

    //rounding of to 4 decimal values
38 z=round(z);
39 z=z/10000;
40 printf("The Decimal equivalent of the Binary number
    given is = %f",f+z);

    //Displaying the final result

```

---

### Scilab code Exa 5.3 decimal equivalent of 2 Mb

```

1 //Example 5.3
2 clc

    //clears the command window .
3 clear

    //clears all the variables .
4 format('v',18);

    //increasing the precision to 18 .
5 n=2; /// given 2 mb
6 dec = n * 2^20 ;
7 printf("The decimal equivalent of 2Mb is = %f ",dec)
    ;

    //displaying the value.

```

---

### Scilab code Exa 5.4 Decimal to binary conversion

```
1 //Example 5.3
2 clc

    //clears the command window
3 clear

    //clears all the variables
4 q=0;
5 b=0;
6 s=0;
7 format('v',18);

    //increasing the precision to 18 .
8 //a=input("Enter the decimal no to be converted to
    its binary equivalent : ");           // accepting
    the decimal input from user
9 a=23.6;
10 d=modulo(a,1);

    //separating the decimal part and the integer
    part
11 a=floor(a);

    //removing the decimal part
12
13 while(a>0)

    //taking integer part in to a matrix and convert
    to equivalent binary
14     x=modulo(a,2);
15     b= b + (10^q)*x;
16     a=a/2;
```

```

17     a=floor(a);
18     q=q+1;
19 end
20
21
22
23 for i=1:10

        // For values after decimal point converting to
        binary
24     d=d*2;
25     q=floor(d);
26     s=s+q/(10^i);
27     if d>=1 then
28         d=d-1;
29     end
30 end
31 k=b+s;
32 printf("The binary equivalent of the given decimal
        number is = %f",k);

        // displaying the final result.

```

---

### Scilab code Exa 5.5 Binary number having all ones

```

1 //Example 5.5
2 clc

        //clears the command window
3 clear

        //clears all the variables
4 format('v',18)

        //increasing the precision

```



```

5 n=32; // given 32 1's
6 dec=2^n - 1 ;
7 printf("The decimal equivalent of 32 bit number with
    all 1s is = %f ",dec); // displaying the
    result

```

---

### Scilab code Exa 5.6 Decimal to binary conversion

```

1 //Example 5.6
2 clc

    //clears the command window
3 clear

    //clears all the variables
4 q=0;
5 b=0;
6 s=0;
7 format('v',18);

    //increasing the precision to 18 .
8 //a=input("Enter the decimal no to be converted to
    its binary equivalent : "); // accepting
    the decimal input from user
9 a=363;

    //taking the value given in problem
10 d=modulo(a,1);

    //separating the decimal part and the integer
    part
11 a=floor(a);

    //removing the decimal part
12

```

```

13 while(a>0)

    //taking integer part in to a matrix and convert
    to equivalent binary
14     x=modulo(a,2);
15     b= b + (10^q)*x;
16     a=a/2;
17     a=floor(a);
18     q=q+1;
19 end
20
21
22
23 for i=1:10

    // For values after decimal point converting to
    binary
24     d=d*2;
25     q=floor(d);
26     s=s+q/(10^i);
27     if d>=1 then
28         d=d-1;
29     end
30 end
31 k=b+s;
32 disp("The give decimal number is 363")
33 printf(" The binary equivalent of the given decimal
    number is = %f",k);

    // displaying the final result.

```

---

#### Scilab code Exa 5.7 binary to hexadecimal

```

1 //Example 5.7
2 clc

```

```

    //clears the command window
3  clear

    //clears all the variables
4  q=1;
5  b=0;
6  f=0;
7  bin=input("Enter the 8-bit binary address :");
    // Taking the input
    binary bits from the user
8  a=floor(bin)
9  while(a>0)

    //Loop to take the binary bits in to a matrix(
    array)
10     r=modulo(a,10);
11     b(1,q)=r;
12     a=a/10;
13     a=floor(a);
14     q=q+1;
15 end
16 for m=1:q-1

    // converrrting to decimal
17     c=m-1;
18     f = f + b(1,m)*(2^c);
19 end
20 c=dec2hex(f);
21 printf("The hexadecimal equivalent of the given
    binary number is : %s",c);
    //
    displaying the value

```

---

Scilab code Exa 5.8 hexadecimal to decimal

```

1 //Example 5.8
2 clc

   //clear the command window
3 clear

   //clear the variables
4 a=input("Enter the hexadecimal number to be
   converted into decimal(enter in a single
   quotation) : ") // taking the input from
   user
5 d=hex2dec(a);
6 printf("The decimal equivalent is : %d",d);

   //displaying the output

```

---

Scilab code Exa 5.9 decimal to hexadecimal and binary

```

1 //Example 5.9
2 clc //clears the
   command window
3 clear //clears
   the variables
4 q=0;
5 b=0;
6 //a=input("enter the decimal no:")
7 a=65535; //
   giving the value specified in the problem
8 temp =a;
9 format('v',18) //
   increasing the precision to 18
10 a=floor(a);
11 h=dec2hex(a);
12 while(a>0) //
   converting to binary

```

```

13     x=modulo(a,2);
14     b= b + (10^q)*x;
15     a=a/2;
16     a=floor(a);
17     q=q+1;
18 end
19 printf("Given decimal number is : %d\n",temp)
20 printf('The hexadecimal equivalent is = %s\n',h);
//
    displaying the results
21 printf('The binary equivalent is = %f',b );

```

---

#### Scilab code Exa 5.10 decimal to hexadecimal and binary

```

1 //Example 5.10
2 clc //clears the
    command window
3 clear //clears
    the variables
4 q=0;
5 b=0;
6 //a=input("enter the decimal no:")
7 a=56000; //
    giving the value specified in the problem
8 temp=a;
9 format('v',18) //
    increasing the precision to 18
10 a=floor(a);
11 h=dec2hex(a);
12 while(a>0) //
    converting to binary
13     x=modulo(a,2);
14     b= b + (10^q)*x;
15     a=a/2;
16     a=floor(a);

```

```

17     q=q+1;
18 end
19 printf("Given decimal number is : %d\n",temp)
20 printf("The hexadecimal equivalent is = %s\n",h)
21 //displaying the results
22 printf("The binary equivalent is = %f\n",b);

```

---

### Scilab code Exa 5.11 decimal to hexadecimal and binary

```

1 //chapter 5
2 //Example 5.11
3 //Q.convert decimal numbers to its hexadecimal and
  binary equivalent ?
4 //solution :
5 clc //clears the
  command window
6 clear //clears
  the variables
7 q=0;
8 b=0;
9 a=input("Enter the decimal no:") //Enter
  the decimal nubcr
10 format('v',18) //
  increasing the precision to 18
11 a=floor(a);
12 h=dec2hex(a);
13 while(a>0) //
  converting to binary
14     x=modulo(a,2);
15     b= b + (10^q)*x;
16     a=a/2;
17     a=floor(a);
18     q=q+1;
19 end
20 printf("The hexadecimal equivalent is = %s\n",h)

```

```
21 // displaying
    the
    results
22 printf("The binary equivalent is = %f",b)
```

---

# Chapter 6

## Arithmetic Circuits

Scilab code Exa 6.1 8bit binary adder

```
1 //exmple 6.1
2 clc;
3 clear;
4 //a=input("enter the first 8 bit number :");
5 //b=input("enter the second 8 bit number :");
6 a=01010111; // taking given inputs
7 b=00110101;
8 for i=1:8
9     a1(i)=modulo(a,10);
10    a=a/10;
11    a=round(a);
12    b1(i)=modulo(b,10);
13    b=b/10;
14    b=round(b);
15 end
16 car(1)=0;
17 for i=1:8 // adding both the inputs (binary
18    addition)
19    c1(i)=car(i)+a1(i)+ b1(i);
20    if c1(i)== 2 then
21        car(i+1)= 1;
```



```

21         c1(i)=0;
22     elseif c1(i)==3 then
23         car(i+1)= 1;
24         c1(i)=1;
25     else
26         car(i+1)=0;
27     end
28 end
29 c1(9)=car(9);
30 re=0;
31 format('v',18);
32 for i=1:9
33     re=re+(c1(i)*(10^(i-1)))
34 end
35 printf('The sum of given two binary numbers is %d\n',
        ,re );
36 q=1;
37 b=0;
38 f=0;
39 a=re;
40 while(a>0)           // converting the result to a
        hexadecimal no
41     r=modulo(a,10);
42     b(1,q)=r;
43     a=a/10;
44     a=floor(a);
45     q=q+1;
46 end
47 for m=1:q-1
48     c=m-1;
49     f = f + b(1,m)*(2^c);
50 end
51 hex=dec2hex(f);
52 printf(' The sum in hexadecimal notation is %s \n',
        hex); //displaying result

```

---

### Scilab code Exa 6.2 16 bit binary adder

```
1 //exmple 6.2
2 clc;
3 clear;
4 //a=input(" enter the first 16 bit binary number :");
5 //b=input(" enter the second 16 bit binarynumber :");
6 a=0000111110101100;
7 b=0011100001111111;
8 for i=1:16
9     a1(i)=modulo(a,10);
10    a=a/10;
11    a=round(a);
12    b1(i)=modulo(b,10);
13    b=b/10;
14    b=round(b);
15 end
16 car(1)=0;
17 for i=1:16    /// adding both the 16-bit inputs (
    binary addition)
18    c1(i)=car(i)+a1(i)+ b1(i);
19    if c1(i)== 2 then
20        car(i+1)= 1;
21        c1(i)=0;
22    elseif c1(i)==3 then
23        car(i+1)= 1;
24        c1(i)=1;
25    else
26        car(i+1)=0;
27    end
28 end
29 c1(17)=car(17);
30 re=0;
31 format('v',25);
```

```

32 for i=1:17
33     re=re+(c1(i)*(10^(i-1)))
34 end
35 printf('The sum of given two binary numbers is %f\n',
        ,re );
36 q=1;
37 b=0;
38 f=0;
39 a=re;
40 while(a>0)    // converting the result to a
                hexadecimal no
41     r=modulo(a,10);
42     b(1,q)=r;
43     a=a/10;
44     a=floor(a);
45     q=q+1;
46 end
47 for m=1:q-1
48     c=m-1
49     f = f + b(1,m)*(2^c);
50 end
51 hex=dec2hex(f);
52 printf(' Sum in decimal notation is %d\n',f);
53 printf(' Sum in hexadecimal notation is %s \n',hex);
    //displaying result

```

---

### Scilab code Exa 6.3 first generation microcomputers addition

```

1 //example 6.3
2 clc;
3 clear;
4 //a=input("enter the first 8 bit number :");
5 //b=input("enter the second 8 bit number :");
6 a=0000111110101100;
7 b=0011100001111111;

```

```

8  for i=1:16
9      a1(i)=modulo(a,10);
10     a=a/10;
11     a=round(a);
12     b1(i)=modulo(b,10);
13     b=b/10;
14     b=round(b);
15 end
16 car(1)=0;
17 for i=1:8
18     c1(i)=car(i)+a1(i)+ b1(i);    // adding the
        Higher bytes
19     if c1(i)== 2 then
20         car(i+1)= 1;
21         c1(i)=0;
22     elseif c1(i)==3 then
23         car(i+1)= 1;
24         c1(i)=1;
25     else
26         car(i+1)=0;
27     end
28 end
29 c1(9)=car(9)
30 re=0;
31 format('v',18);
32 for i=1:9
33     re=re+(c1(i)*(10^(i-1)))
34 end
35
36 printf('The sum of lower bytes of two binary numbers
        is %d\n',re );
37 printf(' with a carry is %d\n',car(9));
38 re=re-(c1(9)*(10^(8)))
39 for i=9:16    // adding the Higher bytes
40     c1(i)=car(i)+a1(i)+ b1(i);
41     if c1(i)== 2 then
42         car(i+1)= 1;
43         c1(i)=0;

```

```

44     elseif c1(i)==3 then
45         car(i+1)= 1;
46         c1(i)=1;
47     else
48         car(i+1)=0;
49     end
50 end
51 c1(17)=car(17);
52 format('v',25);
53 ree=0;
54 for i=9:17
55     ree=ree+(c1(i)*(10^(i-9)));
56 end
57 for i=9:17
58     re=re+(c1(i)*(10^(i-1)))
59 end
60 printf(' The sum of upper bytes of the given
        numbers is %d\n',ree);
61 printf(' with a carry is %d\n',car(17)); //
        displaying results
62 printf(' The total sum is %f',re );

```

---

#### Scilab code Exa 6.4 binary subtraction

```

1 //exmple 6.4
2 clc;
3 clear;
4 a=0;
5 b=0;
6 q=0;
7 //bb=input(" Enter the first no (in decimal) :");
8 //aa=input(" Enter the number from which first no
        has to substracted:");
9 aa=200; // taking the given input
10 bb=125;

```

```

11 while(aa>0)      // converting the inputs in to binary
    numbers
12     x=modulo(aa,2);
13     a= a + (10^q)*x;
14     aa=aa/2;
15     aa=floor(aa);
16     q=q+1;
17 end
18 q=0;
19 while(bb>0)
20     x=modulo(bb,2);
21     b= b + (10^q)*x;
22     bb=bb/2;
23     bb=floor(bb);
24     q=q+1;
25 end
26 printf(' \nThe binary equivalent of first no is %f\n
    \n',b);
27 printf(' The binary equivalent of secnd no is %f\n\n
    ',a);
28 for i=1:40
29     a1(i)=modulo(a,10);
30     a=a/10;
31     a=round(a);
32     b1(i)=modulo(b,10);
33     b=b/10;
34     b=round(b);
35 end
36 bro(1)=0;
37 for i=1:40
38     c1(i)=a1(i)- b1(i)-bro(i); // finding the
        difference of the given inputs
39     if c1(i)== -1 then
40         bro(i+1)= 1;
41         c1(i)=1;
42     elseif c1(i)==-2 then
43         bro(i+1)= 1;
44         c1(i)=0;

```

```

45     else
46         bro(i+1)=0;
47     end
48
49 end
50 re=0;
51 format('v',18);
52 for i=1:40
53     re=re+(c1(i)*(10^(i-1)))
54 end
55 printf(' The diference of given two numbers is %f\n\n
        n',re );
56 q=1;
57 b=0;
58 f=0;
59 a=re;
60 while(a>0)
61     r=modulo(a,10);
62     b(1,q)=r;
63     a=a/10;
64     a=floor(a);
65     q=q+1;
66 end
67 for m=1:q-1
68     c=m-1
69     f = f + b(1,m)*(2^c);
70 end
71 hex=dec2hex(f);
72 printf(' Sum in decimal notation is %d\n\n',f); //
        displaying the results
73 printf(' Sum in hexadecimal notation is %s \n',hex);

```

---

Scilab code Exa 6.5 adding 8 bit unsigned numbers

```
1 //example 6.5
```

```

2
3 clc;
4 clear;
5 a=0;
6 b=0;
7 q=0;
8
9 //aa=input(" Enter the first no (in decimal) :");
10 //bb=input(" Enter the number from which first no
    has to substracted:");
11 aa=150;
12 bb=85;
13 while(aa>0)           // converting the inputs in
    to binary numbers
14     x=modulo(aa,2);
15     a= a + (10^q)*x;
16     aa=aa/2;
17     aa=floor(aa);
18     q=q+1;
19 end
20 q=0;
21 while(bb>0)
22     x=modulo(bb,2);
23     b= b + (10^q)*x;
24     bb=bb/2;
25     bb=floor(bb);
26     q=q+1;
27 end
28 printf(' \n The binary equivalent of first no is %f\
    n\n',a);
29 printf(' The binary equivalent of secnd no is %f\n\n
    ',b);
30 for i=1:40
31     a1(i)=modulo(a,10);
32     a=a/10;
33     a=round(a);
34     b1(i)=modulo(b,10);
35     b=b/10;

```



```

36     b=round(b);
37 end
38
39 car(1)=0;
40 for i=1:40
41     c1(i)=car(i)+a1(i)+ b1(i); // addng both the
        inputs
42     if c1(i)== 2 then
43         car(i+1)= 1;
44         c1(i)=0;
45     elseif c1(i)==3 then
46         car(i+1)= 1;
47         c1(i)=1;
48     else
49         car(i+1)=0;
50     end
51 end
52 c1(41)=car(41);
53 re=0;
54 format('v',18);
55 for i=1:41
56     re=re+(c1(i)*(10^(i-1)))
57 end
58 printf(' The sum of given two binary numbers is %f\n
        \n',re );
59 q=1;
60 b=0;
61 f=0;
62 a=re;
63 while(a>0)
64     r=modulo(a,10);
65     b(1,q)=r;
66     a=a/10;
67     a=floor(a);
68     q=q+1;
69 end
70 for m=1:q-1
71     c=m-1;

```

```

72     f = f + b(1,m)*(2^c);
73 end
74 printf(' Sum in decimal notation is %d\n\n',f); //
    displaying results
75 hex=dec2hex(f);
76 printf(' The sum in hexadecimal notation is %sH \n',
    hex);

```

---

### Scilab code Exa 6.6 subtraction of unsigned numbers

```

1 //exmple 6.6
2 clc;
3 clear;
4 a=0;
5 b=0;
6 q=0;
7 //bb=input(" Enter the first no (in decimal) :");
8 //aa=input(" Enter the number from which first no
    has to substracted:");
9 aa=150;
10 bb=85;
11 while(aa>0)  //// converting the inputs in to binary
    numbers
12     x=modulo(aa,2);
13     a= a + (10^q)*x;
14     aa=aa/2;
15     aa=floor(aa);
16     q=q+1;
17 end
18 nn=a
19 q=0;
20 while(bb>0)
21     x=modulo(bb,2);
22     b= b + (10^q)*x;
23     bb=bb/2;

```

```

24     bb=floor(bb);
25     q=q+1;
26 end
27 printf(' \nThe binary equivalent of first no is %f\n
        \n',b);
28 printf(' The binary equivalent of secnd no is %f\n\n
        ',a);
29 for i=1:40
30     a1(i)=modulo(a,10);
31     a=a/10;
32     a=round(a);
33     b1(i)=modulo(b,10);
34     b=b/10;
35     b=round(b);
36 end
37
38 bro(1)=0;
39 for i=1:40
40     c1(i)=a1(i)- b1(i)-bro(i);
41     if c1(i)== -1 then
42         bro(i+1)= 1;
43         c1(i)=1;
44     elseif c1(i)==-2 then
45         bro(i+1)= 1;
46         c1(i)=0;
47     else
48         bro(i+1)=0;
49     end
50
51 end
52 re=0;
53 format('v',18);
54 for i=1:40
55     re=re+(c1(i)*(10^(i-1)))
56 end
57 printf(' The diference of given two numbers is %f\n\n
        n',re );
58 q=1;

```

```

59 b=0;
60 f=0;
61 a=re;
62 while(a>0) // converting the binary result to
    decimal then to hexadecimal
63     r=modulo(a,10);
64     b(1,q)=r;
65     a=a/10;
66     a=floor(a);
67     q=q+1;
68 end
69 for m=1:q-1
70     c=m-1
71     f = f + b(1,m)*(2^c);
72 end
73 hex=dec2hex(f);
74 printf(' Sum in decimal notation is %d\n\n',f);
75 printf(' Sum in hexadecimal notation is %s \n',hex);

```

---

#### Scilab code Exa 6.7 overflow case

```

1 //example 6.7
2 clc;
3 clear;
4 a=0;
5 b=0;
6 q=0;
7 //aa=input(" Enter the first no (in decimal) :");
8 //bb=input(" Enter the number from which first no
    has to subtracted:");
9 aa=175;
10 bb=118;
11 while(aa>0) // converting the inputs in to binary
    numbers
12     x=modulo(aa,2);

```

```

13     a= a + (10^q)*x;
14     aa=aa/2;
15     aa=floor(aa);
16     q=q+1;
17 end
18 q=0;
19 while(bb>0)
20     x=modulo(bb,2);
21     b= b + (10^q)*x;
22     bb=bb/2;
23     bb=floor(bb);
24     q=q+1;
25 end
26 printf(' \n The binary equivalent of first no is %f\
n\n',a);
27 printf(' The binary equivalent of secnd no is %f\n\n
',b);
28 for i=1:8
29     a1(i)=modulo(a,10);
30     a=a/10;
31     a=round(a);
32     b1(i)=modulo(b,10);
33     b=b/10;
34     b=round(b);
35 end
36
37 car(1)=0;
38 for i=1:8
39     c1(i)=car(i)+a1(i)+ b1(i); //adding the binary
        numbers (binary addtion)
40     if c1(i)== 2 then
41         car(i+1)= 1;
42         c1(i)=0;
43     elseif c1(i)==3 then
44         car(i+1)= 1;
45         c1(i)=1;
46     else
47         car(i+1)=0;

```

```

48     end
49 end
50 c1(9)=car(9);
51 re=0;
52 format('v',18);
53 for i=1:8
54     re=re+(c1(i)*(10^(i-1)))
55 end
56 printf('If only 8 bits are taken the result will be
        as shown below \n\n');
57 printf(' and the sum of given two binary numbers
        will be %f\n\n',re );
58 q=1;
59 b=0;
60 f=0;
61 a=re;
62 while(a>0) //converting the binary output to
        hexadecimal
63     r=modulo(a,10);
64     b(1,q)=r;
65     a=a/10;
66     a=floor(a);
67     q=q+1;
68 end
69 for m=1:q-1
70     c=m-1;
71     f = f + b(1,m)*(2^c);
72 end
73 printf(' Sum in decimal notation is %d\n\n',f);
74 hex=dec2hex(f);
75 printf(' The sum in hexadecimal notation is %sH \n',
        hex);
76 printf(' \n with an overflow of %d\n\n',car(9));

```

---

Scilab code Exa 6.8 2s compliment

```

1 //example 6.8
2 clc;
3 clear;
4 re=0;
5 aaa=input('enter the number(in decimal) :');//
   taking the signed number
6 m=aaa;
7 if aaa<0 then
8     aa=-1*aaa;
9     else aa=aaa;
10 end
11 a=0;
12 q=0;
13 while(aa>0) //converting from decimal to
   binary
14     x=modulo(aa,2);
15     a= a + (10^q)*x;
16     aa=aa/2;
17     aa=floor(aa);
18     q=q+1;
19 end
20 mm=a;
21 for i=1:8
22     a1(i)=modulo(a,10);
23     a=a/10;
24     a=round(a);
25     b1(i)=0;
26 end
27 b1(1)=1;
28 if aaa<0 then // making two's complement if the
   number is less than zero
29     for i=1:8
30         a1(i)=bitcmp(a1(i),1);
31     end
32     car(1)=0;
33 for i=1:8
34     c1(i)=car(i)+a1(i)+ b1(i); // adding one (as a
   part of finding 2's compliment

```

```

35     if c1(i)== 2 then
36         car(i+1)= 1;
37         c1(i)=0;
38     elseif c1(i)==3 then
39         car(i+1)= 1;
40         c1(i)=1;
41     else
42         car(i+1)=0;
43     end;
44 end;
45 c1(9)=car(9);
46 re=0;
47 format('v',18);
48     for i=1:9
49         re=re+(c1(i)*(10^(i-1)))
50     end;
51     printf('\nThe binary contents are %d\n\n',
52           re );
52 else
53     re=mm;
54     end;
55     if(aaa>0)
56     printf('\nThe biary contents are %d\n\n',mm);
57 end;
58 q=1;
59 b=0;
60 f=0;
61 a=re;
62 while(a>0) // converting the result to decimal then
63     to hexadecimal
64     r=modulo(a,10);
65     b(1,q)=r;
66     a=a/10;
67     a=floor(a);
68     q=q+1;
69 end
70 for m=1:q-1
71     c=m-1

```



```

71     f = f + b(1,m)*(2^c);
72 end
73 hex=dec2hex(f);
74 printf('The Hexadecimal contents are %sH',hex) ;//
    displayin the result

```

---

### Scilab code Exa 6.9 2s compliment

```

1 //example 6.9
2 clc;
3 clear;
4 //aaa=input('enter the number(in decimal) :');
5 aaa=-19750 // given input
6 aa=-1*aaa;
7 format('v',18);
8 a=0;
9 q=0;
10 while(aa>0) // converting it to binary
11     x=modulo(aa,2);
12     a= a + (10^q)*x;
13     aa=aa/2;
14     aa=floor(aa);
15     q=q+1;
16 end
17 for i=1:16
18     a1(i)=modulo(a,10);
19     a=a/10;
20     a=round(a);
21     b1(i)=0;
22 end
23 b1(1)=1;
24 for i=1:16 // finding the 2's
    compliment
25     a1(i)=bitcmp(a1(i),1);
26     end

```

```

27     car(1)=0;
28     for i=1:16
29         c1(i)=car(i)+a1(i)+ b1(i);
30         if c1(i)== 2 then
31             car(i+1)= 1;
32             c1(i)=0;
33         elseif c1(i)==3 then
34             car(i+1)= 1;
35             c1(i)=1;
36         else
37             car(i+1)=0;
38         end;
39     end;
40     c1(17)=car(17);
41     re=0;
42     for i=1:17
43         re=re+(c1(i)*(10^(i-1)))
44     end;
45     printf('\n The 2''s compliment is');
46     disp(re);
47     q=1;
48     b=0;
49     f=0;
50     a=re;
51     while(a>0) // converting to hexadecimal
52         r=modulo(a,10);
53         b(1,q)=r;
54         a=a/10;
55         a=floor(a);
56         q=q+1;
57     end
58     for m=1:q-1
59         c=m-1
60         f = f + b(1,m)*(2^c);
61     end
62     hex=dec2hex(f);
63     printf('\n In Hexadecimal notation is %sH\n\n',hex)
        ;// displaying the result

```

```
64 disp('As the memory of a first generation
      microcomputer is organised in bytes . The lower
      byte is stored in 2000 address and the higher
      byte is stored in 2001 address.');
```

---

### Scilab code Exa 6.10 2s compliment subtraction

```
1 //example 6.10
2 clc;
3 clear;
4 format('v',18);
5 //bb=input('enter the first number(in decimal):' );
6 //aaa=input('enter the second number(negative) :');
7 aaa=-12618
8 bb=18357;
9 aa=-1*aaa;
10 a=0;
11 q=0;
12 while(aa>0) // finding the binary
    equivalent
13     x=modulo(aa,2);
14     a= a + (10^q)*x;
15     aa=aa/2;
16     aa=floor(aa);
17     q=q+1;
18 end
19 r=0;
20 b=0;
21 while(bb>0)
22     x=modulo(bb,2);
23     b= b + (10^r)*x;
24     bb=bb/2;
25     bb=floor(bb);
26     r=r+1;
27 end
```

```

28 m=b
29 for i=1:16
30     a1(i)=modulo(a,10);
31     a=a/10;
32     a=round(a);
33     p1(i)=0;
34     b1(i)=modulo(b,10);
35     b=b/10;
36     b=round(b);
37 end
38 p1(1)=1;
39 for i=1:16 // finding the 2's compliment
    of second number
40     a1(i)=bitcmp(a1(i),1);
41     end
42     car(1)=0;
43 for i=1:16
44     c1(i)=car(i)+a1(i)+ p1(i);
45     if c1(i)== 2 then
46         car(i+1)= 1;
47         c1(i)=0;
48     elseif c1(i)==3 then
49         car(i+1)= 1;
50         c1(i)=1;
51     else
52         car(i+1)=0;
53     end;
54 end;
55 re=0;
56 for i=1:16
57     re=re+(c1(i)*(10^(i-1)))
58 end;
59 printf(' The binary representation of first
    number is ');
60 disp(m);
61 printf(' The 2''s compliment of second
    nber is ');
62 disp(re);

```

```

63 a1=c1;
64 ar(1)=0;
65 for i=1:8
66     c1(i)=ar(i)+a1(i)+ b1(i); // addin both the
        numbers (binary addition)
67     if c1(i)== 2 then // lower byte
68         ar(i+1)= 1;
69         c1(i)=0;
70     elseif c1(i)==3 then
71         ar(i+1)= 1;
72         c1(i)=1;
73     else
74         ar(i+1)=0;
75     end
76 end
77 c1(9)=ar(9)
78 re=0;
79 format('v',18);
80 for i=1:8
81     re=re+(c1(i)*(10^(i-1)))
82 end
83 printf(' The sum of lower bytes of two binary
        numbers is %d\n',re );
84 printf(' with a carry is %d\n',ar(9));
85 for i=9:16
86     c1(i)=ar(i)+a1(i)+ b1(i); // upper byte
87     if c1(i)== 2 then
88         ar(i+1)= 1;
89         c1(i)=0;
90     elseif c1(i)==3 then
91         ar(i+1)= 1;
92         c1(i)=1;
93     else
94         ar(i+1)=0;
95     end
96 end
97 c1(17)=ar(17);
98 format('v',25);

```

```

99 ree=0;
100 for i=9:16
101     ree=ree+(c1(i)*(10^(i-9)));
102 end
103 for i=9:16
104     re=re+(c1(i)*(10^(i-1)))
105 end
106 printf(' The sum of upper bytes of the given
        numbers is %d\n',ree);
107 printf(' with a carry is %d\n',ar(17)); // displaying
        results
108 printf(' The total sum is ' );
109 disp(re);
110 printf(' with a carry %d',ar(17));

```

---

#### Scilab code Exa 6.12 final carry in a CLA

```

1 //exmple 6.12
2 clc;
3 clear;
4 //a=input('Enter the first number A (4 bit ) :');
5 //b=input('Enter the first number B (4 bit ) :');
6 a=1111; // given values for a and b
7 b=0001;
8 for i=1:4
9     a1(i)=modulo(a,10);
10    a=a/10;
11    a=round(a);
12    b1(i)=modulo(b,10);
13    b=b/10;
14    b=round(b);
15 end
16 for i=1:4 //finding the generate and propagate
        values for ech bit
17    g(i)=bitand(a1(i),b1(i));

```

```
18     p(i)=bitor(a1(i),b1(i));
19 end
20 c(1)=0;
21 for i=1:4 // finding the carry
22     c(i+1)= bitor(g(i),bitand(p(i),c(i)));
23 end
24 printf('The final carry is C3 = %d', c(5));//
    displaying the result
```

---

# Chapter 7

## Clocks and Timing Circuits

Scilab code Exa 7.1 clock cycle time

```
1 //example 7.1
2 clear;
3 clc;
4 //for 50 kHz clock
5 clk_frq1 = 50000;
6 c_t_500 = 1000/clk_frq1;
7 //for 8-MHz clock
8 clk_frq2 = 8000000;
9 c_t_8 = 10000000/clk_frq2;
10 printf('Cycle time for 500-kHz clock is %f
        milliseconds \n',c_t_500); //displaying results
11 printf(' Cycle time for 8-MHz clock is %f micro
        seconds ',c_t_8);
```

---

Scilab code Exa 7.2 maximum clock frequency

```
1 //example 7.2
2 clc;
```



```

3 clear;
4 //prop_delay = input('Enter the propagation delay in
    nano seconds:');
5 prop_delay=24; // taking the given input
6 format('v',18); //setting the precision
7 max_clk_frq = 1/prop_delay; // making necessary
    calculations
8 max_clk_frq = max_clk_frq*10^3;
9 printf('maximum clock frequency is %f KHz',
    max_clk_frq); // displaying the result

```

---

Scilab code Exa 7.3 frequency limits of the clock

```

1 //example 7.3
2 clc;
3 clear;
4 //ppm = input('Enter the stability in parts per
    million (PPM):');
5 //clk_frq = input('Enter the clock frequency in MHz
    :');
6 ppm= 5//taking the given values
7 clk_frq= 5
8 mill= clk_frq; //making necessary calculations
9 pp = mill*ppm;
10 pp = round(pp);
11 clk_frq = clk_frq*10^6;
12 o(1,1)=clk_frq-pp;
13 o(1,2)=clk_frq+pp;
14 printf('The clock frequency will be somewhere
    between %d and %d Hz',o); //displaying the result

```

---

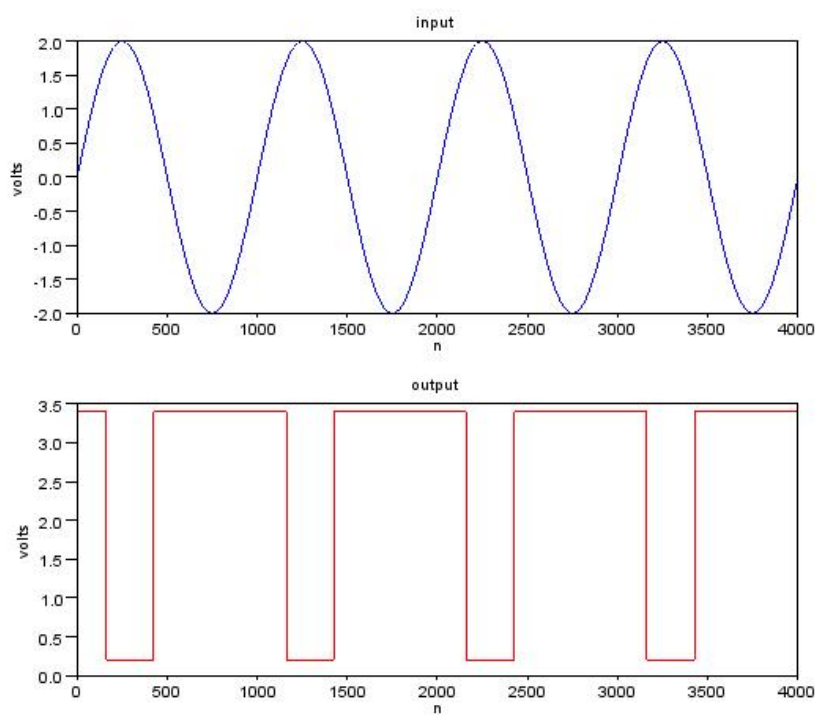


Figure 7.1: Schmitt trigger

### Scilab code Exa 7.4 Schmitt trigger

```
1 //example 7.4
2 //schmitt trigger inverter
3 clear;
4 clc;
5 close;
6 //peak= input('Enter the peak voltage of sine wave
   in volts :');
7 //utp = input('Enter the upper trigger point in
   volts :');
8 //ltp = input('Enter the lower trigger point in
   volts :');
9 peak =2; // taking given values for inputs
10 utp=1.7;
11 ltp=0.9;
12 for i=1:4000
13     sinn(i) = peak * sin(i*2*3.1416/1000); //drawing
       a sin wave with given amplitude
14 end
15 for j=1:4000 // making calculations to plot
       output
16     if modulo(j,1000)< 250 then
17         if sinn(j)<utp then
18             result(j)=3.4;
19         else
20             result(j)=0.2;
21         end
22         elseif sinn(j)>ltp then
23             result(j)=0.2
24         else
25             result(j)=3.4;
26     end
27 end
28 subplot(2,1,1)
29 plot(sinn); //ploting the input and output curves
30 xlabel('n');
31 title('input')
```

```
32 ylabel('volts');
33
34 subplot(2,1,2);
35 plot(result,'r');
36 xlabel('n');
37 title('output')
38 ylabel('volts');
```

---

**Scilab code Exa 7.5** frequency of oscillation for 555 timer

```
1 //example 7.5
2 clc
3 clear
4 //Ra = input('Enter the value of the resistance RA
   in Kohms :');
5 //Rb = input('Enter the value of the resistance RB
   in Kohms :');
6 //C =input('Enter the value of the Capacitance C in
   micro farads :');
7 Ra=1 //taking the given input
8 Rb=1
9 C= 1 *10^-3
10 T= (Ra + 2*Rb)*C;
11 frq = 1.44 * (1/T); //substituting in the equation
12 printf('Frequency of oscillation is %f KHz',frq); //
   displaying result
```

---

**Scilab code Exa 7.6** finding Ra and C in 555 timer circuit

```
1 //example 7.6
2 clc
3 clear
```

```

4 //rb=input('Enter the value of the resistance RB in
    Kohms :');
5 //dc =input('Enter required duty cycle in % :');
6 //clk = input('Enter the provided clock frequency in
    MHz: ');
7 rb=0.75//taking the given values for input
8 dc=25
9 clk=1
10 ra = (rb*100/dc) - 2*rb;//mking neccesary
    calculations
11 format('v',18);
12 t2 = dc/(clk*10^8);
13 C = t2/(693*rb)
14 C=C*10^12;
15 //C=round(C);
16 printf('The value of RA is %f Kohms\n',ra);//
    displaying the output
17 printf('The value of C is %f pico farads',C);

```

---

Scilab code Exa 7.7 output pulse width for the timer

```

1 //example 7.7
2 clc
3 clear
4 //Ra = input('Enter the value of the resistance RA
    in Kohms :');
5 //C =input('Enter the value of the Capacitance C in
    micro farads :');
6 Ra=10//taking given values
7 C=0.1
8 pw = 1.1*Ra*C //substituting in the equation
9 printf('pulse width is %f milliseconds',pw);//
    displaying result

```

---

Scilab code Exa 7.8 value of C necessary to change pulse width to given values

```
1 //example 7.8
2 clc
3 clear
4 //Ra = input('Enter the value of the resistance RA
   in Kohms :');
5 //pw =input('Enter the value of required pulse width
   in millisecondseconds :');
6 Ra=10//taking given values
7 pw=10
8 C = pw/(1.1*Ra); //substituting in the equation
9 printf('The required value of capacitance is %f
   microfarads ',C);//displaying result
```

---

Scilab code Exa 7.9 monostable multivibrator

```
1 //example 7.9
2 clear
3 clc
4 close
5 //R = input('Enter the value of the resistance R in
   Kohms :');
6 //C =input('Enter the value of the Capacitance C in
   micro farads :');
7 sp = input('Enter the spacing between two input
   pulses in micro seconds :');
8 R=1; //taking give values
```

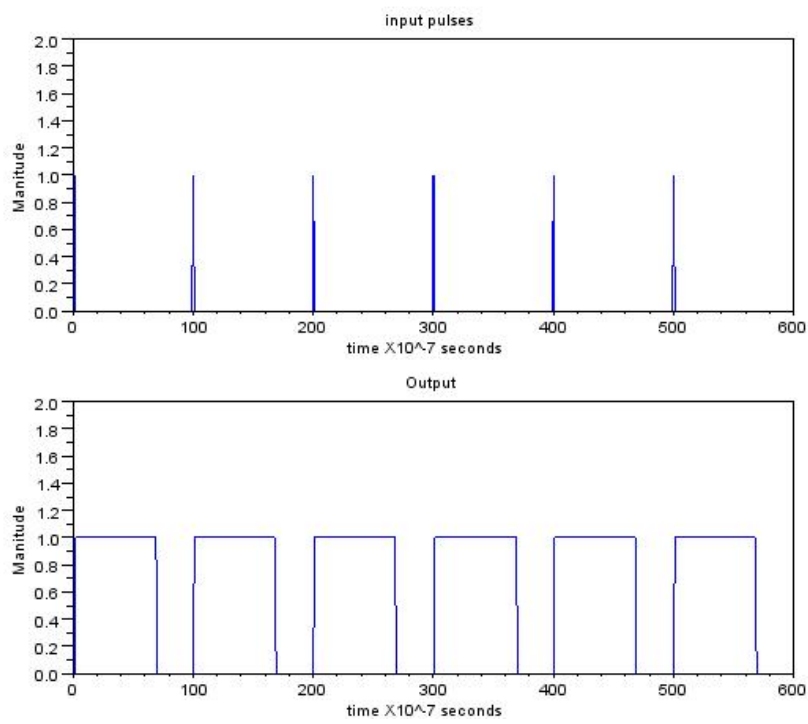


Figure 7.2: monostable multivibrator

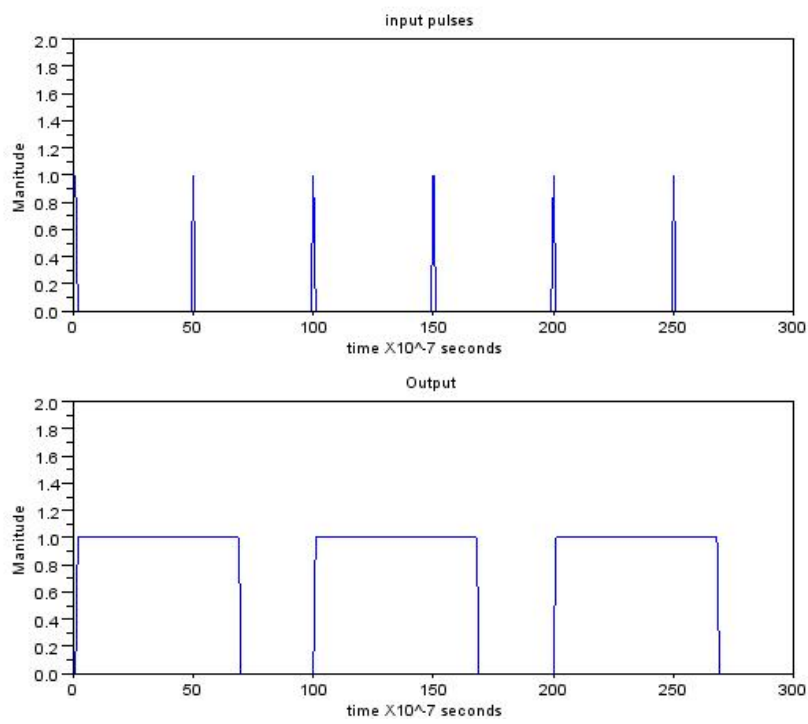


Figure 7.3: monostable multivibrator



```

 9 C=0.01;
10 t= 693*R*C; // calculating time constant
11 tt=t*10;
12 p=1;
13 len=sp*60-1;
14 q=1;
15 for j=1:len //plotin the graphs
16     lo = sp*10;
17     f=modulo(j,lo);
18     if f==0 then
19         inpu(j)=1;
20     else
21         inpu(j)=0;
22     end
23     inpu(1)=1;
24 o(j)=2;
25 end
26 while q<len
27     result(q)=0;
28     q=q+1;
29 end
30 while p<len
31     if inpu(p)==1 then
32         for k=1:tt
33             result(p+k)=1;
34         end
35         p=p+tt;
36     else
37         result(p)=0;
38         p=p+1;
39     end
40 end
41 subplot(2,1,1); // plotting bothe graphs in same
    window
42 plot(o);
43 plot(inpu);
44 xlabel('time X10-7 seconds');
45 ylabel('Manitude') ;

```

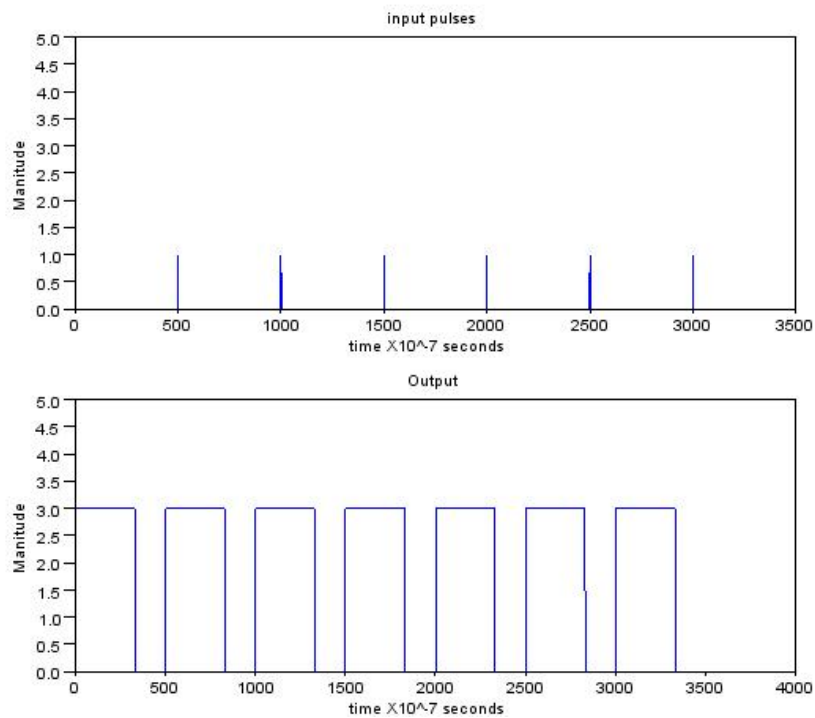


Figure 7.4: 74123

```

46 title('input pulses ');
47     subplot(2,1,2);
48     plot(o);
49 plot(result);
50 xlabel('time X10-7 seconds ');
51 ylabel('Manitude') ;
52 title('Output ');

```

---

Scilab code Exa 7.10 74123

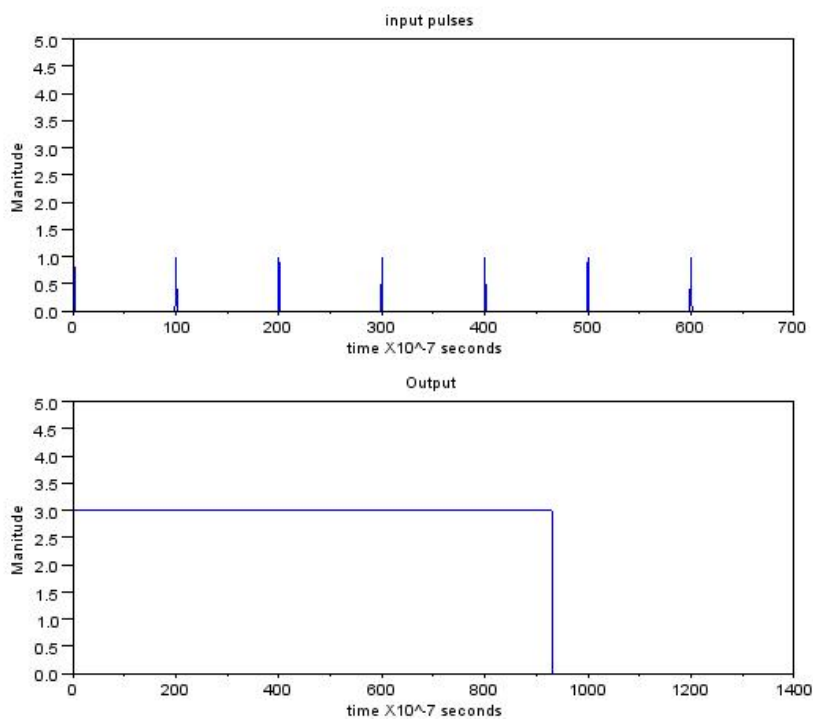


Figure 7.5: 74123

```

1 //example 7.10
2 clear
3 clc
4 close
5 //R = input('Enter the value of the resistance R in
    Kohms :');
6 //C =input('Enter the value of the Capacitance C in
    micro farads :');
7 sp = input('Enter the spacing between two input
    pulses in micro seconds :');
8 R=10;
9 C=0.01;
10 //sp=50;
11 //sp=10;
12 t= 330*R*C; // calculating time constant
13 printf('The output pulse width will be about %f
    micro seconds ',t);
14 tt=t*10;
15 p=1;
16 len=sp*60+1;
17 q=1;
18 for j=1:len // making arrays to plot the graphs
19     lo = sp*10;
20     f=modulo(j,lo);
21     if f==0 then
22         inpu(j)=1;
23     else
24         inpu(j)=0;
25     end
26     inpu(1)=1;
27 o(j)=5;
28 end
29 if sp<40 then
30 while q<2*len
31     result(q)=0;
32     q=q+1;
33 end
34 else

```

```

35 while q<1.2*len
36     result(q)=0;
37     q=q+1;
38 end
39 end
40 while p<len
41     if inpu(p)==1 then
42         for k=1:tt
43             result(p+k)=3;
44         end
45         p=p+1;
46     else
47         p=p+1;
48     end
49
50     end
51 subplot(2,1,1); // plotting bothe graphs in same
    window
52 plot(o);
53 plot(inpu);
54 xlabel('time X10-7 seconds');
55 ylabel('Manitude') ;
56 title('input pulses');
57     subplot(2,1,2);
58     plot(o);
59 plot(result);
60 xlabel('time X10-7 seconds');
61 ylabel('Manitude') ;
62 title('Output');

```

---

Scilab code Exa 7.11 finding timing capacitor values

```

1 //example 7.11
2 clc
3 clear

```

```
4 //R = input('Enter the value of the resistance R in
    Kohms :');
5 //pw =input('Enter the value of required pulse width
    in millisecondseconds :');
6 R=0.5;//taking the given values
7 format('v',18);
8 pw = 1 * 10^-3;
9 C = pw/(0.33*R); //calculating C
10 printf('The required value of capacitance is %f
    microfarads\n',C);
11 printf('The pulse delay capacitor is %f microfarads'
    ,(2*C));
```

---

# Chapter 8

## Flip Flops

Scilab code Exa 8.4 RS flipflop

```
1 //example 8.4
2 clc;
3 clear;
4 en=input("Enter the enable input level(1 or 0) : ")
   ;
5 r=input("enter the R input level(1 or 0) : ");//
   accepting the inputs from the user
6 s=input("enter the S input level(1 or 0) : ");
7 qn=input("Enter the previous output value(1 or 0) :
   ");
8
9 if en == 0 then // calculating the output
10     op = qn;
11 elseif (s==0 & r==0) then
12     op=qn;
13 elseif(s==1&r==1) then
14     disp('The inputs are illegal');
15     return;
16 else
17     op=s;
18
```

```

19 end
20
21 printf('\n \noutput (Qn+1) = %d ',op); // displaying
    the output

```

---

#### Scilab code Exa 8.5 positive edge triggered RS flip flop

```

1 //example 8.5
2 clc;
3 clear;
4 disp('Here what happens at each point in time' );
5 disp('Time t0: S = 0, R = 0, no change in Q (Q
    remains 0) ');
6 disp('Time t1: S = 1, R = 0, Q changes from 0 to 1
    ');
7 disp('Time t2: S = 0, R = 1, Q resets to 0');
8 disp('Time t3: S = 1, R = 0, Q sets to 1 ');
9 disp('Time t4: S = 0, R = 0, no change in Q (Q
    remains 1) ');
10 disp('Notice that either R or S, or both, are
    allowed to change state at any time, whether C is
    high or low. The only time both R and S must be
    stable (unchanging) is during the short PTs of
    the clock. ');

```

---

#### Scilab code Exa 8.6 negative edge triggered RS flip flop

```

1 //example 8.6
2 clc;
3 clear;
4 disp('Here what happens at each point in time' );
5 disp('Time t0: S = 0, R = 0, no change in Q (Q
    remains 0) ');

```



```

6 disp('Time t1: S = 1, R = 0, Q changes from 0 to 1
      ');
7 disp('Time t2: S = 0, R = 1, Q resets to 0');
8 disp('Time t3: S = 1, R = 0, Q sets to 1 ');
9 disp('Time t4: S = 0, R = 0, no change in Q (Q
      remains 1)');
10 disp('Notice that either R or S, or both, are
      allowed to change state at any time, whether C is
      high or low. The only time both R and S must be
      stable (unchanging) is during the short NTs of
      the clock.');
```

---

#### Scilab code Exa 8.7 T flip flop

```

1 //example 8.7
2 clc;
3 clear;
4 close;
5 printf("For input J and K = 0 output Qn+1 = Qn i.e
      output does not change its state And for J = K =
      1, The Output Qn+1 = Qn i.e output toggles ");
```

---

#### Scilab code Exa 8.9 JK master slave

```

1 //example 8.9
2 clc
3 close
4 clear
5 disp("since J=K=1, the flip-flop simply toggles each
      time the clock goes low, The waveform at Q has a
      period twice of that of the waveform. In other
```

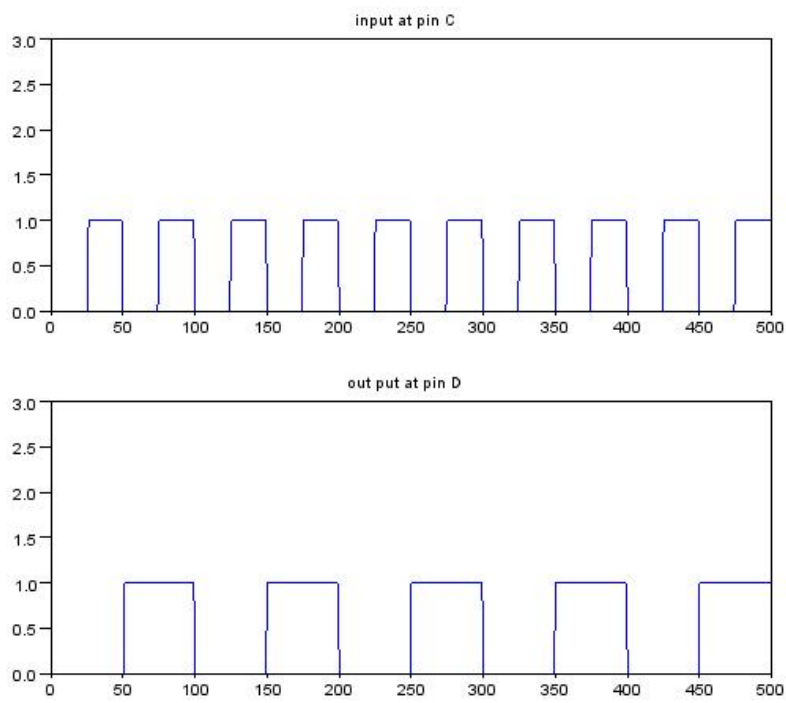


Figure 8.1: JK master slave

words, the frequency of Q is only one-half of that of . This circuit acts as a frequency divider —the output frequency divide by 2. Note that Q changes state on NTs of the clock. The waveforms are as shown in the figure ”);

```

6 t=50; // taken time period
7 p=1;
8 while p<t*10 // taking values for plotting the graph
9     if p==1 | modulo(p,t)==0 then
10         for k=1:t/2
11             cin(p+k)=0;
12         end
13         p=p+t/2;
14     else
15         cin(p)=1;
16         p=p+1;
17     end
18 end
19 t=100;
20 p=1;
21 while p<t*5
22     if p==1 | modulo(p,t)==0 then
23         for k=1:t/2
24             dout(p+k)=0;
25         end
26         p=p+t/2;
27     else
28         dout(p)=1;
29         p=p+1;
30     end
31 end
32 y=[3 3];
33 subplot(2,1,1) //plot in both the plots in a single
    window
34 title('input at pin C')
35 plot(cin)
36 plot(y)
37 subplot(2,1,2)

```

```

38 title('out put at pin D')
39 plot(dout)
40 plot(y)

```

---

Scilab code Exa 8.10 fictitious flip flop excitation table

```

1 //Example 8.10
2 clc
3 clear
4 close
5 b=[0 1 0 1 0 1 0 1]; //given truth tble for the
   fictitious flip flop
6 a = [0 0 1 1 0 0 1 1];
7 qn = [0 0 0 0 1 1 1 1];
8 for i=1:8
9   if(a(i)==0 & b(i)==0) then
10     qn1(i) = 0;
11     end;
12   if (a(i)==1 & b(i)== 1) then
13     qn1(i) = 1 ;
14   end;
15   if (a(i)==1 & b(i)== 0) then
16     qn1(i) = bitcmp(qn(i),1) ;
17   end;
18   if (a(i)==0 & b(i)== 1) then
19     qn1(i) = (qn(i)) ;
20   end;
21 end;
22 for i = 1 : 8 // printin the truth table
23     Y(i,1)=qn(i);
24     Y(i,2)=a(i);
25     Y(i,3)=b(i);
26     Y(i,4)=qn1(i);
27     end
28 disp('The given truth table is :');

```

```

29 disp(' Qn      A      B      Qn+1');
30 disp(Y);
31 disp('The transitions are shown below');
32 c1=0;
33 c2=0;
34 c3=0;
35 c4=0;
36 for j=1:8 //checking all possible cases to make a
           transition table
37         if(qn(j)==0) then
38             if(qn1(j) == 0) then
39                 if(c1==0) then
40                     disp('transition from 0 ----> 0');
41                     disp('  A      B ');
42                     c1=1;
43                     end;
44                     disp(Y(j,2:3));
45                 end;
46             end;
47         end;
48 for j=1:8
49         if(qn(j)==0) then
50             if(qn1(j) == 1) then
51                 if(c2==0) then
52                     disp('transition from 0 ----> 1');
53                     disp('  A      B ');
54                     c2=1;
55                     end;
56                     disp(Y(j,2:3));
57                 end;
58             end;
59         end;
60 for j=1:8
61         if(qn(j)==1) then
62             if(qn1(j) == 0) then
63                 if(c3==0) then
64                     disp('transition from 1 ----> 0');
65                     disp('  A      B ');

```

```

66             c3=1;
67             end;
68             disp (Y(j,2:3));
69         end;
70     end;
71 end;
72 for j=1:8
73     if(qn(j)==1) then
74         if(qn1(j) == 1) then
75             if(c4==0) then
76                 disp('transition from 1 ----> 1');
77                 disp('  A      B ');
78                 c4=1;
79                 end;
80                 disp(Y(j,2:3));
81             end;
82         end;
83 end;

```

---

Scilab code Exa 8.12 state transition diagram for given circuit

```

1 //example 8.12
2
3 clc;
4 clear;
5 close;
6 qn=[0,0,1,1];
7 x=[0,1,0,1];
8 for i=1:4 // calculating Y for all possible cases
9     d(i)= bitxor(x(i),qn(i));
10    qn1(i) =d(i);
11    y(i)=bitand(x(i),bitcmp(qn(i),1));
12 end;
13 for i = 1 : 4 // displaying the state table
14     Y(i,1)=qn(i);

```

```

15         Y(i,2)=x(i);
16         Y(i,3)=d(i);
17         Y(i,4)=qn1(i);
18         Y(i,5)= y(i);
19         end
20 disp('The state table is :');
21 disp('   Qn   X   D   Qn+1   Y');
22 disp(Y);

```

---

### Scilab code Exa 8.13 D flip flop to RS flip flop

```

1 //example 8.13
2 clc;
3 clear;
4 disp('For SR flip flop  $Q_{n+1} = S + R'Q_n$  and for D
      flip-flop  $Q_{n+1} = D$  .');
5 disp('Thus with  $D = S + R'Q_n$  we get circuit which
      behaves like SR flip-flop .');

```

---

# Chapter 9

## Registers

Scilab code Exa 9.1 shift register serial input

```
1 //example 9.1
2 clc;
3 clear;
4 close ;
5 //s = input ('Enter the number to be serially
   shifted in to the shift register ');
6 s=0100; // given serial input
7 for i = 4:-1:1
8     se(i) =modulo(s,10);
9     s=s/10;
10    s=round(s);
11 end
12 se(i+4)=0;
13 k=0;
14 for i = 2:6 // making state table
15     clk(k+1) = k;
16     q(i)=se(i-1);
17     if i>1 then
18         r(i) = q(i-1);
19     else
20         r(i) =0;
```



```

21     end;
22     if i>2 then
23         s(i) = r(i-1);
24     else
25         s(i) =0;
26     end;
27     if i>3 then
28         t(i) =s(i-1);
29     else
30         t(i) =0;
31     end;
32     k=k+1;
33 end
34 for i = 1 : 5 // printing the state table
35     Y(i,1)=clk(i);
36     Y(i,2)=se(i);
37     Y(i,3)=q(i);
38     Y(i,4)=r(i);
39     Y(i,5)= s(i);
40     Y(i,6)=t(i);
41     end
42 disp('The state table is :');
43 disp('Clock   Input   Q       R       S       T');
44 disp(Y);

```

---

**Scilab code Exa 9.2** shift register serial input and output graph

```

1 //example 9.2
2 clc
3 clear
4 close
5 t1=100; // clock period
6 s=0100; //given serial input

```

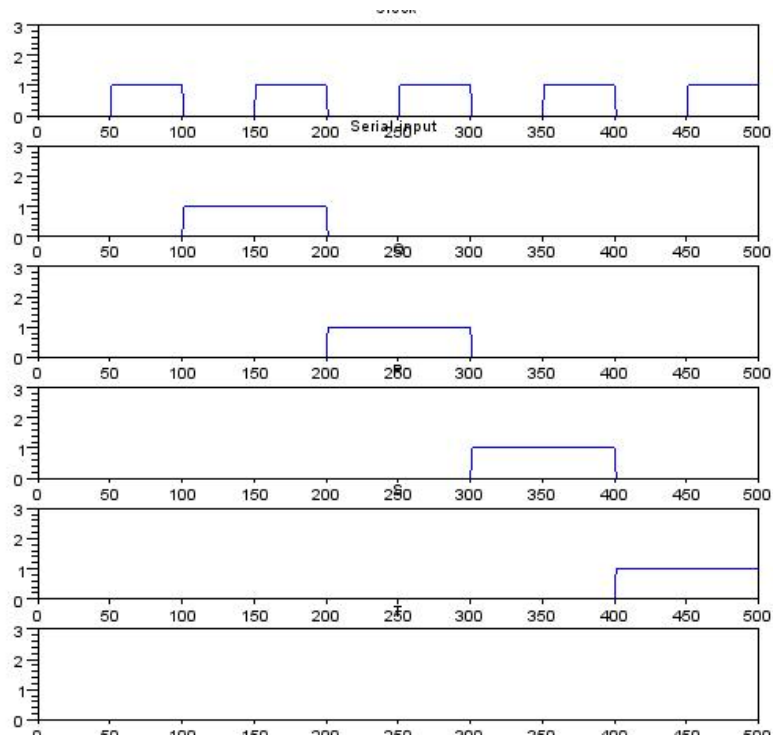


Figure 9.1: shift register serial input and output graph

```

7 for i = 4:-1:1
8     se(i) = modulo(s,10);
9     s=s/10;
10    s=round(s);
11 end
12 se(i+4)=0;
13 k=0;
14 for i = 2:6 //initially making a state table
15     clk(k+1) = k;
16     q(i)=se(i-1);
17     if i>1 then
18         r(i) = q(i-1);
19     else
20         r(i) =0;
21     end;
22     if i>2 then
23         s(i) = r(i-1);
24     else
25         s(i) =0;
26     end;
27     if i>3 then
28         t(i) =s(i-1);
29     else
30         t(i) =0;
31     end;
32     k=k+1;
33 end
34 for m=1:5 // drawing the graph
35     if(se(m)==1) then
36         v= ((m-1).*t1)
37         for u= 1: t1
38             se1(u+v)=1;
39         end
40     else
41         v= ((m-1)*t1)
42         for u= 1: t1
43             se1(u+v)=0;
44         end

```

```

45         end;
46
47         if(q(m)==1) then
48 v= ((m-1).*t1)
49         for u= 1: t1
50             q1(u+v)=1;
51         end
52     else
53         v= ((m-1)*t1)
54         for u= 1: t1
55             q1(u+v)=0;
56         end
57     end;
58     if(r(m)==1) then
59 v= ((m-1).*t1)
60     for u= 1: t1
61         r1(u+v)=1;
62     end
63 else
64     v= ((m-1)*t1)
65     for u= 1: t1
66         r1(u+v)=0;
67     end
68 end;
69     if(s(m)==1) then
70 v= ((m-1).*t1)
71     for u= 1: t1
72         s1(u+v)=1;
73     end
74 else
75     v= ((m-1)*t1)
76     for u= 1: t1
77         s1(u+v)=0;
78     end
79 end;
80     if(t(m)==1) then
81 v= ((m-1).*t1)
82     for u= 1: t1

```

```

83             t11(u+v)=1;
84             end
85         else
86             v= ((m-1)*t1)
87             for u= 1: t1
88                 t11(u+v)=0;
89             end
90         end;
91     end;
92     p=1;
93     while p<t1*5
94         if p==1 | modulo(p,t1) == 1 then
95             for k=1:t1/2
96                 cin(p+k)=0;
97             end
98             p=p+t1/2;
99         else
100             cin(p)=1;
101             p=p+1;
102         end
103     end
104     y=[3 3];
105     subplot(6,1,1) // making subplots to draw all
106                   graphs in a single window
107     title('Clock ')
108     plot(cin)
109     plot(y)
110     subplot(6,1,2)
111     title('Serial input ')
112     plot(se1)
113     plot(y)
114     subplot(6,1,3)
115     title('Q')
116     plot(q1)
117     plot(y)
118     subplot(6,1,4)
119     title('R')
120     plot(r1)

```

```
120 plot(y)
121 subplot(6,1,5)
122 title('S')
123 plot(s1)
124 plot(y)
125 subplot(6,1,6)
126 title('T')
127 plot(t11)
128 plot(y)
```

---

#### Scilab code Exa 9.4 54164 shift register

```
1 //example 9.4
2 clc;
3 clear;
4 //b= input('Enter the number of bits :');
5 //c= input('Enter the clock frequency in Mhz :');
6 b= 8; // given values
7 c=10;
8 t= 1000/c;
9 printf('One clock period takes %d ns\n',t); //
   displaying the results
10 tt=t*b;
11 printf(' Time required by total bits required is %d
   ns',tt);
```

---

#### Scilab code Exa 9.5 54164 shift register

```
1 //example 9.5
2 clc;
3 clear;
4 close;
5 //c= input('Enter the clock frequency in Mhz :');
```

```

6 c=10; //given clock frequency
7 t= 1000/c;
8 printf('The data must be stable for 30 ns\n'); //
    displaying results
9 tc = t-30;
10 printf(' The data may be changing in %d ns',tc);

```

---

#### Scilab code Exa 9.8 741s174

```

1 //exaple 9.8
2 clc;
3 clear;
4 //s=input("Enter the setup time in ns :");
5 //h=input("Enter the hold time in ns :");
6 s=20; //given input values
7 h=5;
8 printf('The data input levels must be held steady
    foor a minimum of %d ns',(s+h)); // displayin
    the results

```

---

#### Scilab code Exa 9.9 7495A

```

1 //example 9.9
2 clc;
3 clear;
4 disp('The mode control line must be high. The data
    lines must be stable for more than 10 ns prior to
    the clock NTs . If the clock is stopped after
    the transition time T, the levels n the input
    data lines may be changed. However , if the clock
    is not stopped , the input data levels must be
    mainted.')
```

---

# Chapter 10

## Counters

Scilab code Exa 10.1 ripple counter clock frequency

```
1 //example 10.1
2 clc;
3 clear;
4 //c= input('Enter the period of the waveform at C in
      micro seconds : ');
5 c=24; // given period of waveform
6 clk= c/8;
7 clkf = 1/(clk*10^-3);
8 printf('The clock period is %f micro seconds \n',clk
      );//displaying the results
9 printf('The clock frequenc must be %f KHz ', clkf);
```

---

Scilab code Exa 10.2 number of flip flops required to construct a counter

```
1 //example 10.2
2 clc;
3 clc
4 c=128; // given counters
```



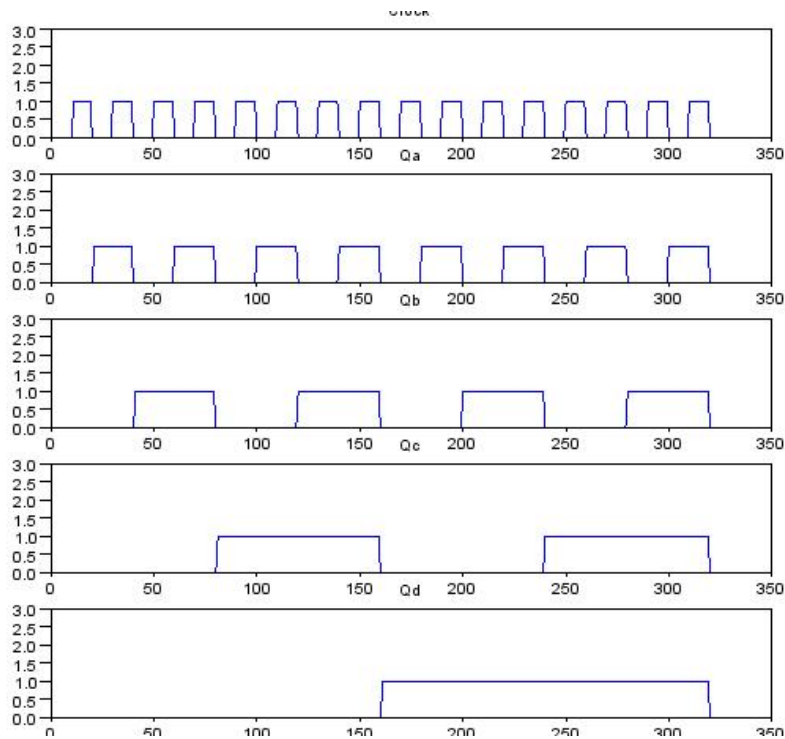


Figure 10.1: Output waveforms for a 7493A connected as a mod 16 counter

```

5 d=32;
6 e=64;
7 fc=log2(c); //making necessar calculations
8 fd=log2(d);
9 printf('A mod-128 conter should have %d flipflops\n'
    , fc);
10 printf(' A mod-32 conter should have %d flipflops\n'
    , fd); //displaying the results
11 fe=log2(e);
12 n=2^fe - 1 ;
13 printf(' The largest decimal no that can be stored
    in a mod-64 counter is %d',n);

```

---

Scilab code Exa 10.3 Output waveforms for a 7493A connected as a mod 16 counter

```
1  clc;
2  clear;
3  close;
4  t=320;
5  p=1;
6  while p<t*1 //making arrays for plotting
7      if p==1 | modulo(p,t)==0 then
8          for k=1:t/2
9              qd(p+k)=0;
10             end
11             p=p+t/2;
12             else
13                 qd(p)=1;
14                 p=p+1;
15             end
16         end
17     t=160;
18     p=1;
19     while p<t*2
20         if p==1 | modulo(p,t)==0 then
21             for k=1:t/2
22                 qc(p+k)=0;
23             end
24             p=p+t/2;
25             else
26                 qc(p)=1;
27                 p=p+1;
28             end
29         end
30     t=80;
31     p=1;
32     while p<t*4
```

```

33     if p==1 | modulo(p,t)==0 then
34         for k=1:t/2
35             qb(p+k)=0;
36         end
37         p=p+t/2;
38     else
39         qb(p)=1;
40         p=p+1;
41     end
42 end
43 t=40;
44 p=1;
45 while p<t*8
46     if p==1 | modulo(p,t)==0 then
47         for k=1:t/2
48             qa(p+k)=0;
49         end
50         p=p+t/2;
51     else
52         qa(p)=1;
53         p=p+1;
54     end
55 end
56 t=20;
57 p=1;
58 while p<t*16
59     if p==1 | modulo(p,t)==0 then
60         for k=1:t/2
61             clk(p+k)=0;
62         end
63         p=p+t/2;
64     else
65         clk(p)=1;
66         p=p+1;
67     end
68 end
69 for i=320:350
70     clk(i)=0;

```

```

71     qa(i)=0;
72     qb(i)=0;
73     qc(i)=0;
74     qd(i)=0;
75 end;
76 y=[3 3]; //ploting the graphs
77 subplot(5,1,1)
78 title('Clock')
79 plot(clk)
80 plot(y)
81 subplot(5,1,2)
82 title('Qa')
83 plot(qa)
84 plot(y)
85 subplot(5,1,3)
86 title('Qb')
87 plot(qb)
88 plot(y)
89 subplot(5,1,4)
90 title('Qc')
91 plot(qc)
92 plot(y)
93 subplot(5,1,5)
94 title('Qd')
95 plot(qd)
96 plot(y)

```

---

Scilab code Exa 10.5 Expression for AND gate connected to the leg of OR gate that

```

1 //example 10.5
2 clc;
3 clear;
4 printf('The correct expression is : (count-up clock)
      ''(Qa)(Qb)(Qc)');

```

---

Scilab code Exa 10.6 Expression for 4 input AND gate connected to the leg of OR gate

```
1 //example 10.6
2
3 clc;
4 clear;
5 printf('The correct logic expression is : (down-up)\'
        \'(Qa)(Qb)(Qc)(enable)\'\'');
```

---

Scilab code Exa 10.7 number of flip flops required to construct a counter

```
1 //example 10.7
2 clc;
3 clear;
4 mod = input("Enter the n value in your desired mod-n
             counter:"); //taking the input
5 m=mod;
6 while 1
7     n= log2(mod); //checking whether the given number
             is a power of 2
8     k=modulo(n,1);
9     if k==0 then
10        printf('The number of flip flops used in mod-%d
                counter are:',m); // if yes the print th
                outpu.
11        printf(' %d',n);
12        return;
13    end
14    mod =mod+1;
15    end
```

---

Scilab code Exa 10.8 what modulus counters can be constructed with given number of

```
1 //example 10.8
2 clc;
3 clear;
4 //ff = input('Enter the no of flip-flops :');
5 ff=4; //given input
6 k=2^ff;
7 if(k==2) then //output display
8     printf('With given flipflop we can only count 2,
9         we can have a modulus 2 counter');
9     else
10    printf('With given number of flip-flops the counter
11        will have a natural count of %d\n',k);
12    printf('We can thus construct any counter that has a
13        modulus between %d and 2',k )
14 end;
```

---

Scilab code Exa 10.9 mod 6 counter

```
1 //example 10.9
2
3 clc;
4 clear;
5 close;
6 c = [0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1
7     0]; //taking the values for a mod -6 counter
8 q = [0 0 1 1 0 0 1 1 0 0 1 1 0 0 1 1 0 0 1 1 0 0 1 1
9     0];
```

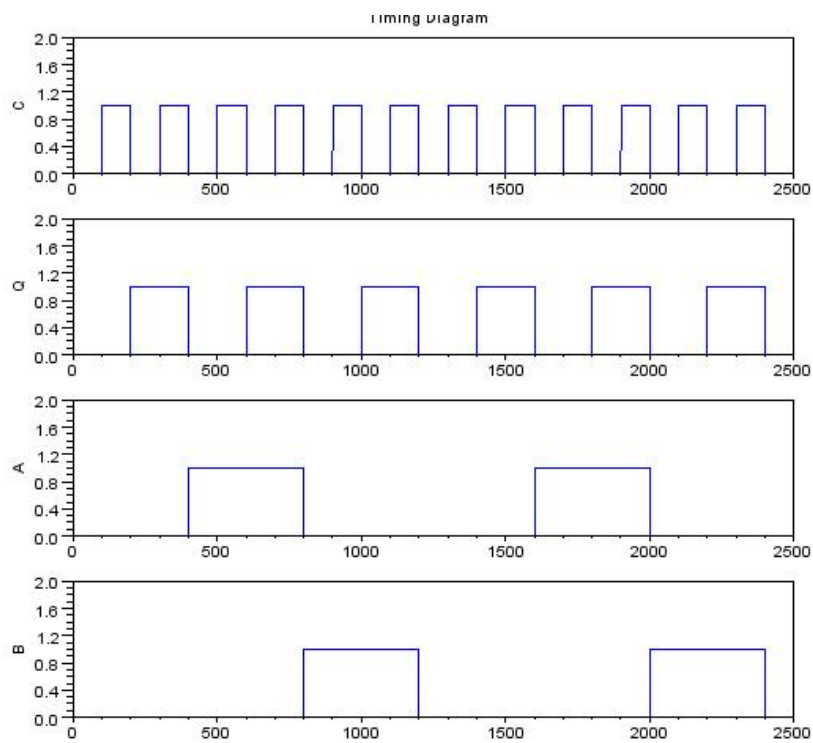


Figure 10.2: mod 6 counter

```

8 a = [0 0 0 0 1 1 1 1 0 0 0 0 0 0 0 0 1 1 1 1 0 0 0 0
      0];
9 b = [0 0 0 0 0 0 0 0 1 1 1 1 0 0 0 0 0 0 0 1 1 1 1
      0];
10 y1=q;
11 y2=a;
12 y3=b;
13 y11p=1;
14 y22p=1;
15 y33p=1;
16 y44p=1;
17 cp=1;
18 yf1p=1;
19 for i=1:25 // making arrays to draw the output
20     if y1(i)==1 then
21         for o=1:100
22             y11(y11p)=1;
23             y11p=y11p+1;
24         end
25     else
26         for o=1:100
27             y11(y11p)=0;
28             y11p=y11p+1;
29         end
30
31 end
32 if y2(i)==1 then
33     for o=1:100
34         y21(y22p)=1;
35         y22p=y22p+1;
36     end
37 else
38     for o=1:100
39         y21(y22p)=0;
40         y22p=y22p+1;
41     end
42
43 end

```



```

44 if y3(i)==1 then
45     for o=1:100
46         y31(y33p)=1;
47         y33p=y33p+1;
48     end
49     else
50         for o=1:100
51             y31(y33p)=0;
52             y33p=y33p+1;
53         end
54
55 end
56 if c(i)==1 then
57     for o=1:100
58         c1(cp)=1;
59         cp=cp+1;
60     end
61     else
62         for o=1:100
63             c1(cp)=0;
64             cp=cp+1;
65         end
66 end
67
68 end
69 z=[2 2];
70 subplot(4,1,1); //ploting the out put
71 title('Timing Diagram');
72 plot(c1);
73 plot(z);
74 ylabel('C');
75 subplot(4,1,2);
76 plot(y11);
77 ylabel('Q');
78 plot(z);
79 subplot(4,1,3);
80 plot(y21);
81 ylabel('A');

```

```
82 plot(z);
83 subplot(4,1,4);
84 plot(z);
85 ylabel('B');
86 plot(y31);
```

---

Scilab code Exa 10.10 Expression for a gate to decode count 8 in a 7492A

```
1 //example 10.10
2 clc;
3 clear;
4 printf('The correct expression is ""8"" = Qd Qc'' Qb
        Qa''');
```

---

Scilab code Exa 10.12 mod 12 counter

```
1 //example 10.12
2 clc
3 clear
4 //pro= ('Enter the value to whic counter should
        progress:');
5 pro =11; // given input
6 q=1;
7 aa=pro;
8 for i=1:4 //converting the given number
        in to binary
9     x=modulo(aa,2);
10    b(q)=x;
11    aa=aa/2;
12    aa=floor(aa);
13    q=q+1;
14 end
15
```

```

16 bi=' '; // then printing the NAND gate inputs
17 for i=1:4
18     if i==1 & b(i)==1 then
19         bi=strcat([bi 'Qa']);
20     elseif i==1 & b(i)== 0 ;
21         bi=strcat([bi 'Qa' ' ']);
22     end
23     if i==2 & b(i)==1 then
24         bi=strcat([bi 'Qb']);
25     elseif i==2 & b(i)== 0 ;
26         bi=strcat([bi 'Qb' ' ']);
27     end
28     if i==3 & b(i)==1 then
29         bi=strcat([bi 'Qc']);
30     elseif i==3 & b(i)== 0 ;
31         bi=strcat([bi 'Qc' ' ']);
32     end
33     if i==4 & b(i)==1 then
34         bi=strcat([bi 'Qd']);
35     elseif i==4 & b(i)== 0 ;
36         bi=strcat([bi 'Qd' ' ']);
37     end
38
39 end
40 disp('The NAND gate inputs must be :');
41 disp(bi)

```

---

#### Scilab code Exa 10.13 4 bit binary counter presettable

```

1 //example 10.13
2 clc;
3 clear
4 //pre=input("Enter the number where the counter is
   preset");
5 pre = 1001; // given preset value

```

```

6 q=1;
7 b=0;
8 f=0;
9 a=pre;
10 while(a>0) //converting to decimal
11     r=modulo(a,10);
12     b(1,q)=r;
13     a=a/10;
14     a=floor(a);
15     q=q+1;
16 end
17 for m=1:q-1
18     c=m-1
19     f = f + b(1,m)*(2^c);
20 end
21 disp("The counter will count down to 15 , Then
        preset back to %d, The resulting state diagram is
        shown below");
22 for k=1:3
23 for i=9:-1:0 // this will print the states
24 printf( '%d  ',i);
25 end;
26 printf( '15  ');
27
28 end;

```

---

check Appendix [AP 8](#) for dependency:

kmap3.sci

check Appendix [AP 2](#) for dependency:

noof.sci

check Appendix [AP 3](#) for dependency:

noof0.sci

### Scilab code Exa 10.14 self correcting modulo 6 counter

```
1 //example 10.14
2 //this program uses the following functions
3 //kmap3.sci
4 //noof.sci and noof0.sci
5 //the above programs should be executed before
   executing these programs
6 clc;
7 n= [ 0 0 0;
8     0 0 1;
9     0 1 0;
10    0 1 1;
11    1 0 0;
12    1 0 1;
13    1 1 0;
14    1 1 1];
15 for i= 1 : 5
16     n1(i,:)= n(i+1,:)
17 end
18 for i=6:8
19     n1(i,:)= [0 0 0]
20 end;
21 p=1;
22 for i= 1:3 //making the state table
23     for j = 1:8
24         if n(j,i)== 0
25             jf(j,p)= n1(j,i);
26             jf(j,p+1)= 2;
27         elseif n(j,i) == 1
28             jf(j,p)=2;
29             jf(j,p+1)=bitcmp(n1(j,i),1);
30         end
31     end
32     p=p+2
33 end;
34 disp('State tabel for mod 6 counter:'); //displaying
   the state table
```

```

35 di= [n n1 jf];
36 disp('   Cn   Bn   An   Cn1   Bn1   An1   Jc
      Kc   Jb   Kb   Ja   Ka');
37 disp(di);
38 disp('Here ''2'' represents a don''t care condition'
      ');
39 disp('These below Karnaugh maps give the design
      equations');
40
41 jc=[0 0 1 0;2 2 2 2] //Krnaugh Maps for the design
      equations
42 JC=kmap3(jc); //calling the 3-variable kmap
43 printf('\n\nJC = %s \n',JC); //displaying the result
44
45 kc=[2 2 2 2;0 1 1 1]
46 KC=kmap3(kc); //calling the 3-variable kmap
47 printf('\n\nKC = %s \n',KC); //displaying the result
48
49 jb=[0 1 2 2;0 0 2 2]
50 JB=kmap3(jb); //calling the 3-variable kmap
51 printf('\n\nJB = %s \n',JB); //displaying the result
52
53 kb=[2 2 1 0;2 2 1 1]
54 KB=kmap3(kb); //calling the 3-variable kmap
55 printf('\n\nKB = %s \n',KB); //displaying the result
56
57 ja=[1 2 2 1;1 2 2 0]
58 JA=kmap3(ja); //calling the 3-variable kmap
59 printf('\n\nJA = %s \n',JA); //displaying the result
60
61 ka=[2 1 1 2;2 1 1 2]
62 KA=kmap3(ka); //calling the 3-variable kmap
63 printf('\n\nKA = %s \n',KA); //displaying the result

```

---

check Appendix [AP 7](#) for dependency:

kmap2.sci

### Scilab code Exa 10.15 sequence generator

```
1 //example 10.15
2 //this program use kmap2.sci
3 //kmap2.sci should be executed before executing this
  program
4 clc;
5 a= [0 0 1 1]
6 b= [0 1 0 1]
7 y= [1 1 0 1]
8 k= [1 1 ; 0 1]
9 bi = kmap2(k); // calling 2-variable kmap
10 disp("  A      B      Y");
11 for i=1:3
12     Y(i,1) = a(i);
13     Y(i,2) = b(i);
14     Y(i,3) = y(i);
15 end
16 disp(Y);
17 disp('The minimised expression from karnaugh map is '
      ');// displaying the result
18 disp(bi);
```

---

# Chapter 11

## Design of Sequential Circuit

check Appendix [AP 6](#) for dependency:

```
kmap3a.sci
```

check Appendix [AP 2](#) for dependency:

```
noof.sci
```

check Appendix [AP 3](#) for dependency:

```
noof0.sci
```

**Scilab code Exa 11.1** synchronous sequential logic circuit

```
1 //example 11.1
2 clc;
3 //this program requires
4 //kmap3a.sci to find the kmap
5 //noof.sci function used inside kmap
6 //noof0.sci function used inside kmap
7 n= [ 0 0 0;
8      0 0 1;
9      0 1 0;
10     0 1 1;
```



```

11     1 0 0;
12     1 0 1;
13     1 1 0;
14     1 1 1];
15     for i= 1:8      //printing the state synthesis
        table
16         an1(i,1)=n(i,3);
17         dn(i,1)=n(i,3);
18         if n(i,1)==1 & n(i,2) ==1 & n(i,3)==0 then
19             z(i,1)=1;
20         else
21             z(i,1)=0;
22         end
23 end;
24 dis=[n an1 dn z];
25 disp('State Synthesis table :');
26 disp('  An      X      Y      An1      Dn      Z');
27 disp(dis);
28 printf('\n\n Design equations :\n');
29 Dn = [ 0 1 1 0;0 1 1 0];
30 Z= [ 0 0 0 0;0 0 0 1];
31 dn1= kmap3a(Dn);           // finding
        the 3 variable kmap of Dn
32 printf('\n      Dn = %s \n\n',dn1);           //displaying
        the minimized expression
33 z1= kmap3a(Z);           //finding
        the 3 variable kmap of Z
34 printf('\n      Z = %s \n\n',z1);           //displaying
        the minimized expression

```

---

check Appendix [AP 4](#) for dependency:

donkmapij.sci

check Appendix [AP 5](#) for dependency:

noof1.sci

## Scilab code Exa 11.2 vending machine

```

1 // example 11.2
2 //this code needs
3 //donkmapij.sci // function to minimize given
  expression using a kmap
4 //noof1.sci
5 //above two should be executed before executing this
  code
6 clc;
7 tt=[0 0 0 0 0 0 0 0 0 0 0; // given state synthesis
  table
8     0 0 0 1 0 0 0 0 0 0 0;
9     0 0 1 0 0 1 0 0 0 0 1;
10    0 0 1 1 1 0 0 0 0 1 0;
11    0 1 0 0 0 1 0 0 0 0 1;
12    0 1 0 1 0 1 0 0 0 0 1;
13    0 1 1 0 1 0 0 0 0 1 0;
14    0 1 1 1 0 0 1 0 0 0 0;
15    1 0 0 0 1 0 0 0 0 1 0;
16    1 0 0 1 1 0 0 0 0 1 0;
17    1 0 1 0 0 0 0 1 0 0 0;
18    1 0 1 1 0 0 0 1 1 0 0];
19 disp('State synthesis table for Vending machine
  problem'); //printing the state synthesis table
20 disp('Present state   input       Next state   Output
  Db   Da');
21 disp('   Bn   An   I       J       Bn1   An1   X       Y
  ');
22 disp(tt);
23 printf('\n\n Design equations :\n');
24
25 printf('\n Design equation for DB\n');
26 db =[ 0 0 2 1 ; 0 0 2 1 ;1 0 2 0 ;0 1 2 0];
27 DB =donkmapij(db); // minimizing the expression using
  4 variable kmap
28 printf('\n      DB = %s \n\n',DB); //displaying result
29

```

```

30 printf('\n Design equation for DA\n');
31 da =[ 0 1 2 0;0 1 2 0 ;0 0 2 0;1 0 2 0];
32 DA =donkmapij(da);// minimizing the expression using
    4 variable kmap
33 printf('\n      DA = %s \n\n',DA);//displaing result
34
35 printf('\n Design equation for X\n');
36 x =[0 0 2 0;0 0 2 0;0 1 2 1; 0 0 2 1];
37 X =donkmapij(x);// minimizing the expression using 4
    variable kmap
38 printf('\n      X = %s \n\n',X);//displaing result
39
40 printf('\n Design equation for Y\n');
41 y=[0 0 2 0;0 0 2 0;0 0 2 1;0 0 2 0];
42 Y =donkmapij(y);// minimizing the expression using 4
    variable kmap
43 printf('\n      Y = %s \n\n',Y);//displaing result

```

---

### Scilab code Exa 11.5 Reducing state transition diagrams

```

1 //example 11.5
2 clc;
3 clear;
4 disp('Original table :'); //displaying original
    table
5 disp('Present State    Next State    Present Output');
6 disp('                X=0    X=1    ');
7 disp('      a            a      b            0    ');
8 disp('      b            c      d            0    ');
9 disp('      c            d      e            1    ');
10 disp('      d            c      b            0    ');
11 disp('      e            b      c            1    ');
12 disp('For states b and d except for next state X=1
    rest are same. NOW b and d would have been
    equivalent if these next states are equivalent.

```

For b next state is d and d, next state is b.  
 Thus bd are equivalent if next states db are  
 equivalent which can always be true. Thus b and d  
 are equivalent and state b is retained.'

```

13 disp('Table after first row elimination :'); //after
    first row elimination
14 disp('Present State   Next State   Present Output');
15 disp('           X=0   X=1   ');
16 disp('      a       a     b       0   ');
17 disp('      b       c     b       0   ');
18 disp('      c       b     e       1   ');
19 disp('      e       b     c       1   ');
20 disp('Now repeating the same above step for c and e
    . Retaining c and replacing arll c''s with e we
    get the below table ');
21 disp('Table after second row elimination :');//after
    second row elimination
22 disp('Present State   Next State   Present Output');
23 disp('           X=0   X=1   ');
24 disp('      a       a     b       0   ');
25 disp('      b       c     b       0   ');
26 disp('      c       b     c       1   ');
27
28 disp('Implication table method'); // by implication
    method
29 printf('d:d\nc:d(ce)\nb:d(Ce)(bd)\na:(ce)(bd)a\nP=(
    ce)(bd)(a)');

```

---

### Scilab code Exa 11.6 asynchronous sequential circuit

```

1 //example 11.6
2 clc;
3 clear;
4 disp('To analyse the circuit we consider  $x = X(t-)$ 
    where is the cummulative propagatin delay from

```

```

        input side up to X. For all possible
        combinations of xAB we get X and Y following the
        logic relation as shown in the circuit and
        prepare the following Karnaugh map');
5  disp('Karnaugh map'); // displaying the kmap
6  disp('  AB')
7  disp('x   00      01      11      10');
8  disp('0   0''/0   0''/0   1/0   0''/0');
9  disp('1   0/0     0/1     1''/1   1''/0');
10 disp('State where X = x are stable and primed.
        Outputs corresponding to each state and input
        combination are shown beside. ');

```

---

**Scilab code Exa 11.7** asynchronous sequential circuit problem in operation

```

1 //example 11.7
2 clc;
3 clear;
4 disp('Given karnaugh map '); //given kmap
5 disp('   00      01      11      10');
6 disp('00  11      00''      11      00''');
7 disp('01  01''      11      11      01''');
8 disp('11  10      11      11''      10');
9 disp('10  10''      10''      11      11');
10 disp('Yes, the circuit may face problem in its
        operation. When the circuit is at stable state
        xyAB = 1111 and input AB changes from 11 --> 10
        the circuit oscillates between xyAB = 1110 and xy
        AB = 1010. Also there can be a critical race
        problem if at stable state xyAB = 0001, input AB
        chnge from 01 to 00. The circuit may settle at
        xyAB = 0100 or xyAB = 1000 depending on which of
        x and y changes first at the feedback path. Non-
        critical race situatuion occurs if at stable
        state xyAB = 0010 the input AB change from 10 to

```

```
00. ');
```

---

check Appendix [AP 1](#) for dependency:

```
kmap3abx.sci
```

check Appendix [AP 2](#) for dependency:

```
noof.sci
```

check Appendix [AP 3](#) for dependency:

```
noof0.sci
```

#### Scilab code Exa 11.8 asynchronous sequential circuit

```
1 //example 11.8
2 // this program requires
3 //kmp3abx.sci
4 //noof.sci
5 //noof0.sci
6 //above three functions are first execute before
   executing this program
7 clc;
8 disp('State table through karnaugh map'); // state
   table through kmap
9 disp('    00    01    11    10');
10 disp('a    a''    a''    b    b');
11 disp('b    a    b''    b''    b''');
12 disp('If we represent current state a as x = 0 and b
   as x =1 the noutput X can be expressed as ');
13 j=[0 0  1 1 ; 0 1 1 1];
14 J= kmap3abx(j); // finding the minimized expression
   using 3-variable kmap
15 disp('The minimised expression J');
16 disp(J); // displaying the minimized expression
```

---

# Chapter 12

## D to A Conversion and A to D conversion

Scilab code Exa 12.1 binary equivalent weight of each bit in a 4bit system

```
1 //chapter 12
2 //Example 12.1
3 //Q.Find the binary equivalent of each bit in a 4-
  bit system
4 //solution:
5 clc;
6 clear;
7 LSB = 1/(2^4-1); // calculating binary weights
8 LSB2 = 2*1/(2^4-1);
9 LSB3 = 4*1/(2^4-1);
10 MSB = 8*1/(2^4-1);
11 disp("Binary weight of each bit in a 4-bit system");
12 disp("LSB ="); // displaing the result
13 disp(LSB);
14 disp("LSB2 =");
15 disp(LSB2)
16 disp("LSB3 =");
17 disp(LSB3)
18 disp("MSB =");
```

19 disp(MSB) ;

---

### Scilab code Exa 12.2 5 bit resistive divider

```
1 //Example 12.2
2 clc;
3 clear;
4 LSB = 1/(2^5-1); // calculating weights
5 LSB2 = 2*1/(2^5-1);
6 LSB3 = 4*1/(2^5-1);
7 change_LSB = 10 *LSB;
8 change_LSB2 = 10 *LSB2;
9 change_LSB3 = 10*LSB3;
10 op_vol= (10*2^0 + 0*2^1 + 10*2^2 +0*2^3 +10*2^4)
    /(2^5-1); // calculating output voltage
11 disp("(a) LSB =");
12 disp(LSB);
13 disp("(b) Second LSB =");
14 disp(LSB2)
15 disp(" Third LSB3 =");
16 disp(LSB3)
17 disp('(c) change in output voltage caused by ');
18 disp("change in LSB =");
19 disp(change_LSB);
20 disp("change in second LSB =");
21 disp(change_LSB2)
22 disp("change in third LSB =");
23 disp(change_LSB3)
24 disp("(d) output voltage for a digital input of 10101
    =");
25 disp(op_vol) ;
```

---

### Scilab code Exa 12.3 5 bit ladder



```

1 //Example 12.3
2 clc;
3 clear;
4 for i=1:5
5     op_v(1,i)= 10/2^i; // calculating output voltages
                        corresponding to each bit
6 end
7 disp("output voltages corresponding to each bit are
      ") //displaying result
8 disp(op_v);

```

---

Scilab code Exa 12.4 5 bit ladder

```

1 //Example 12.4
2 clc
3 clear
4 V_A=0
5 //a=input("Enter the binary digit(5 bits) :");
6 a=11010
7 for i=1:5
8     r=modulo(a,10);
9     b(1,i)=r;
10    a=a/10;
11    a=floor(a);
12 end
13 for j=1:5
14 V_A = V_A + 10*b(1,j)*2^(j-1);
15 end;
16 V_A=V_A/2^5;
17 disp("The output voltage in volts is ");
18 disp(V_A); // displaying the value

```

---

Scilab code Exa 12.5 5 bit ladder

```

1 //Example 12.5
2 clc;
3 clear;
4 ful_scale_voltage = 0 ;
5 for i=1:5
6     op_v(1,i)= 10/2^i;
7     ful_scale_voltage = ful_scale_voltage + op_v(1,i
            ); // calculating the full scale voltage
8 end
9 disp("full scale output voltage in volts is =");
10 disp(ful_scale_voltage)

```

---

#### Scilab code Exa 12.6 5 bit ladder

```

1 //Example 12.6
2 clc;
3 clear;
4 I=10/(3*10^3);
5 printf("Current each input digital voltage must be
        capable of supplying is =%f mA\n",I*1000)
6 ful_scale_voltage = 0 ;
7 for i=1:5
8     op_v(1,i)= 10/2^i;
9     ful_scale_voltage = ful_scale_voltage + op_v(1,i
            );
10 end
11 V_A=ful_scale_voltage * (2*1000)/(1000+(2*1000));
12 printf("\n Output voltage Va = %f V",V_A);

```

---

#### Scilab code Exa 12.8 DAC0808

```

1 //Example 12.8
2 clc;

```

```

3 clear;
4 V0= 10*((1/2)+(1/4)+(1/32)+(1/128)); // calculating
    voltage when 1,2,5,7 re high
5 printf("Output voltage = %f V",V0)//displayin result

```

---

Scilab code Exa 12.9 resolution of 9 bit D to A

```

1 //Example 12.9
2 clc;
3 clear
4 resol = 1/512 * 100 ; // calculating resol
5 vol_resol = 1/512 *5 *1000;
6 printf("Resolution in percentage = %f",resol);
7
8 printf("\n\n Voltage resolution = %f mV",vol_resol);

```

---

Scilab code Exa 12.10 resolution

```

1 //Example 12.10
2 clc;
3 clear;
4 disp("The LSB of an 11-bit system has a resolution
    of 1/2048");
5 re =ceil(10000/2048); // calculating the resolution
6 printf("\n\n 1/2048 x 10 = %d mV",re);

```

---

Scilab code Exa 12.11 counter type A to D converter

```

1 //Example 12.11
2 clc;

```

```

3 clear;
4 //bit = input("Specify the converter bit length :")
5 //fre = input("specify the clock frequency in kHz
   :")
6 bit =8;
7 fre = 500
8 max_conv_time = 2^bit * (1/(fre*1000));
9 avg_conv_time = 0.5 *max_conv_time;
10 max_conv_rate = 1/max_conv_time;
11 disp("Maximum Conversion Time = ");
12 disp(max_conv_time);
13 disp("Average Conversion Time =");
14 disp(avg_conv_time);
15 disp("Maximum Conversion Rate =");
16 disp(max_conv_rate);

```

---

**Scilab code Exa 12.13** 10 bit A to D converter

```

1 //Example 12.13
2 clc
3 clear
4 quat_err = 1/1024 *100 ;
5 disp("If the analog portion to be constructed ti an
   accuracy of 0.1")
6 printf("\nThe overall accuracy is in percentage = %f
   ",0.1 + quat_err)

```

---

# Chapter 13

## Memory

Scilab code Exa 13.2 structure of binary address

```
1 //example 13.2
2 clc;
3 clear;
4 close;
5 //cp = input('enter the capacity of the memory
    system in bits :');
6 cp=1024; // given capacity
7 n= log2(cp);
8 printf('The no of bits in the address word are : %d\
    n',n);
9 printf('The number of required rows are : %d\n',2^(n
    /2));
10 printf('The number of required columns are : %d',2^(
    n/2));
```

---

Scilab code Exa 13.3 decimal and hexadecimal address for the given binary address

```
1 //example 13.3
```

```
2 clc
3 clear
4 //bin(1,1) = input('Enter the first half string of
    binary number :');
5 //bin(1,2) = input('Enter the second half string of
    binary number :');
6 bin=['10110' '01101']; // given binry address
7 dec=bin2dec(bin); // finding decimal equivlent
8 hex=dec2hex(dec); //findin hexadecimal equivalent
9 disp('The decimal address is :');
10 disp(dec);
11 disp('The hexadecimal address is :');
12 disp(hex);
```

---

# Chapter 14

## Digital Integrated circuits

Scilab code Exa 14.1 diode forward or reverse

```
1 //example 14.1
2 clc;
3 clear;
4 close;
5 vdc = input('Enter the value of DC voltage Vdc in
   volts :');
6 r = input('Enter the value of resistace in K ohms :'
   );
7 v = input(' Enter the value of voltage across diode
   in volts :');
8
9 i = (vdc-v)/r ;
10 format('v',4);
11 if(i>0) // checking whether the diode is forward or
   reverse biased by checking current
12     disp('The diode is in forward bias');
13     disp('The diode current in mA is :');
14     disp(i);
15 else
16     disp('The diode is in Reverse bias');
17     disp('The diode current in mA is : 0.0');
```

```
18 end;
```

---

#### Scilab code Exa 14.2 Diode current

```
1 //example 14.2
2 clc;
3 clear;
4 close;
5 vdc = input('Enter the value of DC voltage Vdc in
             volts :');
6 r = input('Enter the value of resistace in K ohms :'
           );
7 //v = input(' Enter the value of voltage across
             diode in volts:');
8 v= 1.6;
9 i = (vdc-v)/r ; /// calculating the current
10     disp('The diode current in mA is :');
11     disp(i);
```

---

#### Scilab code Exa 14.3 current in the given circuit

```
1 //example 14.3
2 clc;
3 clear;
4 close;
5 v1 = input('Enter the value of V1 in volts :'); //
      taking the inpt voltage
6 disp('CASE - a');// case a
7 if (v1==0) then
8     disp('V2 = 5 V');
9     disp('I = 0 mA');
10 else
11     disp('V2 = 0 V')
```



```

12         disp('I = 5 mA ')
13     end
14     disp('CASE - b'); // case b
15     if (v1==0) then
16         disp('V2 = 5 V')
17         disp('I = 5 mA' )
18     else
19         disp('V2 = 0 V');
20         disp('I = 0 mA');
21
22     end

```

---

#### Scilab code Exa 14.4 n channel MOSFET inverter

```

1 //example 14.4
2 clc;
3 clear;
4 v1 = input('Enter the value of V1 in volts :'); //
      part a : v1 =0 ; part b : v1 =5v
5 if (v1==0) then // checking for V1
6     disp('V2 = 5 V');
7     disp('I = 0 mA');
8 else
9     disp('V2 = 0 V');
10    disp('I = 0.5 mA ');
11 end

```

---

# Chapter 15

## Applications

Scilab code Exa 15.1 Timing of a six digit display

```
1 //example 15.1
2 //timing for a six digit display
3 clc;
4 clear;
5 //f=input('Enter the repetition rate in Hz :');
6 //d= input('Enter the length of display :');
7 f=125; // given inputs
8 d=6;
9 format('v',5); //changing the precision of the
   calculation
10 k=1000/f;
11 l=1000/(f*d);//making necessary calculations
12 m=k-l;
13 printf('All digits must be serviced once every %f
   milliseconds \n',k);
14 printf('Each digit will be ON for : %f milliseconds\
   n',l) ;
15 printf('and OFF for : %f milliseconds ',m); //
   displaying results
```

---

#### Scilab code Exa 15.4 Basic frequency counter

```
1 //example 15.4
2 clc;
3 clear;
4 //f=input('Enter the input square wave signal
    frequency in kHz:');
5 //t=input('Enter the gate enable time in seconds ');
6 //first part :
7 f=7.50;
8 t=0.1;
9 format('v',18);
10 m=t*f*1000; //making necessary calculations
11 printf('For t = %f seconds\n',t); // displaying
    results
12 printf('The counter will count up to : %f\n',m);
13 //part2
14 t=1;
15 printf('\n\nFor t = %f seconds\n',t);
16 m=t*f*1000;
17 printf('The counter will count up to : %f\n',m);
18 //part3
19 t=10;
20 m=t*f*1000;
21 printf('\n\nFor t = %f seconds\n',t);
22 printf('The counter will count up to : %f\n',m);
```

---

#### Scilab code Exa 15.5 4 decimal digit frequency counter

```
1 //example 15.5
2 clc;
```

```
3 disp('Assuming the counter began at 0000, the
    display would read 200 at the end of the first
    measurment period.It will read 400,then 600 and
    so on at the end of succeeding periods. This is
    because the counter capacity is exceeded each
    time, and it simply recycles through 0000.');
```

---

**Scilab code Exa 15.6** instrument to measure time period

```
1 //example 15.6
2 clc;
3 clear;
4 //s= input('Enter the clk frequency in kHz : ');
5 //f=input('Enter the frequency of the unknown input
    in Hz :');
6 s=100;// taking the inputs
7 f=200;
8 g=1000000/f; // making neccessary calculations
9 c=g*s/1000;
10 c=round(c);
11 p=c*1000/s;
12 disp('Assuming that the conter and the display are
    initially at 00000');//displaying results
13 printf('Enable gate time in micro secnds will be :
    %d\n',g);
14 printf('During the gate time the counter will be
    advcd by (number of counts ) %d \n',c);
15 printf('The time period of the unknown input in
    micro-seconds is : %d',p);
```

---

**Scilab code Exa 15.9** ADC0804

```
1 //example 15.9
```

```

2  clc ;
3  clear;
4  //part (a)
5  //an=input('Enter the analog input in volts :');
6  format('v',12); // changing the precision of
    calculation
7  an=2.5;
8  k=an*1000/19.53;
9  k= round(k);
10 m=dec2bin(k); // converting from decimal to binary
11 printf('The digital output is :%s\n',m);
12 //part(b)
13 //dg=input('Enter the digital output as a string:');
14 dg='00100010';
15 f=bin2dec(dg); // converting binary to decimal
16 y=f*19.53*10-3;
17 printf(' The analog input in volts is :%f',y);

```

---

#### Scilab code Exa 15.10 ADC3511

```

1  //example 15.10
2  clc;
3  clear;
4  //v = input('Enter the reference voltage in volts
    :');
5  //an = input('Enter the analog input voltage in
    volts :');
6  v=2; // taking given input
7  an=1.25;
8  count = 2000*an/v ;
9  count = round(count);
10 printf('The ccount held in the counter for given
    analog input will be : %f',count);
11 d= an/v;
12 printf(' \n The duty cycle is : %f',d);

```

---

**Scilab code Exa 15.11 ADC3511**

```
1 // example 15.11
2 clc;
3 clear;
4 disp('The full scale count for ADC3511 is 1999 and
      for the ADC3711 is 3999. So, the largest value
      possible for the MSD in either case is 3 = 0011.
      clearly the MSB is not needed for th magnitue of
      the MSD. It is thus convenient to specif
      positive number when this bit is a 0 and a
      negative number when this bit is a 1 .');
```

---

**Scilab code Exa 15.12 ADD3501**

```
1 //exmple 15.12
2 clc ;
3 clear;
4 disp('These two components establish the internal
      oscillator frequency used as the clock frequency
      in the cnverter according to the relationship fi
      =0.6/RC. In this case fi=320 kHz.');
```

---

# Chapter 16

## A Simple Computer Design

Scilab code Exa 16.1 size of PC IR ACC MAR MDR

```
1 //example 16.1
2 clc
3 clear
4 //len = input('Enter the length of each memory
      location in bits :');
5 //op = input('Enter the length of Opcode:');
6 len =16;
7 op=4;
8 nop= 2^4; // calculating
9 nab = len-op;
10 memloc = 2^nab;
11 memsize = memloc*16;
12 mem=memsize/1024;
13 printf('(a)Maximum Number of Opcodes = %d \n',nop);
      // displaying
14 printf(' (b)Size of memory in Kilo bits = %d \n',mem
      );
15 printf(' (c)Size of PC and MAR = %d \n',nab );
16 printf('      Size of IR = %d\n',op);
17 printf('      Size of ACC and MDR = %d \n',len);
```

---

Scilab code Exa 16.6 Number of clock cycles needed to execute a program

```
1 //example 16.6
2 lda= input('Enter the number of LDA instructions :')
   ; // accepting the input from the user
3 add= input('Enter the number of ADD instructions :')
   ;
4 sub= input('Enter the number of SUB instructions :')
   ;
5 sta= input('Enter the number of STA instructions :')
   ;
6 shl= input('Enter the number of SHL instructions :')
   ;
7 hlt= input('Enter the number of HLT instructions :')
   ;
8 k= lda+add+sub+sta;
9 l=shl+HLT;
10 c= k*5 + l*4; // calculating the total no.of clock
   cycles required
11 printf('Total clock cycles required to execute are =
   %d ',c); //displaying result.
```

---



# Appendix

Scilab code AP 1 3-variable kmap(abx)

```
1 //3-VARIABLE KMAP
2 //uses noof.sci and noof0.sci
3 //above two functions should be executed before
  executing this function .
4 function bi = kmap3abx(k)
5     n=4;
6     m=2
7
8 //k=[0 0 0 1;
9 // 0 1 1 1];
10 k(:, :, 2)=zeros(m,n);
11     var=['x' 'A' 'B'];
12     //var=['w' 'x' 'y' 'z'];
13     p1=['x' '' 'x'];
14     p2=['A' 'B' '' ; 'A' 'B'; 'AB'; 'AB' ''];
15     cmn4=4;
16     cmn2=2;
17     temp=1;
18     // printf('The minimal epression of the given
      Kmap ');
19     disp(k(:, :, 1));
20     //disp(" is :");
21     //printf('f');
22     // printf("=");
23
24 bi = ' ';
```

```

25 //8 cells
26 for i=1:m
27     for j=1:n
28         if(k(i,j)~=1 & k(i,j)~=2)
29             temp=0;
30             break;
31         end
32     end
33 end
34 if(temp==1)
35     bi = strcat([bi "1"]);
36     return;
37 end
38 //4 cells
39 z1=ones(1,4);
40 z2=ones(4,1);
41 z3=ones(2,2);
42 temp1=['0' '1'];
43 temp2=['00';'01';'11';'10'];
44 for t=1:m
45     z=k(t, :, 1);
46     no=noof(k(t, :, 2));
47     if(noof0(z)==0 & no<cmn4 & noof(z)>0)
48         k(t, :, 2)=z1;
49         a=strsplit(temp1(1,t));
50         for in=1:max(size(a))
51             if(a(in)=='0')
52                 bi = strcat([bi var(in) ' ']);
53             end
54             if(a(in)=='1')
55                 bi = strcat([bi var(in)]);
56             end
57         end
58         bi = strcat([bi " + "]);
59     end
60 end
61 for i=1:m-1

```

```

62         for j=1:n
63             t1=i+1;
64             if(j==n)
65                 t2=1;
66             else
67                 t2=j+1;
68             end
69             z4=[k(i,j,1) k(i,t2,1);k(t1,j,1) k(t1,t2
70                 ,1)];
71             z5=[k(i,j,2) k(i,t2,2);k(t1,j,2) k(t1,t2
72                 ,2)];
73             no=noof(z5);
74             if(noof0(z4)==0 & no<cmn4 & noof(z4)>0)
75                 k(i,j,2)=1;
76                 k(i,t2,2)=1;
77                 k(t1,j,2)=1;
78                 k(t1,t2,2)=1;
79                 a=strsplit(temp2(j,1));
80                 b=strsplit(temp2(t2,1));
81                 c=strcmp(a,b);
82                 for in=1:max(size(c))
83                     if(c(in)==0 & a(in)=='0')
84                         bi = strcat([bi var(1+in) ' '
85                                     ' ' ]);
86                     end
87                     if(c(in)==0 & a(in)=='1')
88                         bi = strcat([bi var(1+in)])
89                             ;
90                     end
91                 end
92             end
93         //2 cells
94         z6=[1 1];
95         z7=z6';

```

```

96     for i=1:m
97         for j=1:n
98             t1=i+1;
99             if(j==n)
100                 t2=1;
101             else
102                 t2=j+1;
103             end
104             z8=[k(i,j,1) k(i,t2,1)];
105             z9=[k(i,j,2) k(i,t2,2)];
106             no1=noof(z9);
107             if(noof0(z8)==0 & no1<cmn2 & noof(z8)>0)
108                 k(i,j,2)=1;
109                 k(i,t2,2)=1;
110                 bi = strcat([bi p1(1,i)]);
111                 a=strsplit(temp2(j,1));
112                 b=strsplit(temp2(t2,1));
113                 c=strcmp(a,b);
114                 for in=1:max(size(c))
115                     if(c(in)==0 & a(in)=='0')
116                         bi = strcat([bi var(1+in) ''
117                                     '']);
118                         bi = strcat([bi '' + '']);
119                     end
120                     if(c(in)==0 & a(in)=='1')
121                         bi = strcat([bi var(1+in)])
122                             ;
123                         bi = strcat([bi '' + '']);
124                     end
125                 end
126             end
127         for i=1:m-1
128             for j=1:n
129                 t1=i+1;
130                 if(j==n)
131                     t2=1;

```

```

132         else
133             t2=j+1;
134         end
135         z10=[k(i,j,1);k(t1,j,1)];
136         z11=[k(i,j,2);k(t1,j,2)];
137         no2=noof(z11);
138         if(noof0(z10)==0 & no2<cmn2 & noof(z10)
           >0)
139             k(i,j,2)=1;
140             k(t1,j,2)=1;
141             bi = strcat([bi p2(j,1)]);
142             bi = strcat([bi " + "]);
143         end
144     end
145 end
146 //single cell
147 for i=1:m
148     for j=1:n
149         if(k(i,j,2)==0 & k(i,j,1)==1)
150             bi = strcat([bi p1(1,i)]);
151             bi = strcat([bi p2(j,1)]);
152             bi = strcat([bi " + "]);
153         end
154     end
155 end
156 bi = strcat([bi " 0 "]);
157 //disp(" ")
158
159 endfunction

```

---

Scilab code AP 2 returns number of 1s in a matrix

```

1 function res=noof(a) // this function returns the
   no of 1's in the given matrix
2     res=0;
3     for i=1:max(size(a(:,1)))
4         for j=1:max(size(a(1,:)))
5             if(a(i,j)==1)

```

```

6             res=res+1;
7         end
8     end
9 end
10 endfunction

```

---

Scilab code AP 3 returns number of 0s in a matrix

```

1 function res=noof0(a) // to find the no of zeros in
   given matrix
2     res=0;
3     for i=1:max(size(a(:,1)))
4         for j=1:max(size(a(1,:)))
5             if(a(i,j)==0)
6                 res=res+1;
7             end
8         end
9     end
10 endfunction

```

---

Scilab code AP 4 4-variable kmap with don't cares

```

1 //4-Variable KMAP
2 //returns a string of the minimized expression
3 //requires noof1.sci
4 //noof1.sci should be executed before executing this
   function
5 function bi = donkmapij(k)
6     n=4;
7     k(:,:,2)=zeros(n,n);
8     var=['I' 'J' 'Bn' 'An'];
9     p1=['I''J'' ' 'I''J' 'IJ' 'IJ'''];
10    p2=['Bn''An'' ' 'Bn''An'; 'BnAn'; 'BnAn'''];
11    cmn4=4;
12    cmn2=2;
13    temp=1;
14    bi= ' ';
15    disp(k(:,:,1));

```

```

16     for i=1:n
17         for j=1:n
18             if(k(i,j)~=1 | k(i,j)~=2)
19                 temp=0;
20                 break;
21             end
22         end
23     end
24     if(temp==1)
25         printf("1");
26         abort;
27     end
28     //checking the 8 cells cases
29     z1=ones(2,4);
30     z2=ones(4,2);
31     temp1=['00' '01' '11' '10'];
32     temp2=temp1';
33     for i=1:n
34         if(i==4)
35             t=1;
36         else
37             t=i+1;
38         end
39         z=[k(i,:,1);k(t,:,1)];
40         if(noof1(z,0)==0 & noof1(z,1)>1)
41             k(i,:,2)=[1 1 1 1];
42             k(t,:,2)=[1 1 1 1];
43             a=strsplit(temp2(i,1));
44             b=strsplit(temp2(t,1));
45             c=strcmp(a,b);
46             for in=1:max(size(c))
47                 if(c(in)==0 & a(in)=='0')
48                     bi = strcat([bi var(in) ' ']);
49                     bi = strcat([bi " + "]);
50                     break;
51                 else
52                     if(c(in)==0 & a(in)=='1')

```

```

53             bi = strcat([bi var(in)])
54             ;
55             bi = strcat([bi " + "]);
56             break;
57         end
58     end
59 end
60 end
61 for j=1:n
62     if(j==4)
63         t=1;
64     else
65         t=j+1;
66     end
67     z=[k(:,j,1) k(:,t,1)];
68     if(noof1(z,0)==0 & noof1(z,1)>0)
69         k(:,j,2)=[1;1;1;1];
70         k(:,t,2)=[1;1;1;1];
71         a=strsplit(temp1(1,j));
72         b=strsplit(temp1(1,t));
73         c=strcmp(a,b);
74         for in=1:max(size(c))
75             if(c(in)==0 & a(in)=='0')
76                 bi = strcat([bi var(2+in) ' ']);
77                 ;
78                 bi = strcat([bi " + "]);
79                 break;
80             else
81                 if(c(in)==0 & a(in)=='1')
82                     bi = strcat([bi var(2+in)])
83                     ;
84                     bi = strcat([bi " + "]);
85                     break;
86                 end
87             end
88         end
89     end
90 end

```



```

88     end
89     //checking the 4 cells cases
90     z1=ones(1,4);
91     z2=ones(4,1);
92     z3=ones(2,2);
93     temp1=['00' '01' '11' '10'];
94     temp2=temp1';
95     for t=1:n
96         z=k(t, :, 1);
97         no=noof1(k(t, :, 2), 1);
98         if(noof1(z,0)==0 & no<cmn4 & noof1(z,1)
99             >0)
100             k(t, :, 2)=z1;
101             a=strsplit(temp1(1,t));
102             for in=1:max(size(a))
103                 if(a(in)=='0')
104                     bi = strcat([bi var(in) ' ']);
105                 end
106                 if(a(in)=='1')
107                     bi = strcat([bi var(in)]);
108                 end
109                 bi = strcat([bi " + "]);
110             end
111         end
112     for t=1:n
113         z=k(:, t, 1);
114         no=noof1(k(:, t, 2), 1);
115         if(noof1(z,0)==0 & no<cmn4 & noof1(z,1)>0)
116             k(:, t, 2)=z2;
117             a=strsplit(temp2(t,1));
118             for in=1:max(size(a))
119                 if(a(in)=='0')
120                     bi = strcat([bi var(2+in) ' ']);
121                 end
122                 if(a(in)=='1')

```

```

123             bi = strcat([bi var(2+in)]);
124             end
125         end
126         bi = strcat([bi " + "]);
127     end
128 end
129 for i=1:n
130     for j=1:n
131         if(i==n)
132             t1=1;
133         else
134             t1=i+1;
135         end
136         if(j==n)
137             t2=1;
138         else
139             t2=j+1;
140         end
141         z4=[k(i,j,1) k(i,t2,1);k(t1,j,1) k(t1,t2
142             ,1)];
143         z5=[k(i,j,2) k(i,t2,2);k(t1,j,2) k(t1,t2
144             ,2)];
145         no=noof1(z5,1);
146         if(noof1(z4,0)==0 & no<cmn4 & noof1(z4
147             ,1)>0)
148             k(i,j,2)=1;
149             k(i,t2,2)=1;
150             k(t1,j,2)=1;
151             k(t1,t2,2)=1;
152             a=strsplit(temp2(i,1));
153             b=strsplit(temp2(t1,1));
154             c=strcmp(a,b);
155             for in=1:max(size(c))
156                 if(c(in)==0 & a(in)=='0')
157                     bi = strcat([bi ,var(in) ' '
158                         ']);
159                 end
160                 if(c(in)==0 & a(in)=='1')

```

```

157             bi = strcat([bi var(in)]);
158         end
159     end
160     a=strsplit(temp1(1,j));
161     b=strsplit(temp1(1,t2));
162     c=strcmp(a,b);
163     for in=1:max(size(c))
164         if(c(in)==0 & a(in)=='0')
165             bi = strcat([bi ,var(2+in) '
166                 ''']);
167             end
168             if(c(in)==0 & a(in)=='1')
169                 bi = strcat([bi var(2+in)])
170             ;
171         end
172     end
173     end
174 end
175 //2 cells
176 z6=[1 1];
177 z7=z6';
178 for i=1:n
179     for j=1:n
180         if(i==n)
181             t1=1;
182         else
183             t1=i+1;
184         end
185     if(j==n)
186         t2=1;
187     else
188         t2=j+1;
189     end
190     z8=[k(i,j,1) k(i,t2,1)];
191     z9=[k(i,j,2) k(i,t2,2)];
192     no1=noof1(z9,1);

```

```

193         if(noof1(z8,0)==0 & no1<cmn2 & noof1(z8
           ,1)>0)
194             k(i,j,2)=1;
195             k(i,t2,2)=1;
196             a=strsplit(temp1(1,j));
197             b=strsplit(temp1(1,t2));
198             c=strcmp(a,b);
199             for in=1:max(size(c))
200                 if(c(in)==0 & a(in)=='0')
201                     bi = strcat([bi p1(1,i)]);
202                     bi = strcat([bi ,var(2+in) '
                        ' ' ']);
203                     bi = strcat([bi " + "]);
204                 end
205                 if(c(in)==0 & a(in)=='1')
206                     bi = strcat([bi p1(1,i)]);
207                     bi = strcat([bi var(2+in)])
                        ;
208                     bi = strcat([bi " + "]);
209                 end
210             end
211         end
212     end
213 end
214 for i=1:n
215     for j=1:n
216         if(i==n)
217             t1=1;
218         else
219             t1=i+1;
220         end
221         if(j==n)
222             t2=1;
223         else
224             t2=j+1;
225         end
226         z10=[k(i,j,1);k(t1,j,1)];
227         z11=[k(i,j,2);k(t1,j,2)];

```

```

228         no2=noof1(z11,1);
229         if(noof1(z10,0)==0 & no2<cmn2 & noof1(
           z10,1)>0)
230             k(i,j,2)=1;
231             k(t1,j,2)=1;
232             a=strsplit(temp2(i,1));
233             b=strsplit(temp2(t1,1));
234             c=strcmp(a,b);
235             for in=1:max(size(c))
236                 if(c(in)==0 & a(in)=='0')
237                     bi = strcat([bi p2(j,1)]);
238                     bi = strcat([bi var(in) ' ' ' '
                                   ]);
239                     bi = strcat([bi " + "]);
240                 end
241                 if(c(in)==0 & a(in)=='1')
242                     bi = strcat([bi p2(j,1)]);
243                     bi = strcat([bi var(in)]);
244                     bi = strcat([bi " + "]);
245                 end
246             end
247         end
248     end
249 end
250 //checking the single cell cases
251 for i=1:n
252     for j=1:n
253         if(k(i,j,2)==0 & k(i,j,1)==1)
254             a=strsplit(temp1(1,j));
255             b=strsplit(temp2(i,1));
256             for in=1:max(size(a(:,1)))
257                 if(a(in,1)=='1')
258                     bi = strcat([bi var(in+2)]);
259                 else
260                     if(a(in,1)=='0')
261                         bi = strcat([bi var(2+in)
                                       ) ' ' ' ']);
262                     end

```

```

263         end
264     end
265     for in=1:max(size(b(:,1)))
266         if(b(in,1)=='1')
267             bi = strcat([bi var(in)]);
268         else
269             if(b(in,1)=='0')
270                 bi = strcat([bi var(in)
271                             '''']);
272             end
273         end
274         bi = strcat([bi " + "]);
275     end
276 end
277 end
278 bi = strcat([bi "0 "]);
279 endfunction

```

---

#### Scilab code AP 5 number of zeros and ones

```

1 function res=noof1(a,z) //this function returns both
   the no of zeros and ones in given matrix
2     res=0;
3     for i=1:max(size(a(:,1)))
4         for j=1:max(size(a(1,:)))
5             if(a(i,j)==z)
6                 res=res+1;
7             end
8         end
9     end
10 endfunction

```

---

#### Scilab code AP 6 3-variable kmap(a)

```

1
2 //3-VARIABLE KMAP

```

```

3 //this function returns the a string containing the
  minimized expression for the given 3 variable
  kmap
4 //this function requires
5 //noof.sci
6 //noof0.sci
7 function bi = kmap3a(k)
8     n=4;
9     m=2;
10 k(:, :, 2)=zeros(m,n);
11     var=['An' 'X' 'Y'];
12     p1=['An' '' 'An'];
13     p2=['X' 'Y' '' ; 'X' 'Y' ; 'XY' ; 'XY' '' ];
14
15     cmn4=4;
16     cmn2=2;
17     temp=1;
18     disp(k(:, :, 1));
19 bi = ' ';
20 //checking all the 8 1's cases
21     for i=1:m
22         for j=1:n
23             if(k(i,j)~=1 & k(i,j)~=2)
24                 temp=0;
25                 break;
26             end
27         end
28     end
29     if(temp==1)
30         bi = strcat([bi "1"]);
31         return;
32     end
33 //checking all the 4 1's cases
34     z1=ones(1,4);
35     z2=ones(4,1);
36     z3=ones(2,2);
37     temp1=['0' '1'];
38     temp2=['00' ; '01' ; '11' ; '10'];

```

```

39     for t=1:m
40         z=k(t, :, 1);
41         no=noof(k(t, :, 2));
42         if(noof0(z)==0 & no<cmn4 & noof(z)>0)
43             k(t, :, 2)=z1;
44             a=strsplit(temp1(1,t));
45             for in=1:max(size(a))
46                 if(a(in)=='0')
47                     bi = strcat([bi var(in) ' ']);
48                 end
49                 if(a(in)=='1')
50                     bi = strcat([bi var(in)]);
51                 end
52             end
53             bi = strcat([bi " + "]);
54         end
55     end
56     for i=1:m-1
57         for j=1:n
58             t1=i+1;
59             if(j==n)
60                 t2=1;
61             else
62                 t2=j+1;
63             end
64             z4=[k(i, j, 1) k(i, t2, 1);k(t1, j, 1) k(t1, t2
65                 , 1)];
66             z5=[k(i, j, 2) k(i, t2, 2);k(t1, j, 2) k(t1, t2
67                 , 2)];
68             no=noof(z5);
69             if(noof0(z4)==0 & no<cmn4 & noof(z4)>0)
70                 k(i, j, 2)=1;
71                 k(i, t2, 2)=1;
72                 k(t1, j, 2)=1;
73                 k(t1, t2, 2)=1;
74                 a=strsplit(temp2(j,1));
75                 b=strsplit(temp2(t2,1));

```



```

74         c=strcmp(a,b);
75         for in=1:max(size(c))
76             if(c(in)==0 & a(in)=='0')
77                 bi = strcat([bi var(1+in) ' '
78                             ' ' ]);
79             end
80             if(c(in)==0 & a(in)=='1')
81                 bi = strcat([bi var(1+in)])
82             ;
83         end
84         bi = strcat([bi " + "]);
85     end
86 end
87 //checking all the 2 1's cases
88 z6=[1 1];
89 z7=z6';
90 for i=1:m
91     for j=1:n
92         t1=i+1;
93         if(j==n)
94             t2=1;
95         else
96             t2=j+1;
97         end
98         z8=[k(i,j,1) k(i,t2,1)];
99         z9=[k(i,j,2) k(i,t2,2)];
100        no1=noof(z9);
101        if(noof0(z8)==0 & no1<cmn2 & noof(z8)>0)
102            k(i,j,2)=1;
103            k(i,t2,2)=1;
104            bi = strcat([bi p1(1,i)]);
105            a=strsplit(temp2(j,1));
106            b=strsplit(temp2(t2,1));
107            c=strcmp(a,b);
108            for in=1:max(size(c))

```

```

110         if(c(in)==0 & a(in)=='0')
111             bi = strcat([bi var(1+in) ' '
112                          ' ']);
112             bi = strcat([bi " + "]);
113         end
114         if(c(in)==0 & a(in)=='1')
115             bi = strcat([bi var(1+in)])
116                 ;
116             bi = strcat([bi " + "]);
117         end
118     end
119 end
120 end
121 end
122 for i=1:m-1
123     for j=1:n
124         t1=i+1;
125         if(j==n)
126             t2=1;
127         else
128             t2=j+1;
129         end
130         z10=[k(i,j,1);k(t1,j,1)];
131         z11=[k(i,j,2);k(t1,j,2)];
132         no2=noof(z11);
133         if(noof0(z10)==0 & no2<cmn2 & noof(z10)
134            >0)
134             k(i,j,2)=1;
135             k(t1,j,2)=1;
136             bi = strcat([bi p2(j,1)]);
137             bi = strcat([bi " + "]);
138         end
139     end
140 end
141 //checking if any single isolated 1's are left
142 for i=1:m
143     for j=1:n
144         if(k(i,j,2)==0 & k(i,j,1)==1)

```

```

145             bi = strcat([bi p1(1,i)]);
146             bi = strcat([bi p2(j,1)]);
147             bi = strcat([bi " + "]);
148         end
149     end
150 end
151     bi = strcat([bi " 0 "]);
152 endfunction

```

---

### Scilab code AP 7 2-variable kmap

```

1 //this function minimizes a two variable boolean
  expression using kmap
2 function bi =kmap2(k)
3 var=['A','B',' ','A','B','AB','AB',' ']
4 temp =1
5 for i=1:2 // intially checking for all 1's
6     for j=1:2
7         if k(i,j)==1 then
8             temp = temp + 1;
9         end
10    end
11 end
12 v=0;
13 bi = ' ' ;
14 if temp == 5 then
15     disp("The minimal expression is : 1');
16     v=1;
17 else
18 for i= 1 : 2 // considering all 2 1's cases
19     if k(i,1) == 1 & k(i,2) == 1 then
20         if i== 1 then
21             bi = strcat([ bi 'A' ' ' ] );v=1;
22         else
23             bi = strcat([ bi 'A' ] );v=1;
24         end
25     bi = strcat([ bi " + " ]);
26 end

```

```

27     if k(1,i) == 1 & k(2,i) == 1 then
28         if i== 1 then
29             bi = strcat ([ bi 'B' ' ' ] );v=1;
30         else
31             bi = strcat ([ bi 'B' ] );v=1;
32         end
33     end
34 end
35 end;
36 one(1)=k(2,1);
37 f=2;m=2;i=1;
38     for j=1:2
39         one(f)=k(i,j)
40         f=f+1;
41     end
42     i=2;
43     for j=2:-1:1
44         one(f)=k(i,j)
45         f=f+1;
46     end
47 one(6)=k(1,1);
48 if v==0 then // for isolated 1's
49     for i =2:5
50         if one(i)==1 & one(i+1)== 0 & one(i-1) ==0
51             then
52                 if m>0
53                     bi = strcat ([bi " + " ] );
54                     end;
55                     bi = strcat ([bi var(i-1)]);
56                     m=m+1;
57                 end
58             end
59 endfunction // final result will be stored in bi

```

---

**Scilab code AP 8 3-variable kmap**

1 //3-VARIABLE KMAP

```

2 //this function returns the a string containing the
   minimized expression for the given 3 variable
   kmap
3 //this function requires
4 //noof.sci
5 //noof0.sci
6 function bi = kmap3(k)
7     n=4;
8     m=2
9 k(:, :, 2)=zeros(m,n);
10    var=['Cn' 'Bn' 'An'];
11    p1=['Cn' '' 'Cn'];
12    p2=['Bn' 'An' '' ; 'Bn' 'An' ; 'BnAn' ; 'BnAn' '' ];
13    cmn4=4;
14    cmn2=2;
15    temp=1;
16    disp(k(:, :, 1));
17 bi = ' ';
18 //checking all the 8 1's cases
19     for i=1:m
20         for j=1:n
21             if(k(i,j)~=1 & k(i,j)~=2)
22                 temp=0;
23                 break;
24             end
25         end
26     end
27     if(temp==1)
28         bi = strcat([bi "1"]);
29         return;
30     end
31 //checking all the 4 1's cases
32 z1=ones(1,4);
33 z2=ones(4,1);
34 z3=ones(2,2);
35 temp1=['0' '1'];
36 temp2=['00' ; '01' ; '11' ; '10'];
37 for t=1:m

```

```

38         z=k(t, :, 1);
39         no=noof(k(t, :, 2));
40         if(noof0(z)==0 & no<cmn4 & noof(z)>0)
41             k(t, :, 2)=z1;
42             a=strsplit(temp1(1,t));
43             for in=1:max(size(a))
44                 if(a(in)=='0')
45                     bi = strcat([bi var(in) ' ']);
46                 end
47                 if(a(in)=='1')
48                     bi = strcat([bi var(in)]);
49                 end
50             end
51             bi = strcat([bi " + "]);
52         end
53     end
54     for i=1:m-1
55         for j=1:n
56             t1=i+1;
57             if(j==n)
58                 t2=1;
59             else
60                 t2=j+1;
61             end
62             z4=[k(i, j, 1) k(i, t2, 1);k(t1, j, 1) k(t1, t2
63                 ,1)];
64             z5=[k(i, j, 2) k(i, t2, 2);k(t1, j, 2) k(t1, t2
65                 ,2)];
66             no=noof(z5);
67             if(noof0(z4)==0 & no<cmn4 & noof(z4)>0)
68                 k(i, j, 2)=1;
69                 k(i, t2, 2)=1;
70                 k(t1, j, 2)=1;
71                 k(t1, t2, 2)=1;
72                 a=strsplit(temp2(j, 1));
73                 b=strsplit(temp2(t2, 1));
74                 c=strcmp(a, b);

```

```

73         for in=1:max(size(c))
74             if(c(in)==0 & a(in)=='0')
75                 bi = strcat([bi var(1+in) ' '
76                             ' ' ]);
77             end
78             if(c(in)==0 & a(in)=='1')
79                 bi = strcat([bi var(1+in)])
80                 ;
81             end
82         end
83     end
84 end
85 //checking all the 2 1's cases
86 z6=[1 1];
87 z7=z6';
88 for i=1:m
89     for j=1:n
90         t1=i+1;
91         if(j==n)
92             t2=1;
93         else
94             t2=j+1;
95         end
96         z8=[k(i,j,1) k(i,t2,1)];
97         z9=[k(i,j,2) k(i,t2,2)];
98         no1=noof(z9);
99         if(noof0(z8)==0 & no1<cmn2 & noof(z8)>0)
100             k(i,j,2)=1;
101             k(i,t2,2)=1;
102             bi = strcat([bi p1(1,i)]);
103             a=strsplit(temp2(j,1));
104             b=strsplit(temp2(t2,1));
105             c=strcmp(a,b);
106             for in=1:max(size(c))
107                 if(c(in)==0 & a(in)=='0')

```

```

109             bi = strcat([bi var(1+in) ''
110                          '']);
111             bi = strcat([bi " + "]);
112         end
113         if(c(in)==0 & a(in)=='1')
114             bi = strcat([bi var(1+in)])
115             ;
116             bi = strcat([bi " + "]);
117         end
118     end
119 end
120 for i=1:m-1
121     for j=1:n
122         t1=i+1;
123         if(j==n)
124             t2=1;
125         else
126             t2=j+1;
127         end
128         z10=[k(i,j,1);k(t1,j,1)];
129         z11=[k(i,j,2);k(t1,j,2)];
130         no2=noof(z11);
131         if(noof0(z10)==0 & no2<cmn2 & noof(z10)
132            >0)
133             k(i,j,2)=1;
134             k(t1,j,2)=1;
135             bi = strcat([bi p2(j,1)]);
136             bi = strcat([bi " + "]);
137         end
138     end
139 //checking if any single isolated 1's are left
140 for i=1:m
141     for j=1:n
142         if(k(i,j,2)==0 & k(i,j,1)==1)
143             bi = strcat([bi p1(1,i)]);

```



```

144             bi = strcat([bi p2(j,1)]);
145             bi = strcat([bi " + "]);
146         end
147     end
148 end
149     bi = strcat([bi " 0 "]);
150 endfunction

```

---

#### Scilab code AP 9 4-variable kmap(sx1x2)

```

1 function []=kmapsx(k) // this fncions prints the
   minimized expression for the given kmap .
2 // it requires noof.sci
3 //so the above mentioned function shoub be execute
   before executing this function .
4
5     n=4;
6     k(:,:,2)=zeros(n,n);
7     var=['X2' 'X3' 'S' 'X1'];
8
9     p1=['X2' 'X3'' ' 'X2' 'X3' 'X2X3' 'X2X3'' '];
10    p2=['S' 'X1'' ' 'S' 'X1' 'SX1' 'SX1'' '];
11    cmn4=4;
12    cmn2=2;
13    temp=1;
14    disp(k(:,:,1));
15    disp(" is :");
16    disp(" ")
17    //checking the 16 cells case
18    for i=1:n
19        for j=1:n
20            if(k(i,j)~=1)
21                temp=0;
22                break;
23            end
24        end
25    end
26    printf(' ');

```

```

27     if(temp==1)
28         printf("1");
29         abort;
30     end
31     //checking the 8 cells cases
32     z1=ones(2,4);
33     z2=ones(4,2);
34     temp1=['00' '01' '11' '10'];
35     temp2=temp1';
36     for i=1:n
37         if(i==4)
38             t=1;
39         else
40             t=i+1;
41         end
42         z=[k(i, :, 1);k(t, :, 1)];
43         if(z==z1)
44             k(i, :, 2)=[1 1 1 1];
45             k(t, :, 2)=[1 1 1 1];
46             a=strsplit(temp2(i,1));
47             b=strsplit(temp2(t,1));
48             c=strcmp(a,b);
49             for in=1:max(size(c))
50                 if(c(in)==0 & a(in)=='0')
51                     printf('%s' ',var(in));
52                     printf(' + ');
53                     break;
54                 else
55                     if(c(in)==0 & a(in)=='1')
56                         printf(var(in));
57                         printf(' + ');
58                         break;
59                     end
60                 end
61             end
62         end
63     end
64     for j=1:n

```

```

65         if(j==4)
66             t=1;
67         else
68             t=j+1;
69         end
70         z=[k(:,j,1) k(:,t,1)];
71         if(z==z2)
72             k(:,j,2)=[1;1;1;1];
73             k(:,t,2)=[1;1;1;1];
74             a=strsplit(temp1(1,j));
75             b=strsplit(temp1(1,t));
76             c=strcmp(a,b);
77             for in=1:max(size(c))
78                 if(c(in)==0 & a(in)=='0')
79                     printf('%s''',var(2+in));
80                     printf(' + ');
81                     break;
82                 else
83                     if(c(in)==0 & a(in)=='1')
84                         printf(var(2+in));
85                         printf(' + ');
86                         break;
87                     end
88                 end
89             end
90         end
91     end
92     //checking the 4 cells cases
93     z1=ones(1,4);
94     z2=ones(4,1);
95     z3=ones(2,2);
96     temp1=['00' '01' '11' '10'];
97     temp2=temp1';
98     for t=1:n
99         z=k(t,(:,1));
100        no=noof(k(t,(:,2)));
101        if(z==z1 & no<cmn4)
102            k(t,(:,2))=z1;

```

```

103         a=strsplit(temp1(1,t));
104         for in=1:max(size(a))
105             if(a(in)=='0')
106                 printf('%s''',var(in));
107             end
108             if(a(in)=='1')
109                 printf(var(in));
110             end
111         end
112         printf(" + ");
113     end
114 end
115 for t=1:n
116     z=k(:,t,1);
117     no=noof(k(:,t,2));
118     if(z==z2 & no<cmn4)
119         k(:,t,2)=z2;
120         a=strsplit(temp2(t,1));
121         for in=1:max(size(a))
122             if(a(in)=='0')
123                 printf('%s''',var(2+in));
124             end
125             if(a(in)=='1')
126                 printf(var(2+in));
127             end
128         end
129         printf(" + ");
130     end
131 end
132 for i=1:n
133     for j=1:n
134         if(i==n)
135             t1=1;
136         else
137             t1=i+1;
138         end
139         if(j==n)
140             t2=1;

```

```

141         else
142             t2=j+1;
143         end
144         z4=[k(i,j,1) k(i,t2,1);k(t1,j,1) k(t1,t2
,1)];
145         z5=[k(i,j,2) k(i,t2,2);k(t1,j,2) k(t1,t2
,2)];
146         no=noof(z5);
147         if(z4==z3 & no<cmn4)
148             k(i,j,2)=1;
149             k(i,t2,2)=1;
150             k(t1,j,2)=1;
151             k(t1,t2,2)=1;
152             a=strsplit(temp2(i,1));
153             b=strsplit(temp2(t1,1));
154             c=strcmp(a,b);
155             for in=1:max(size(c))
156                 if(c(in)==0 & a(in)=='0')
157                     printf('%s''',var(in));
158                 end
159                 if(c(in)==0 & a(in)=='1')
160                     printf(var(in));
161                 end
162             end
163             a=strsplit(temp1(1,j));
164             b=strsplit(temp1(1,t2));
165             c=strcmp(a,b);
166             for in=1:max(size(c))
167                 if(c(in)==0 & a(in)=='0')
168                     printf('%s''',var(2+in));
169                 end
170                 if(c(in)==0 & a(in)=='1')
171                     printf(var(2+in));
172                 end
173             end
174             printf(" + ");
175         end
176     end

```

```

177     end
178     //checking all the 2 cells cases
179     z6=[1 1];
180     z7=z6';
181     for i=1:n
182         for j=1:n
183             if(i==n)
184                 t1=1;
185             else
186                 t1=i+1;
187             end
188             if(j==n)
189                 t2=1;
190             else
191                 t2=j+1;
192             end
193             z8=[k(i,j,1) k(i,t2,1)];
194             z9=[k(i,j,2) k(i,t2,2)];
195             no1=noof(z9);
196             if(z8==z6 & no1<cmn2 )
197                 k(i,j,2)=1;
198                 k(i,t2,2)=1;
199                 a=strsplit(temp1(1,j));
200                 b=strsplit(temp1(1,t2));
201                 c=strcmp(a,b);
202                 for in=1:max(size(c))
203                     if(c(in)==0 & a(in)=='0')
204                         printf(p1(1,i));
205                         printf('%s''',var(2+in));
206                         printf(" + ");
207                     end
208                     if(c(in)==0 & a(in)=='1')
209                         printf(p1(1,i));
210                         printf(var(2+in));
211                         printf(" + ");
212                     end
213                 end
214             end
end

```

```

215         end
216     end
217     for i=1:n
218         for j=1:n
219             if(i==n)
220                 t1=1;
221             else
222                 t1=i+1;
223             end
224             if(j==n)
225                 t2=1;
226             else
227                 t2=j+1;
228             end
229             z10=[k(i,j,1);k(t1,j,1)];
230             z11=[k(i,j,2);k(t1,j,2)];
231             no2=noof(z11);
232             if(z10==z7 & no2<cmn2)
233                 k(i,j,2)=1;
234                 k(t1,j,2)=1;
235                 a=strsplit(temp2(i,1));
236                 b=strsplit(temp2(t1,1));
237                 c=strcmp(a,b);
238                 for in=1:max(size(c))
239                     if(c(in)==0 & a(in)=='0')
240                         printf(p2(j,1));
241                         printf('%s''',var(in));
242                         printf(" + ");
243                     end
244                     if(c(in)==0 & a(in)=='1')
245                         printf(p2(j,1));
246                         printf(var(in));
247                         printf(" + ");
248                     end
249                 end
250             end
251         end
252     end

```

```

253 // checking all the single cell cases
254 for i=1:n
255     for j=1:n
256         if(k(i,j,2)==0 & k(i,j,1)==1)
257             a=strsplit(temp1(1,j));
258             b=strsplit(temp2(i,1));
259             for in=1:max(size(a(:,1)))
260                 if(a(in,1)=='1')
261                     printf(var(in+2));
262                 else
263                     if(a(in,1)=='0')
264                         printf('%s''',var(2+in))
265                             ;
266                             end
267                             end
268                             for in=1:max(size(b(:,1)))
269                                 if(b(in,1)=='1')
270                                     printf(var(in));
271                                 else
272                                     if(b(in,1)=='0')
273                                         printf('%s''',var(in));
274                                     end
275                                     end
276                                 end
277                                 if(i~=4 & j~=4)
278                                     printf(" + ");
279                                 end
280                             end
281                         end
282                     end
283                 end
284             printf("0");
285 endfunction

```

---

Scilab code AP 10 4-variable kmap



```

2 //this funtion prints the minimal expression of a
   given 4-vriable kmap
3 //this program requires noof.sci
4 function []=kmap(k)
5     n=4;
6     k(:,:,2)=zeros(n,n);
7     var=['A' 'B' 'C' 'D'];
8     p1=['A' 'B' '' 'A' 'B' 'AB' 'AB' '']
9     p2=['C' 'D' '' ; 'C' 'D'; 'CD'; 'CD' ''];
10    cmn4=4;
11    cmn2=2;
12    temp=1;
13    disp(k(:,:,1));
14    disp(" is ");
15    disp(" ")
16    //checking for 16 cells
17    for i=1:n
18        for j=1:n
19            if(k(i,j)~=1)
20                temp=0;
21                break;
22            end
23        end
24    end
25    printf(' ');
26    if(temp==1)
27        printf("1");
28        abort;
29    end
30    //checking 8 cells cases
31    z1=ones(2,4);
32    z2=ones(4,2);
33    temp1=['00' '01' '11' '10'];
34    temp2=temp1';
35    for i=1:n
36        if(i==4)
37            t=1;
38        else

```

```

39         t=i+1;
40     end
41     z=[k(i,:,1);k(t,:,1)];
42     if(z==z1)
43         k(i,:,2)=[1 1 1 1];
44         k(t,:,2)=[1 1 1 1];
45         a=strsplit(temp2(i,1));
46         b=strsplit(temp2(t,1));
47         c=strcmp(a,b);
48         for in=1:max(size(c))
49             if(c(in)==0 & a(in)=='0')
50                 printf('%s''',var(in));
51                 printf(' + ');
52                 break;
53             else
54                 if(c(in)==0 & a(in)=='1')
55                     printf(var(in));
56                     printf(' + ');
57                     break;
58                 end
59             end
60         end
61     end
62 end
63 for j=1:n
64     if(j==4)
65         t=1;
66     else
67         t=j+1;
68     end
69     z=[k(:,j,1) k(:,t,1)];
70     if(z==z2)
71         k(:,j,2)=[1;1;1;1];
72         k(:,t,2)=[1;1;1;1];
73         a=strsplit(temp1(1,j));
74         b=strsplit(temp1(1,t));
75         c=strcmp(a,b);
76         for in=1:max(size(c))

```

```

77         if(c(in)==0 & a(in)=='0')
78             printf('%s' ',var(2+in));
79             printf(' + ');
80             break;
81         else
82             if(c(in)==0 & a(in)=='1')
83                 printf(var(2+in));
84                 printf(' + ');
85                 break;
86             end
87         end
88     end
89 end
90 end
91 //checking all 4 cells cases
92 z1=ones(1,4);
93 z2=ones(4,1);
94 z3=ones(2,2);
95 temp1=['00' '01' '11' '10'];
96 temp2=temp1';
97 for t=1:n
98     z=k(t,:,1);
99     no=noof(k(t,:,2));
100    if(z==z1 & no<cmn4)
101        k(t,:,2)=z1;
102        a=strsplit(temp1(1,t));
103        for in=1:max(size(a))
104            if(a(in)=='0')
105                printf('%s' ',var(in));
106            end
107            if(a(in)=='1')
108                printf(var(in));
109            end
110        end
111        printf(" + ");
112    end
113 end
114 for t=1:n

```

```

115         z=k(:,t,1);
116         no=noof(k(:,t,2));
117         if(z==z2 & no<cmn4)
118             k(:,t,2)=z2;
119             a=strsplit(temp2(t,1));
120             for in=1:max(size(a))
121                 if(a(in)=='0')
122                     printf('%s''',var(2+in));
123                 end
124                 if(a(in)=='1')
125                     printf(var(2+in));
126                 end
127             end
128             printf(" + ");
129         end
130     end
131     for i=1:n
132         for j=1:n
133             if(i==n)
134                 t1=1;
135             else
136                 t1=i+1;
137             end
138             if(j==n)
139                 t2=1;
140             else
141                 t2=j+1;
142             end
143             z4=[k(i,j,1) k(i,t2,1);k(t1,j,1) k(t1,t2
144                 ,1)];
145             z5=[k(i,j,2) k(i,t2,2);k(t1,j,2) k(t1,t2
146                 ,2)];
147             no=noof(z5);
148             if(z4==z3 & no<cmn4)
149                 k(i,j,2)=1;
150                 k(i,t2,2)=1;
151                 k(t1,j,2)=1;
152                 k(t1,t2,2)=1;

```

```

151         a=strsplit(temp2(i,1));
152         b=strsplit(temp2(t1,1));
153         c=strcmp(a,b);
154         for in=1:max(size(c))
155             if(c(in)==0 & a(in)=='0')
156                 printf('%s ',' ',var(in));
157             end
158             if(c(in)==0 & a(in)=='1')
159                 printf(var(in));
160             end
161         end
162         a=strsplit(temp1(1,j));
163         b=strsplit(temp1(1,t2));
164         c=strcmp(a,b);
165         for in=1:max(size(c))
166             if(c(in)==0 & a(in)=='0')
167                 printf('%s ',' ',var(2+in));
168             end
169             if(c(in)==0 & a(in)=='1')
170                 printf(var(2+in));
171             end
172         end
173         printf(" + ");
174     end
175 end
176 end
177 //checking all 2 cells cases
178 z6=[1 1];
179 z7=z6';
180 for i=1:n
181     for j=1:n
182         if(i==n)
183             t1=1;
184         else
185             t1=i+1;
186         end
187         if(j==n)
188             t2=1;

```

```

189         else
190             t2=j+1;
191         end
192         z8=[k(i,j,1) k(i,t2,1)];
193         z9=[k(i,j,2) k(i,t2,2)];
194         no1=noof(z9);
195         if(z8==z6 & no1<cmn2 )
196             k(i,j,2)=1;
197             k(i,t2,2)=1;
198             a=strsplit(temp1(1,j));
199             b=strsplit(temp1(1,t2));
200             c=strcmp(a,b);
201             for in=1:max(size(c))
202                 if(c(in)==0 & a(in)=='0')
203                     printf(p1(1,i));
204                     printf('%s' ',var(2+in));
205                     printf(" + ");
206                 end
207                 if(c(in)==0 & a(in)=='1')
208                     printf(p1(1,i));
209                     printf(var(2+in));
210                     printf(" + ");
211                 end
212             end
213         end
214     end
215 end
216 for i=1:n
217     for j=1:n
218         if(i==n)
219             t1=1;
220         else
221             t1=i+1;
222         end
223         if(j==n)
224             t2=1;
225         else
226             t2=j+1;

```

```

227         end
228         z10=[k(i,j,1);k(t1,j,1)];
229         z11=[k(i,j,2);k(t1,j,2)];
230         no2=noof(z11);
231         if(z10==z7 & no2<cmn2)
232             k(i,j,2)=1;
233             k(t1,j,2)=1;
234             a=strsplit(temp2(i,1));
235             b=strsplit(temp2(t1,1));
236             c=strcmp(a,b);
237             for in=1:max(size(c))
238                 if(c(in)==0 & a(in)=='0')
239                     printf(p2(j,1));
240                     printf(' %s ',var(in));
241                     printf(" + ");
242                 end
243                 if(c(in)==0 & a(in)=='1')
244                     printf(p2(j,1));
245                     printf(var(in));
246                     printf(" + ");
247                 end
248             end
249         end
250     end
251 end
252 //checking for isolated cell
253 for i=1:n
254     for j=1:n
255         if(k(i,j,2)==0 & k(i,j,1)==1)
256             a=strsplit(temp1(1,j));
257             b=strsplit(temp2(i,1));
258             for in=1:max(size(a(:,1)))
259                 if(a(in,1)=='1')
260                     printf(var(in+2));
261                 else
262                     if(a(in,1)=='0')
263                         printf(' %s ',var(2+in))

```

```

264         end
265     end
266     end
267     for in=1:max(size(b(:,1)))
268         if(b(in,1)=='1')
269             printf(var(in));
270         else
271             if(b(in,1)=='0')
272                 printf('%s''',var(in));
273             end
274         end
275     end
276     if(i~=4 & j~=4)
277         printf(" + ");
278     end
279 end
280     end
281 end
282     printf("0");
283 endfunction

```

---

#### Scilab code AP 11 4-variable kmap pos

```

1 //this funtion prints the minimal expression in the
  Pruduct of sums form for a given 4-vriable kmap
2 //this program requires noof.sci
3 function []=kmappos(k)
4     n=4;
5     k(:,:,2)=zeros(n,n);
6     var=['A' 'B' 'C' 'D'];
7 p1=['A + B' 'A + B'' 'A'' + B'' 'A'' + B'];
8 p2=['C + D'; 'C + D'''; 'C'' + D'''; 'C'' + D'];
9     cmn4=4;
10    cmn2=2;
11    temp=1;
12    disp(k(:,:,1));
13    disp(" is ");
14    disp(" ")

```



```

15 //checking for 16 cells
16 for i=1:n
17     for j=1:n
18         if(k(i,j)~=1)
19             temp=0;
20             break;
21         end
22     end
23 end
24 printf(' ');
25 if(temp==1)
26     printf("1");
27     abort;
28 end
29 for i=1:n
30     for j=1:n
31         if(k(i,j)~=0)
32             temp=0;
33             break;
34         end
35     end
36 end
37 printf(' ');
38 if(temp==1)
39     printf("0");
40     abort;
41 end
42
43 //checking for 8 cells cases
44 z1=zeros(2,4);
45 z2=zeros(4,2);
46 temp1=['00' '01' '11' '10'];
47 temp2=temp1';
48 for i=1:n
49     if(i==4)
50         t=1;
51     else
52         t=i+1;

```

```

53         end
54         z=[k(i, :, 1);k(t, :, 1)];
55         if(z==z1)
56             printf(' ');
57             k(i, :, 2)=[1 1 1 1];
58             k(t, :, 2)=[1 1 1 1];
59             a=strsplit(temp2(i,1));
60             b=strsplit(temp2(t,1));
61             c=strcmp(a,b);
62             for in=1:max(size(c))
63                 if(c(in)==0 & a(in)=='0')
64                     printf(var(in));
65                     break;
66                 else
67                     if(c(in)==0 & a(in)=='1')
68                         printf('%s', var(in));
69                         break;
70                     end
71                 end
72             end
73             printf(' ');
74         end
75     end
76     for j=1:n
77         if(j==4)
78             t=1;
79         else
80             t=j+1;
81         end
82         z=[k(:, j, 1) k(:, t, 1)];
83         if(z==z2)
84             printf(' ');
85             k(:, j, 2)=[1;1;1;1];
86             k(:, t, 2)=[1;1;1;1];
87             a=strsplit(temp1(1,j));
88             b=strsplit(temp1(1,t));
89             c=strcmp(a,b);
90             for in=1:max(size(c))

```

```

91         if(c(in)==0 & a(in)=='0')
92             printf(var(2+in));
93             break;
94         else
95             if(c(in)==0 & a(in)=='1')
96                 printf('%s' ',var(2+in));
97                 break;
98             end
99         end
100     end
101     printf(')')
102 end
103 end
104 //checking for 4 cells cases
105 z1=zeros(1,4);
106 z2=zeros(4,1);
107 z3=zeros(2,2);
108 temp1=['00' '01' '11' '10'];
109 temp2=temp1';
110 for t=1:n
111     z=k(t, :, 1);
112     no=noof(k(t, :, 2));
113     if(z==z1 & no<cmn4)
114         printf('( ')
115         k(t, :, 2)=[1 1 1 1];
116         a=strsplit(temp1(1,t));
117         for in=1:max(size(a))
118             if(a(in)=='0')
119                 if in ~= 1 then
120                     printf(' + ');
121                 end
122                 printf(var(in));
123             end
124             if(a(in)=='1')
125                 if in ~= 1 then
126                     printf(' + ');
127                 end
128                 printf('%s' ',var(in));

```

```

129             end
130             end
131             printf(")");
132         end
133     end
134     for t=1:n
135         z=k(:,t,1);
136         no=noof(k(:,t,2));
137         if(z==z2 & no<cmn4)
138             printf(' ');
139             k(:,t,2)=[1;1;1;1];
140             a=strsplit(temp2(t,1));
141             for in=1:max(size(a))
142                 if(a(in)=='0')
143                     if in ~= 1 then
144                         printf(' + ');
145                     end
146                     printf(var(2+in));
147                 end
148                 if(a(in)=='1')
149                     if in ~= 1 then
150                         printf(' + ');
151                     end
152                     printf('%s',var(2+in));
153                 end
154             end
155             printf(")");
156         end
157     end
158     for i=1:n
159         for j=1:n
160             if(i==n)
161                 t1=1;
162             else
163                 t1=i+1;
164             end
165             if(j==n)
166                 t2=1;

```

```

167         else
168             t2=j+1;
169         end
170         z4=[k(i,j,1) k(i,t2,1);k(t1,j,1) k(t1,t2
171             ,1)];
172         z5=[k(i,j,2) k(i,t2,2);k(t1,j,2) k(t1,t2
173             ,2)];
174         no=noof(z5);
175         if(z4==z3 & no<cmn4)
176             printf('(')
177             k(i,j,2)=1;
178             k(i,t2,2)=1;
179             k(t1,j,2)=1;
180             k(t1,t2,2)=1;
181             a=strsplit(temp2(i,1));
182             b=strsplit(temp2(t1,1));
183             c=strcmp(a,b);
184             for in=1:max(size(c))
185                 if(c(in)==0 & a(in)=='0')
186                     printf(var(in));
187                 end
188                 if(c(in)==0 & a(in)=='1')
189                     printf('%s''',var(in));
190                 end
191             a=strsplit(temp1(1,j));
192             b=strsplit(temp1(1,t2));
193             c=strcmp(a,b);
194             for in=1:max(size(c))
195                 if(c(in)==0 & a(in)=='0')
196                     printf(' + ');
197                     printf(var(2+in));
198                 end
199                 if(c(in)==0 & a(in)=='1')
200                     printf(' + ');
201                     printf('%s''',var(2+in));
202             end

```

```

203             end
204             printf(")");
205         end
206     end
207 end
208 //checking for 2 cells
209 z6=[0 0];
210 z7=z6';
211 for i=1:n
212     for j=1:n
213         if(i==n)
214             t1=1;
215         else
216             t1=i+1;
217         end
218         if(j==n)
219             t2=1;
220         else
221             t2=j+1;
222         end
223         z8=[k(i,j,1) k(i,t2,1)];
224         z9=[k(i,j,2) k(i,t2,2)];
225         no1=noof(z9);
226         if(z8==z6 & no1<cmn2 )
227
228             printf('(');
229             k(i,j,2)=1;
230             k(i,t2,2)=1;
231             a=strsplit(temp1(1,j));
232             b=strsplit(temp1(1,t2));
233             c=strcmp(a,b);
234             for in=1:max(size(c))
235                 if(c(in)==0 & a(in)=='0')
236                     printf(p1(1,i));
237                     printf(' + ');
238                     printf(var(2+in));
239                     printf(")");
240             end

```

```

241         if(c(in)==0 & a(in)=='1')
242             printf(p1(1,i));
243             printf(" + ");
244             printf('%s',var(2+in));
245
246             printf(")");
247         end
248     end
249 end
250
251 end
252 for i=1:n
253     for j=1:n
254         if(i==n)
255             t1=1;
256         else
257             t1=i+1;
258         end
259         if(j==n)
260             t2=1;
261         else
262             t2=j+1;
263         end
264         z10=[k(i,j,1);k(t1,j,1)];
265         z11=[k(i,j,2);k(t1,j,2)];
266         no2=noof(z11);
267         if(z10==z7 & no2<cmn2)
268             printf(' ');
269             k(i,j,2)=1;
270             k(t1,j,2)=1;
271             a=strsplit(temp2(i,1));
272             b=strsplit(temp2(t1,1));
273             c=strcmp(a,b);
274             for in=1:max(size(c))
275                 if(c(in)==0 & a(in)=='0')
276                     printf(p2(j,1));
277                     printf(" + ");
278                     printf(var(in));

```

```

279         printf(")");
280     end
281     if(c(in)==0 & a(in)=='1')
282         printf(p2(j,1));
283         printf(" + ");
284         printf('%s',var(in));
285
286         printf(")");
287     end
288 end
289     end
290 end
291 end
292 //for single cell
293 for i=1:n
294     for j=1:n
295         if(k(i,j,2)==0 & k(i,j,1)==0)
296             printf('( ');
297             a=strsplit(temp1(1,j));
298             b=strsplit(temp2(i,1));
299             for in=1:max(size(a(:,1)))
300                 if(a(in,1)=='1')
301                     printf('%s',var(2+in));
302                     printf(' + ');
303                 else
304                     if(a(in,1)=='0')
305                         printf(var(in+2));
306
307                         printf(' + ');
308                     end
309                 end
310             end
311             for in=1:max(size(b(:,1)))
312                 if(b(in,1)=='1')
313                     printf('%s',var(in));
314                     if(in~=max(size(b(:,1))))
315                         printf(' + ');
316                     end

```



```
317         else
318             if (b(in,1) == '0')
319                 printf(var(in));
320
321                 if (in ~ = max(size(b(:,1))))
322                     printf(' + ');
323                 end
324
325             end
326         end
327     end
328     printf(")");
329 end
330 end
331 end
332 endfunction
```

---