CHAPTER 7

Exercises

E7.1 (a) For the whole part, we have:

	Quotient	Remainders
23/2	11	1
11/2	5	1
5/2	2	1
2/2	1	0
1/2	0	1

Reading the remainders in reverse order, we obtain:

$23_{10} = 10111_2$

For the fractional part we have

$2 \times 0.75 = 1 + 0.5$

$2 \times 0.50 = 1 + 0$

Thus we have

$0.75_{10} = 0.110000_2$

Finally, the answer is $23.75_{10} = 10111.11_2$

(b) For the whole part we have:

	Quotient	Remainders
17/2	8	1
8/2	4	0
4/2	2	0
2/2	1	0
1/2	0	1

Reading the remainders in reverse order we obtain:

$17_{10} = 10001_2$

For the fractional part we have

$2 \times 0.25 = 0 + 0.5$

$2 \times 0.50 = 1 + 0$

Thus we have

$0.25_{10} = 0.010000_2$

Finally, the answer is $17.25_{10} = 10001.01_2$

(c) For the whole part we have:

	Quotient	Remainders
4/2	2	0
2/2	1	0
1/2	0	1

Reading the remainders in reverse order we obtain:

$4_{10} = 100_2$

For the fractional part, we have

$2 \times 0.30 = 0 + 0.6$

$2 \times 0.60 = 1 + 0.2$

$2 \times 0.20 = 0 + 0.4$

$2 \times 0.40 = 0 + 0.8$

$2 \times 0.80 = 1 + 0.6$

$2 \times 0.60 = 1 + 0.2$

Thus we have

$0.30_{10} = 0.010011_2$

Finally, the answer is $4.3_{10} = 100.010011_2$

E7.2 (a) $1101.111_2 = 1\times2^3 + 1\times2^2 +0\times2^1 +1\times2^0 +1\times2^{-1} +1\times2^{-2} +1\times2^{-3} = 13.875_{10}$

(b) $100.001_2 = 1\times2^2 +0\times2^1 +0\times2^0 +0\times2^{-1} +0\times2^{-2} +1\times2^{-3} = 4.125_{10}$

E7.3 (a) Using the procedure of Exercise 7.1, we have

$97_{10} = 1100001_2$

Then adding two leading zeros and forming groups of three bits we have

$001\ 100\ 001_2 = 141_8$

Adding a leading zero and forming groups of four bits we obtain

$0110\ 0001 = 61_{16}$

(b) Similarly

$229_{10} = 11100101_2 = 345_8 = E5_{16}$

E7.4 (a) $72_8 = 111\ 010 = 111010_2$

(b) $FA6_{16} = 1111\ 1010\ 0110 = 111110100110_2$

E7.5 $197_{10} = 0001\ 1001\ 0111 = 000110010111_{BCD}$

E7.6 To represent a distance of 20 inches with a resolution of 0.01 inches, we need 20/0.01 = 2000 code words. The number of code words in a Gray code is 2^L in which L is the length of the code words. Thus we need L = 11, which produces 2048 code words.

E7.7 (a) First we convert to binary

$$22_{10} = 16 + 4 + 2 = 10110_2$$

Because an eight-bit representation is called for, we append three leading zeros. Thus +22 becomes

00010110

in two's complement notation.

(b) First we convert +30 to binary form

$$30_{10} = 16 + 8 + 4 + 2 = 11110_2$$

Attaching leading zeros to make an eight-bit result we have

$$30_{10} = 00011110_2$$

Then we take the ones complement and add 1 to find the two's complement:

$$\begin{array}{lr} \text{one's complement:} & 11100001 \\ \text{add 1} & \underline{+1} \\ & 11100010 \end{array}$$

Thus the eight-bit two's complement representation for -30_{10} is 11100010.

E7.8 First we convert 19_{10} and -4_{10} to eight-bit two's complement form then we add the results.

$$\begin{array}{ll} 19 & 00010011 \\ \underline{-4} & \underline{111111100} \\ 15 & 00001111 \end{array}$$

Notice that we neglect the carry out of the left-most bit.

E7.9 See Tables 7.3 and 7.4 in the book.

E7.10 See Table 7.5 in the book.

E7.11 (a) To apply De Morgan's laws to the expression

$$AB + \bar{B}C$$

first we replace each logic variable by its inverse

$$\overline{A}\,\overline{B} + B\overline{C}$$

then we replace AND operations by OR operations and vice versa

$$(\overline{A} + \overline{B})(B + \overline{C})$$

finally we invert the entire expression so we have

$$D = AB + \overline{B}C = \overline{(\overline{A} + \overline{B})(B + \overline{C})}$$

(b) Following the steps of part (a) we have

$$\overline{[F(G + \overline{H}) + F\overline{G}]}$$

$$\overline{[\overline{F}(\overline{G} + H) + \overline{F}G]}$$

$$\overline{[(\overline{F} + \overline{G}H)(\overline{F} + G)]}$$

$$E = \overline{[F(G + \overline{H}) + F\overline{G}]} = [(\overline{F} + \overline{G}H)(\overline{F} + G)]$$

E7.12 For the AND gate we use De Morgan's laws to write

$$AB = \overline{(\overline{A} + \overline{B})}$$

See Figure 7.21 in the book for the logic diagrams.

E7.13 The truth table for the exclusive-OR operation is

A	B	$A \oplus B$
0	0	0
0	1	1
1	0	1
1	1	0

Focusing on the rows in which the result is 1, we can write the SOP expression

$$A \oplus B = \overline{A}B + A\overline{B}$$

The corresponding logic diagram is shown in Figure 7.25a in the book.

Focusing on the rows in which the result is 0, we can write the POS expression

$$A \oplus B = (A + B)(\overline{A} + \overline{B})$$

The corresponding logic diagram is shown in Figure 7.25b in the book.

E7.14 The truth table is shown in Table 7.7 in the book. Focusing on the rows in which the result is 1, we can write the SOP expression

$$A = \sum m(3, 6, 7, 8, 9, 12)$$

$$= \overline{F}\,\overline{D}GR + \overline{F}\,DG\overline{R} + \overline{F}DGR + F\overline{D}\,\overline{G}\,\overline{R} + F\overline{D}\,\overline{G}R + FD\overline{G}\,\overline{R}$$

Focusing on the rows in which the result is 0, we can write the POS expression

$$A = \prod M(0, 1, 2, 4, 5, 10, 11, 13, 14, 15)$$

$$= (F + D + G + R)(F + D + G + \overline{R})(F + D + \overline{G} + R)\cdots(\overline{F} + \overline{D} + \overline{G} + \overline{R})$$

E7.15 The Truth table is shown in Table 7.8.

E7.16 (a) $\overline{A}\,\overline{B}C\overline{D}$

(b) $\overline{A}\,\overline{B}\,\overline{C}\,D$

E7.17 See Figure 7.34 in the book.

E7.18 See Figure 7.35 in the book.

E7.19 Because S is high at $t = 0$, Q is high and remains so until R becomes high at $t = 3$. Q remains low until S becomes high at $t = 7$. Then Q remains high until R becomes high at $t = 11$.

E7.20 See Table 7.9.

E7.21 See Figure 7.49 in the book.

Answers for Selected Problems

P7.1* 1. When noise is added to a digital signal, the logic levels can still be exactly determined, provided that the noise is not too large in amplitude. Usually, noise cannot be completely removed from an analog signal.

2. Component values in digital circuits do not need to be as precise as in analog circuits.

3. Very complex digital logic circuits (containing millions of components) can be economically produced. Analog circuits often call for large capacitances and/or precise component values that are impossible to manufacture as large-scale integrated circuits.

P7.6 (a)* 5.625
(f)* 21.375

P7.7 (c)* $9.75_{10} = 1001.11_2 = 1001.01110101_{BCD}$

P7.9 (a)* 1101.11 + 101.111 = 10011.101

P7.10 (a)* $10010011.0101_{BCD} + 00110111.0001_{BCD} = 93.5_{10} + 37.1_{10} = 130.6_{10} =$
000100110000.0110_{BCD}

P7.11 (d)* $313.0625_{10} = 100111001.0001_2 = 471.04_8 = 139.1_{16}$

P7.12 (c)* 75 = 01001011
(d)* -87 = 10101001

P7.17*

0000
0001
0011
0010
0110
0111
0101
0100
1100
1101
1111
1110
1010
1011
1001
1000

P7.18 (a)* $FA5.6_{16} = 4005.375_{10}$

(b)* $725.3_8 = 469.375_{10}$

P7.19 (a)* $11101000 \Rightarrow \overbrace{00010111}^{\text{One's Complement}} \Rightarrow \overbrace{00011000}^{\text{Two's Complement}}$

P7.20 (c)*

$$\begin{array}{r@{\qquad\qquad}r} 33 & 00100001 \\ \underline{-37} & \underline{11011011} \\ -4 & 11111100 \end{array}$$

P7.23* If the variables in a logic expression are replaced by their inverses, the AND operation is replaced by OR, the OR operation is replaced by AND, and the entire expression is inverted, the resulting logic expression yields the same values as before the changes. In equation form, we have:

$$ABC = \overline{\overline{A}+\overline{B}+\overline{C}} \qquad\qquad (A+B+C) = \overline{\overline{A}\,\overline{B}\,\overline{C}}$$

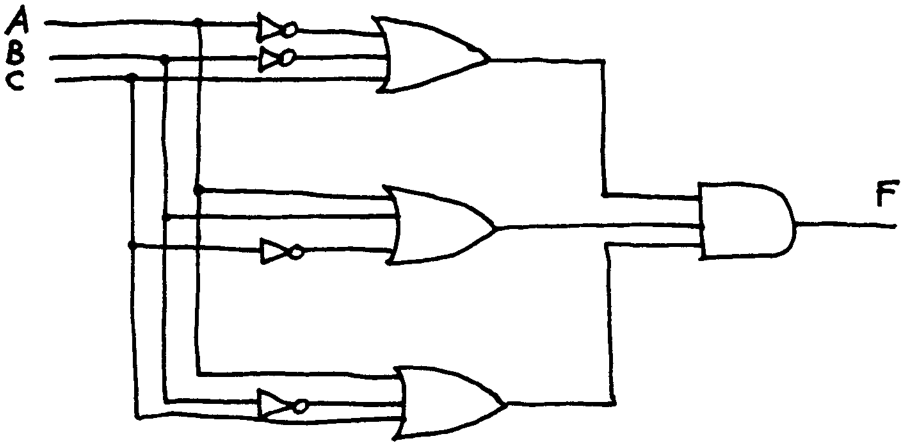
P7.26 (b)* $E = AB + A\overline{B}C + \overline{C}D$

A	B	C	D	E
0	0	0	0	0
0	0	0	1	1
0	0	1	0	0
0	0	1	1	0
0	1	0	0	0
0	1	0	1	1
0	1	1	0	0
0	1	1	1	0
1	0	0	0	0
1	0	0	1	1
1	0	1	0	1
1	0	1	1	1
1	1	0	0	1
1	1	0	1	1
1	1	1	0	1
1	1	1	1	1

P7.28*

A	B	C	$(A+B)(A+C)$	$A+BC$
0	0	0	0	0
0	0	1	0	0
0	1	0	0	0
0	1	1	1	1
1	0	0	1	1
1	0	1	1	1
1	1	0	1	1
1	1	1	1	1

P7.32 (c)* $F = \left(\overline{A}+\overline{B}+C\right)\left(A+B+\overline{C}\right)\left(A+\overline{B}+C\right)$



P7.33 (d)* $F = (A+B+C)(A+\overline{B}+C)(\overline{A}+B+\overline{C}) = \overline{\overline{A}\,\overline{B}\,\overline{C}+\overline{A}\,B\overline{C}+A\overline{B}\,C}$

(e)* $F = ABC + A\overline{B}\,C + \overline{A}\,B\overline{C} = \overline{(\overline{A}+\overline{B}+\overline{C})(\overline{A}+B+\overline{C})(A+\overline{B}+C)}$

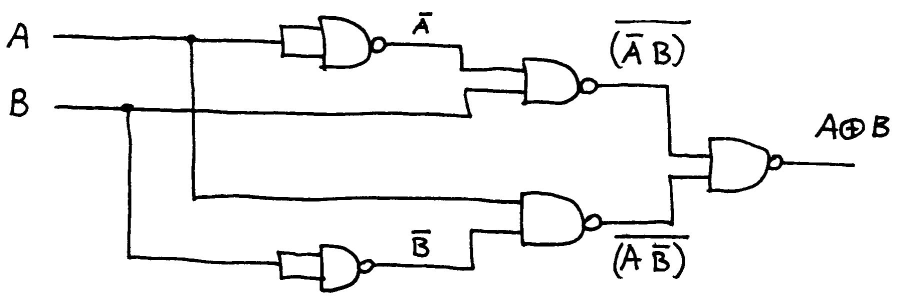
P7.40* $F = \overline{A}\,\overline{B}\,\overline{C} + \overline{A}B\overline{C} + A\overline{B}C + ABC$

$= \sum m(0,2,5,7)$

$F = \left(A+B+\overline{C}\right)\left(A+\overline{B}+\overline{C}\right)\left(\overline{A}+B+C\right)\left(\overline{A}+\overline{B}+C\right)$

$= \prod M(1,3,4,6)$

P7.49*

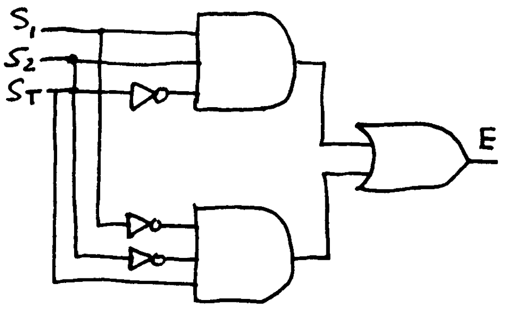


P7.52* (a)

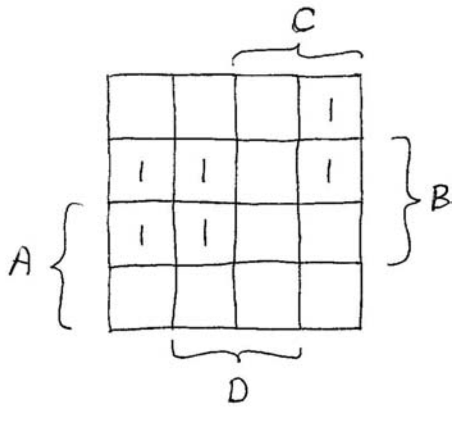
S_1	S_2	S_T	E
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	0

(b) $E = \sum m(1,6) = \overline{S_1}\,\overline{S_2}S_T + S_1S_2\overline{S_T}$

(c) Circuit diagram:



P7.53* (a) The Karnaugh map is:

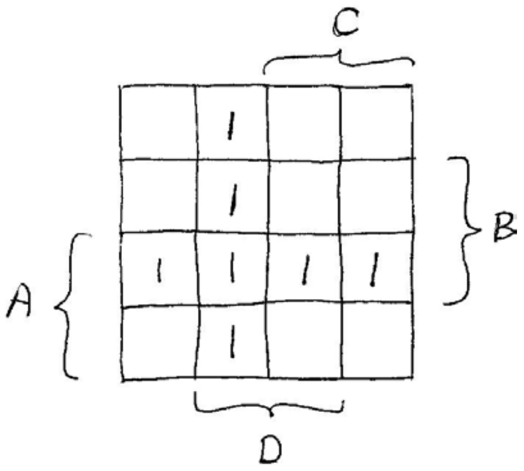


			1
1	1		1
1	1		

(b) $F = B\overline{C} + \overline{A}\,C\overline{D}$

(c) Inverting the map, and writing the minimum SOP expression yields $\overline{F} = AC + \overline{B}\,\overline{C} + CD$. Then applying DeMorgan's laws gives
$F = (\overline{A} + \overline{C})(B + C)(\overline{C} + \overline{D})$

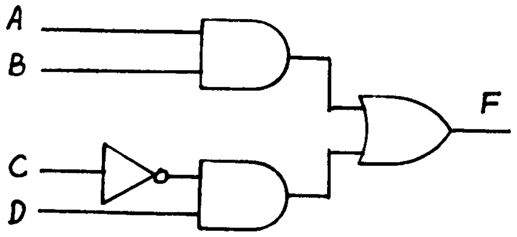
P7.58* (a) The Karnaugh map is:



	1		
	1		
1	1	1	1
	1		

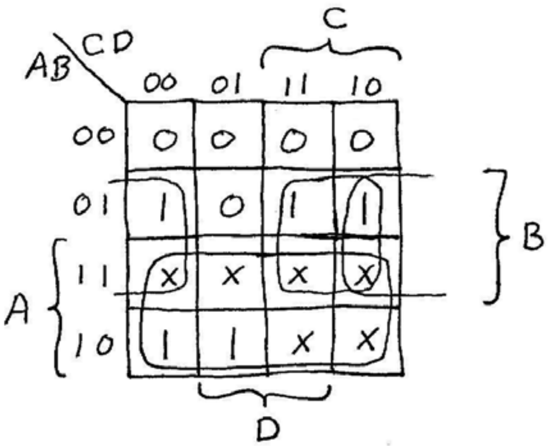
(b) $F = AB + \overline{C}D$

(c) The circuit is:



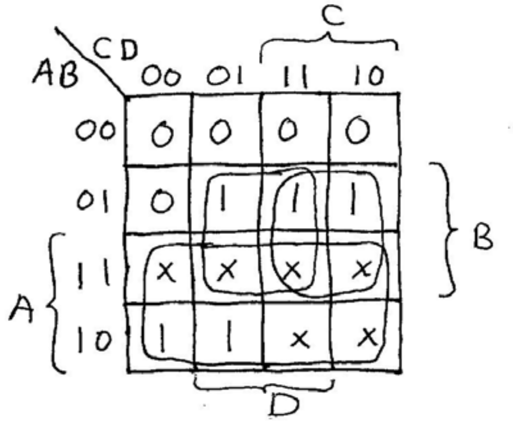
(d) Inverting the map, and writing the minimum SOP expression yields
$\overline{F} = \overline{A}\,C + \overline{A}\,\overline{D} + \overline{B}\,C + \overline{B}\,\overline{D}$. Then, applying DeMorgan's laws gives
$F = (A + \overline{C})(A + D)(B + \overline{C})(B + D)$

P7.71* (a) $F = A + BC + B\overline{D}$



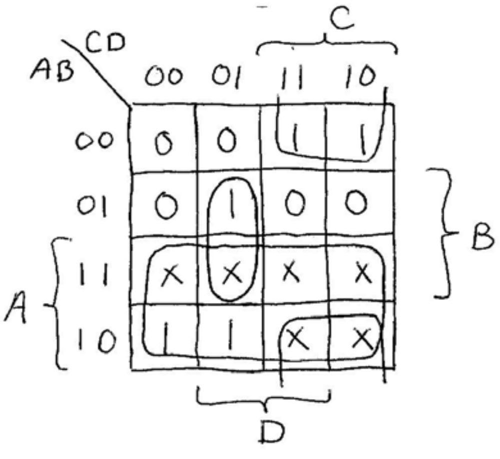
AB \ CD	00	01	11	10
00	0	0	0	0
01	1	0	1	1
11	X	X	X	X
10	1	1	X	X

(b) $G = A + BD + BC$



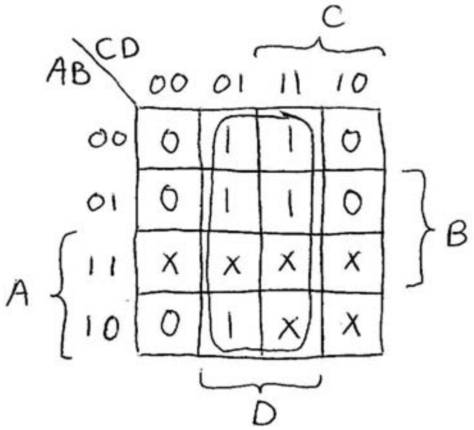
AB \ CD	00	01	11	10
00	0	0	0	0
01	0	1	1	1
11	X	X	X	X
10	1	1	X	X

(c) $H = A + \overline{B}C + B\overline{C}D$



AB \ CD	00	01	11	10
00	0	0	1	1
01	0	1	0	0
11	X	X	X	X
10	1	1	X	X

(d) $I = D$



AB \ CD	00	01	11	10
00	0	1	1	0
01	0	1	1	0
11	X	X	X	X
10	0	1	X	X

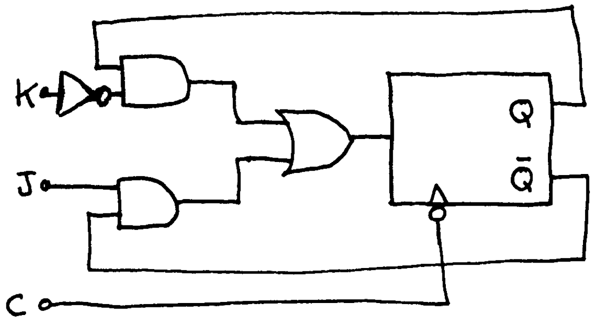
P7.81* The successive states are:

Q_0	Q_1	Q_2
1	0	0
0	1	0
1	0	1
1	1	0
1	1	1
0	1	1
0	0	1

(repeats)

Thus, the register returns to the initial state after seven shifts.

P7.85*



Practice Test

T7.1 (a) 12, (b) 19 (18 is incorrect because it omits the first step, inverting the variables), (c) 20, (d) 23, (e) 21, (f) 24, (g) 16, (h) 25, (i) 7, (j) 10, (k) 8, (l) 1 (the binary codes for hexadecimal symbols *A* through *F* do not occur in BCD).

T7.2 (a) For the whole part, we have:

	Quotient	Remainders
353/2	176	1
176/2	88	0
88/2	44	0
44/2	22	0
22/2	11	0
11/2	5	1
5/2	2	1
2/2	1	0
1/2	0	1

Reading the remainders in reverse order, we obtain:

$353_{10} = 101100001_2$

For the fractional part, we have

$2 \times 0.875 = 1 + 0.75$

$2 \times 0.75 = 1 + 0.5$

$2 \times 0.5 = 1 + 0$

Thus, we have

$0.875_{10} = 0.111_2$

Finally, combining the whole and fractional parts, we have

$353.875_{10} = 101100001.111_2$

(b) For the octal version, we form groups of three bits, working outward from the decimal point, and then write the octal symbol for each group.

$101\ 100\ 001.111_2 = 541.7_8$

(c) For the hexadecimal version, we form groups of four bits, working outward from the decimal point, and then write the hexadecimal symbol for each group.

$0001\ 0110\ 0001.1110_2 = 161.E_{16}$

(d) To obtain binary coded decimal, we simply write the binary equivalent for each decimal digit.

$$353.875_{10} = 0011\ 0101\ 0011.1000\ 0111\ 0101_{BCD}$$

T7.3 (a) Because the left-most bit is zero, this is a positive number. We simply convert from binary to decimal:

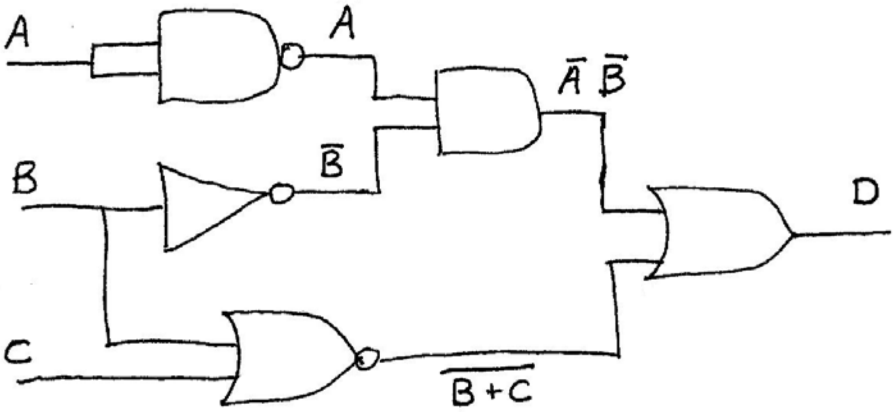
$$01100001_2 = 1 \times 2^6 + 1 \times 2^5 + 1 \times 2^0 = 64 + 32 + 1 = +97_{10}$$

(b) Because the left-most bit is one, this is a negative number. We form the two's complement, which is 01000110. Then, we convert from binary to decimal:

$$01000110_2 = 1 \times 2^6 + 1 \times 2^2 + 1 \times 2^1 \quad = 64 \ + 4 + 2 = +70_{10}$$

Thus, the decimal equivalent for eight-bit signed two's complement integer 10111010 is –70.

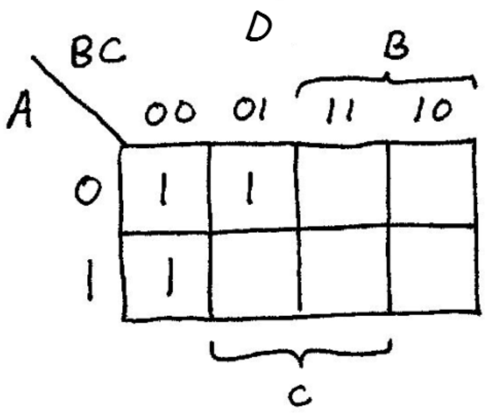
T7.4. (a) The logic expression is $D = \overline{A}\,\overline{B} + \overline{(B + C)}$.



(b) The truth table is:

A	*B*	*C*	*D*
0	0	0	1
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	0

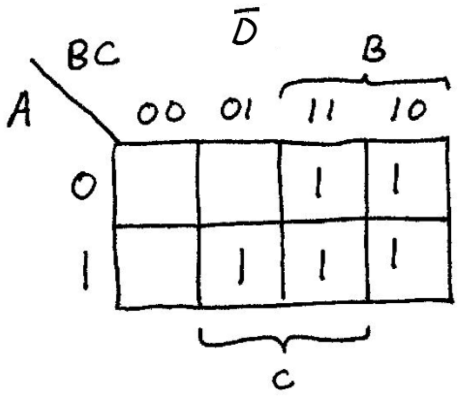
The Karnaugh map is:



A \ BC	00	01	11	10
0	1	1		
1	1			

(c) The map can be covered by two 2-cubes and the minimum SOP expression is $D = \overline{A}\,\overline{B} + \overline{B}\,\overline{C}$.

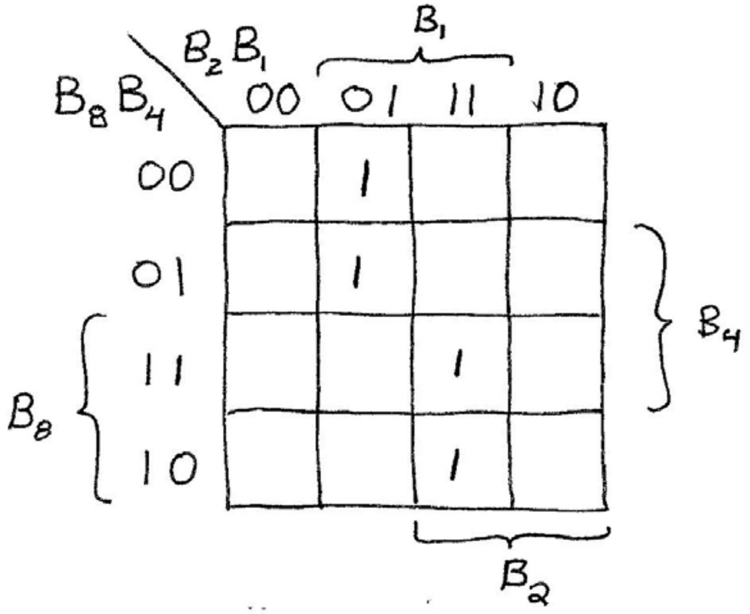
(d) First, we invert the map to find:



A \ BC	00	01	11	10
0			1	1
1		1	1	1

This map can be covered by one 4-cube and one 2-cube and the minimum SOP expression is $\overline{D} = B + AC$. Applying DeMorgan's laws to this yields the minimum POS expression $D = \overline{B}(\overline{A} + \overline{C})$.

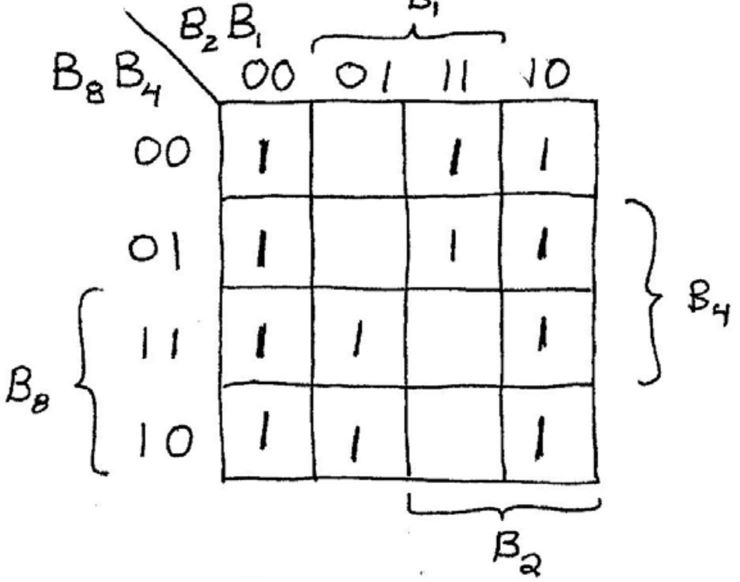
T7.5 (a) The completed Karnaugh map is:



B_8B_4 B_2B_1	00	01	11	10
00		1		
01		1		
11			1	
10			1	

(b) The map can be covered by two 2-cubes and the minimum SOP expression is $G = B_1\overline{B_2}\ \overline{B_8} + B_1B_2B_8$.

(c) First, we invert the map to find:



B_8B_4 B_2B_1	00	01	11	10
00	1		1	1
01	1		1	1
11	1	1		1
10	1	1		1

This map can be covered by one 8-cube and two 4-cubes and the minimum SOP expression is $\overline{G} = \overline{B_1} + B_2\overline{B_8} + \overline{B_2}B_8$. Applying DeMorgan's laws to this yields the minimum POS expression $G = B_1(\overline{B_2} + B_8)(B_2 + \overline{B_8})$.

T7.6 Clearly, the next value for Q_0 is the NAND combination of the current values of Q_1 and Q_2. The next value for Q_2 is the present value for Q_1. Similarly, the next value for Q_1 is the present value for Q_0. Thus, the successive states (Q_0 Q_1 Q_2) of the shift register are:

100 (initial state)
110
111
011
001
100
111

The state of the register returns to its initial state after 5 shifts.