[2] Brayton, R. K., G. D. Hachtel, C. T. McMullen, and A. L. Sangiovanni-Vincentelli: *Logic Minimization Algorithms for VLSI Synthesis*, Kluwer Academic, Boston, 1984.
[3] Hong, S. J., R. G. Cain, and D. L. Ostapko: "MINI: a heuristic approach for logic minimization," *IBM J. Research & Development*, vol. 18, pp. 443–458, September 1974.
[4] Karnaugh, M.: "The map method for synthesis of combinational logic circuits," *Trans. AIEE* part I, vol. 72, no. 9, pp. 593–599, 1953.
[5] McCluskey, E. J., Jr: "Minimization of Boolean functions," *Bell System Tech. J.*, vol. 35, no. 6, pp. 1417–1444, November 1956.
[6] McCluskey, E. J., and H. Schorr: "Essential multiple-output prime implicants," in *Mathematical Theory of Automata*, *Proc. Polytech. Inst. Brooklyn Symp.*, vol. 12, pp. 437–457, 1962.
[7] Muroga, S.: *Logic Design and Switching Theory*, John Wiley & Sons, New York, 1979.
[8] Quine, W. V.: "The problem of simplifying truth functions," *Am. Math. Monthly*, vol. 59, no. 8, pp. 521–531, October 1952.
[9] Quine, W. V.: "A way to simplify truth functions," *Am. Math. Monthly*, vol. 62, no. 9, pp. 627–631, November 1955.
[10] Veitch, E. W.: "A chart method for simplifying truth functions," in *Proc. ACM*, Pittsburgh, pp. 127–133, May 1952.

Problems

Problem 4.1. With the aid of a four-variable Karnaugh map, derive minimal sum-of-products expressions for each of the following functions:
(a) $f_1(w, x, y, z) = \sum(0, 1, 2, 3, 4, 6, 8, 9, 10, 11)$;
(b) $f_2(w, x, y, z) = \sum(0, 1, 5, 7, 8, 10, 14, 15)$;
(c) $f_3(w, x, y, z) = \sum(0, 2, 4, 5, 6, 8, 10, 12)$.

Problem 4.2
(a) Find the minimal sum-of-products and minimal product-of-sums expressions for

$$f(w, x, y, z) = \prod(1, 4, 5, 6, 11, 12, 13, 14, 15).$$

Is your answer unique?
(b) Determine the minimal sum-of-products expression for

$$f(w, x, y, z) = \sum(0, 2, 4, 9, 12, 15) + \sum_{\phi}(1, 5, 7, 10).$$

Problem 4.3. Given the function $T(w, x, y, z) = \sum(1, 2, 3, 5, 13) + \sum_{\phi}(6, 7, 8, 9, 11, 15)$:
(a) find a minimal sum-of-products expression;
(b) find a minimal product-of-sums expression;
(c) compare the expressions obtained in (a) and (b); if they do not represent identical functions, explain why.

Problem 4.4. Find all minimal four-variable functions that assume the value 1 when the minterms 4, 10, 11, 13 are equal to 1, and the value 0 when the minterms 1, 3, 6, 7, 8, 9, 12, 14 are equal to 1.

Problem 4.5. Each of the following functions actually represents a set of four functions, corresponding to the possible assignments of the don't-care terms.

$$f_1(w, x, y, z) = \sum(1, 3, 4, 5, 9, 10, 11) + \sum_{\phi}(6, 8),$$

$$f_2(w, x, y, z) = \sum(0, 2, 4, 7, 8, 15) + \sum_{\phi}(9, 12).$$

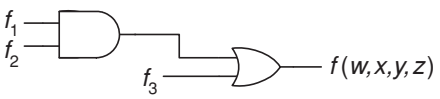
(a) Find $f_3 = f_1 \cdot f_2$. How many functions does f_3 represent?
(b) Find $f_4 = f_1 + f_2$. How many functions does f_4 represent?
(c) Simplify the above functions, their product, and their sum.

Problem 4.6. Let $f = \sum(5, 6, 13)$ and $f_1 = \sum(0, 1, 2, 3, 5, 6, 8, 9, 10, 11, 13)$. Find f_2 such that $f = f_1 \cdot f_2'$. Is f_2 unique? If not, indicate all possibilities.

Problem 4.7. Given the network of Fig. P4.7, determine the functions f_2 and f_3 if $f_1 = xz' + x'z$ and the overall transmission function is to be

$$f(w, x, y, z) = \sum(0, 4, 9, 10, 11, 12).$$

Fig. P4.7



Problem 4.8. A binary-coded-decimal (BCD) message appears in four input lines of a switching circuit. Design an AND, OR, NOT gate network that produces an output value 1 whenever the input combination is 0, 2, 3, 5, or 8.

Problem 4.9. Find the simplest function $g(A, B, C, D)$ that will make the function $f = A'BC + (AC + B)D + g(A, B, C, D)$ self-dual.
Hint: Determine first the properties of maps of self-dual functions.

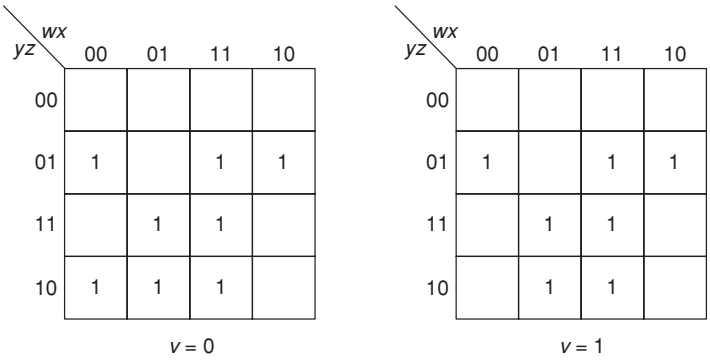
Problem 4.10. Use the map method to simplify each of the following functions:
(a) $f_1(v, w, x, y, z)$
$= \sum(3, 6, 7, 8, 10, 12, 14, 17, 19, 20, 21, 24, 25, 27, 28)$;
(b) $f_2(v, w, x, y, z)$
$= \sum(0, 1, 2, 4, 5, 9, 11, 13, 15, 16, 18, 22, 23, 26, 29, 30, 31)$.

Problem 4.11. The five-variable map can be constructed from two disjoint four-variable maps that correspond to the fifth variable and its complement, as shown in Fig. P4.11.
(a) Devise an algorithm that specifies the minimization procedure using such maps.
(b) Simplify the function

$$T(v, w, x, y, z) = \sum(1, 2, 6, 7, 9, 13, 14, 15, 17, 22, 23, 25, 29, 30, 31).$$

whose maps are given in Fig. P4.11.

Fig. P4.11



Problem 4.12. Construct a six-variable map and show the representation of

$$T(u, v, w, x, y, z) = u'w'y' + uwy + w'xy'z.$$

Problem 4.13. For the function $T(w, x, y, z) = \sum(0, 1, 2, 3, 4, 6, 7, 8, 9, 11, 15)$:
(a) Show the map;
(b) Find all prime implicants and indicate which are essential;
(c) Find a minimal expression for T and determine whether it is unique.

Problem 4.14. Given the function $T(w, x, y, z) = \sum(1, 3, 4, 5, 7, 8, 9, 11, 14, 15)$:
(a) use the map to obtain the set of all prime implicants and indicate specifically the essential ones;
(b) find three distinct minimal expressions for T;
(c) find the complement T' directly from the map;
(d) assume that only unprimed variables are available and construct a circuit that realizes T and requires no more than 13 gate inputs and two NOT gates.
Hint: Use the result obtained in part (c).

Problem 4.15. Show maps for four-variable functions with the following specifications. If this is impossible, explain why.
(a) A function with eight minterms for which
 (i) there are no essential prime implicants.
 (ii) all the prime implicants are essential.
(b) Repeat (a) for functions with nine minterms.
(c) A function with an even number of prime implicants, of which exactly half are essential.
(d) A function with six prime implicants, of which four are essential and two are covered by essential ones.

Problem 4.16. Prove or show a counterexample to each of the following statements.
(a) If a function f has a unique minimal sum-of-products expression then all its prime implicants are essential.
(b) If a function f has a unique minimal sum-of-products expression then it also has a unique minimal product-of-sums expression.

(c) If the pairwise product of all prime implicants of f is 0 then it has a unique minimal expression.
(d) For every prime implicant p that is not essential, there is an irredundant expression that *does not* contain p.
(e) If a function f does not have any essential prime implicant then it has at least two minimal sum-of-products forms.

Problem 4.17
(a) Give the map of an irreducible four-variable function whose sum-of-products representation consists of 2^3 minterms.
(b) Prove that there exists a function of n variables whose minimal sum-of-products form consists of 2^{n-1} minterms and that no function when expressed in sum-of-products form requires more than 2^{n-1} product terms.
(c) Derive a bound on the number of literals needed to express *any* n-variable function.

Problem 4.18
(a) Let $f(x_1, x_2, \ldots, x_n)$ be equal to 1 if and only if exactly k of the variables equal 1. How many prime implicants does this function have?
(b) Repeat (a) for the case where f assumes the value 1 if and only if k or more of the variables are equal to 1.
(*Note:* The above functions are known as *symmetric*.)

Problem 4.19
(a) Let $T(A, B, C, D) = A'BC + B'C'D$. Prove that *any* expression for T must contain at least one instance of the literal D *or* of the literal D'.
(b) If, in a minimal sum-of-products expression, each variable appears either in a primed form or in an unprimed form but not in both then the function is said to be *unate*. Prove that the minimal sum-of-products form of a unate function is unique.
(c) Is the converse true, i.e., if the minimal sum-of-products expression is unique then the function is unate?
Hint: The function $f = w'z + x'y + x'z$ is unate. If you relabel the variables, the function may be transformed into another function whose variables are all in an unprimed form.

Problem 4.20 Use the tabulation procedure to generate the set of prime implicants and to obtain *all* minimal expressions for the following functions:
(a) $f_1(w, x, y, z) = \sum(1, 5, 6, 12, 13, 14) + \sum_\phi(2, 4)$
(b) $f_2(v, w, x, y, z) = \sum(0, 1, 3, 8, 9, 13, 14, 15, 16, 17, 19, 24, 25, 27, 31)$
(c) $f_3(w, x, y, z) = \sum(0, 1, 4, 5, 6, 7, 9, 11, 15) + \sum_\phi(10, 14)$
(d) $f_4(v, w, x, y, z) = \sum(1, 5, 6, 7, 9, 13, 14, 15, 17, 18, 19, 21, 22, 23, 25, 29, 30)$
(e) $f_5(w, x, y, z) = \sum(0, 1, 5, 7, 8, 10, 14, 15)$

Problem 4.21 Apply the branching method to find a minimal expression for

$$f(v, w, x, y, z) = \sum(0, 4, 12, 16, 19, 24, 27, 28, 29, 31).$$

Problem 4.22
(a) Prove that if x and y are switching variables, then:
(i) $x + y = x \oplus y \oplus xy$;
(ii) $x' = x \oplus 1$.

(b) Using the equations in (a), any switching expression can be converted to an equivalent expression containing only the operations EXCLUSIVE OR and AND. Demonstrate the conversion procedure by transforming the expression

$$f = xyz' + xy'z + x'z.$$

(c) Derive a procedure to transform an expression containing the EXCLUSIVE-OR operation to an equivalent switching expression containing only AND, OR, and NOT operations. Apply your procedure to the expression

$$f = x \oplus y \oplus z.$$

Problem 4.23. Consider the minimization of modulo-2 sum-of-products expressions by means of a Karnaugh map. Since for every such expression the following are valid,

$$x \oplus x \oplus \cdots \oplus x = \begin{cases} 0 & \text{for an even number of } x\text{'s,} \\ x & \text{for an odd number of } x\text{'s,} \end{cases}$$
$$xy \oplus xy' = x,$$

then, when forming cubes, every 1-cell *must* be included in an *odd* number of cubes while any 0-cell *may* be included in selected cubes as long as it is included in an *even* number of such cubes. For example, the map for the function

$$f(x, y, z) = x'y'z' \oplus x'yz \oplus xy'z \oplus xyz'$$

is shown in Fig. P4.23. From the three cubes shown, it is evident that the minimal expression is

$$f = x \oplus y \oplus z'.$$

(a) Derive an algorithm for simplifying modulo-2 sum-of-products expressions by means of the map.[2]

(b) Apply your algorithm to simplify the following expressions:

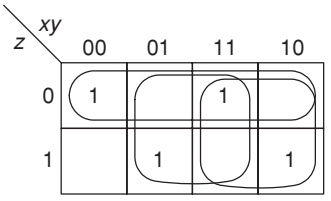
$$f_1(w, x, y, z) = w'xy'z' \oplus w'xyz' \oplus wx'y'z \oplus wx'yz \oplus wxy'z' \oplus wxy'z$$

(note that three terms containing seven literals constitute a minimum);

$$f_2(w, x, y, z) = w'x'yz \oplus w'xy'z \oplus w'xyz' \oplus wx'y'z \oplus wx'yz' \oplus wxy'z'$$

(note that five terms containing 14 literals constitute a minimum).

Fig. P4.23

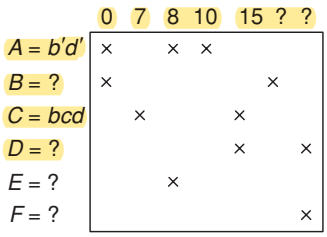


[2] For a reference, see Even, S., I. Kohavi, and A. Paz: "On minimal modulo 2 sums of products for switching functions," *IEEE Trans. Electron. Computers*, vol. EC-16, October 1967.

Problem 4.24. Shown in Fig. P4.24 is a prime implicant chart for $f(a, b, c, d)$ in which some of the row and column headings are unknown. It is known, however, that the chart has a row for each prime implicant of f and a column for each minterm for which f has a value 1.

(a) Find with the aid of a map all the minterms and prime implicants that correspond, respectively, to the columns and rows with unknown headings.
(b) Is your solution to (a) unique?
(c) Give the minterms for which f must be equal to 0.
(d) Find a minimal expression for f.

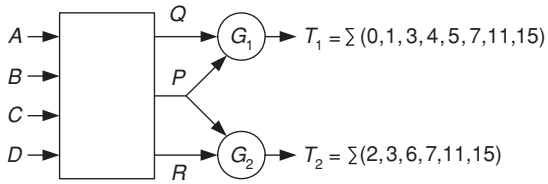
Fig. P4.24



Problem 4.25. A combinational network with four inputs A, B, C, and D, three intermediate outputs Q, P, and R, and final two outputs T_1 and T_2 is shown in Fig. P4.25.

(a) Assuming that G_1 and G_2 are both AND gates, show the map for the smallest function $P_{\min}$ (i.e., with the minimum number of minterms) that makes it possible to produce T_1 and T_2.
(b) Show the maps for Q and R that correspond to the above $P_{\min}$. Indicate explicitly the don't-care positions.
(c) Assuming that G_1 and G_2 are both OR gates, find the largest $P_{\max}$ and show the corresponding maps for Q and R.
(d) Can both T_1 and T_2 be produced if G_1 is an AND gate and G_2 is an OR gate? Or if G_1 is an OR gate and G_2 is an AND gate?

Fig. P4.25



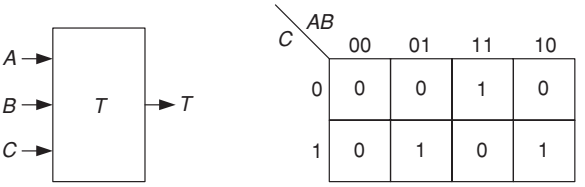
Problem 4.26. A gate T has logical properties that are defined by the map in Fig. P4.26.

(a) Prove that if the logic value 1 is given then any switching function can be realized by means of T gates, that is, T gates plus the logic value 1 are functionally complete.
(b) Realize, by means of two T gates, the function

$$f(w, x, y, z) = \sum(0, 1, 2, 4, 7, 8, 9, 10, 12, 15).$$

Hint: Realize the 0's of f.

Fig. P4.26



Problem 4.27. The initial covering of minterms for the function $f = \sum(0, 2, 3, 4, 5, 7)$ is shown on the left in Fig. P4.27. It needs to be converted into the covering shown on the right. Find a sequence of reduce, expand, and irredundant steps needed to do so. This sequence is not unique.

Fig. P4.27

x	y	z	f
0	–	0	1
–	1	1	1
1	0	–	1

$\overset{?}{\Longrightarrow}$

x	y	z	f
–	0	0	1
0	1	–	1
1	–	1	1

Problem 4.28. For the three functions shown below, obtain a multi-output minimized two-level implementation using an augmented prime implicant chart. Assume that minimizing the total number of gates is the sole objective.

$$f_1 = \sum(2, 3);$$

$$f_2 = \sum(2, 3, 4, 5, 6, 7);$$

$$f_3 = \sum(1, 3, 5, 7).$$

Problem 4.29. The initial covering of minterms for two functions, f_1 and f_2, is shown on the left in Fig. P4.29. It needs to be converted into the covering shown on the right. Find a sequence of reduce, expand, and irredundant steps that will achieve this.

Fig. P4.29

x	y	z	f_1	f_2
–	0	1	1	0
0	1	–	1	0
1	–	0	1	0
0	0	–	0	1
–	1	0	0	1
1	–	1	0	1

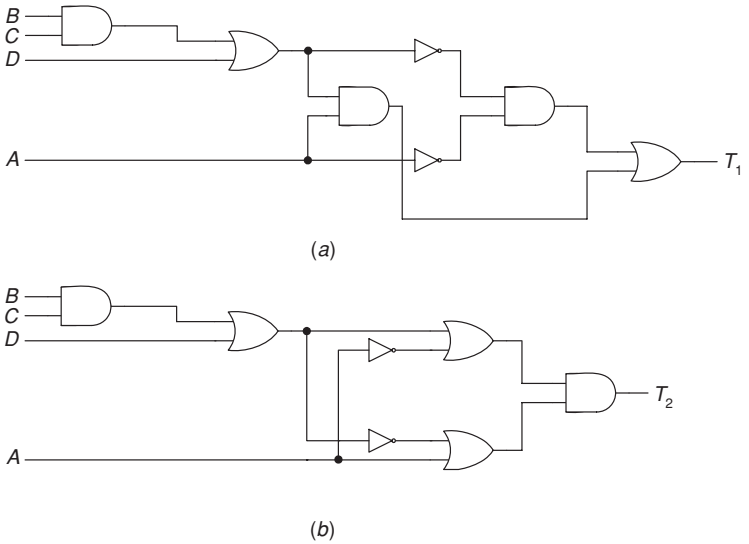
$\overset{?}{\Longrightarrow}$

x	y	z	f_1	f_2
0	–	1	1	0
1	0	–	1	0
–	1	0	1	1
0	0	–	0	1
1	–	1	0	1

Problems

Problem 5.1. Express T_1 and T_2 (see Fig. P5.1a, b) as functions of A, B, C, and D.

Fig. P5.1



Problem 5.2

(a) Design a two-level code converter from BCD to the *2-out-of-5 code* shown in Table P5.2a.

(b) Design a two-level code converter from the *Ringtail code* shown in Table P5.2b to BCD.

Table P5.2

Decimal	*2-out-of-5*				
0	1	1	0	0	0
1	0	0	0	1	1
2	0	0	1	0	1
3	0	0	1	1	0
4	0	1	0	0	1
5	0	1	0	1	0
6	0	1	1	0	0
7	1	0	0	0	1
8	1	0	0	1	0
9	1	0	1	0	0

(a)

Decimal	Ringtail				
0	0	0	0	0	0
1	0	0	0	0	1
2	0	0	0	1	1
3	0	0	1	1	1
4	0	1	1	1	1
5	1	1	1	1	1
6	1	1	1	1	0
7	1	1	1	0	0
8	1	1	0	0	0
9	1	0	0	0	0

(b)

Problem 5.3. Design a circuit with four inputs, x_1, x_2, x_3, x_4, and seven outputs, p_1, p_2, m_1, p_3, m_2, m_3, m_4, that receives BCD code words and generates the corresponding Hamming code words defined in Table 1.8.

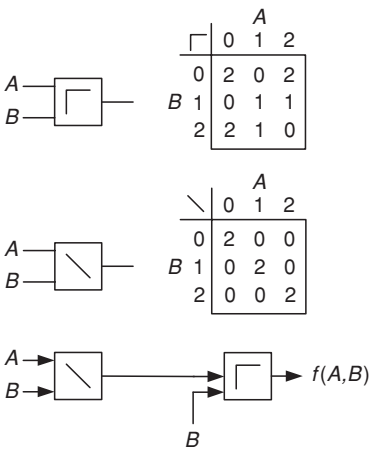
Problem 5.4. You are supplied with just one NOT gate and an unlimited amount of AND and OR gates and are required to design a circuit that realizes the expression

$$T(w, x, y, z) = w'x + x'y + xz'.$$

Only unprimed variables are available as inputs.
Hint: You may find the map of T helpful.

Problem 5.5. The tables shown in Fig. P5.5 define two devices whose inputs and outputs may assume any one of the *three* values 0, 1, or 2.

Fig. P5.5



Give the equivalent of a Karnaugh-map description of the function $f(A, B)$ that is realized by the network of Fig. P5.5.

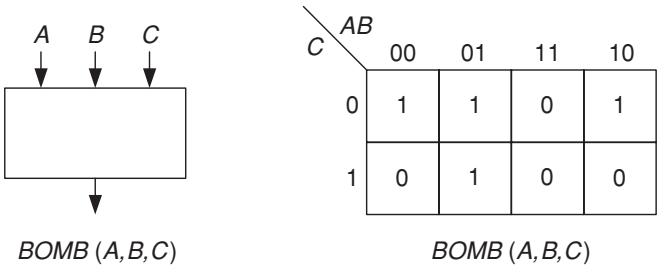
Problem 5.6. A certain four-input gate, called a LEMON gate, realizes the switching function $LEMON(A, B, C, D) = BC(A + D)$. Assume that the input variables are available in both primed and unprimed form.
(a) Show a realization of the function

$$f(w, x, y, z) = \sum(0, 1, 6, 9, 10, 11, 14, 15)$$

with only three LEMON gates and one OR gate.
(b) Can all switching functions be realized with LEMON and OR logic?
Hint: Draw the map for LEMON and utilize possible "patches" (coverings of the minterms of f with the LEMON function) on the map of f.

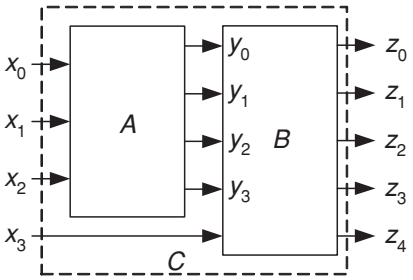
Problem 5.7. A three-input gate, BOMB, whose characteristics are shown in Fig. P5.7, has been mass-produced by an unfortunate company. Experimental evidence shows that input combinations 101 and 010 cause the gate to physically explode. Your task is to determine whether the gate is completely useless or can be externally modified such that it may be efficiently used to implement any switching function without causing explosions.

Fig. P5.7



Problem 5.8. A logic module A, shown in Fig. P5.8, operates as follows: output $y_i = 1$ iff i inputs out of x_0, x_1, x_2 are equal to 1. Design unit B in such a way that the overall logic function of unit C will be to produce an output $z_i = 1$ iff i inputs out of x_0, x_1, x_2, x_3 are equal to 1.

Fig. P5.8

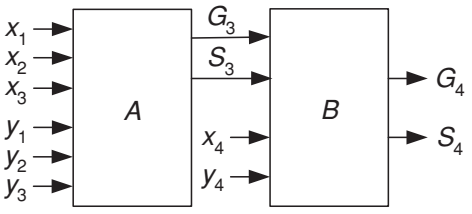


Problem 5.9. Given a logic module A that compares the magnitudes of two 3-bit numbers, $X_3 = x_1x_2x_3$ and $Y_3 = y_1y_2y_3$, where x_3 and y_3 are the least significant bits. Module A has two outputs G_3 and S_3, such that: $G_3 = 1$ iff $X_3 > Y_3$; $S_3 = 1$ iff $X_3 < Y_3$; and $G_3 = S_3 = 0$ iff $X_3 = Y_3$.

(a) Design a logic unit B such that together with module A it will serve as a comparator for two four-bit numbers, $X_4 = x_1x_2x_3x_4$ and $Y_4 = y_1y_2y_3y_4$, as shown in Fig. P5.9. Find expressions for G_4 and S_4 in terms of the inputs to unit B and show a realization of these expressions using only NAND gates.

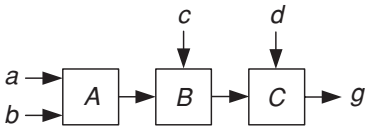
(b) Show a realization of module A by means of only units of type B. Assume that the constants 0 and 1 are available.

Fig. P5.9



Problem 5.10. Given a function $g(x_1, x_2, x_3, x_4) = \sum(4, 6, 7, 15) + \sum_{\phi}(2, 3, 5, 11)$, realize g in the form shown in Fig. P5.10, i.e., find the correspondence between the x_i and a, b, c, d, and determine the functions A, B, and C.

Fig. P5.10



Problem 5.11. A *half adder* is a device capable of performing the addition of two bits. It has two binary inputs, A and B, and two outputs, S and C_0. (Note that there is no carry into the half adder.)

(a) Write truth tables that define the half adder and derive logic expressions for S and C_0.

(b) Assuming that only uncomplemented inputs are available, show an implementation of the half adder that requires only three two-input AND or OR gates and one NOT gate.

(c) Under the above assumption, design the half adder using no more than five NAND gates or NOR gates, but not both together.

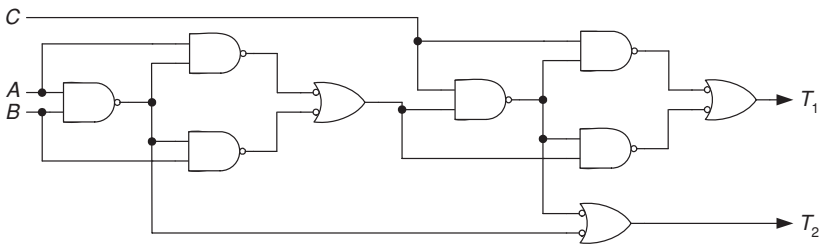
Problem 5.12. Construct a full adder using only two half adders and one OR gate.

Problem 5.13. A *half subtractor* is a device capable of subtracting one binary digit from the other. Show a realization of the half subtractor using AND, OR, NOT logic.

Problem 5.14. Define a *full subtractor*, show its truth tables, and derive logic expressions for difference (D) and borrow (B) outputs.

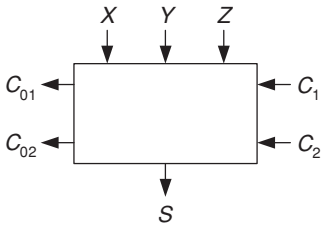
Problem 5.15. Analyze the two-output circuit shown in Fig. P5.15. Indicate the logic expression associated with every gate output.

Fig. P5.15



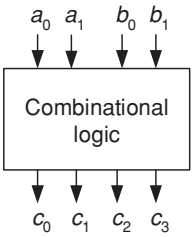
Problem 5.16. Design a device capable of adding *three* binary digits simultaneously. The device has five inputs, as shown in Fig. P5.16; X, Y, and Z are the arguments, C_1 is the carry-in from the preceding stage, and C_2 is the carry-in from the next-to-the-preceding stage. The output S designates the sum, while C_{01} and C_{02} designate the carry-outs to the succeeding stage and to the next-to-the-succeeding stage, respectively. Express explicitly the sum and carry-out functions and show a circuit diagram.

Fig. P5.16



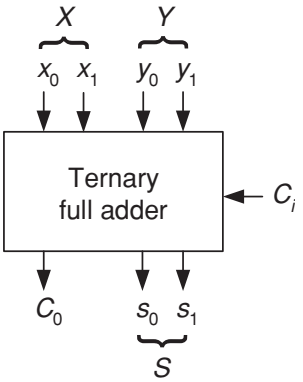
Problem 5.17. The schematic diagram in Fig. P5.17 shows a multiplier capable of multiplying two two-digit binary numbers. The digits of the two numbers are designated a_0 and a_1, b_0 and b_1, while c_0, c_1, c_2, and c_3 designate the digits of the product. Design the combinational logic.

Fig. P5.17



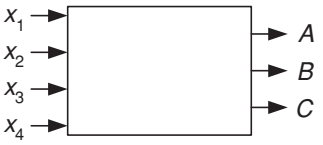
Problem 5.18. The schematic diagram shown in Fig. P5.18 shows a *ternary full adder* that receives two ternary digits X and Y plus a carry-in C_i and produces their sum S in base 3 plus a carry-out C_0. The ternary digits are coded in binary, that is, each of the three ternary digits 0, 1, 2 is coded by two binary digits: 0 by 00, 1 by 01, and 2 by 10. Thus, for example, if X and Y are each equal to 2 in base 3 and C_i equals 1 then the ternary full adder is required to perform the ternary addition of $(2)_3 + (2)_3 + (1)_3 = (12)_3$. Accordingly, the sum S must be 2 while the carry-out must be 1. Design the circuit assuming that you have as many gates as necessary as well as binary half and full adders.

Fig. P5.18



Problem 5.19. A communication system is designed to transmit just two code words, $A = 0010$ and $B = 1101$. However, owing to noise in the system, the received word may have as many as two errors. Design a combinational circuit that receives the words and that can correct one error and detect the existence of two errors. Specifically, design the circuit in Fig. P5.19 in such a way that output A will be equal to 1 if the received word is A, output B will be equal to 1 if the received word is B, and output C will be equal to 1 if the word received has two errors and thus cannot be corrected.

Fig. P5.19



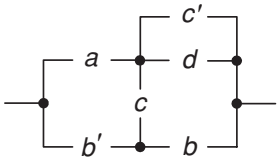
Problem 5.20

(a) Find all cut and tie sets for the circuit shown in Fig. P5.20. What function T is realized by this circuit?

(b) Prove that any network realization of T must contain at least one branch d. Generalize your arguments to determine the necessity of branches for other literals.

(c) Find a minimum-branch series–parallel network for T.

Fig. P5.20

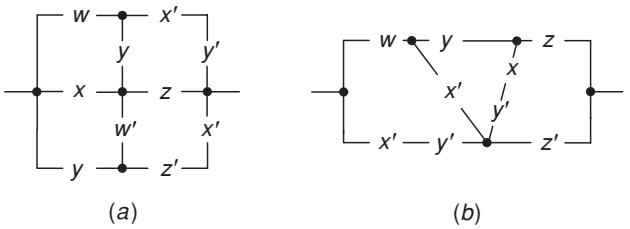


Problem 5.21

(a) Find a minimal network equivalent to that shown in Fig. P5.21a. It requires only five branches.

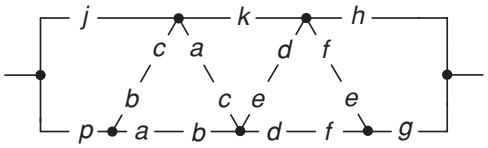
(b) Find a minimal complex CMOS gate which realizes a function that is the same as the transmission function realized by the network in Fig. P5.21b. It requires only 14 transistors.

Fig. P5.21



Problem 5.22. For the network of Fig. P5.22, find an equivalent network with only 11 branches.

Fig. P5.22



Problem 5.23. Design a minimal complementary-switch-based CMOS implementation that can turn a lamp on or off from three different locations independently. Denote the switches as x, y, and z.

Problem 5.24. For each of the following functions, find a network realization that requires as few branches as possible:

(a) $T(w, x, y, z) = \sum(0, 4, 6, 8, 9, 12)$;

(b) $T(w, x, y, z) = \sum(3, 7, 8, 9, 13)$;

(c) $T(w, x, y, z) = \sum(5, 6, 7, 9, 10, 11, 13, 14)$;

(d) $T(w, x, y, z) = \sum(5, 6, 9, 10, 11, 12, 13, 14, 15)$;

(e) $T(w, x, y, z) = \sum(5, 6, 7, 9, 10, 11, 12)$.

Problem 5.25. In a meeting of a board of directors, four resolutions, A, B, C, and D, are to be put to the vote. The decisions are complicated, however, by the fact that the resolutions are not mutually independent. In fact, voting must be governed by the following rules.

1. Those who vote for resolution B must also vote for resolution C.
2. It is possible to vote for both resolutions A and C only if a vote for either B or D is also cast.
3. Those who vote for either resolution C or D or vote against resolution A must vote for resolution B.

Each member of the board has four switches, A, B, C, and D, which he presses or releases, depending on whether he is in favor of or against the resolution under consideration. The switches of each member are inputs to a complex CMOS gate associated with that member. The gate produces a red signal at the end of the vote if the member *did not* vote according to the rules. Design such a gate with as few transistors as possible.

Problem 5.26. Four people, w, x, y, and z, own a company. Their shares in the company are: w, 40%; x, 30%; y, 20%; z, 10%. A 60% majority of the shares is required to pass a resolution. Around their conference table are mounted four buttons, w, x, y, and z. Each person presses his button to vote in favor of, or releases it to oppose, the resolution under consideration. Design a complex CMOS gate whose output gives a signal whenever a resolution is passed.

Problem 5.27. For $f = w'x' + w'v'z' + v'x'y' + y'z'$, derive a static CMOS complex gate that has a total of only 10 transistors.

Hint: Both nMOS and pMOS networks would need to be nonseries–parallel.