Problem Set 5

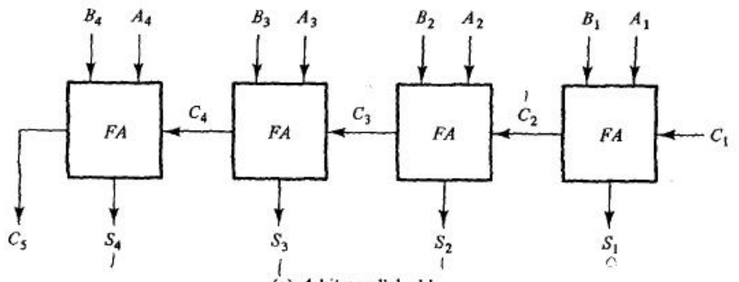
4-28 Design a combinational circuit that converts a 4-bit Gray code number (Table 1-4) to a 4-bit straight binary number. Implement the circuit with exclusive-OR gates.

TABLE 1-4
Four-bit Gray code

Gray code	Decimal equivalent
0000	0
0001	1
0011	2
0010	3
0110	4
0111	5
0101	6
0100	7
1100	8
1101	9
1111	10
1110	11
1010	12
1011	13
1001	14
1000	15

4-7 Design a combinational circuit that multiplies two 2-bit numbers, a_1a_0 and b_1b_0, to produce a 4-bit product, $c_3c_2c_1c_0$. Use AND gates and half-adders.

3: Using two four-bit adder chips, and other gates, develop a circuit for adding two BCD digits plus carry-in. Hint: If the sum is over 9 it is necessary to subtract 10 to get the BCD digit. (The circuit for a four-bit adder is shown in Figure 5-2a, page 156 of Mano.)



(a) 4-bit parallel adder

4: Develop a circuit for generating decimal "carry generate" and "carry propagate" signals for a BCD adder. Hint: Use a four-bit adder chip, and think about what output conditions represent "carry generate" and "carry propagate."

5-15 A combinational circuit is defined by the following three Boolean functions. Design the circuit with a decoder and external gates.

$$F_1 = x'y'z' + xz$$

$$F_2 = xy'z' + x'y$$

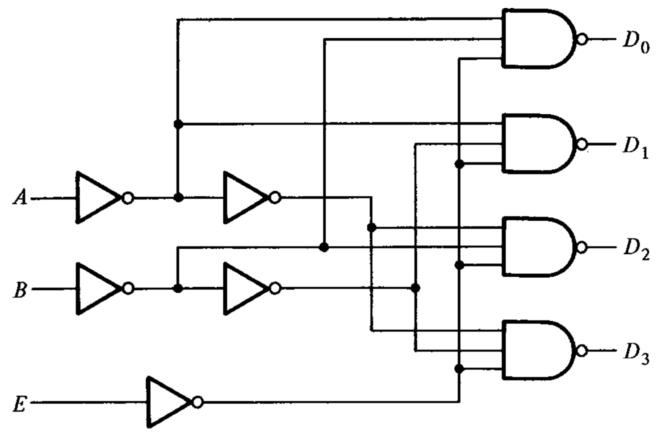
$$F_3 = x'y'z + xy$$

5-16 A combinational circuit is specified by the following three Boolean functions. Implement the circuit with a 3×8 decoder constructed with NAND gates (similar to Fig. 5-10) and three external NAND or AND gates. Use a block diagram for the decoder. Minimize the number of inputs in the external gates.

$$F_1(A, B, C) = \Sigma(2, 4, 7)$$

$$F_2(A, B, C) = \Sigma(0, 3)$$

$$F_3(A, B, C) = \Sigma(0, 2, 3, 4, 7)$$



(a) Logic diagram

E	A	B	D_0	D_1	D_2	D_3
1	X	X	1	1	1	1
0	0	0	0	1	1	1
0	0	1	1	0	1	1
0	1	0	1	1	0	1
0	1	1	1	1	1	0

(b) Truth table

FIGURE 5-10
A 2-to-4-line decoder with enable (E) input

5-18 Construct a 5 × 32 decoder with four 3 × 8 decoders with enable and one 2 × 4 decoder. Use block diagrams similar to Fig. 5-12.

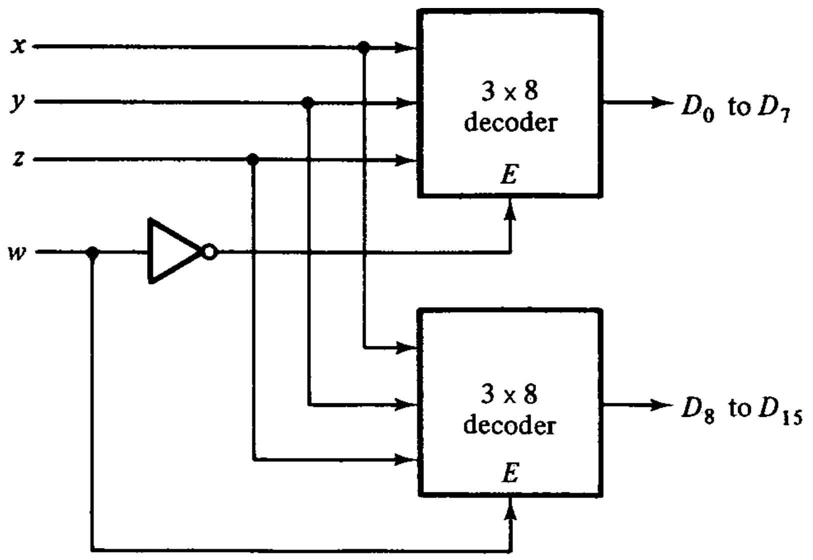


FIGURE 5-12
A 4 × 16 decoder constructed with two 3 × 8 decoders

5-24 Implement the following Boolean function with an 8 × 1 multiplexer.

$$F(A, B, C, D) = \Sigma(0, 3, 5, 6, 8, 9, 14, 15)$$

5-26 Implement the Boolean function of Example 5-2 with an 8 × 1 multiplexer, but with inputs A, B, and C connected to selection inputs s_2, s_1, and s_0, respectively.

$$F(A, B, C, D) = \Sigma(0, 1, 3, 4, 8, 9, 15)$$

5-27 An 8 × 1 multiplexer has inputs A, B, and C connected to the selection inputs s_2, s_1, and s_0, respectively. The data inputs, I_0 through I_7, are as follows: $I_1 = I_2 = I_7 = 0$; $I_3 = I_5 = 1$; $I_0 = I_4 = D$; and $I_6 = D'$. Determine the Boolean function that the multiplexer implements.

5-28 Implement the following Boolean function with a 4 × 1 multiplexer and external gates. Connect inputs A and B to the selection lines. The input requirements for the four data lines will be a function of variables C and D. These values are obtained by expressing F as a function of C and D for each of the four cases when AB = 00, 01, 10, and 11. These functions may have to be implemented with external gates.

$$F(A, B, C, D) = \Sigma(1, 3, 4, 11, 12, 13, 14, 15)$$

5-32 Tabulate the truth table for an 8 × 4 ROM that implements the following four Boolean functions:

$$A(x, y, z) = \Sigma(1, 2, 4, 6,)$$
$$B(x, y, z) = \Sigma(0, 1, 6, 7)$$
$$C(x, y, z) = \Sigma(2, 6)$$
$$D(x, y, z) = \Sigma(1, 2, 3, 5, 7)$$

5-33 Tabulate the PLA programming table for the four Boolean functions listed in Problem 5-32. Minimize the number of product terms.

5-34 Derive the PLA programming table for the combinational circuit that squares a 3-bit number. Minimize the number of product terms.

9: Design a code converter that converts 84-2-1 code to BCD

(a) using a four input decoder,

(b) using a PLA, and

(c) using only a four-bit binary adder chip.

In all cases you may assume that only valid inputs occur.
In (b) make full use of the resulting "don't care" conditions.