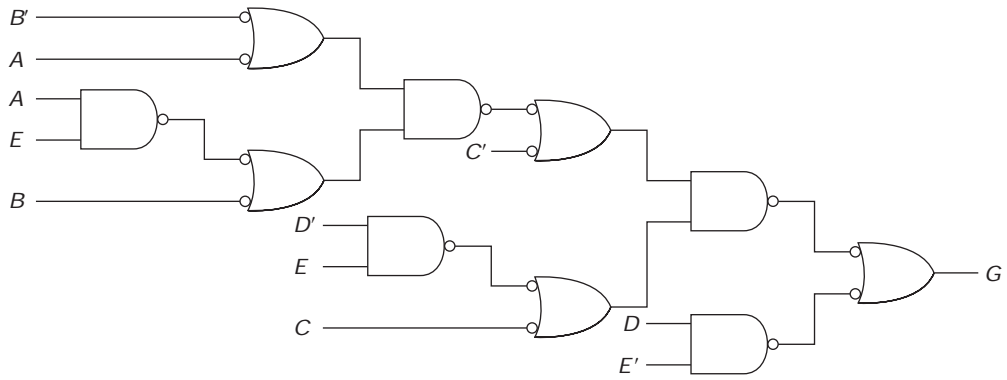
5.10 EXERCISES

1. For the following circuit:



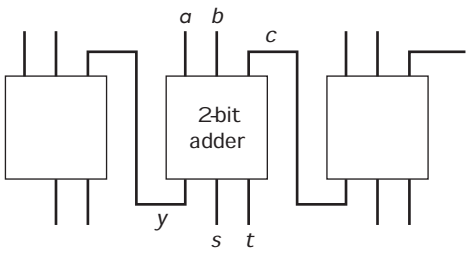
a. Compute the maximum delay,
   i. Assuming that all inputs are available both uncomplemented and complemented
   ii. Assuming only uncomplemented inputs are available and an additional gate must be added to complement each input

b. Compute the maximum delay from input C to the output, assuming that all inputs are available both uncomplemented and complemented.

***2.** We are building an adder to add the 32-bit constant

10101010101010101010101010101010

to an arbitrary 32-bit number. We will implement this with 16 identical adder modules, each of which will add 2 bits of the number to the constant (10) and a carry from the next lower pair of bits and produce 2 bits of the sum and the carry to the next bits. A block diagram of part of this is shown below:



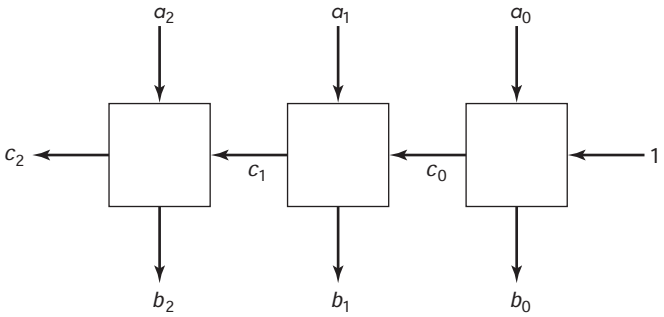
The problem each 2-bit adder solves is

$$\begin{array}{ccc} & & c \\ & a & b \\ & 1 & 0 \\ \hline y & s & t \end{array}$$

a. Show a truth table for that 2-bit adder (it has three inputs, a, b, and c, and it has three outputs, y, s, and t), and find minimum SOP expressions for each output.

b. Compute the delay from the c input of each module to the y output of that module and the total delay for the 32 bits.

3. We want to build a circuit to compute the two's complement of an n-bit number. We will do this with n modules, each of which complements that bit and then adds the carry from the next lower bit. Thus, the first three bits of a block diagram of the circuit will look like



a. Show a block diagram for each of the boxes using NAND gates. (Design the first—on the right—box specially.)

b. Compute the delay for n bits.

c. Improve the speed by designing 2 bits at a time. Show a NAND gate circuit and compute the total delay.

4. We want to build an adder to simultaneously add three multidigit binary numbers. Design a single bit of that adder. It has three inputs for that digit, x, y, and z, plus two carry inputs, u and v (since you may have a carry of 0, 1, or 2). There are three outputs, a sum, s, and two carries, f and g. Show a truth table and find the minimum sum of products expressions for the three outputs.

5. Design a circuit to multiply two 2-bit numbers—a, b and c, d and produce a 4-bit product—w, x, y, z. Show a truth table and the equations.

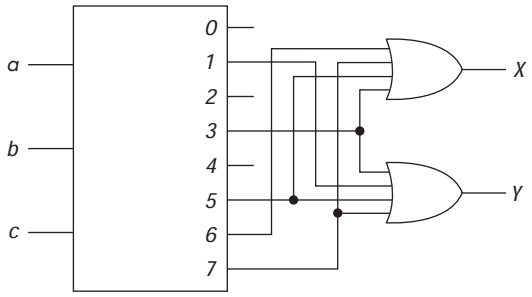
6. We need to determine whether a three-bit number, a_3, a_2, a_1, is equal to another number, b_3, b_2, b_1, or if it is greater than that number. (We do not need an output for less than.)

a. Show how the 7485 would be connected to accomplish this.

b. Implement this with AND and OR gates.

c. Assuming that the 7485 costs $1, what must 7400 series AND and OR gate packages cost to make the AND/OR implementation less expensive?

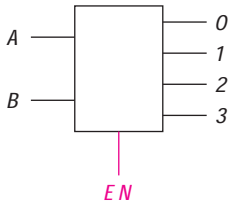
***7.** Consider the following circuit with an active high output decoder. Draw a truth table for *X* and *Y* in terms of *a*, *b*, and *c*.



8. We wish to design a decoder, with three inputs, *x*, *y*, *z*, and eight active high outputs, labeled *0*, *1*, *2*, *3*, *4*, *5*, *6*, *7*. There is no enable input required. (For example, if *xyz* = 011, then output *3* would be 1 and all other outputs would be 0.)

The **only** building block is a two-input, four-output decoder (with an active high enable), the truth table for which is shown below.

EN	*A*	*B*	*0*	*1*	*2*	*3*
0	X	X	0	0	0	0
1	0	0	1	0	0	0
1	0	1	0	1	0	0
1	1	0	0	0	1	0
1	1	1	0	0	0	1



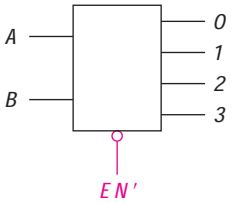
Draw a block diagram of the system using as many of these building blocks as are needed.

***9.** We want to implement a full adder; we'll call the inputs *a*, *b*, and *c* and the outputs *s* and c_{out}. As always, the adder is described by the following equations:

$$s(a, b, c) = \Sigma m(1, 2, 4, 7)$$
$$c_{out}(a, b, c) = \Sigma m(3, 5, 6, 7)$$

To implement this, all we have available are two decoders (as shown below) and two OR gates. Inputs *a* and *b* are available both uncomplemented and complemented; *c* is available only

uncomplemented. Show a block diagram for this system. Be sure to label all of the inputs to the decoders.



EN	A	B	0	1	2	3
1	X	X	0	0	0	0
0	0	0	1	0	0	0
0	0	1	0	1	0	0
0	1	0	0	0	1	0
0	1	1	0	0	0	1

10. Show the block diagram for a decoder, the truth table for which is shown below. The available components are one-, two-, and three-input NAND gates. (A one-input NAND is an inverter.)

Inputs				Outputs		
E1	E2	a	b	1	2	3
0	X	X	X	1	1	1
X	0	X	X	1	1	1
1	1	0	0	1	1	1
1	1	0	1	0	1	1
1	1	1	0	1	0	1
1	1	1	1	1	1	0

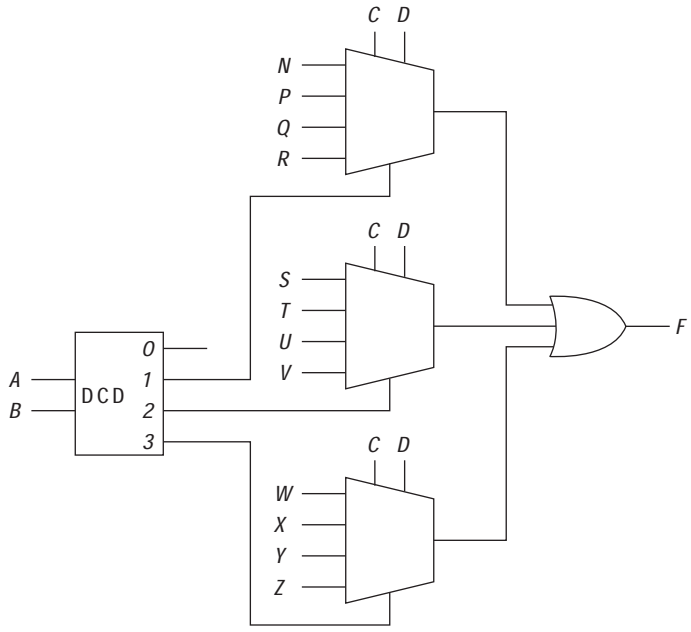
11. Design, using AND, OR, and NOT gates, a priority encoder with seven active low inputs, $1', \ldots, 7'$ and three active high outputs, CBA that indicate which is the highest priority line active. Input $1'$ is highest priority; $7'$ is lowest. If none of the inputs are active, the output is 000. There is a fourth output line, M, which is 1 if there are multiple active inputs.

***12.** Implement the function

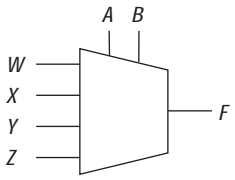
$$f(x, y, z) = \Sigma m(0, 1, 3, 4, 7)$$

using two-way multiplexers.

13. In the following circuit, the decoder (DCD) has two inputs and four (active high) outputs (such that, for example, output 0 is 1 if and only if inputs A and B are both 0). The three multiplexers each have two select inputs (shown on the top of the box), four data inputs (shown on the left) and an active high enable input (shown on the bottom). Inputs A, B, C, and D are select inputs; inputs N through Z are data inputs. Complete a truth giving the value of F for each of the 16 possible select input combinations. (Comment: For some values, $F = 0$, for one value, $F = W$.)



14. The following circuit includes a multiplexer with select inputs, A and B, and data inputs, W, X, Y, and Z:



Write an algebraic equation for F.

15. For the following sets of functions, design a system

i. Using a ROM
ii. Using a PLA with the number of product terms shown
iii. Using a PAL

a. $F(A, B, C) = \Sigma m(3, 4, 5, 7)$
$G(A, B, C) = \Sigma m(1, 3, 5, 6, 7)$
$H(A, B, C) = \Sigma m(1, 4, 5)$ (4 product terms)

b. $W(A, B, C) = \Sigma m(0, 1, 4)$
$X(A, B, C) = \Sigma m(0, 3, 4, 7)$
$Y(A, B, C) = \Sigma m(1, 2, 6)$
$Z(A, B, C) = \Sigma m(2, 3, 6, 7)$ (4 product terms)

c. $f(a, b, c, d) = \Sigma m(3, 5, 6, 7, 8, 11, 13, 14, 15)$
$g(a, b, c, d) = \Sigma m(0, 1, 5, 6, 8, 9, 11, 13, 14)$
(6 product terms)

d. $F(A, B, C, D) = \Sigma m(1, 2, 6, 7, 8, 9, 12, 13)$
$G(A, B, C, D) = \Sigma m(1, 8, 9, 10, 11, 13, 15)$
$H(A, B, C, D) = \Sigma m(1, 6, 7, 8, 11, 12, 14, 15)$
(8 product terms)

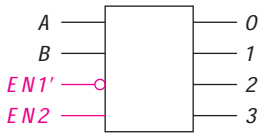
16. We have found a minimum sum of products expression for each of two functions, F and G, minimizing them individually (no sharing):

$$F = WX'Y' + XY'Z + W'Z$$
$$G = WY'Z + X'Y'$$

a. Implement them with a ROM.

b. Implement them with a PLA with four terms.

c. For the same functions, we have available as many of the decoders described below as are needed plus 2 eight-input OR gates. Show a block diagram for this implementation. All inputs are available both uncomplemented and complemented.



EN1	*EN2*	*A*	*B*	*0*	*1*	*2*	*3*
X	0	X	X	0	0	0	0
1	X	X	X	0	0	0	0
0	1	0	0	1	0	0	0
0	1	0	1	0	1	0	0
0	1	1	0	0	0	1	0
0	1	1	1	0	0	0	1

Note that this chip is enabled only when $EN1' = 0$ and $EN2 = 1$.

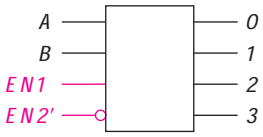
17. Consider the following three functions, f, g, and h of the four variables, a, b, c, and d, whose minimum solutions (treating each as a separate problem) are listed below. Throughout, all variables are available *only uncomplemented*:

$$f = b'c'd' + bd + a'cd$$
$$g = c'd' + bc' + bd' + a'b'cd$$
$$h = bd' + cd + ab'd$$

a. Implement them with a ROM.

b. Implement them on a PLA with six terms.

c. Implement them using only decoders of the type shown below (as many as needed) and three OR gates (each with as many inputs as you need). (No other gates are allowed.) Logic 0 and logic 1 are available.



EN1	EN2	A	B	0	1	2	3
0	X	X	X	0	0	0	0
X	1	X	X	0	0	0	0
0	1	0	0	1	0	0	0
0	1	0	1	0	1	0	0
0	1	1	0	0	0	1	0
0	1	1	1	0	0	0	1

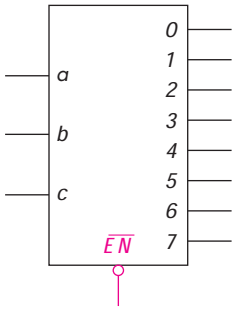
***18.** We have three functions, X, Y, Z of the four variables, A, B, C, D. Note: Each part can be solved without the other:

$$X(A, B, C, D) = \Sigma m(0, 2, 6, 7, 10, 13, 14, 15)$$
$$Y(A, B, C, D) = \Sigma m(2, 6, 7, 8, 10, 12, 13, 15)$$
$$Z(A, B, C, D) = \Sigma m(0, 6, 8, 10, 13, 14, 15)$$

a. Implement with a two-level NAND gate circuit. This can be done using only prime implicants of the individual functions with 13 gates. With sharing, it can be done with 10 gates. Assume that all variables are available both complemented and uncomplemented.

b. Implement these functions using a ROM.

c. Implement this with 2 three-input (plus active low enable) decoders as shown below, plus a minimum number of AND, OR, and NOT gates.



d. Implement it with a PLA with eight terms. (You may not need to use all of them.)

e. Implement them with the PAL shown in the text.

19. Implement the 2-bit adder of Section 5.1.2 using the PAL of Section 5.6.3. The problem is that one of the output functions requires 7 terms and another 12. This can be overcome by building the carry between the 2 bits and using that output as another input to compute s_1 and c_{out}.

20. In Solved Problem 16, we designed a converter from excess 3 to 2 of 5 code. In this exercise, we want to do the reverse, that is

design a converter from 2 of 5 code to excess 3. There will be four functions of five variables. We will assume that only legitimate digit codes are input; thus, there will be 22 don't cares on each map. All inputs are available both uncomplemented and complemented.

a. Map each of the four functions and find all minimum sum of products and product of sums solutions for each of the four functions individually.

b. Our building blocks consist of integrated circuit chips. We can buy any of the following chips:

7404:	6 inverters		
7400:	4 two-input NAND gates	7402:	4 two-input NOR gates
7410:	3 three-input NAND gates	7427:	3 three-input NOR gates
7420:	2 four-input NAND gates	7425:	2 four-input NOR gates

All chips cost the same, 25¢ each.

Find one of the least expensive ($1.00) implementations of the four outputs. (The gates on any chip may be used as part of the implementation of more than one of the outputs.) Show the algebraic expression and the block diagram for the solution.

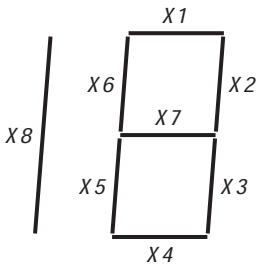
c. Find two solutions, one of which uses only 7400 and 7410 packages, and a solution that uses only NOR gates. Each of these must cost no more than $1.00. (Of course, one of these is the solution to part b.)

d. Implement this with a ROM.

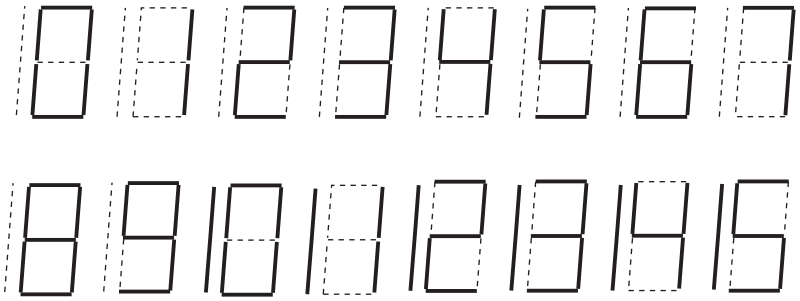
e. Implement this with a PLA.

f. Implement this with the PAL described in the text.

***21.** We have a special eight-segment display, as shown below.



We want to display the numbers from 0 to 15, as shown on the next figure, where a dashed line means an unlit segment and a solid line a lit one. Note that for 6 and 9, one segment each may be lit or unlit, as you wish.



Design three versions of a system that accepts as an input a 4-bit number, A, B, C, D and produces the eight outputs, $X1$, $X2$, . . . , $X8$ under each of the following constraints. (All inputs are available both complemented and uncomplemented.)

a. Each output is minimized independently using a two-level NAND gate circuit, where minimum is minimum number of gates, and among those with the same number of gates, minimum number of gate inputs. (Each function must be a sum of prime implicants of that function. A gate can be shared among functions only if it implements a prime implicant of each function.) (Minimum solution: 32 gates, 95 inputs.)

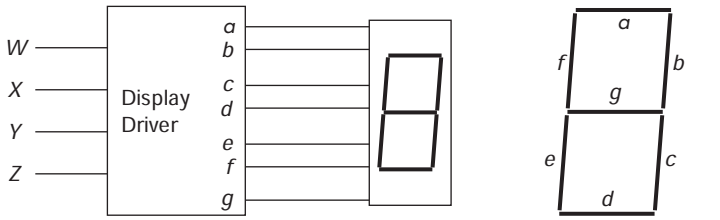
b. Two-level NAND gates, using a minimum number of the following modules:

Type 7400: 4 two-input NAND gates
Type 7410: 3 three-input NAND gates
Type 7420: 2 four-input NAND gates
Type 7430: 1 eight-input NAND gate

(There is a solution that uses 11 modules.) (Note: The solution to part a uses 13 modules.)

c. A PLA with the minimum number of terms.
For parts a and b, show the maps, the equations, and a block diagram.

22. We have a decimal digit stored in excess 3 code. The bits of the code are labeled w, x, y, z (from left to right). We wish to display that digit on a seven-segment display. The layout follows. Note that there are two ways to display a 6, a 7, and a 9; choose whichever is most convenient. The display requires a 1 to light a segment and a 0 for it not to be lit.



Design the four-input, seven-output device that takes the code for the digit produces the signals to drive the display. If any of the unused input combinations are applied to your device, the display is to be blank, that is all the outputs from the device are to be 0. Assume that the four inputs are available both uncomplemented and complemented.

We are looking for three different designs for this. For each, show the maps, the algebraic equations, and a block diagram. Please use color to make your solutions readable. For the first two parts, indicate how many of each type of package you need (7400, 7410, 7420, 7430). But minimum is defined as minimum number of gates, and among those with the same number of gates, minimum number of gate inputs.

a. First, find a minimum cost two-level NAND gate solution such that all terms are prime implicants of the individual functions. Only terms that are prime implicants of each function are to be shared. When there are multiple solutions, one answer will often lead to more sharing than others.

b. Second, reduce the number of gates by doing more sharing (including terms that are not prime implicants).

c. Third, implement this with a PLA, which has four inputs, seven outputs, and 12 product terms.

23. For the following three functions (of five variables)

$$f(a, b, c, d, e) = \Sigma m(0, 2, 5, 7, 8, 10, 13, 15, 16, 21, 23, 24, 29, 31)$$

$$g(a, b, c, d, e) = \Sigma m(2, 5, 7, 10, 13, 15, 16, 18, 20, 21, 22, 23, 25, 27)$$
$$h(a, b, c, d, e) = \Sigma m(2, 9, 10, 12, 13, 14, 16, 18, 20, 22, 28, 29, 30, 31)$$

a. Find a minimum sum of products solution for each. Show the maps and the algebraic equations for each.

b. Find a minimum solution, assuming a two-level NAND gate circuit. All variables are available both uncomplemented and complemented. Show the maps, the equations and a block diagram of the circuit. Also, indicate how many 7400 series packages you need (that is 7400, 7410, 7420, 7430). (It can be done with no more than 12 gates.)

c. Find an implementation that uses as few two-input NAND gates as possible. No gate may be used as a NOT. Show the equations and a block diagram of the circuit. (Comment: The solution may be derived from part a or from part b or some combination thereof.)

d. Show an implementation with a PLA with five inputs, three outputs, and 10 product terms.

24. Consider the following three functions:

$$f(a, b, c, d, e) = \Sigma m(2, 3, 4, 5, 8, 9, 12, 20, 21, 24, 25, 31)$$
$$g(a, b, c, d, e) = \Sigma m(2, 3, 4, 5, 6, 7, 10, 11, 12, 20, 21, 26, 27, 31)$$
$$h(a, b, c, d, e) = \Sigma m(0, 2, 3, 4, 5, 8, 10, 12, 16, 18, 19, 20, 21, 22, 23, 24, 28, 31)$$

All variables are available both uncomplemented and complemented.

a. Consider each as a separate problem and find all the minimum SOP expression(s). Both *f* and *h* have multiple solutions.

b. Assume that 7400, 7410, 7420, and 7430 packages are available at 25¢ each. Show the number of each size gate, how many of each package is required, and the total cost for a two-level solution. (Take advantage of sharing ONLY if the same term is a prime implicant of more than one function.)

c. For each function (again using the solutions of part a), find a solution that only uses 7400 and 7410 packages (25¢ each) (no four- or eight-input gates). Show the maps, the equations,

indicating sharing, and a block diagram. Show the number of each size gate, how many of each package is required and the total cost for a two-level solution.

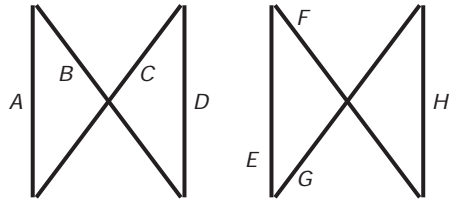
d. Take maximum advantage of sharing to try to reduce the cost of a two-level solution. Use 7400, 7410, 7420, and 7430 packages (25¢ each). Show the maps, the equations, indicating sharing, and a block diagram. Show the number of each size gate, how many of each package is required, and the total cost for a two-level solution.

e. Implement this using a ROM and also using a PLA with five inputs, 12 product terms, and three outputs.

25. Design a system that has as its inputs a number from 1 to 10 and provides as its outputs (eight of them) the signals to drive the display described below. The inputs are labeled *W*, *X*, *Y*, and *Z* and are normal binary. The input combinations 0000, 1011, 1100, 1101, 1110, and 1111 will never occur; they are to be treated as don't cares. The available building blocks are 7400, 7410, and 7420 integrated circuits. The design should use the minimum number of packages (which is five for all cases). The solution should include the maps for each of the functions and a block diagram of the circuit.

The display allows for the representation of Roman numerals (except that IIX is used to represent 8, whereas it is normally written as VIII).

There are a total of eight segments in the display, labeled *A* through *H*, as shown below.



Version High: To light a segment, a 1 is placed on the appropriate display input (*A*, *B*, . . . , *H*).

Version Low: To light a segment, a 0 is placed on the appropriate display input (*A*, *B*, . . . , *H*). Note that for this version, each input is just the complement of the one for Version High.

There are two ways to represent a 5 on this display.

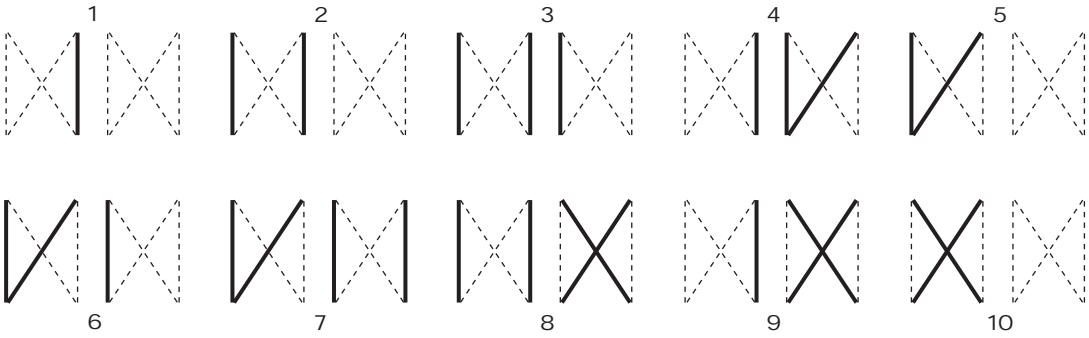
Left: Light segments *A* and *C* (or *E* and *G*).

Right: Light segments *B* and *D* (or *F* and *H*).

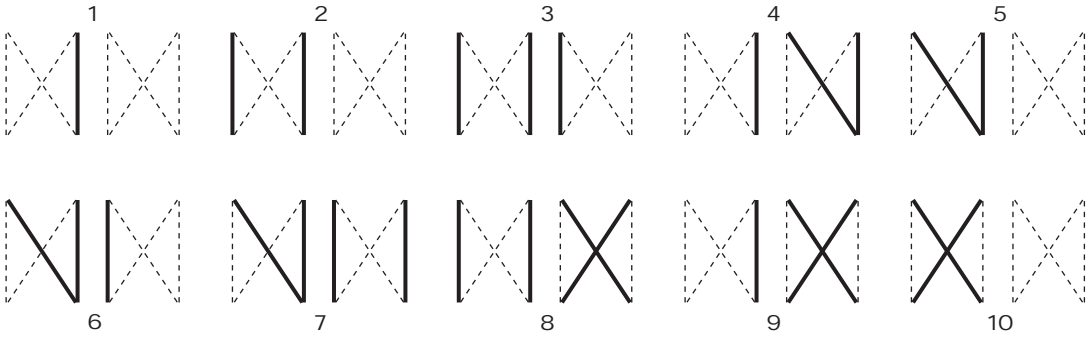
EXERCISES

The following illustration shows all digits as they should be coded for each of these, with a lit segment represented by a bold line and an unlit segment represented by a dashed line.

Left



Right



This is really four separate problems, one for each version of the design.

5.11 CHAPTER 5 TEST (60 MINUTES)

CHAPTER TEST

1. Implement the following functions using only two of the decoders described below and two 8-input OR gates.

$$f(w, x, y, z) = \Sigma m(0, 4, 5, 6, 7, 12, 15)$$
$$g(w, x, y, z) = \Sigma m(1, 3, 12, 13, 14, 15)$$