## SECTION 8.2 REVIEW

### TECHNIQUES

- Find the truth function corresponding to a given Boolean expression or logic network.
- Ⓦ Construct a logic network with the same truth function as a given Boolean expression.
- Write a Boolean expression with the same truth function as a given logic network.
- Ⓦ Write the Boolean expression in canonical sum-of-products form for a given truth function.
- Find a network composed only of NAND gates that has the same truth function as a given network with AND gates, OR gates, and inverters.
- Find a truth function that satisfies the description of a particular problem.

### MAIN IDEAS

- We can effectively convert information from any of the following three forms to any other form:

  truth function ↔ Boolean expression ↔ logic network

- A Boolean expression can sometimes be converted to a simpler, equivalent expression using the properties of Boolean algebra, thus producing a simpler network for a given truth function.

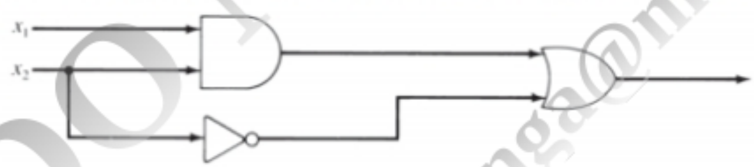


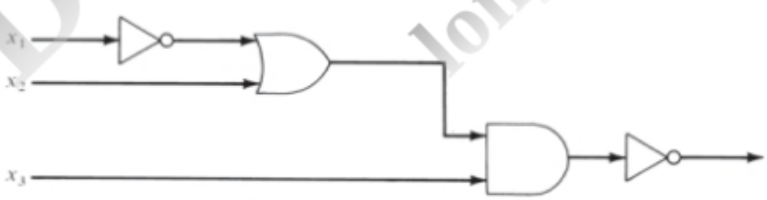
### EXERCISES 8.2

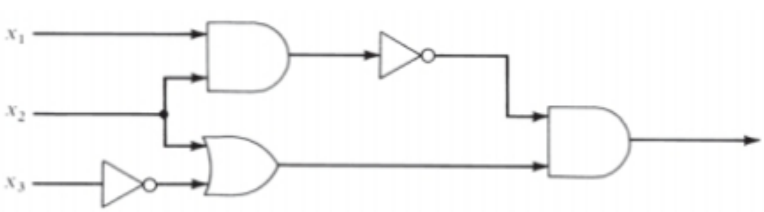
For Exercises 1–4, write a truth function and construct a logic network using AND gates, OR gates, and inverters for each of the given Boolean expressions.

1. $(x_1' + x_2)x_3$
2. $(x_1 + x_2') + x_1'x_3$
3. $x_1'x_2 + (x_1x_2)'$
4. $(x_1 + x_2)'x_3 + x_3'$

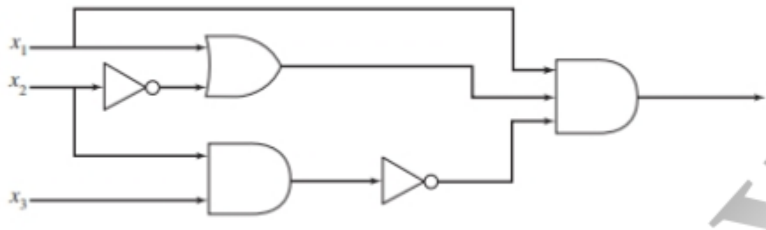
For Exercises 5–8, write a Boolean expression and a truth function for each of the logic networks shown.

5. 

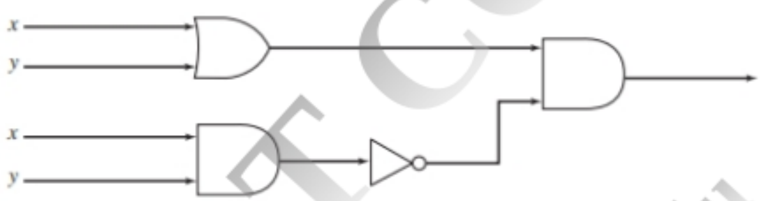
6. 

7. 

8.




9. a. Write the truth function for the Boolean operation $x \oplus y = xy' + yx'$.
   b. Draw the logic network for $x \oplus y$.
   c. Show that the network of the accompanying figure also represents $x \oplus y$. Explain why the network illustrates that $\oplus$ is the **exclusive OR** operation. (Recall that a bitwise exclusive-OR operation is used in DES encoding, as discussed in Section 5.6.)




10. a. Write the truth function for the Boolean expression

$$(xy')'(yx')'$$

   b. Draw the logic network for this expression.
   c. By looking at either the truth function or the logic network, what propositional logic connective does this Boolean expression represent?

For Exercises 11–20, find the canonical sum-of-products form for the truth functions in the given tables.

11.

| $x_1$ | $x_2$ | $f(x_1, x_2)$ |
|---|---|---|
| 1 | 1 | 0 |
| 1 | 0 | 0 |
| 0 | 1 | 0 |
| 0 | 0 | 1 |

12.

| $x_1$ | $x_2$ | $f(x_1, x_2)$ |
|---|---|---|
| 1 | 1 | 1 |
| 1 | 0 | 0 |
| 0 | 1 | 1 |
| 0 | 0 | 0 |

13.

| $x_1$ | $x_2$ | $x_3$ | $f(x_1, x_2, x_3)$ |
|---|---|---|---|
| 1 | 1 | 1 | 1 |
| 1 | 1 | 0 | 0 |
| 1 | 0 | 1 | 0 |
| 1 | 0 | 0 | 0 |
| 0 | 1 | 1 | 1 |
| 0 | 1 | 0 | 0 |
| 0 | 0 | 1 | 0 |
| 0 | 0 | 0 | 0 |

14.

| $x_1$ | $x_2$ | $x_3$ | $f(x_1, x_2, x_3)$ |
|---|---|---|---|
| 1 | 1 | 1 | 0 |
| 1 | 1 | 0 | 1 |
| 1 | 0 | 1 | 1 |
| 1 | 0 | 0 | 0 |
| 0 | 1 | 1 | 1 |
| 0 | 1 | 0 | 0 |
| 0 | 0 | 1 | 0 |
| 0 | 0 | 0 | 1 |

15.

| $x_1$ | $x_2$ | $x_3$ | $f(x_1, x_2, x_3)$ |
|---|---|---|---|
| 1 | 1 | 1 | 0 |
| 1 | 1 | 0 | 0 |
| 1 | 0 | 1 | 1 |
| 1 | 0 | 0 | 1 |
| 0 | 1 | 1 | 0 |
| 0 | 1 | 0 | 1 |
| 0 | 0 | 1 | 0 |
| 0 | 0 | 0 | 0 |

16.

| $x_1$ | $x_2$ | $x_3$ | $f(x_1, x_2, x_3)$ |
|---|---|---|---|
| 1 | 1 | 1 | 0 |
| 1 | 1 | 0 | 1 |
| 1 | 0 | 1 | 1 |
| 1 | 0 | 0 | 0 |
| 0 | 1 | 1 | 0 |
| 0 | 1 | 0 | 1 |
| 0 | 0 | 1 | 0 |
| 0 | 0 | 0 | 0 |

17.

| $x_1$ | $x_2$ | $x_3$ | $x_4$ | $f(x_1, x_2, x_3, x_4)$ |
|---|---|---|---|---|
| 1 | 1 | 1 | 1 | 1 |
| 1 | 1 | 1 | 0 | 0 |
| 1 | 1 | 0 | 1 | 1 |
| 1 | 1 | 0 | 0 | 0 |
| 1 | 0 | 1 | 1 | 1 |
| 1 | 0 | 1 | 0 | 0 |
| 1 | 0 | 0 | 1 | 1 |
| 1 | 0 | 0 | 0 | 0 |
| 0 | 1 | 1 | 1 | 0 |
| 0 | 1 | 1 | 0 | 0 |
| 0 | 1 | 0 | 1 | 0 |
| 0 | 1 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 | 1 |
| 0 | 0 | 1 | 0 | 1 |
| 0 | 0 | 0 | 1 | 0 |
| 0 | 0 | 0 | 0 | 0 |

18.

| $x_1$ | $x_2$ | $x_3$ | $x_4$ | $f(x_1, x_2, x_3, x_4)$ |
|---|---|---|---|---|
| 1 | 1 | 1 | 1 | 1 |
| 1 | 1 | 1 | 0 | 0 |
| 1 | 1 | 0 | 1 | 1 |
| 1 | 1 | 0 | 0 | 0 |
| 1 | 0 | 1 | 1 | 1 |
| 1 | 0 | 1 | 0 | 1 |
| 1 | 0 | 0 | 1 | 0 |
| 1 | 0 | 0 | 0 | 0 |
| 0 | 1 | 1 | 1 | 1 |
| 0 | 1 | 1 | 0 | 0 |
| 0 | 1 | 0 | 1 | 1 |
| 0 | 1 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 | 0 |
| 0 | 0 | 1 | 0 | 0 |
| 0 | 0 | 0 | 1 | 0 |
| 0 | 0 | 0 | 0 | 0 |

19.

| $x_1$ | $x_2$ | $x_3$ | $x_4$ | $f(x_1, x_2, x_3, x_4)$ |
|---|---|---|---|---|
| 1 | 1 | 1 | 1 | 0 |
| 1 | 1 | 1 | 0 | 0 |
| 1 | 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 0 | 0 |
| 1 | 0 | 1 | 1 | 0 |
| 1 | 0 | 1 | 0 | 1 |
| 1 | 0 | 0 | 1 | 0 |
| 1 | 0 | 0 | 0 | 0 |
| 0 | 1 | 1 | 1 | 1 |
| 0 | 1 | 1 | 0 | 0 |
| 0 | 1 | 0 | 1 | 1 |
| 0 | 1 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 | 1 |
| 0 | 0 | 1 | 0 | 1 |
| 0 | 0 | 0 | 1 | 1 |
| 0 | 0 | 0 | 0 | 0 |

20.

| $x_1$ | $x_2$ | $x_3$ | $x_4$ | $f(x_1, x_2, x_3, x_4)$ |
|---|---|---|---|---|
| 1 | 1 | 1 | 1 | 1 |
| 1 | 1 | 1 | 0 | 1 |
| 1 | 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 0 | 1 |
| 1 | 0 | 1 | 1 | 1 |
| 1 | 0 | 1 | 0 | 0 |
| 1 | 0 | 0 | 1 | 0 |
| 1 | 0 | 0 | 0 | 0 |
| 0 | 1 | 1 | 1 | 0 |
| 0 | 1 | 1 | 0 | 1 |
| 0 | 1 | 0 | 1 | 0 |
| 0 | 1 | 0 | 0 | 1 |
| 0 | 0 | 1 | 1 | 0 |
| 0 | 0 | 1 | 0 | 0 |
| 0 | 0 | 0 | 1 | 0 |
| 0 | 0 | 0 | 0 | 0 |

21. a. Find the canonical sum-of-products form for the truth function in the accompanying table.
   b. Draw the logic network for the expression of part (a).
   c. Use properties of a Boolean algebra to reduce the expression of part (a) to an equivalent expression whose network requires only two logic elements. Draw the network.

| $x_1$ | $x_2$ | $x_3$ | $f(x_1, x_2, x_3)$ |
|---|---|---|---|
| 1 | 1 | 1 | 0 |
| 1 | 1 | 0 | 1 |
| 1 | 0 | 1 | 0 |
| 1 | 0 | 0 | 1 |
| 0 | 1 | 1 | 0 |
| 0 | 1 | 0 | 0 |
| 0 | 0 | 1 | 0 |
| 0 | 0 | 0 | 0 |

22. a. Find the canonical sum-of-products form for the truth function in the accompanying table.
   b. Draw the logic network for the expression of part (a).
   c. Use properties of a Boolean algebra to reduce the expression of part (a) to an equivalent expression whose network requires only three logic elements. Draw the network.

| $x_1$ | $x_2$ | $x_3$ | $f(x_1, x_2, x_3)$ |
|---|---|---|---|
| 1 | 1 | 1 | 1 |
| 1 | 1 | 0 | 0 |
| 1 | 0 | 1 | 0 |
| 1 | 0 | 0 | 0 |
| 0 | 1 | 1 | 1 |
| 0 | 1 | 0 | 1 |
| 0 | 0 | 1 | 0 |
| 0 | 0 | 0 | 0 |

23. a. Show that the two Boolean expressions

$$(x_1 + x_2)(x_1' + x_3)(x_2 + x_3)$$

and

$$(x_1x_3) + (x_1'x_2)$$

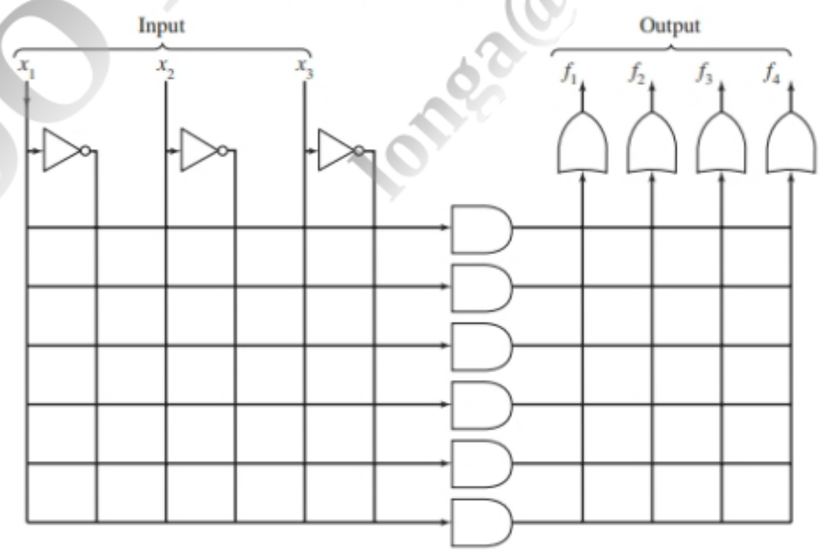
are equivalent by writing the truth function for each.

b. Write the canonical sum-of-products form equivalent to the two expressions of part (a).

c. Use properties of a Boolean algebra to reduce one of the expressions of part (a) to the other.

24. The accompanying figure shows an unprogrammed PLD for three inputs, $x_1$, $x_2$, and $x_3$. Program this PLD to generate the truth functions $f_1$ and $f_3$ represented by

$$f_1\colon\ x_1x_2x_3 + x_1'x_2x_3' + x_1'x_2'x_3$$
$$f_3\colon\ x_1x_2'x_3' + x_1'x_2'x_3 + x_1'x_2'x_3'$$



25. There is also a *canonical product-of-sums form* (*conjunctive normal form*) for any truth function. This expression has the form

$$()() \cdots ()$$

with each factor a sum of the form

$$\alpha + \beta + \cdots + \omega$$

where $\alpha = x_1$ or $x_1'$, $\beta = x_2$ or $x_2'$, and so on. Each factor is constructed to have a value of 0 for the input values of exactly one of the rows of the truth function having value 0. Thus, the entire expression has value 0 for these inputs and no others. Find the canonical product-of-sums form for the truth functions of Exercises 11–15.

26. Consider the truth function given here. Because there are many more 1*s* than 0*s*. the canonical sum-of-products form might seem tedious to compute. Instead, create a Boolean sum-of-products expression using the same formula as before but on the 0 rows instead of the 1 rows. This expression will give outputs of 1*s* for those rows and no others, exactly the opposite of what you want. Then complement the expression (equivalent to sticking an inverter on the end of the network).

| $x_1$ | $x_2$ | $x_3$ | $f(x_1, x_2, x_3)$ |
|---|---|---|---|
| 1 | 1 | 1 | 1 |
| 1 | 1 | 0 | 1 |
| 1 | 0 | 1 | 1 |
| 1 | 0 | 0 | 0 |
| 0 | 1 | 1 | 1 |
| 0 | 1 | 0 | 1 |
| 0 | 0 | 1 | 0 |
| 0 | 0 | 0 | 1 |

a. Use this approach to create a Boolean expression for this truth function.

b. Prove that the resulting expression is equivalent to the canonical product-of-sums form described in Exercise 25.

27. The *2's complement* of an $n$-bit binary number $p$ is an $n$-bit binary number $q$ such that $p + q$ equals an $n$-bit representation of zero (any carry bit to column $n + 1$ is ignored). Thus 01110 is the 2's complement of 10010 because

$$\begin{array}{r} 10010 \\ +\ 01110 \\ \hline (1)00000 \end{array}$$

The 2's complement idea can be used to represent negative integers in binary form. After all, the negative of $p$ is by definition a number that, when added to $p$, results in zero.

Given a binary number $p$, the 2's complement of $p$ is found by scanning $p$ from low-order to high-order bits (right to left). As long as bit $i$ of $p$ is 0, bit $i$ of $q$ is 0. When the first 1 of $p$ is encountered, say at bit $j$, then bit $j$ of $q$ is 1, but for the remaining bits, $j < i \leq n$, $q_i = p_i'$. For $p = 10010$, for instance, the rightmost 0 bit of $p$ stays a 0 bit in $q$, and the first 1 bit stays a 1 bit. The remaining bits of $q$, however, are the reverse of the bits in $p$.

First 1

$$\begin{array}{rl} p = 100 & \vdots\ 10 \\ q = \underbrace{011} & \vdots\ \underbrace{10} \\ q_i = p_i' & \vdots\ q_i = p_i \end{array}$$

For each binary number $p$, find the 2's complement of $p$, namely $q$, and then calculate $p + q$.

a. 1100 b. 1001 c. 001

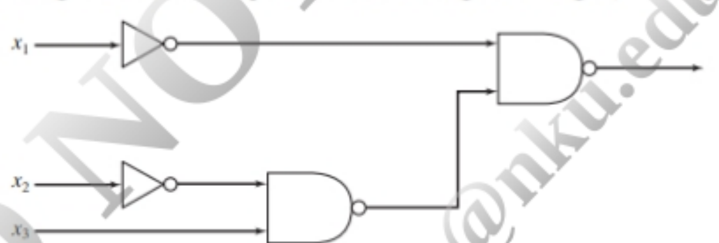
28. For any bit $x_i$ in a binary number $p$, let $r_i$ be the corresponding bit in $q$, the 2's complement of $p$ (see Exercise 27). The value of $r_i$ depends on the value of $x_i$ and also on the position of $x_i$ relative to the first 1 bit in $p$. For the $i$th bit, let $c_{i-1}$ denote a 0 if the bits $p_j$, $1 \le j \le i - 1$, are 0 and a 1 otherwise. A value $c_i$ must be computed to move on to the next bit.
   a. Give a truth function for $r_i$ with inputs $x_i$ and $c_{i-1}$. Give a truth function for $c_i$ with inputs $x_i$ and $c_{i-1}$.
   b. Write Boolean expressions for the truth functions of part (a). Simplify as much as possible.
   c. Design a circuit module to output $r_i$ and $c_i$ from inputs $x_i$ and $c_{i-1}$.
   d. Using the modules of part (c), design a circuit to find the 2's complement of a three-bit binary number zyx. Trace the operation of the circuit in computing the 2's complement of 110.

29. a. Construct a network for the following expression using only NAND elements. Replace the AND and OR gates and inverters with the appropriate NAND networks.

$$x_3'x_1 + x_2'x_1 + x_3'$$

   b. Use the properties of a Boolean algebra to reduce the expression of part (a) to one whose network would require only three NAND gates. Draw the network.

30. Replace the following network with an equivalent network using one AND gate, one OR gate, and one inverter.




31. Using only NOR elements, construct networks that can replace (a) an inverter, (b) an OR gate, and (c) an AND gate.

32. Find an equivalent network for the half-adder module that uses exactly five NAND gates. Draw the network.

33. A thermostat controls a heating system that should raise the temperature above 67°F during working hours (between 7:00 AM and 6:00 PM). The input values are

$x_1$ = 1 when the temperature is less than 67°F
$x_1$ = 0 otherwise
$x_2$ = 1 when the time is less than 7:00 AM or greater than 6:00 PM
$x_2$ = 0 otherwise

Find a truth function, a Boolean expression, and a logic network for when the heating system should be on (value 1) or off (value 0).