

Exercises

1. Find a Boolean product of the Boolean variables x , y , and z , or their complements, that has the value 1 if and only if

a) $x = y = 0, z = 1$. $\bar{x}\bar{y}z$ ✓ b) $x = 0, y = 1, z = 0$. $\bar{x}y\bar{z}$ ✓
c) $x = 0, y = z = 1$. $\bar{x}yz$ ✓ d) $x = y = z = 0$. $\bar{x}\bar{y}\bar{z}$ ✓

2. Find the sum-of-products expansions of these Boolean functions.

a) $F(x, y) = \bar{x} + y$ b) $F(x, y) = x\bar{y}$
c) $F(x, y) = 1$ d) $F(x, y) = \bar{y}$

3. Find the sum-of-products expansions of these Boolean functions.

a) $F(x, y, z) = x + y + z$ $x\bar{y}\bar{z} + x\bar{y}z + x\bar{y}\bar{z} + x\bar{y}z + x\bar{y}\bar{z} + x\bar{y}z + x\bar{y}\bar{z} + x\bar{y}z$ ✓
b) $F(x, y, z) = (x + z)y$ $xy\bar{z} + x\bar{y}\bar{z} + \bar{x}yz$ ✓
c) $F(x, y, z) = x$ $x\bar{y}\bar{z} + x\bar{y}z + x\bar{y}\bar{z} + x\bar{y}z$ ✓
d) $F(x, y, z) = x\bar{y}$ $x\bar{y}\bar{z} + x\bar{y}z$ ✓

4. Find the sum-of-products expansions of the Boolean function $F(x, y, z)$ that equals 1 if and only if

a) $x = 0$. b) $xy = 0$.
c) $x + y = 0$. d) $xyz = 0$.

5. Find the sum-of-products expansion of the Boolean function $F(w, x, y, z)$ that has the value 1 if and only if an odd number of w, x, y , and z have the value 1. $w\bar{x}\bar{y}\bar{z} + w\bar{x}y\bar{z} + w\bar{x}y\bar{z} + w\bar{x}y\bar{z} + w\bar{x}y\bar{z} + w\bar{x}y\bar{z} + w\bar{x}y\bar{z} + w\bar{x}y\bar{z}$

6. Find the sum-of-products expansion of the Boolean function $F(x_1, x_2, x_3, x_4, x_5)$ that has the value 1 if and only if three or more of the variables x_1, x_2, x_3, x_4 , and x_5 have the value 1.

Another way to find a Boolean expression that represents a Boolean function is to form a Boolean product of Boolean sums of literals. Exercises 7–11 are concerned with representations of this kind.

7. Find a Boolean sum containing either x or \bar{x} , either y or \bar{y} , and either z or \bar{z} that has the value 0 if and only if

a) $x = y = 1, z = 0$. $\bar{x} + \bar{y} + z$ b) $x = y = z = 0$. $x + y + z$ ✓

c) $x = z = 0, y = 1$. $x + \bar{y} + z$

8. Find a Boolean product of Boolean sums of literals that has the value 0 if and only if $x = y = 1$ and $z = 0$, $x = z = 0$ and $y = 1$, or $x = y = z = 0$. [Hint: Take the Boolean product of the Boolean sums found in parts (a), (b), and (c) in Exercise 7.]

9. By definition $+$, the sum $y_1 + y_2 + \dots + y_n$ has the value 0 iff each $y_i = 0$. This happens precisely when $x_i = 0$ for those cases in which $y_i = x_i$ and $x_i = 1$ in those cases in which $y_i = \bar{x}_i$.

9. Show that the Boolean sum $y_1 + y_2 + \dots + y_n$, where $y_i = x_i$ or $y_i = \bar{x}_i$, has the value 0 for exactly one combination of the values of the variables, namely, when $x_i = 0$ if $y_i = x_i$ and $x_i = 1$ if $y_i = \bar{x}_i$. This Boolean sum is called a **maxterm**.

10. Show that a Boolean function can be represented as a Boolean product of maxterms. This representation is called the **product-of-sums expansion** or **conjunctive normal form** of the function. [Hint: Include one maxterm in this product for each combination of the variables where the function has the value 0.]

11. Find the product-of-sums expansion of each of the Boolean functions in Exercise 3. ... that's ex.3's question

12. Express each of these Boolean functions using the operators \cdot and $\bar{}$.

a) $x + y + z$ b) $x + \bar{y}(\bar{x} + z)$
c) $\bar{x} + \bar{y}$ d) $\bar{x}(x + \bar{y} + \bar{z})$

13. Express each of the Boolean functions in Exercise 12 using the operators $+$ and $\bar{}$. $\rightarrow ?$ it is originally like that \rightarrow true for (a)(c) (b)(d) too complicated since why

14. Show that

a) $\bar{x} = x \downarrow x$. b) $xy = (x \downarrow y) \downarrow (x \downarrow y)$.

c) $x + y = (x \downarrow x) \downarrow (y \downarrow y)$.

15. Show that

a) $\bar{x} = x \downarrow x$. $1 \downarrow 1 = 0$ $0 \downarrow 0 = 1$ ✓

b) $xy = (x \downarrow x) \downarrow (y \downarrow y)$. $x \downarrow y = 0: 0 \downarrow 0 = 1$
 $x \downarrow y = 1: 1 \downarrow 1 = 0$

c) $x + y = (x \downarrow y) \downarrow (x \downarrow y)$. $x \downarrow y = 1: 1 \downarrow 1 = 0$
 $x \downarrow y = 0: 0 \downarrow 0 = 1$

16. Show that $\{\downarrow\}$ is functionally complete using Exercise 15.

17. Express each of the Boolean functions in Exercise 3 using the operator \downarrow .
3(a). $((x \downarrow x) \downarrow (y \downarrow y)) \downarrow ((x \downarrow x) \downarrow (y \downarrow y)) \downarrow (z \downarrow z)$
(b). $((x \downarrow x) \downarrow (z \downarrow z)) \downarrow y \downarrow ((x \downarrow x) \downarrow (z \downarrow z)) \downarrow y$
(c). no need (d). $(x \downarrow (y \downarrow y)) \downarrow (x \downarrow (y \downarrow y))$

18. Express each of the Boolean functions in Exercise 3 using the operator \downarrow .

19. Show that the set of operators $\{+, \cdot\}$ is not functionally complete. ...

20. Are these sets of operators functionally complete?

a) $\{+, \oplus\}$ b) $\{\bar{}, \oplus\}$ c) $\{\cdot, \oplus\}$

12.3 Logic Gates

Introduction

Boolean algebra is used to model the circuitry of electronic devices. Each input and each output of such a device can be thought of as a member of the set $\{0, 1\}$. A computer, or other electronic device, is made up of a number of circuits. Each circuit can be designed using the rules of Boolean algebra that were studied in Sections 12.1 and 12.2. The **basic elements of circuits**

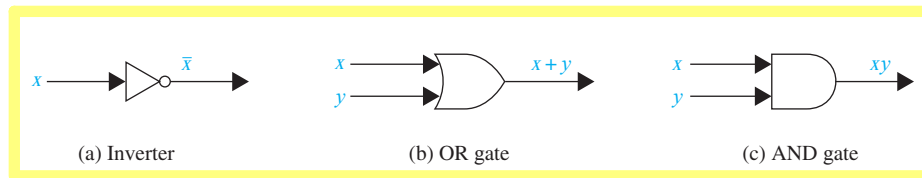


FIGURE 1 Basic Types of Gates.

are called **gates**, and were introduced in Section 1.2. Each type of gate implements a Boolean operation. In this section we define several types of gates. Using these gates, we will apply the rules of Boolean algebra to design circuits that perform a variety of tasks. The circuits that we will study in this chapter give output that depends only on the input, and not on the current state of the circuit. In other words, these circuits have no memory capabilities. Such circuits are called **combinational circuits** or **gating networks**.

We will construct combinational circuits using three types of elements. The first is an **inverter**, which accepts the value of one Boolean variable as input and produces the complement of this value as its output. The symbol used for an inverter is shown in Figure 1(a). The input to the inverter is shown on the left side entering the element, and the output is shown on the right side leaving the element.

The next type of element we will use is the **OR gate**. The **inputs** to this gate are the values of **two or more Boolean variables**. The **output** is the Boolean **sum of their values**. The symbol used for an OR gate is shown in Figure 1(b). The inputs to the OR gate are shown on the left side entering the element, and the output is shown on the right side leaving the element.

The third type of element we will use is the **AND gate**. The **inputs** to this gate are the values of **two or more Boolean variables**. The **output** is the Boolean **product of their values**. The symbol used for an AND gate is shown in Figure 1(c). The inputs to the AND gate are shown on the left side entering the element, and the output is shown on the right side leaving the element.

We will permit multiple inputs to AND and OR gates. The inputs to each of these gates are shown on the left side entering the element, and the output is shown on the right side. Examples of AND and OR gates with n inputs are shown in Figure 2.

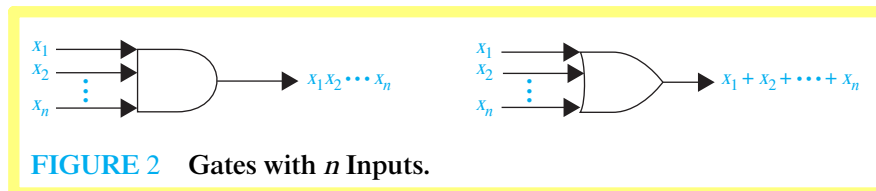


FIGURE 2 Gates with n Inputs.

Combinations of Gates

Combinational circuits can be constructed using a combination of inverters, OR gates, and AND gates. When combinations of circuits are formed, some gates may share inputs. This is shown in one of two ways in **depictions** of circuits. One method is to use branchings that indicate all the gates that use a given input. The other method is to indicate this input separately for each gate. Figure 3 illustrates the two ways of showing gates with the same input values. Note also that output from a gate may be used as input by one or more other elements, as shown in Figure 3. Both drawings in Figure 3 depict the circuit that produces the output $xy + \bar{x}y$.

EXAMPLE 1 Construct circuits that produce the following outputs: (a) $(x + y)\bar{x}$, (b) $\bar{x}(y + \bar{z})$, and (c) $(x + y + z)(\bar{x}\bar{y}\bar{z})$.

Solution: Circuits that produce these outputs are shown in Figure 4.

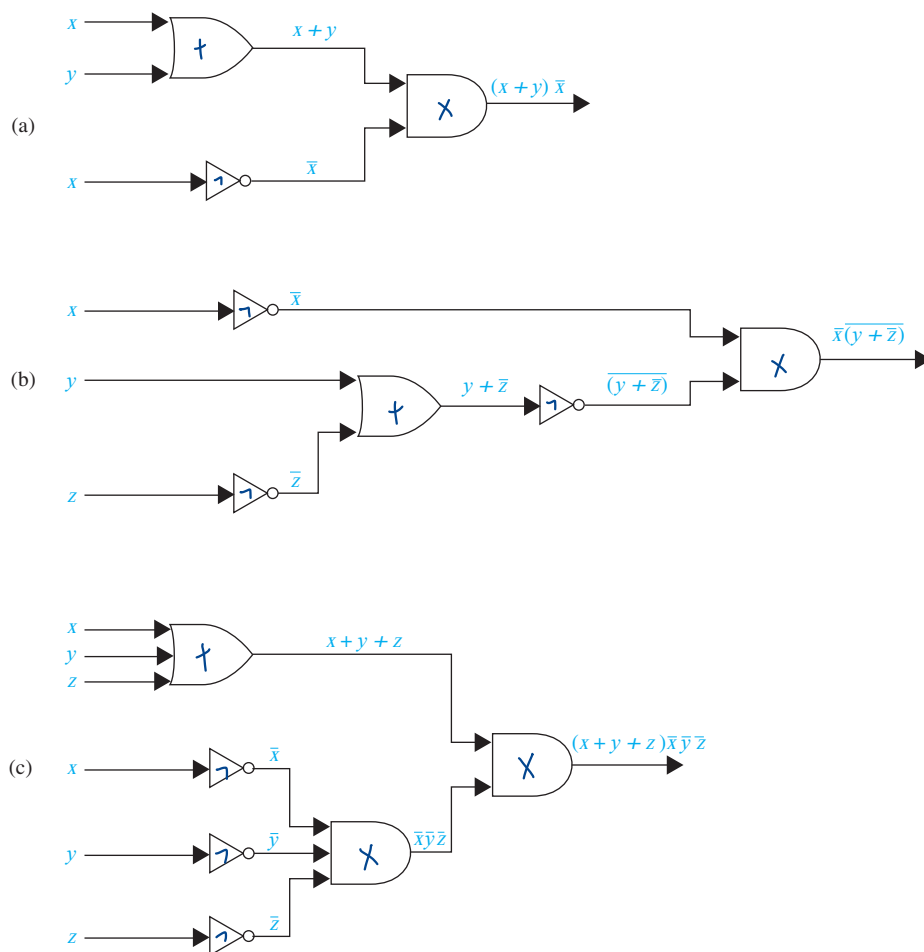
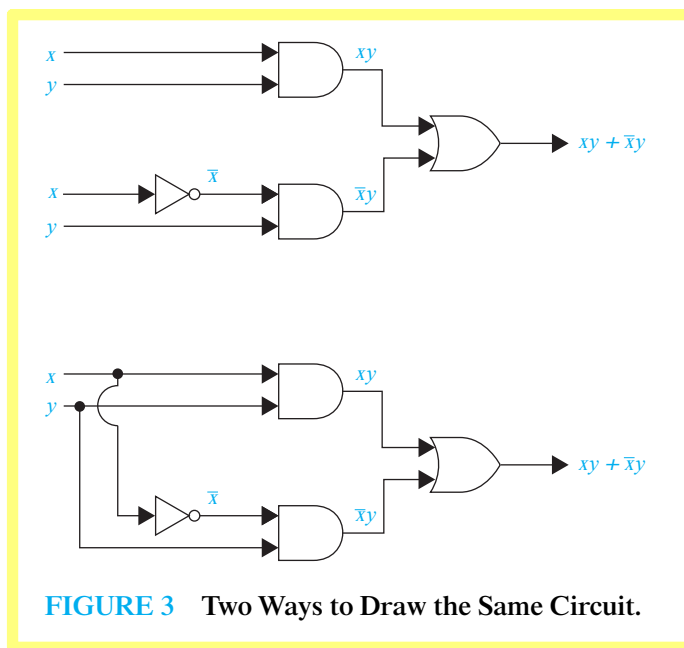


FIGURE 4 Circuits that Produce the Outputs Specified in Example 1.

Examples of Circuits

We will give some examples of circuits that perform some useful functions.

EXAMPLE 2 A committee of three individuals decides issues for an organization. Each individual votes either yes or no for each proposal that arises. A proposal is passed if it receives at least two yes votes. Design a circuit that determines whether a proposal passes.



Solution: Let $x = 1$ if the first individual votes yes, and $x = 0$ if this individual votes no; let $y = 1$ if the second individual votes yes, and $y = 0$ if this individual votes no; let $z = 1$ if the third individual votes yes, and $z = 0$ if this individual votes no. Then a circuit must be designed that produces the output 1 from the inputs x , y , and z when two or more of x , y , and z are 1. One representation of the Boolean function that has these output values is $xy + xz + yz$ (see Exercise 12 in Section 12.1). The circuit that implements this function is shown in Figure 5.

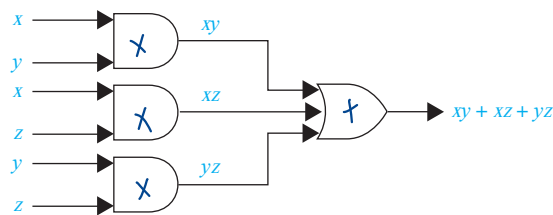


FIGURE 5 A Circuit for Majority Voting.

EXAMPLE 3 Sometimes light fixtures are controlled by more than one switch. Circuits need to be designed so that flipping any one of the switches for the fixture turns the light on when it is off and turns the light off when it is on. Design circuits that accomplish this when there are two switches and when there are three switches.

a bit confused ←

TABLE 1		
x	y	$F(x, y)$
1	1	1
1	0	0
0	1	0
0	0	1

Solution: We will begin by designing the circuit that controls the light fixture when two different switches are used. Let $x = 1$ when the first switch is closed and $x = 0$ when it is open, and let $y = 1$ when the second switch is closed and $y = 0$ when it is open. Let $F(x, y) = 1$ when the light is on and $F(x, y) = 0$ when it is off. We can arbitrarily decide that the light will be on when both switches are closed, so that $F(1, 1) = 1$. This determines all the other values of F . When one of the two switches is opened, the light goes off, so $F(1, 0) = F(0, 1) = 0$. When the other switch is also opened, the light goes on, so $F(0, 0) = 1$. Table 1 displays these values. Note that $F(x, y) = xy + \bar{x}\bar{y}$. This function is implemented by the circuit shown in Figure 6.

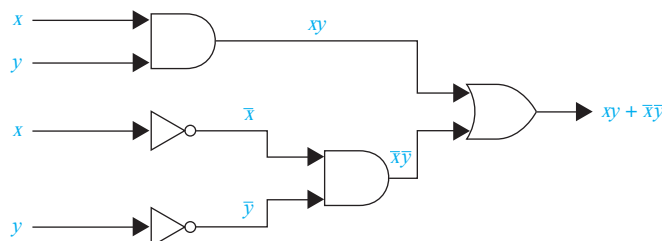


FIGURE 6 A Circuit for a Light Controlled by Two Switches.

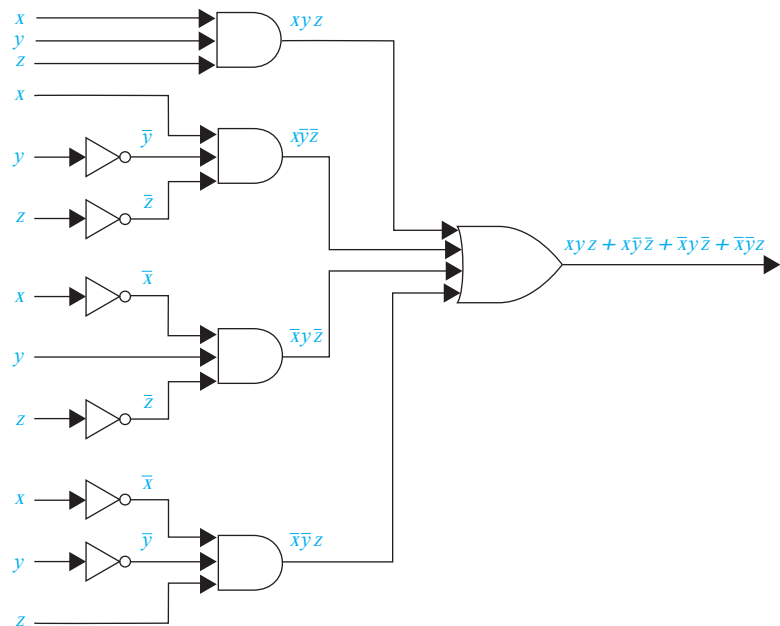


FIGURE 7 A Circuit for a Fixture Controlled by Three Switches.

TABLE 2			
x	y	z	$F(x, y, z)$
1	1	1	1
1	1	0	0
1	0	1	0
1	0	0	1
0	1	1	0
0	1	0	1
0	0	1	1
0	0	0	0

We will now design a circuit for three switches. Let x , y , and z be the Boolean variables that indicate whether each of the three switches is closed. We let $x = 1$ when the first switch is closed, and $x = 0$ when it is open; $y = 1$ when the second switch is closed, and $y = 0$ when it is open; and $z = 1$ when the third switch is closed, and $z = 0$ when it is open. Let $F(x, y, z) = 1$ when the light is on and $F(x, y, z) = 0$ when the light is off. We can arbitrarily specify that the light be on when all three switches are closed, so that $F(1, 1, 1) = 1$. This determines all other values of F . When one switch is opened, the light goes off, so $F(1, 1, 0) = F(1, 0, 1) = F(0, 1, 1) = 0$. When a second switch is opened, the light goes on, so $F(1, 0, 0) = F(0, 1, 0) = F(0, 0, 1) = 1$. Finally, when the third switch is opened, the light goes off again, so $F(0, 0, 0) = 0$. Table 2 shows the values of this function.

The function F can be represented by its sum-of-products expansion as $F(x, y, z) = xyz + x\bar{y}\bar{z} + \bar{x}yz + \bar{x}\bar{y}z$. The circuit shown in Figure 7 implements this function. ◀

Adders



TABLE 3 Input and Output for the Half Adder.			
Input		Output	
x	y	s	c
1	1	0	1
1	0	1	0
0	1	1	0
0	0	0	0

We will illustrate how logic circuits can be used to carry out addition of two positive integers from their binary expansions. We will build up the circuitry to do this addition from some component circuits. First, we will build a circuit that can be used to find $x + y$, where x and y are two bits. The input to our circuit will be x and y , because these each have the value 0 or the value 1. The output will consist of two bits, namely, s and c , where s is the sum bit and c is the carry bit. This circuit is called a **multiple output circuit** because it has more than one output. The circuit that we are designing is called the **half adder**, because it adds two bits, without considering a carry from a previous addition. We show the input and output for the half adder in Table 3. From Table 3 we see that $c = xy$ and that $s = x\bar{y} + \bar{x}y = (x + y)(\bar{x}y)$. Hence, the circuit shown in Figure 8 computes the sum bit s and the carry bit c from the bits x and y .

We use the **full adder** to compute the sum bit and the carry bit when two bits and a carry are added. The inputs to the full adder are the bits x and y and the carry c_i . The outputs are the sum bit s and the new carry c_{i+1} . The inputs and outputs for the full adder are shown in Table 4.

how to know which one is which ??
why is xy the carry bit not xy
why is there multiple outputs?

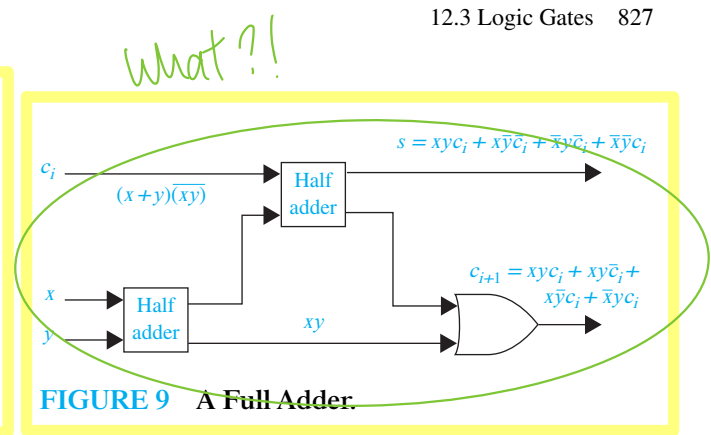
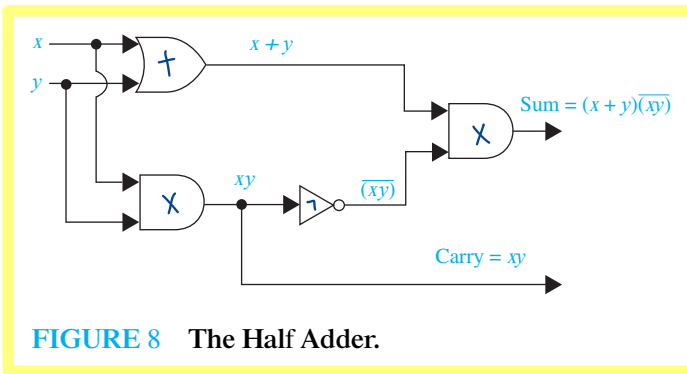
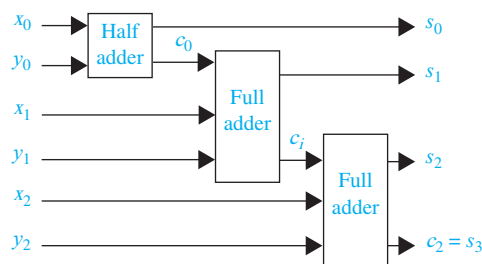


TABLE 4
Input and
Output for
the Full Adder.

Input			Output	
x	y	ci	s	ci+1
1	1	1	1	1
1	1	0	0	1
1	0	1	0	1
1	0	0	1	0
0	1	1	0	1
0	1	0	1	0
0	0	1	1	0
0	0	0	0	0

The two outputs of the full adder, the sum bit s and the carry c_{i+1} , are given by the sum-of-products expansions $xy c_i + x \bar{y} \bar{c}_i + \bar{x} y \bar{c}_i + \bar{x} \bar{y} c_i$ and $xy c_i + xy \bar{c}_i + x \bar{y} c_i + \bar{x} y c_i$, respectively. However, instead of designing the full adder from scratch, we will use half adders to produce the desired output. A full adder circuit using half adders is shown in Figure 9.

Finally, in Figure 10 we show how full and half adders can be used to add the two three-bit integers $(x_2 x_1 x_0)_2$ and $(y_2 y_1 y_0)_2$ to produce the sum $(s_3 s_2 s_1 s_0)_2$. Note that s_3 , the highest-order bit in the sum, is given by the carry c_2 .



Exercises

In Exercises 1–5 find the output of the given circuit.

