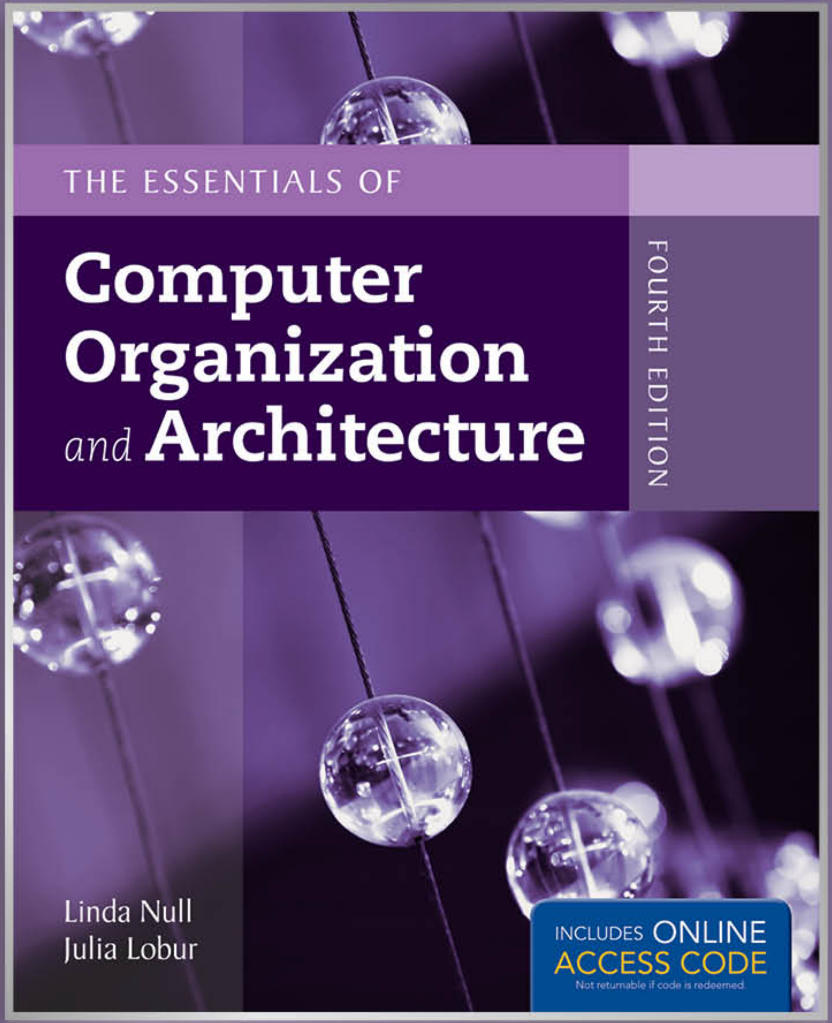
Chapter 3

Boolean Algebra and Digital Logic



© Odua Images/ShutterStock, Inc. Copyright © 2014 by Jones & Bartlett Learning, LLC an Ascend Learning Company
www.jblearning.com

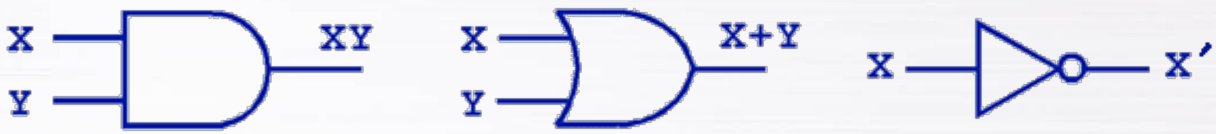
3.3 Logic Gates

- **Boolean algebra** is an **abstract** system.
- In this section, we see that **Boolean functions** can be **implemented in digital circuits** consisting of gates.
- A **gate** is an **electronic device** that produces a result based on two or more input values.

© Odua Images/ShutterStock, Inc. Copyright © 2014 by Jones & Bartlett Learning, LLC an Ascend Learning Company
www.jblearning.com

3.3 Logic Gates

- The three simplest gates are the **AND**, **OR**, and **NOT** gates.



X AND Y

X	Y	XY
0	0	0
0	1	0
1	0	0
1	1	1

X OR Y

X	Y	X+Y
0	0	0
0	1	1
1	0	1
1	1	1

NOT X

X	X′
0	1
1	0

- They **correspond** directly to their respective **Boolean operations**, as you can see by their truth tables.

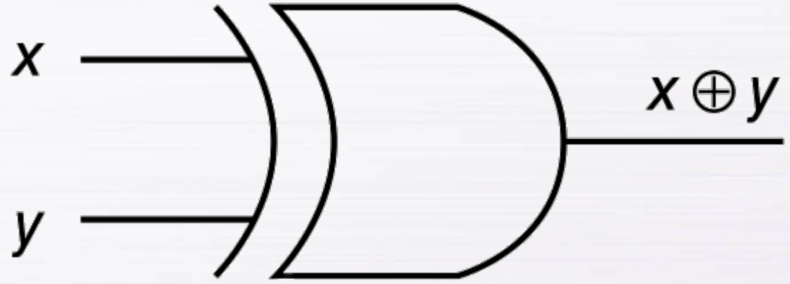
© Odua Images/ShutterStock, Inc. Copyright © 2014 by Jones & Bartlett Learning, LLC an Ascend Learning Company
www.jblearning.com

3.3 Logic Gates

- Another very useful gate is the **exclusive OR** (**XOR**) gate.
- The output of the XOR operation is **true** only **when** the values of the **inputs differ**.

X XOR Y

X	Y	X ⊕ Y
0	0	0
0	1	1
1	0	1
1	1	0



Note the special symbol ⊕ for the XOR operation.

© Odua Images/ShutterStock, Inc. Copyright © 2014 by Jones & Bartlett Learning, LLC an Ascend Learning Company
www.jblearning.com

3.3 Logic Gates

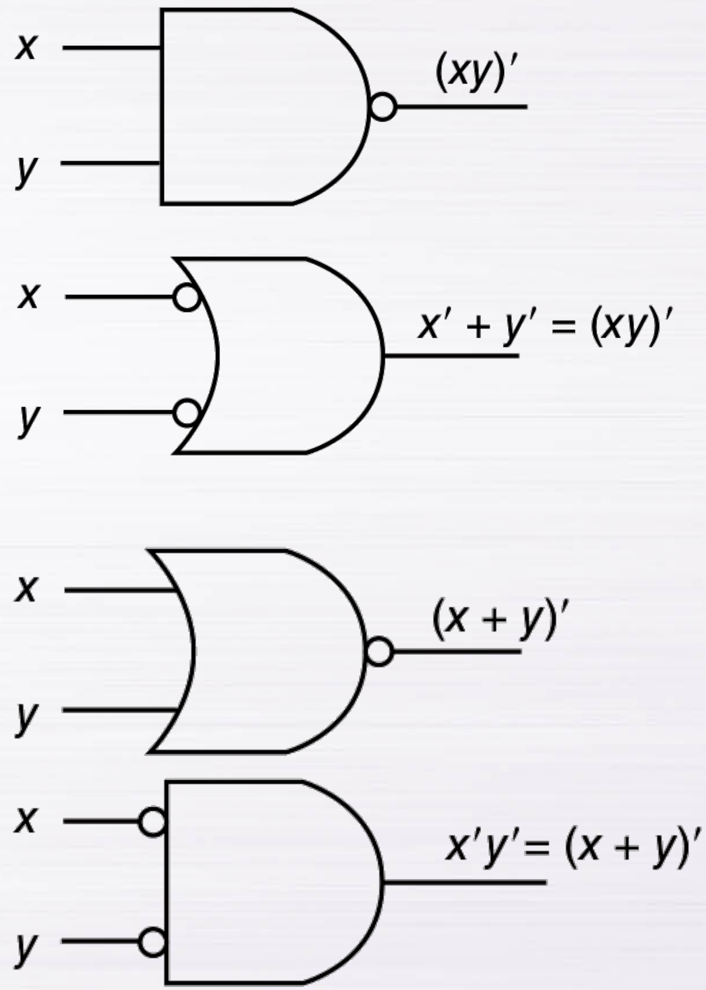
- **NAND** and **NOR** are two very important gates. Their symbols and truth tables are shown at the right.

X NAND Y

X	Y	X NAND Y
0	0	1
0	1	1
1	0	1
1	1	0

X NOR Y

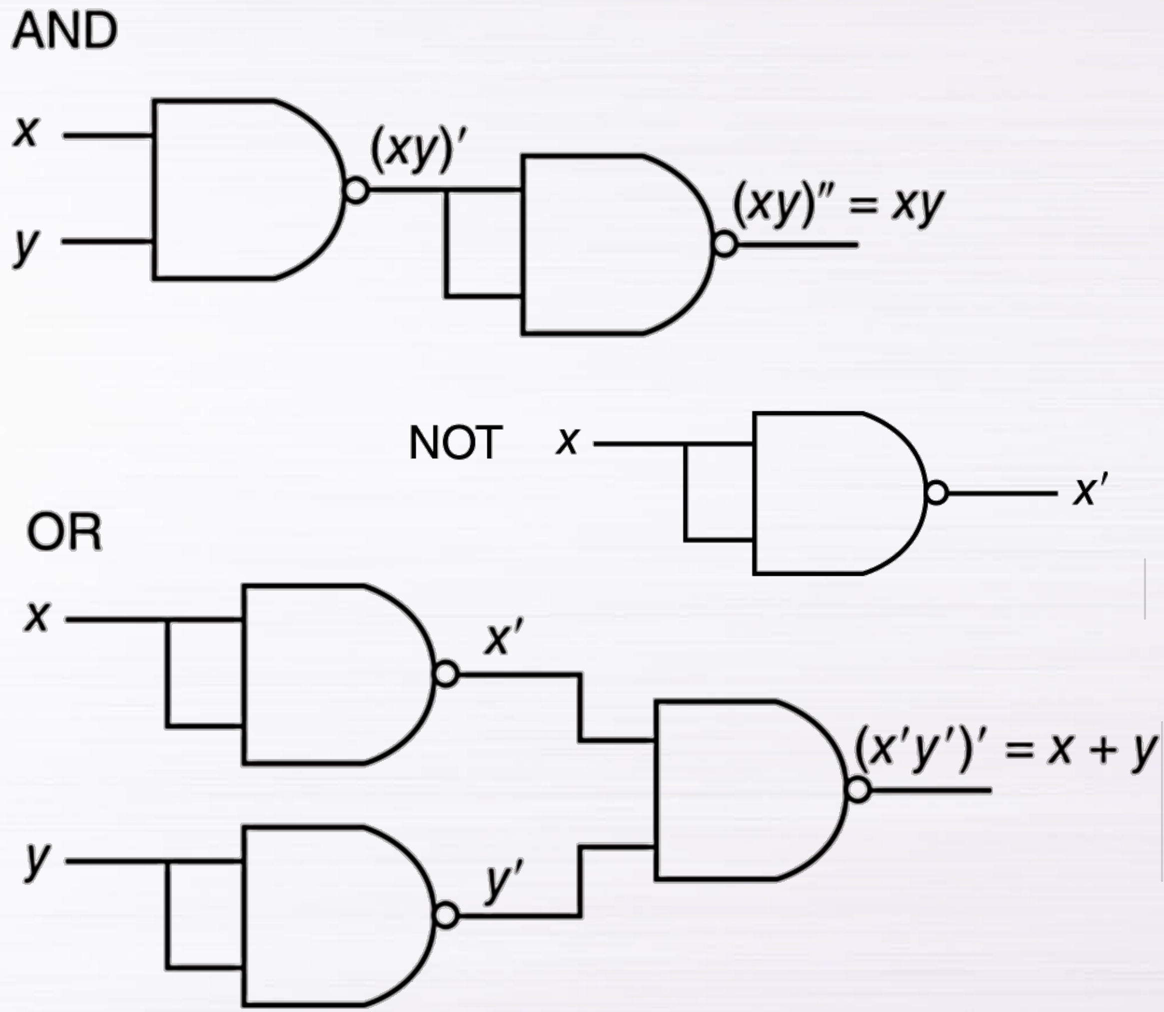
X	Y	X NOR Y
0	0	1
0	1	0
1	0	0
1	1	0



© Odua Images/ShutterStock, Inc. Copyright © 2014 by Jones & Bartlett Learning, LLC an Ascend Learning Company
www.jblearning.com

3.3 Logic Gates

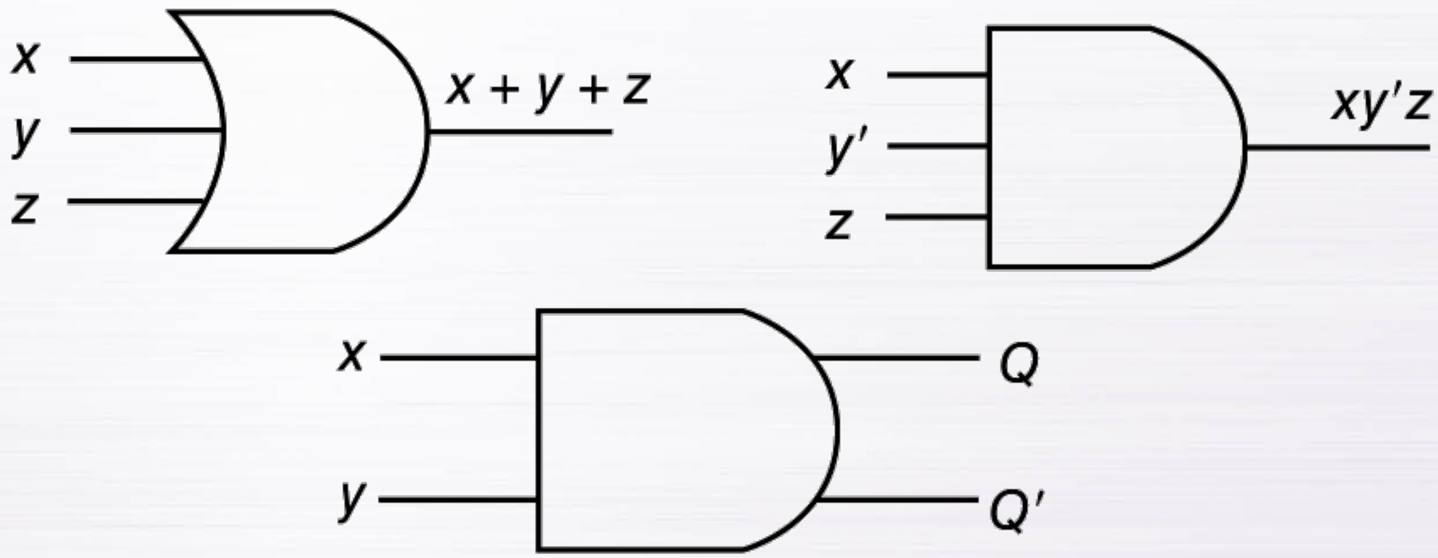
- **NAND** and **NOR** are known as *universal gates* because they are **cheap to manufacture** and **any Boolean function** can be constructed **using** only **NAND** or only **NOR** gates.



© Odua Images/ShutterStock, Inc. Copyright © 2014 by Jones & Bartlett Learning, LLC an Ascend Learning Company
www.jblearning.com

3.3 Logic Gates

- Gates can have **multiple inputs** and more than one output.
  - A second **output** can be provided for the complement of the operation.



© Odua Images/ShutterStock, Inc. Copyright © 2014 by Jones & Bartlett Learning, LLC an Ascend Learning Company
www.jblearning.com

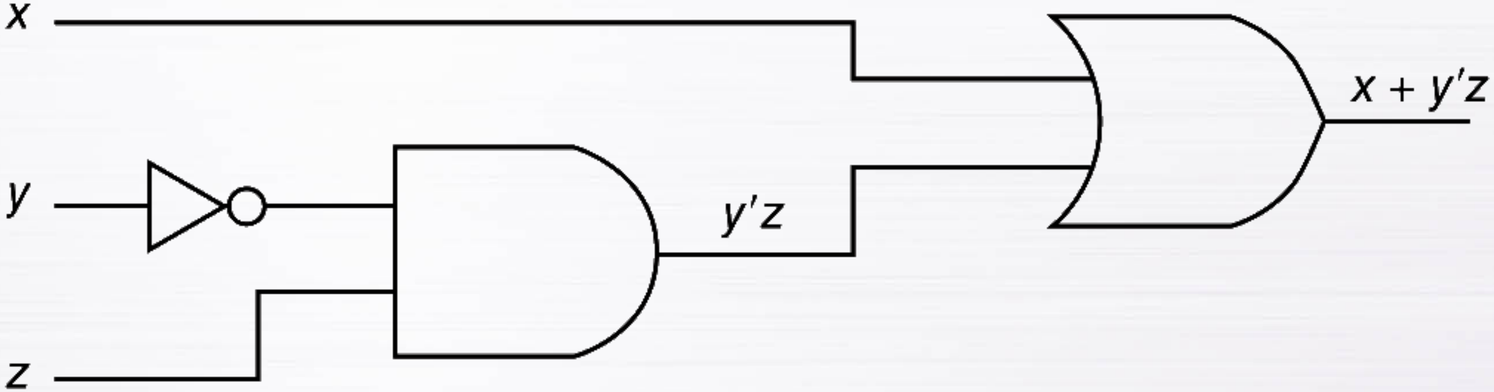
3.3 Logic Gates

- The main thing to remember is that **combinations of gates implement Boolean functions**.
- The circuit below implements the Boolean function $\mathbf{F(x,y,z) = x + y'z}$:

© Odua Images/ShutterStock, Inc. Copyright © 2014 by Jones & Bartlett Learning, LLC an Ascend Learning Company
www.jblearning.com

3.3 Logic Gates

- The main thing to remember is that **combinations of gates implement Boolean functions**.
- The circuit below implements the Boolean function $F(x,y,z) = x + y'z$:



We simplify our Boolean expressions so that we can create simpler circuits.

© Odua Images/ShutterStock, Inc. Copyright © 2014 by Jones & Bartlett Learning, LLC an Ascend Learning Company
www.jblearning.com

3.5 Combinational Circuits

- We have designed a combinational circuit that implements the Boolean function:

$$F(X,Y,Z) = X+\overline{Y}Z$$

- Combinational logic circuits **produce** a specified **output** (almost) at the **instant** when input values are applied.
  - In a later section, we will explore circuits where this is not the case.

© Odua Images/ShutterStock, Inc. Copyright © 2014 by Jones & Bartlett Learning, LLC an Ascend Learning Company
www.jblearning.com

3.5 Combinational Circuits

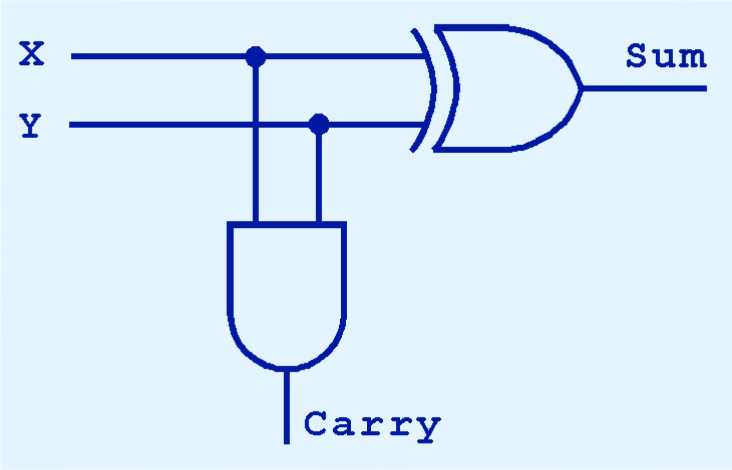
- Combinational logic circuits give us many **useful** devices.
- One of the simplest is the ***half adder***, which finds the **sum of two bits**.
- We can gain some insight as to the construction of a half adder by looking at its truth table, shown at the right.

Inputs		Outputs	
X	Y	Sum	Carry
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

© Odua Images/ShutterStock, Inc. Copyright © 2014 by Jones & Bartlett Learning, LLC an Ascend Learning Company
www.jblearning.com

3.5 Combinational Circuits

- As we see, the **sum** can be found using the **XOR** operation and the **carry** using the **AND** operation.



Inputs		Outputs	
X	Y	Sum	Carry
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

© Odua Images/ShutterStock, Inc. Copyright © 2014 by Jones & Bartlett Learning, LLC an Ascend Learning Company
www.jblearning.com

3.5 Combinational Circuits

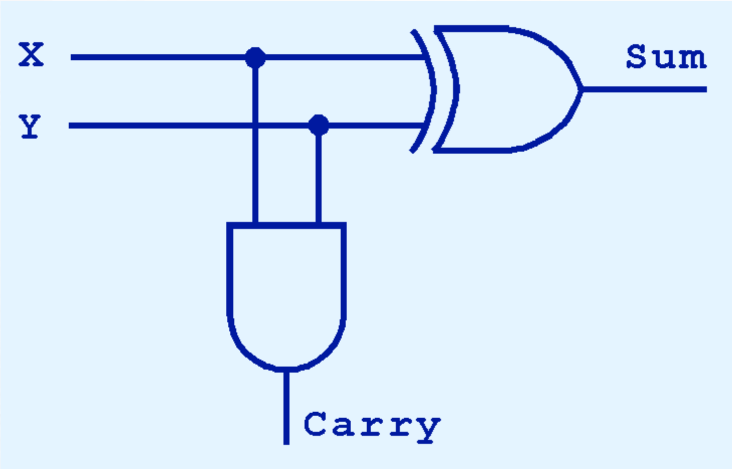
- We can extend our half adder to to a **full adder** by including gates for **processing the carry bit**.
- The truth table for a full adder is shown at the right.

Inputs			Outputs	
X	Y	Carry In	Sum	Carry Out
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

© Odua Images/ShutterStock, Inc. Copyright © 2014 by Jones & Bartlett Learning, LLC an Ascend Learning Company
www.jblearning.com

3.5 Combinational Circuits

- How can we change the half adder shown below to make it a full adder?

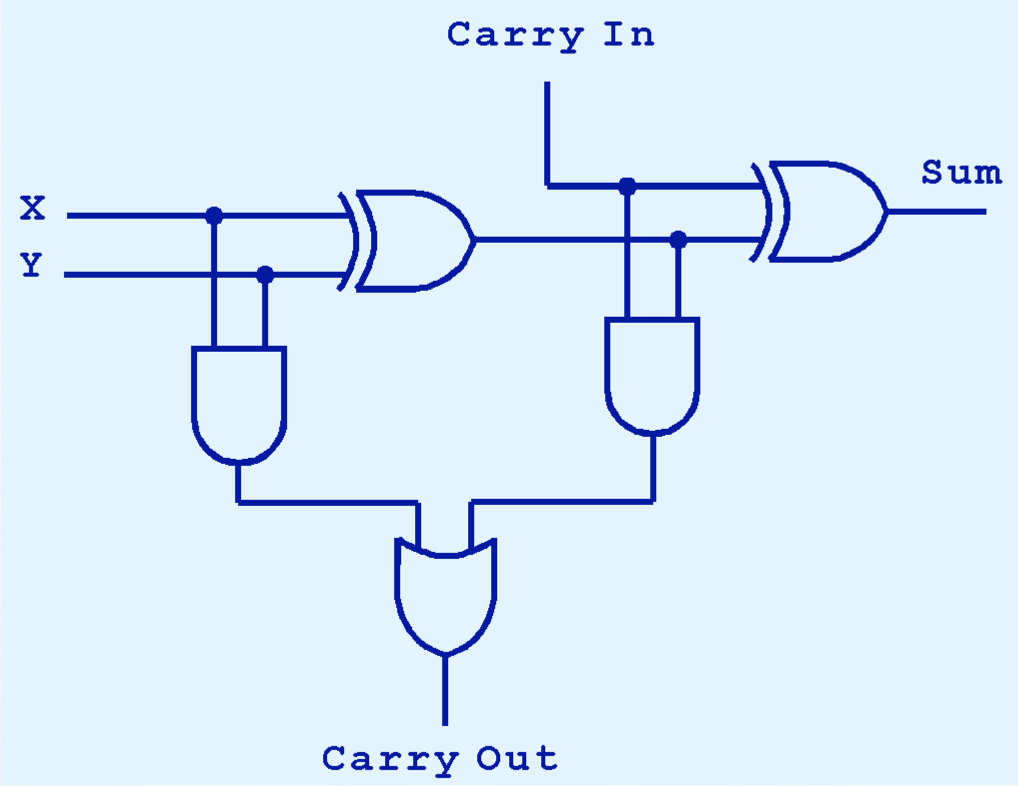


Inputs			Outputs	
X	Y	Carry In	Sum	Carry Out
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

© Odua Images/ShutterStock, Inc. Copyright © 2014 by Jones & Bartlett Learning, LLC an Ascend Learning Company
www.jblearning.com

3.5 Combinational Circuits

- Here's our completed full adder.

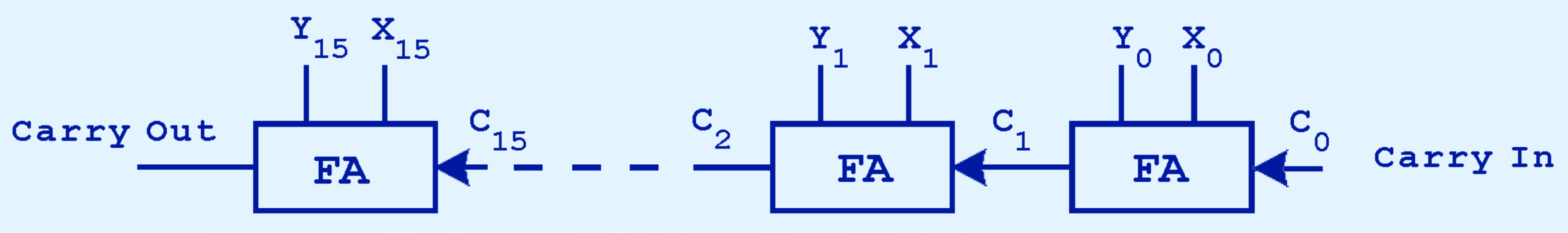


Inputs			Outputs	
X	Y	Carry In	Sum	Carry Out
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

© Odua Images/ShutterStock, Inc. Copyright © 2014 by Jones & Bartlett Learning, LLC an Ascend Learning Company
www.jblearning.com

3.5 Combinational Circuits

- Just as we combined half adders to make a full adder, **full adders can connected in series**.
- The **carry bit "ripples"** from one adder to the next; hence, this configuration is called a *ripple-carry adder*.

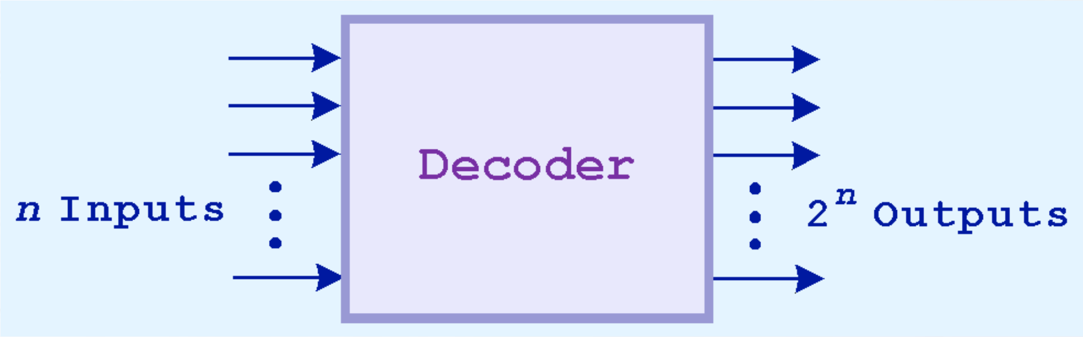


© Odua Images/ShutterStock, Inc. Copyright © 2014 by Jones & Bartlett Learning, LLC an Ascend Learning Company
www.jblearning.com

3.5 Combinational Circuits

- **Decoders** are another important type of **combinational circuit**.
- Among other things, they are **useful in selecting a memory location according a binary value** placed on the address lines of a memory bus.
- Address decoders with ***n* inputs** can select any of **2^n locations**.

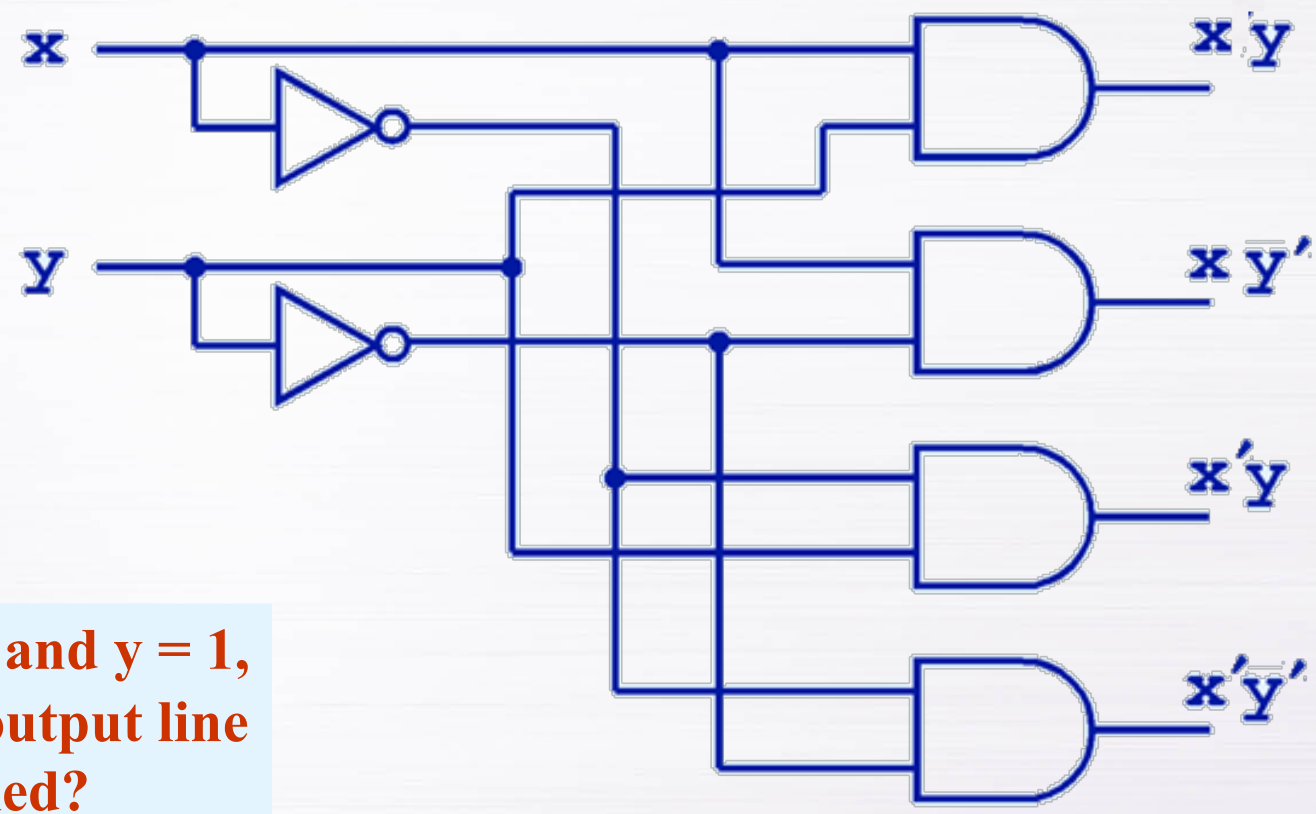
This is a block diagram for a decoder.



© Odua Images/ShutterStock, Inc. Copyright © 2014 by Jones & Bartlett Learning, LLC an Ascend Learning Company
www.jblearning.com

3.5 Combinational Circuits

- This is what a 2-to-4 decoder looks like on the inside.

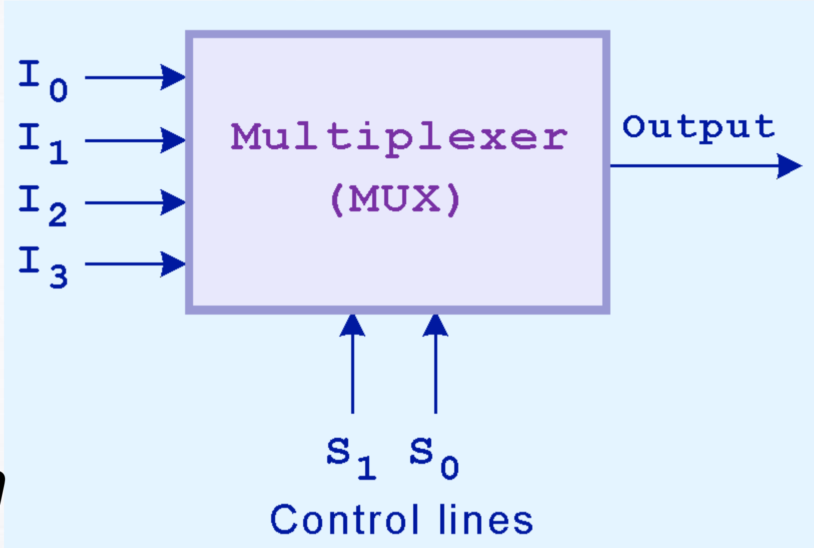


If x = 0 and y = 1, which output line is enabled?

© Odua Images/ShutterStock, Inc. Copyright © 2014 by Jones & Bartlett Learning, LLC an Ascend Learning Company
www.jblearning.com

3.5 Combinational Circuits

- A **multiplexer selects a single output from several inputs.**
- The particular **input chosen for output** is determined by the **value of the multiplexer's control lines**.
- To be able to select among *n* inputs, $\log_2 n$ control lines are needed.

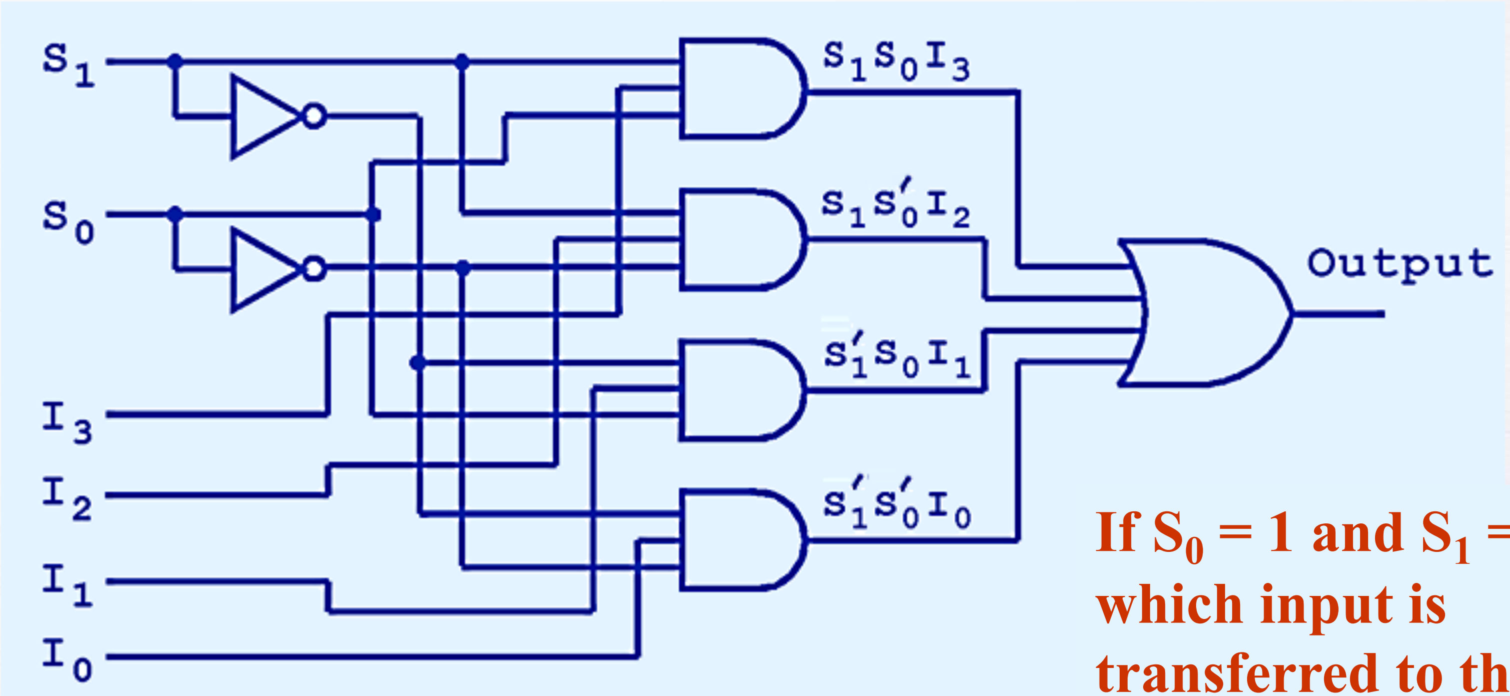


This is a block diagram for a multiplexer.

© Odua Images/ShutterStock, Inc. Copyright © 2014 by Jones & Bartlett Learning, LLC an Ascend Learning Company
www.jblearning.com

3.5 Combinational Circuits

- This is what a 4-to-1 multiplexer looks like on the inside.

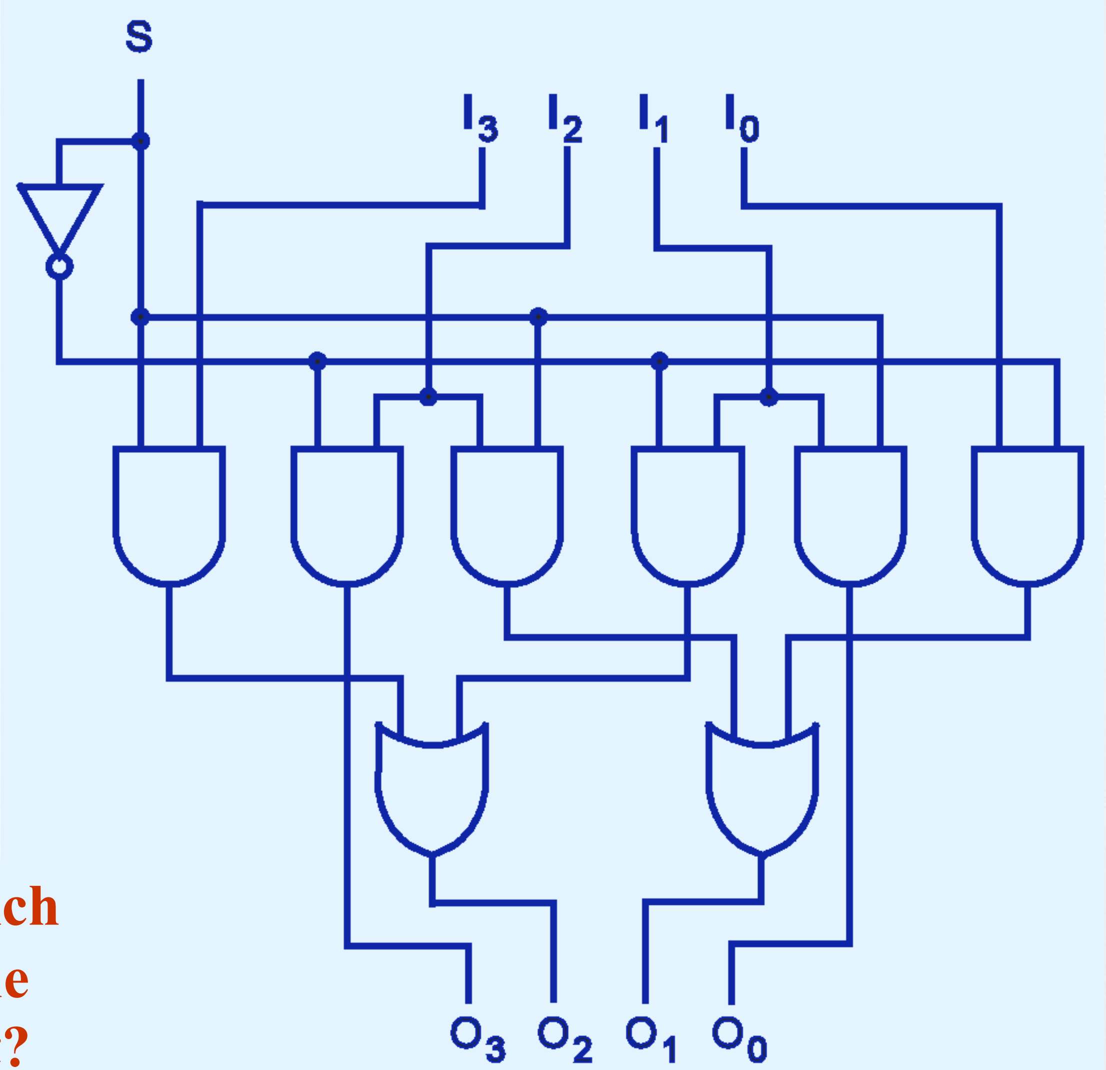


If $S_0 = 1$ and $S_1 = 0$, which input is transferred to the output?

© Odua Images/ShutterStock, Inc. Copyright © 2014 by Jones & Bartlett Learning, LLC an Ascend Learning Company
www.jblearning.com

3.5 Combinational Circuits

- This **shifter moves the bits** of a nibble one position to the **left or right**.



If S = 0, in which direction do the input bits shift?

© Odua Images/ShutterStock, Inc. Copyright © 2014 by Jones & Bartlett Learning, LLC an Ascend Learning Company
www.jblearning.com