

# Lecture No.1

## LECTURE OUTLINE

- 2-1 Decimal Numbers
- 2-2 Binary Numbers
- 2-3 Decimal-to-Binary Conversion
- 2-4 Binary Arithmetic

## LECTURE OBJECTIVES

- Review the decimal number system
- Count in the binary number system
- Convert from decimal to binary and from binary to decimal
- Apply arithmetic operations to binary numbers

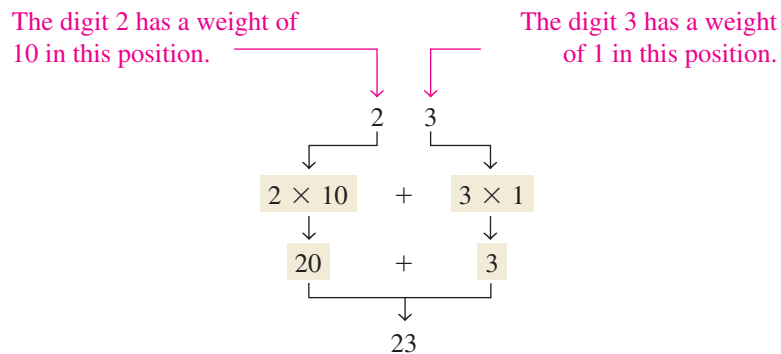
## 2-1 Decimal Numbers

You are familiar with the decimal number system because you use decimal numbers every day. Although decimal numbers are commonplace, their weighted structure is often not understood. In this section, the structure of decimal numbers is reviewed. This review will help you more easily understand the structure of the binary number system, which is important in computers and digital electronics.

After completing this section, you should be able to

- ◆ Explain why the decimal number system is a weighted system
- ◆ Explain how powers of ten are used in the decimal system
- ◆ Determine the weight of each digit in a decimal number

In the **decimal** number system each of the ten digits, 0 through 9, represents a certain quantity. As you know, the ten symbols (**digits**) do not limit you to expressing only ten different quantities because you use the various digits in appropriate positions within a number to indicate the magnitude of the quantity. You can express quantities up through nine before running out of digits; if you wish to express a quantity greater than nine, you use two or more digits, and the position of each digit within the number tells you the magnitude it represents. If, for example, you wish to express the quantity twenty-three, you use (by their respective positions in the number) the digit 2 to represent the quantity twenty and the digit 3 to represent the quantity three, as illustrated below.



The position of each digit in a decimal number indicates the magnitude of the quantity represented and can be assigned a **weight**. The weights for whole numbers are positive powers of ten that increase from right to left, beginning with  $10^0 = 1$ .

$$\dots 10^5 10^4 10^3 10^2 10^1 10^0$$

For fractional numbers, the weights are negative powers of ten that decrease from left to right beginning with  $10^{-1}$ .

$$10^2 10^1 10^0 . 10^{-1} 10^{-2} 10^{-3} \dots$$

↑  
Decimal point

The value of a decimal number is the sum of the digits after each digit has been multiplied by its weight, as Examples 2-1 and 2-2 illustrate.

**EXAMPLE 2-1**

Express the decimal number 47 as a sum of the values of each digit.

**Solution**

The digit 4 has a weight of 10, which is  $10^1$ , as indicated by its position. The digit 7 has a weight of 1, which is  $10^0$ , as indicated by its position.

$$\begin{aligned} 47 &= (4 \times 10^1) + (7 \times 10^0) \\ &= (4 \times 10) + (7 \times 1) = \mathbf{40} + \mathbf{7} \end{aligned}$$

**Related Problem\***

Determine the value of each digit in 939.

**EXAMPLE 2-2**

Express the decimal number 568.23 as a sum of the values of each digit.

**Solution**

The whole number digit 5 has a weight of 100, which is  $10^2$ , the digit 6 has a weight of 10, which is  $10^1$ , the digit 8 has a weight of 1, which is  $10^0$ , the fractional digit 2 has a weight of 0.1, which is  $10^{-1}$ , and the fractional digit 3 has a weight of 0.01, which is  $10^{-2}$ .

$$\begin{aligned} 568.23 &= (5 \times 10^2) + (6 \times 10^1) + (8 \times 10^0) + (2 \times 10^{-1}) + (3 \times 10^{-2}) \\ &= (5 \times 100) + (6 \times 10) + (8 \times 1) + (2 \times 0.1) + (3 \times 0.01) \\ &= \mathbf{500} + \mathbf{60} + \mathbf{8} + \mathbf{0.2} + \mathbf{0.03} \end{aligned}$$

**Related Problem**

Determine the value of each digit in 67.924.

**SECTION 2-1 CHECKUP**

Answers are at the end of the chapter.

- What weight does the digit 7 have in each of the following numbers?  
 (a) 1370    (b) 6725    (c) 7051    (d) 58.72
- Express each of the following decimal numbers as a sum of the products obtained by multiplying each digit by its appropriate weight:  
 (a) 51    (b) 137    (c) 1492    (d) 106.58

**2-2 Binary Numbers**

The binary number system is another way to represent quantities. It is less complicated than the decimal system because the binary system has only two digits. The decimal system with its ten digits is a base-ten system; the binary system with its two digits is a base-two system. The two binary digits (bits) are 1 and 0. The position of a 1 or 0 in a binary number indicates its weight, or value within the number, just as the position of a decimal digit determines the value of that digit. The weights in a binary number are based on powers of two.

After completing this section, you should be able to

- ♦ Count in binary
- ♦ Determine the largest decimal number that can be represented by a given number of bits
- ♦ Convert a binary number to a decimal number

## Counting in Binary

To learn to count in the binary system, first look at how you count in the decimal system. You start at zero and count up to nine before you run out of digits. You then start another digit position (to the left) and continue counting 10 through 99. At this point you have exhausted all two-digit combinations, so a third digit position is needed to count from 100 through 999.

A comparable situation occurs when you count in binary, except that you have only two digits, called *bits*. Begin counting: 0, 1. At this point you have used both digits, so include another digit position and continue: 10, 11. You have now exhausted all combinations of two digits, so a third position is required. With three digit positions you can continue to count: 100, 101, 110, and 111. Now you need a fourth digit position to continue, and so on. A binary count of zero through fifteen is shown in Table 2–1. Notice the patterns with which the 1s and 0s alternate in each column.

Decimal Number	Binary Number			
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0
9	1	0	0	1
10	1	0	1	0
11	1	0	1	1
12	1	1	0	0
13	1	1	0	1
14	1	1	1	0
15	1	1	1	1

As you have seen in Table 2–1, four bits are required to count from zero to 15. In general, with  $n$  bits you can count up to a number equal to  $2^n - 1$ .

$$\text{Largest decimal number} = 2^n - 1$$

For example, with five bits ( $n = 5$ ) you can count from zero to thirty-one.

$$2^5 - 1 = 32 - 1 = 31$$

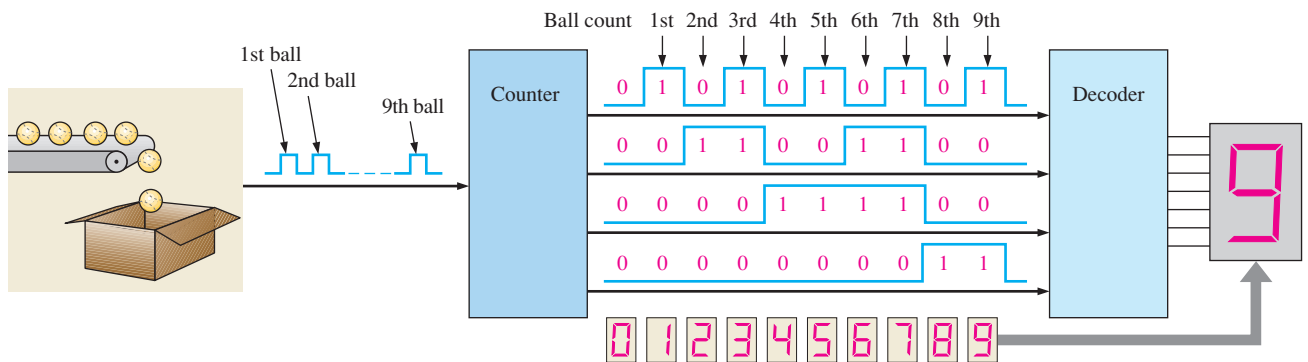
With six bits ( $n = 6$ ) you can count from zero to sixty-three.

$$2^6 - 1 = 64 - 1 = 63$$

## An Application

Learning to count in binary will help you to basically understand how digital circuits can be used to count events. Let's take a simple example of counting tennis balls going into a box from a conveyor belt. Assume that nine balls are to go into each box.

The counter in Figure 2-1 counts the pulses from a sensor that detects the passing of a ball and produces a sequence of logic levels (digital waveforms) on each of its four parallel outputs. Each set of logic levels represents a 4-bit binary number (HIGH = 1 and LOW = 0), as indicated. As the decoder receives these waveforms, it decodes each set of four bits and converts it to the corresponding decimal number in the 7-segment display. When the counter gets to the binary state of 1001, it has counted nine tennis balls, the display shows decimal 9, and a new box is moved under the conveyor belt. Then the counter goes back to its zero state (0000), and the process starts over. (The number 9 was used only in the interest of single-digit simplicity.)



**FIGURE 2-1** Illustration of a simple binary counting application.

## The Weighting Structure of Binary Numbers

A binary number is a weighted number. The right-most bit is the **LSB** (least significant bit) in a binary whole number and has a weight of  $2^0 = 1$ . The weights increase from right to left by a power of two for each bit. The left-most bit is the **MSB** (most significant bit); its weight depends on the size of the binary number.

Fractional numbers can also be represented in binary by placing bits to the right of the binary point, just as fractional decimal digits are placed to the right of the decimal point. The left-most bit is the MSB in a binary fractional number and has a weight of  $2^{-1} = 0.5$ . The fractional weights decrease from left to right by a negative power of two for each bit.

The weight structure of a binary number is

$$2^{n-1} \dots 2^3 2^2 2^1 2^0 . 2^{-1} 2^{-2} \dots 2^{-n}$$

↑ Binary point

where  $n$  is the number of bits from the binary point. Thus, all the bits to the left of the binary point have weights that are positive powers of two, as previously discussed for whole numbers. All bits to the right of the binary point have weights that are negative powers of two, or fractional weights.

The powers of two and their equivalent decimal weights for an 8-bit binary whole number and a 6-bit binary fractional number are shown in Table 2-2. Notice that the weight doubles for each positive power of two and that the weight is halved for each negative power of two. You can easily extend the table by doubling the weight of the most significant positive power of two and halving the weight of the least significant negative power of two; for example,  $2^9 = 512$  and  $2^{-7} = 0.0078125$ .

**TABLE 2-2**

Binary weights.

Positive Powers of Two (Whole Numbers)									Negative Powers of Two (Fractional Number)					
$2^8$	$2^7$	$2^6$	$2^5$	$2^4$	$2^3$	$2^2$	$2^1$	$2^0$	$2^{-1}$	$2^{-2}$	$2^{-3}$	$2^{-4}$	$2^{-5}$	$2^{-6}$
256	128	64	32	16	8	4	2	1	1/2	1/4	1/8	1/16	1/32	1/64
									0.5	0.25	0.125	0.625	0.03125	0.015625

## Binary-to-Decimal Conversion

The decimal value of any binary number can be found by adding the weights of all bits that are 1 and discarding the weights of all bits that are 0.

### EXAMPLE 2-3

Convert the binary whole number 1101101 to decimal.

#### Solution

Determine the weight of each bit that is a 1, and then find the sum of the weights to get the decimal number.

$$\begin{aligned}
 \text{Weight: } & 2^6 \ 2^5 \ 2^4 \ 2^3 \ 2^2 \ 2^1 \ 2^0 \\
 \text{Binary number: } & 1 \ 1 \ 0 \ 1 \ 1 \ 0 \ 1 \\
 1101101 = & 2^6 + 2^5 + 2^3 + 2^2 + 2^0 \\
 = & 64 + 32 + 8 + 4 + 1 = \mathbf{109}
 \end{aligned}$$

#### Related Problem

Convert the binary number 10010001 to decimal.

### EXAMPLE 2-4

Convert the fractional binary number 0.1011 to decimal.

#### Solution

Determine the weight of each bit that is a 1, and then sum the weights to get the decimal fraction.

$$\begin{aligned}
 \text{Weight: } & 2^{-1} \ 2^{-2} \ 2^{-3} \ 2^{-4} \\
 \text{Binary number: } & 0 \ . \ 1 \ 0 \ 1 \ 1 \\
 0.1011 = & 2^{-1} + 2^{-3} + 2^{-4} \\
 = & 0.5 + 0.125 + 0.0625 = \mathbf{0.6875}
 \end{aligned}$$

#### Related Problem

Convert the binary number 10.111 to decimal.

### SECTION 2-2 CHECKUP

1. What is the largest decimal number that can be represented in binary with eight bits?
2. Determine the weight of the 1 in the binary number 10000.
3. Convert the binary number 10111101.011 to decimal.

## 2-3 Decimal-to-Binary Conversion

In Section 2-2 you learned how to convert a binary number to the equivalent decimal number. Now you will learn two ways of converting from a decimal number to a binary number.

After completing this section, you should be able to

- ◆ Convert a decimal number to binary using the sum-of-weights method
- ◆ Convert a decimal whole number to binary using the repeated division-by-2 method
- ◆ Convert a decimal fraction to binary using the repeated multiplication-by-2 method

### Sum-of-Weights Method

One way to find the binary number that is equivalent to a given decimal number is to determine the set of binary weights whose sum is equal to the decimal number. An easy way to remember binary weights is that the lowest is 1, which is  $2^0$ , and that by doubling any weight, you get the next higher weight; thus, a list of seven binary weights would be 64, 32, 16, 8, 4, 2, 1 as you learned in the last section. The decimal number 9, for example, can be expressed as the sum of binary weights as follows:

$$9 = 8 + 1 \quad \text{or} \quad 9 = 2^3 + 2^0$$

Placing 1s in the appropriate weight positions,  $2^3$  and  $2^0$ , and 0s in the  $2^2$  and  $2^1$  positions determines the binary number for decimal 9.

$$\begin{array}{cccc} 2^3 & 2^2 & 2^1 & 2^0 \\ 1 & 0 & 0 & 1 \end{array} \quad \text{Binary number for decimal 9}$$

#### EXAMPLE 2-5

Convert the following decimal numbers to binary:

- (a) 12      (b) 25  
(c) 58      (d) 82

#### Solution

- (a)  $12 = 8 + 4 = 2^3 + 2^2$  —————→ **1100**  
 (b)  $25 = 16 + 8 + 1 = 2^4 + 2^3 + 2^0$  —————→ **11001**  
 (c)  $58 = 32 + 16 + 8 + 2 = 2^5 + 2^4 + 2^3 + 2^1$  —————→ **111010**  
 (d)  $82 = 64 + 16 + 2 = 2^6 + 2^4 + 2^1$  —————→ **1010010**

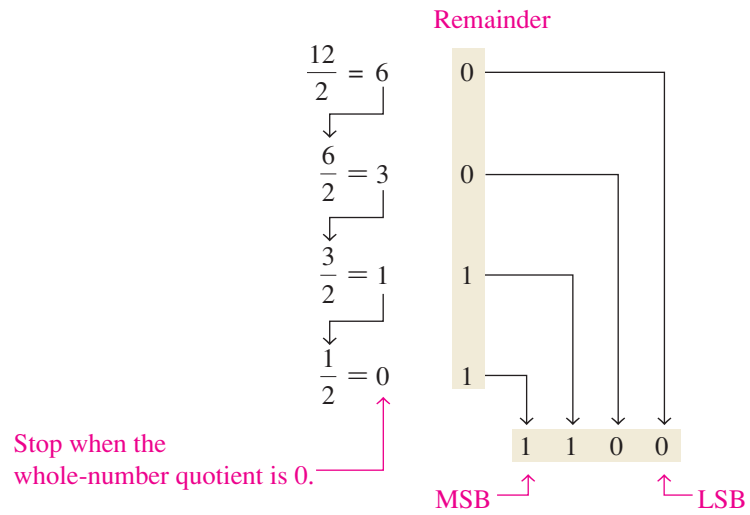
#### Related Problem

Convert the decimal number 125 to binary.

### Repeated Division-by-2 Method

A systematic method of converting whole numbers from decimal to binary is the *repeated division-by-2* process. For example, to convert the decimal number 12 to binary, begin by dividing 12 by 2. Then divide each resulting quotient by 2 until there is a 0 whole-number quotient. The **remainders** generated by each division form the binary number. The first remainder to be produced is the LSB (least significant bit) in the binary number, and the

last remainder to be produced is the MSB (most significant bit). This procedure is illustrated as follows for converting the decimal number 12 to binary.

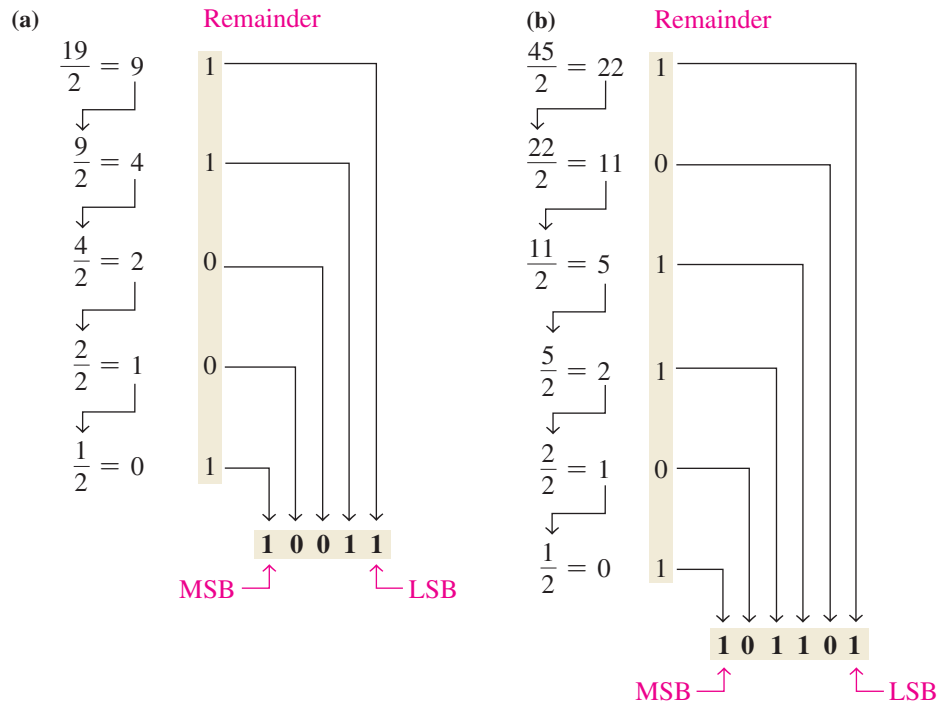


#### EXAMPLE 2-6

Convert the following decimal numbers to binary:

- (a) 19      (b) 45

#### Solution



#### Related Problem

Convert decimal number 39 to binary.



## Converting Decimal Fractions to Binary

Examples 2–5 and 2–6 demonstrated whole-number conversions. Now let's look at fractional conversions. An easy way to remember fractional binary weights is that the most significant weight is 0.5, which is  $2^{-1}$ , and that by halving any weight, you get the next lower weight; thus a list of four fractional binary weights would be 0.5, 0.25, 0.125, 0.0625.

### Sum-of-Weights

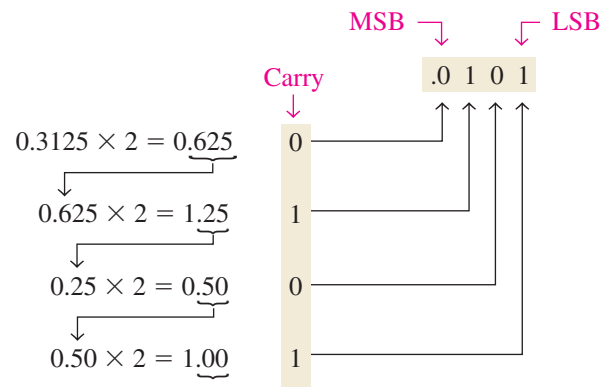
The sum-of-weights method can be applied to fractional decimal numbers, as shown in the following example:

$$0.625 = 0.5 + 0.125 = 2^{-1} + 2^{-3} = 0.101$$

There is a 1 in the  $2^{-1}$  position, a 0 in the  $2^{-2}$  position, and a 1 in the  $2^{-3}$  position.

### Repeated Multiplication by 2

As you have seen, decimal whole numbers can be converted to binary by repeated division by 2. Decimal fractions can be converted to binary by repeated multiplication by 2. For example, to convert the decimal fraction 0.3125 to binary, begin by multiplying 0.3125 by 2 and then multiplying each resulting fractional part of the product by 2 until the fractional product is zero or until the desired number of decimal places is reached. The carry digits, or **carries**, generated by the multiplications produce the binary number. The first carry produced is the MSB, and the last carry is the LSB. This procedure is illustrated as follows:



Continue to the desired number of decimal places or stop when the fractional part is all zeros.

#### SECTION 2-3 CHECKUP

- Convert each decimal number to binary by using the sum-of-weights method:
  - 23
  - 57
  - 45.5
- Convert each decimal number to binary by using the repeated division-by-2 method (repeated multiplication-by-2 for fractions):
  - 14
  - 21
  - 0.375

## 2-4 Binary Arithmetic

Binary arithmetic is essential in all digital computers and in many other types of digital systems. To understand digital systems, you must know the basics of binary addition, subtraction, multiplication, and division. This section provides an introduction that will be expanded in later sections.

After completing this section, you should be able to

- ◆ Add binary numbers
- ◆ Subtract binary numbers
- ◆ Multiply binary numbers
- ◆ Divide binary numbers

### Binary Addition

The four basic rules for adding binary digits (bits) are as follows:

$0 + 0 = 0$	Sum of 0 with a carry of 0
$0 + 1 = 1$	Sum of 1 with a carry of 0
$1 + 0 = 1$	Sum of 1 with a carry of 0
$1 + 1 = 10$	Sum of 0 with a carry of 1

Notice that the first three rules result in a single bit and in the fourth rule the addition of two 1s yields a binary two (10). When binary numbers are added, the last condition creates a sum of 0 in a given column and a carry of 1 over to the next column to the left, as illustrated in the following addition of  $11 + 1$ :

$$\begin{array}{r}
 \text{Carry} \quad \text{Carry} \\
 \begin{array}{r}
 1 \leftarrow 1 \leftarrow \\
 0 \quad 1 \quad 1 \\
 + 0 \quad 0 \quad 1 \\
 \hline
 1 \quad 0 \quad 0
 \end{array}
 \end{array}$$

In the right column,  $1 + 1 = 0$  with a carry of 1 to the next column to the left. In the middle column,  $1 + 1 + 0 = 0$  with a carry of 1 to the next column to the left. In the left column,  $1 + 0 + 0 = 1$ .

When there is a carry of 1, you have a situation in which three bits are being added (a bit in each of the two numbers and a carry bit). This situation is illustrated as follows:

Carry bits	↓	
1	+	$0 + 0 = 01$ Sum of 1 with a carry of 0
1	+	$1 + 0 = 10$ Sum of 0 with a carry of 1
1	+	$0 + 1 = 10$ Sum of 0 with a carry of 1
1	+	$1 + 1 = 11$ Sum of 1 with a carry of 1

#### EXAMPLE 2-7

Add the following binary numbers:

- (a)  $11 + 11$       (b)  $100 + 10$   
 (c)  $111 + 11$       (d)  $110 + 100$

**Solution**

The equivalent decimal addition is also shown for reference.

$$\begin{array}{r}
 \text{(a)} \quad 11 \quad 3 \\
 + 11 \quad + 3 \\
 \hline
 110 \quad 6
 \end{array}
 \qquad
 \begin{array}{r}
 \text{(b)} \quad 100 \quad 4 \\
 + 10 \quad + 2 \\
 \hline
 110 \quad 6
 \end{array}$$

$$\begin{array}{r}
 \text{(c)} \quad 111 \quad 7 \\
 + 11 \quad + 3 \\
 \hline
 1010 \quad 10
 \end{array}
 \qquad
 \begin{array}{r}
 \text{(d)} \quad 110 \quad 6 \\
 + 100 \quad + 4 \\
 \hline
 1010 \quad 10
 \end{array}$$

**Related Problem**

Add 1111 and 1100.

**Binary Subtraction**

The four basic rules for subtracting bits are as follows:

$$\begin{array}{l}
 0 - 0 = 0 \\
 1 - 1 = 0 \\
 1 - 0 = 1 \\
 10 - 1 = 1 \quad 0 - 1 \text{ with a borrow of } 1
 \end{array}$$

When subtracting numbers, you sometimes have to borrow from the next column to the left. A borrow is required in binary only when you try to subtract a 1 from a 0. In this case, when a 1 is borrowed from the next column to the left, a 10 is created in the column being subtracted, and the last of the four basic rules just listed must be applied. Examples 2–8 and 2–9 illustrate binary subtraction; the equivalent decimal subtractions are also shown.

**EXAMPLE 2–8**

Perform the following binary subtractions:

$$\text{(a)} \quad 11 - 01 \qquad \text{(b)} \quad 11 - 10$$

**Solution**

$$\begin{array}{r}
 \text{(a)} \quad 11 \quad 3 \\
 - 01 \quad - 1 \\
 \hline
 10 \quad 2
 \end{array}
 \qquad
 \begin{array}{r}
 \text{(b)} \quad 11 \quad 3 \\
 - 10 \quad - 2 \\
 \hline
 01 \quad 1
 \end{array}$$

No borrows were required in this example. The binary number 01 is the same as 1.

**Related Problem**

Subtract 100 from 111.

**EXAMPLE 2–9**

Subtract 011 from 101.

**Solution**

$$\begin{array}{r}
 101 \quad 5 \\
 - 011 \quad - 3 \\
 \hline
 010 \quad 2
 \end{array}$$

Let's examine exactly what was done to subtract the two binary numbers since a borrow is required. Begin with the right column.

<p>Left column: When a 1 is borrowed, a 0 is left, so <math>0 - 0 = 0</math>.</p>	$\begin{array}{r} 0 \\ 1^1 01 \\ - 011 \\ \hline 010 \end{array}$	<p>Middle column: Borrow 1 from next column to the left, making a 10 in this column, then <math>10 - 1 = 1</math>.</p> <p>Right column: <math>1 - 1 = 0</math></p>
---	---	--

### Related Problem

Subtract 101 from 110.

## Binary Multiplication

The four basic rules for multiplying bits are as follows:

$0 \times 0 = 0$
$0 \times 1 = 0$
$1 \times 0 = 0$
$1 \times 1 = 1$

Multiplication is performed with binary numbers in the same manner as with decimal numbers. It involves forming partial products, shifting each successive partial product left one place, and then adding all the partial products. Example 2-10 illustrates the procedure; the equivalent decimal multiplications are shown for reference.

### EXAMPLE 2-10

Perform the following binary multiplications:

(a)  $11 \times 11$       (b)  $101 \times 111$

### Solution

(a)	$\begin{array}{r} 11 \\ \times 11 \\ \hline 11 \\ +11 \\ \hline 1001 \end{array}$	(b)	$\begin{array}{r} 111 \\ \times 101 \\ \hline 111 \\ 000 \\ +111 \\ \hline 100011 \end{array}$
-----	---	-----	--

### Related Problem

Multiply  $1101 \times 1010$ .

## Binary Division

Division in binary follows the same procedure as division in decimal, as Example 2-11 illustrates. The equivalent decimal divisions are also given.

### EXAMPLE 2-11

Perform the following binary divisions:

(a)  $110 \div 11$       (b)  $110 \div 10$

**Solution**

$$\begin{array}{r}
 \mathbf{10} \quad 2 \\
 \text{(a) } 11 \overline{)110} \quad 3 \overline{)6} \\
 \underline{11} \quad \underline{6} \\
 000 \quad 0
 \end{array}
 \qquad
 \begin{array}{r}
 \mathbf{11} \quad 3 \\
 \text{(b) } 10 \overline{)110} \quad 2 \overline{)6} \\
 \underline{10} \quad \underline{6} \\
 10 \quad 0 \\
 \underline{10} \\
 00
 \end{array}$$

**Related Problem**

Divide 1100 by 100.

**SECTION 2-4 CHECKUP**

- Perform the following binary additions:  
 (a)  $1101 + 1010$       (b)  $10111 + 01101$
- Perform the following binary subtractions:  
 (a)  $1101 - 0100$       (b)  $1001 - 0111$
- Perform the indicated binary operations:  
 (a)  $110 \times 111$       (b)  $1100 \div 011$