and throughout the written material that supports the system, so it is very important that you are able to interpret values in any system and convert between any of these numeric representations. Other codes that use 1s and 0s to represent things such as alphanumeric characters will be covered because they are so common in digital systems.

2-1 BINARY-TO-DECIMAL CONVERSIONS

OUTCOMES

Upon completion of this section, you will be able to:

- Convert binary numbers to decimal.
- Identify the weight of each bit in a binary number.

As explained in Chapter 1, the binary number system is a positional system where each binary digit (bit) carries a certain weight based on its position relative to the LSB. Any binary number can be converted to its decimal equivalent simply by summing together the weights of the various positions in the binary number that contain a 1. To illustrate, let's change 11011_2 to its decimal equivalent.

$$\begin{aligned} &1 \quad\; 1 \quad\; 0 \quad\; 1 \quad\; 1_2 \\ &2^4 + 2^3 + 0 + 2^1 + 2^0 = 16 + 8 + 2 + 1 \\ &\qquad\qquad\qquad\qquad\;\; = 27_{10} \end{aligned}$$

Let's try another example with a greater number of bits:

$$\begin{aligned} &1 \quad 0 \quad 1 \quad 1 \quad 0 \quad 1 \quad 0 \quad 1_2 = \\ &2^7 + 0 + 2^5 + 2^4 + 0 + 2^2 + 0 + 2^0 = 181_{10} \end{aligned}$$

Note that the procedure is to find the weights (i.e., powers of 2) for each bit position that contains a 1, and then to add them up. Also note that the MSB has a weight of 2^7 even though it is the eighth bit; this is because the LSB is the first bit and has a weight of 2^0.

Another method of binary-to-decimal conversion that avoids the addition of large numbers and keeping track of column weights is called the double-dabble method. The procedure is as follows:

1. Write down the left-most 1 in the binary number.
2. Double it and add the next bit to the right.
3. Write down the result under the next bit.
4. Continue with steps 2 and 3 until finished with the binary number.

Let's use the same binary numbers to verify this method.

Given: $1 \quad 1 \quad 0 \quad 1 \quad 1_2$

Results:

$$\begin{aligned} &1 \times 2 = 2 \\ &\underline{+\,1} \\ &3 \times 2 = 6 \\ &\underline{+\,0} \\ &6 \times 2 = 12 \\ &\underline{+1} \\ &13 \times 2 = 26 \\ &\underline{+1} \\ &27_{10} \end{aligned}$$

Given: $1 \quad 0 \quad 1 \quad 1 \quad 0 \quad 1 \quad 0 \quad 1_2$

Results: $1 \rightarrow \; 2 \rightarrow \; 5 \rightarrow \; 11 \rightarrow \; 22 \rightarrow \; 45 \rightarrow \; 90 \rightarrow \; \mathbf{181_{10}}$

OUTCOME ASSESSMENT QUESTIONS

1. Convert 100011011011_2 to its decimal equivalent by adding the products of digits and weights.
2. What is the weight of the MSB of a 16-bit number?
3. Repeat the conversion in Question 1 using the double-dabble method.

2-2 DECIMAL-TO-BINARY CONVERSIONS

OUTCOMES

Upon completion of this section, you will be able to:

- Convert decimal numbers to binary.
- Identify the number of bits needed for a given range of values.
- Identify the range of values given the number of bits.

There are two ways to convert a decimal *whole* number to its equivalent binary-system representation. The first method is the reverse of the first process described in Section 2-1. The decimal number is simply expressed as a sum of powers of 2, and then 1s and 0s are written in the appropriate bit positions. To illustrate:

$$
\begin{aligned}
45_{10} = 32 + 8 + 4 + 1 &= 2^5 + 0 + 2^3 + 2^2 + 0 + 2^0 \\
&= 1 \quad 0 \quad 1 \quad 1 \quad 0 \quad 1_2
\end{aligned}
$$

Note that a 0 is placed in the 2^1 and 2^4 positions, since all positions must be accounted for. Another example is the following:

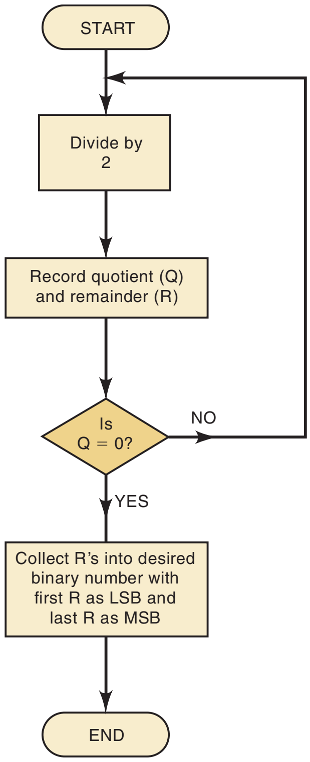
$$
\begin{aligned}
76_{10} = 64 + 8 + 4 &= 2^6 + 0 + 0 + 2^3 + 2^2 + 0 + 0 \\
&= 1 \quad 0 \quad 0 \quad 1 \quad 1 \quad 0 \quad 0_2
\end{aligned}
$$

Another method for converting decimal integers uses repeated division by 2. The conversion, illustrated below for 25_{10}, requires repeatedly dividing the decimal number by 2 and writing down the remainder after each division until a quotient of 0 is obtained. Note that the binary result is obtained by writing the first remainder as the LSB and the last remainder as the MSB. This process, diagrammed in the flowchart of Figure 2-1, can also be used to convert from decimal to any other number system, as we shall see.

$$
\begin{aligned}
\frac{25}{2} &= 12 + \text{remainder of } 1 \quad \text{LSB} \\
\frac{12}{2} &= 6 + \text{remainder of } 0 \\
\frac{6}{2} &= 3 + \text{remainder of } 0 \\
\frac{3}{2} &= 1 + \text{remainder of } 1 \\
\frac{1}{2} &= 0 + \text{remainder of } 1 \quad \text{MSB}
\end{aligned}
$$

$$25_{10} = 1 \quad 1 \quad 0 \quad 0 \quad 1_2$$

FIGURE 2-1 Flowchart for repeated-division method of decimal-to-binary conversion of integers. The same process can be used to convert a decimal integer to any other number system.



CALCULATOR HINT:

If you use a calculator to perform the divisions by 2, you can tell whether the remainder is 0 or 1 by whether or not the result has a fractional part. For instance, $\frac{25}{2}$ would produce 12.5. Since there is a fractional part (the .5), the remainder is a 1. If there were no fractional part, such as $\frac{12}{2} = 6$, then the remainder would be 0. Example 2-1 illustrates this.

EXAMPLE 2-1

Convert 37_{10} to binary. Try to do it on your own before you look at the solution.

Solution

$$\frac{37}{2} = 18.5 \longrightarrow \text{remainder of 1 (LSB)}$$

$$\frac{18}{2} = 9.0 \longrightarrow \quad 0$$

$$\frac{9}{2} = 4.5 \longrightarrow \quad 1$$

$$\frac{4}{2} = 2.0 \longrightarrow \quad 0$$

$$\frac{2}{2} = 1.0 \longrightarrow \quad 0$$

$$\frac{1}{2} = 0.5 \longrightarrow \quad 1 \text{ (MSB)}$$

Thus, $37^{10} = \mathbf{100101_2}$.

Counting Range

Recall that using N bits, we can count through 2^N different decimal numbers ranging from 0 to $2^N - 1$. For example, for $N = 4$, we can count from 0000_2 to 1111_2, which is 0_{10} to 15_{10}, for a total of 16 different numbers. Here, the largest decimal value is $2^4 - 1 = 15$, and there are 2^4 different numbers.

In general, then, we can state:

Using N bits, we can represent decimal numbers ranging from 0 to $2^N - 1$, a total of 2^N different numbers.

EXAMPLE 2-2

(a) What is the total range of decimal values that can be represented in eight bits?

(b) How many bits are needed to represent decimal values ranging from 0 to 12,500?

Solution

(a) Here we have $N = 8$. Thus, we can represent decimal numbers from 0 to $2^8 - 1 = 255$. We can verify this by checking to see that 11111111_2 converts to 255_{10}.

(b) With 13 bits, we can count from decimal 0 to $2^{13} - 1 = 8191$. With 14 bits, we can count from 0 to $2^{14} - 1 = 16{,}383$. Clearly, 13 bits aren't enough, but 14 bits will get us up beyond 12,500. Thus, the required number of bits is **14**.

OUTCOME ASSESSMENT QUESTIONS

1. Convert 83_{10} to binary using both methods.
2. Convert 729_{10} to binary using both methods. Check your answer by converting back to decimal.
3. How many bits are required to count up to decimal 1 million?

2-3 HEXADECIMAL NUMBER SYSTEM

OUTCOMES

Upon completion of this section, you will be able to:

- Identify the weight of each hexadecimal digit.
- Convert between any of the following number systems: binary, decimal, hexadecimal.
- Count in hexadecimal.
- Identify range of numbers (in all systems) for a given number of digits.
- Identify number of digits needed for a given range of values.
- Memorize the value of each hexadecimal digit in binary and decimal.
- Cite advantages of the hexadecimal number system.

The **hexadecimal number system** uses base 16. Thus, it has 16 possible digit symbols. It uses the digits 0 through 9 plus the letters A, B, C, D, E, and F as the 16 digit symbols. The digit positions are weighted as powers of 16 as shown below, rather than as powers of 10 as in the decimal system.

16^4	16^3	16^2	16^1	16^0	16^{-1}	16^{-2}	16^{-3}	16^{-4}

Hexadecimal point

Table 2-1 shows the relationships among hexadecimal, decimal, and binary. Note that each hexadecimal digit represents a group of four binary digits. It is important to remember that hex (abbreviation for "hexadecimal") digits A through F are equivalent to the decimal values 10 through 15.

TABLE 2-1

Hexadecimal	Decimal	Binary
0	0	0000
1	1	0001
2	2	0010
3	3	0011
4	4	0100
5	5	0101
6	6	0110
7	7	0111
8	8	1000
9	9	1001
A	10	1010
B	11	1011
C	12	1100
D	13	1101
E	14	1110
F	15	1111

Hex-to-Decimal Conversion

A hex number can be converted to its decimal equivalent by using the fact that each hex digit position has a weight that is a power of 16. The LSD has a weight of $16^0 = 1$; the next higher digit position has a weight of $16^1 = 16$; the next has a weight of $16^2 = 256$; and so on. The conversion process is demonstrated in the examples below.

CALCULATOR HINT:

You can use the y^x calculator function to evaluate the powers of 16.

$$\begin{aligned} 356_{16} &= 3 \times 16^2 + 5 \times 16^1 + 6 \times 16^0 \\ &= 768 + 80 + 6 \\ &= 854_{10} \end{aligned}$$

$$\begin{aligned} 2AF_{16} &= 2 \times 16^2 + 10 \times 16^1 + 15 \times 16^0 \\ &= 512 + 160 + 15 \\ &= 687_{10} \end{aligned}$$

Note that in the second example, the value 10 was substituted for A and the value 15 for F in the conversion to decimal.

For practice, verify that $1BC2_{16}$ is equal to 7106_{10}.

Decimal-to-Hex Conversion

Recall that we did decimal-to-binary conversion using repeated division by 2. Likewise, decimal-to-hex conversion can be done using repeated division by 16 (Figure 2-1). The following example contains two illustrations of this conversion.

EXAMPLE 2-3

(a) Convert 423_{10} to hex.

Solution

$$\frac{423}{16} = 26 + \text{remainder of } 7$$

$$\frac{26}{16} = 1 + \text{remainder of } 10$$

$$\frac{1}{16} = 0 + \text{remainder of } 1$$

$$423_{10} = 1A7_{16}$$

(b) Convert 214_{10} to hex.

Solution

$$\frac{214}{16} = 13 + \text{remainder of } 6$$

$$\frac{13}{16} = 0 + \text{remainder of } 13$$

$$214_{10} = D6_{16}$$

Again note that the remainders of the division processes form the digits of the hex number. Also note that any remainders that are greater than 9 are represented by the letters A through F.

CALCULATOR HINT:

If a calculator is used to perform the divisions in the conversion process, the results will include a decimal fraction instead of a remainder. The remainder can be obtained by multiplying the fraction by 16. To illustrate, in Example 2-3(b), the calculator would have produced

$$\frac{214}{16} = 13.375$$

The remainder becomes $(0.375) \times 16 = 6$.

Hex-to-Binary Conversion

The hexadecimal number system is used primarily as a "shorthand" method for representing binary numbers. It is a relatively simple matter to convert a hex number to binary. *Each* hex digit is converted to its four-bit binary equivalent (Table 2-1). This is illustrated below for $9F2_{16}$.

$$
\begin{aligned}
9F2_{16} &= \quad 9 \qquad\qquad\quad F \qquad\qquad\quad 2 \\
&\quad\ \ \downarrow \qquad\qquad\quad \downarrow \qquad\qquad\quad \downarrow \\
&= 1\ 0\ 0\ 1 \quad 1\ 1\ 1\ 1 \quad 0\ 0\ 1\ 0 \\
&= 100111110010_2
\end{aligned}
$$

For practice, verify that $BA6_{16} = 101110100110_2$.

Binary-to-Hex Conversion

Conversion from binary to hex is just the reverse of the process above. The binary number is grouped into groups of *four* bits, and each group is converted to its equivalent hex digit. Zeros (shown shaded) are added, as needed, to complete a four-bit group.

$$
\begin{aligned}
1\,1\,1\,0\,1\,0\,0\,1\,1\,0_2 &= \underbrace{\boxed{0\,0}\,1\,1}_{3}\ \underbrace{1\,0\,1\,0}_{A}\ \underbrace{0\,1\,1\,0}_{6} \\
&= 3A6_{16}
\end{aligned}
$$

To perform these conversions between hex and binary, it is necessary to know the four-bit binary numbers (0000 through 1111) and their equivalent hex digits. Once these are mastered, the conversions can be performed quickly without the need for any calculations. This is why hex is so useful in representing large binary numbers.

For practice, verify that $101011111_2 = 15F_{16}$.

Counting in Hexadecimal

When counting in hex, each digit position can be incremented (increased by 1) from 0 to F. Once a digit position reaches the value F, it is reset to 0, and the next digit position is incremented. This is illustrated in the following hex counting sequences:

(a) 38, 39, 3A, 3B, 3C, 3D, 3E, 3F, 40, 41, 42
(b) 6F8, 6F9, 6FA, 6FB, 6FC, 6FD, 6FE, 6FF, 700

Note that when there is a 9 in a digit position, it becomes an A when it is incremented.

With N hex digit positions, we can count from decimal 0 to $16^N - 1$, for a total of 16^N different values. For example, with three hex digits, we can count from 000_{16} to FFF_{16}, which is 0_{10} to 4095_{10}, for a total of $4096 = 16^3$ different values.

Usefulness of Hex

Hex is often used in a digital system as sort of a "shorthand" way to represent strings of bits. In computer work, strings as long as 64 bits are not uncommon. These binary strings do not always represent a numerical value, but—as you will find out—can be some type of code that conveys nonnumerical information. When dealing with a large number of bits, it is more convenient and less error-prone to write the binary numbers in hex and, as we have seen, it is relatively easy to convert back and forth between binary and hex. To illustrate the advantage of hex representation of a binary string, suppose you had in front of you a printout of the contents of 50 memory locations, each of which was a 16-bit number, and you were checking it against a list. Would you rather check 50 numbers like this one: 0110111001100111, or 50 numbers like this one: 6E67? And which one would you be more apt to read incorrectly? It is important to keep in mind, though, that digital circuits all work in binary. Hex is simply used as a convenience for the humans involved. You should memorize the four-bit binary pattern for each hexadecimal digit. Only then will you realize the usefulness of this tool in digital systems.

EXAMPLE 2-4

Convert decimal 378 to a 16-bit binary number by first converting to hexadecimal.

Solution

$$\frac{378}{16} = 23 + \text{remainder of } 10_{10} = A_{16}$$
$$\frac{23}{16} = 1 + \text{remainder of } 7$$
$$\frac{1}{16} = 0 + \text{remainder of } 1$$

Thus, $378_{10} = 17A_{16}$. This hex value can be converted easily to binary 000101111010. Finally, we can express 378_{10} as a 16-bit number by adding four leading 0s:

$$378_{10} = 0000 \quad 0001 \quad 0111 \quad 1010_2$$

EXAMPLE 2-5

Convert $B2F_{16}$ to decimal.

Solution

$$\begin{aligned} B2F_{16} &= B \times 16^2 + 2 \times 16^1 + F \times 16^0 \\ &= 11 \times 256 + 2 \times 16 + 15 \\ &= 2863_{10} \end{aligned}$$

Summary of Conversions

Right now, your head is probably spinning as you try to keep straight all of these different conversions from one number system to another. You probably realize that many of these conversions can be done *automatically* on your calculator just by pressing a key, but it is important for you to master these conversions so that you understand the process. Besides, what happens if your calculator battery dies at a crucial time and you have no handy replacement? The following summary should help you, but nothing beats practice, practice, practice!

1. When converting from binary [or hex] to decimal, use the method of taking the weighted sum of each digit position or follow the double-dabble procedure.
2. When converting from decimal to binary [or hex], use the method of repeatedly dividing by 2 [or 16] and collecting remainders (Figure 2-1).

3. When converting from binary to hex, group the bits in groups of four, and convert each group into the correct hex digit.
4. When converting from hex to binary, convert each digit into its four-bit equivalent.

OUTCOME ASSESSMENT QUESTIONS

1. Convert $24CE_{16}$ to decimal.
2. Convert 3117_{10} to hex, then from hex to binary.
3. Convert 1001011110110101_2 to hex.
4. Write the next four numbers in this hex counting sequence: E9A, E9B, E9C, E9D, ____, ____, ____, ____.
5. Convert 3527_{16} to binary.
6. What range of decimal values can be represented by a four-digit hex number?

2-4 BCD CODE

OUTCOMES

Upon completion of this section, you will be able to:

- Convert decimal numbers to BCD code.
- Convert BCD code to decimal.
- Cite the pros and cons of using BCD.
- Cite advantages/disadvantages of BCD versus binary in digital systems.

When numbers, letters, or words are represented by a special group of symbols, we say that they are being encoded, and the group of symbols is called a *code.* Probably one of the most familiar codes is the Morse code, where a series of dots and dashes represents letters of the alphabet.

We have seen that any decimal number can be represented by an equivalent binary number. The group of 0s and 1s in the binary number can be thought of as a code representing the decimal number. When a decimal number is represented by its equivalent binary number, we call it **straight binary coding.**

Digital systems all use some form of binary numbers for their internal operation, but the external world is decimal in nature. This means that conversions between the decimal and binary systems are being performed often. We have seen that the conversions between decimal and binary can become long and complicated for large numbers. For this reason, a means of encoding decimal numbers that combines some features of both the decimal and the binary systems is used in certain situations.

Binary-Coded-Decimal Code

If *each* digit of a decimal number is represented by its binary equivalent, the result is a code called **binary-coded decimal** (hereafter abbreviated BCD). Since a decimal digit can be as large as 9, four bits are required to code each digit (the binary code for 9 is 1001).