Lecture – 2

Number Systems, Operations

Lesson Outcomes

After completing this lecture, students will be able to

- *Express the number systems and covert one form to another*
- *Perform the arithmetic operations in various number systems*
- *Express decimal numbers to binary coded decimal (BCD) form*

Key Terms

- **Alphanumeric** Consisting of numerals, letters, and other characters.
- **ASCII** American Standard Code for Information Interchange; the most widely used alphanumeric code.
- **BCD** Binary coded decimal; a digital code in which each of the decimal digits, 0 through 9, is represented by a group of four bits.
- **Byte** A group of eight bits.
- **Cyclic redundancy check (CRC)** A type of error detection code.
- **Floating-point number** A number representation based on scientific notation in which the number consists of an exponent and a mantissa.
- **Hexadecimal** Describes a number system with a base of 16.
- **LSB** Least significant bit; the right-most bit in a binary whole number or code.
- **MSB** Most significant bit; the left-most bit in a binary whole number or code.
- **Octal** Describes a number system with a base of eight.
- **Parity** In relation to binary codes, the condition of evenness or oddness of the number of 1s in a code group.

Decimal numbers

- The decimal number system has a base of 10.
- The value of a digit is determined by its position in the number.

$$10^2\ 10^1\ 10^0.10^{-1}\ 10^{-2}\ 10^{-3}\ \ldots$$

Decimal point

- The value of a decimal number is the sum of the digits after each digit has been multiplied by its weight

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

Binary numbers

TABLE 2–1

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

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

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

Binary point

for example, $2^9 = 512$ and $2^{-7} = 0.0078125$.

Binary numbers – application (counting)

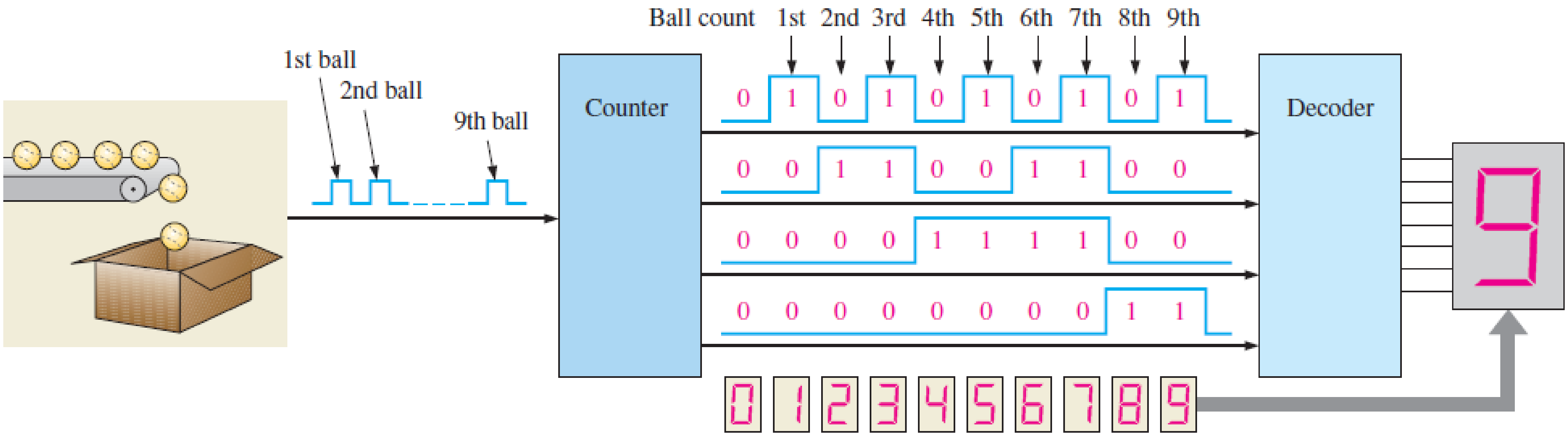


FIGURE 2–1 Illustration of a simple binary counting application.

Binary to decimal number

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.

Weight:	$2^6\ 2^5\ 2^4\ 2^3\ 2^2\ 2^1\ 2^0$
Binary number:	1 1 0 1 1 0 1

$$1101101 = 2^6 + 2^5 + 2^3 + 2^2 + 2^0$$
$$= 64 + 32 + 8 + 4 + 1 = \mathbf{109}$$

Decimal to binary – sum of weights method

EXAMPLE 2–5

Convert the following decimal numbers to binary:

(a) 12 **(b)** 25

(c) 58 **(d)** 82

Solution

(a) $12 = 8 + 4 = 2^3 + 2^2 \longrightarrow$ **1100**

(b) $25 = 16 + 8 + 1 = 2^4 + 2^3 + 2^0 \longrightarrow$ **11001**

(c) $58 = 32 + 16 + 8 + 2 = 2^5 + 2^4 + 2^3 + 2^1 \longrightarrow$ **111010**

(d) $82 = 64 + 16 + 2 = 2^6 + 2^4 + 2^1 \longrightarrow$ **1010010**

Decimal to binary - division by 2 method

Remainder

$$\frac{12}{2} = 6 \qquad 0$$

$$\frac{6}{2} = 3 \qquad 0$$

$$\frac{3}{2} = 1 \qquad 1$$

$$\frac{1}{2} = 0 \qquad 1$$

Stop when the whole-number quotient is 0.

1 1 0 0

MSB LSB

Decimal to binary conversion

EXAMPLE 2–6

Convert the following decimal numbers to binary:

(a) 19 (b) 45

Solution

(a)

	Remainder
$\frac{19}{2} = 9$	1
$\frac{9}{2} = 4$	1
$\frac{4}{2} = 2$	0
$\frac{2}{2} = 1$	0
$\frac{1}{2} = 0$	1

1 0 0 1 1

MSB LSB

(b)

	Remainder
$\frac{45}{2} = 22$	1
$\frac{22}{2} = 11$	0
$\frac{11}{2} = 5$	1
$\frac{5}{2} = 2$	1
$\frac{2}{2} = 1$	0
$\frac{1}{2} = 0$	1

1 0 1 1 0 1

MSB LSB

	Carry
$0.3125 \times 2 = 0.625$	0
$0.625 \times 2 = 1.25$	1
$0.25 \times 2 = 0.50$	0
$0.50 \times 2 = 1.00$	1

MSB LSB

.0 1 0 1

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

Binary operation

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

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

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

Binary Subtraction

The four basic rules for subtracting bits are as follows:

$0 - 0 = 0$	
$1 - 1 = 0$	
$1 - 0 = 1$	
$10 - 1 = 1$	$0 - 1$ with a borrow of 1

(a) $\begin{array}{r} 11 \\ -01 \\ \hline \mathbf{10} \end{array} \qquad \begin{array}{r} 3 \\ -1 \\ \hline 2 \end{array}$ (b) $\begin{array}{r} 11 \\ -10 \\ \hline \mathbf{01} \end{array} \qquad \begin{array}{r} 3 \\ -2 \\ \hline 1 \end{array} \qquad \begin{array}{r} 101 \\ -011 \\ \hline \mathbf{010} \end{array} \qquad \begin{array}{r} 5 \\ -3 \\ \hline 2 \end{array}$

Binary operation

Binary Multiplication

The four basic rules for multiplying bits are as follows:

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

(a)
$$\begin{array}{r} 11 \\ \times\ 11 \\ \hline \text{Partial products}\left\{ \begin{array}{r} 11 \\ +11 \end{array} \right. \\ \hline \mathbf{1001} \end{array} \qquad \begin{array}{r} 3 \\ \times\ 3 \\ \hline 9 \end{array}$$

(b)
$$\begin{array}{r} 111 \\ \times\ 101 \\ \hline \text{Partial products}\left\{ \begin{array}{r} 111 \\ 000 \\ +111 \end{array} \right. \\ \hline \mathbf{100011} \end{array} \qquad \begin{array}{r} 7 \\ \times\ 5 \\ \hline 35 \end{array}$$

Binary Division

Division in binary follows the same procedure as division in decimal,

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

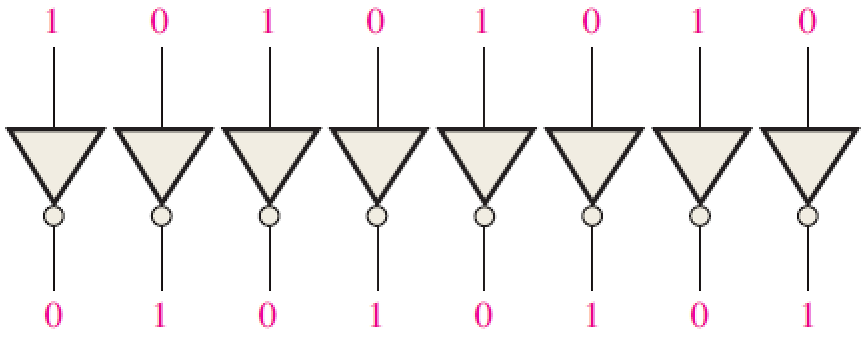
(b)
$$\begin{array}{r} \mathbf{11} \\ 10\overline{)110} \\ \underline{10} \\ 10 \\ \underline{10} \\ 00 \end{array} \qquad \begin{array}{r} 3 \\ 2\overline{)6} \\ \underline{6} \\ 0 \end{array}$$

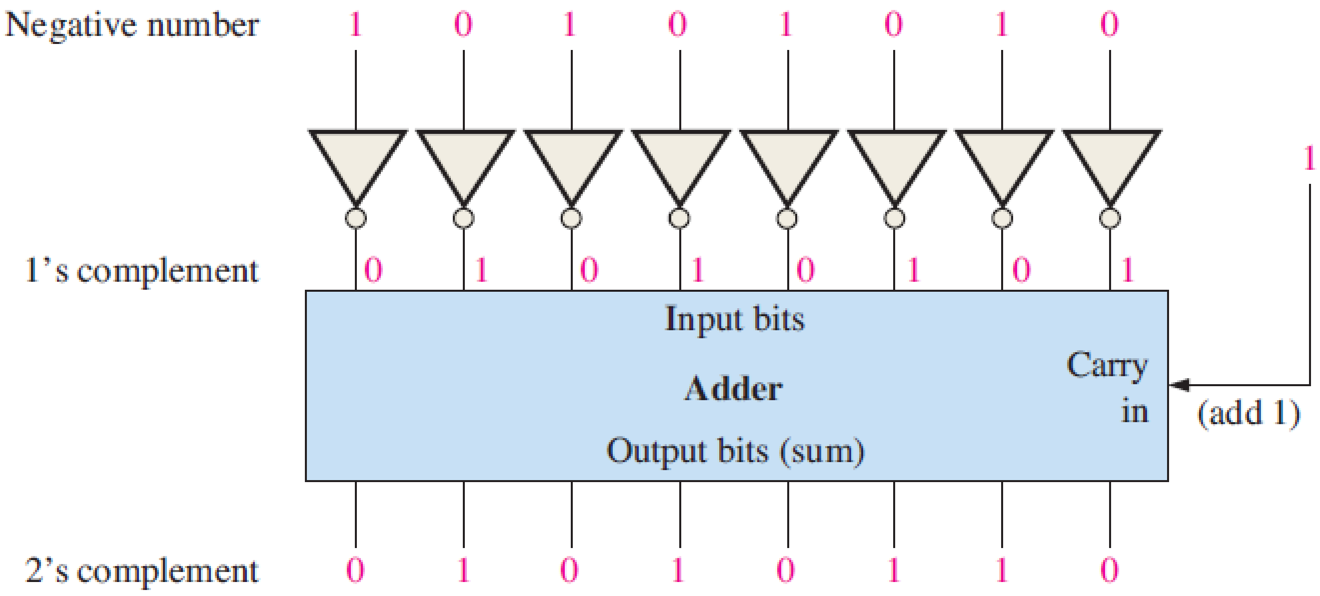
Complements in binary numbers

- The 1's complement and the 2's complement of a binary number are important because they permit the representation of negative numbers.
- The method of 2's complement arithmetic is commonly used in computers to handle negative numbers.

10110010	Binary number
01001101	1's complement
+ 1	Add 1
01001110	2's complement

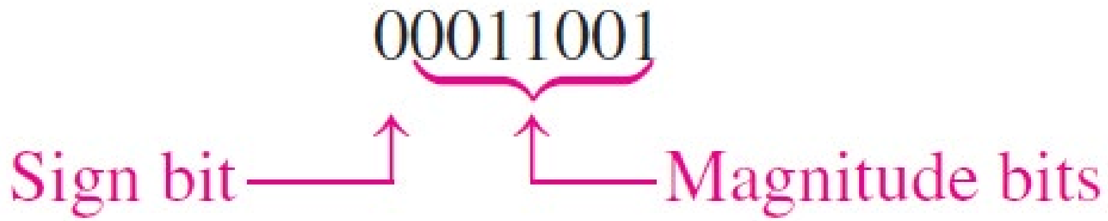
1 0 1 1 0 0 1 0	Binary number
↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓	
0 1 0 0 1 1 0 1	1's complement



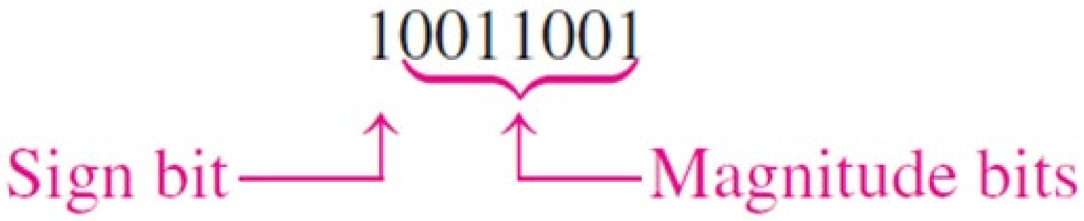


The signed numbers

- **Sign Bit:** The left-most bit in a signed binary number is the **sign bit**, which tells you whether the number is positive or negative.
- A 0 sign bit indicates a positive number, and a 1 sign bit indicates a negative number.
- Using 8-bit signed binary number, the decimal number **+25** is expressed as



- Using 8-bit signed binary number, the decimal number **-25** is expressed as



1’s and 2’s complements in signed numbers

- In the 1’s complement form, a negative number is the 1’s complement of the corresponding positive number.
- In the 2’s complement form, a negative number is the 2’s complement of the corresponding positive number.
- 8-bit number for +39 is: **00100111**
- In the *sign-magnitude form,* -39 is: **10100111**
- In the *1’s complement form,* 239 is: **11011000**
- In the *2’s complement form,* 239 is:

11011000	1’s complement
+ 1	
11011001	2’s complement

Range of signed integer numbers

- The formula for finding the number of different combinations of *n* bits is

$$\boxed{\text{Total combinations} = 2^n}$$

- For 2's complement signed numbers, the range of values for *n*-bit numbers is

$$\boxed{\text{Range} = -(2^{n-1}) \text{ to } +(2^{n-1} - 1)}$$

- For example, with 8 bits the range is: -128 to +127

- With 16 bits, the range is: -32768 to + 32767

Floating-point numbers

- To represent very large **integer** numbers, many bits are required. There is also a problem when numbers with both integer and fractional parts, such as 23.5618, need to be represented. The floating-point number system, based on scientific notation, is capable of representing very large and very small numbers without an increase in the number of bits and also for representing numbers that have both integer and fractional components.

- A **floating-point number** (also known as a *real number*) consists of two parts plus a sign. The **mantissa** is the part of a floating-point number that represents the magnitude of the number and is between 0 and 1. The **exponent** is the part of a floating-point number that represents the number of places that the decimal point (or binary point) is to be moved.

- Let's consider a decimal number which, in integer form, is 241,506,800. The mantissa is .2415068 and the exponent is 9. The floating-point number is written as $\mathbf{0.2415068 \times 10^9}$.

Single-precision floating-point binary numbers

- In the standard format for a single-precision binary number, the sign bit (S) is the left-most bit, the exponent (E) includes the next eight bits, and the mantissa or fractional part (F) includes the remaining 23 bits,

32 bits

S	Exponent (E)	Mantissa (fraction, F)
1 bit	8 bits	23 bits

- The general approach to determining the value of a floating-point number is expressed by

$$\boxed{\text{Number} = (-1)^S(1 + F)(2^{E-127})}$$

Decimal to floating-point binary number

EXAMPLE 2–18

Convert the decimal number 3.248×10^4 to a single-precision floating-point binary number.

Solution

Convert the decimal number to binary.

$$3.248 \times 10^4 = 32480 = 111111011100000_2 = 1.11111011100000 \times 2^{14}$$

The MSB will not occupy a bit position because it is always a 1. Therefore, the mantissa is the fractional 23-bit binary number 11111011100000000000000 and the biased exponent is

$$14 + 127 = 141 = 10001101_2$$

The complete floating-point number is

0	**10001101**	**11111011100000000000000**

Example 2-1

PROBLEM: Determine the binary value of the following floating-point binary number.

1	10010001	10001110001000000000000

PROBLEM:

S	E	F
1	10010001	10001110001000000000000

The sign bit is 1. The biased exponent is 10010001 $=$ 145. Applying the formula, we get

$$\begin{aligned}\text{Number} &= (-1)^1\,(1.10001110001)(2^{145-127})\\ &= (-1)(1.10001110001)(2^{18}) = -1100011100010000000\end{aligned}$$

Arithmetic operations in signed numbers

EXAMPLE 2–20

Perform each of the following subtractions of the signed numbers:

(a) 00001000 − 00000011 **(b)** 00001100 − 11110111

(c) 11100111 − 00010011 **(d)** 10001000 − 11100010

Solution

Like in other examples, the equivalent decimal subtractions are given for reference.

(a) In this case, $8 - 3 = 8 + (-3) = 5$.

	00001000	Minuend (+8)
	+ 11111101	2's complement of subtrahend (−3)
Discard carry →	**1 00000101**	Difference (+5)

Arithmetic operations in signed numbers

(b) In this case, $12 - (-9) = 12 + 9 = 21$.

00001100	Minuend (+12)
+ 00001001	2's complement of subtrahend (+9)
00010101	Difference (+21)

(c) In this case, $-25 - (+19) = -25 + (-19) = -44$.

	11100111	Minuend (−25)
	+ 11101101	2's complement of subtrahend (−19)
Discard carry	**1 11010100**	Difference (−44)

(d) In this case, $-120 - (-30) = -120 + 30 = -90$.

10001000	Minuend (−120)
+ 00011110	2's complement of subtrahend (+30)
10100110	Difference (−90)

Binary multiplication

The basic steps in the partial products method of binary multiplication are as follows:

- **Step 1:** Determine if the signs of the multiplicand and multiplier are the same or different. This determines what the sign of the product will be.
- **Step 2:** Change any negative number to true (uncomplemented) form. Because most computers store negative numbers in 2's complement, a 2's complement operation is required to get the negative number into true form.
- **Step 3:** Starting with the least significant multiplier bit, generate the partial products.
  - ✓ When the multiplier bit is 1, the partial product is the same as the multiplicand.
  - ✓ When the multiplier bit is 0, the partial product is zero. Shift each successive partial product one bit to the left.
- **Step 4:** Add each successive partial product to the sum of the previous partial products to get the final product.
- **Step 5:** If the sign bit that was determined in step 1 is negative, take the 2's complement of the product. If positive, leave the product in true form. Attach the sign bit to the product.

Example 2-2

PROBLEM: Multiply the signed binary numbers: 01010011 (multiplicand) and 11000101 (multiplier).

SOLUTION

Step 1: The sign bit of the multiplicand is 0 and the sign bit of the multiplier is 1. The sign bit of the product will be 1 (negative).

Step 2: Take the 2's complement of the multiplier to put it in true form. 11000101→ 00111011

Step 3 and 4: The multiplication proceeds as follows. Notice that only the magnitude bits are used in these steps.

Step 5: Since the sign of the product is a 1 as determined in step 1, take the 2's complement of the product.

1001100100001 → 0110011011111

Attach the sign bit →

1 0110011011111

1010011	Multiplicand
× 0111011	Multiplier
1010011	1st partial product
+ 1010011	2nd partial product
11111001	Sum of 1st and 2nd
+ 0000000	3rd partial product
011111001	Sum
+ 1010011	4th partial product
1110010001	Sum
+ 1010011	5th partial product
100011000001	Sum
+ 1010011	6th partial product
1001100100001	Sum
+ 0000000	7th partial product
1001100100001	Final product

Hexadecimal numbers

- The **hexadecimal** number system has a base of sixteen; that is, it is composed of 16 numeric and alphabetic characters.
- Most digital systems process binary data in groups that are multiples of four bits, making the hexadecimal number very convenient because each hexadecimal digit represents a 4-bit binary number (as listed in Table 2–3).
- Counting hexadecimal numbers: ….., E, F, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 1A, 1B, 1C, 1D, 1E, 1F, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 2A, 2B, 2C, 2D, 2E, 2F, 30, 31, …..

TABLE 2–3

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

EXAMPLE 2–24

Convert the following binary numbers to hexadecimal:

(a) 1100101001010111 **(b)** 111111000101101001

Solution

(a) $\underbrace{1100}_{\displaystyle C}\underbrace{1010}_{\displaystyle A}\underbrace{0101}_{\displaystyle 5}\underbrace{0111}_{\displaystyle 7} = \mathbf{CA57_{16}}$

(b) $\underbrace{0011}_{\displaystyle 3}\underbrace{1111}_{\displaystyle F}\underbrace{0001}_{\displaystyle 1}\underbrace{0110}_{\displaystyle 6}\underbrace{1001}_{\displaystyle 9} = \mathbf{3F169_{16}}$

Two zeros have been added in part (b) to complete a 4-bit group at the left.

EXAMPLE 2–25

Determine the binary numbers for the following hexadecimal numbers:

(a) $10A4_{16}$ **(b)** $CF8E_{16}$ **(c)** 9742_{16}

Solution

(a) $\overset{1}{1}\overbrace{0000}^{0}\overbrace{1010}^{A}\overbrace{0100}^{4}$

(b) $\overbrace{1100}^{C}\overbrace{1111}^{F}\overbrace{1000}^{8}\overbrace{1110}^{E}$

(c) $\overbrace{1001}^{9}\overbrace{0111}^{7}\overbrace{0100}^{4}\overbrace{0010}^{2}$

In part (a), the MSB is understood to have three zeros preceding it, thus forming a 4-bit group.

Hexadecimal to decimal conversion

EXAMPLE 2–26

Convert the following hexadecimal numbers to decimal:

(a) $1C_{16}$ **(b)** $A85_{16}$

Solution

Remember, convert the hexadecimal number to binary first, then to decimal.

(a)

$$\overbrace{0001}^{1}\overbrace{1100}^{C} = 2^4 + 2^3 + 2^2 = 16 + 8 + 4 = \mathbf{28}_{10}$$

(b)

$$\overbrace{1010}^{A}\overbrace{1000}^{8}\overbrace{0101}^{5} = 2^{11} + 2^9 + 2^7 + 2^2 + 2^0 = 2048 + 512 + 128 + 4 + 1 = \mathbf{2693}_{10}$$

Hexadecimal to decimal conversion (weighted method)

EXAMPLE 2–27

Convert the following hexadecimal numbers to decimal:

(a) $E5_{16}$ **(b)** $B2F8_{16}$

Solution

Recall from Table 2–3 that letters A through F represent decimal numbers 10 through 15, respectively.

(a) $E5_{16} = (E \times 16) + (5 \times 1) = (14 \times 16) + (5 \times 1) = 224 + 5 = \mathbf{229_{10}}$

(b)
$$\begin{aligned} B2F8_{16} &= (B \times 4096) + (2 \times 256) + (F \times 16) + (8 \times 1) \\ &= (11 \times 4096) + (2 \times 256) + (15 \times 16) + (8 \times 1) \\ &= 45{,}056 + 512 + 240 + 8 = \mathbf{45{,}816_{10}} \end{aligned}$$

Decimal to hexadecimal

EXAMPLE 2–28

Convert the decimal number 650 to hexadecimal by repeated division by 16.

Solution

Hexadecimal remainder

$$\frac{650}{16} = 40.625 \rightarrow 0.625 \times 16 = 10 = \text{A}$$

$$\frac{40}{16} = 2.5 \rightarrow 0.5 \times 16 = 8 = 8$$

$$\frac{2}{16} = 0.125 \rightarrow 0.125 \times 16 = 2 = 2$$

Stop when whole number quotient is zero.

2 8 A Hexadecimal number

MSD LSD

Hexadecimal addition

$$\begin{array}{r} 58_{16} \\ +\ 22_{16} \\ \hline \mathbf{7A_{16}} \end{array}$$

right column: $8_{16} + 2_{16} = 8_{10} + 2_{10} = 10_{10} = A_{16}$

left column: $5_{16} + 2_{16} = 5_{10} + 2_{10} = 7_{10} = 7_{16}$

$$\begin{array}{r} 2B_{16} \\ +\ 84_{16} \\ \hline \mathbf{AF_{16}} \end{array}$$

right column: $B_{16} + 4_{16} = 11_{10} + 4_{10} = 15_{10} = F_{16}$

left column: $2_{16} + 8_{16} = 2_{10} + 8_{10} = 10_{10} = A_{16}$

$$\begin{array}{r} DF_{16} \\ +\ AC_{16} \\ \hline \mathbf{18B_{16}} \end{array}$$

right column: $F_{16} + C_{16} = 15_{10} + 12_{10} = 27_{10}$

$27_{10} - 16_{10} = 11_{10} = B_{16}$ with a 1 carry

left column: $D_{16} + A_{16} + 1_{16} = 13_{10} + 10_{10} + 1_{10} = 24_{10}$

$24_{10} - 16_{10} = 8_{10} = 8_{16}$ with a 1 carry

Octal number

The **octal** number system is composed of eight digits, which are

$$0, 1, 2, 3, 4, 5, 6, 7$$

To count above 7, begin another column and start over:

$$10, 11, 12, 13, 14, 15, 16, 17, 20, 21, \ldots$$

Octal to decimal number

$$\begin{array}{rl} \text{Weight:} & 8^3\ 8^2\ 8^1\ 8^0 \\ \text{Octal number:} & 2\ \ 3\ \ 7\ \ 4 \end{array}$$

$$\begin{aligned} 2374_8 &= (2 \times 8^3) + (3 \times 8^2) + (7 \times 8^1) + (4 \times 8^0) \\ &= (2 \times 512) + (3 \times 64) + (7 \times 8) + (4 \times 1) \\ &= 1024 + 192 + 56 + 4 = 1276_{10} \end{aligned}$$

Decimal to octal conversion

Remainder

$$\frac{359}{8} = 44.875 \longrightarrow 0.875 \times 8 = 7$$

$$\frac{44}{8} = 5.5 \longrightarrow 0.5 \times 8 = 4$$

$$\frac{5}{8} = 0.625 \longrightarrow 0.625 \times 8 = 5$$

Stop when whole number quotient is zero.

5 4 7 Octal number

MSD LSD

Octal to binary conversion

TABLE 2–4

Octal/binary conversion.

Octal Digit	0	1	2	3	4	5	6	7
Binary	000	001	010	011	100	101	110	111

EXAMPLE 2–31

Convert each of the following octal numbers to binary:

(a) 13_8 (b) 25_8 (c) 140_8 (d) 7526_8

Solution

(a) $\underbrace{1}_{001}\ \underbrace{3}_{011} \rightarrow 001011$

(b) $\underbrace{2}_{010}\ \underbrace{5}_{101} \rightarrow 010101$

(c) $\underbrace{1}_{001}\ \underbrace{4}_{100}\ \underbrace{0}_{000} \rightarrow 001100000$

(d) $\underbrace{7}_{111}\ \underbrace{5}_{101}\ \underbrace{2}_{010}\ \underbrace{6}_{110} \rightarrow 111101010110$

Binary to octal conversion

EXAMPLE 2–32

Convert each of the following binary numbers to octal:

(a) 110101 **(b)** 101111001 **(c)** 100110011010 **(d)** 11010000100

Solution

(a) $\underbrace{110}_{6}\underbrace{101}_{5}$ $= \mathbf{65}_8$

(b) $\underbrace{101}_{5}\underbrace{111}_{7}\underbrace{001}_{1}$ $= \mathbf{571}_8$

(c) $\underbrace{100}_{4}\underbrace{110}_{6}\underbrace{011}_{3}\underbrace{010}_{2}$ $= \mathbf{4632}_8$

(d) $\underbrace{011}_{3}\underbrace{010}_{2}\underbrace{000}_{0}\underbrace{100}_{4}$ $= \mathbf{3204}_8$

Binary coded decimal (BDC)

- **The 8421 BCD Code:** The 8421 code is a type of **BCD** (binary coded decimal) code. Binary coded decimal means that each decimal digit, 0 through 9, is represented by a binary code of four bits. The designation 8421 indicates the binary weights of the four bits (2^3, 2^2, 2^1, 2^0).

- **Invalid Codes:** You should realize that, with four bits, sixteen numbers (0000 through 1111) can be represented but that, in the 8421 code, only ten of these are used. The six code combinations that are not used—1010, 1011, 1100, 1101, 1110, and 1111—are invalid in the 8421 BCD code.

- **Applications:** Digital clocks, digital thermometers, digital meters, and other devices with seven-segment displays typically use BCD code to simplify the displaying of decimal numbers.

TABLE 2–5

Decimal/BCD conversion.

Decimal Digit	0	1	2	3	4	5	6	7	8	9
BCD	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001

Decimal to BCD and vice versa

EXAMPLE 2–33

Convert each of the following decimal numbers to BCD:

(a) 35 (b) 98 (c) 170 (d) 2469

Solution

(a) $\overbrace{0011}^{3}\overbrace{0101}^{5}$

(b) $\overbrace{1001}^{9}\overbrace{1000}^{8}$

(c) $\overbrace{0001}^{1}\overbrace{0111}^{7}\overbrace{0000}^{0}$

(d) $\overbrace{0010}^{2}\overbrace{0100}^{4}\overbrace{0110}^{6}\overbrace{1001}^{9}$

EXAMPLE 2–34

Convert each of the following BCD codes to decimal:

(a) 10000110 (b) 001101010001 (c) 1001010001110000

Solution

(a) $\underbrace{1000}_{8}\underbrace{0110}_{6}$

(b) $\underbrace{0011}_{3}\underbrace{0101}_{5}\underbrace{0001}_{1}$

(c) $\underbrace{1001}_{9}\underbrace{0100}_{4}\underbrace{0111}_{7}\underbrace{0000}_{0}$

BCD addition

EXAMPLE 2–36

Add the following BCD numbers:

(a) 1001 + 0100 (b) 1001 + 1001

(c) 10000110 + 00010011 (d) 010001010000 + 010000010111

(a)

	1001		9
	+ 0100		+4
	1101	Invalid BCD number (>9)	13
	+ 0110	Add 6	
0001	**0011**	Valid BCD number	
1	3		

(b)

	1001		9
	+ 1001		+ 9
1	0010	Invalid because of carry	18
	+ 0110	Add 6	
0001	**1000**	Valid BCD number	
1	8		

(c)

1000	0110	86
+ 0001	0011	+ 13
1001	**1001**	**99**

(d)

0100	0101	0000	450
+ 0100	0001	0111	+ 417
1000	**0110**	**0111**	**867**

References

1. ***Digital Fundamentals*** by Thomas Floyd, Pearson International Edition, 11th Edition, Chapter 2, Page 65-109.

Next class

Logic Gates