ALU

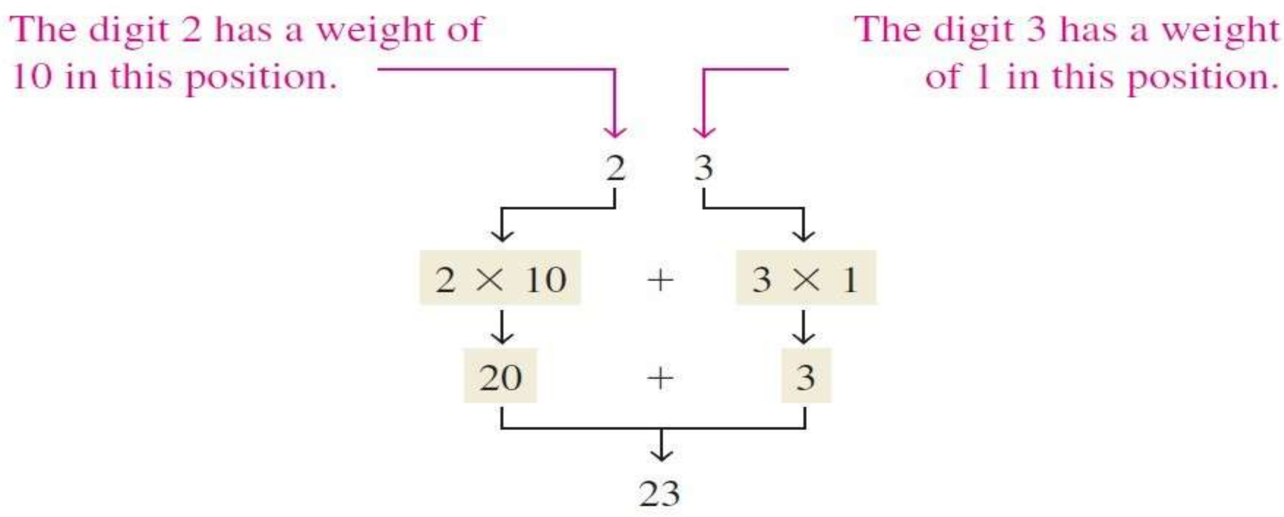
LECTURE 4: NUMBER SYSTEM

INTRODUCTION

- The binary number system and digital codes are fundamental to computers and to digital electronics in general.
- In this Lecture, the binary number system and its relationship to other number systems such as decimal, hexadecimal, and octal are presented.
- Arithmetic operations with binary numbers are covered to provide a basis for understanding how computers and many other types of digital systems work

DECIMAL SYSTEM

- In the **decimal number system each of the ten digits, 0 through 9, represents a certain** 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 in the next slide.



- 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

  $10^0 = 1$. $\ldots 10^5\ 10^4\ 10^3\ 10^2\ 10^1\ 10^0$
- For fractional numbers, the weights are negative powers of ten that decrease from left to right beginning with 10^{-1}.

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

Examples

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.

$$47 = (4 * 10^1) + (7 * 10^0)$$

$$= (4 * 10) + (7 * 1) = \mathbf{40} + \mathbf{7}$$

Related Problem*

- Determine the value of each digit in **939**.

Examples cont'd

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}.

$$568.23 = (5*10^2) + (6*10^1) + (8*10^0) + (2*10^{-1}) + (3*10^{-2})$$

$$= (5*100) + (6*10) + (8*1) + (2*0.1) + (3*0.01)$$

$$= \mathbf{500} + \mathbf{60} + \mathbf{8} + \mathbf{0.2} + \mathbf{0.03}$$

Related Problem*

- Determine the value of each digit in 67.924.

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.

BINARY NUMBERS

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

BINARY NUMBERS

- As you **MIGHT** have seen in the Table, four bits are required to count from zero to 15. In general, with *n bits you can count up to a number equal to* $2^n - 1$.
- 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.*
- $2^6 - 1 = 64 - 1 = 63$

BINARY NUMBER APPLICATION

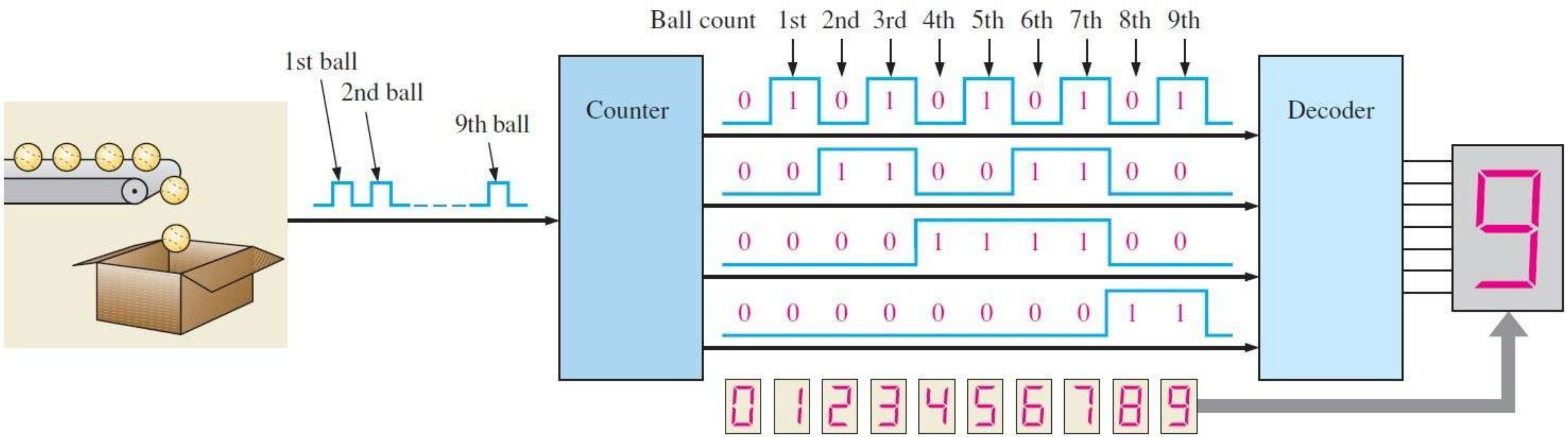


FIGURE 2–1 Illustration of a simple binary counting application.

- 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.

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.

- The weight structure of a binary number is

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

where *n is the number of bits from the binary point.*

Binary-to-Decimal Conversion

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 0.5	1/4 0.25	1/8 0.125	1/16 0.625	1/32 0.03125	1/64 0.015625

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:

1. 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 = **109**

Related Problem *

1. Convert the binary number 10010001 to decimal.
2. Convert the binary number 10.111 to decimal.

Decimal-to-Binary Conversion

- 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.
- The decimal number 9, for example, can be expressed as the sum of binary weights as follows:
- $9 = 8 + 1$ or $9 = 2^3 + 2^0$
- 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.
- 2^3 2^2 2^1 2^0
- 1 0 0 1 Binary number for decimal 9

Decimal-to-Binary Conversion

1. 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**

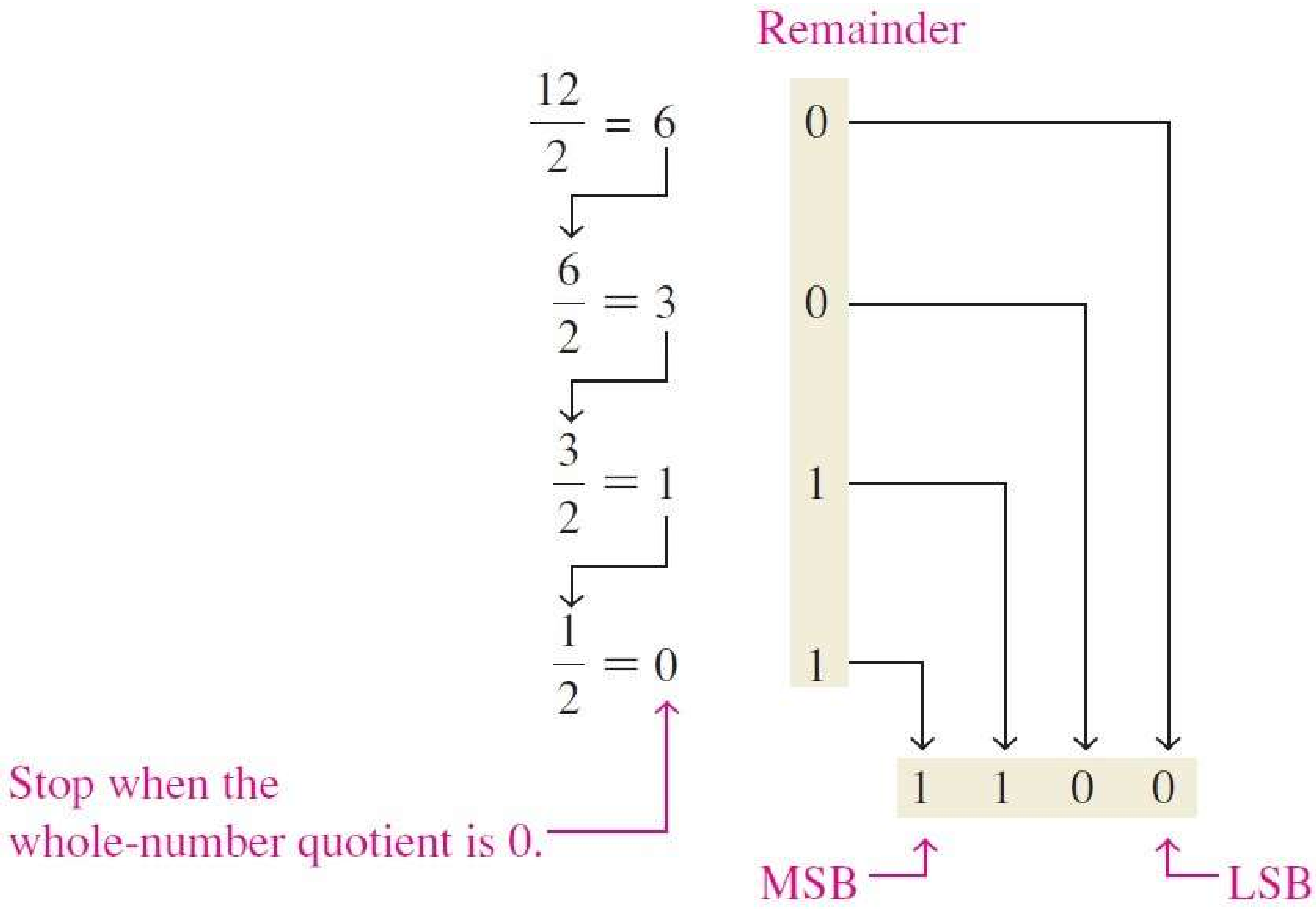
(c) $58 = 32 + 16 + 8 + 2 = 2^5 + 2^4 + 2^3 + 2^1$ **111010**

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

Related Problem*

Convert the decimal number 125 to binary.

Decimal-to-Binary Conversion



Example:

Convert the following decimal numbers to binary:

(a) 19 [10011] **(b) 45 [101101]**

Decimal Fractions-to-Binary Conversion

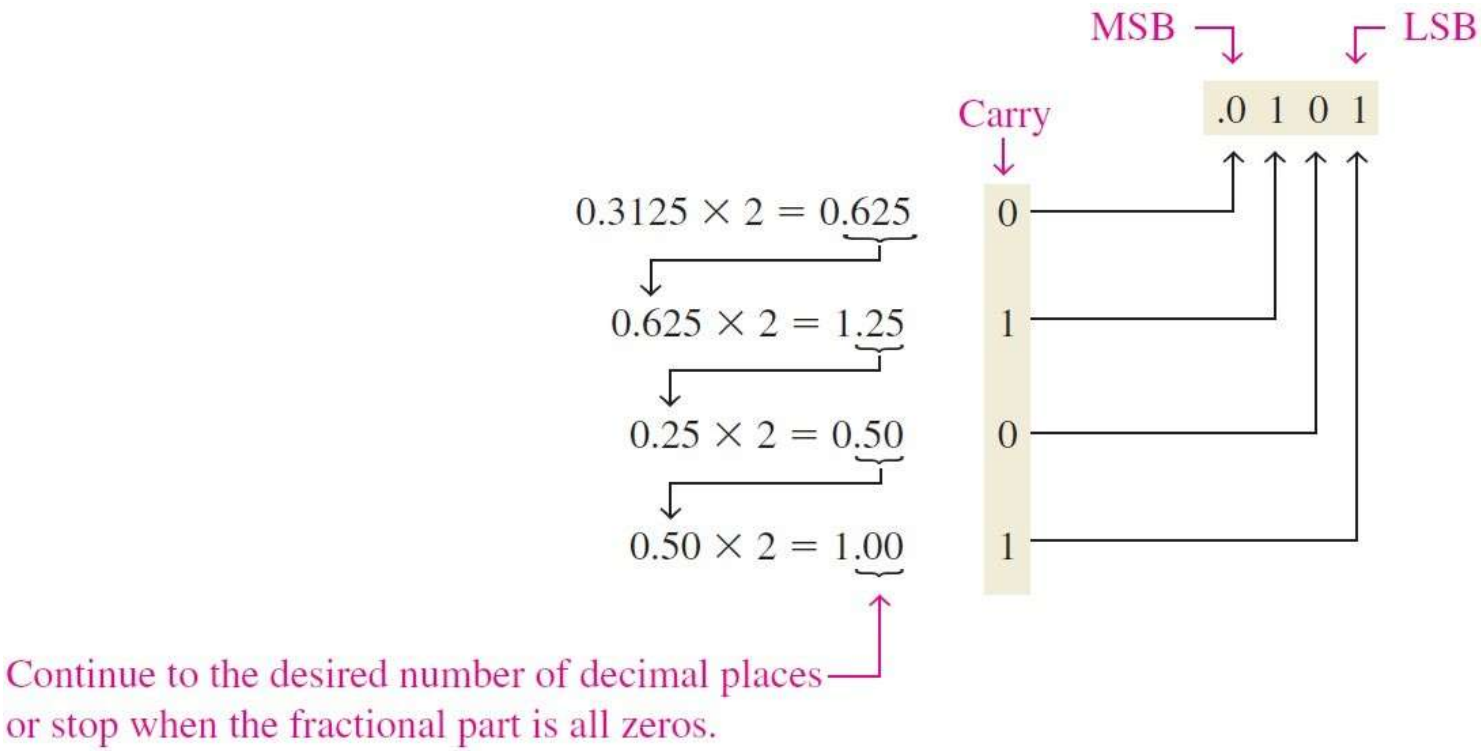
The sum-of-weights method can be applied to fractional decimal numbers, as shown in the following example:

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

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

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.



Arithmetic Operations of Binary Numbers

ADDITION

$0 + 0 = 0$	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

DIVISION

Division in binary follows the same procedure as division in decimal

SUBTRACTION

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

MULTIPLICATION

$0 \times 0 = 0$

$0 \times 1 = 0$

$1 \times 0 = 0$

$1 \times 1 = 1$

EXAMPLES

1. Add the following binary numbers:

(a) 11 + 11 [110] **(b) 100 + 10 [110]**

(c) 111 + 11 [1010] **(d) 110 + 100 [1010]**

2. Subtract 011 from 101. **[010]**

3. Perform the following binary multiplications:

(a) 11 * 11 [1001] **(b) 101 * 111 [100011]**

4. Perform the following binary divisions:

(a) 110/11 [10] **(b) 110 /10 [11]**

Complements of 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.
- The 1’s **complement of a binary number is found by changing all 1s to 0s and all 0s to 1s**
- The 2’s complement of a binary number is found by adding 1 to the LSB of the 1’s complement.
- 2’s complement = (1’s complement) + 1

Complements of Binary Numbers

1. Find the 2's complement of 10110010. **[01001110]**
2. Find the 1's complement of 10110010. **[01001101]**

- An alternative method of finding the 2's complement of a binary number is as follows:

1. Start at the right with the LSB and write the bits as they are up to and including the first 1.

2. Take the 1's complements of the remaining bits.

Eg. 10111000 ---- 01001000

Signed Numbers

- Digital systems, such as the computer, must be able to handle both positive and negative numbers.
- A signed binary number consists of both sign and magnitude information.
- The sign indicates whether a number is positive or negative, and the magnitude is the value of the number.
- There are three forms in which signed integer (whole) numbers can be represented in binary: sign-magnitude, 1’s complement, and 2’s complement.

Signed Numbers and 1’s Complement

- 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.**

00011001 ----- +25

10011001 ----- -25

- Positive numbers in 1’s complement form are represented the same way as the positive sign-magnitude numbers. Negative numbers, however, are the 1’s complements of the corresponding positive numbers. (2’s complement follows its own rules as well)

00011001 ---- +25

11100110 ---- -25

Signed Numbers and 1's Complement

1. Determine the decimal value of this signed binary number expressed in sign-magnitude: 10010101.

[-21]

2. Determine the decimal values of the signed binary numbers expressed in 1's complement:

(a) 00010111 [23] **(b) 11101000 [-23]**

Decimal values of negative numbers are determined by assigning a negative value to the weight of the sign bit, summing all the weights where there are 1s, and adding 1 to the result.

Signed Numbers and 1's Complement

1. Determine the decimal values of the signed binary numbers expressed in 2's complement:

(a) 01010110 [86] **(b) 10101010 [-86]**

** The weight of the sign bit in a negative number is given a negative value.**

MODULO 2 OPERATION

- Modulo-2 addition (or subtraction) is the same as binary addition with the carries discarded, as shown in the table below.

Modulo-2 operation.

Input Bits	Output Bit
0 0	0
0 1	1
1 0	1
1 1	0

- **Truth tables are widely used to** describe the operation of logic circuits, as you will learn in the next lecture.
- With two bits, there is a total of four possible combinations, as shown in the table.
- This particular table describes the modulo-2 operation also known as *exclusive-OR and can be implemented with a logic gate*

- Another convention is called *BCD (binary coded decimal"). In this case each decimal* digit is separately converted to binary. Therefore, since 7 = 0111_2 and 9 = 1001_2, then 79 = 01111001 (BCD).

- It is very often quite useful to represent blocks of 4 bits by a single digit. Thus in base 16 there is a convention for using one digit for the numbers 0,1,2,. . .,*15 which is called hexadecimal. It follows decimal for 0-9, then uses letters A-F.*
- How do you count in hexadecimal once you get to F? Simply start over with another column and continue as follows:
- …, 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
- With two hexadecimal digits, you can count up to FF_{16}, which is decimal 255. To count beyond this, three hexadecimal digits are needed. For instance, 100_{16} is decimal 256, 101_{16} is decimal 257, and so forth. The maximum 3-digit hexadecimal number is FFF_{16}, or decimal 4095. The maximum 4-digit hexadecimal number is $FFFF_{16}$, which is decimal 65,535.

Binary to Hexadecimal

- Converting a binary numb ecimal is a straightforward procedure ak the binary number into 4-bit groups, he right-most bit and replace each 4-bit the equivalent hexadecimal symbol.

Decimal	Binary	Hex
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:

1. Convert the following bin to hexadecimal:

(a) 1100101001010111 (b) 1 01101001

Solution

(a) 1100101001010111 (b) 0011111100010110101001

$CA57_{16}$ **$3F169_{16}$**

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

Hexadecimal to Binary

- To convert from a hexadecimal number to a binary number, reverse the process and replace each hexadecimal symbol with the appropriate four bits.

EXAMPLE:

1. Determine the binary numbers for the following hexadecimal numbers:

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

Solution

(a)1000010100100 **(b)1100111110001110**
(c)1001011101000010

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

ASSIGNMENT

1. Convert the following hexadecimal numbers to decimal:

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