Digital Design

CS302

Ref: Digital Design. M. Morris Mano , and Michael D. Ciletti. Pearson, FIFTH EDITION, 2013

Digital Systems and Binary Numbers

CHAPTER 1

1.1 DIGITAL SYSTEMS

- Digital systems are used in communication, traffic control, medical treatment, weather monitoring, …
- Discrete information: any set that is restricted to a finite number of elements.
- Digital systems is a treatment of discrete elements of information.
- Discrete elements of information are represented in a digital system by physical quantities called signals.
- The variables in the analog computer are represented by continuous signals usually electric voltages that vary with time.

1.1 DIGITAL SYSTEMS

- The signals in electronic digital systems use just two discrete values (said to be *binary*).
- A binary digit, called a bit, has two values: 0 and 1.
- Discrete elements of information are represented with groups of bits called *binary codes*.
- **Example**

The decimal digits 0 through 9 are represented in a digital system with a code of four bits (e.g., the number 7 is represented by 0111).

1.2 BINARY NUMBERS

The decimal number 392 is a shorthand notation for what should be written as

$$3 \times 10^2 + 9 \times 10^1 + 2 \times 10^0$$

The decimal number system is said to be of *base* (*radix*) 10.

The *binary system* : the coefficients of the binary number system have only two values: 0 and 1 and is of *radix 2*

1.2 BINARY NUMBERS

Each coefficient is multiplied by a power of the radix 2

The number 1010.11 is 10.75:

$(1010.\ 11)_2$

$=1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 0 \times 2^0 + 1 \times 2^{-1} + 1 \times 2^{-2}$

$= 8 + 2 + 0.5 + 0.25 = (10.\ 75)_{10}$

1.2 BINARY NUMBERS

- A major trend in digital design methodology is the use of a HDL to describe and simulate the functionality of a digital circuit.
- An HDL (hardware description language) resembles a programming language and is suitable for describing digital circuits in textual form.
- It is used to design a digital system to verify its operation before hardware is built.

1.2 BINARY NUMBERS

In general, a number expressed in a base-r system has coefficients multiplied by powers of r:

$$a_n \cdot r^n + a_{n-1} \cdot r^{n-1} + \cdots + a_2 \cdot r^2 + a_1 \cdot r + a_0 + a_{-1} \cdot r^{-1} + a_{-2} \cdot r^{-2} + \cdots + a_{-m} \cdot r^{-m}$$

The coefficients *aj* range in value from 0 to *r - 1.*

It is customary to borrow the needed *r digits for the coefficients from the decimal* system when the base of the number is less than 10. The letters of the alphabet are used to supplement the 10 decimal digits when the base of the number is greater than 10.

For example, in the *hexadecimal (base-16) number system, the first 10 digits are borrowed*

1.3 NUMBER-BASE CONVERSIONS

	Integer Quotient		Remainder	Coefficient
$41/2 =$	20	+	$\frac{1}{2}$	$a_0 = 1$
$20/2 =$	10	+	0	$a_1 = 0$
$10/2 =$	5	+	0	$a_2 = 0$
$5/2 =$	2	+	$\frac{1}{2}$	$a_3 = 1$
$2/2 =$	1	+	0	$a_4 = 0$
$1/2 =$	0	+	$\frac{1}{2}$	$a_5 = 1$

Therefore, the answer is $(41)_{10} = (a_5a_4a_3a_2a_1a_0)_2 = (101001)_2$.

1.3 NUMBER-BASE CONVERSIONS

Integer	Remainder	
41		
20	1	
10	0	
5	0	
2	1	
1	0	
0	1	101001 = answer

1.3 NUMBER-BASE CONVERSIONS

Table 1.2
Numbers with Different Bases

Decimal (base 10)	Binary (base 2)	Octal (base 8)	Hexadecimal (base 16)
00	0000	00	0
01	0001	01	1
02	0010	02	2
03	0011	03	3
04	0100	04	4
05	0101	05	5
06	0110	06	6
07	0111	07	7
08	1000	10	8
09	1001	11	9
10	1010	12	A
11	1011	13	B
12	1100	14	C
13	1101	15	D
14	1110	16	E
15	1111	17	F

1.9 BINARY LOGIC

- Binary logic deals with variables that take on two discrete values and with operations that assume logical meaning.
- The two values the variables assume may be called by different names (true and false, yes and no, etc.), but for our purpose, it is convenient to think in terms of bits and assign the values 1 and 0.
- The binary logic introduced in this section is equivalent to an algebra called Boolean algebra.
- There are three basic logical operations: AND, OR, and NOT.

1.9 BINARY LOGIC

Table 1.8
Truth Tables of Logical Operations

AND				OR				NOT	
x	y	$x \cdot y$		x	y	$x + y$		x	x'
0	0	0		0	0	0		0	1
0	1	0		0	1	1		1	0
1	0	0		1	0	1			
1	1	1		1	1	1			

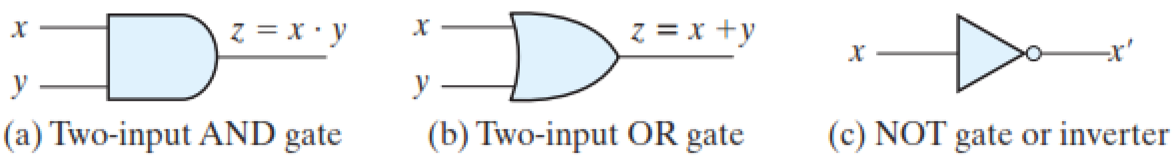

FIGURE 1.4
Symbols for digital logic circuits