Number Systems

The electrical circuits on an IC have one of two states, off or on. Therefore, the *binary number system* (base 2), which uses only two digits (0 and 1), was adopted for use in computers. To represent numbers and letters, a code was developed with eight binary digits grouped together to represent a single number or letter. Each 0 or 1 in the binary code is called a *bit* (BInary digiT) and an 8-bit unit is called a *byte*.

Our most familiar number system is the decimal, or *base 10*, system. It uses ten digits: 0 through 9. Each place represents a power of ten, with the first place to the left of the decimal point representing 10^0, the next place representing 10^1, the next 10^2, and so on (remember that any number raised to the zero power is 1). In the decimal number 485, the 4 represents 4×10^2, the 8 represents 8×10^1, and the 5 represents 5×10^0. The number 485 represents the sum 4×100 + 8×10 + 5×1 (400 + 80 + 5):

Decimal	Base 10 Equivalent
485	$4\times10^2 + 8\times10^1 + 5\times10^0 = 400 + 80 + 5$

The binary, or *base 2*, system works identically except that each place represents a power of two instead of a power of ten. For example, the binary number 101 represents the sum $1\times2^2 + 0\times2^1 + 1\times2^0$ or 5 in base ten. Some decimal numbers and their binary equivalents are:

Decimal	Binary	Base 2 Equivalent		
0	0	$= 0\times2^1 + 0\times2^0$	= 0×2 + 0×1	= 0 + 0
1	1	$= 0\times2^1 + 1\times2^0$	= 0×2 + 1×1	= 0 + 1
2	10	$= 1\times2^1 + 0\times2^0$	= 1×2 + 0×1	= 2 + 0
3	11	$= 1\times2^1 + 1\times2^0$	= 1×2 + 1×1	= 2 + 1
4	100	$= 1\times2^2 + 0\times2^1 + 0\times2^0$	= 1×4 + 0×2 + 0×1	= 4 + 0 + 0

The hexadecimal system is used to represent groups of four binary digits. The *hexadecimal*, or *base 16*, system is based on 16 digits: 0 through 9, and the letters A through F representing 10 through 15 respectively. Each place represents a power of sixteen. For example, the hexadecimal number 1F represents the sum $1\times16^1 + 15\times16^0$. Some decimal numbers and their hexadecimal equivalents are:

Decimal	Binary	Hexadecimal	Base 16 Equivalent		
0	0000 0000	0	$= 0\times16^0$	= 0×1	= 0
10	0000 1010	A	$= 10\times16^0$	= 10×1	= 10
25	0001 1001	19	$= 1\times16^1 + 9\times16^0$	= 1×16 + 9×1	= 16 + 9
30	0001 1110	1E	$= 1\times16^1 + 14\times16^0$	= 1×16 + 14×1	= 16 + 14

For clarity, a non-base 10 number should have the base subscripted after the number. For example, to show the difference between 100 in base 10 and 100 in base 2 (which represents 4), the base 2 number should be written as 100_2.

Every letter of an alphabet (Latin, Japanese, Cherokee, and so on) and symbols of every culture (=, @, ½, and so on) have been given a representation in a digital code called Unicode. *Unicode* uses a set of sixteen 1s and 0s to form a 16-bit binary code for each symbol. For example, the uppercase letter V is Unicode 00000000 01010110, which can be thought of as the base 10 number 86 (86_{10}). Lowercase v has a separate code of 00000000 01110110, or 11810.

ASCII and EBCDIC

ASCII (American Standard Code for Information Interchange) and EBCDIC (Extended Binary Coded Decimal Interchange Code) are two other digital coding schemes. Unlike Unicode, ASCII and EBCDIC are not large enough to support Asian and other languages that use a different alphabet.