NUMBER SYSTEM

Number systems are the technique to represent numbers in the computer system architecture, every value that you are saving or getting into/from computer memory has a defined number system.

Computer architecture supports following number systems.

→ Binary number system
→ Octal number system
→ Decimal number system
→ Hexadecimal (hex) number system

BINARY NUMBER SYSTEM

A Binary number system has only two digits that are 0 and 1. Every number (value) represents with 0 and 1 in this number system. The base of binary number system is 2, because it has only two digits.

OCTAL NUMBER SYSTEM

Octal number system has only eight (8) digits from 0 to 7. Every number (value) represents with 0,1,2,3,4,5,6 and 7 in this number system. The base of octal number system is 8, because it has only 8 digits.

DECIMAL NUMBER SYSTEM

Decimal number system has only ten (10) digits from 0 to 9. Every number (value) represents with 0,1,2,3,4,5,6, 7,8 and 9 in this number system. The base of decimal number system is 10, because it has only 10 digits.

HEXADECIMAL NUMBER SYSTEM

A Hexadecimal number system has sixteen (16) alphanumeric values from 0 to 9 and A to F. Every number (value) represents with 0,1,2,3,4,5,6, 7,8,9,A,B,C,D,E and F in this number system. The base of hexadecimal number system is 16, because it has 16 alphanumeric values. Here A is 10, B is 11, C is 12, D is 14, E is 15 and F is 16.

Number system	Base(Radix)	Used digits	Example
Binary	2	0,1	(11110000)2
Octal	8	0,1,2,3,4,5,6,7	(360)8
Decimal	10	0,1,2,3,4,5,6,7,8,9	(240)10
Hexadecimal	16	0,1,2,3,4,5,6,7,8,9, A,B,C,D,E,F	(F0)16

Decimal and Binary Numbers

When we write decimal (base 10) numbers, we use a positional notation system. Each digit is multiplied by an appropriate power of 10 depending on its position in the number:

For example:

843 = 8 x 102 + 4 x 101 + 3 x 100

= 8 x 100 + 4 x 10 + 3 x 1

= 800 + 40 + 3

For whole numbers, the rightmost digit position is the one’s position (100 = 1). The numeral in that position indicates how many ones are present in the number. The next position to the left is ten’s, then hundred’s, thousand’s, and so on. Each digit position has a weight that is ten times the weight of the position to its right.

In the decimal number system, there are ten possible values that can appear in each digit position, and so there are ten numerals required to represent the quantity in each digit position. The decimal numerals are the familiar zero through nine (0, 1, 2, 3, 4, 5, 6, 7, 8, 9).

In a positional notation system, the number base is called the radix. Thus, the base ten system that we normally use has a radix of 10. The term radix and base can be used interchangeably.

When writing numbers in a radix other than ten, or where the radix isn't clear from the context, it is customary to specify the radix using a subscript. Thus, in a case where the radix isn't understood, decimal numbers would be written like this:
12710   1110   567310

The binary number system is also a positional notation numbering system, but in this case, the base is not ten, but is instead two. Each digit position in a binary number represents a power of two. So, when we write a binary number, each binary digit is multiplied by an appropriate power of 2 based on the position in the number:
For example:
101101 = 1 x 25 + 0 x 24 + 1 x 23 + 1 x 22 + 0 x 21 + 1 x 20
= 1 x 32 + 0 x 16 + 1 x 8 + 1 x 4 + 0 x 2 + 1 x 1
= 32 + 8 + 4 + 1

convert 110112 to decimal
1 1 0 1 1
\ \ \ _________1 x 20 = 1
\ \ __________1 x 21 = 2
\ _____________1 x 23 = 8
______________1 x 24 = 16
___
27

Converting a decimal to binary
The method for converting a decimal number to binary is one that can be used to convert from decimal to any number base. It involves using successive division by the radix until the dividend reaches 0. At each division, the remainder provides a digit of the converted number, starting with the least significant digit
An example of the process: convert 3710 to binary
37 / 2 = 18 remainder 1 (least significant digit)
18 / 2 = 9 remainder 0
9 / 2 = 4 remainder 1
4 / 2 = 2 remainder 0
2 / 2 = 1 remainder 0
1 / 2 = 0 remainder 1 (most significant digit)
The resulting binary number is: 100101

Another example: convert 9310 to binary
93 / 2 = 46 remainder 1 (least significant digit)
46 / 2 = 23 remainder 0
23 / 2 = 11 remainder 1
11 / 2 = 5 remainder 1
5 / 2 = 2 remainder 1
2 / 2 = 1 remainder 0
1 / 2 = 0 remainder 1 (most significant digit)
The resulting binary number is: 1011101

Hexadecimal Digits

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
10 .... A 1010
11 .... B 1011
12 .... C 1100
13 .... D 1101
14 .... E 1110
15 ....F 1111

Powers of 2:

2^0 1
2^1 2
2^2 4
2^3 8
2^4 16
2^5 32
2^6 64
2^7 128
2^8 256
2^9 512
2^10 1024
2^11 2048
2^12 4096
2^13 8192
2^14 16384
2^15 32768
2^16 65536

Convert the binary number 10110101 to a hexadecimal number

Divide into groups for 4 digits 1011 0101
Convert each group to hex digit B 5
B516

Convert the binary number 0110101110001100 to hexadecimal

Divide into groups of 4 digits 0110 1011 1000 1100
Convert each group to hex digit 6 B 8 C
6B8C16