Number Base Conversions

By Donald W. Larson
dwlarson@mac.com
http://www.timeoutofmind.com

Numbering systems have been arbitrarily created by mathematicians to represent the counting of objects in an organized manner. In this paper the discussion will revolve around the Decimal, Binary, and Hexadecimal numbering systems.

Decimal or base ten (10) is the numbering system that we are most familiar with. Whenever you are counting and reach nine (9), we simply use the zero (0) as a placeholder and move one position to the left to continue as in:

1
2
3
4
5
6
7
8
9
10 (notice the 'zero' used as a "one's" unit's place holder and a 'one' has been put in the ten's position by moving the 'one' to the left. This now is interpreted as one ten's unit plus zero one's units.)

11
... (continued)
99 (interpreted as 'nine' ten's plus 'nine' one's)
100 (interpreted as 'one' hundred's plus 'zero' ten's plus 'zero' one's)

Notice as we to count upwards and reach ninety-nine (99), we put a zero in the unit's and ten's position and move one place to the left and enter a one (1) in the hundred's position, creating the number one hundred (100). It should be noted that each unit position is a power of ten. This is the method of using base ten or decimal counting.

Other numbering systems follow the same basic rule; that is when you exceed the maximum number within the unit position of the base, place a zero (0) in that location, and move to the left one position to continue.

Binary or base two (2) is used to represent objects as either being 'true' (usually represented as a 1) to a particular condition or 'false' (usually represented as a 0) to a particular condition. Computer designers use Binary as a mathematical way to represent whether an electronic circuit has power on it or not. The important point concerning Binary is that there are only two numerals that make up this system, a 'zero' and a 'one'. We still count by the same rule mentioned above by shifting to the left as needed each time a one is incremented and placing a zero in the previous position. Only now we are counting by powers of two. Therefore to represent a decimal digit using Binary math, we need more positions for the Binary digits. An example is shown below:

0001 (a '1' in the first position [(2^0) * 1 = 1])

0010 (a ‘0’ in the first position indicating that no one’s are present; and a ‘1’ in the second position indicating that the number has been incremented to ’2’, resulting in a move to the left and adding a ’0’ as a place holder on the right-most positions {[(2^0) * 0 =0] + [(2^1) * 1 = 2]} totaling 2)

0011 (the right-most position has been incremented therefore we add 1 plus 2 which equals 3 {[(2^0) * 1 = 1] + [(2^1) * 1 = 2]} totaling 3)

0100 (the right-most position has been incremented again causing the second position to be incremented also, so a third position has been created to hold the value. This number represents {[(2^0) * 0 = 0] + [(2^1) * 0 = 0] + [(2^2) * 1 = 4]} totaling 4)

Hexadecimal or base sixteen (16) commonly called Hex for short, is used as shorthand to represent Binary digits remembering that now each number position is a power of 16. The normal convention is to use a ‘$’ proceeding Hex digits to avoid confusion with decimal digits. At this time, review the equivalency table illustrated below before proceeding.

Decimal (base 10)	Binary (base 2)	Hexadecimal (base 16)
0	0000 0000	$00
1	0000 0001	$01
2	0000 0010	$02
3	0000 0011	$03
4	0000 0100	$04
5	0000 0101	$05
6	0000 0110	$06
7	0000 0111	$07
8	0000 1000	$08
9	0000 1001	$09
10	0000 1010	$0A
11	0000 1011	$0B
12	0000 1100	$0C
13	0000 1101	$0D
14	0000 1110	$0E
15	0000 1111	$0F
16	0001 0000	$10
...		
100	0110 0100	$64
...		
255	1111 1111	$FF
256	0001 0000 0000	$100

Notice that the Hex digit $0F uses only 2 characters of space (not counting the ‘$’) to represent the Binary number ‘0000 1111’ which takes up 8 characters. Also notice that $64 represents 100 decimal (6 times 16 then adding 4 to equal 100). And finally, recognizing that $100 using only three characters is representing 12 Binary characters. As the Binary numbers continue to increase, the Hex numbers are really beginning to aid in the comprehension of the Binary values. You should be beginning to understand the reason why Hex is used by programmers instead of Binary when dealing with computers.

In closing, whether or not you become proficient in using the Binary or Hex numbering systems in addition to our common decimal system is important only if you intend to study computer hardware or software in a deeper manner. Our point here is that the mathematics involved is not as mysterious as one would imagine at first glance. On the other hand, the casual user of the computer can accomplish a great many tasks without ever knowing anything about Binary or Hex. The fact that you have had this limited exposure to the underlying information could provide you with a certain advantage when you're involved in a technical conversation concerning computers.