Base Conversion

Our regular counting system is the decimal (base 10) system. This is because we use **10** distinct digits, zero through nine. In general, the numerical value of a number is what you were taught in elementary school. For example,

$$2713 = 2 \times 10^3 + 7 \times 10^2 + 1 \times 10^1 + 3 \times 10^0$$

Each digit's value is determined by which place it's in. Each place is a perfect power of the base, with the least significant at the end, counting up by one as you go through the number from right to left.

Although this seems to be the only possible number system, it turns out that the number of digits used is arbitrary. We could have just as easily chose to use **5** digits $(0 - 4)$, in which case the value of a number would be as follows:

$$314_5 = 3 \times 5^2 + 1 \times 5^1 + 4 \times 5^0 = 84_{10}.$$

Thus, this is how we convert from a different base to base 10. In general, we can write our conversion as follows:

$$d_{n-1}d_{n-2}\ldots d_2d_1d_0 \text{ (in base b)} = d_{n-1} \times b^{n-1} + d_{n-2} \times b^{n-2} + \ldots + d_2 \times b^2 + d_1 \times b + d_0$$

(Note, b raised to the 1 and 0 powers were simplified above.)

Let's look at a couple quick examples:

$$781_9 = 7 \times 9^2 + 8 \times 9^1 + 1 \times 9^0 = 640_{10}$$

$$1110101_2 = 1 \times 2^6 + 1 \times 2^5 + 1 \times 2^4 + 0 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 = 117_{10}$$

(Note: Base 2 is so common, it has a name: binary.)

Hexadecimal and conversion from Hexadecimal to Binary

The most common base with more than 10 digits is base 16, or hexadecimal. The sixteen digits used in this base are 0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F, with A = 10, B = 11, C = 12, D = 13, E = 14 and F = 15.

Although everything internally in a computer is stored in base 2 (binary), often times when we view the contents of memory or are assigning values (such as RGB values for colors), we actually view numbers in hexadecimal. This is one reason why it's important to be familiar with and be able to convert to and from hexadecimal to other bases. In particular, the easiest conversions are to and from other bases that are also perfect powers to 2.

Note that converting from base 16 to base 10 is done exactly as shown on the previous page. Consider the following example:

$A3D_{16} = A \times 16^2 + 3 \times 16^1 + D = 10 \times 16^2 + 3 \times 16 + 13 = 2621_{10}$.

Since 16 is a perfect power of two, converting to base 2 is relatively easy, because each hexadecimal digit is perfectly represented by 4 binary digits (since $16 = 2^4$.) Here's a chart with the conversions between each hexadecimal digit and base two:

Hex	0	1	2	3	4	5	6	7
Bin	0000	0001	0010	0011	0100	0101	0110	0111
Hex	8	9	A	B	C	D	E	F
Bin	1000	1001	1010	1011	1100	1101	1110	1111

Thus, $A3D_{16} = 1010\ 0011\ 1101_2$.

Converting from Decimal to Another Base

Now, let's consider the problem of being given a number in decimal and having to convert it to another base, say binary. Let's use the example of converting 117. We know that

$$117_{10} = d_6 x2^6 + d_5 x2^5 + d_4 x2^4 + d_3 x2^3 + d_2 x2^2 + d_1 x2^1 + d_0 x2^0$$

where each digit is either a 0 or a 1. (Technically we don't yet know how many digits the final answer will be in base two.)

But, with very little calculation, here is what we CAN determine:

All of the terms on the right-hand side are divisible by 2 except for the last. Notice that 117%2 = 1. Now, consider calculating the remainder when you divide the right-hand side by 2. All of the terms except for the last leave a remainder of 0 since 2 divides evenly into these terms. Thus, we know that $d_0 = 117\%2$.

Now, consider dividing the right-hand side by 2, using int. division:

$$d_6 x2^5 + d_5 x2^4 + d_4 x2^3 + d_3 x2^2 + d_2 x2^1 + d_1 x2^0$$

This number must equal 117/2 = 58. From this point on, we can repeat the process to determine d_1, which is 58%2 = 0, then d_2, etc. Here is the whole example worked out:

117 % 2 = 1 (d_0)	117/2 = 58	Now, read the answer from the
58 % 2 = 0 (d_1)	58/2 = 29	bottom up: 1110101.
29 % 2 = 1 (d_2)	29/2 = 14	
14 % 2 = 0 (d_3)	14/2 = 7	
7 % 2 = 1 (d_4)	7/2 = 3	
3 % 2 = 1 (d_5)	3/2 = 1	
1 % 2 = 1 (d_6)	1/2 = 0 (we can stop)	

Although the example above was worked out for base 2, none of the logic used was specific to base 2. The same logic would apply any other base. When you mod by the base on the right-hand side, all terms drop out except for the last digit. Thus, repeated mod and dividing operations by the value of the base will extract each digit in the converted number, in reverse order.

Let's look at a couple more examples:

$381_{10} = $ _____ $_{16}$

$381 \ \% \ 16 = 13 \ (D)$ $381/16 = 23$
$23 \ \ \% \ 16 = \ \ 7$ $23/16 \ \ = 1$
$1 \ \ \ \ \% \ 16 \ = \ 1$ $1/16 \ \ \ \ = 0$, so $381_{10} = 17D_{16}$.

$175_{10} = $ _____ $_{3}$

$175 \ \% \ 3 = 1$ $175/3 = 58$
$58 \ \ \% \ 3 = 1$ $58/3 \ \ = 19$
$19 \ \ \% \ 3 = 1$ $19/3 \ \ = 6$
$6 \ \ \ \ \% \ 3 = 0$ $6/3 \ \ \ = 2$
$2 \ \ \ \ \% \ 3 = 2$ $2/3 \ \ \ \ = 0$, so $175_{10} = 20111_{3}$.

Converting from any base (B_1) to any other base (B_2) where neither base is base 10

To convert from any base (B_1) to any other base (B_2, except for base 10), use a two step process:

1) Convert from base B_1 to base 10.
2) Convert from base 10 to base B_2.

This should be fairly clear and straight-forward since it's literally putting together the two algorithms shown in this lecture.

If you are converting between two bases that are perfect powers of 2, the following procedure works more quickly:

1) Convert from base B_1 to base 2.
2) Convert from base 2 to base B_2.

Consider the following example:

$A3D_{16} = $ _____ $_8$

We have already done step 1 previously, and found that

$A3D_{16} = 1010\ 0011\ 1101_2$.

Now, we must convert the right-hand side to base 8. Remember that $8 = 2^3$, so that three binary digits perfectly represent one octal (term for base 8) digit. Group the binary digits in sets of three, from right to left, and then convert each set of three binary digits to its octal equivalent.

$101\quad 000\quad 111\quad 101_2 = 5075_8$

Why the base conversion between bases of powers of 2 works

This example, drawn out should clarify why the base conversion works:

$$A3D_{16} = 10 \times 16^2 + 3 \times 16^1 + 13 \times 16^0$$
$$= (1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 0 \times 2^0) \times 16^2 +$$
$$(0 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 1 \times 2^0) \times 16^1 +$$
$$(1 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0)$$

$$= (1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 0 \times 2^0) \times 2^8 +$$
$$(0 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 1 \times 2^0) \times 2^4 +$$
$$(1 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0) \text{, by rewriting 16 as a power of 2.}$$

$$= 1 \times 2^{11} + 0 \times 2^{10} + 1 \times 2^9 + 0 \times 2^8 +$$
$$0 \times 2^7 + 0 \times 2^6 + 1 \times 2^5 + 1 \times 2^4 +$$
$$1 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0$$
$$= 101000111101_2.$$

Note that this only works when one base is a perfect power of the second base.

The reverse transformation, grouping three binary digits at a time and replacing them with one octal digit is very similar to the one above. (The same issue is true for this transformation; it only works when the second base is a perfect power of the first.)