1. Number Systems

All popular number systems use an Arabic-style representation, i.e. you write a numeric value as a string of digits, for example $N_3 N_2 N_1 N_0$, which is really shorthand for the actual value being represented. The actual value is a polynomial in the number system's Base, where the coefficients are the shorthand digits. For example, if B is the base of a number system and a value in that system is represented as $N_3 N_2 N_1 N_0$, then:

$\mathbf{N_3 N_2 N_1 N_0}$ is actually shorthand for the polynomial $\mathbf{N_3}*B^3 + \mathbf{N_2}*B^2 + \mathbf{N_1}*B^1 + \mathbf{N_0}*B^0$.

So goes the general theory. Here are specific examples in the common bases we use:

Base 2:	**1010 0110**	$= \mathbf{1}*2^7 + \mathbf{0}*2^6 + \mathbf{1}*2^5 + \mathbf{0}*2^4 + \mathbf{0}*2^3 + \mathbf{1}*2^2 + \mathbf{1}*2^1 + \mathbf{0}*2^0$
Base 10:	**106**	$= \mathbf{1}*10^2 + \mathbf{0}*10^1 + \mathbf{6}*10^0$
Base 16:	**A60**	$= \mathbf{A}*16^2 + \mathbf{6}*16^1 + \mathbf{0}*16^0$

To differentiate base 16 numbers from base 10 numbers and from numbers in other number systems, it is customary to prefix base 16 numbers with 0x, for example, A60 is written **0xA60**.

1.1 Number System Base

The Base of a number system may be any integer greater than unity (1), i.e., any number from the list 2, 3, 4, 5, etc. To each unique Base, there corresponds a unique number system. For example, Base 2 corresponds to the binary number system; Base 3 corresponds to the ternary number system, Base 4 to the quaternary number system, Base 8 to the octal number system, Base 10 to the decimal number system, Base 16 to the hexadecimal number system, Base 60 to the sexagesimal (time and angular measure) number system, etc.

An important identity used by all number systems can be stated as follows:

Any number system base B, raised to the zeroth power, i.e. $\mathbf{B^0}$, equals unity (1).

1.2 Number System Digits

Digits are the unique symbols of the number system. The number of digits in a number system equals the BASE of the number system. Every number system has a value for zero and unity. For example:

Base 2 digits:	**0 1**	(two digits)
Base 10 digits:	**0 1 2 3 4 5 6 7 8 9**	(ten digits)
Base 16 digits:	**0 1 2 3 4 5 6 7 8 9 A B C D E F**	(sixteen digits)

1.3 Conversion between Number System Digits

Before we consider how to convert a number from one system to another, we need to know how to convert the digits of the number systems. The following table shows how to do this (memory is required here). Note: the basic digits of each number system are shown in **boldface**.

Base Two (binary digits)	Base Ten (decimal digits)	Base Sixteen (hexadecimal digits)
0	**0**	**0**
1	**1**	**1**

Base Two (binary digits)	Base Ten (decimal digits)	Base Sixteen (hexadecimal digits)
10	**2**	**2**
11	**3**	**3**
110	**4**	**4**
101	**5**	**5**
110	**6**	**6**
111	**7**	**7**
1000	**8**	**8**
1001	**9**	**9**
1010	10	**A**
1011	11	**B**
1100	12	**C**
1101	13	**D**
1110	14	**D**
1111	15	**F**

(The usual abbreviations are "hex" for hexadecimal, and "bit" for binary digit.)

1.4 Conversion between Number Systems

There is one general method, called "quotient and remainder division", which can be used to convert Arabic-style numbers in any number system. Though you can always use this conversion method, there are special optimized methods available for certain conversions. These methods are easier and faster than the division method. Below, we show the easiest methods of number system conversion for the three most common systems in use.

Base 2 → Base 16:

Method: transcription.

1) Starting from right side of binary number, make small hash marks every four bits.
2) Using above table, transcribe from binary to hexadecimal.

Example: **1101110001101010111100010101001110**

1) 11,0111,0001,1010,1011,1100,0101,0100,1110
2) 3 7 1 A B C 5 4 E

Result: **0x371ABC54E**

Base 16 → Base 2:

Method: transcription.

1) Starting from either end of a hexadecimal number, simply transcribe each hex digit to its equivalent four bits using above table.

Example: **0xC0DE426**

2) C 0 D E 4 2 6

Result: **1100000011011110010000100110**

Base 16 → Base 10:

Method: polynomial expansion.

3) Convert the shorthand hex number to its polynomial expansion as shown under **"1. Number Systems"** above.
4) Replace the hex digits A through F by their decimal equivalents.
5) Carry out the multiplication and addition to arrive at the decimal result.

Example: **0xEA51E**

1) $E*16^4 + A*16^3 + 5*16^2 + 1*16^1 + E*16^0$
2) $(14)*16^4 + (10)*16^3 + 5*16^2 + 1*16^1 + (14)*16^0$
3) 14*65,536 + 10*4,096 + 5*256 + 1*16 + 14*1 = 959,774

Result: **959,774**

Base 10 → Base 16

Method: division.

1) Divide 16 (in general, the base you're converting TO) into the decimal number (in general then number you're converting from). You will get a quotient (Q) and a remainder (R).
2) Convert remainders 10 through 15 to hex digits, if necessary. The remainder (even if it's a zero) becomes the next digit on the LEFT of the resulting hex number. In other words, you're building the hex number one digit at a time from right to left.
3) If Q is greater than 15 (0xF), take it as the new decimal number and go back to the first step. Otherwise, you're done—just place the quotient to the left of the hex result.

Example: **4125**

1) 4125 16: Q=257, R=13 (0xD). "D" becomes the rightmost digit of the result. Since, Q=257 is greater than 15, go back to first step.
2) 257 16: Q=16, R=1. "1" becomes the next digit (on the left) of the result, which is now 0x1D. Since Q=16 is greater than 15, go back to first step.
3) 16 16: Q=1, R=0. "0" becomes the next digit (on the left) of the result, which is now 0x01D. Since Q=1 is not greater than 15, put it on the left of the result, and you are finished.

Result: **0x101D**

Base 2 → Base 10:

Method 1: polynomial expansion.

1) Convert the shorthand binary number to its polynomial expansion as shown under **"1. Number Systems"** above.
2) Carry out the multiplication and addition to arrive at the decimal result.

This is too tedious. Though simple, there are too many operations.

Method 2: combination of above (preferred).

1) Base 2 → Base 16
2) Base 16 → Base 10

Base 10 → Base 2:

Method: combination of above (preferred to minimize number of operations).

1) Base 10 → Base 16
2) Base 16 → Base 2

2. A Convenient Relationship between Base 2 and Base 10 Numbers

It is a happy coincidence of arithmetic that $\mathbf{2^{3.333...}}$ is approximately equal to **10** (actually, it is almost 10.08). As a result, we can write:

$$2^{10} = 2^{3.333... * 3} = (2^{3.333...})^3 \sim 10^3$$

From this result, we can build the following very helpful table of approximations, which you can use to perform calculations quickly in your head:

Power of 2	Power of 10	Common Name and Abbreviation
2^0	10^0	Unity
2^{10}	10^3	Kilo K (as in KB or kilobytes)
2^{20}	10^6	Mega M (as in MB or megabytes)
2^{30}	10^9	Giga G (as in GB or gigabytes)
2^{40}	10^{12}	Tera T (as in TB or terabytes)
2^{50}	10^{15}	Peta P (as in PB or petabytes)
2^{60}	10^{18}	Exa E (as in EB or exabytes)

When you add exponents, you multiply numbers, and vice-versa. For example, suppose someone asks you to prove that the practical limit to the size of a HDD using standard partition tables is two terabytes. You would reply that there are 32 bits in standard partition table entries for the relative (beginning) sector number of the partition, and each sector has 512 bytes. Since 512 = 2^9, the maximum size of a standard HDD is currently limited to $2^{32} * 2^9 = 2^{41} = 2^1 * 2^{40}$ = 2 TB.