## Hexadecimal

Internally, most modern computer memories are organized as a collection of 8 bit bytes grouped into words of 4 or 8 bytes each. However, when printing the contents of memory, representing bytes and words as binary yields inconveniently long sequences of 1s and 0s. To avoid this we use hexadecimal, that is, base 16 numbers.

To represent a byte in hexadecimal, we first note that a byte consists of two 4 bit fields and that in 4 bits we can count from 0 to 15 (16 values). A single digit hexadecimal number also ranges from 0 to 15. Thus, we can represent the contents of any byte with exactly 2 hexadecimal digits, one digit for the first 4 bits and a second digit for the second 4 bits. Since 16 is itself a power of two, conversion between a digit in base 16 and the corresponding value in base 2 and back is a trivial substitution as can be seen in Table Table.

In order to use base 16 we need 16 symbols to represent the digits of the numbering system (the equivalent of 010 through 1510). For the first 10 digits, we use the same symbols as the decimal numbering system (0 through 9). For digits above nine, representing the values 1010 through 1510, we use the upper (or lower) case letters *A, B, C, D, E,* and *F* (see Table Table).

When we have a binary number longer than 4 bits, we group the bits, usually from right to left, into groups of 4 bits each and then substitute. Thus, if you have a binary number such as:

0000 0001 1010 1011 0010 0011 0111 1000

Using the values from Table Table, it can be represented in hexadecimal as:

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0000 | 0001 | 1010 | 1011 | 0010 | 0011 | 0111 | 1000 | binary |
| 0 | 1 | A | B | 2 | 3 | 7 | 8 | hex |

Conversion from a hexadecimal number (also referred to as *hex*) to its equivalent decimal notation, however, requires more work because there is a fundamental change in base. When you want the decimal equivalent of a hex number, you need to sum, in decimal, the contributions of each hex digit. To do this, a table such as that in Table Table is useful. It gives, for each hex digit position (up to an eight digit hex number), the contribution of that digit expressed in decimal.

Thus, if you have the hexadecimal number such as: 01 AB 23 78, it can be converted to decimal by adding the positional decimal equivalent contributions of each of the hex digits as shown in Table Table.

Conversion from decimal to hex can be done in several ways. The easiest, if you have Table Table, is to successively find the largest decimal value in the table and subtract it from the decimal number until the remainder is zero. Each corresponding hex digit from each step is retained as part of the final answer. If you skip a hex position in the process, insert a zero in the skipped position. For example, if you have the decimal number: 123456789, this can be rendered in hex as shown in Table Table.

|  |  |
| --- | --- |
| Hex | Decimal Equivalent by Digit Position |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 268435456 | 16777216 | 1048576 | 65536 | 4096 | 256 | 16 | 1 |
| 2 | 536870912 | 33554432 | 2097152 | 131072 | 8192 | 512 | 32 | 2 |
| 3 | 805306368 | 50331648 | 3145728 | 196608 | 12288 | 768 | 48 | 3 |
| 4 | 1073741824 | 67108864 | 4194304 | 262144 | 16384 | 1024 | 64 | 4 |
| 5 | 1342177280 | 83886080 | 5242880 | 327680 | 20480 | 1280 | 80 | 5 |
| 6 | 1610612736 | 100663296 | 6291456 | 393216 | 24576 | 1536 | 96 | 6 |
| 7 | 1879048192 | 117440512 | 7340032 | 458752 | 28672 | 1792 | 112 | 7 |
| 8 | 2147483648 | 134217728 | 8388608 | 524288 | 32768 | 2048 | 128 | 8 |
| 9 | 2415919104 | 150994944 | 9437184 | 589824 | 36864 | 2304 | 144 | 9 |
| A | 2684354560 | 167772160 | 10485760 | 655360 | 40960 | 2560 | 160 | 10 |
| B | 2952790016 | 184549376 | 11534336 | 720896 | 45056 | 2816 | 176 | 11 |
| C | 3221225472 | 201326592 | 12582912 | 786432 | 49152 | 3072 | 192 | 12 |
| D | 3489660928 | 218103808 | 13631488 | 851968 | 53248 | 3328 | 208 | 13 |
| E | 3758096384 | 234881024 | 14680064 | 917504 | 57344 | 3584 | 224 | 14 |
| F | 4026531840 | 251658240 | 15728640 | 983040 | 61440 | 3840 | 240 | 15 |
| Col | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 |
| Table 1 Hex to Decimal Conversion Chart |

Of course, another method to convert between binary, decimal and hexadecimal is to use the pro­gramming mode of a calculator such as that which comes with many Linux distributions.

|  |  |  |  |
| --- | --- | --- | --- |
| from table col 8 corresponding to | 0 | add |  0 (0) |
| from table col 7 corresponding to | 1 | add | 16777216 (1) |
| from table col 6 corresponding to | A | add |  10485760 (A) |
| from table col 5 corresponding to  | B | add |  720896 (B) |
| from table col 4 corresponding to | 2 | add |  8192 (2) |
| from table col 3 corresponding to  | 3 | add |  768 (3) |
| from table col 2 corresponding to | 7 | add |  112 (7) |
| from table col 1 corresponding to | 8 | add |  8 (8) |
|  | sum | 27,992,952 |
| Table 2 Conversion from Hex to Decimal |

The above discussion concerns positive integers only. Negative integers and numbers with fractional parts (floating point numbers) and binary coded decimals (BCD) numbers will be discussed later.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| from 123456789 | subtract | 117440512 | 716 (col 7) | leaving | 6016277 |
| from 6016277 | subtract | 5242880 | 516 (col 6) | leaving | 773397 |
| from 773397 | subtract | 720896 | B16 (col 5) | leaving | 52501 |
| from 52501 | subtract | 49152 | C16 (col 4) | leaving | 3349 |
| from 3349 | subtract | 3328 | D16 (col 3) | leaving | 21 |
| from 21 | subtract | 16 | 116 (col 2) | leaving | 5 |
| from 5 | subtract | 5 | 516 (col 1) | leaving | 0 |
| Result: 07 5B CD 1516 |
| Table 3 Decimal 123456789 to Hex Conversion |

# Data Types

Modern computers use a variety of data types to represent and manipulate information. These include character and bit strings, addresses, signed and unsigned binary integers, signed floating point numbers, binary coded decimals and so forth. For each of these, there are hardware instructions to operate on instances of the data.

In this section, we review the basic data types and how they are represented in the IBM mainframe. Some of these data types are used on all computers, from cell phones to mainframes, while others are peculiar only to larger systems.