Numbering Systems

Decimal

Decimal is a Base 10 numbering system, meaning that you can use any number from 0-9 for each number place (there are up to 10 possible values).

Binary

The numbering system that computers use is Base 2, meaning that there are up to 2 possible values for each number place. Each number value is a bit that is either 0 or 1.

Base 2 = Binary

0 000
1 001
010
011
100
101
110
111

- A three-bit binary number has 8 possible combinations (000. 001, 002, etc)
- A eight-bit binary number has 256 possible combinations, and it is 2 to the eight power

10101100

$2^8 = 256$

Hexadecimal

Base 16 numbering system (16 dif character options)

- 10 - 15 are represented by letters

Base <u>16</u> = Hexadecimal

0	A=10
1	B=11
2	C=12
3	D=13
4	E=14
5	F=15
6	
7	
8	
9	

Conversion

Converting binary to decimal

One way to convert is manually:

BIN		DEC
110	000	= 0
	001	= 1
	010	= 2
	011	= 3
	100	= 4
	101	= 5
	110	= 6
	111	= 7

A better approach would be to create a conversion grid that lists the decimal equivalents for each number place in the binary number. In this case, those numbers would be 128, 64, 32, 16, 8, 4, 2, and 1 for an eight bit binary number. What you do then is put the 1s and 0s in the binary number above the corresponding slots of this conversion grid. If there's a 1 above a given number, you bring its value down; if there's a 0, you don't. Then you add the numbers together and that gives you the decimal equivalent. In this case, the decimal equivalent of the binary number up here is 172.

BIN

10101100 $2^8 = 256$

1	0	1	0	1	1	0	0	
128	64	32	16	8	4	2	1	
128	+	32	+	8	+ 4			= 172

Converting decimal to binary

Let's say the decimal number is 196 and you need to figure out the eight-bit binary equivalent. Using the conversion grid, first you'd ask yourself 'Does 128 fit into 196?' If it does, we put a 1 above the 128. Then you take 196 and subtract the 128 from it, leaving 68. Next ask, 'Does 64 fit into 68?' In this case, it does. So we put a 1 above the 64. Then we subtract 68 from 64, and we're left with 4. Does 32 fit into 4? No. 16? Nope. 8? No. However, 4 matches up. So we put a 1 above that and then the rest are 0s. There's the eight-bit binary number that is the equivalent of the decimal of 196.

DEC

196 =

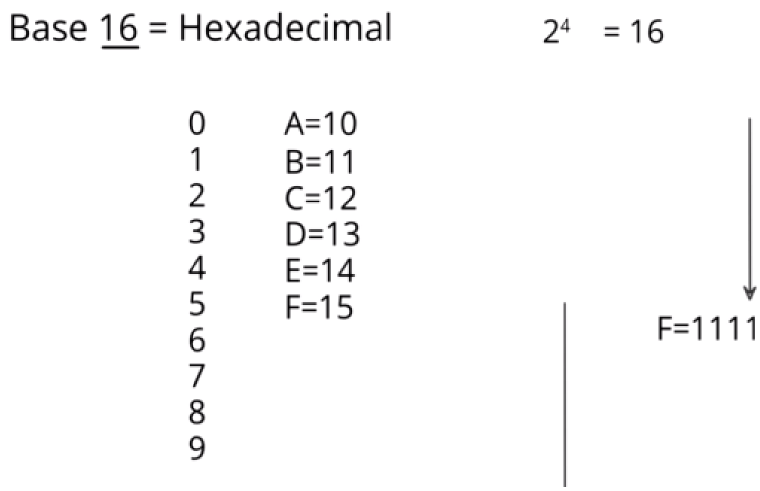
1	1	0	0	0	1	0	0
128	64	32	16	8	4	2	1

$$\begin{array}{r} 196 \\ -128 \\ \hline 68 \\ -64 \\ \hline 4 \end{array}$$

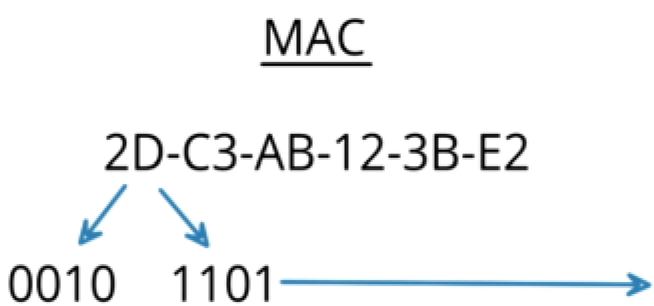
Convert Binary to Hexadecimal

With hexadecimal you're working with Base 16 numbers, so there are 16 total possible combinations for each number place. To represent one hexadecimal number place in binary, you need a four-bit binary number because two to the fourth equals 16.

In this example, you would convert all these hexadecimal characters to four-bit binary numbers, all of the way up to the last possible hexadecimal character, which is F. You would represent it with four 1s. This is probably the fastest way to convert from binary to hexadecimal and vice versa.



For example, let's suppose you need to convert a MAC address on the computer to its binary equivalent. A MAC address is usually a 48-bit address that is usually written out in hexadecimal notation, as shown here. To convert this MAC address to binary, take each character and break it out into its four-bit binary equivalent. In this case, the hexadecimal number two would work out to be 0010 in binary. Hexadecimal D would work out to be 1011, and so on.



Hexadecimal Value	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
Binary Value	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
Decimal Value	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

Hex Value	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
Decimal Value	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

Hex Value	10	11	12	13	14	15	16	17	18	19	1A	1B	1C	1D	1E	1F
Decimal Value	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31

Numbering System Facts

Binary Number System (Base 2)

Computers natively use the binary numbering system to represent and process data. In binary, there are only two possible numbers for every number place: 0 or 1. For example, a three-bit binary number contains a combination of 0s and 1s. Because there are only two possible numbers for each slot, there are eight possible combinations for a three-bit binary number:

- 000
- 001
- 010
- 011
- 100
- 101
- 110
- 111

The following table lists several binary values and their decimal equivalents:

Binary Value	Decimal Value
10000000	128
01000000	64
00100000	32
00010000	16
00001000	8
00000100	4

00000010	2
00000001	1

To find the decimal value of a binary number, simply add the decimal values of the 1 bits in the number. For example, the decimal value of the binary number 10010101 is:

10000000 = 128

00010000 = 16

00000100 = 4

00000001 = 1

Total: 128 + 16 + 4 + 1 = 149

Because there are only two possible values in a binary number (0 and 1), you can express binary numbers in terms of powers of two:

# of bits	Exponent	Exponent value
1	2^1	2
2	2^2	4
3	2^3	8
4	2^4	16
5	2^5	32
6	2^6	64
7	2^7	128
8	2^8	256
9	2^9	512
10	2^{10}	1024
11	2^{11}	2048

12	2^{12}	4096

Octal Number System (Base 8)

The octal numeral system is a base-8 number system and uses the digits 0 to 7. Octal numerals can be made from binary numerals by grouping consecutive binary digits into groups of three (starting from the right). For example, the binary representation for 74 is 1001010. You can add two zeros to the left: (00)1 001 010. This makes the corresponding octal digits 1 1 2, creating the octal representation 112.

Hexadecimal Number System (Base 16)

The hexadecimal numbering system is also frequently used with computers and networking. Hexadecimal is a base-16 numbering system, which means there are 16 different characters possible for each number place. These characters go from 0 to 9, as decimal does; however, hexadecimal uses the letter A to represent the decimal number 10. B represents 11 and so on up to F, which represents 15. The easiest way to convert between decimal, binary, and hexadecimal is to memorize the corresponding values for each hexadecimal number using the following table:

Hexadecimal Value	Binary Value	Decimal Value
0	0000	0
1	0001	1
2	0010	2
3	0011	3
4	0100	4
5	0101	5
6	0110	6
7	0111	7
8	1000	8
9	1001	9

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