MAKE
SCHOOL

NUMBERS

All your base are belong to us.

MAKE SCHOOL

BASE-10

Decimal

We speak base-10.

We have 10 digits in base-10. (0 - 9)

9 rolls over to 10. 99 rolls over to 100.

MAKE SCHOOL

BASE-2

Binary

Computers speak base-2.

There are 2 digits in base-2. (0, 1)

1 rolls over to 10. 11 rolls over to 100.

“There are 10 kinds of people in the world. Those who understand binary, and those who don’t.”

–Old Programming Proverb

COUNTING IN BINARY

0, 1, 10, 11, 100, 101, 110, 111, 1000, 1001, 1010, 1011, 1100, 1101, 1110, 1111, 10000, 10001, 10010, 10011, 10100, 10101, 10110, 10111, 11000, 11001, 11010, 11100, 11101, 11110, 11111, 100000, 100001, 100010, 100011, 100100, 100101, 100110, 100111, 101000, 101001, 101010, 101011, 101100, 101101, 101110, 101111, 110000, 110001, 110010, 110011, 110100, 110101, 110110, 110111, 111000, 111001, 111010, 111011, 111100, 111101, 111110, 111111...

BINARY TO DECIMAL

1 0 1 0 1 0 0 1 0 0 1 1

*

2^{11} 2^{10} 2^{9} 2^{8} 2^{7} 2^{6} 2^{5} 2^{4} 2^{3} 2^{2} 2^{1} 2^{0}

=

2048 0 512 0 128 0 0 16 0 0 2 1

=

2707

BASE-16

Hexadecimal

There are 16 digits in base-16 (0 - F)

$A_{16} = 10_{10}$

$B_{16} = 11_{10}$

$C_{16} = 12_{10}$

$D_{16} = 13_{10}$

$E_{16} = 14_{10}$

$F_{16} = 15_{10}$

BASE-16

Hexadecimal

Hex is often prefixed **`0x`**

$$\texttt{0xAF320100} == \texttt{AF320100}_{16}$$

COUNTING IN HEX

0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 1A, 1B, 1C, 1D, 1E, 1F, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 2A, 2B, 2C, 2D, 2E, 2F, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 3A, 3B, 3C, 3D, 3E, 3F, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 5A, 5B, 5C, 5D, 5E, 5F

HEX TO BINARY

Every hex digit is 4 binary digits

$5_{16} \leftrightarrow 0101_2$	$72_{16} \leftrightarrow 0111\ 0010_2$
$B_{16} \leftrightarrow 1011_2$	$A6_{16} \leftrightarrow 1010\ 0110_2$
$F_{16} \leftrightarrow 1111_2$	$FF_{16} \leftrightarrow 1111\ 1111_2$
$8_{16} \leftrightarrow 1000_2$	$10_{16} \leftrightarrow 0001\ 0000_2$

NEGATIVE INTEGERS

SIGNED MAGNITUDE

Most significant bit (leftmost) indicates sign

`1` is negative, `0` is positive

Range is $-(2^{n-1} - 1)$ to $2^{n-1} - 1$

Two `0’s`: `10000000` and `00000000`

SIGNED MAGNITUDE

$72_{10} \leftrightarrow 0100\ 1000_2$

$-72_{10} \leftrightarrow 1100\ 1000_2$

$-1_{10} \leftrightarrow 1000\ 0001_2$

$-0_{10} \leftrightarrow 1000\ 0000_2$

ONES' COMPLEMENT

Negative number is positive one with bitwise NOT applied

Range is $-(2^{n-1} - 1)$ to $2^{n-1} - 1$

Two `0`'s `11111111` and `00000000`

ONES' COMPLEMENT

$72_{10} \leftrightarrow 0100\ 1000_2$

$-72_{10} \leftrightarrow 1011\ 0111_2$

$-1_{10} \leftrightarrow 1111\ 1110_2$

$-0_{10} \leftrightarrow 1111\ 1111_2$

ONES' COMPLEMENT ADDITION

Add the two numbers

If there's a carry, do an *end-around carry* (add it back in to the sum)

ONES' COMPLEMENT ADDITION

$$
\begin{array}{rrr}
 & 1111\ 1010_2 & -5_{10} \\
+ & 0000\ 1000_2 & 8_{10} \\
\hline
 & {\color{blue}1}\ 0000\ 0010_2 & 2_{10} \\
+ & 1_2 & 1_{10} \\
\hline
 & 0000\ 0011_2 & 3_{10}
\end{array}
$$

MAKE SCHOOL

TWOS' COMPLEMENT

Standard way most processors represent negative numbers

Addition, subtraction, multiplication algorithms are the same as the ones for unsigned integers

Range is $-(2^{n-1})$ to $2^{n-1} - 1$

One representation of **`0: 00000000`**

TWOS' COMPLEMENT NEGATION

To negate a twos' complement number, flip all the bits and add one.

Slightly Easier Version

1. Starting from the right, find the first 1

2. Invert all the bits to the left of that 1

TWOS' COMPLEMENT

$72_{10} \leftrightarrow 0100\ 1000_2$

$-72_{10} \leftrightarrow 1011\ 1000_2$

$-1_{10} \leftrightarrow 1111\ 1111_2$

$0_{10} \leftrightarrow 0000\ 0000_2$

RESOURCES

http://en.wikipedia.org/wiki/Signed_number_representations

http://en.wikipedia.org/wiki/Two%27s_complement

MAKE
SCHOOL