Integer Numbers

- Integers are almost universally stored as binary numbers.
- When you enter an integer from the keyboard, software converts the ASCII or Unicode characters to a binary integer.

Decimal Numbers

- Some systems *(i.e. Intel Pentium)* support decimal numbers in packed or unpacked format.

packed decimal uses 4 bit fields for each digit

9375 =

1001,0011,0111,0101

unpacked decimal uses a byte per digit (ASCII)

9375 =

00111001,00110011,00110111,00110101

Negative Integers

- Almost all systems for storing negative binary numbers set the left most bit (MSB) to indicate the sign of a number.

Common formats:

- Signed Magnitude
- Ones Complement
- Twos Complement

Signed Magnitude

- Negative number are the same as positive with the sign bit set

Three bit example

000	001	010	011	100	101	110	111
0	1	2	3	-0	-1	-2	-3

Normal addition does not work for negative numbers.

001 + 101 = 110 1 + (-1) = -2

Adding negative zero changes a number.

Ones Complement

- Negative number are the logical inverse of positive numbers

Three bit example

000	001	010	011	100	101	110	111
0	1	2	3	-3	-2	-1	-0

Mathematically positive and negative zero are the same, but they are different bit patterns.

Ones Complement Arithmetic

- The carry out of the sign position needs to be added to the number

101	-2	
+110	+ -1	
011		
+1		add carry out of sign
100	-3	

Twos Complement

- Negative number are the logical inverse of positive numbers plus 1.

Three bit example

000	001	010	011	100	101	110	111
0	1	2	3	-4	-3	-2	-1

Normal binary arithmetic works for positive and negative numbers.

Byte Ordering

- Some systems store the least significant byte first (Little Endian). Others store the most significant byte first (Big Endian).
- The decimal number 258 (0100000010_2) would be stored in as a 16 bit binary number.

00000001 00000010 Big Endian

00000010 00000001 Little Endian

Floating Point

- Almost all systems today use the IEEE 754 standard for floating point numbers.
- The standard supports 32 bit (single precision) and 64 bit (double precision).
- Standard committee is working on 128 bit
- In many engineering applications, accuracy is extremely important.

Floating Point Format

Single Precision up to 2.0×10^{38}

sign	exponent	mantissa
1	8 bits	23 bits

Double Precision up to 2.0×10^{308}

sign	exponent	mantissa
1	11 bits	53 bits

Normalization

- The mantissa or fraction part of a floating point number is normalized. This means that the fraction is shifted and the exponent adjusted so that the left most bit is a one.
- Since the left most bit is always a one, it is not stored in the number. It is assumed to always be there.

Special Values

Exponent	**Mantissa**	**Purpose**
0	0	Zero
255	0	±Infinity
255	nonzero	NaN not a number