SOME MORE COMPUTER ARITHMETIC (OOPS !)

TWO'S COMPLEMENT

The method used to represent signed numbers (positive and negative signs) in microprocessors is called *two's complement*. In this system, positive numbers are represented as regular binary numbers. Negative numbers are represented as the two's complement of positive numbers.

The two's complement of a number is formed by taking the one's complement and then adding 1_2 . The one's complement is formed by replacing all 1's with 0's, and all 0's with 1's. For example, the two's complement of 04_{10} is found in the following sequence:

00000100	⇐	04_{10} in binary
11111011	⇐	one's complement of 04_{10}
11111011		
+ 1	⇐	add 1_2 to one's complement of 04_{10}
11111100	⇐	resultant two's complement of 04_{10}
	=	-04_{10}

Two's Complement Arithmetic

The real beauty of the two's complement system is illustrated when you add numbers with unlike signs. For example, assume an 8-bit microprocessor is to add +7 and –3. Since these are singed numbers they must be represented in two's complement form. The arithmetic is shown below:

00000111	⇐	$+07_{10}$ in binary
+11111101	⇐	-03_{10} in two's complement form
100000100	⇐	result is $+04_{10}$
⇑		
Discard final entry		

Notice that the sum is correct if you ignore the final carry bit. Keep in mind that microprocessors adds the two numbers as if they were unsigned binary numbers. It is merely our interpretation of the answer that makes the systems work for signed numbers.

00000111	⇐	$+07_{10}$ in binary
+11111101	⇐	-03_{10} in two's complement form
100000100	⇐	result is $+04_{10}$
⇑		
Discard final entry		

The system also works when the magnitude of the negative number is larger for example, when −9 is added to +8 the result is −1. The procedure follows:

11110111	$\Leftarrow$	-09_{10} in two's complement form
$\underline{+00001000}$	$\Leftarrow$	$+08_{10}$
11111111	$\Leftarrow$	result is -01_{10} in two's complement form

The last example involves adding two negative numbers:

11111101	$\Leftarrow$	-03_{10} in two's complement form
$\underline{+11111100}$	$\Leftarrow$	-04_{10} in two's complement form
$\underline{1}11111001$	$\Leftarrow$	result is -07_{10} in two's complement form

$\Uparrow$
Discard final carry

Remember the capacity of the microprocessor when you add two negative numbers. The largest negative number that can be represented by 8-bits is -128_{10} .