MITOCW | MIT6_004S17_01-02-12-02_300k

Two's complement representation is a way of representing both positive and negative numbers in binary.

Let's take a look at an example of a positive number represented in two's complement in order to begin understanding how this representation works.

If we look at the number, 001000, the way that we determine the value of this two's complement number is as follows.

Each bit represents a power of 2, with the right most bit representing 20, the next bit is 21, then 22, 23, 24, and finally -25.

Having the most significant bit represent a negative power of 2 rather than a positive power allows us to express both positive and negative numbers with our notation.

If your number has a 0 in a certain position, then the power of 2 associated with that position is not present in your number.

If the number has a 1, then that tells us that the corresponding power of 2 makes up part of the number's value.

To figure out the value of the number 001000, we see that there is a single 1 which is in the 23 position.

So this number is equal to 23 which equals 8.

Note that if I wanted to represent the same number 8 using a larger number of binary bits, I would simply add some leading zeroes to my positive number.

For example, the number 8 represented in 8-bits two's complement is the same as the 6-bit representation but with 2 additional leading zeroes.

Now let's try a negative number.

Negative numbers in 2's complement representation always have a 1 in their most significant bit (MSB).

What is the value of 101100?

We apply the same logic as before, now we have a 1 in bit positions 2, 3, and 5.

That means that this number is equal to -25 + 23 + 22 which equals -32 + 8 + 4 = -20.

In the case of negative numbers, if we wanted to represent the same value, -20 using more bits, one would simply need to append leading 1's to the number to end up with the same value.

For example, -20, using 8 bits is 11101100.

Let's check that this is correct.

This number is equal to -27 + 26 + 25 + 23 + 22 which is equal to -128 + 64 + 32 + 8 + 4 which is -128 + 108 and that is equal to -20.

Let's take a look at the range of numbers that can be represented using N-bit 2's complement.

Let's begin with our example of 6-bits.

The largest positive number in that case is 0 followed by five 1's (011111) which is equal to 20 + 21 + 22 + 23 + 24 which is equal to 1 + 2 + 4 + 8 + 16 which equals 31 which is also equal to 25-1.

The most negative number using 6 bits is 1 followed by five 0's (100000) which equals -25 with no positive component added to the number and that is equal to -32.

So in general for N-bits, the 2's complement number can range from -2N-1 through 2N-1-1.

An easy way of converting between positive and negative 2's complement numbers is by flipping all the bits and adding 1.

For example, earlier we calculated that 101100 = -20.

If I take this number and flip all of the bits, I get 010011.

I then add 1 to that, which results in 010100.

This number is = 22 + 24 = 4 + 16 = 20.

This tells us that our original number 101100 was equal to -20.

The same methodology can be applied in reverse.

If you start with a positive number 20 = 010100, and you flip all the bits and add 1, you get 101011 + 1 = 101100.

This brings us back to our original number which is -20.

So to easily determine the value of a negative 2's complement number, one flips all the bits and adds 1 to find the corresponding positive number, say X, which in turn means that your original negative number had a value of -X.