

Representation of negative numbers

Signed-Magnitude representation:

- This is the simplest method.
- Write the magnitude of the number in binary. Then add a 1 to the front of it if the number is negative and a 0 if it is positive.
- Examples: +7 would be 111 and then a 0 in front so 00000111 for an 8-bit representation.
-9 would be 1001 (+9) and then a 1 so 10001001 for an 8-bit representation.
- It is not the best method or representation because it makes computation awkward.

2's complement representation:

- Most widely used method of representation.
- Positive numbers are represented as they are (simple binary).
- To get a negative number, write the positive number in binary, then change all 0's to 1's and 1's to 0's. Then add 1 to the number.
- Example : +7 would be 0111 in 4-bit 2's complement.
To represent -5 we take +5 (0101) and then invert the digits (1010) and add 1 (1011). -5 is thus 1011.
- Suppose you already have a number that is in two's complement representation and want to find its value in binary.

If the number starts with a 1 it is a negative number. If it starts with a 0 it is a positive number.

If it is a negative number, take the 2's complement of that number. You will get the number in ordinary binary. The sign you already know. Let's take 1101. Take the 2's complement and you get 0011. Since it started with a 1, it was negative and the value is 0011 which is 3. The number represented by 1101 is -3 in 2's complement.

Lets see how this system is better:

If we add +5 and -5 in decimal we get 0.

Let's add them in 4-bit signed-magnitude. +5 is 0101 and -5 is 1101. On adding we get 10010. That is not zero.

Let's do the same thing in 2's complement. Adding 0101 (+5) and 1011 (-5) gives 10000. If we discard the carry of 1 we get 0000 – i.e. 0.

Thus addition works out ok for negative numbers in 2's complement whereas it doesn't in sign magnitude.

Similarly, you can show that multiplication and subtraction all work in 2's complement but do not in other representations. The other number systems require much more complicated hardware to implement basic mathematical functions. i.e. add/subtract/multiply.