|  |  |
| --- | --- |
|  | How does 0 have two values in one's complement?  In **1's complement** you just invert all the bits.  Consider these 2 examples (assuming 8 bits):   * 4=000001004=00000100, so −4=11111011−4=11111011 * 0=000000000=00000000, so −0=11111111−0=11111111.   So you have 2 ways to represent the number 0  In **2's complement** you add 1 to the 1's complement representation of the negative number   * −4−4 that in 1's complement was 1111101111111011 becomes 1111110011111100 * −0−0 that in 1's complement was 1111111111111111 becomes 0000000000000000 same as 0   So you have just one way to represent the 0 in this case  As you can see from the examples the difference is that:   * in 1's complement, with 8 bits you can just express numbers from −27+1−27+1 to 27−127−1(from -127 to 127) * in 2's complement with 8 bits you can express numbers from −27−27 to 27−127−1 (from -128 to 127), so one more number   Reason for end around carry to do  End-around carry is actually rather simple: it changes the modulus of the addition operation from rn to rn–1, if you think of the numbers as unsigned. To simplify things, let's talk about binary.  Let's compute (-2) + (-4) using four-bit two's complement arithmetic:  1 1 1 0 (-2)  + 1 1 0 0 + (-4)  --------- ------  1 1 0 1 0 (-6)  Let's keep the carry bit where it is for now. If you look at the numbers as unsigned integers, we're computing 14 + 12 = 26. However, addition is done modulo 16, so we get 10, which is the unsigned number which represents -6 (the correct result).  In ones' complement, the numbers have different representations:  1 1 0 1 (-2)  + 1 0 1 1 + (-4)  --------- ------  1 1 0 0 0 (-6)  Again, let's keep the carry bit where it is. If you look at the numbers as unsigned integers, we're computing 13 + 11 = 24. However, due to the wrap-around carry, addition is done modulo 15, so we end up with 9, which represents -6 (the correct result).  So in four-bit two's complement, -2 is equivalent to 14 modulo 16, -4 is equivalent to 12 modulo 16, and -6 is equivalent to 10 modulo 16.  And in four-bit ones' complement, -2 is equivalent to 13 modulo 15, -4 is equivalent to 11 modulo 15, and -6 is equivalent to 9 modulo 15.  **Signed zero:** The reason you get "signed zero" is because there are 16 possible numbers in four bit, but if you're doing modulo-15 arithmetic, then 0 and 15 are equivalent. That's all there is to it. |