The Two's Complement System (D)

The next system to be discussed is also a "Signed magnitude" system ***(at least when positive)***. However, the **Two's Complement** system has a method of representing a negative # in such a way that it can be **treated just like it was a positive number**, thereby:

- saving on hardware costs and
- reducing the speed loss,

during subtraction as well as during addition.

Method of converting into the 2's comp. system

The first thing to remember about the **two's complement system** is that it **IS A SYSTEM**. That means it has both positive and negative numbers in it. It is denoted by a subscript of **2cm** and its positive numbers resemble exactly the positive numbers in the signed magnitude system. And just as in the **2sm** system, a **positive number** is represented by a **'0',** and a **negative number** is represented by a '**1**'. However, unlike in the 2sm system, this time <u>**the '1' carries the position weight that it occupies**</u>.

Remember that in the "**2sm**" system the sign bit didn't have a weight. You just applied a negative sign, a '1', to the binary number if it was desired to make it negative. The 2's complement system is MUCH different!

Let's look at how we go about converting a positive number into **the Two's Complement System.**

2's Complement Example 1

Convert **+23**$_{10}$ into the **2's complement system** (**N=7**).

- Just as with the **2sm** system, the value of **N** (length of number and sign bit) is very important. The value of **N** must be large enough for all numbers in the operation, including the result.
- Convert the decimal number into binary.

$$23_{10} \Rightarrow \quad 10111_2$$

- The binary result is **5-bits** long. Since we will need to have a total of 7 bits and the sign bit uses up one of those, we will need to pad the left side of the number with a **0**.

$$23_{10} \Rightarrow \quad 010111_2$$

- Now all that is necessary is to tack on the sign bit, a '**0**', along with a comma to separate it from the rest of the number. The addition of the **2cm** subscript will complete the conversion of **+23** into the **2's complement system**.

$$+23_{10} \Rightarrow \quad \underbrace{0{,}010111_{2cm}}_{N=7}$$

- That is all there is to it! It will be necessary to start out all **2's complement** conversions this way. Even if the number is negative, it will be necessary to start out with it as a **positive 2's complement number** before finding its **negative 2's complement equivalent**.

Finding the negative 2's complement (Long-hand method)

Now let's add to the problem by finding the **negative 2's complement**. There is a **Long-hand** and a **Short-hand** method. We will start out with the Long-hand method.

2's Complement Example 2

Find the **negative 2's complement** of **+23_{10} (N = 7)** using the **Long-hand method**.

- In the last example we found the **positive 2's complement** of **+23 (N = 7)** to be:

$$+23 = 0{,}010111_{2cm}$$

- Next we find the 1's Complement (**1cm**). The 1's Complement is found by inverting each bit.

$$\begin{array}{cccccccc} 0, & 0 & 1 & 0 & 1 & 1 & 1_{2cm} \\ 1, & 1 & 0 & 1 & 0 & 0 & 0_{1cm} \end{array}$$

- Now all that is necessary is to **add a one** to the 1's Complement to find its **negative 2's Complement.**

$$\begin{array}{lcccccc} 1, & 1 & 0 & 1 & 0 & 0 & 0_{1cm} \\ + & & & & & & 1 \\ \hline 1, & 1 & 0 & 1 & 0 & 0 & 1_{2cm} \end{array}$$

$$-23_{10} = \boxed{1{,}101001_{2cm}}$$

The method we just covered was the "**long hand**" method for finding the negative of a number in the 2's complement system. However, there is a much easier and faster method called the **shorthand method**. The use of this method is required for this course.

Finding the negative number in 2's comp. system (Shorthand method)

Now we need to take a look at a short-hand method to achieve the same result.

The procedure is as follows:

- **Start at the farthest bit to the right (the LSB)**
- **Copy all bits up to and including the first 1.**
- **Invert the rest of the bits.**

2's Complement Example 3

Find the **negative 2's Complement** of **+23_{10} (N = 7)** using the **Short-hand Method**

- We will start out with the number already in the positive 2's complement form as derived in an earlier example:

$$+23_{10} = \underbrace{0,010111_{2cm}}_{N=7}$$

- Now it is time to find the negative 2's complement of our number. Remember to start at the LSB and copy all the bits up to and including the first 1. In this case, all that was necessary was to copy the LSB.

$$\begin{array}{r} 0,\ 0\ 1\ 0\ 1\ 1\ 1_{2cm} \\ \hline 1_{2cm} \end{array}$$

- Now, we complete the negation by inverting the rest of the bits:

$$\begin{array}{r} 0,\ 0\ 1\ 0\ 1\ 1\ 1_{2cm} \\ \hline 1,\ 1\ 0\ 1\ 0\ 0\ 1_{2cm} \end{array}$$

$$so, \quad -23_{10} = 1,101001_{2cm}$$

Let's take a look at a number with some 0's in the lower nibble.

2's Complement Example 4

Find the **negative 2's complement** of $0,11000110000_{2cm}$ using the **Short-hand method.**

- Start with the **LSB, copy all bits up to and including the first 1:**

$$\begin{array}{r} 0,\ 1\ 1\ 0\ 0\ 0\ 1\ 1\ 0\ 0\ 0\ 0_{2cm} \\ \hline 1\ 0\ 0\ 0\ 0_{2cm} \end{array}$$

- Now we complete the negation by inverting the rest of the bits:

$$\begin{array}{r} 0,\ 1\ 1\ 0\ 0\ 0\ 1\ 1\ 0\ 0\ 0\ 0_{2cm} \\ \hline 1,\ 0\ 0\ 1\ 1\ 1\ 0\ 1\ 0\ 0\ 0\ 0_{2cm} \end{array}$$

- Note that the resulting value did indeed have a negative bit on the front.

Converting from a neg. 2's complement system number to base 10.

Before we do another **2cm** example, let's look at how we can convert **directly** from a **negative 2cm** to **decimal**.

- **First, we determine the position weight of the sign bit**. In the example below, the position weight of the sign bit in the 7-bit 2cm representation of (-23) is 2^6 or **64**. Since it was negative, we make it -64.

$$-23_{10} = \underbrace{1,101001_{2cm}}_{N=7} = \underbrace{\overset{2^6}{1}}_{-64},\overset{2^5}{1}\,0\,\overset{2^3}{1}\,00\,\overset{2^0}{1}_{2cm}$$

- Then we throw in the positive weights of the rest of the 1's:

$$-23_{10} = \underbrace{1,101001_{2cm}}_{N=7} = \underbrace{\overset{2^6}{1}}_{-64},\underbrace{\overset{2^5}{1}}_{+32}\,0\,\underbrace{\overset{2^3}{1}}_{+8}\,00\,\underbrace{\overset{2^0}{1}_{2cm}}_{+1}$$

- Then we add to it the weights together:

$$-23_{10} = \underbrace{1,101001_{2cm}}_{N=7} = \underbrace{\overset{2^6}{1}}_{-64},\underbrace{\overset{2^5}{1}}_{+32}\,0\,\underbrace{\overset{2^3}{1}}_{+8}\,00\,\underbrace{\overset{2^0}{1}_{2cm}}_{+1} = -64 + 32 + 8 + 1 = \boxed{-23_{10}}$$

- We ended up with what we started with.

More examples from start to finish in the 2's complement system.

2's Complement Example 5

Find the **negative 2's complement** of 13.75_{10} **(N=8)** and then verify your answer by converting the result **directly** back to **base 10**.

- 1^{st} find the **positive 2's complement** of the number by finding the 7-bit binary equivalent and tacking on a **0** as a sign bit.

$$+13.75_{10} \Rightarrow \underbrace{0,\ 1\ 1\ 0\ 1\ .\ 1\ 1\ 0_{2cm}}_{N=8}$$

- Next start with the **LSB** and **copy all bits up to and including the first 1**:

$$+13.75_{10} \Rightarrow 0,\ 1\ 1\ 0\ 1\ .\ 1\ 1\ 0_{2cm}$$
$$1\ 0_{2cm}$$

- Now, invert the rest of the bits:

$$+13.75_{10} \Rightarrow 0,\ 1\ 1\ 0\ 1\ .\ 1\ 1\ 0_{2cm}$$
$$-13.75_{10} \Rightarrow 1,\ 0\ 0\ 1\ 0\ .\ 0\ 1\ 0_{2cm}$$

- Now, verify the result by converting the result directly back to base 10:

$$-13.75_{10} \Rightarrow \underbrace{\overbrace{1}^{2^4}}_{-16},\ 0\ 0\ \underbrace{\overbrace{1}^{2^1}}_{+2}\ 0\ .\ 0\ \underbrace{\overbrace{1}^{2^{-2}}}_{+.25}\ 0_{2cm}$$

$$-16 + 2 + .25 = -14 + .25 = -13.75_{10} \Leftarrow \textit{VERIFIED}$$

2's Complement Example 6

Find the **negative 2's complement** of 44.25_{10} **(N=9)** and then verify your answer by converting the result **directly** back to **base 10**.

- 1st find the **positive 2's complement** of the number by finding the 7-bit binary equivalent and tacking on a **0** as a sign bit.

$$+44.25_{10} \Rightarrow \underbrace{0,\ 1\ 0\ 1\ 1\ 0\ 0\ .\ 0\ 1_{2cm}}_{N=9}$$

- Next start with the **LSB** and **copy all bits up to and including the first 1**:

$$+44.25_{10} \Rightarrow 0,\ 1\ 0\ 1\ 1\ 0\ 0\ .\ 0\ 1_{2cm}$$
$$1_{2cm}$$

- Now, invert the rest of the bits:

$$+44.25_{10} \Rightarrow 0,\ 1\ 0\ 1\ 1\ 0\ 0\ .\ 0\ 1_{2cm}$$
$$-44.25_{10} \Rightarrow 1,\ 0\ 1\ 0\ 0\ 1\ 1\ .\ 1\ 1_{2cm}$$

- Now, verify the result by converting the result directly back to base 10:

$$-44.25_{10} \Rightarrow \underset{-64}{\overset{2^6}{1,}}\ 0\ \underset{+16}{\overset{2^4}{1}}\ 0\ 0\ \underset{+2}{\overset{2^1}{1}}\ \underset{+1}{\overset{2^0}{1}}\ .\ \underset{0.5}{\overset{2^{-1}}{1}}\ \underset{0.25}{\overset{2^{-2}}{1_{2cm}}}$$

$$-64 + 16 + 2 + 1 + 0.5 + 0.25 =$$
$$-64 + 19 + 0.75 =$$
$$-45 + 0.75 = -44.25_{10} \Leftarrow \textit{VERIFIED}$$