9's Complement Summary
==========
As you are aware, when dealing with Unsigned Numbers, with 3 digits in Decimal system, we can generate numbers from 000 to 999.

But these are all positive (Unsigned) numbers.

Setting up the system
Imagine we are designing a 3 digit Decimal Based computer.

To introduce the Signed numbers (Both Negative and Positive) let's assume that we will use half of these numbers as negative and half as positive.

Dividing the range
We can then arbitrarily choose a criterion for the distribution (but, in reality it is of more benefit to generate roughly equal numbers of negative and positive numbers) - let's say - we will divide the range in middle:

Unsigned Range: Total 1000 Numbers
000 999

Signed Range: Total 1000 Numbers, 500 on the negative side, 500 on positive side.

-499 .....-000 | +000 .....+499

Now that we have decided how we divide the numbers on the both side of zero, our next challenge is how can we tell our decimal computer about the negative and positive signs – remember it is easy for us to see the +/- sign and interpret – computers don't have that ability – only thing it see/understand is numbers.

Setting up the Representation Rule
So our challenge now is to come up with a way to represent the negative numbers (we are not worried about the positive numbers because of the following logic: any number is not negative, must be a positive number). This introduces the concept of "Representation of the numbers". For now, we will concentrate on 9's complement representation method.

A general formula we can use for the negative number representation is to obtain the representative number by subtracting the negative number we want to represent from the highest radix value of the system.

For example, in our 3 digit decimal system, our highest radix value is 999.
To find a representation for -499, we can then use the formula stated above.
9's complementary representation of -499 is, therefore:

```
 999
-499
------
 500
```

Another example: 9's complementary representation of -000 is therefore:

```
 999
-000
--------
 999
```

Now we can actually store the representation numbers on our 3 digit decimal computer and found a way to deal with the -/+ signs.

Let's review what we have done using the number scale using 9's complement method:

Number to represent: -499 .....-000 | +000 .....+499
Number's representation: 500 ……. 999 | 000 ……..499

Identifying negative/positive numbers from the representation/Finding Sign and Magnitude of a represented number:

From the above scale, one thing should be very clear now: In our representation, any number beginning with 0 to 4 will represent positive numbers (000 represents 0 and 499 represents 499), and any number beginning with 5 to 9 will represent negative number (500 represents -499, 999 represents -0).

Let's take a look at some more examples:

We need to add -350 with +250 in our 3 digit decimal system. Of course, by looking at it, you know the answer is -100. Again, the computer does not know how to deal with the (–) sign, we need to represent the negative numbers in 9's complimentary form and do the math.

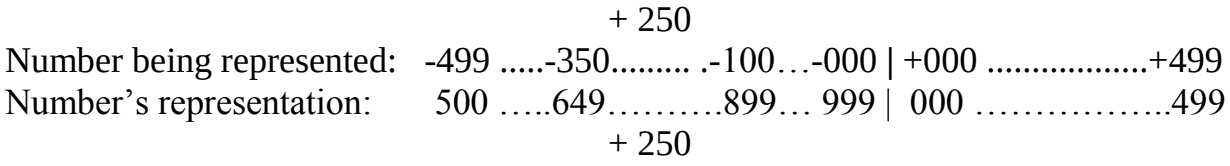
-350 can be represented as 999-350 = 649
We add 250 with 649 (649 + 250) = 899

899 is a representation of a negative number. Let's find out what it represents.

999 – 899 = 100. Since the representation was 899, whose MSD is 8, which is more than 4, it must have been a negative number. Therefore 899 represents -100.

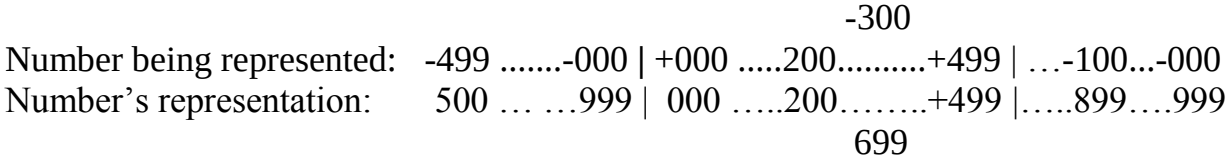
Therefore, now we are able to find the sign and magnitude of the number that was being represented.

The whole discussion in this section is summarized below using the number scale:



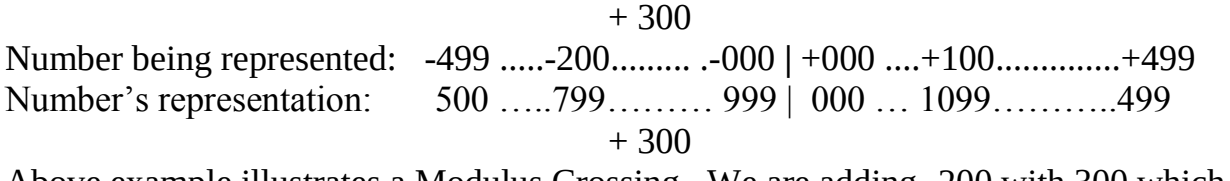
Next Topic: Wraparound and Overflow.

Wraparound:



Above example shows the wraparound case. We are adding -300 with 200. 200 – 300 is -100. To show this in action, we can represent -300 in 9's complement form 999-300 = 699. When we add 200 with 699, we get 899 which crosses the highest number 499 on the representation scale (remember, highest we can represent for a positive number in this system is 499 and positive numbers do not need any representation). With the rule set up in previous section, we now know 899 is representing a negative number (Most Significant Digit > 4). Therefore, it must be representing 999 – 899 = -100!

Modulus Crossing



Above example illustrates a Modulus Crossing. We are adding -200 with 300 which results in 100. Representation of -200 is 999-200 = 799 and adding 300 to it is 1099. But remember, we are only dealing with a 3-digit decimal system, which can only handle 3-digits max. Therefore the carry, [1]099 will be dropped and added back to the number 099 + 1 = 100. This process is known as End-Around-Carry.