Binary Numbers Representation

We can make the binary numbers into the following two groups − **Unsigned numbers** and **Signed numbers**.

Unsigned Numbers

Unsigned numbers contain only magnitude of the number. They don't have any sign. That means all unsigned binary numbers are positive. As in decimal number system, the placing of positive sign in front of the number is optional for representing positive numbers. Therefore, all positive numbers including zero can be treated as unsigned numbers if positive sign is not assigned in front of the number.

Signed Numbers

Signed numbers contain both sign and magnitude of the number. Generally, the sign is placed in front of number. So, we have to consider the positive sign for positive numbers and negative sign for negative numbers. Therefore, all numbers can be treated as signed numbers if the corresponding sign is assigned in front of the number.

If sign bit is zero, which indicates the binary number is positive. Similarly, if sign bit is one, which indicates the binary number is negative.

Representation of Un-Signed Binary Numbers

The bits present in the un-signed binary number holds the **magnitude** of a number. That means, if the un-signed binary number contains **'N'** bits, then all **N** bits represent the magnitude of the number, since it doesn't have any sign bit.

Example

Consider the **decimal number 108**. The binary equivalent of this number is **1101100**. This is the representation of unsigned binary number.

$$108108_{10} = 11011001101100_2$$

It is having 7 bits. These 7 bits represent the magnitude of the number 108.

Representation of Signed Binary Numbers

The Most Significant Bit MSBMSB of signed binary numbers is used to indicate the sign of the numbers. Hence, it is also called as **sign bit**. The positive sign is represented by placing '0' in the sign bit. Similarly, the negative sign is represented by placing '1' in the sign bit.

If the signed binary number contains 'N' bits, then $\mathrm{N-1}$N−1 bits only represent the magnitude of the number since one bit MSBMSB is reserved for representing sign of the number.

There are three **types of representations** for signed binary numbers

Sign-Magnitude form

1's complement form

2’s complement form

Representation of a positive number in all these 3 forms is same. But, only the representation of negative number will differ in each form.

Example

Consider the **positive decimal number +108**. The binary equivalent of magnitude of this number is 1101100. These 7 bits represent the magnitude of the number 108. Since it is positive number, consider the sign bit as zero, which is placed on left most side of magnitude.

$$+108+108_{10} = 0110110001101100_2$$

Therefore, the **signed binary representation** of positive decimal number +108 is $\mathbf{0\ 1101100}$. So, the same representation is valid in sign-magnitude form, 1’s complement form and 2’s complement form for positive decimal number +108.

Sign-Magnitude form

In sign-magnitude form, the MSB is used for representing **sign** of the number and the remaining bits represent the **magnitude** of the number. So, just include sign bit at the left most side of unsigned binary number. This representation is similar to the signed decimal numbers representation.

Example

Consider the **negative decimal number -108**. The magnitude of this number is 108. We know the unsigned binary representation of 108 is 1101100. It is having 7 bits. All these bits represent the magnitude.

Since the given number is negative, consider the sign bit as one, which is placed on left most side of magnitude.

$$-108-108_{10} = 1110110011101100_2$$

Therefore, the sign-magnitude representation of -108 is **11101100**.

1’s complement form

The 1’s complement of a number is obtained by **complementing all the bits** of signed binary number. So, 1’s complement of positive number gives a negative number. Similarly, 1’s complement of negative number gives a positive number.

That means, if you perform two times 1’s complement of a binary number including sign bit, then you will get the original signed binary number.

Example

Consider the **negative decimal number -108**. The magnitude of this number is 108. We know the signed binary representation of 108 is 01101100.

It is having 8 bits. The MSB of this number is zero, which indicates positive number. Complement of zero is one and vice-versa. So, replace zeros by ones and ones by zeros in order to get the negative number.

$$-108-108_{10} = 1001001110010011_2$$

Therefore, the **1’s complement of** 108108_{10} is 1001001110010011_2.

2’s complement form

The 2’s complement of a binary number is obtained by **adding one to the 1’s complement** of signed binary number. So, 2’s complement of positive number gives a negative number. Similarly, 2’s complement of negative number gives a positive number.

That means, if you perform two times 2’s complement of a binary number including sign bit, then you will get the original signed binary number.

Example

Consider the **negative decimal number -108**.

We know the 1’s complement of (**108**)$_{10}$ is (**10010011**)$_2$

2’s compliment of $108$108$_{10}$ *= 1’s compliment of* $108$108$_{10}$ *+ 1.*

= 10010011 + 1

= 10010100

Therefore, the **2’s complement of** $108$108$_{10}$ is $10010100$1001010 $_{2}$.