LECTURE NO:-07

LECTURE NOTES ON

DIGITAL ELECTRONICS

Prepared by:
Arpita Singh
Roll no. 07
Msc. 1^{st} semester
DDU Gorakhpur University, GKP.

Under supervision of
Prof. Manish Mishra
Department of Electronics
DDU Gorakhpur University
Gorakhpur, (U.P.)

COMPLEMENTS

Complements are used in the digital computers in order to simplify the subtraction operation and for the logical manipulations. For each radix-r system (radix r represents base of number system) there are two types of complements.

There are two types of complements:

1. R's complement/Radix
2. (R-1)'s complement/Diminished Radix

Radix Complement and Diminished Radix Complement

The mostly used complements are 1's, 2's, 9's, and 10's complement. Apart from these complements, there are many more complements from which mostly peoples ares not familiar. For finding the subtraction of the number base system, the complements are used. If **r** is the base of the number system, then there are two types of complements that are possible, i.e., r's and (r-1)'s. We can find the r's complement, and (r-1)'s complement of the number, here r is the radix. The r's complement is also known as **Radix complement** (r-1)'s complement, is known as **Diminished Radix complement**.

If the base of the number is 2, then we can find 1's and 2's complement of the number. Similarly, if the number is the octal number, then we can find 7's and 8's complement of the number.

There is the following formula for finding the r's and (r-1)'s complement:

r' s= complement= $(r^n)_{10}$-N
(r-1)' s complement=$\{(r^n)_{10}-1\}$-N

In the above formulas,

- The n is the number of digits in the number.
- The N is the given number.

- The r is the radix or base of the number.

Advantages of r's complement

These are the following advantages of using r's complement:

- In r's complement, we can further use existing addition circuitry means there is no special circuitry.
- There is no need to determine whether the minuend and subtrahend are larger or not because the result has the right sign automatically.
- The negative zeros are eliminated by r's complement.

Let's take some examples to understand how we can calculate the r's and (r-1)'s complement of binary, decimal, octal, and hexadecimal numbers.

Example 1: $(1011000)_2$

This number has a base of 2, which means it is a binary number. So, for the binary numbers, the value of r is 2, and r-1 is 2-1=1. So, we can calculate the 1's and 2's complement of the number.

1's complement of the number 1011000 is calculated as:

$=\{(2^7)_{10}-1\}-(1011000)_2$
$=\{(128)_{10}-1\}-(1011000)_2$
$=\{(127)_{10}\}-(1011000)_2$
$=1111111_2-1011000_2$
$=0100111$

2's complement of the number 1011000 is calculated as:

$=(2^7)_{10}-(1011000)_2$
$=(128)_{10}-(1011000)_2$
$=10000000_2-1011000_2$
$=0101000_2$

Example 2: $(155)_{10}$

This number has a base of 10, which means it is a decimal number. So, for the decimal numbers, the value of r is 10, and r-1 is 10-1=9. So, we can calculate the 10's and 9's complement of the number.

9's complement of the number 155 is calculated as:

$=\{(10^3)_{10}-1\}-(155)_{10}$
$=(1000-1)-155$
$=999-155$
$=(844)_{10}$

10's complement of the number 1011000 is calculated as:

$=(10^3)_{10}-(155_{10}$
$=1000-155$
$=(845)_{10}$

Example 3: $(172)_8$

This number has a base of 8, which means it is an octal number. So, for the octal numbers, the value of r is 8, and r-1 is 8-1=7. So, we can calculate the 8's and 7's complement of the number.

7's complement of the number 172 is calculated as:

$=\{(8^3)_{10}-1\}-(172)_8$
$=((512)_{10}-1)-(132)_8$
$=(511)_{10}-(122)_{10}$
$=(389)_{10}$
$=(605)_8$

8's complement of the number 172 is calculated as:

$=(8^3)_{10}-(172)_8$
$=(512)_{10}-172_8$
$=512_{10}-122_{10}$

$=390_{10}$
$=606_8$

Example 4: $(F9)_{16}$

This number has a base of 16, which means it is a hexadecimal number. So, for the hexadecimal numbers, the value of r is 16, and r-1 is 16-1=15. So, we can calculate the 16's and 15's complement of the number.

15's complement of the number F9 is calculated as:

$\{(16^2)_{10}-1\}-(F9)_{16}$
$(256-1)_{10}-F9_{16}$
$255_{10}-249_{10}$
$(6)_{10}$
$(6)_{16}$

16's complement of the number F9 is calculated as:

$\{(16^2)_{10}\}-(F9)_{16}$
$256_{10}-249_{10}$
$(7)_{10}$
$(7)_{16}$

1) **r's complement**

The r's complement of a non-zero number in any number system with base r can be calculated by adding 1 to the LSB of its (r-1)'s complement.

For Example: In binary number system, **2's** complement of **001** can be calculated by adding 1 to the LSB of its 1'complement (i.e., **110 + 1**) = **$(111)_2$**.

Similarly, in octal number system, **8's** complement of **347** can be calculated by adding 1 to the LSB of its **7'complement** (i.e., **430 + 1**) = **$(431)_8$**.

2) (r-1)'s complement

The **(r-1)'s complement of a number** in any number system with base **r** can be found out by subtracting every single digit of a number by **r-1**.

For Example: In the binary number system, the base is **2**. Hence, its **(r-1)'s** i.e., **(2-1 =1)'s** complement can be obtained by subtracting each bit from 1, i.e., **1's** complement for **001** can also be calculated by subtracting **001** from **111** which will be **(111-001) = $(110)_2$**.

Similarly, in the octal number system, the base is **8** so its **7's** complement can be calculated by subtracting each bit by **7**, i.e., **7's** complement for **347** in octal number system can be calculated by subtracting **347** from **777** which will yield **(777 - 347) = $(430)_8$**.

9's and 10's complement in Decimal Number System

We already know that the decimal number system has its base as 10, As we have already discussed above, 9's complement of decimal number can be found out by subtracting its each by 9.

Example 1: Calculate 9's complement of $(2457)^{10}$

Solution: 9999 - 2457 = $(7542)^{10}$

Now, 10's complement of a decimal number can be calculated by adding 1 to the LSB of the 9's complement.

Example 2: Calculate 10's complement of $(2457)^{10}$

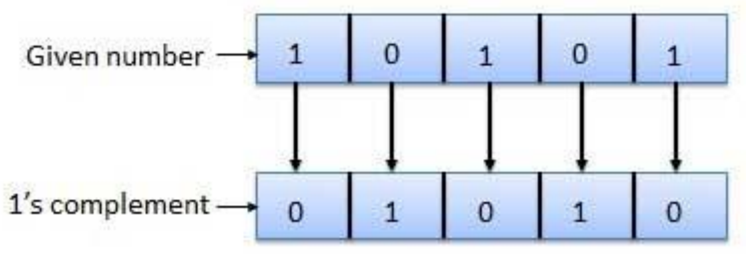
Solution: 10's complement for $(2457)^{10}$ is calculated by adding 1 to $(7542)^{10}$ which is the 9's complement. Therefore, 10's complement of $(2457)^{10}$ is $(7543)^{10}$.

Binary system complements

As the binary system has base r = 2. So the two types of complements for the binary system are 2's complement and 1's complement.

1's complement

The 1's complement of a number is found by changing all 1's to 0's and all 0's to 1's. This is called as taking complement or 1's complement. Example of 1's Complement is as follows.

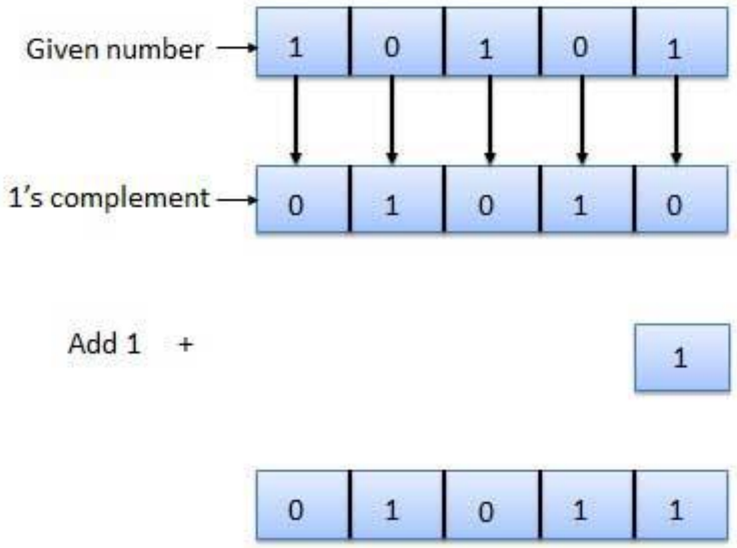


2's complement

The 2's complement of binary number is obtained by adding 1 to the Least Significant Bit (LSB) of 1's complement of the number.

2's complement = 1's complement + 1

Example of 2's Complement is as follows.



Differences between 1's complement and 2's complement

1's complement	2's complement
To get 1's complement of a binary number, simply invert the given number.	To get 2's complement of a binary number, simply invert the given number and add 1 to the least significant bit (LSB) of given result.
1's complement of binary number 110010 is 001101	2's complement of binary number 110010 is 001110
Simple implementation which uses only NOT gates for each input bit.	Uses NOT gate along with full adder for each input bit.
Can be used for signed binary number representation but not suitable as	Can be used for signed binary number representation and most suitable as

1’s complement	2’s complement
ambiguous representation for number 0.	unambiguous representation for all numbers.
0 has two different representation one is -0 (e.g., 1 1111 in five bit register) and second is +0 (e.g., 0 0000 in five bit register).	0 has only one representation for -0 and +0 (e.g., 0 0000 in five bit register). Zero (0) is considered as always positive (sign bit is 0)
For k bits register, positive largest number that can be stored is $(2^{(k-1)}-1)$ and negative lowest number that can be stored is $-(2^{(k-1)}-1)$.	For k bits register, positive largest number that can be stored is $(2^{(k-1)}-1)$ and negative lowest number that can be stored is $-(2^{(k-1)})$.
end-around-carry-bit addition occurs in 1’s complement arithmetic operations. It added to the LSB of result.	*end-around-carry-bit* addition does not occur in 2’s complement arithmetic operations. It is ignored.
1’s complement arithmetic operations are not easier than 2’s complement because of addition of *end-around-carry-bit.*	2’s complement arithmetic operations are much easier than 1’s complement because of there is no addition of *end-around-carry-bit.*
Sign extension is used for converting a signed integer from one size to another.	Sign extension is used for converting a signed integer from one size to another.

Use of 1's complement

1's complement plays an important role in representing the signed binary numbers. The main use of 1's complement is to represent a signed binary number. Apart from this, it is also used to perform various arithmetic operations such as addition and subtraction.

In signed binary number representation, we can represent both positive and negative numbers. For representing the positive numbers, there is nothing to do. But for representing negative numbers, we have to use 1's complement technique. For representing the negative number, we first have to represent it with a positive sign, and then we find the 1's complement of it.

Let's take an example of a positive and negative number and see how these numbers are represented.

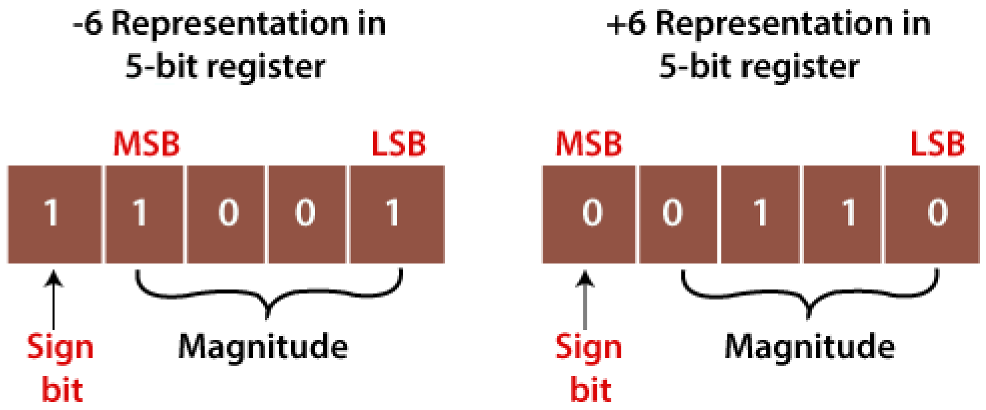
Example 1: +6 and -6

The number +6 is represented as same as the binary number. For representing both numbers, we will take the 5-bit register.

So the +6 is represented in the 5-bit register as 0 0110.

The -6 is represented in the 5-bit register in the following way:

1. +6=0 0110
2. Find the 1's complement of the number 0 0110, i.e., 1 1001. Here, MSB denotes that a number is a negative number.



Here, MSB refers to Most Significant Bit, and LSB denotes the Least Significant Bit.

Example 2: +120 and -120

The number +120 is represented as same as the binary number. For representing both numbers, take the 8-bit register.

So the +120 is represented in the 8-bit register as 0 1111000

.

The -120 is represented in the 8-bit register in the following way:

1. +120=0 1111000

Now, find the 1's complement of the number 0 1111000, i.e., 1 0000111. Here, the MSB denotes the number is the negative number.

Use of 2's complement

2's complement is used for representing signed numbers and performing arithmetic operations such as subtraction, addition, etc. The positive number is simply represented as a magnitude form. So there is nothing to do for representing positive numbers. But if we represent the negative number, then we have to choose either 1's complement or 2's complement technique. The 1's complement is an ambiguous technique, and 2's complement is an unambiguous technique. Let's see an example to understand how we can calculate the 2's complement in signed binary number representation.

Example 1: +6 and -6

The number +6 is represented as same as the binary number. For representing both numbers, take the 5-bit register.

So the +6 is represented in the 5-bit register as 0 0110.

The -6 is represented in the 5-bit register in the following way:

1. +6=0 0110
2. Now, find the 1's complement of the number 0 0110, i.e., 1 1001.
3. Now, add 1 to its LSB. When we add 1 to the LSB of 11001, the newly generated number comes out 11010. Here, the sign bit is one which means the number is the negative number.

Example 2: +120 and -120

The number +120 is represented as same as the binary number. For representing both numbers, take the 8-bit register.

So, the +120 is represented in the 8-bit register as 0 1111000.

The -120 is represented in the 8-bit register in the following way:

1. +120=0 1111000
2. Now, find the 1's complement of the number 0 1111000, i.e., 1 0000111. Here, the MSB denotes the number is the negative number.
3. Now, add 1 to its LSB. When we add 1 to the LSB of 1 0000111, the newly generated number comes out 1 0001000. Here, the sign bit is one, which means the number is the negative number.