NED University of Engineering & Technology

Department of Software Engineering

Course Coordinator : Ms.Simra Najm
Designation : ***Assistant Professor***
Email: ***simranajm@neduet.edu.pk***

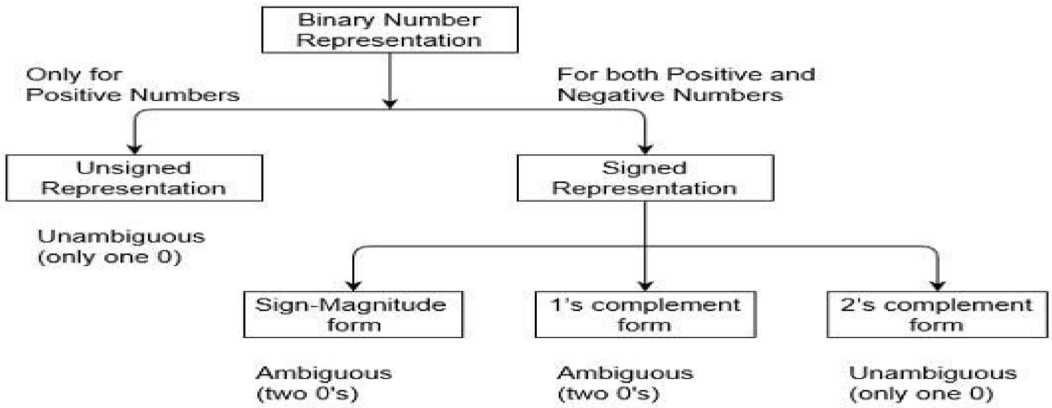
Lec # 4

DATA REPRESENTATION

- ✔ Variables such as integers can be represent in two ways, i.e., Signed and Unsigned.
- ✔ Signed numbers use sign flag or can be distinguish between negative values and positive values.
- ✔ Whereas Unsigned numbers stored only positive numbers but not negative numbers.
- ✔ Number representation techniques like: Binary, Octal, Decimal and Hexadecimal number representation techniques can represent numbers in both signed and unsigned ways.
- ✔ Binary Number System is one the type of Number Representation techniques.
- ✔ It is most popular and used in digital systems.
- ✔ Binary system is used for representing binary quantities which can be represented by any device that has only two operating states or possible conditions. For example, a switch has only two states: open or close.
- ✔ In the Binary System, there are only two symbols or possible digit values, i.e., 0 and 1. Represented by any device that only 2 operating states or possible conditions..

Representation of Binary Numbers:

- Binary numbers can be represented in signed and unsigned way.
- Unsigned binary numbers do not have sign bit (only represent positive numbers)
- Whereas **Signed** binary numbers uses signed bit as well or these can be distinguishable between positive and negative numbers. A Signed binary is a specific data type of a signed variable.



SIGNED AND UNSIGNED NUMBERS

- Unsigned binary numbers are, by definition, positive numbers and thus do not require an arithmetic sign.
- An n-bit Unsigned number represents all numbers in the range **0 to $2^n - 1$.**

For example,

The range of 8-bit unsigned binary numbers is from 0 to 255_{10} in decimal and from 00 to FF_{16} in hexadecimal. Similarly, the range of 16-bit unsigned binary numbers is from 0 to $65,535_{10}$ in decimal and from 0000 to $FFFF_{16}$ in hexadecimal.

- Signed numbers, on the other hand, require an arithmetic sign. The most significant bit of a binary number is used to represent the sign bit.
- If the sign bit is equal to zero, the signed binary number is positive; otherwise, it is negative.
- The remaining bits represent the actual number. There are three ways to represent negative numbers.

1. Unsigned Numbers:

- Unsigned numbers don't have any sign,
- Since there is no sign bit in this unsigned binary number, so
- Representation of unsigned binary numbers are all positive numbers only.
- N bit binary number represent its magnitude only.
- Zero (0) is also unsigned number.
- This representation has only one zero (0), which is always positive.
- Every number in unsigned number representation has only one unique binary equivalent form, so this is unambiguous representation technique.
- Representation of positive decimal numbers is positive by default.

- **We always assume that there is a positive sign symbol in front of every number.**

The range of unsigned binary number is from 0 to (2^n-1).

Example-1: Represent decimal number 92 in unsigned binary number.
Simply convert it into Binary number, it contains only magnitude of the given number.
$= (92)_{10}$
$= (1x2^6+0x2^5+1x2^4+1x2^3+1x2^2+0x2^1+0x2^0)_{10}$
$= (1011100)_2$
It's 7 bit binary magnitude of the decimal number 92. (**We always assume that there is a positive sign symbol in front of every number.)**

Example-2: Find range of **5 bit** Unsigned binary numbers. Also, find **minimum** and **maximum** value in this range. Since, range of unsigned binary number is from 0 to (2^n-1).

Therefore, range of 5 bit unsigned binary number is *from* 0 to (2^5-1) which is equal from **minimum value 0 (00000) to**

maximum value 31 (11111).

2. Signed Numbers:

- Signed numbers contain sign flag, this representation distinguish positive and negative numbers.
- This technique contains both sign bit and magnitude of a number.
- For example, in representation of negative decimal numbers, we need to put negative symbol in front of given decimal number.

Representation of Signed Binary Numbers:

- There are three types of representations for signed binary numbers.
- Because of extra signed bit,
- Binary number zero has two representation, either positive (0) or negative (1), so ambiguous representation.
- But 2's complementation representation is unambiguous representation because of there is no double representation of number 0.
- These are**:** **Sign-Magnitude form**, **1's complement form**, and **2's complement form** which are explained as following below.

2.(a) *Sign-Magnitude form:*

- For n bit binary number, 1 bit is reserved for sign symbol.
- If the value of sign bit is 0, then the given number will be positive, else
- If the value of sign bit is 1, then the given number will be negative.
- Remaining (n-1) bits represent magnitude of the number. Since magnitude of number zero (0) is always 0, so there can be two representation of number zero (0), positive (+0) and negative (-0), which depends on value of sign bit.
- Hence these representations are ambiguous generally because of two representation of number zero (0).
- Generally sign bit is a most significant bit (MSB) of representation.

The range of Sign-Magnitude form is from $(2^{(n-1)}-1)$ to $(2^{(n-1)}-1)$.

For example,

- Range of 6 bit Sign-Magnitude form binary number is from (2^5-1) to (2^5-1) which is equal from minimum value -31 (i.e., 1 11111) to
- maximum value +31 (i.e., 0 11111).
- And zero (0) has two representation, -0 (i.e., 1 00000) and +0 (i.e., 0 00000).

2.(b) 1's Complement form:

- Since, 1's complement of a number is obtained by inverting each bit of given number. So,
- we represent positive numbers in binary form and negative numbers in 1's complement form. There is extra bit for sign representation.
- If value of sign bit is 0, then number is positive and you can directly represent it in simple binary form, but
- if value of sign bit 1, then number is negative and you have to take 1's complement of given binary number.
- You can get negative number by 1's complement of a positive number and positive number by using 1's complement of a negative number.
- Therefore, in this representation, zero (0) can have two representation, that's why 1's complement form is also ambiguous form.

The range of 1's complement form is *from* $-(2^{(n-1)}-1)$ to $(2^{(n-1)}-1)$.

For example, range of 6 bit 1's complement form binary number is from (2^5-1) to (2^5-1) which is equal from minimum value -31 (i.e., 1 00000) to maximum value +31 (i.e., 0 11111). And zero (0) has two representation, -0 (i.e., 1 11111) and +0 (i.e., 0 00000).

2.(c) 2's complement form:

- Since, 2's complement of a number is obtained by inverting each bit of given number plus 1 to least significant bit (LSB). So,

- we represent positive numbers in binary form and negative numbers in 2’s complement form. There is extra bit for sign representation.
- If value of sign bit is 0, then number is positive and you can directly represent it in simple binary form, but
- if value of sign bit 1, then number is negative and you have to take 2’s complement of given binary number.
- You can get negative number by 2’s complement of a positive number and positive number by directly using simple binary representation.
- If value of most significant bit (MSB) is 1, then take 2’s complement from, else not.
- Therefore, in this representation, zero (0) has only one (unique) representation which is always positive.

The range of 2’s complement form is *from* $(2^{(n-1)})$ to $(2^{(n-1)}-1)$.

For example, range of 6 bit 2’s complement form binary number is from (2^5) to (2^5-1) which is equal from minimum value -32 (i.e., 1 00000) to maximum value +31 (i.e., 0 11111). And zero (0) has two representation, -0 (i.e., 1 11111) and +0 (i.e., 0 00000).

Subtraction in 1_s and 2_s Complement

To perform a binary subtraction you first have to represent the number to be subtracted in its negative form. This is known as its two's OR one’s complement.
The two's complement of a binary number is obtained by:

1. Replacing all the 1_s with 0_s and the 0_s with 1_s. This is known as its one's complement.
2. Adding 1 to this number by the rules of binary addition.

Now you have the two's complement.

Subtracting using 1_s complement

1s complement method is as follows:

1. Determine the 1_s complement of the number to be subtracted.
2. Perform the addition i.e. Add the 1_s complement to the number .

3(a). IF there is a carry out, it means answer is positive. Remove the final carry and add it to the result. This is called the end-around carry.
(b). If there is no carry. It means answer is in negative and in 1’s compliment form

Example 1:
11001-10011

Result from Step1: 01100
Result from Step2: 1 00101{Carry out}

Result from Step3: 00110

To verify, note that 25 - 19 = 6

Example 2 :

1001 - 1101

Result from Step1: 0010

Result from Step2: 1011 {No Carry}

Result from Step3: 1011(-4)

To verify, note that 9 - 13 = - 4

Subtracting using 2s complement

2s complement method is as follows:

1. Determine the 2s complement of the number to be subtracted.
2. Perform the addition i.e. Add the 2s complement.

3(a).If there is a carry then result is positive Discard the final carry (there is always one in this case)

(b).If there is no carry from the left-most column. The result is in 2s complement form and is negative. Change the sign and take the 2s complement of the result to get the final answer

.

Example 1:

11001 - 10011

Result from Step1: 01101
Result from Step2: 100110
Result from Step3: 00110
Again, to verify, note that 25 - 19 = 6

Example 2:

1001 - 1101

Result from Step1: 0011
Result from Step2: 1100
Result from Step3: **-0100 or (1100)**

Again to verify, note that 9 - 13 = - 4