REPRESENTING NEGATIVE NUMBERS

Presented by Nabanita Das

BENGAL INSTITUTE OF TECHNOLOGY

Representing Negative Numbers

- Are the negative numbers just numbers with a minus sign in the front? This is probably true…but there are issues to represent negative numbers in computing systems.
- Common schemas:
  - Sign-magnitude
  - Complementary representations:
    - 1's complement
    - ***2's complement – most common & important***

Representation Of Negative Number

1'S & 2'S COMPLIMENT → Binary Number System

9'S & 10'S COMPLIMENT → Decimal Number System

Sign Magnitude

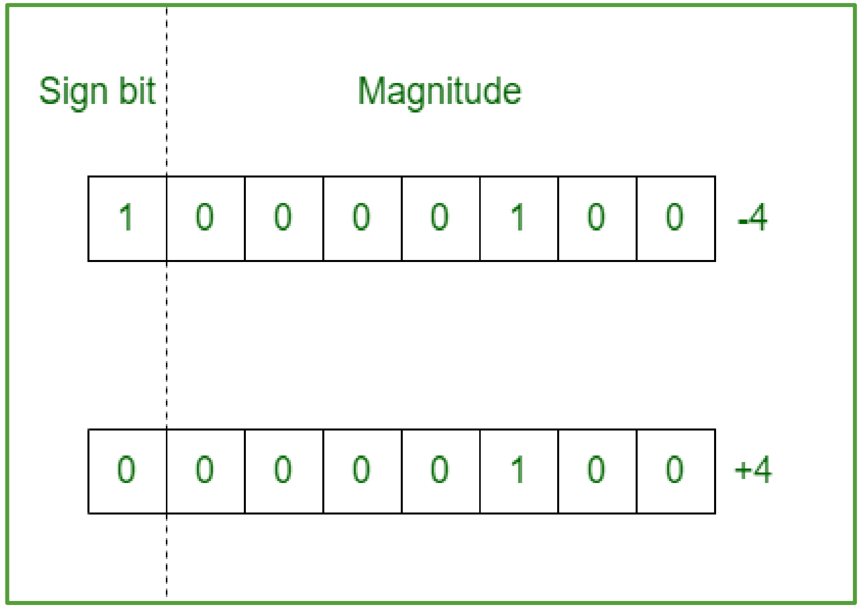
- An extra bit in the most significant position is designated as the ***sign*** *bit* which is to the left of an unsigned integer. The unsigned integer is the ***magnitude***.

x	xxx xxxx

- A **0** in the sign bit means positive, and a **1** means negative.
- There is a problem with the sign-magnitude representation is a ***plus zero*** and ***minus zero***, which causes unnecessary complexity.

Sign Magnitude

- In the signed magnitude method number is divided into two parts: Sign bit and magnitude. Sign bit is 1 for negative number and 0 for positive number. Magnitude of number is represented with the binary form of the number.
- Left most bit used to represent sign
  - **0 = positive value**
  - **1 = negative value**
- Given an n+1-bit sign magnitude number the range of values that it can represent is $-(2^n-1)$ to $+(2^n-1)$



1's Complement

- For positive numbers, the representation is the same as for unsigned integers where the most significant bit is always zero.
- The additive inverse of a one's complement representation is found by inverting each bit.
- Inverting each bit is also called taking the one's complement.
- **Shortcut for base 2?**
  - **All combinations used, but 2 zeros!**

	$+N$	$-N$
0	0000	1111
1	0001	1110
2	0010	1101
3	0011	1100
4	0100	1011
5	0101	1010
6	0110	1001
7	0111	1000

2'S Complement Process

The steps in the **2's Complement** process uses base-2 (binary) numbers.

First, complement all of the digits in a number.

- A digit's complement is the number you add to the digit to make it equal to the largest digit in the base (i.e., 1 for binary). In binary language, the complement of 0 is 1, and the complement of 1 is 0.

Second, add 1.

- Without this step, our number system would have two zeroes (+0 & -0), which no number system has.

2's Complement

- If we consider x is a positive number.
- Then –x is represented **by $b^D+(-x)$**.
  - Ex. Let b=2 and D=4. Then -1 is represented by $2^4-1 = 15$ or 1111_2.
  - Ex. Let b=2 and D=4. Then -5 is represented by $2^4 - 5 = 11$ or 1011_2.

Solution for the two ZERO's

- 0000 $\longrightarrow$ 1's complement $\longrightarrow$ 1111 $\longrightarrow$ adding 1 $\longrightarrow$ 10000 discharge carry.

	$+N$	$-N$
0	0000	0000
1	0001	1111
2	0010	1110
3	0011	1101
4	0100	1100
5	0101	1011
6	0110	1010
7	0111	1001

2's Complement

First, find the equivalent binary number.
If the decimal number was positive: then ok.
If the decimal number was negative: invert all the bits, and add 1 (with carries as needed).

Examples

25 decimal is 00011001 binary. It's positive, so all done.
-25 decimal:

Begin with binary:	**0 0 0 1 1 0 0 1**
Invert all the bits to get:	**1 1 1 0 0 1 1 0**
	+ 1
Add 1 to get:	**1 1 1 0 0 1 1 1** (decimal value -25)

9's & 10's Complement Process

- Similarly **2's complement**, the **10's Complement** process uses base-10 (decimal) numbers. In decimal number system we have the **9's** and **10's** complement, the **9's** is obtained by **subtracting** each digit **from 9**.

- 10's complement = 9's complement +1.

First, complement all of the digits in a number.

- A digit's complement is the number you add to the digit to make it equal to the largest digit in the base (i.e., 9 for decimal). The complement of 0 is 9, 1 is 8, 2 is 7, etc.

Second, add 1.

- Without this step, our number system would have two zeroes (+0 & -0), which no number system has.

- If we consider x is a positive number.

- Then –x is represented **by b^D+(-x)**.
  - Then the 10's complement of 12389 = 10^5 – 12389
    = 100000 – 12389 = 87611.

10’s Complement Examples

Example #1

```
-003
↓↓↓
 996
  +1
____
 997
```

Complement Digits

Add 1

Example #2

```
-214
↓↓↓
 785
  +1
____
 786
```

Complement Digits

Add 1

Binary Arithmetic

1. BINARY ADDITION
2. BINARY SUBTRACTION
3. BINARY MULTIPLICATION
4. BINARY DIVISION

Binary Addition

0 + 0 = 0
1 + 0 = 1
0 + 1 = 1
1 + 1 = 0 1 (Carry bit)

1 1 0 1	(13 decimal)
+0 0 0 1	(+1 decimal)
1 1 1 0	(14 decimal)

Binary Subtraction

0 - 0 = 0
1 - 0 = 1
0 - 1 = 1 1 (Borrow bit)
1 - 1 = 0

Borrow

1 1 0 1	(13 decimal)
-0 0 1 1	(3 decimal)
1 0 1 0	(10 decimal)

Binary Multiplication

```
             1 0 0 0        =8₁₀
          X  0 1 1 0        =6₁₀
_____________________
             0 0 0 0
         + 1 0 0 0
       + 1 0 0 0
     + 0 0 0 0
_____________________
0 1 1 0 0 0 0               = 48₁₀
```

Binary Division

```
011 ) 0 1 1 0 0 1 0    ( 1
      0 1 1            (0
        0 0 0
        0 0 0          (0
          0 0 0
          0 0 0
            0 0 1      (0
            0 0 0
              0 1 0
```

Q=1000=16$_{10}$

R=10= 2$_{10}$

Using The 2's Compliment Process

Use the 2's complement process to add together the following numbers.

POS + POS = POS	⇒	9 + 5 = 14	
NEG + POS = NEG	⇒	(-9) + 5 = - 4	
POS + NEG = POS	⇒	9 + (-5) = 4	
NEG + NEG = NEG	⇒	(-9) + (-5) = - 14	

POS + POS → POS Answer

If no 2's complement is needed, use regular binary addition.

$$
\begin{array}{rcr}
9 & \longrightarrow & 00001001 \\
+\quad 5 & \longrightarrow & +\ 00000101 \\
\hline
14 & \longleftarrow & 00001110
\end{array}
$$

POS + NEG → POS Answer

Take the 2's complement of the negative number and use regular binary addition.

$$\begin{array}{rcr} 9 & \longrightarrow & 00001001 \\ +\ (-5) & & +\ 11111011 \\ \hline 4 & \longleftarrow & \cancel{1}]\,00000100 \end{array}$$

8th Bit = 0: Answer is Positive
Disregard 9th Bit

$$\begin{array}{r} 00000101 \\ \downarrow\downarrow\downarrow\downarrow\downarrow\downarrow\downarrow\downarrow \\ 11111010 \\ +1 \\ \hline 11111011 \end{array}$$

2's Complement Process

POS + NEG $\longrightarrow$ NEG Answer

Take the 2's complement of the negative number and use regular binary addition.

$$\begin{array}{r} (-9) \\ +\quad 5 \\ \hline -\boxed{4} \end{array} \qquad \begin{array}{r} 11110111 \\ +\ 00000101 \\ \hline 11111100 \end{array}$$

8^{th} Bit = 1: Answer is Negative

To Check: Perform 2's Complement On Answer

$$\begin{array}{r} 11111100 \\ \downarrow\downarrow\downarrow\downarrow\downarrow\downarrow\downarrow\downarrow \\ 00000011 \\ +1 \\ \hline 00000100 \end{array}$$

2's Complement Process

$$\boxed{\begin{array}{r} 00001001 \\ \downarrow\downarrow\downarrow\downarrow\downarrow\downarrow\downarrow\downarrow \\ 11110110 \\ +1 \\ \hline 11110111 \end{array}}$$

NEG + NEG → NEG Answer

Take the 2's complement of both negative numbers and use regular binary addition.

(-9) → 11110111

\+ (-5) → + 11111011

-14 ← ~~1~~]11110010

2's Complement Numbers, See Conversion Process In Previous Slides

8th Bit = 1: Answer is Negative
Disregard 9th Bit

To Check: Perform 2's Complement On Answer

11110010

↓↓↓↓↓↓↓↓

00001101

+1

00001110