2`s Complement Arithmetic

Sparsh Mittal

How To Create A Negative Number

- In digital electronics you cannot simply put a minus sign in front of a number to make it negative.
- You must represent a negative number in a *fixed-length* binary number system. All signed arithmetic must be performed in a *fixed-length* number system.
- A physical *fixed-length* device (usually memory) contains a fixed number of bits (usually 4-bits, 8-bits, 16-bits) to hold the number.

8-Bit Binary Number System

How do you represent negative numbers in this 8-bit binary system?

→Cut the range in half.

→Use 00000001 – 01111111 to indicate positive numbers.

→Use 10000000 – 11111111 to indicate negative numbers.

→Notice that 00000000 is not positive or negative.

+127	01111111	pos(+)
+126	01111110	
+125	01111101	
⋮	⋮	
+1	00000001	
0	00000000	
-1	11111111	
-2	11111110	
⋮	⋮	
-127	10000001	
-128	10000000	neg(-)

Sign Bit

- What did do you notice about the most significant bit of the binary numbers?
- The MSB is (0) for all positive numbers.
- The MSB is (1) for all negative numbers.
- The MSB is called the sign bit.
- In a signed number system, this allows you to instantly determine whether a number is positive or negative.

pos(+)

+127	01111111
+126	01111110
+125	01111101
⋮	⋮
+1	00000001
0	00000000
-1	11111111
-2	11111110
⋮	⋮
-127	10000001
-128	10000000

neg(-)

2'S Complement Process

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 Examples

Example #1

```
 5 = 00000101
     ↓↓↓↓↓↓↓↓
     11111010
           +1
     --------
-5 = 11111011
```

Complement Digits

Add 1

Example #2

```
-13 = 11110011
      ↓↓↓↓↓↓↓↓
      00001100
            +1
      --------
 13 = 00001101
```

Complement Digits

Add 1

Solved example

Compute 2’s complement of 0000 0100 (binary), which is 4 (decimal).

Answer: First compute 1’s complement, which is 1111 1011.

Now add 1, so we get 1111 1100. In hexadecimal, it is FC.

Understanding 2’s complement

- Consider a 3-bit number 6, which is 110.
- On taking 2’s complement, we get 010, which is simply 2. We should have got -6.
- Where is the error?
- **Explanation:** The original number 110 itself was not 6 actually! In 2’s complement, it is actually -2.
- With 3 bits, we can represent only -4 to 3. The number -6 is outside the range for 3 bits, so overflow happens.
- We have to use 4-bit number system.
- Now, 6 is 0110 and its 2’s complement would be 1010.

Property of 2's complement

- The 2's complement of 2's complement of a number is the number itself. Just like (-(-2)) = 2
- With say 12 bits, you can represent integers from -2^{11} to $2^{11}-1$.