

Computer Arithmetic I

Professor: Yang Peng

The slides are re-produced by the courtesy of
Dr. Arnie Berger and Dr. Wooyoung Kim

Topic

Computer Arithmetic

- Negative binary number
- Chapter 2.4, 2.5 (Null)

Negative Integer in Binary

- Subtraction in a processor is done by **changing positive numbers to negative number** and then **adding** them
- A processor **always** assumes “**an addition is on signed numbers**”

How to represent a negative integer in a computer system?

Negative Integer – Two's Complement

- Define the **negative number** as a **complement number**
- 10's complement of 1718 in a 4-decimal system
 $10^4 - 1718 = 10000 - 1718 = 8282.$
 8282 is 10's complement of 1718 in a 4-decimal system
- 2's complement of 7 in an 8-bit system
 $2^8 - 7 = 10000000_2 - 00000111_2 = 11111001_2$
 Define 11111001_2 as a negative integer of 7 (00000111_2)
- Subtraction is confusing? Then,
 $2^8 - 7 = (2^8 - 1) - 7 + 1$
 $= 11111111_2 - 111_2 + 1_2 = 11111000_2 + 1_2 = 11111001_2$
- It's the same as **"flip bits** (1's to zero, zero's to 1) **and add 1 at the end"**

Negative Integer – Two's Complement

- How to get the **2's complement of K** in **n-bit system**
 - $2^n - K$, or
 - $(2^n - 1) - K + 1$, or
 - **Flip** (Complement) all bits of K, **then add 1**
- Example: In an 8-bit system, compute the 2's complement of 0x5E
 - Step 1: Convert to binary: $0x5E = 0101\ 1110$
 - Step 2: Flip bits $\rightarrow 1010\ 0001$
 - Step 3: Add 1 $\rightarrow = 1010\ 0001 + 1 = 1010\ 0010$
 - Step 4: Convert to hex again: $1010\ 0010_2 \rightarrow 0xA2$

Signed Number Range

- Signed number in a computer system
 - A negative number is in the format of 2's complement
- Two's complement negative numbers imply
 - All arithmetic operations are converted to addition
 - The **MSB** is *always* 1 if it is a **negative number** (zero is positive)
 - Range of an n-bit number is -2^{n-1} to $+(2^{n-1}-1)$
 - E.g., range of 4-bit numbers is -2^3 to $+(2^3-1) \rightarrow -8$ to 7
- Exercise: in a 4-bit system
 - What is the signed number 1000_2 in decimal?
 - The MSB is 1, so it is a negative number
 - 2's complement of 1000_2 is $0111_2 + 1 = 1000_2 = 8$
 - So it is -8

We cannot have a positive 8 in 4-bit signed system.

Similarly, we cannot have a positive 128 in 8-bit signed system.

Repeat the question with 2's complement

- Question (in a 4-bit system)
 1. How to represent zero? 0000 or 1000 ?
0000 is zero, and 1000 is -8
 2. How many unsigned numbers you can have in a 4-bit system?
unsigned: 0000 to 1111 (0 to 15)
 3. How many signed numbers (in 2's complement) you can have in a 4-bit system?
Positive: 0000 to 0111 (0 to 7)
Negative: 1000 to 1111 (-8 to -1): So, total 16 numbers
 4. With this representation in a 4-bit system,
 - 1) What is $7 - 5$ in binary?
 $0111_2 - 0101_2 = 0010_2$
 - 2) What is $7 + (-5)$ in binary?
 $0111_2 + 1011_2 = 1\ 0010_2$
(Since it is a 4-bit system, the carry bit won't appear, so $0010_2 = 2_{10}$)

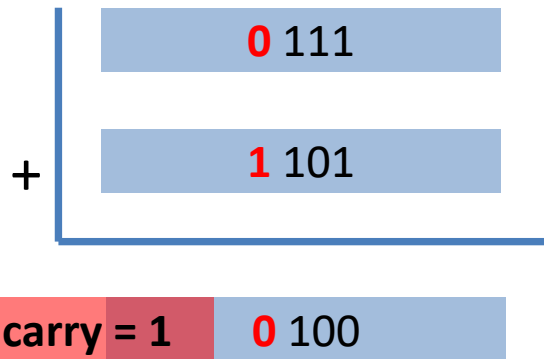
Two's Complement Arithmetic

- Arithmetic **overflow**
 - Adding **two positive** number **results in a negative** number
 - Adding **two negative** number **results in a positive** number
 - *How can this be possible?*
- If the result is **out of range**, the computer results in an **incorrect answer**.
- For example, in a 4-bit system (range is -8 to 7)
 - $7 + 7 = 0111 + 0111 = 1110$ (It is not 14 but -2) → incorrect, negative
 - $(-7) + (-8) = 1001 + 1000 = 10001$ (it has a carry bit and the result is 1) → incorrect, positive

Two's Complement Arithmetic

- In a 4-bit system (range -8 to 7)

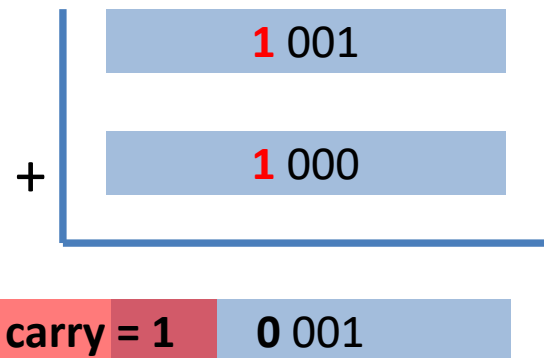
1. $7 + (-3) = 4$



No overflow

carry-out bit is invisible.
Then the answer is 4 (correct)

2. $(-7) + (-8)$



Overflow (sign bit changed to 0)

Incorrect result, error.