Computer Systems
Lecture 17

Overview

- Addition
- Subtraction
- Overflow testing
- Two’s complement
- Numerical types in Java

10’s complement representation

- For the 10’s complementary representation we have to specify the number of digits (some *n*).
- The number being represented depends on that *n*. For example:
  - 3-digit complement: ‘567’ represents -433.
  - 4-digit complement: ‘0567’ represents 567.

Addition

- Operation of addition for *n*-digit numbers represented by 10's complementary convention:
  - Addition modulo $1\underbrace{0\ldots0}_{n}$.
  - Use existing addition procedure, except that any carry beyond the specified number of digits (*n*) is "thrown away".

Examples of addition in 3 digits

10's complementary codes	Actual values
699	- 301
400	400
-------	------
1099	99
099	

Examples of addition (cont.)

10's complementary codes	Actual values
814	- 186
923	- 77
-------	-------
1737	- 263
737	

Subtraction

- To subtract *B* from *A* one may first compute negated *B* and add it to *A*.
  - *A* - *B* = *A* + (-*B*).
- Given *B* in 10's complement, how to compute -*B*:
  - Just subtract *B* from 10…0.
  - For example, negated value of '250' is '750' (= -250).

Overflow testing

- Example.
  - Codes: ‘347’ + ‘230’ = ‘577’.
  - Values: 347 + 230 = - 423.
- What is wrong here?
- It is **overflow**.
  - The result 577 is too big to be fit into 3-digit complement representation (-500 – 499).

Testing for overflow

Simple rule:

If both inputs to an addition have the same sign, and the output sign is different, overflow has occurred.

Positive and negative

- In 10’s complement system the range is **unevenly** divided between positive and negative integers.
- For example, 4-digit complement system represents all the integer values between -5000 and 4999.

Two's complement in 8 bits

- Two's complement representation for binary (n=8) is similar to 10's complement representation for decimal.

binary codes:

$$10000000 \ldots 11111111 \mid 00000000 \ldots 01111111$$

two's-complement:

$$-128_{10} \quad \ldots \quad -1_{10} \mid 0_{10} \quad \ldots \quad 127_{10}$$

Two’s complement (cont.)

- As in the case of 10’s complementary system we have to specify the number of digits (here: 8 bits).

- Numbers that begin with 0 are considered to be positive representing **themselves**.
- Numbers that begin with 1 are representing **negative** numbers.

Two's complement (cont.)

- A complement of the number in 2's complementary *n*-digit encoding can be found in one of two ways.
  - Either subtract the value from the modulus:

$$1\underbrace{0\ldots0}_{n}$$

  - Or, invert all bits and add 1 to the result.
- Examples:
  - 11011101 represents -00100011

Example

- Consider 2's complementary code 11011101.
  - What is the binary number represented?
- The code started with 1, it follows a **negative** number is represented.
- To find the number:
  - We first invert all bits of code: 11011101 → 00100010.
  - Then add 1, 00100010 + 1 = 00100011.
- The binary number represented is -00100011.

Addition

- Addition modulo $1\underbrace{0\ldots0}_{n}$

- Particularly simple for computers:
  – "Don't care about the leftmost carry bit".
- Example:

$$\begin{array}{rr} 011011 & 27 \\ \underline{101100} & \underline{-20} \\ \not1000111 & 7 \end{array}$$

Subtraction and overflow

- Subtraction and overflow are handled as in the case of 10's complementary.
- $A - B = A + (-B)$.

- **If both inputs to an addition have the same sign, and the output sign is different, overflow has occurred.**

Numerical types in Java

- In Java we have the following numerical data types for integers.
  - **`byte`** 8-bit: integers from -2^{8-1} to $2^{8-1}-1$.
  - **`short`** 16-bit: integers from -2^{16-1} to $2^{16-1}-1$.
  - **`int`** 32-bit: integers from -2^{32-1} to $2^{32-1}-1$.
  - **`long`** 64-bit: integers from -2^{64-1} to $2^{64-1}-1$.
- 2's complement representation of signed integers is used.

- Q. Addition for *n*-digit numbers represented by 10’s complementary convention is done based upon addition modulo n. (T or F)

- Q. How is overflow detected during addition?

- Q. Can overflow result from subtractions?

- Q. Under 2’s complement system, numbers that begin with 1 are representing negative numbers. (T or F)

- How many bits are required for a Java short integer?

- Q. How many bits is required for a Java long integer?

- Q. An ‘int’ type under Java encodes integers under this range: -2^{32} --- $2^{32}-1$

Readings

- Wikipedia article for 10's complement:
  - http://en.wikipedia.org/wiki/Method_of_complements
- Wikipedia article for 2's complement:
  - http://en.wikipedia.org/wiki/Two%27s_complement