Lecture 2:
The Magical Base-2

CSE 370, Autumn 2007
Benjamin Ylvisaker

Daily Quiz

• Have you added yourself to the class mailing list?
• Do it by 5:30 this afternoon to get a 4 on today’s daily quiz
• Tell classmates who didn’t make it to class on time at your own discretion

Administrivia

• Office hours
  Monday Ramkumar ??? lab
  Tuesday Josh 1:30-2:30 lab
  Wednesday Benjamin 1:30-2:30 210
  Thursday Benjamin 9:30-10:30 210
  Friday Nikhil 11:30-12:30 lab

Elementary Math Review

• Positional number notation
  \[ 2,104 = 2 \times 10^3 + 1 \times 10^2 + 0 \times 10 + 4 \times 1 = 2 \times 10^3 + 1 \times 10^2 + 0 \times 10 + 4 \times 10^0 \]

• Generalize to arbitrary base \( b \)
  \[ \text{XYZ} = X \times b^2 + Y \times b^1 + Z \times b^0 \]
  where \( X, Y \) and \( Z \) are digits with values in the range \([0..b-1] \)
Bases of Interest

- In 370, we are interested in the following bases:
  - Binary [0,1]
  - Octal [0..7]
  - Decimal [0..9]
  - Hexadecimal [0..9,A..F]
    - A=10, B=11, C=12, D=13, E=14, F=15

Conversion to Decimal

- \[1001101_2\]
  \[= 1 \times 2^6 + 0 \times 2^5 + 0 \times 2^4 + 1 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0\]
  \[= 64 + 0 + 0 + 8 + 4 + 1\]
  \[= 77\]

- \[92A70_{16}\]
  \[= 9 \times 16^4 + 2 \times 16^3 + 10 \times 16^2 + 7 \times 16^1 + 0 \times 16^0\]
  \[= 9 \times 65536 + 2 \times 4096 + 10 \times 256 + 7 \times 16 + 0 \times 1\]
  \[= 589824 + 8192 + 2560 + 112\]
  \[= 600688\]

Arithmetic is the Same in All Bases

- \[1101101_2\]
  \[32175_8\]
  \[= 27AA32_{16}\]
  \[+ 101011_2\]
  \[1622_8\]
  \[+ 92A70_{16}\]
  \[0000000_2\]
  \[34017_8\]
  \[30D4A2_{16}\]

- \[1001101_2\]
  \[32175_8\]
  \[= 27AA32_{16}\]
  \[- 101011_2\]
  \[1622_8\]
  \[- 92A70_{16}\]
  \[0000000_2\]
  \[30353_8\]
  \[1E7FC2_{16}\]

Multiplication, Too

- \[\begin{array}{c}
  1101101_2 \\
  \times 101011_2 \\
  \hline
  0000000_2 \\
  \hline
  \end{array}\]
  \[\begin{array}{c}
  1101101_2 \\
  \times 17_{16} \\
  \hline
  1101101_2 \\
  +A3_{16} \\
  \hline
  EA5_{16} \\
  \end{array}\]
Conversion to Binary by Successive Division

\[ 154_{10} \div 2_{10} = 77_{10} \quad \text{Remainder} 0 \]
\[ 77_{10} \div 2_{10} = 38_{10} \quad \text{Remainder} 1 \]
\[ 38_{10} \div 2_{10} = 19_{10} \quad \text{Remainder} 0 \]
\[ 19_{10} \div 2_{10} = 9_{10} \quad \text{Remainder} 1 \]
\[ 9_{10} \div 2_{10} = 4_{10} \quad \text{Remainder} 1 \]
\[ 4_{10} \div 2_{10} = 2_{10} \quad \text{Remainder} 0 \]
\[ 2_{10} \div 2_{10} = 1_{10} \quad \text{Remainder} 0 \]
\[ 1_{10} \div 2_{10} = 0_{10} \quad \text{Remainder} 1 \]

Read the result “up”

The Trouble with Negative Numbers

\[ 10011010_2 \div 1010_2 = 1111_2 \quad \text{Remainder} 100_2 \]
\[ 1111_2 \div 1010_2 = 1_2 \quad \text{Remainder} 101_2 \]
\[ 1_2 \div 1010_2 = 0_2 \quad \text{Remainder} 1_2 \]

• The symbol “-” for negative can be used in any base, when doing arithmetic by hand
• Computers only have two symbols: 1, 0. No “-”
• Also, computers usually do arithmetic with numbers that are a fixed number of bits “wide” (like, 8, 16, 32, 64)
Sign/Magnitude Representation

- High-order (left-most) bit is the sign. 0 = positive, 1 = negative
- Remaining bits are the magnitude
- With $N$ bits, represent numbers between $-2^{N-1} + 1$ and $2^{N-1} - 1$
- Two representations of 0!

Two's Complement

- High-order (left-most) bit is the sign. 0 = positive, 1 = negative
- Remaining bits are the magnitude (encoded in a funny way)
- With $N$ bits, represent numbers between $-2^{N-1} + 1$ and $2^{N-1} - 1$
- Just one representation of 0!

Sign/Magnitude

- Pro: easy to read and write for humans
- Con: harder to do basic arithmetic correctly with a computer
- Result: rarely used

Negation in 2’s Complement

- Flip the bits and add 1
Addition in 2’s Complement

- 0011 (3) + 0101 (5) = 1000 (-8)
- 0011 (3) + 1011 (-5) = 1110 (-2)

Subtraction is just addition with the second operand negated first.

Later in the Course

- Efficient circuits for implementing arithmetic
- Detecting overflow/underflow
- Changing the width of numbers without changing the number

Fractional Numbers

- We might want to represent non-integral numbers
- Two popular approaches:
  - Fixed-point
  - Floating-point
- Not covered in 370

Thank You for Your Attention

- Lab 1 has changed slightly, I’ll post an update soon (and send a mail to the class mailing list)
- Continue reading the book
- Continue/start homework 1
- Next time: the fundamentals of Boolean logic