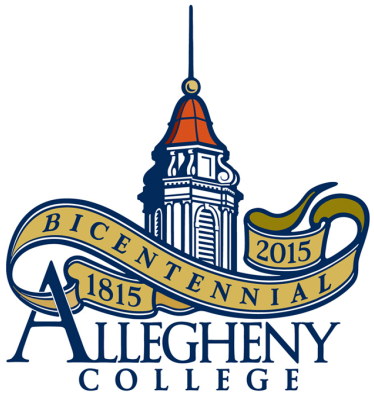
CMPSC 210

Lecture 8: Internal Respresentation Of Data - Part 02

Dr. Aravind Mohan

Allegheny College

September 20, 2017



Last Time

- Binary Numbers
- Decimal Numbers
- HexaDecimal Numbers
- Number Coversions

Reminder

- Don't forget to complete the Mastery Quiz to get today's Attendance points.

Fractions - Converting Decimal to Binary

- Multiply repeatedly by 2 and subtract the whole numbers until the multiplicand = 0.
- For example, 0.6875_{10}

  - $0.6885 \times 2 = 1.375$ Most Significant Bit
  - $0.375 \times 2 = 0.75$
  - $0.75 \times 2 = 1.5$
  - $0.5 \times 2 = 1.0$ Least Significant Bit

  - Solution is $0.6875_{10} = 0.1011_2$

Converting Binary to Decimal

- Starting with the least significant bit, divide the value by 2 and add the next bit. Continue to the binary point.
- For example, 0.01101_2

- $(0 + 1)/2 = 1/2$
- $(1/2 + 0)/2 = 1/4$
- $(1/4 + 1)/2 = 5/8$
- $(5/8 + 1)/2 = 13/16$
- $(13/16 + 0)/2 = 13/32$

- Solution: $0.01101_2 = 13/32$

Binary Artithmetic

- Just like decimal addition, when the sum equals or exceeds the base, we have a carry.
- 00110111

  +01010110

  111 11 carries

  10001101

Note: If the sum exceeds 8 bits, we have a carry out.

Binary Subtraction

- Just like decimal subtraction, when the minuend digit is less than the subtrahend digit, we have a borrow.
- 11001010

  -10011011

  1 1 1 borrows

  00101111

Note: If the subtrahend exceeds the minuend, we have a carry out.

Binary Multiplication

- Easy to compute, since we have just two cases, multiply by 1 or 0.
- $$\begin{array}{r} 01101 \\ \times 01010 \\ \hline 00000 \\ 01101 \\ 00000 \\ 01101 \\ 00000 \\ \hline 0010000010 \end{array}$$

Note: If the product is too large, we have a carry out.

Negative Binary Numbers

- In decimal we are quite familiar with placing a ”-” sign in front of a number to denote that it is negative.
- The same is true for binary numbers a computer won’t understand that.
- What happens in memory then?

Binary Negative Numbers

- There are several representations
  - Signed magnitude
  - One's complement
  - Two's complement
- Two's complement is the system used in microprocessors
- Most significant bit becomes important

Signed Magnitude

- Represent the decimal number as binary.
- Left bit (MSB) used as the sign bit.
- Only have 7 bits to express the number.

12_{10} = 00001100

-12_{10} = 10001100

One's Complement

- Method: Invert the ones and zeros

  11_{10} = 00001011

  -11_{10} = 11110100
- 0 in MSB implies positive
- 1 in MSB implies negative

Two's Complement

- Method: Take the one's complement and add 1

  1110 = 00001011

  -1110 = 11110100 one's comp

  -1110 = 11110101 two's comp

Try out on your own

- Convert $(0.01110010)_2$ to Decimal
- Convert $(0.567821)_{10}$ to Binary
- Calculate the sum of $(100)_2$ and $(101)_2$
- Calculate the difference between $(1000)_2$ and $(10)_2$
- Multiply $(1000)_2$ and $(111)_2$
- Calculate the difference between $(11001001)_2$ and $(10110111)_2$

Any Questions

- Reminder: Dont forget to complete the Review Form at the end to let me know how todays lecture went.
- Midterm exam in two weeks. October 4^{th}.