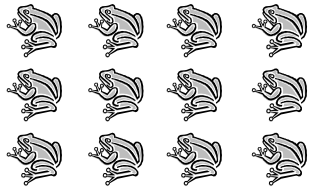
Class 2: Numbering Systems

- Numbering systems
  - Decimal
  - Binary
  - Hexadecimal
- Hexadecimal to Binary Conversion
- Data Organization
- Homework 1 -> due a week from today!

Why different numbering systems?

- When we do arithmetic, we normally use the decimal numbering system (base 10) with digits from 0 to 9.
- Most computer systems operate using binary logic where values are represented using voltage levels (usually 0v and +5 v).
- This corresponds well to the binary numbering system which represents numbers using 0 and 1.
- Working with binary is very cumbersome (the numbers get very long) so other numbering systems that work with powers of two are usually used. The most commonly used is hexadecimal (base 16) with digits from 0 to F.

Different ways of representing the same value



- How many frogs?
  - 12 decimal
  - C hexadecimal
  - 14 octal
  - 1100 binary

Decimal System (a review)

- Commonly used by: people!
- Each digit represents a power of 10
- Digits go from 0 to 9
- Also known as Base 10
- Example:

123_{10} =

If moving to the right of the decimal point, powers decrease:

2.45_{10} =

Binary System

- Used by: computer internal representation.
- In binary, each digit represents a power of 2.
- Binary numbers are made entirely of 0 and 1.
- Also known as Base 2.
- Binary digits are also known as bits.

Binary System, continued

- For convenience, a numeric value can be assigned to each bit position:

  $x_7\ x_6\ x_5\ x_4\ x_3\ x_2\ x_1\ x_0$
- The right most bit is position zero, also known as the low order bit.
- The left most bit known as the high order bit.
- Example:

  0101_2 =

Binary to Decimal Conversion

- Since each digit is a power of two, conversion is done by multiplying the digit by 2 to the power of the bit position.
- Examples:

  101_2 =

  11001010_2 =

Decimal to Binary Conversion: Method 1

- Divide by Decreasing Powers of 2
  - Find the largest power of 2 that fits the number
  - The bit for that power will be set
  - Repeat with the remainder from the division

Example: Converting 76_{10} to Binary

Decimal to Binary Conversion: Method 2

- That method works well by hand but you need to know the powers of two.
- A better method to automate on the computer is to repeatedly divide by 2

-- picture from Irvine, p 575

Hexadecimal System

- Used by: Debug utility.
- Also used as a more compact way to represent numbers rather than binary.
- In hexadecimal (or hex), each digit represents a power of 16.
- Digits are 0 to 9 and A to F.
- Hexadecimal numbers are denoted by the number, followed by an H or h.

Decimal to Hex Conversion

- A good method is to repeatedly divide by 16.
- Example: Converting $15{,}268_{10}$ to hexadecimal

Hex to Decimal Conversion

- Convert each digit to decimal.
- Multiply the digit by 16 to the power of the bit position.
- Example:

  3BA4h =

Converting between Hex and Binary

- This is where the advantage of a hexadecimal representation is clear!
- Each hexadecimal digit represents four binary digits.
- Lookup tables (or, better yet, memorized bit patterns) can be used to do the conversions.
- No multiplication or addition required!

Binary/Hexadecimal Conversion Table

- from AoA

Binary to Hexadecimal

- First, break the number up into four digit (bit) pieces, starting from the least significant (right-most) bit
- Then convert each four digit segment into the corresponding hexadecimal digit
- Example:

1010101110010111100011011100101

Hexadecimal to Binary

- Convert each hex digit into the four digit (bit) binary representation
- Example:

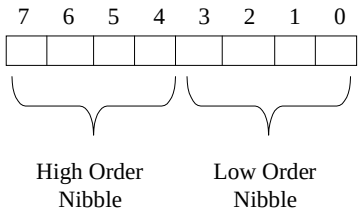
  8A2640

Data Organization

- Computers work with groups of bits (binary digits)
- Common groups of bits are:
  - Single bits
  - Groups of 4 bits – nibbles
  - Groups of 8 bits – bytes
  - Groups of 16 bits – words
  - Groups of 32 bits – double words

Bytes

Bit numbering in a byte:

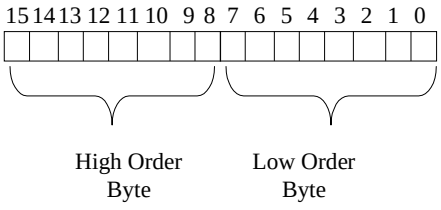


A byte can represent values:
0..255 (unsigned)
-128..+127 (signed)

or, special data types that require no more than 256 different values

Words

Bit numbering in a word:



A word can represent values:
0..65,535 (unsigned)
-32,768..+32,767 (signed)