CS61C: Clarifications & Questions

CS61C Fall2007 - Discussion #7

Greg Gibeling

Unsigned in MIPS

- Overloaded Term
  - It meaning is context dependant
    - What else in this class is like this?
  - A bad design practice (cf C static & extern)
- Meanings
  - Unsigned = Unsigned Integer
    - Multiplication, Division & Comparison
    - These operations must actually interpret bits differently
  - Unsigned = No overflow check
    - Addition, Subtraction
    - These operations result in identical bit patterns whether signed or unsigned
  - Unsigned = No sign extension
    - lbu/sbu
- Unrelated
  - Add/subtract immediate **always** sign extend
  - Logical immediate operations **never** sign extend

Translation & Interpretation

- Examples
  - Compiler: Translates an HLL into an LLL
  - Assembler: Mechanical translation from assembly to machine code
  - Linker
    - A "translator"
    - Truthfully it doesn't translate anything
  - BASH/TCSH/CMD: Shell interpreters
  - CPU: A machine for interpreter implemented in hardware
- Translation
  - Turn code into a different kind of code
- Interpretation
  - Imperative languages: do what the commands say
    - All languages we've see to date are imperative
    - Programs are a list of commands
  - Declarative languages: ?
    - Verilog, the next language we'll learn is declarative
    - "Programs" are statements of existence, not commands

Floating Point Tips

- Stop relying on 10 so much
  - It's a great base to use for the mantissa
  - Don't translate the exponents to be 10 based
  - When comparing representations stick to binary
  - Learn: 0.06125, 0.125, 0.25, 0.5, 1, 2, 4, 8, 16, 32, 64, 128, 256
- Another way of looking at it
  - Mantissa is a binary number
  - Exponent determines where the binary point goes
- Lining up binary points
  - Place the point using the exponent, then line them up
  - Make it clear which bits are stored and which aren't
  - Helps compare representations

Options

- Questions on topics covered so far…
  - Memory & Pointers
    - Pointers (in C and Assembly)
    - Malloc & Free
    - Stack vs. Heap (Text, Static Storage, Object file sections)
    - K&R Allocator
  - Assembly & Machine Code
    - Translation from C
    - Location vs. Address (Memory vs. Pointer)
    - Stored program computing: interpretation of bits
    - Java variables vs. C variables vs. Assembly locations
  - Numbers
    - Integers: Signed, Twos Complement, Ones Complement, Biased
    - Fixed Point
    - Floating Point (Roundoff & Denormals)
  - Compilation
    - Translation & Interpretation
    - Object Files
    - Relocation
    - Dynamic Linking
- Midterm Questions
- Project Solutions
- Review Requests

Midterm Questions

- Design a CS61C Midterm1
  - Cover material through 10/5 only
  - You can write questions, solutions, outlines, whatever you want
  - Groups of no more than 4
  - I'll answer questions on anything class related