Floating-point multiplication
IA-32 and PDP-8 arithmetic
Digital logic basics

CS207, Fall 2004
October 4, 2004

Floating-point multiplication

- Add exponents
  - subtract the bias!
- Multiply mantissas (include leading 1's)
- Normalize
- Check for overflow/underflow
- Round (may need to re-normalize)
- Set sign of product

MIPS floating-point instructions

- 32 floating point registers $fn
  - pairs store double-precision numbers (use even register as name)
- lwcl, swcl
- {add, sub, mul, div}.s, {add, sub, mul, div}.d:
  single- and double-precision ops
- c.{lt, le, gt, ge, eq, ne}.s, c.{lt, le, gt, ge, eq, ne}.d:
  single- and double-precision comparisons
- bclt, bclf: branch on {true, false}
  - PC-relative address

IA-32 floating point

- Multiplication and division operate directly on registers
- Floating-point arithmetic is stack-based
  - operands are pushed onto stack, popped off by operators
  - 80-bit “extended precision” operands
- Includes trig functions, sqrt, abs
- Slower than other instruction set architectures
PDP-8 arithmetic

- Overflow:
  - link bit = 1
  - check sign of accumulator
- Subtraction: 2's complement is CMA IAC
- Example: a - b in PAL:
  CLA CLL
  TAD B
  CMA IAC
  TAD A

PDP-8 arithmetic

- Multiplication and division: repeated +/-
- Double precision:
  - 2 contiguous memory locations
  - operate on “lower” registers first, deal with carry, then operate on “upper” registers
    - carry in link bit; add to accumulator by left-shift
  - 2's complement: invert both halves, CIA lower, CMA upper

Logic paradigms

- Combinational logic: output depends on the immediate inputs
  - arithmetic
- Sequential logic: output depends on previous inputs
  - memory

Boolean algebra

- Branch of arithmetic where inputs and outputs take on one of two values (true or false)
  - corresponds to high and low voltage (sampled)
- Basic operations:
  - AND: true if all inputs are true
  - OR: true if any inputs are true
  - NOT: true if input is false
Boolean arithmetic functions

- Three ways to express:
  - truth table
  - algebraic expression
  - logic diagram

Boolean algebraic expressions

- AND: *
  - “product”
- OR: +
  - “sum”
- NOT: line over the top of the variable

Universal gates

- So named because all boolean functions can be expressed using all of one or the other types of gates
- NAND: NOT AND
  - true except if both inputs are true
- NOR: NOT OR
  - true only if neither input is true