COMP 2401/2001 – day 4

Today's agenda

- Bit models
  - integers
  - approximate reals
- Memory model
  - variables, memory addresses, pointers
- File I/O
  - printf and scanf

Bit models – Integral types

- unsigned integers
  - magnitude-only (base-2 representation)
- signed integers
  - sign-magnitude
  - one's complement
  - two's complement

magnitude-only (unsigned)

- base-2 representaion of number
- can represent all integers in the range
  - $0 \rightarrow 2^n - 1$
  - 2^n numbers in total

sign-magnitude (signed)

- leftmost bit (msb) is for the sign
  - msb = 0 → non-negative number
  - msb = 1 → non-positive number
- all other bits are for magnitude of number
  - same as sign-magnitude for unsigned integers
  - magnitude ranges from $0 \rightarrow 2^{(n-1)} - 1$
- IBM 7090/7094 used by NASA in the 60's
  - Mercury/Gemini space flights, Apollo missions

sign-magnitude (signed)

- a binary number is negated by flipping its msb
  - if $M = b_7 b_6 b_5 b_4 b_3 b_2 b_1 b_0$

    then $-M = \overline{b}_7 b_6 b_5 b_4 b_3 b_2 b_1 b_0$

sign-magnitude (signed)

- a binary number is negated by flipping its msb
  - if $\quad M = b_7 b_6 b_5 b_4 b_3 b_2 b_1 b_0$

    then $\quad -M = \overline{b}_7 b_6 b_5 b_4 b_3 b_2 b_1 b_0$
- 00000000 and 1000000 both represent zero

  (this is not a good property to have)

one's complement (signed)

- a binary number is negated by taking its "complement" (flipping all the bits)

if $M \rightarrow b_7 b_6 b_5 b_4 b_3 b_2 b_1 b_0$

then $-M \rightarrow \overline{b}_7 \overline{b}_6 \overline{b}_5 \overline{b}_4 \overline{b}_3 \overline{b}_2 \overline{b}_1 \overline{b}_0$

- when msb = 0
  - same as magnitude-only (non-negative number)

one's complement (signed)

- when msb = 1
  - flip all the bits to obtain the (magnitude-only) magnitude of the number
- can represent $-(2^{(n-1)}-1) \rightarrow (2^{(n-1)}-1)$
- 00000000 and 11111111 both represent zero
- Univac 1100 series

two's complement (signed)

- a binary number is negated by flipping all of its bits and then adding 1
- if $M = b_7 b_6 b_5 b_4 b_3 b_2 b_1 b_0$

  then $-M = \overline{b_7}\,\overline{b_6}\,\overline{b_5}\,\overline{b_4}\,\overline{b_3}\,\overline{b_2}\,\overline{b_1}\,\overline{b_0} + 00000001$

  (this is equivalent to subtracting the number from 2^n)

two's complement (signed)

- pretty much everything uses two's complement now
- why?
  - only one representation for zero
  - uses the same hardware for +/-/* as unsigned integers
  - easy detection for overflow (correct computation)

two's complement (signed)

- only one zero (0000...0000)
  - what does inverting zero and adding 1 give?
- number range is not symmetric about zero now
  - can represent all numbers

$$-2^{(n-1)} \rightarrow 2^{(n-1)}-1$$

  - n=8 (1 byte) : -128 → 127

Bit models – approximate reals

- we can only approximate most real numbers on a computer
  - how do you represent π or even 0.3 exactly?

- fixed point bit model
- floating point bit model

fixed point (reals)

- extend our notion of base-2 numbers to include negetive powers of 2
  - $2^0 = 1$
    $2^{-1} = 0.5$
    $2^{-2} = 0.25$
    $2^{-3} = 0.125$
    $2^{-4} = 0.0625$
    $2^{-5} = 0.03125$
    $2^{-6} = 0.015625$
    $2^{-7} = 0.0078125$

fixed point (reals)

- extend our notion of base-2 numbers to include negetive powers of 2
  - $2^{0} = 1$

    $2^{-1} = 0.5$

    $2^{-2} = 0.25$

    $2^{-3} = 0.125$

    $2^{-4} = 0.0625$

    $2^{-5} = 0.03125$

    $2^{-6} = 0.015625$

    $2^{-7} = 0.0078125$

$b_0.b_{-1}b_{-2}b_{-3}b_{-4}b_{-5}b_{-6}b_{-7}$

what numbers can we represent with this?

fixed point (reals)

- $b_0.b_{-1}b_{-2}b_{-3}b_{-4}b_{-5}b_{-6}b_{-7}$

  $\longrightarrow$

  $b_0*2^0 + b_{-1}*2^{-1} + b_{-2}*2^{-2} + b_{-3}*2^{-3} + b_{-4}*2^{-4}$
  $+ b_{-5}*2^{-5} + b_{-6}*2^{-6} + b_{-7} *2^{-7}$

  $\longrightarrow$

  $b_0 + 0.5\ b_{-1} + 0.25\ b_{-2} + 0.125\ b_{-} + 0.0625\ b_{-4}$
  $+ 0.03125\ b_{-5} + 0.015625\ b_{-6} + 0.0078125\ b_{-7}$

fixed point (reals)

- $b_0.b_{-1}b_{-2}b_{-3}b_{-4}b_{-5}b_{-6}b_{-7}$

  →

  $b_0 {*} 2^0 + b_{-1} {*} 2^{-1} + b_{-2} {*} 2^{-2} + b_{-3} {*} 2^{-3} + b_{-4} {*} 2^{-4}$
  $+ b_{-5} {*} 2^{-5} + b_{-6} {*} 2^{-6} + b_{-7} {*} 2^{-7}$

  →

  $b_0 + 0.5\, b_{-1} + 0.25\, b_{-2} + 0.125\, b_{-} + 0.0625\, b_{-4}$
  $+ 0.03125\, b_{-5} + 0.015625\, b_{-6} + 0.0078125\, b_{-7}$

- 0, 0.0078125, …, 1.9921875
  (2^8 = 256 possible numbers)

fixed point (reals)

- fixed point extend our notion of base-2 numbers to include negetive powers of 2
  - $2^{0} = 1$
    $2^{-1} = 0.5$
    $2^{-2} = 0.25$
    $2^{-3} = 0.125$
    $2^{-4} = 0.0625$
    $2^{-5} = 0.03125$
    $2^{-6} = 0.015625$
    $2^{-7} = 0.0078125$

fixed point (reals)

- fixed point representation uses **fixed** set of powers
  - $2^0 = 1$
    $2^{-1} = 0.5$
    $2^{-2} = 0.25$
    $2^{-3} = 0.125$
    $2^{-4} = 0.0625$
    $2^{-5} = 0.03125$
    $2^{-6} = 0.015625$
    $2^{-7} = 0.0078125$

floating point (reals)

- why fix the set of powers (of 2) that we can use?
- let the "decimal point" float...
- scientific notation in base-2

$$+/- 1.f * 2^{e}$$

  - f is the mantissa (fraction part)
  - e is the exponent

floating point (reals)

- $+/- \texttt{1.f} * 2^{\texttt{e-127}}$
- C float ~ 4 byte (IEEE standard)
(1 sign bit, 8 exponent bits, 23 fraction bits)

- bit 31 ~ sign bit (msb)
- bits 30-23 ~ exponent
  - unsigned
  - 0 → 255 maps to exponent range -127 → 128
- bits 22-0 ~ mantissa (fraction)
  - negative powers of 2 (-1 → -23)

floating point (reals)

- +/– $1.f * 2^{e-127}$
- C float ~ 4 byte
  - 1 sign bit, 8 exponent bits, 23 fraction bits
- C double ~ 8 bytes
  - 1 sign bit, 11 exponent bits, 52 fraction bits