Floating Point Arithmetic

Most scientific and engineering computations require "decimal" arithmetic, i.e. numbers containing a decimal point. Floating point is the computer implementation of "real" arithmetic with limited precision. Until recently, only the largest computers had floating point hardware as standard equipment.

Q. *What connects Bellevue to Seattle?*

A. *The Evergreen Floating Point Bridge.*

@1997 LS All Rights Reserved

Terminology

Scientific notation: 3.1557×10^{9}, 3.1416×10^{0}, 3.1557×10^{-9}

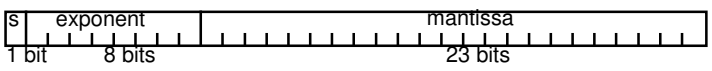
Normalized number: 31.557×10^{8} --> 0.31557×10^{10}

Decimal fractions:		Binary fractions:		
0.1_{10}	1×10^{-1}	0.1_{2}	1×2^{-1}	1/2
0.44_{10}	44×10^{-2}	0.01_{2}	1×2^{-2}	1/4
0.612_{10}	612×10^{-3}	0.101_{2}	5×2^{-3}	5/8

General form of floating point: $1.fffffffffffx2^{eee}$

What is the first digit of every normalized binary fp number?
Constituents are: sign, exponent, significand or mantissa.

Form of floating point: $(-1)^{s}Fx2^{E}$



@1997 LS All Rights Reserved

Further Float Facts

Floating point is well understood: IEEE 754 Floating Point Standard.

Single Precision -- one word representation for fp.

Double Precision -- two word representation for fp.

1 bit sign, 11 bit exponent, 20+32=52 matissa

Range:

Single: 2.0×10^{-38} through 2.0×10^{38}

Double: 2.0×10^{-308} through 2.0×10^{308}

MSB of normalized mantissa not represented: 24, 53.

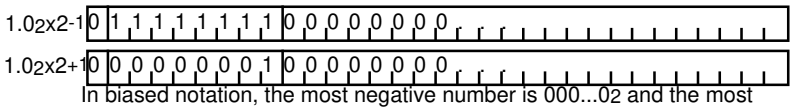
Zero is represented as 000. . . 0, i.e. it has no implied MSB.

Floating point representation is designed so that when a<b as a signed magnitude bit sequence, a<b as floating point numbers -- easy as sorting integers.

@1997 LS All Rights Reserved

Biased Representation

Signed exponents would complicate comparisons.



In biased notation, the most negative number is 000...0_2 and the most positive number is 111...1_2.

Since the single precision exponent field is 8 bits, allowing 256 different configurations, the bias for sp fp is 127.

+2 is represented as 2+127=129 = 1000 0001

-2 is represented as -2+127=125 = 0111 1101

The bias for double precision is 1023

The formula for computing single precision fp is thus:

(-1) x (1 + mantissa) x 2 (exponent - bias)

@1997 LS All Rights Reserved

Example Representation

Find the floating point for 5.125?

$5.125 = 5 + 0.125 = 5 + 1/8 = 5 + 1x2^{-3} = 101_2 + .001_2 = 101.001$

Normalize $101.001_2 x 2^0$ to 1.01001_{2x2^2}

Thus $5.125 = (-1)^0 \times (1+.0100\ 1000\ 0\ ...) \times 2^{129}$

In reverse, what floating point number is

$(-1)^1 \times (1+.0111\ 0000\ 0\ ...) \times 2^{130}$

In binary scientific notation it is -1.0111×2^3

Reducing the exponent to 0, yields $-1011.1_2 \times 2^0 = -1011.1_2$

$= -11_{10} + 2^{-1} = -11_{10} + 1/2 = -11.5_{10}$.

@1997 LS All Rights Reserved

Multiplying FP Numbers

Recall that fp is scientific notation, so arithmetic is logarithmic:

Add the exponents (reduce by the bias), multiply the mantissas, and renormalize if needed.

0.75 x 24 = 0.11 x 11000
1.1x2^{-1} times 1.1x2^4
Add exponents: -1 + 4 = 3
Multiply fractions:

```
      1.1
      1.1
      1 1
    1 1
  1 0.0 1
```

Result: 10.01x2^3 = 10010
or 1.001x2^4 normalized

0.75 x 24 = 0.11 x 11000
1.1x2$^{(126-127)}$ times 1.1x2$^{(131-127)}$
Add exponents:
126-127+131-127=257-254 = 3
Multiply fractions:

```
      1.1
      1.1
      1 1
    1 1
  1 0.0 1
```

Result: 10.01x2^3 = 10010
or 1.001x2$^{(131-127)}$ normalized

@1997 LS All Rights Reserved

Adding FP Numbers

Requires that the binary points to be aligned, which is equivalent to having the same exponent. . . shift the mantissa of the smaller right, raising its exponent.

1.000x2^{-1} plus 1.011x2^{2}

Shift smaller right: 1.000x2^{-1}=0.100x2^{0}=0.0100x2^{1}=0.001x2^{2}
Add: 1.011 + 0.001=1.100
Renormalize: 1.100

Result: 1.100x2^{2}

@1997 LS All Rights Reserved

FP Instructions

There are 32 separate floating point registers, $f0, $f1, $f2, . . ., $f31.

The even numbered registers are used for single precision.

An even/odd pair is used for double precision, with the odd numbered register holding the least significant mantissa bits.

Special load/store instructions move fp data to/from memory:

`l.s, s.s, l.d, s.d`.

Arithmetic operations (R-type) come in single/double forms.

`add.s, add.d, sub.s, sub.d, mul.s, mul.d`

Comparisons make direct tests and set a "condition bit"

`c.le.s, c.lt.s, c.eq.s, c.ne.s, c.gt.s, c.ge.s`

`c.le.d, c.lt.d, c.eq.d, c.ne.d, c.gt.d, c.ge.d`

Branch if true, `bc1t`, and branch if false, `bc1f`.

@1997 LS All Rights Reserved