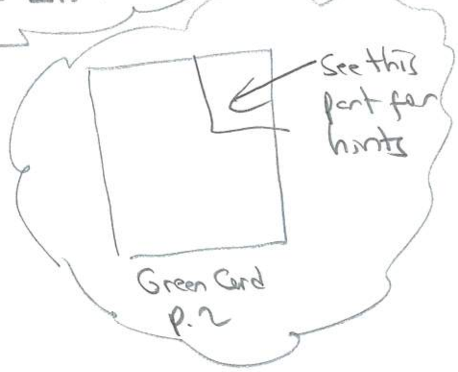
Floating Point Numbers

IEEE 754 Representation

Single Precision: 32 bits

Double Precision: 64 bits

Read FDIV problem of Intel 1994



Single Precision

1	8	23	← No. of bits
s	exp	mantissa	

Sign: 0 = +, 1 = −

Exponent: Excess 127, 127 = 7F

significand / mantissa / fraction } diff. names for the same thing

Normalized representation
1<x<10 scientific notation

A2B = A.2B x 16^2

101.01 = 1.0101 x 2^2

0.001 = 1.0x2^(-3)

mantissa = after .

before . digit implied not stored

Example Dec 80.5 in Single Precision

$\frac{80}{16} = 5$ Remainder = 0

$\frac{5}{16} = 0$ Remainder: 5

↑ write numbers in this order

$80 = 50_{16}$

Fractional Part

0.5
x16
8.0 ← when 0 stop else continue

$0.5 = 0.8_{16}$

$80.5 = 50.8_{16}$ → 0101 0000 . 1000

$1010000.10 \Rightarrow 1.\overbrace{01000010}^{\text{mantissa}} \times 2^6$ (1. not stored)

$127 + 6 = 133$ $\quad$ $7F + 6 = 85_{16}$ ← exponent

S
0 1000 0101 0100001000 . . . 0
4 2 A 1 . . . ← we need to have 32 bits (mantissa)

↑ +ve number

$\boxed{42A10000_{16}}$

FP Nos/1 - Oct 20, '20 (5 pages)

Double Precision

Example 1

1	11	52	← size in bits
	exp.	mantissa	

excess $1023 = 3FF_{16}$

Exponent is stored with an excess value of $1023\ (3FF_{16})$

$$80.5 \Rightarrow 50.8_{16}$$

$$\underbrace{101\ 0000.1000 \Rightarrow 1.\overbrace{01000010}^{\text{Mantissa}} \times 2^6}_{}$$

Normalized Scientific Representation

Excess 1023 representation for the exponent 6

$$\begin{array}{r} 1023 \\ +\ \ 6 \\ \hline 1029 \end{array} \qquad \begin{array}{r} 3FF \\ +\ 6 \\ \hline 405_{16} \end{array}$$

For example for decimal number

$0.072 \Rightarrow 7.2\times10^{-2}$

$1234.5 \Rightarrow 1.2345\times10^{3}$

Normalized or Scientific Representation

→ mantissa ← 64 bits

$$\overset{S}{\underbrace{0\ 100}_{4}}\ \underbrace{0000}_{0}\ \underbrace{0101}_{5}\ \underbrace{0100}_{4}\ \underbrace{0010}_{2} \cdots 0$$

$$\boxed{40\ 54\ 20\ 00\ 00\ 00\ 00\ 00_{16}}$$

Example 2

-0.05_{16} show in single precision

$$0.00000101 \Rightarrow 1.01 \times 2^{-6}$$

↑ not stored

$$\begin{array}{r} 7F \\ -6 \\ \hline 79 \end{array}$$

→ Mant

$$\overset{S}{\underbrace{1\ 011}_{B}}\ \underbrace{1\ 100}_{C}\ \underbrace{1\ 010}_{A}\ \underbrace{0000}_{0} \cdots$$

$$\boxed{BC\ A0\ 00\ 00_{16}}$$

Example 3

Try double precision: Is it $\boxed{BF\ 94\ 00\ 00\ 00\ 00\ 00\ 00_{16}}$?

From IEEE 754 Single Precision Representation

To Decimal Conversion

$42\ E4\ 00\ 00_{16}$ is given $\Rightarrow X_{10}$?

0100 0010 1110 0100 0000 ...

S:+

85 is exponent

mantissa

1.1100100

not stored implied

85 excess 127 ($7F_{16}$) representation

$$\begin{array}{r} 85 \\ -\ 7F \\ \hline 06 \end{array}$$

$1.1100100 \times 2^6 \Rightarrow 1110\ 0100$

$\Downarrow$

$\underbrace{111}_{7}\ \underbrace{0010}_{2}.0 \Rightarrow 72_{16}$

$72_{16} = 7 \times 16 + 2 = 112 + 2 \Rightarrow \boxed{114_{10}}$

Problem to be solved

Given $A1\ 49\ 00\ 00_{16}$

Interpret this number as a

1. Floating point number
2. Interpret as an integer
3. Interpret as a sign magnitude number

S magnitude

1: -ve 0 = +ve

0100 ⇒ +4

1100 ⇒ -4

0000 ⇒ +0

1000 ⇒ -0

FP No/3 - Oct. 20, '20

Floating point addition using scientific notation

Decimal example

$$9.999 \times 10^1 + 1.610 \times 10^{-1}$$

Step 1: Align the decimal point of the numbers with the larger exponent

9.999×10^1 $\qquad$ $1.610 \times 10^{-1} \Rightarrow 0.161 \times 10^0$

$\Rightarrow 0.016 \times 10^1$

lost a digit

Step 2 Add the new form of the numbers

$$\begin{array}{r} 9.999 \times 10^1 \\ 0.016 \times 10^1 \\ \hline 10.015 \times 10^1 \end{array}$$

Step 3 Normalize the result

$$10.015 \times 10^1 \Rightarrow 1.0015 \times 10^2$$

Assuming that we are allowed to keep three digits after decimal point

$\Rightarrow 1.001 \times 10^2$ lost precision

(subtraction)
Binary addition using scientific notation

Example $1.000_2 \times 2^{-1} - 1.110 \times 2^{-2}$

called binary point

Step 1: Align the binary point of the numbers

$1.000_2 \times 2^{-1}$ $1.110 \times 2^{-2} \Rightarrow 0.111 \times 2^{-1}$

Step 2 Add numbers

$$\begin{array}{r} 1.000 \times 2^{-1} \\ -0.111 \times 2^{-1} \\ \hline 0.001 \times 2^{-1} \end{array}$$

borrow 1 comes as 10 -> decimal
borrow 1 comes as 2 -> binary

Step 4 Normalize result

$0.001 \times 2^{-1} \Rightarrow 1.000 \times 2^{-4}$

Example

$1.011 \times 2^{-1} + 1.011 \times 2^{-6}$

$1.011 \times 2^{-6} \Rightarrow 0.1011 \times 2^{-5}$
0.01011×2^{-4}
0.001011×2^{-3}
0.0001011×2^{-2}
0.00001011×2^{-1}

$$\begin{array}{r} 1.011 \times 2^{-1} \\ 0.000 \times 2^{-1} \\ \hline 1.011 \times 2^{-1} \end{array}$$

← lost

FP No.15 - Oct 20, '20