Number Representations

Note Title 9/10/2007

Make binary more "human friendly"

Hexadecimal notation commonly called "hex"

not base 6 is base 16

char. set 0-9, a-f (A-F)

a = 10 b = 11 c = 12

d = 13 e = 14 f = 15

C notation prefix 0x 0x12 0xa3

positional notation

base b $\quad xyz_b = x \cdot b^2 + y \cdot b + z$

$= x \cdot 256 + y \cdot 16 + z$

so $0x2c = 2 \cdot 16 + 12 = 44_{10}$

hex $\longleftrightarrow$ binary very easy

hex → binary expand each digit to 4 bit binary rep

0x3d50

3 d 5 0

0011 1101 0101 0000$_2$

binary to hex 4-bit groups of binary bits get converted to hex digits

0100 1000

4 8

0x48

What about fractions?

positional # systems work on both sides of the radix point.

radix r (n integer digits, m fractional)

$$val = a_{n-1} \cdot r^{n-1} + a_{n-2} r^{n-2} + \cdots a_0 r^0 + a_{-1} r^{-1} + a_{-2} r^{-2} + \cdots + a_{-m} \cdot r^{-m}$$

$$WX.YZ_{16} = W \cdot 16 + X + Y \cdot \frac{1}{16} + Z \cdot \frac{1}{16^2}$$

Integers — radix point assumed to be at far right

Fixed-point — have radix point at some specified location

Q notation

Qn.m — number with n+m bits (digits), n integer and m fractional

sign bit is often in addition to n+m bits

e.g. $Q_{0.15}$ is very common # rep.

2's comp. # w/ 15 bits to right of rp.

Qm notation $m+1$ bits, m are frac.

$Q_{15} \equiv Q_{0.15}$

Q_3 4-bits

$wxyz = w.xyz = w \cdot (-1) + x \cdot \frac{1}{2} + y \cdot \frac{1}{4} + z \cdot \frac{1}{8}$

range -1 to $+\frac{7}{8}$ w/ resolution $\frac{1}{8}$

Floating Point Rep.

represents reals — use scientific notation

base 10 $x \cdot 10^y$ $0.32 \times 10^{-3} = 0.00032$

base 2 $x \cdot 2^y$ 3.2×10^{-4}

y ← exponent

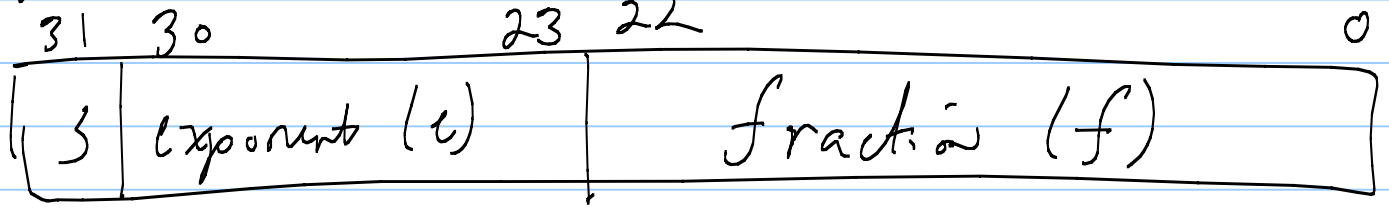
x ← mantissa

IEEE std 754

single precision 32 bit

double " 64 bit

S.P. 32 bits



1 8 23

sign

$$value = (-1)^s \times 2^{e-127} \times 1.f$$

(hidden "1")

range $\pm 2 \times 10^{\pm 38}$

$s = 0$, $e = 0$, $f = 0$ $\Rightarrow$ value = zero

$e = 255 = 11111111_2$

$f = 0$ $\Rightarrow$ value = $(-1)^s \times$ infinity

$f \neq 0$ $\Rightarrow$ value = "not a number"

NaN

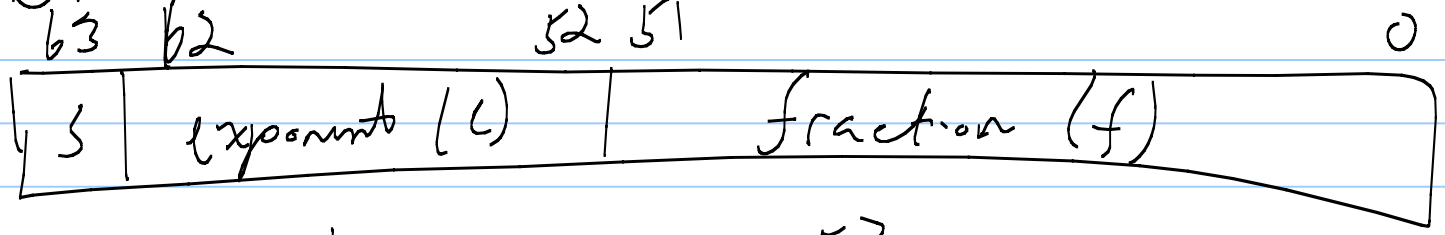
trigger exception

$e = 0$, $f \neq 0$ denormalized number

value $(-1)^s \times 2^{-126} \times 0.f$

↑ hidden "0"

DP 64 bits



bits 1 11 52

$$value = (-1)^s \times 2^{e-1023} \times 1.f$$

$$range = \pm 2 \times 10^{\pm 308}$$

$e = 0$, $f \neq 0$ denormalized

$$value = (-1)^s \times 2^{-1022} \times 0.f$$

Text - Characters + Strings

2 common stds.

ASCII American Standard Code for Information Interchange

7-bit code rep basic Latin chars. and numbers [A-Z, a-z, 0-9], some common punc., and control chars.

Unicode - 16-bit code extends to much wider char set / languages

1st 128 code coincide w/ ASCII

C chars / strings assume ASCII

string as seq. of chars, stored one byte per char, terminated by NULL char (0).

"Hello!"

H	0x48
e	0x65
l	0x6c
l	0x6c
o	0x6f
!	0x21
NULL	0x0

Numbers are assigned in order of incr. value

'0'	0x30
'1'	0x31
⋮	
'9'	0x39

diff of 0x30

Letters also assigned in lexicographical order

'A' 0x41 0,1000001

'B' 0x42

'Z' 0x5a

diff of 0x20

'a' 0x61

'b' 0x62

:

'z' 0x7a

0010 0000

0110 0001

b a

First 32 chars. 0 - 0x1f are control codes

0x00	NULL
0x07	bell
0x0a	line feed
0x0d	carriage return

End of line conventions are OS dep.

Windows	0x0a, 0x0d
Linux	0x0a

In C '\n' is mapped to OS specific eol