Computer Organization

Floating Point (Part I)

Prof. Roger Luis Uy
De La Salle University
College of Computer Studies

Floating Point

- Real numbers are difficult to represent. This involves a certain amount of approximation with knowledge of a few significant digits at the start of the number, *e.g.*,

$$\pi \approx 3.1415926536$$

$$c \approx 299{,}792{,}458$$

Floating Point

- One approach is to use *floating-point representation* where the number is given according to a fixed number of decimal places (and precision)

Floating Point

- Scientists and engineers use scientific notation where a number is expressed as

$$M \times 10^{E}$$

Where M is the mantissa, E is the exponent and 10 is the base.

10^{E} is also known as scaling factor

e.g.,

$$c \approx 2.998 \times 10^{8}$$

$$\text{Avogadro's Number} \approx 6.0247 \times 10^{23}$$

$$\text{Planck's Constant} \approx 6.6254 \times 10^{-27}$$

Floating Point

- Computers represent real numbers using scientific notation in base 2. $\Rightarrow$ *floating-point*

$$(-1)^S \times M \times 2^E$$

Floating Point

- The mantissa is also typically *normalized* ($1 \le |M| < 2$) and can be represented as 1 plus a fraction
- Floating point standard: IEEE-754 (Institute of Electrical and Electronics Engineers Standard 754). Originally 1985, current version is 2008.

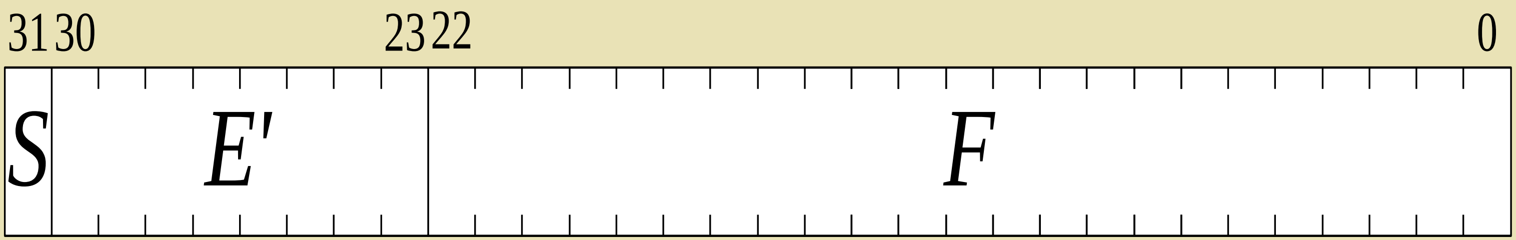
IEEE-754

- IEEE-754 defines the following:
  - Arithmetic format: binary and decimal floating point data
  - Rounding rules
  - Exception handling (divide by zero, etc.)
  - Floating point operations

IEEE-754 Arithmetic Format

- IEEE-754 arithmetic format
  – 16-bit binary: IEEE Half Precision (not basic)
  – 32-bit binary: IEEE Single Precision
  – 64-bit binary: IEEE Double Precision
  – 128-binary: IEEE Quadruple Precision
  – 32-bit decimal: Decimal-32
  – 64-bit decimal: Decimal-64
  – 128-bit decimal: Decimal-128

IEEE Single Precision



$$(-1)^S \times 1.F \times 2^{E'-127}$$

IEEE Single Precision

- Uses 32 bits: 1-bit sign, 23-bit mantissa, 8-bit exponent.
- The fraction is normalized (i.e. 1.M).
- The exponent is biased (i.e., excess 127).
- Base of the exponent is 2
- Mantissa has 24-bits of precision = 7.225 decimal digits ($\log_{10}(2^{24})$)
- The smallest normalized number is $\pm\ 1.0\text{x}\ 2^{-126}$ or $\pm 1.17 \times 10^{-38}$.
- The largest normalized number is $\pm$ $1.11111111111111111111111\text{x}\ 2^{127}$ or $\pm 3.40 \times 10^{38}$.

Single Precision (Special Value)

Sign Bit	E’	Mantissa	Value
0	0000 0000	000 0000 0000 0000 0000 0000	+0 (Positive Zero)
1	0000 0000	000 0000 0000 0000 0000 0000	-0 (Negative Zero)
0/1	0000 0000	<>0	Denomalized
0	1111 1111	000 0000 0000 0000 0000 0000	+ Infinity
1	1111 1111	000 0000 0000 0000 0000 0000	- Infinity
x	1111 1111	0xx xxxx xxxx xxxx xxxx xxxx	sNaN
x	1111 1111	1xx xxxx xxxx xxxx xxxx xxxx	qNaN

Example (single precision)

- 1.00010000×2^5
- Sign bit = 0
- Exponent representation = 5+127=132 = 1000 0100
- Mantissa = 000 1000 0000 0000 0000 0000
- => 0 10000100 00010000000000000000000
- or 42080000h

Example (single precision)

- 11001.1111 x 2^{12}
- (normalize): 1.10011111x 2^{16}
- Sign bit = 0
- Exponent representation = 16+127=143 = 1000 1111
- Mantissa = 100 1111 1000 0000 0000 0000
- => 0 1000010 10011111000000000000000
- or 47CF8000h

Example (single precision)

- $0.0000001101 \times 2^{-3}$
- (normalize): 1.101×2^{-10}
- Sign bit = 0
- Exponent representation = -10+127=117 = 0111 0101
- Mantissa = 101 0000 0000 0000 0000 0000
- => 0 01110101 10100000000000000000000
- or 3AD00000h

Smallest Denormalized Number (Single Precision)

- Denormalized number: 0.00000000000000000000001 $\times 2^{-126}$
- 1.0 $\times 2^{-149}$ ~ 1.4 $\times 10^{-45}$
- Sign bit = 0
- Exponent = 0000 0000
- Mantissa = 00000000000000000000001

IEEE Double Precision

63	62 … 52	51 … 0
S	E'	F

$$(-1)^S \times 1.F \times 2^{E'-1023}$$

IEEE Double Precision

- Uses 64 bits: 1-bit sign, 52-bit mantissa, 11-bit exponent.
- The fraction is normalized (i.e. 1.M).
- The exponent is biased (i.e., excess 1023).
- Base of the exponent is 2
- Mantissa has 53-bits of precision = 15.955 decimal digits ($\log_{10}(2^{53})$)
- The smallest normalized number is $\pm\ 1.0\text{x}\ 2^{-1022}$ or is $\pm 2.23\times 10^{+308}$.
- The largest normalized number is $\pm\ 1.11111\ldots 1\text{x}\ 2^{+1023}$ or $\pm 1.80\times 10^{-308}$

Double Precision (Special Value)

Sign Bit	E’	Mantissa	Value
0	000 0000 0000	0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000	+0 (Positive Zero)
1	000 0000 0000	0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000	-0 (Negative Zero)
0/1	000 0000 0000	<>0	Denomalized
0	111 1111 1111	0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000	+ Infinity
1	111 1111 1111	0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000	- Infinity
x	111 1111 1111	0xxx xxxx xxxx xxxx xxxx xxxx xxxx xxxx xxxx xxxx xxxx xxxx xxxx	sNaN
x	111 1111 1111	1xxx xxxx xxxx xxxx xxxx xxxx xxxx xxxx xxxx xxxx xxxx xxxx xxxx	qNaN

Example (double precision)

- 1.00010000 x 2^5
- Sign bit = 0
- Exponent representation = 5+1023=1028 = 100 0000 0100
- Mantissa = 00010…….0
- => 0 10000000100000100….0000000
- or 4041000000000000h

Example (double precision)

- $11001.1111 \text{ x } 2^{12}$
- (normalize): $1.10011111\text{x } 2^{16}$
- Sign bit = 0
- Exponent representation = 16+1023=1039 = 100 0000 1111
- Mantissa = 10011111000….0
- => 0 10000001111 10011111000000…00
- or 40F9F000000000000h

Example (double precision)

- $0.0000001101 \times 2^{-3}$
- (normalize): 1.101×2^{-10}
- Sign bit = 0
- Exponent representation = -10+1023=1013 = 01111110101
- Mantissa = 1010…….0
- => 0 01111110101 1010…0
- or 3F5A000000000000h

Denormalized (Double Precision)

- Denormalized number: $0.000\ldots001 \times 2^{-1022}$
- $1.0 \times 2^{-1074} \sim 4.94 \times 10^{-324}$
- Sign bit = 0
- Exponent = 000 0000 0000
- Mantissa = 00000……1

IEEE Half Precision

- Uses 16 bits: 1-bit sign, 10-bit mantissa, 5-bit exponent.
- The fraction is normalized (i.e. 1.M).
- The exponent is biased (i.e., excess 15).
- Base of the exponent is 2
- Mantissa has 11-bits of precision = 3.3 decimal digits ($\log_{10}(2^{11})$)
- The smallest normalized number is $\pm$ 1.0x 2^{-14} or is $\pm 6.103515625\times10^{-5}$.
- The largest normalized number is $\pm$ 1.11111…1x 2^{+15} or $\pm$65535

IEEE Quadruple Precision

- Uses 128 bits: 1-bit sign, 112-bit mantissa, 15-bit exponent.
- The fraction is normalized (i.e. 1.M).
- The exponent is biased (i.e., excess 16383).
- Base of the exponent is 2
- Mantissa has 113-bits of precision = 34.016 decimal digits ($\log_{10}(2^{113})$)
- The smallest normalized number is $\pm$ 1.0x 2^{-16382} or is $\pm 3.36 \times 10^{-4932}$.
- The largest normalized number is $\pm$ 1.11111…1x 2^{+16383} or $\pm 1.18 \times 10^{4932}$