Computer Architecture

Spring 2020

Hamed Farbeh

farbeh@aut.ac.ir

Department of Computer Engineering

Amirkabir University of Technology

Copyright Notice

Lectures adopted from

- Computer Organization and Design: The Hardware/Software Interface, 5th edition, David A. Patterson, John L. Hennessy, MK pub., 2014
  - Chapter 3: Arithmetic for Computers

Chapter 3

Arithmetic for Computers

Floating Point

- Representation for non-integral numbers
  - Including very small and very large numbers
- Like scientific notation
  - -2.34×10^{56} — normalized
  - $+0.002 \times 10^{-4}$ — not normalized
  - $+987.02 \times 10^{9}$ — not normalized
- In binary
  - $\pm 1.xxxxxxx_2 \times 2^{yyyy}$
- Types `float` and `double` in C

Floating Point Standard

- Defined by IEEE Std 754-1985
- Developed in response to divergence of representations
  - Portability issues for scientific code
- Now almost universally adopted
- Two representations
  - Single precision (32-bit)
  - Double precision (64-bit)

IEEE Floating-Point Format

	single: 8 bits double: 11 bits	single: 23 bits double: 52 bits
S	Exponent	Fraction

$$x = (-1)^S \times (1 + \text{Fraction}) \times 2^{(\text{Exponent} - \text{Bias})}$$

- S: sign bit (0 ⇒ non-negative, 1 ⇒ negative)
- Normalize significand: 1.0 ≤ |significand| < 2.0
  - Always has a leading pre-binary-point 1 bit, so no need to represent it explicitly (hidden bit)
  - Significand is Fraction with the “1.” restored (Unsigned)
- Exponent: excess representation: actual exponent + Bias
  - Ensures exponent is unsigned
  - Single: Bias = 127; Double: Bias = 1023

Bias = $(2^{E-1}-1)$

Biased Exponent

Decimal Exponent	Signed-2's Complement	Biased Notation (Excess-16)	Decimal Value of Biased Notation
15	01111	11111	31
14	01110	11110	30
...	...	...	...
1	00001	10001	17
0	00000	10000	16 (bias)
-1	11111	01111	15
...	...	...	...
-15	10001	00001	1
-16	10000	00000	0

Single-Precision Range

- Exponents 00000000 and 11111111 reserved
- Smallest value
  - Exponent: 00000001
    $\Rightarrow$ actual exponent = 1 – 127 = –126
  - Fraction: 000…00 $\Rightarrow$ significand = 1.0
  - $\pm 1.0 \times 2^{-126} \approx \pm 1.2 \times 10^{-38}$
- Largest value
  - exponent: 11111110
    $\Rightarrow$ actual exponent = 254 – 127 = +127
  - Fraction: 111…11 $\Rightarrow$ significand ≈ 2.0
  - $\pm 2.0 \times 2^{+127} \approx \pm 3.4 \times 10^{+38}$

Double-Precision Range

- Exponents 0000…00 and 1111…11 reserved
- Smallest value
  - Exponent: 00000000001
    ⇒ actual exponent = 1 – 1023 = –1022
  - Fraction: 000…00 ⇒ significand = 1.0
  - $\pm 1.0 \times 2^{-1022} \approx \pm 2.2 \times 10^{-308}$
- Largest value
  - Exponent: 11111111110
    ⇒ actual exponent = 2046 – 1023 = +1023
  - Fraction: 111…11 ⇒ significand ≈ 2.0
  - $\pm 2.0 \times 2^{+1023} \approx \pm 1.8 \times 10^{+308}$

Floating-Point Precision

- Relative precision
  - all fraction bits are significant
  - Single: approx 2^{-23}
    - Equivalent to $23 \times \log_{10}2 \approx 23 \times 0.3 \approx 6$ decimal digits of precision
  - Double: approx 2^{-52}
    - Equivalent to $52 \times \log_{10}2 \approx 52 \times 0.3 \approx 16$ decimal digits of precision

Floating-Point Example

- Represent –0.75
  - $-0.75 = (-1)^1 \times 1.1_2 \times 2^{-1}$
  - S = 1
  - Fraction = $1000\ldots00_2$
  - Exponent = –1 + Bias
    - Single: –1 + 127 = 126 = 01111110_2
    - Double: –1 + 1023 = 1022 = 01111111110_2
- Single: 1011111101000…00
- Double: 1011111111101000…00

Floating-Point Example

- What number is represented by the single-precision float

  11000000101000…00
  - S = 1
  - Fraction = $01000\ldots00_2$
  - Exponent = $10000001_2 = 129$
- $x = (-1)^1 \times (1 + 01_2) \times 2^{(129 - 127)}$

  $= (-1) \times 1.25 \times 2^2$

  $= -5.0$

Denormal Numbers

- Exponent = 000...0 $\Rightarrow$ hidden bit is 0

$$x = (-1)^S \times (0 + \text{Fraction}) \times 2^{-\text{Bias}}$$

- Smaller than normal numbers
  - allow for gradual underflow, with diminishing precision
- Denormal with fraction = 000...0

$$x = (-1)^S \times (0 + 0) \times 2^{-\text{Bias}} = \pm 0.0$$

Two representations of 0.0!

Infinities and NaNs

- Exponent = 111...1, Fraction = 000...0
  - ±Infinity
  - Can be used in subsequent calculations, avoiding need for overflow check
- Exponent = 111...1, Fraction ≠ 000...0
  - Not-a-Number (NaN)
  - Indicates illegal or undefined result
    - e.g., 0.0 / 0.0
  - Can be used in subsequent calculations

Single precision		Double precision		Object represented
Exponent	Fraction	Exponent	Fraction	
0	0	0	0	0
0	Nonzero	0	Nonzero	± denormalized number
1–254	Anything	1–2046	Anything	± floating-point number
255	0	2047	0	± infinity
255	Nonzero	2047	Nonzero	NaN (Not a Number)

FIGURE 3.13 EEE 754 encoding of floating-point numbers. A separate sign bit determines the sign. Denormalized numbers are described in the *Elaboration* on page 222. This information is also found in Column 4 of the MIPS Reference Data Card at the front of this book.

	single	double	extended	full quadruple
Format length	32	64	80	128
Stored fraction bits	23	52	64	112
Precision (p)	24	53	64	113
Biased-exponent bits	8	11	15	15
Minimum exponent	−126	−1022	−16382	−16382
Maximum exponent	127	1023	16383	16383
Exponent bias	127	1023	16383	16383
`macheps` (2^{-p+1})	2^{-23}	2^{-52}	2^{-63}	2^{-112}
	$\approx$ 1.19e-07	$\approx$ 2.22e-16	$\approx$ 1.08e-19	$\approx$ 1.93e-34
Largest finite	$(1-2^{-24})2^{128}$	$(1-2^{-53})2^{1024}$	$(1-2^{-64})2^{16384}$	$(1-2^{-113})2^{16384}$
	$\approx$ 3.40e+38	$\approx$ 1.80e+308	$\approx$ 1.19e+4932	$\approx$ 1.19e+4932
Smallest normalized	2^{-126}	2^{-1022}	2^{-16382}	2^{-16382}
	$\approx$ 1.18e-38	$\approx$ 2.23e-308	$\approx$ 3.36e-4932	$\approx$ 3.36e-4932
Smallest denormalized	2^{-149}	2^{-1074}	2^{-16446}	2^{-16494}
	$\approx$ 1.40e-45	$\approx$ 4.94e-324	$\approx$ 1.82e-4951	$\approx$ 6.48e-4966