

ECE 645: Lecture 2

Number Representations

Little-Endian vs. Big-Endian

Residue Number Systems

Floating Point Representations

Galois Field Representations

Little-Endian vs. Big-Endian Representation of Integers

Little-Endian vs. Big-Endian Representation

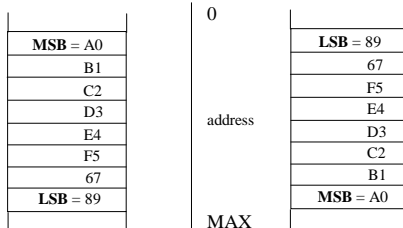
A0 B1 C2 D3 E4 F5 67 89₁₆

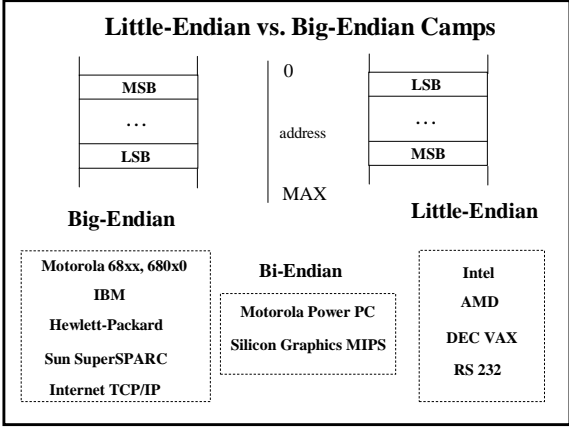
MSB

LSB

Big-Endian

Little-Endian



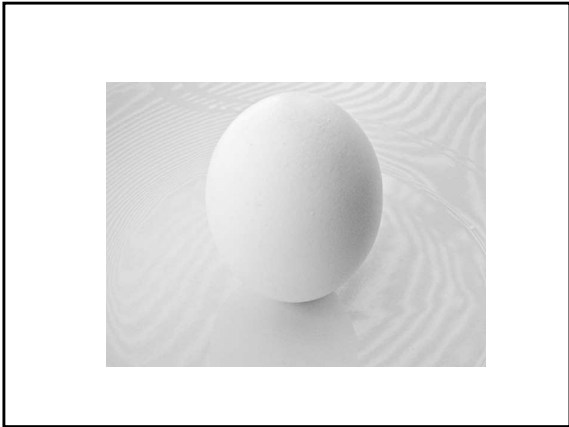


Little-Endian vs. Big-Endian

Origin of the terms

Jonathan Swift, *Gulliver's Travels*

- A law requiring all citizens of Lilliput to break their soft-eggs at the little ends only
- A civil war breaking between the Little Endians and the Big-Endians, resulting in the Big Endians taking refuge on a nearby island, the kingdom of Blefuscu
- Satire over holy wars between Protestant Church of England and the Catholic Church of France



Little-Endian vs. Big-Endian
Advantages and Disadvantages

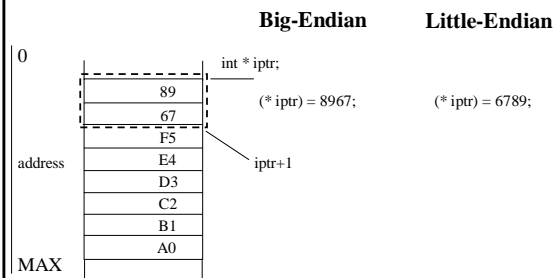
Big-Endian

- easier to determine a sign of the number
- easier to compare two numbers
- easier to divide two numbers
- easier to print

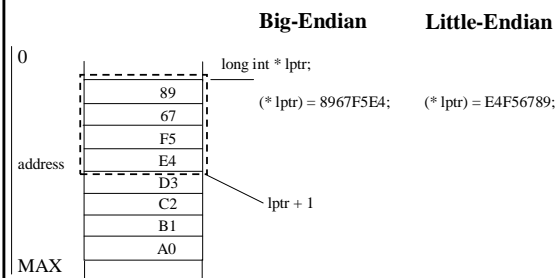
Little-Endian

- easier addition and multiplication of multiprecision numbers

Pointers (1)



Pointers (2)



Divisibility Rules and Operations Modulo

Divisibility

$a \mid b$ a divides b
 a is a divisor of b

$a \mid b$ iff $\exists c \in \mathbb{Z}$ such that $b = c \cdot a$

$a \nmid b$ a does not divide b
 a is not a divisor of b

True or False?

-3 | 18 14 | 7 7 | 63 -13 | 65 14 | 21 14 | 14
0 | 63 7 | 0 -5 | 0 0 | 0

Quotient and remainder

Given integers a and n , $n > 0$

$\exists!$ $q, r \in \mathbf{Z}$ such that

$$a = q \cdot n + r \quad \text{and} \quad 0 \leq r < n$$

q – **quotient** $q = \left\lfloor \frac{a}{n} \right\rfloor = a \operatorname{div} n$

r – **remainder** $r = a - q \cdot n = a - \left\lfloor \frac{a}{n} \right\rfloor \cdot n =$
(of a divided by n) $= a \bmod n$

$$32 \bmod 5 =$$

$$-32 \bmod 5 =$$

Integers congruent modulo n

Two integers a and b are **congruent modulo n**
(**equivalent modulo n**)

written $a \equiv b$

iff

$$a \bmod n = b \bmod n$$

or

$$a = b + kn, \quad k \in \mathbf{Z}$$

or

$$n \mid a - b$$

Rules of addition, subtraction and multiplication modulo n

$$a + b \bmod n = ((a \bmod n) + (b \bmod n)) \bmod n$$

$$a - b \bmod n = ((a \bmod n) - (b \bmod n)) \bmod n$$

$$a \cdot b \bmod n = ((a \bmod n) \cdot (b \bmod n)) \bmod n$$

$$9 \cdot 13 \bmod 5 =$$

$$25 \cdot 25 \bmod 26 =$$

**Residue Number Systems
RNS**

Chinese puzzle, 1500 years ago:
 What number has the remainders of 2, 3, and 2 when divided by the numbers 7, 5, and 3, respectively?

Chinese Remainder Theorem

Let
$$N = n_1 \cdot n_2 \cdot n_3 \cdot \dots \cdot n_M$$

and for any i, j $\gcd(n_i, n_j) = 1$

Then, any number $0 \leq A \leq N-1$ can be represented uniquely by

$A \leftrightarrow (a_1 = A \bmod n_1, a_2 = A \bmod n_2, \dots, a_M = A \bmod n_M)$

A can be reconstructed from (a_1, a_2, \dots, a_M) using equation

$$A = \sum_{i=1}^M (a_i \cdot N_i \cdot N_i^{-1} \bmod n_i) \bmod N$$
 where $N_i = \frac{N}{n_i} = n_1 \cdot n_2 \cdot \dots \cdot n_{i-1} \cdot n_{i+1} \cdot \dots \cdot n_M$

RNS(8 | 7 | 5 | 3)

RNS Arithmetic

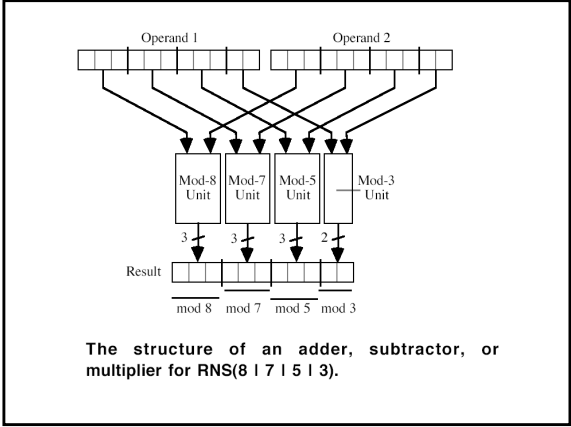
$(5 | 5 | 0 | 2)_{RNS}$ Represents $x = +5$

$(7 | 6 | 4 | 2)_{RNS}$ Represents $y = -1$

$(4 | 4 | 4 | 1)_{RNS}$ $x + y : \langle 5 + 7 \rangle_8 = 4, \langle 5 + 6 \rangle_7 = 4$, etc.

$(6 | 6 | 1 | 0)_{RNS}$ $x - y : \langle 5 - 7 \rangle_8 = 6, \langle 5 - 6 \rangle_7 = 6$, etc.
 (alternatively, find $-y$ and add to x)

$(3 | 2 | 0 | 1)_{RNS}$ $x \times y : \langle 5 \times 7 \rangle_8 = 3, \langle 5 \times 6 \rangle_7 = 2$, etc.



**Chinese Remainder Theorem
for $N=P \cdot Q$**

$N = P \cdot Q \quad \text{gcd}(P, Q) = 1$

$M \leftrightarrow (M_p = M \bmod P, M_Q = M \bmod Q)$

$$M = M_p \cdot \frac{N}{P} \cdot \left[\left[\frac{N}{P} \right]^{-1} \bmod P \right] + M_Q \cdot \frac{N}{Q} \cdot \left[\left[\frac{N}{Q} \right]^{-1} \bmod Q \right] \bmod N$$

$$= M_p \cdot Q \cdot ((Q^{-1}) \bmod P) + M_Q \cdot P \cdot ((P^{-1}) \bmod Q) \bmod N =$$

$$= M_p \cdot R_Q + M_Q \cdot R_P \bmod N$$

**Fast modular exponentiation
using Chinese Remainder Theorem**

$$M = C \bmod N$$

$C_p = C \bmod P$
 $d_p = d \bmod (P-1)$

$C_Q = C \bmod Q$
 $d_Q = d \bmod (Q-1)$

$$M_p = C_p \bmod P \quad M_Q = C_Q \bmod Q$$

$$M = M_p \cdot R_Q + M_Q \cdot R_P \bmod N$$

where

$$R_p = (P^{-1} \bmod Q) \cdot P = P^{Q-1} \bmod N$$

$$R_Q = (Q^{-1} \bmod P) \cdot Q = Q^{P-1} \bmod N$$

**Time of exponentiation
without and with Chinese Remainder Theorem**

SOFTWARE

Without CRT

$$t_{\text{EXP}}(k) = c_s \cdot k^3$$

With CRT

$$t_{\text{EXP-CRT}}(k) \approx 2 \cdot c_s \cdot \left(\frac{k}{2}\right)^3 = \frac{1}{4} t_{\text{EXP}}(k)$$

HARDWARE

Without CRT

$$t_{\text{EXP}}(k) = c_h \cdot k^2$$

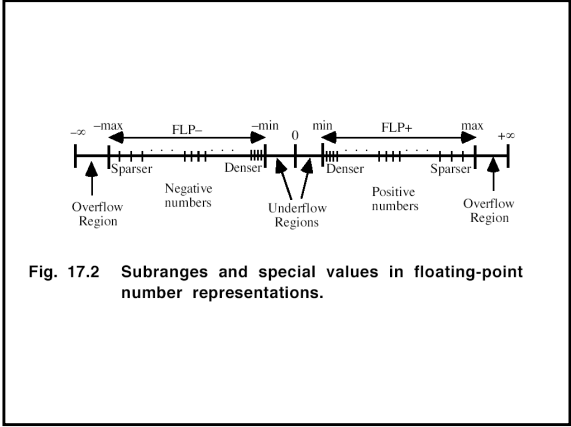
With CRT

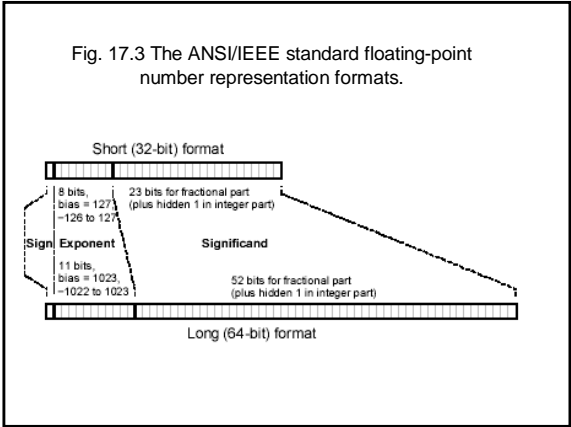
$$t_{\text{EXP-CRT}}(k) \approx c_h \cdot \left(\frac{k}{2}\right)^2 = \frac{1}{4} t_{\text{EXP}}(k)$$

Floating Point Representations

No finite number system can represent all real numbers
 Various systems can be used for a subset of real numbers

Fixed-point	$\pm w . f$	low precision and/or range
Rational	$\pm p / q$	difficult arithmetic
Floating-point	$\pm s \times b^e$	most common scheme
Logarithmic	$\pm \log_b x$	limiting case of floating-point





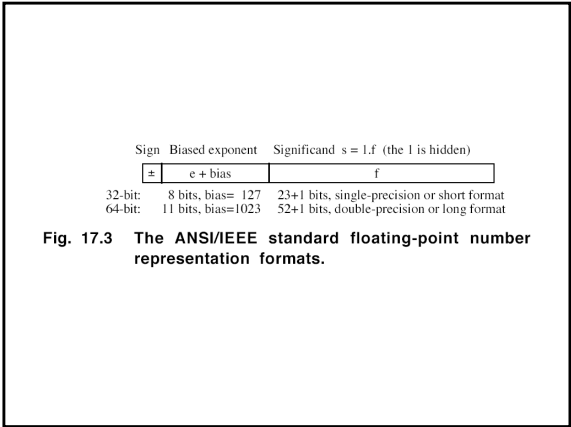


Table 17.1 Some features of the ANSI/IEEE standard floatingpoint number representation formats

Feature	Single/Short	Double/Long
Word width (bits)	32	64
Significand bits	23 + 1 hidden	52 + 1 hidden
Significand range	$[1, 2 - 2^{-23}]$	$[1, 2 - 2^{-52}]$
Exponent bits	8	11
Exponent bias	127	1023
Zero (± 0)	$e + bias = 0, f = 0$	$e + bias = 0, f = 0$
Denormal	$e + bias = 0, f \neq 0$ represents $\pm 0.f \cdot 2^{-126}$	$e + bias = 0, f \neq 0$ represents $\pm 0.f \cdot 2^{-1022}$
Infinity ($\pm \infty$)	$e + bias = 255, f = 0$	$e + bias = 2047, f = 0$
Not-a-number (NaN)	$e + bias = 255, f \neq 0$	$e + bias = 2047, f \neq 0$
Ordinary number	$e + bias \in [1, 254]$ $e \in [-126, 127]$ represents $1.f \times 2^e$	$e + bias \in [1, 2046]$ $e \in [-1022, 1023]$ represents $1.f \times 2^e$
<i>min</i>	$2^{-126} \approx 1.2 \times 10^{-38}$	$2^{-1022} \approx 2.2 \times 10^{-308}$
<i>max</i>	$\approx 2^{128} \approx 3.4 \times 10^{38}$	$\approx 2^{1024} \approx 1.8 \times 10^{308}$

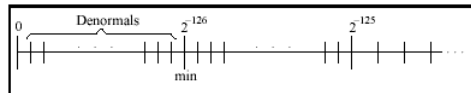
Exceptions

- divide by zero
- overflow
- underflow
- inexact result: rounded value not same as original
- invalid operation: examples include
 - addition $(+\infty) + (-\infty)$
 - multiplication $0 \times \infty$
 - division $0 / 0$ or ∞ / ∞
 - square-root operand < 0

Operations on special operands:

- Ordinary number $\div (+\infty) = \pm 0$
- $(+\infty) \times$ Ordinary number $= \pm \infty$
- NaN + Ordinary number = NaN

Fig. 17.4 Denormals in the IEEE single-precision format.



The IEEE floating-point standard also defines

The four basic arithmetic operations (+, -, ×, ÷) and \sqrt{x}
(must match the results that would be obtained if
intermediate computations were infinite-precision)

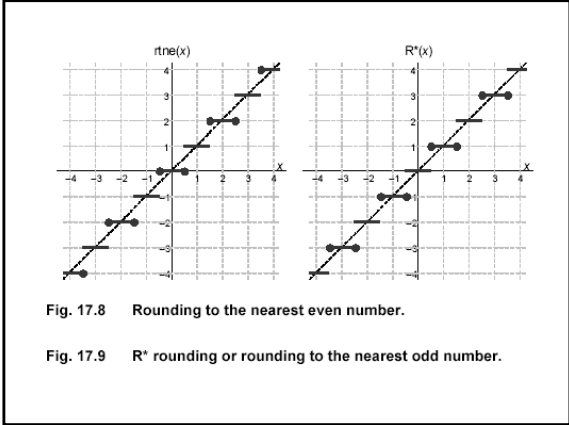
Extended formats for greater internal precision

Single-extended: ≥ 11 bits for exponent
 ≥ 32 bits for significand
bias unspecified, but
exp range $\supset [-1022, 1023]$

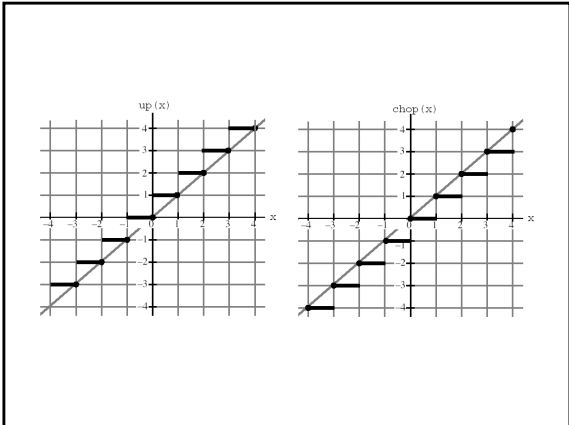
Double-extended: ≥ 15 bits for exponent
 ≥ 64 bits for significand
exp range $\supset [-16382, 16383]$

ANSI/IEEE standard includes four rounding modes:

- Round to nearest even (default mode)
- Round toward zero (inward)
- Round toward $+\infty$ (upward)
- Round toward $-\infty$ (downward)



We may need computation errors to be in a known direction
 Example: in computing upper bounds,
 larger results are acceptable,
 but results that are smaller than correct values
 could invalidate the upper bound
 This leads to the definition of
 upward-directed rounding (round toward $+\infty$) and
 downward-directed rounding (round toward $-\infty$)
 (optional features of the IEEE floating-point standard)



**Polynomial Representation
of the Galois Field
elements**

Evariste Galois (1811-1832)



Evariste Galois (1811-1832)

Studied the problem of finding algebraic solutions for the general equations of the degree ≥ 5 , e.g.,

$$f(x) = a_5x^5 + a_4x^4 + a_3x^3 + a_2x^2 + a_1x + a_0 = 0$$

Answered definitely the question which specific equations of a given degree have algebraic solutions

On the way, he developed **group theory**, one of the most important branches of modern mathematics.

Evariste Galois (1811-1832)

1829 Galois submits his results for the first time to the French Academy of Sciences

Reviewer 1
Augustin-Louis Cauchy forgot or lost the communication

1830 Galois submits the revised version of his manuscript, hoping to enter the competition for the Grand Prize in mathematics

Reviewer 2
Joseph Fourier – died shortly after receiving the manuscript

1831 Third submission to the French Academy of Sciences

Reviewer 3
Simeon-Denis Poisson – did not understand the manuscript and rejected it.

Evariste Galois (1811-1832)

May 1832 Galois provoked into a duel

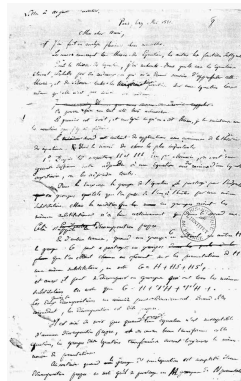
The night before the duel he writes a letter to his friend containing the summary of his discoveries.

The letter ends with a plea:
“Eventually there will be, I hope, some people who will find it profitable to decipher this mess.”

May 30, 1832 Galois is grievously wounded in the duel and dies in the hospital the following day.

1843 Galois manuscript rediscovered by Joseph Liouville

1846 Galois manuscript published for the first time in a mathematical journal



Field

Set F , and two operations typically denoted by
(but not necessarily equivalent to)
 $+$ and $*$

Set F , and definitions of these two operations must
fulfill special conditions.

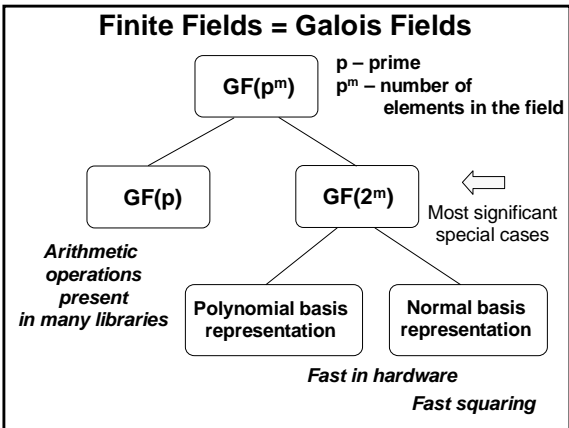
Examples of fields

Infinite fields

{
 R = set of real numbers,
 $+$ addition of real numbers
 $*$ multiplication of real numbers
}

Finite fields

{
set $Z_p = \{0, 1, 2, \dots, p-1\}$,
 $+$ (mod p): addition modulo p ,
 $*$ (mod p): multiplication modulo p
}



Elements of the Galois Field GF(2^m)

Binary representation
(used for storing and processing in computer systems):

$$A = (a_{m-1}, a_{m-2}, \dots, a_2, a_1, a_0) \quad a_i \in \{0, 1\}$$

Polynomial representation
(used for the definition of basic arithmetic operations):

$$A(x) = \sum_{i=0}^{m-1} a_i \cdot x^i = a_{m-1} \cdot x^{m-1} + a_{m-2} \cdot x^{m-2} + \dots + a_2 \cdot x^2 + a_1 \cdot x + a_0$$

- multiplication
- + addition modulo 2 (XOR)

Addition and Multiplication in the Galois Field GF(2^m)

Inputs

$$A = (a_{m-1}, a_{m-2}, \dots, a_2, a_1, a_0) \quad a_i, b_i \in \{0, 1\}$$

$$B = (b_{m-1}, b_{m-2}, \dots, b_2, b_1, b_0)$$

Output

$$C = (c_{m-1}, c_{m-2}, \dots, c_2, c_1, c_0) \quad c_i \in \{0, 1\}$$

Addition in the Galois Field GF(2^m)

Addition

$$A \rightarrow A(x)$$

$$B \rightarrow B(x)$$

$$C \leftarrow C(x) = A(x) + B(x) =$$

$$= (a_{m-1} + b_{m-1}) \cdot x^{m-1} + (a_{m-2} + b_{m-2}) \cdot x^{m-2} + \dots +$$

$$+ (a_2 + b_2) \cdot x^2 + (a_1 + b_1) \cdot x + (a_0 + b_0) =$$

$$= c_{m-1} \cdot x^{m-1} + c_{m-2} \cdot x^{m-2} + \dots + c_2 \cdot x^2 + c_1 \cdot x + c_0$$

- multiplication
- + addition modulo 2 (XOR)

$$c_i = a_i + b_i = a_i \text{ XOR } b_i$$

$$C = A \text{ XOR } B$$

Multiplication in the Galois Field $GF(2^m)$

Multiplication

A \rightarrow $A(x)$

B \rightarrow $B(x)$

C \leftarrow $C(x) = A(x) \cdot B(x) \bmod P(X)$
 $= C_{m-1} \cdot X^{m-1} + C_{m-2} \cdot X^{m-2} + \dots + C_2 \cdot X^2 + C_1 \cdot X + C_0$

$P(x)$ - irreducible polynomial of the degree m

$P(x) = p_m \cdot x^m + p_{m-1} \cdot x^{m-1} + \dots + p_2 \cdot x^2 + p_1 \cdot x + p_0$
