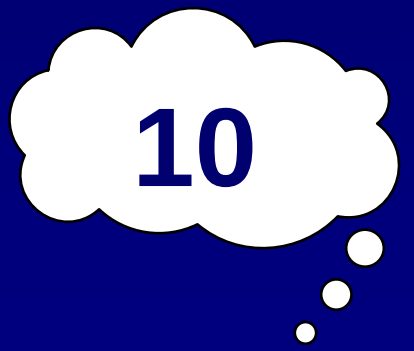
Rexx Arithmetic – inspiration for a Standard

Rexx Symposium – 31 August 2016

Mike Cowlishaw

http://speleotrove.com

mfc@speleotrove.com

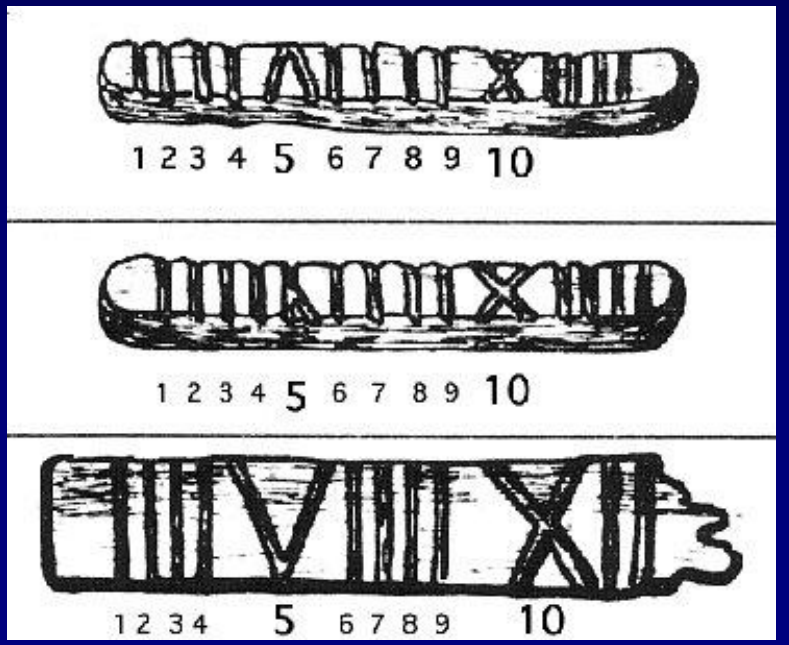


Overview

- Rexx arithmetic
- Status in 2001
  - Software
  - Hardware
  - Standards
- The IEEE 754 standard 2001–2008
  - Current status
- Questions?

Why is decimal arithmetic important?

- Decimal arithmetic represents numbers in base ten, so uses the same number system that people have used for thousands of years



- Pervasive for financial and other commercial applications; often a legal requirement

- 55% of numeric data in commercial databases are decimal (and a further 43% are integers)

The trouble with binary

Decimal:

$$0.1 = 1/10 = 1 \times 10^{-1} = 1E\text{-}1$$

Binary:

$$= 0.0001100110011\ldots$$

$$= 1/16 + 1/32 + 1/256 + 1/512 + 1/4096 + 1/8192 + \ldots$$

Repeated division by 10; what users expect:

Decimal
9
0.9
0.09
0.009
0.0009
0.00009
0.000009
9E-7
9E-8

Repeated division by 10; what they get:

Decimal	float (binary)
9	9
0.9	0.9
0.09	0.089999996
0.009	0.0090
0.0009	9.0E-4
0.00009	9.0E-5
0.000009	9.0E-6
9E-7	9.0000003E-7
9E-8	9.0E-8

Where it costs real money …

- Add 5% sales tax to a $ 0.70 telephone call, rounded to the nearest cent
- 1.05 x 0.70 using binary double is exactly

  0.734999999999999867732370449812151491641998291015625

  (should have been 0.735)
- rounds to $ 0.73, instead of $ 0.74

Rexx arithmetic, 1979-1999

Rexx 1.0 [May 1979]

- 'Minimal' arithmetic – minimum necessary to be useful (loop counters, *etc*.)
- Integers only
- This was a 'holding' implementation, while other parts of the interpreter were designed and implemented

Rexx 1.10 [January 1980]

- Plain decimal arithmetic (no exponents)
- Up to 9 digits after the decimal point
- Precision of result is determined by the more precise of the two terms involved in an operation

  8/3 → 2 8/3.00 → 2.67

- Worked well, but results could often surprise

Rexx 2.50 – the 'new' arithmetic [May – July 1981]

- Developed primarily by e-mail
- Initially controversial (because it changed the behaviour of existing programs)
- Widely discussed and researched (REX *[sic]* was in use in 43 countries by then)
- Essentially the same as the arithmetic in *"The Rexx Language"* book (1985 & 1990)

The choice of arithmetic

- The principle:

"REX arithmetic attempts to carry out the usual operations in as 'natural' way as possible. What this really means is the rules which are followed are those which are conventionally taught in schools and colleges."

(7 Oct. 1981)

Rexx arithmetic

- Full-function finite decimal floating point arithmetic
- Preserves significand length, *etc*. For example,
  1.23 + 1.17 gives 2.40 (not 2.4)
- Integers are a seamless subset of all numbers
- Precision is user-selectable (**numeric digits**)
- Exponents from E-999999999 through E+999999999

ANSI (X3-J18) refinements

- Trigger to exponential notation after 0.000001 (not dependent on DIGITS setting)
- LostDigits condition (raised if input data too precise)
- Input data rounded to DIGITS (not DIGITS+1)
- Published as ANSI X3.274-1996 (see www.rexxla.org), refined through 1999

Decimal arithmetic in 2001

Decimal fixed scale numbers

- Example: 1234.50
- Integer and scale

123450	2

Status in 2001 (software)

- C/C++, PL/I, *etc*: 15-31 digit fixed-scale decimal (*e.g*., 2 digits after the decimal point)
- COBOL: 31 digit fixed-scale decimal
- Databases: 31 or 38 digits, various arithmetics
- Java: unlimited finite floating point decimal (by Sun and IBM)
- C#, VB, *etc.* (Microsoft .Net platform): 28 digit partly-floating-point decimal
- Rexx family: unlimited finite floating point decimal

Status in 2001 (hardware)

- z-Series (IBM S/390): decimal integer instructions (Store-to-Store) used for fixed-scale arithmetic
- Most non-RISC processors (Intel x86, Motorola 68xxx, *etc.*) had decimal adjust instructions to aid decimal integer arithmetic
- In general, decimal arithmetic had to be carried out in software; 100x to 1000x slower than hardware (or worse)

Standards in 2001

- ANSI X3.274-1996 (Programming Language REXX)
  - floating-point arbitrary precision decimals
- IEEE 854-1987 (Radix-Independent Floating-Point Arithmetic)
  - generalization of IEEE 754, to allow for base-10
  - fixed precision

Decimal representations

- Example: 1234.50
- Traditional fixed-scale numbers: integer and scale

123450	2

- Floating point : coefficient and exponent

123450	–2

  – many advantages in the coefficient being an integer

Rationalizing decimal arithmetic

- General Decimal Arithmetic specification:
  - Arithmetic and encodings, suitable for hardware or software implementation
  - Core floating point and integer operations based on ANSI X3-274 (Rexx)
  - Specification extended to comply with IEEE 854
- Open specification; (still) available on the web

IEEE 754 revision, 2001–2008

- Proposal based on Rexx arithmetic
- UC Berkeley key supporters (Prof. Kahan, *etc.*)
- Refined to allow efficient hardware (registers *etc*.)
- By 2005, agreed decimal formats and arithmetic
- Ballot process took another three years …

64-bit Decimal encoding

- Sign, exponent, and coefficient encoded:

- First bit is sign
- Combination field indicates NaN or Infinity, or holds first digit of coefficient and ~two bits of exponent
- Third field has further 8 bits of exponent
- Fourth field has 3x10 bits (3x5 digits) of coefficient (densely packed decimal; DPD)

IEEE 754 decimal formats

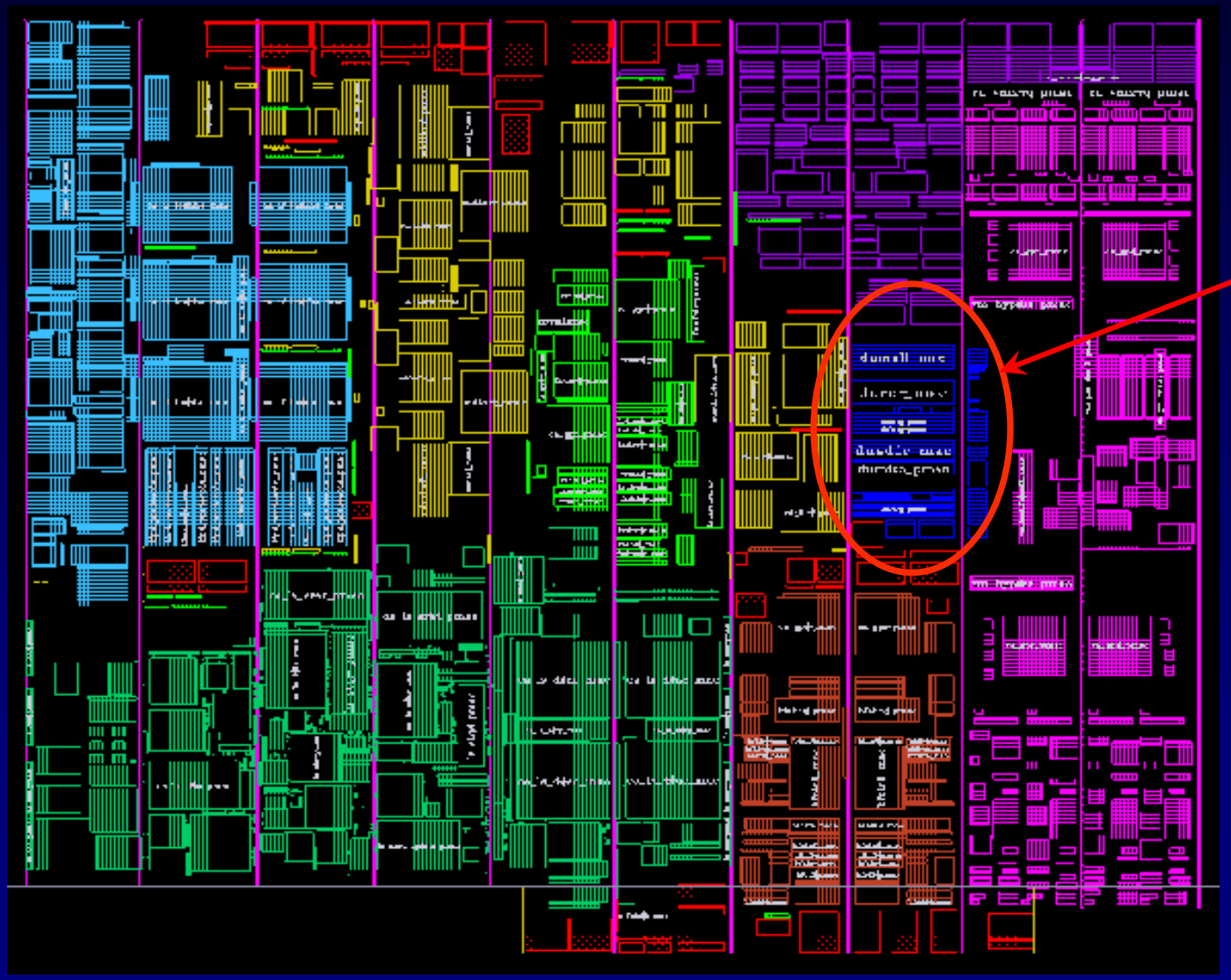
size (bits)	digits	exponent range
32	7	−95 to +96
64	16	−383 to +384
128	34	−6143 to +6144

(range is always larger than same-size binary format)

Benefits of the new types

- Hardware performance
  - Individual operations up to 150x faster than software packages
  - Applications up to 2x faster overall
  - Almost all commercial and financial applications get a performance boost

Power6 processor core layout



DFPU
(Decimal
Floating-
Point Unit)

Benefits of the new types [2]

- Standard data types for decimal bring all the benefits that binary applications enjoy:
  - known, standard, formats, so no conversions (less checking needed at interfaces)
  - faster processing (especially with hardware)
  - well-defined arithmetic, rounding rules, *etc*.
  - safer and easier-to-write applications

IEEE 754 revision, 2015 – 2018?

- Start point IEEE 754-2008 (ISO/IEC 60559:2011)
- Clarifications and minor additions only
- No changes to decimal arithmetic (so far…)
- New Chairman (David Hough); the Editor is still Mike Cowlishaw

Summary

- Decimal data and arithmetic predominate in commercial calculation
- IEEE 754-2008 includes decimal floating-point arithmetic, based on Rexx arithmetic
- Now well-established, in hardware and software
- New revision of IEEE 754 is in progress

Questions?

http://speleotrove.com/decimal