IEEE standard: extended precision I

- It is a format that offers just a little extra precision and exponent
- Motivation for extended precision: some operations benefit from using more digits internally
- Example: some calculators display 10 digits but use 13 internally for calculation
- Example: binary-decimal conversion

  Think about writing/reading numbers to/from files

  When writing a binary number to a decimal number and read it back, can we get the same binary number?

IEEE standard: extended precision II

- Writing 9 digits is enough for short

  Though $10^8 > 2^{24}$, 8 digits are not enough (details not discussed)
- From Section 2.1.2 of Goldberg [1990], in reading the 9-digit number, if extended precision is available, an efficient algorithm can be done so that the original binary representation is recovered (details not shown)
- 17 digits for double precision (proof not provided).

  Example:

  numbers in a data set from Matrix market:

IEEE standard: extended precision III

```
> tail s1rmq4m1.dat
  8.2511736085618438E+01  2.513452865992495
 -6.0042951255041466E+00  8.659944220661552
  1.0026197619563723E+01 -1.313683766184450
 -1.5108331040361231E+01  5.142317399695508
 -1.1690286345961363E+03  1.625072665580781
  8.2511736074473220E+01  1.510833104036122
```

- Matrix market:

  http://math.nist.gov/MatrixMarket/

  A collection of matrix data

IEEE standard: exactly rounded operations I

- Operations: IEEE standard requires results of addition, subtraction, multiplication and division exactly rounded.
- Exactly rounded: an array of words or floating-point numbers, expensive
- Goldberg [1990] showed using 2 guard digits and one sticky bit the result is the same as using exactly rounded operations

  Only little more cost

IEEE standard: exactly rounded operations II

- Reasons to specify operations

  run on different machines $\Rightarrow$ results the same
- IEEE: square root, remainder, conversion between integer and floating-point, internal formats and decimal are correctly rounded (i.e. exactly rounded operations)
- IEEE does not require transcendental functions to be exactly rounded
- Transcendental numbers:

IEEE standard: exactly rounded operations III

e.g., exp, log

- Reason: cannot specify the precision because they are arbitrarily long

Special quantities I

- On some computers (e.g., IBM 370) every bit pattern is a valid floating-point number
- For IBM 370, $\sqrt{-4} = 2$ and it prints an error message
- IEEE : `NaN`, not a number

  Thus not every bit pattern is a valid number
- Special value of IEEE:

  $+0, -0$, denormalized numbers, $+\infty$, $-\infty$, NaNs (more than one NaN)
- A summary

Special quantities II

Exponent	significand	represents
$e = e_{\min} - 1$	$f = 0$	$+0, -0$
$e = e_{\min} - 1$	$f \neq 0$	$0.f \times 2^{e_{\min}}$
$e_{\min} \leq e \leq e_{\max}$		$1.f \times 2^{e}$
$e = e_{\max} + 1$	$f = 0$	$\pm\infty$
$e = e_{\max} + 1$	$f \neq 0$	NaN

- Why IEEE has NaN

  Sometimes even $0/0$ occurs, the program can continue
- Example: find $f(x) = 0$, try different x's, even $0/0$ happens, other values may be ok.

Special quantities III

- If $b^2 - 4ac < 0$

$$\frac{-b + \sqrt{b^2 - 4ac}}{2a}$$

  returns NaN

  $-b+$ NaN should be NaN

  In general when a NaN is in an operation, result is NaN

- Examples producing NaN:

Special quantities IV

Operation	NaN by
$+$	$\infty + (-\infty)$
$\times$	$0 \times \infty$
$/$	$0/0, \infty/\infty$
REM	x REM 0, ∞ REM y
$\sqrt{}$	$\sqrt{x}$ when $x < 0$

Infinity I

- $\beta = 10, p = 3, e_{\max} = 98$, $x = 3 \times 10^{70}$,
  x^2 overflow and replaced by 9.99×10^{98}??
  In IEEE, the result is ∞
- Note $0/0 =$ NaN, $1/0 = \infty, -1/0 = -\infty$
  $\Rightarrow$ nonzero divided by 0 is ∞ or $-\infty$
  Similarly, $-10/0 = -\infty$, and $-10/-0 = +\infty$
  (± 0 will be explained later)
- $3/\infty = 0, 4 - \infty = -\infty, \sqrt{\infty} = \infty$
- How to know the result?
  replace ∞ with x, let $x \to \infty$

Infinity II

Example:

$$\frac{3}{\infty}: \quad \lim_{x \to \infty} 3/x = 0$$

If limit does not exist $\Rightarrow$ NaN

- $x/(x^2+1)$ vs $1/(x+x^{-1})$

  $x/(x^2+1)$: if x is large, x^2 overflow, $x/\infty = 0$ but not $1/x$.

  $1/(x+x^{-1})$: x large, $1/x$ ok

  $1/(x+x^{-1})$ looks better but what about $x=0$?

  $x = 0$, $1/(0+0^{-1}) = 1/(0+\infty) = 1/\infty = 0$

Infinity III

- If no infinity arithmetic, an extra instruction is needed to test if $x = 0$. This may interrupt the pipeline