Unifying Tests for Square Root

Michael Parks
Sun Microsystems

Real Numbers and Computers 7
Nancy, France
July 10 to 12, 2006

Outline

- Machine arithmetic
- Rounding boundaries
- Test strategy – all modes simultaneously
- Hensel's Recurrence
- Software speed comparison
- P-adic arithmetic

Verification

- Test the accuracy of FSQRT(x) subroutine
- Huge input space (10^9 single, 10^19 double)
- Correct answer is not obvious
- Need large number of test cases
- Previously: Tang, K.C. Ng, W. Kahan ('88-'94), Parks
- Now:
  - all 4 modes folded together
  - order of magnitude faster C implementation

IEEE 754 Arithmetic

- Fixed precision n
- Representation = (sign, exponent, significand)
- Value $\pm 2^e \, m$
- Range $2^{n-1} \le m \le 2^n - 1$
- Ignored: 0, denormals, infinity, NaN
- Partition the line into categories
  - Representable
  - Non-representable

Rounding modes

$$e = \lfloor \log_2 s \rfloor$$

$$trunc\,(s) = 2^{e-n+1} \lfloor s / 2^{e-n+1} \rfloor$$

$$near\,(s) = 2^{e-n+1} \left\lfloor s / 2^{e-n+1} \right\rceil$$

$$inf\,(s) = 2^{e-n+1} \lceil s / 2^{e-n+1} \rceil$$

$$(minf = trunc)$$

Normalize

- Scale $2^n \sqrt{x} = \sqrt{4^n x}$

- Should be enforced by good FSQRT algorithm design
- Implies equivalence classes:

$$\ldots \sim \frac{x}{16} \sim \frac{x}{4} \sim x \sim 4x \sim 16x \sim \ldots$$

- Algebra made easier by assuming

$$2^{2n-2} < x < 2^{2n}$$

$$2^{n-1} < \sqrt{x} < 2^n$$

Table-Maker's Dilemma

- Seeking cases for which

$$\sqrt{x} = N + .00001\ldots$$
$$\sqrt{x} = N + .01111\ldots$$
$$\sqrt{x} = N + .10000\ldots$$
$$\sqrt{x} = N + .11111\ldots$$

- FSQRT depends only upon on precision

Test strategy

- Land close to the rounding boundaries
- Seek integer x with
- Mode near: $\sqrt{x} \approx y \pm 1/2$
- Mode trunc: $\sqrt{x} \approx z$

- By scaling, y and z are integers too
- Data for round-down, round-up same as truncation

Derivation, 1

- Fundamental integer range $[2^{n-1}, 2^n - 1]$
- The test argument should take the form

$$x = 2^{n-1} R$$

$$x = 2^n R$$

- with R fundamental
- Binary form of x has n-1 or n trailing zeros

Derivation, 2

- Formulate boundary cases
- For nearest mode, want x so that $\sqrt{x} = y \pm (\frac{1}{2} - \epsilon)$
- For directed, want $\sqrt{x} = z \pm \epsilon$
- Tiny deflection

$$\epsilon > 0$$

Derivation, 3

- For directed modes, seek

$$\sqrt{x} = z \pm \epsilon$$

$$x = z^2 \pm 2\,\epsilon\, z + \epsilon^2$$

$$z^2 - x = \pm 2\,\epsilon\, z - \epsilon^2 = k$$

- Fractional terms eliminated
- Purely integer equation

Derivation, 4

- Recast the problem: Given k, find R and z with

$$z^2 - k = 2^n R$$

- Restrictions
- z and R in the fundamental range $[2^{n-1}, 2^n - 1]$
- This process yields test case x
- Small k means a good test case, since sqrt(x) is near z
- Larger k means sqrt(x) lies farther from the boundary

Derivation, 5

- Directed modes set up already
- For nearest mode, seek

$$\sqrt{x} = y \pm (\frac{1}{2} - \epsilon)$$

$$4\,x = (2\,y \pm 1)^2 - k$$

$$4\,x = z^2 - k$$

$$4\,x = 2^{n+2}\,R = z^2 - k$$

- Same problem as previous, with a scaling factor

Derivation, 6

- Transformed the problem into simple integer equations
- The user chooses a small k, and must find (z, R) pairs

$$z^2 - k = 2^{n-1} R$$

$$z^2 - k = 2^{n} R$$

$$z^2 - k = 2^{n+1} R$$

$$z^2 - k = 2^{n+2} R$$

Derivation, 7

- Hensel's technique
  - > Choose a small k
  - > Solve each in turn

$$z_1^2 - k = 2^1 R_1$$

$$z_2^2 - k = 2^2 R_2$$

- Pairs (z, R) for each j
- Each z has j bits
- No R gets very big

$$z_n^2 - k = 2^n R_n$$

Derivation, 8

- Equivalently, solve

$$z^2 \equiv k\,(2^j)$$

- Each solution admits two more

$$\bar{z} = 2^{j-1} + z$$

$$\tilde{z} = 2^{j-1} - z$$

Derivation, 9

- Hensel's Recurrence

$$z_j^2 \equiv k \, (2^j)$$

$$z_3 = 1,\ R_3 = (1-k)/8$$

$$for\ \ j = 4, \ldots n$$

$$R_{j-1}\ even:$$

$$z_j = z_{j-1}$$

$$R_j = R_{j-1}/2$$

$$else$$

$$z_j = 2^{j-2} - z_{j-1}$$

$$R_j = 2^{j-4} + (R_{j-1} - z_{j-1})/2$$

Test summary

- A little algebra puts the test arguments in the right range
- Produces correctly rounded results y and z too
- Test requirements are minimal
- basic operations (add, mult by 2, shift)
- integer ops, but no multi-precision
- inner loop costs O(n), in return ~ 4 test cases
- Guaranteed to find data x and results y or z
- Loop: increase k
- Tests performed in descending order of difficulty

Example

- Single precision n = 24 bits, about 7 digits
- Take k = 1 and run Hensel's Recurrence
- Results in a test point: x = 2^{23} * 8388610
- s = sqrt(x) exactly
- = 8388608.9999994... // base 10
- = 10----0.11111111111111111111110... // base 2
- Excellent test case: 24 consecutive ones past the point
- Provably closest to a rounding boundary

Implementation in C language

- User selects:
- #define PRECISION
- #define TEST_NEAREST, TEST_DIRECTED
- #define MAXK
- Computes:
  - all test data x for |k| < MAXK in sequence
  - correctly rounded results (y and z) too
- Compare to FSQRT(x) for correctness
- Repeats billions of times

Comparison

- Existing UCBTEST implementation sqrtest.c, nearest only
- New implementation fastsqrtest.c handles all modes

Mode	sqrtest.c	fastsqrtest.c
Nearest	6443 sec	560 sec
Directed	-	688 sec
All modes	-	750 sec

- UCBTEST/sqrtest.c by K.C. Ng works fine, but lacks the normalizing factor and resorts to multi-precision arithmetic
- Also available dirmultest.c, dirdivtest.c by Parks

Software

- Test programs for
  - FMUL directed
  - FDIV directed
  - FSQRT near and directed
- Sun Legal working on it

Core: p-adic arithmetic

- Hensel defined

$$x = \sum_{k=N}^{\infty} x_k \, p^k$$

$$|x|_p = \frac{1}{p^N}$$

$$d_p(x, y) = |x - y|_p$$

$$\mathbb{Q}_p = completion\ of\ \mathbb{Q}\ using\ metric\ d_p$$

Core: p-adic arithmetic

$$|x|_p = \frac{1}{p^N}$$

$$\mathbb{Q}_p \; completion \; of \; (\mathbb{Q}, d_p)$$

$$\mathbb{Q}_p \; is \; a \; field$$

$$\mathbb{Z}_p = ring \; of \; p-adic \; integers$$

$$\mathbb{Z}_p = \{ x \in \mathbb{Q}_p : |x|_p \leq 1 \}$$

Hensel Lifting in p-adic Arithmetic

- Restated, if

$$f(Z) = \sum_{k=0}^{M} a_k Z^k \qquad f'(z) = unit$$

$$f(z) \equiv 0 \left(2^j\right) \qquad u = \left(f'(z)\right)^{-1}$$

- then

$$z' = z - u\, f(z) \qquad f(z') \equiv 0 \left(2^{j+1}\right)$$

Hensel Lifting in p-adic Arithmetic

- Newton's method converges in the p-adic metric, constructing a 2-adic integer solution to

$$z^2 \equiv k \ (2^n)$$

- 1 bit at a time, from right to left
- Unify: No surprise that test data for modes are related
- Further expedient, compute data for n = 53 from n = 24

Motto

Seek singularities!

There lie all the errors.

-W. Kahan

Results

... if any.

- Nothing found yet:
  - SPARC
  - AMD Opteron
  - Itanium
- FMA and formal methods are helping get the last bit right

References

- Companion to
  - W. Kahan, “A Test for Correctly Rounded Square Root”, Computer Science Dept, Berkeley, 1994
  - M. Parks, “Number-Theoretic Test Generation for Directed Rounding”, IEEE Trans. Computers, 2000
  - P. Tang, “Testing Computer Arithmetic by Elementary Number Theory”, Argonne National Lab, 1989
  - K.C. Ng, notes within UCBTEST below
- Software
  - UCBTEST, http://www.netlib.org/fp

Conclusions

- Fold near and directed tests together compactly
- Re-derive Hensel's Recurrence with a unified vision
- Significant speed improvement in C
- Check conformance to IEEE Standard 754
- Verification, if run long enough
- P-adic arithmetic
- Software release

Merci beaucoup!

- Appreciated

```
ieee754 @ yahoo.com
          gmail.com
          ...
```