The Loosest Slots In Reno

Processors for consumer applications promise outstanding single-precision floating-point performance for scientific applications at commodity prices. However, depending on the specific operations and data values, limited precision in floating-point arithmetic can lead to significant loss of accuracy. Even double precision is not safe, but single precision is often sufficient. A big payoff can be had by speculatively using the lowest viable precision and resorting to more expensive higher precisions only in the relatively rare cases when that fails to deliver sufficient accuracy.

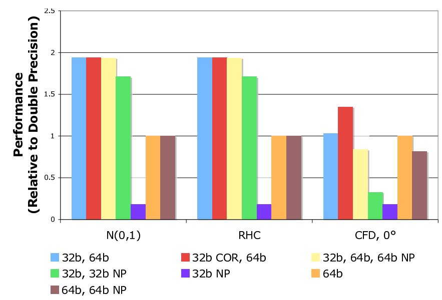
Speculative Precision. The ability to inexpensively detect (potential) loss of accuracy at run time is the enabling technology that allows the programmer or compiler to use *speculative precision.* No general-purpose method in known by which loss of accuracy can be cheaply detected; however, the special case of summation is commonly used and prone to loss of accuracy due to rounding and cancellation. Although rounding can slowly eat away at accuracy, cancellation is less predictable and often more extreme.

We suggest that a PEAK EXPONENT REGISTER should be incorporated into floating point units to warn of loss of precision due to cancellation. When $c = a+b$ is computed and cancellation occurs, the difference between the exponent of c and the exponent of a is the number of bits that canceled, assuming a is the addend with the larger magnitude. The peak exponent register stores the largest addend exponent seen during a summation. After computing the sum, software can check whether cancellation has occurred using the difference between the peak exponent register and the sum exponent. In a sample FPGA implementation, the peak exponent register requires a trivial amount of hardware and has no impact on clock cycle time of the FPU.

What The Slots Show. The “slot machines” in our exhibit demonstrate just how often single precision is sufficient in computing summations by showing how single, double, and 304-bit arbitrary precision arithmetic sum various different data sets. The data sets are modeled after those commonly used to test accuracy of summation algorithms:

- The first 32 terms of the Taylor series for $e^{-2\pi}$
- *Heavy Cancellation* sums $1 + ... + 1 + 1 \times 10^{-18} + 1 \times 10^{-18} - 1 - ... - 1$
- 32 *Uniform Spacing* values between 1.0 and 2.0
- $N(0,1)$adds Gaussian random numbers with $\mu = 0, \sigma = 1$
- *Inverse Square* is $\sum_{i=1}^{32} \frac{1}{i^2}$
- *Random Heavy Cancellation* is $\sum_{i=1}^{32} \pm 10^{x_i}$, where x_i is Gaussian with $\mu = 0, \sigma = 35$, but clipped to [-35, +35]

Briefly pulling the lever sums the current type of sequence; holding the lever down for 5s or longer steps to the next type of sequence. See for yourself how often single precision is sufficient to give 5 decimal digits of accuracy.



Speedup of speculative precision over double

How Well It Works In General. As the above graph shows, it works very well – with as much as 2× speedup over using native double precision... and double precision was not always sufficient. For the random data sets, $N(0,1)$ and *Random Heavy Cancellation*, 32-bit speculation failures are rare, so even recomputing the failed cases using software *native pair* arithmetic yields good speedup. A real “worst case” summation data set came from a computational fluid dynamics (CFD) code, but even in that case speculation still delivered modest improvements over directly using a higher precision.

How To Make It Work Better. The native pair method for improving accuracy can be accelerated significantly by augmenting a floating point unit with a simple RESIDUAL REGISTER. Like the PEAK EXPONENT REGISTER, this has minimal impact on the hardware complexity, speed, or power consumption, making it a much more efficient solution than implementing higher precisions directly in hardware.

This document should be cited as:

```
@techreport{sc07peakexp,
author={William Dieter and Raymond P. LeBeau and
Henry Dietz},
title={The Loosest Slots In Reno},
institution={University of Kentucky},
address={http://aggregate.org/WHITE/sc07peakexp.pdf},
month={Nov}, year={2007}}
```