Beyond FLOPS: Why Exact Arithmetic is Faster

James Douglas Boyd
Founder and CEO/CTO, SciSci Research

SciSci's AI chip, the exact processing unit (EPU), will deliver exact precision with a $\sim 15\times$ speed gain per operation per ALU over floating-point. Here, I'll explain why the EPU will be so fast.

Micro-operations required to perform IEEE standard floating-point (FP) arithmetic, such as rounding and shifting, impose an overlooked "drag" on execution. Replacing floating-point with a new architecture based on exact arithmetic, SciSci's ISA eliminates this micro-operation drag and benefits from novel parallelization per operation.

Micro-Operation "Drag"

One might think of an operation (e.g., addition) as the smallest unit of processor execution, but it isn't. With the FP format, even arithmetic as rudimentary as addition requires a series of supporting micro-operations (μ-ops). The IEEE standard encodes floats using a sign, exponent, and mantissa (i.e., significands). Addition requires several micro-operation "steps": exponents are compared and subtracted; mantissas are compared and shifted; results are normalized and rounded. That's a high-level description; the finer details are hairier. Generously, let's say that FP addition involves five steps: exponents, mantissa shifts, mantissa addition, normalization, and rounding.

How do we know that μ-ops impose "drag"? Operations with fewer μ-ops are faster. FP multiplication on AMD is 133% faster than addition because it requires fewer μ-op steps; it doesn't require mantissa shifts. If we say, crudely, that FP addition involves five μ-op steps and multiplication involves four, we would expect multiplication to enjoy a 125% efficiency gain, a slight underestimate. To take another example, fixed-point arithmetic only requires (in the best case) 3 steps (adjusting scaling factors, arithmetic, and rounding) so we would expect $\sim 167\%$ efficiency over FP, and, indeed, INT8 is $\leq 180\%$ more efficient than FP8. So, we'll presume conservatively that operations with N% fewer μ-op steps are $\frac{100}{100-\mathrm{N}}\%$ faster.

Beyond Floating-Point: Faster Operations

Suppose we removed all of the μ-op "drag" from arithmetic. It's easy enough to imagine. Consider addition on integers in binary: $2 + 4 = 6$ is simply $100 + 010 = 110$. SciSci's ISA gives short, exact encodings directly in binary, without the exponent/mantissa formatting of FP, without the scaling that fixed-point arithmetic requires, and without rounding. Arithmetic is direct. Furthermore, the ISA enjoys novel parallelization on exact encodings that make arithmetic even faster.

How Much Faster?

Let's say we're doing arithmetic that requires rounding in FP. Sometimes, mantissas don't need to be shifted, and sometimes normalization isn't needed, so we can say that FP requires between 3 and 5 μ-ops per operation. Likewise, EPU parallelization can range between $2\times$ and $3\times$. So, the upper bound advantage is $3\times$ EPU parallelization against an FP operation requiring all 5 nontrivial μ-ops: a **15× speed advantage**.

How about below the upper bound? As an empirical exercise, let's count the number of operations performed for a fixed number of μ-ops, say, 500. FP needs 3-5 μ-ops per operation, and the EPU can deliver 2-3 operations per μ-op thanks to exact parallelization. We draw randomly from these possibilities 500 times and count the number of completed operations. Running this combinatorial test $50,000$ times, one can compare the distributions of completed operations for FP and EPU (which are Gaussian). The average number of operations for EPU exact arithmetic was 1250 ± 124 operations, and the average number of operations for FP was 144 ± 14 operations. So, within two standard deviations, the EPU performance gain is between **6×** and **13×**. That's per operation per ALU; then, one can multiply over tens of thousands of ALUs...

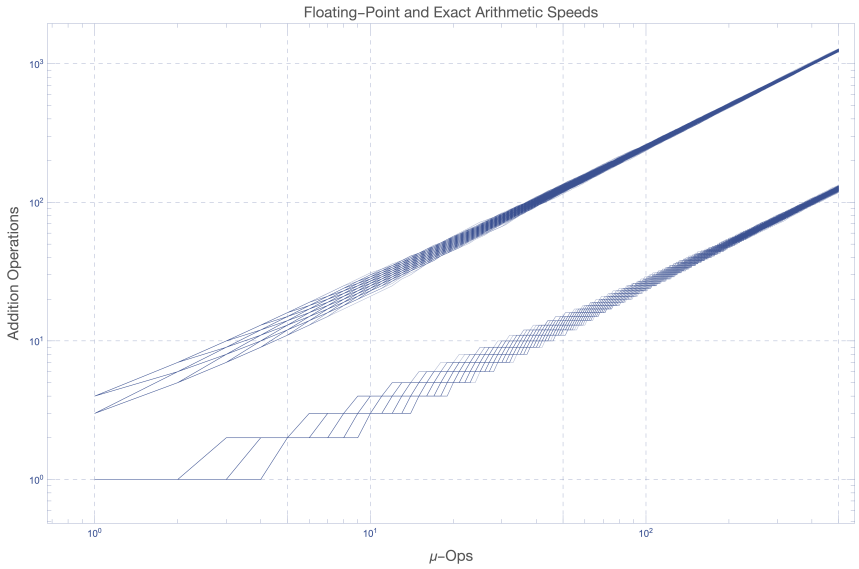

Figure 1: A comparison of cumulative number of completed operations per μ-op on a log-log plot. The EPU trajectory is above and the FP trajectory is below.

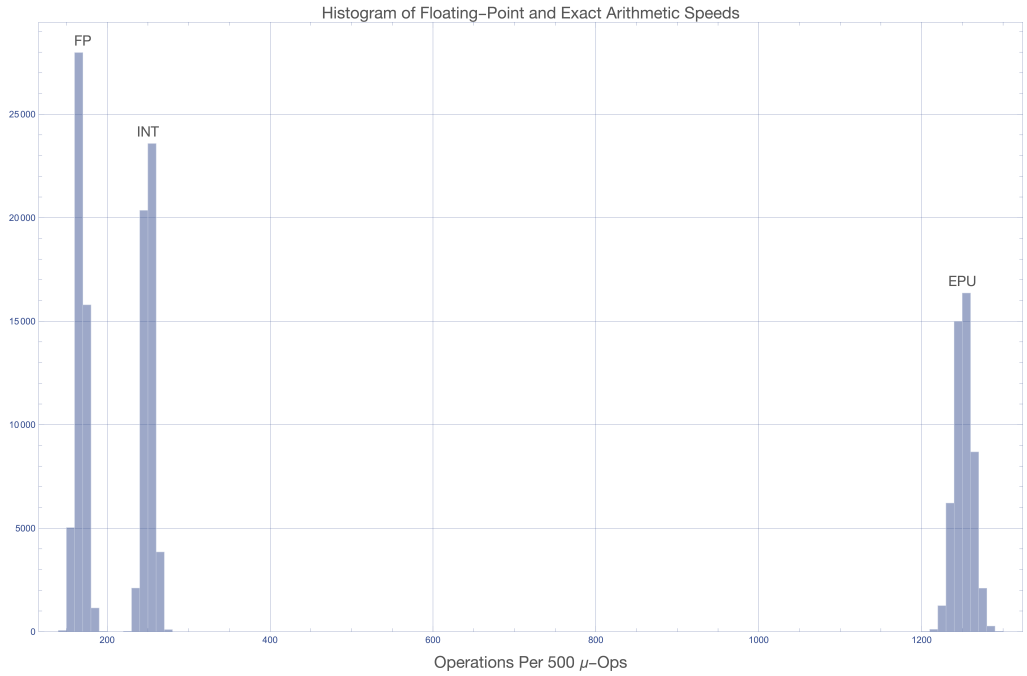

Figure 2: A histogram comparison of the distribution of completed operations after 500 μ-ops, drawn from 50,000 samples. FP and EPU are compared, as well as fixed-point arithmetic ("INT"), which is modeled as requiring 1-3 μ-ops per operation.