Wanted: Floating-Point Add Round-off Error Instruction

Marat Dukhan Richard Vuduc Jason Riedy

School of Computational Science and Engineering
College of Computing
Georgia Institute of Technology

June 23, 2016

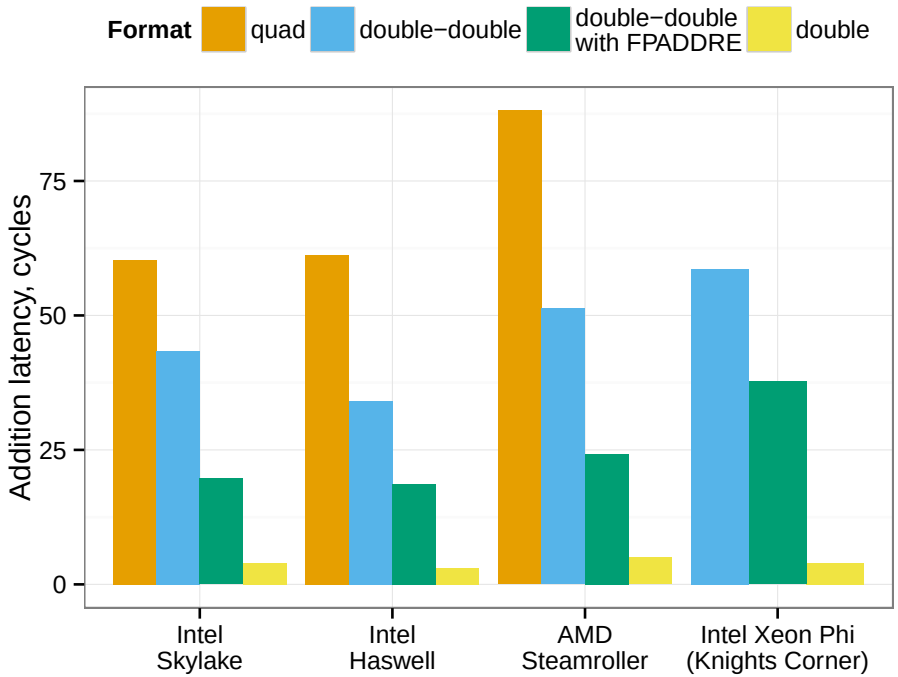
Outline

High-Precision Arithmetic in High Demand

- Numerical reproducibility
  - Dynamic work distribution across threads
  - Variations in SIMD- and instruction-level parallelism
- Mathematical functions
  - IEEE754-2008 recommends correct rounding for LibM functions
- Growing number of scientific applications
  - David Bailey's presetations: 8 areas in (2005): 8 areas of science
  - His recent presentation on SC BoF (2014): 12 areas of science

High-Precision Arithmetic Algorithms

- Quadruple precision
  - Software implementation using integer arithmetic
- Double-double arithmetic
  - Represent a number as an unevaluated sum of two doubles: $x = x_{hi} + x_{lo}$
- Compensated algorithms
  - High-precision summation, dot product, polynomial evaluation



Outline

Error-Free Multiplication

$$p + e = a \cdot b$$

where

$$p = \text{double}(a \cdot b)$$

Error-Free Multiplication with FMA

```
p := FPMUL a * b
e := FMA a * b - p
```

Error-Free Addition

$$s + t = a + b$$

where

$$s = \text{double}(a + b)$$

Error-Free Addition (Knuth, 1997)

```
s := FPADD a + b
b_virtual := FPADD s - b
a_virtual := FPADD s - b_virtual
b_roundoff := FPADD b - b_virtual
a_roundoff := FPADD a - a_virtual
e := FPADD a_roundoff + b_roundoff
```

FPADD3 Instruction

Ogita et al (2005) suggested FPADD3 instruction to accelerate Error-Free Addition.

- FPADD3 adds 3 floating-point numbers without intermediate rounding
- No general-purpose CPU or GPU ever implemented this instruction

Error-Free Addition with FPADD3 (Ogita et al, 2005)

```
s := FPADD a + b
e := FPADD3 a + b - s
```

FPADDRE Instruction

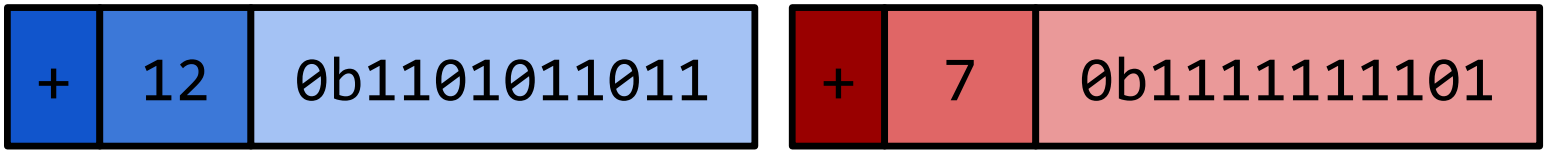
We suggest an instruction, Floating-Point Add Round-off Error (FPADDRE) to compute the roundoff error of floating-point addition. The instruction offers two benefits for error-free addition:

- Replace 5 FPADD instructions with 1 FPADDRE
- Break dependency chain between the sum and the roundoff error

Error-Free Addition with FPADDRE

```
s := FPADD a + b
e := FPADDRE a + b
```

Reusing FPADD logic in FPADDRE



Mantissa alignment	+ 11101011011 11111101101
Summation	11110111010111101
Rounding bit	± 1
FPADD(a, b)	11110111011
FPADDRE(a, b)	01101

Schema of FPADD and FPADDRE operations (the case of operands with the same sign and overlapping mantissas). The operations differ only in two aspects: addition or subtraction of a sticky bit and the bits copied to the resulting mantissa.

Outline

Simulation

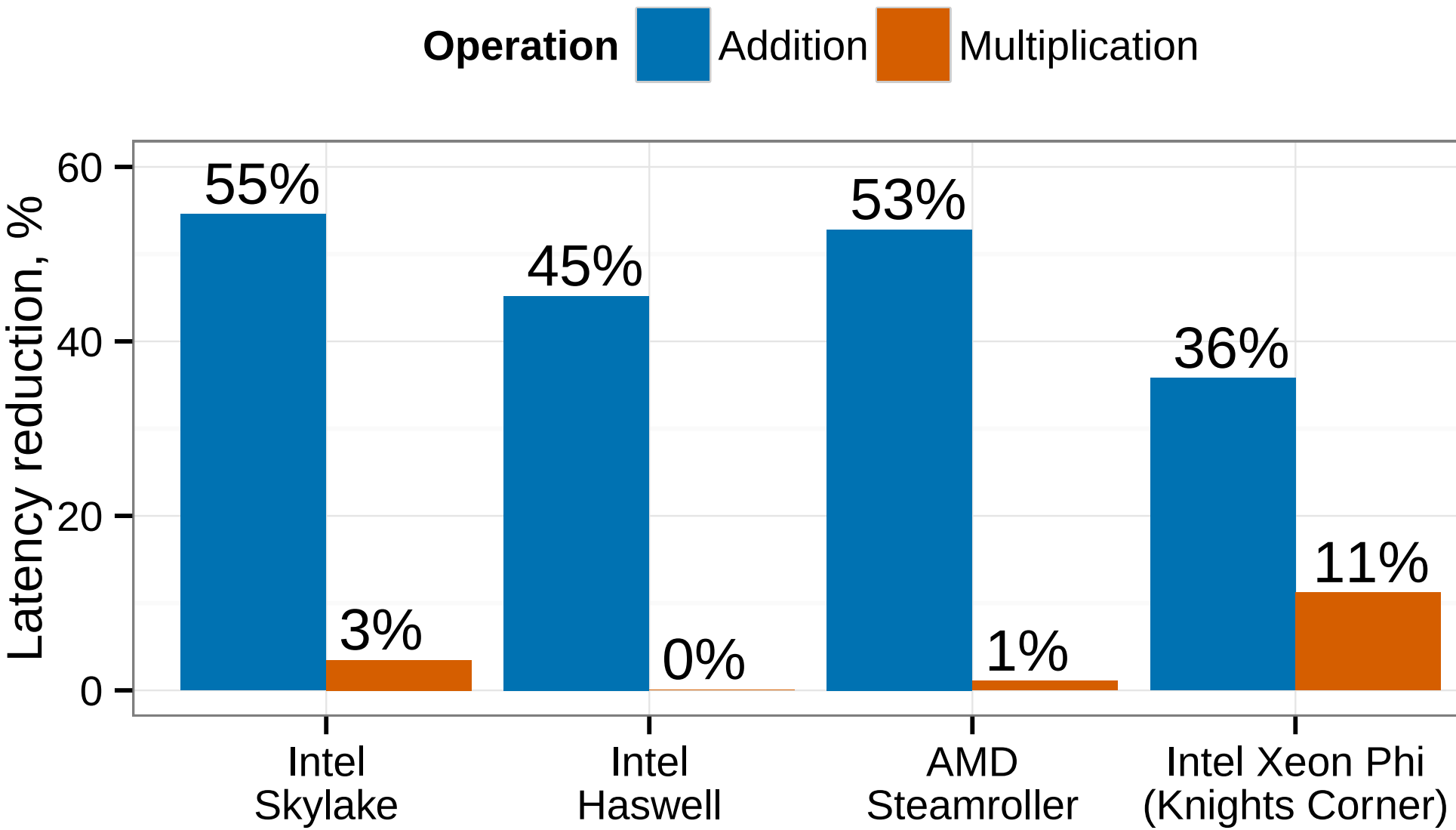
To estimate performance effect of the FPADDRE instruction, we implemented a several of high-precision algorithms:

- Double-double scalar addition and multiplication
- Double-double matrix multiplication
- Compensated dot product
- Polynomial evaluation via compensated Horner scheme

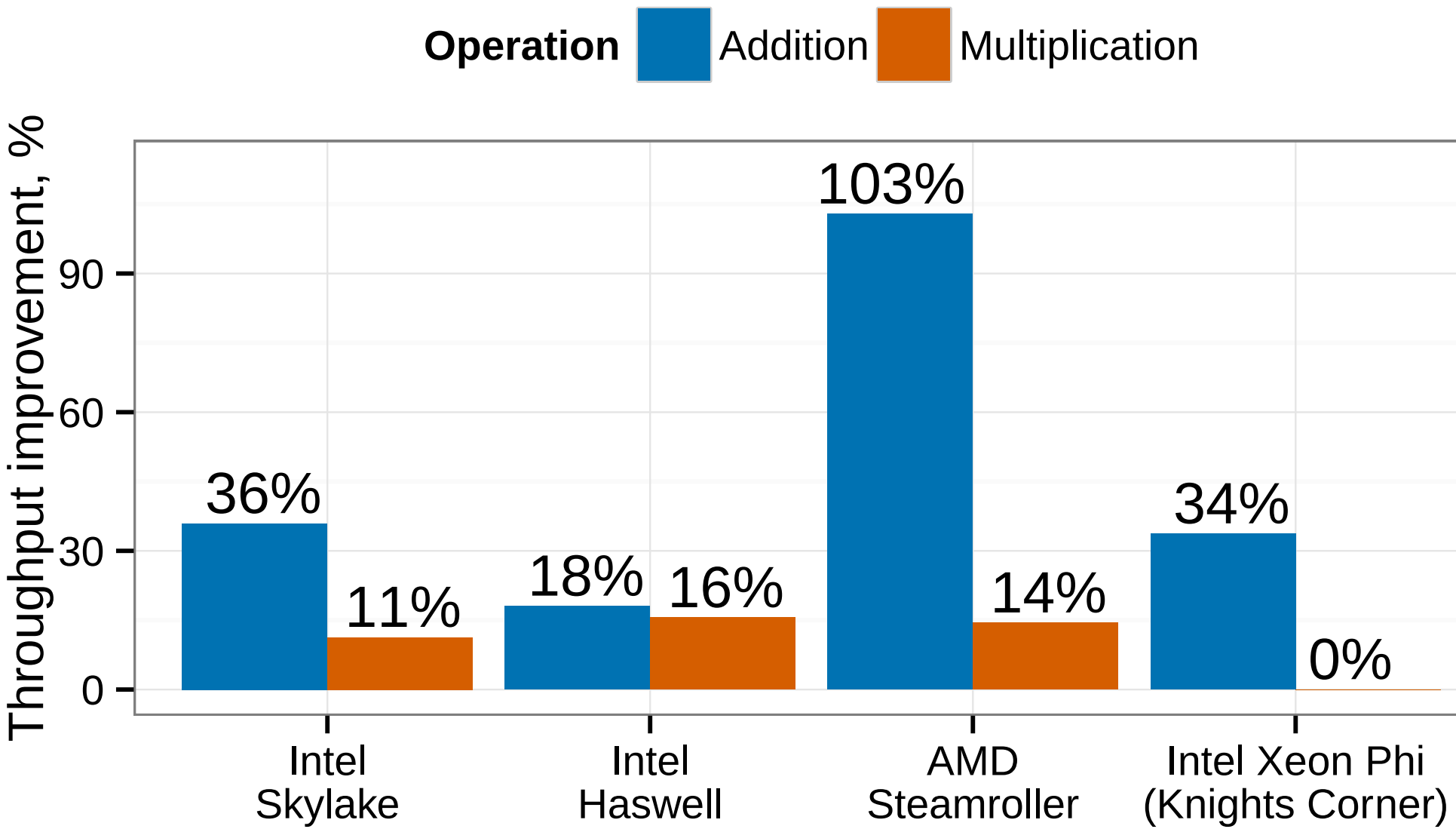
Then we replaced FPADDRE with an instruction with performance characteristics of addition and benchmarked the algorithms on four microarchitectures:

- Intel Haswell
- Intel Skylake
- AMD Steamroller
- Intel Knights Corner co-processor

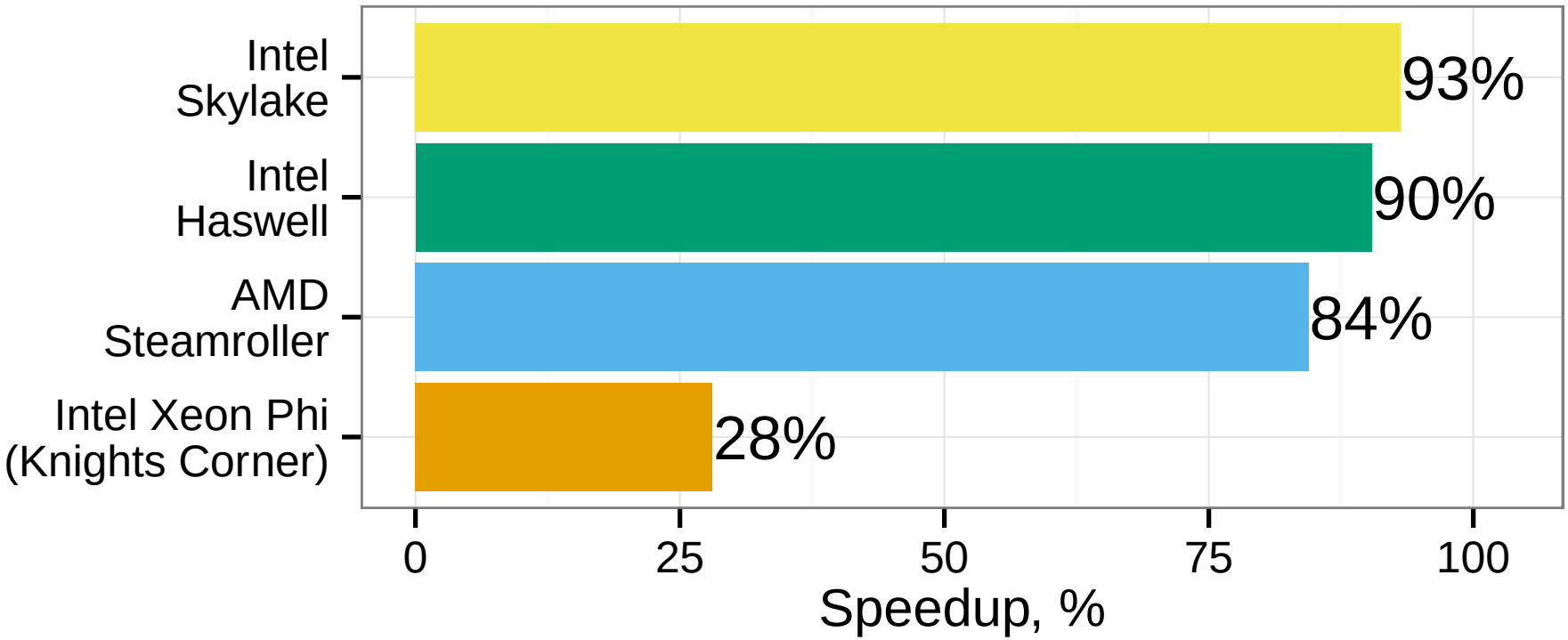
Double-double Latency



Double-double Throughput

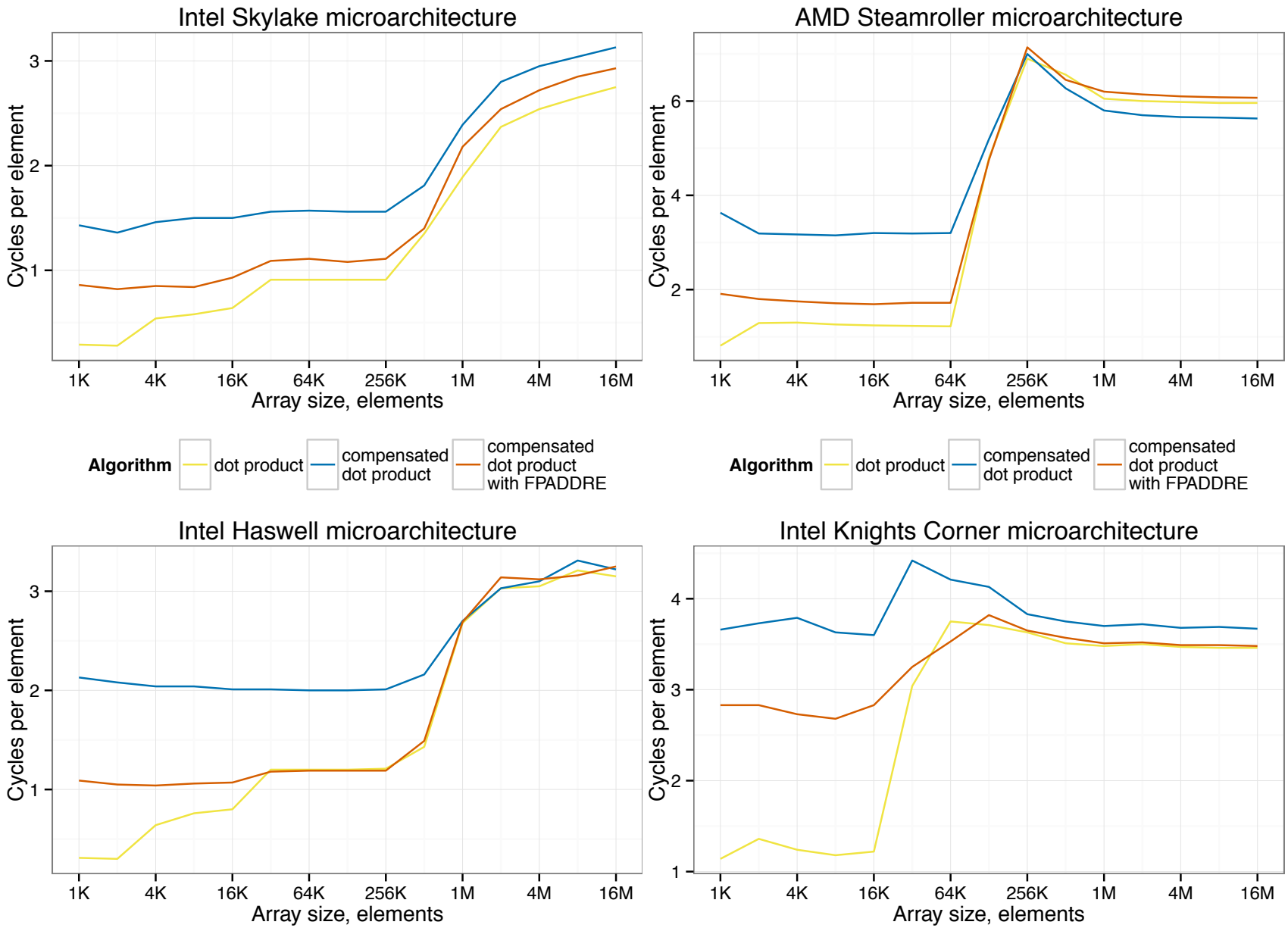


Double-double Matrix Multiplication

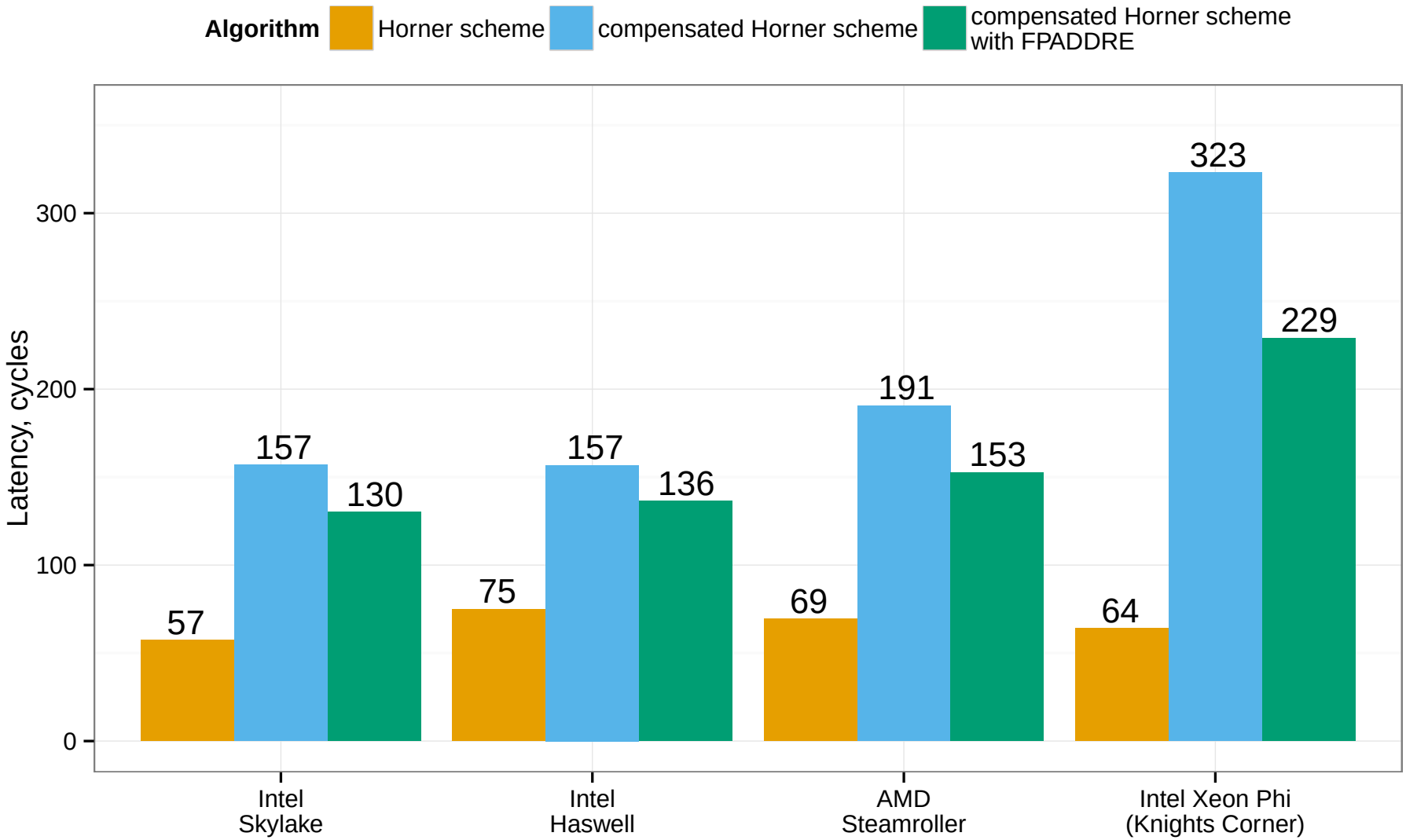


Double-double matrix multiplication acceleration with FPADDRE instruction

Compensated Dot Product



Compensated Polynomial Evaluation



Public release

We open-sourced the software which was deloped as a part of this research

- The implementation, unit tests, and benchmarks, are available at github.com/Maratyszcza/FPplus
- The paper preprint is on arxiv.org/abs/1603.00491

Summary

- We suggest a new instruction, Floating-Point Add Round-off Error, to compute the roundoff error of floating-point addition
- Performance simulations suggest that the proposed instruction could accelerate high-precision computations by up to 2x

Funding

This research was supported in part by

The National Science Foundation (NSF) under NSF CAREER award number 1339745.

The U.S. Dept. of Energy (DOE), Office of Science, Advanced Scientific Computing Research under award DE-FC02-10ER26006/DE-SC0004915.

Defense Advanced Research Projects Agency (DARPA) under agreement #HR0011-13-2-0001

Declaimer

Any opinions, conclusions or recommendations expressed in this presentation are those of the authors and not necessarily reflect those of NSF, DOE, or DARPA.