

# Design Exploration for Decimal Floating-Point Arithmetic IBM University Partnership Program Proposal

Professor Michael J. Schulte  
University of Wisconsin-Madison

IBM Technical Sponsor:  
Dr. Eric Schwarz  
IBM Systems and Technology Group

## 1. Technical Areas of Research

Computer Architecture/System Design, Digital Circuit Design, Servers and Embedded Systems.

## 2. Description of Project

Commercial applications and databases typically store numerical data in decimal format. Currently, however, microprocessors do not provide instructions or hardware support for decimal floating-point arithmetic [1]. Consequently, decimal numbers are often read into computers, converted to binary numbers, and then processed using binary floating-point arithmetic. Results are then converted back to decimal before being stored. Besides being time-consuming, this process is error-prone, since most decimal numbers cannot be exactly represented as binary numbers [2,3]. Thus, if binary floating-point arithmetic is used to process decimal data, unexpected results may occur after a few computations [4].

In many commercial applications, including financial analysis, banking, tax calculation, currency conversions, insurance, accounting, and e-commerce, the errors introduced by converting between decimal and binary numbers are unacceptable and may violate legal accuracy requirements. Therefore, these applications often use software to perform decimal floating-point arithmetic. Although this approach eliminates errors due to converting between binary and decimal numbers, it leads to long execution times for numerically intensive commercial applications, since software implementations of decimal floating-point operation are typically 100 to 1,000 times slower than equivalent binary floating-point operations in hardware [5]. Due the growing importance of decimal floating-point arithmetic, specifications for it have recently been added to the draft revision of the IEEE 754 Standard for Floating-Point Arithmetic [6].

For this project, we will research, design, and evaluate instruction set extensions, arithmetic algorithms, and hardware designs for decimal floating-point arithmetic. We will also enhance computer simulators and compilers to better evaluate the benefits of hardware support for decimal floating-point arithmetic. This research will include:

- **Investigating decimal floating-point instruction set extensions.** We will select these instructions based on (a) their usefulness in reducing the execution times of commercial applications, (b) their ability to meet the requirements of the revised IEEE 754 Standard, and (c) their anticipated ease of implementation on high-performance processors. To estimate the execution times of current decimal floating-point software, we will use the decNumber C library [7], commercial benchmarks, the SimpleScalar simulator [8], and the *prof* and *gprof* profiling tools.
- **Developing and analyzing novel arithmetic algorithms and hardware designs.** We will investigate the use of specialized encodings, hardware sharing, parallel speculative result computation, carry-free decimal addition, and other techniques to improve the area, critical path delay, performance, and/or power consumption of decimal floating-point arithmetic. We will also

investigate micro-architecture modifications that need to be made to allow high-performance processors to support decimal floating-point instructions. Our preliminary research in this area has led to novel designs for decimal adders [9], multipliers [10], and multioperand adders [11].

- **Modeling, simulating, and synthesizing promising hardware designs.** We will model hardware designs for decimal floating-point arithmetic using Verilog and VHDL. These designs will be simulated to verify correct performance, and then synthesized, placed, and routed to obtain area, delay, and power estimates. Similar estimates will be made for high-speed binary floating-point arithmetic units using the same technology to gain a better understanding of the relative cost of decimal floating-point arithmetic. Furthermore, theoretical models, similar to those presented in [12,13] will be developed to estimate the area, critical path delay, and power consumption of our proposed designs in a technology independent manner. For certain designs, such as decimal floating-point adders, we will also perform custom layout to take advantage of high-speed and low-power circuit techniques.
- **Measuring the performance benefits of the designs.** To obtain accurate estimates of potential performance improvements, we will run commercial benchmarks on enhanced versions of the SimpleScalar [14] and PHARMSim [15] simulators, which will be modified to simulate processor support for decimal floating-point arithmetic. We will also extend the gcc compiler to support decimal data types and operations in order to test large-scale commercial applications.

### 3. Project objectives and goals

The goals of this research are to:

- Develop efficient arithmetic algorithms and hardware designs for decimal fixed-point and decimal floating-point arithmetic
- Determine useful architecture and microarchitecture extensions for decimal arithmetic
- Evaluate the potential costs and benefits of decimal arithmetic
- Provide an improved understanding of the fundamental similarities and differences between binary and decimal floating-point arithmetic
- Examine the potential usefulness of decimal arithmetic for non-commercial applications
- Develop simulator and compiler extensions for decimal floating-point arithmetic
- Provide challenging and exciting research opportunities related to decimal arithmetic for a diverse group of undergraduate and graduate students
- Assist researchers at IBM to develop support for decimal floating-point arithmetic on future IBM processors

We anticipate that this research will help answer a variety of important questions, including:

- What are efficient algorithms and hardware designs for implementing decimal floating-point arithmetic?
- What decimal arithmetic algorithms scale well with higher frequencies and larger operand sizes?
- What intermediate encodings are most efficient for processing decimal data?
- What micro-architecture enhancements are necessary to allow decimal floating-point arithmetic to be implemented on future microprocessors?
- What instruction set extensions for decimal floating-point arithmetic are likely to be most effective at improving the performance of commercial applications?
- What are the relative costs in terms of area, performance, and power dissipation of decimal floating-point arithmetic units compared to binary floating-point arithmetic units?
- What is the potential speedup from implementing decimal floating-point arithmetic in hardware?

Our long-term vision for this research is that it will lead to efficient hardware support for decimal floating-point arithmetic on future IBM microprocessors.

#### **4. Long term impact to the information/computing industry and IBM**

While working on this project, we will interact closely with Eric Schwarz, Mike Cowlshaw, Mark Erle, and other experts from IBM to help ensure that the results from our research are useful in the design of future IBM processors. Our interactions with experts at IBM have already begun, since I am currently advising Mark Erle on his Ph.D. dissertation on decimal floating-point arithmetic, while he works full-time at IBM with Eric Schwarz as his IBM mentor. Mark's research has already led to two papers in major IEEE conferences [10,16] and it is anticipated that his Ph.D. dissertation will provide an important basis for the design of future decimal arithmetic hardware. We have sent Eric Schwarz and Mike Cowlshaw copies of our early publications on decimal floating-point arithmetic and will continue to do so as part of this project.

For this project, we will also be working with a diverse group of top-notch undergraduate and graduate students from the University of Wisconsin-Madison on topics related to decimal arithmetic. I have a strong record of mentoring and advising members of under represented groups and have received awards and support for these efforts from the National Society of Black Engineers, the Alfred P. Sloan Foundation, and the National Science Foundation. Similar mentoring and advising will be incorporated into this project. In addition to the graduate student to be directly supported on this project, we anticipate that several other students will contribute to this project through senior design projects, honors projects, independent studies, Master's projects or theses, and a summer research program. Our hope is that these students will have opportunities to work at IBM as interns, co-ops, and full-time hires to apply the experienced gained through this research project. To facilitate this, we will continue to send Dr. Schwarz resumes of our best students and explore the possibility of establishing an IBM speed team to research and develop decimal arithmetic hardware.

This project will help guide efficient implementations of decimal floating-point arithmetic on future IBM microprocessors. Anticipated research contributions include

- A better understanding of the role of decimal arithmetic in commercial applications and of the potential performance benefits due to decimal floating-point hardware
- Instruction set extensions for decimal floating-point arithmetic that can be incorporated into the design of future microprocessors
- Novel arithmetic algorithms, hardware designs, and micro-architecture enhancements for decimal floating-point arithmetic
- Enhanced simulators and compilers for evaluating the performance impact of decimal floating-point arithmetic
- An improved understanding of the fundamental differences between binary and decimal floating-point arithmetic

Processor support for decimal floating-point arithmetic will improve the accuracy and performance of commercial applications, and will allow more numerically intensive commercial applications to be developed. Results from this research have the potential to benefit several business-related fields including financial analysis, banking, insurance, accounting and e-commerce, which make up an important customer base for IBM. If our project goals are successful, this research could help fundamentally change the way computers process decimal data. Since IBM is the clear industry leader in high-end processors for commercial applications, this research project is expected to provide significant direct benefit to IBM by helping them to maintain their competitive advantage over other companies in this area.

## References:

- [1] M. F. Cowlishaw, "Decimal Arithmetic FAQ: Part 3 – Hardware Questions," IBM Corporation, 2003. Available at <http://www2.hursley.ibm.com/decimal/decifaq3.html>.
- [2] W. D. Clinger, "How to Read Floating Point Numbers Accurately," *Proceedings of the ACM SIGPLAN 1990 Conference on Programming Language Design and Implementation*, pp. 92-101, June 1990.
- [3] R. G. Burger and R. K. Dybvig, "Printing Floating-Point Numbers Quickly and Accurately," *Proceedings of the ACM SIGPLAN 1996 Conference on Programming Language Design and Implementation*, pp. 108-116 1996.
- [4] M. F. Cowlishaw, "General Decimal Arithmetic," IBM Corporation, 2004. Available at <http://www2.hursley.ibm.com/decimal/>.
- [5] M. F. Cowlishaw, "Decimal Floating-Point: Algorithm for Computers," *Proceedings of the 16th IEEE Symposium on Computer Arithmetic*, pp. 104-111, June 2003.
- [6] Institute of Electrical and Electronics Engineers, inc., *Draft IEEE Standard for Floating-Point Arithmetic*, The Institute of Electrical and Electronics Engineers, Inc., New York, 2004. Working draft available at <http://754r.ucbtest.org/drafts/754r.pdf>.
- [7] M. Cowlishaw, "The decNumber C library," IBM Corporation, 2003. Available at <http://www2.hursley.ibm.com/decimal/>.
- [8] T. Austin, E. Larson, and D. Ernst, "SimpleScalar: An Infrastructure for Computer System Modeling," *IEEE Computer*, pp. 59-67, vol. 35, no 2, February, 2002.
- [9] J. Thompson, N. Karra, and M. J. Schulte, "A 64-bit Decimal Floating-Point Adder," *Proceedings of the IEEE Computer Society Annual Symposium on VLSI*, pp. 297-298, February, 2004. Extended version available at <http://mesa.ece.wisc.edu>.
- [10] M. A. Erle and M. J. Schulte, "Decimal Multiplication Via Carry-Save Addition," *Proceedings of the IEEE International Conference on Application-Specific Systems, Architectures, and Processors*, pp. 348-358, June, 2003.
- [11] R. D. Kenney and M. J. Schulte, "Multioperand Decimal Addition," *Proceedings of the IEEE Computer Society Annual Symposium on VLSI*, pp. 251-253, February, 2004. Extended version available at <http://mesa.ece.wisc.edu>.
- [12] V. G. Oklobdzija, B. R. Zeydel, H. Dao, S. Mathew, and R. Krishnamurthy, "Energy-Delay Estimation Technique for High-Performance Microprocessor VLSI Adders," *IEEE Symposium on Computer Arithmetic*, pp. 272-279, June 2003.
- [13] M. J. Schulte, P. I. Balzola, A. Akkas, and R. W. Brocato, "Integer Multiplication with Overflow Detection or Saturation," *IEEE Transactions on Computers*, vol. 49, pp. 681-691, July, 2000.
- [14] T. Austin, E. Larson, and D. Ernst, "SimpleScalar: An Infrastructure for Computer System Modeling," *IEEE Computer*, pp. 59-67, vol. 35, no 2, February, 2002.
- [15] H. W. Cain, K. M. Lepak, B. A. Schwartz, and M. H. Lipasti, "Precise and Accurate Processor Simulation," *5th Workshop On Computer Architecture Evaluation Using Commercial Workloads*, February 2002
- [16] M. A. Erle, M. J. Schulte, and J. G. Linebarger, "Potential Speedup with Decimal Floating-Point Hardware," *Proceedings of the Thirty Sixth Asilomar Conference on Signals, Systems, and Computers*, 1073-1077, November, 2002.