UCLA Computer Science Department: CS 252A
Arithmetic Algorithms and Processors
Winter 2019, MW 10:00 - 11:55 am, Boelter 4283
Office hours: M2:00 - 3:00pm or by appointment

Prof. Miloš D. Ercegovac
Engineering 6 468B
825-5414; e-mail: milos@cs.ucla.edu
http://www.cs.ucla.edu/˜milos

Course material: Lecture viewgraphs, papers, handouts, etc. posted on **CCLE Course.**
Textbook: M.D. Ercegovac and T. Lang, *Digital Arithmetic*, Morgan and Kaufmann, 2004.
Related resources: Books, journals, conferences, Web sites, simulators - listed below.
Grading: Design explorations, reports and presentations 30%, a survey of papers related to your project and presentation 20%, project 50%.

Updated: December 15, 2018

COURSE DESCRIPTION

Computer arithmetic has always played an important and critical role in the design of general-purpose and special-purpose processors. It has been an active research area since the early days of computers (1950s), albeit relatively small compared to other areas in computer architecture and computer science. Computer arithmetic, in brief, investigates theoretical/hardware/software aspects in number representation, arithmetic algorithms for basic operations, function evaluation, and fixed-point and floating-point arithmetic. Computer arithmetic is a synergy of applied mathematics, algorithms, digital design, VLSI implementation, software and compilers, and applications. Every new generation of processors, such as GPUs, super-scalar and multi-core processors, digital signal processors, and, most recently, processors for AI and machine learning, rely on progress in computer arithmetic to increase performance and reduce power and energy, and cost.

After discussing fundamentals, this course will focus on arithmetic techniques relevant to several research areas of growing importance:
A1. Domain-specific accelerators and composite arithmetic,
A2. Architectures for machine learning and neural networks,
A3. Approximate computing and variable-precision arithmetic.

Class style will be interactive and participatory. I will mostly use the blackboard and show slides as outlines and visuals. My blackboard notes will be included with lecture slides. Lecture slides are based on the textbook and will be posted. Textbook readings and practice exercises will be suggested (solutions provided). You should work on these exercises in groups - no submission expected. Lectures will take 90 minutes, leaving 20 minutes for solving problems and discussing your questions.

PLAN

Activity 1: Weeks 1 - 4: Fundamentals

The fundamentals of computer arithmetic discussed in the class will include : (i) number representation systems, and (ii) algorithms and design of basic operations (addition, multiplication, division, and floating-point addition and multiplication).

Activity 2: Weeks 2 - 6; Design Explorations

Design explorations will help you master some of the key aspects of arithmetic design:
DE-1: Design and compare two 64-bit adders. Pick two out of the following list: Carry-lookahead adder, Parallel prefix adder, Carry-skip adder, Carry-select, and Conditional-sum adder. *Due end of the 2nd week.*
DE-2: Design and compare radix-4 24-bit it right-to-left (LSBF) and left-to-right array (MSBF) multiplier. *Due end of the 4th week.*
DE-3: Design a 24-bit radix-4 SRT divider with prescaling. Do two approaches: sequential and combinational (i.e., unfolded sequential) and compare. *Due end of the 6th week.*

For each design exploration, you will write a report describing (a) Design and simulation of arithmetic schemes [your choice of tools]; and (b) Performance evaluation for different parameters and discussion. You will also make a short class presentation.

Activity 3: Weeks 4 - 7: Study of Arithmetic in Areas A1 - A3

In this activity we will study two research papers from each of these areas, identify arithmetic operators and performance issues, and discuss relevant arithmetic algorithms and designs. You should form teams of two students to study papers, prepare short summaries, 15-minute presentations, and class discussions. Note that the objective is to study issues related to arithmetic, not the research areas themselves. The papers you choose may relate to your project. A selection of papers will be posted on CCLE site. You may also select your own papers related to these areas.

Activity 4: Weeks 6 - 10: Projects

The project (individual or team) should be related to computer arithmetic. You will propose a project topic, carry out the proposed research, and produce a conference-type research paper as report (8 pages). In your project proposal do: (a) Define the problem you propose to investigate and prepare a short bibliography of 2-4 most relevant papers to study in detail for comparisons with your project. (b) Discuss the work and results described in the related papers and what do you expect your original contribution will be. (c) Describe your approach in achieving project objectives. (e) Describe how will you evaluate and compare your results. The choice of tools is yours.

In this period we will have:

- Consultations with me regarding projects during office hours or by appointment. Discussions in class on projects in progress.

- Project presentations and report drafts detailed enough to allow peer reviews (ready by the end of Week 8). The project presentations will be during Weeks 9 and 10.
- Final project reports are due March 20, 2019. A tentative organization of the project paper:

  0 - Project title; names; summary.
  1 - Introduction (problem statement, project objectives, related work).
  2 - Your approach in solving the problem.
  3 - Main body of your work.
  4 - Results.
  5 - Evaluation and discussion of results. Comparisons with related work.
  6 - Future work. What would you do differently if you are to do it again?
  7 - References.

RESOURCES

Books on Digital Arithmetic

- *Digital Arithmetic*, M.D. Ercegovac and T. Lang, Morgan and Kaufmann, 2004. (Main textbook for this course)
- *Division and Square Root: Digit-Recurrence Algorithms and Implementations*, M. Ercegovac and T. Lang, Kluwer Academic Publishers, 1994.
- *Handbook of Floating-Point Arithmetic*, J.M. Muller et al., Birkhauser, Boston, 2009. **Floating-Point Arithmetic.**
- *Elementary Functions, Algorithms and Implementations*, J.M. Muller, 3rd. Edition, Birkhauser, 2016. **Elementary Functions.**
- *Seminumerical Algorithms*, D.E. Knuth, Addison Wesley, 1997. A classic work.
- *I-64 and Elementary Functions*, P. Markstein, HP Press, 2002.

IEEE Conference Proceedings Available on IEEE Xplore

ACM Conference Proceedings Available on ACM Portal

- **IEEE Symposium on Computer Arithmetic** - the principal resource for arithmetic.
- Asilomar Conference on Circuits, Systems and Computers
- IEEE Application-Specific Architectures and Processors (ASAP)
- SPIE Conference on Algorithms and Implementations for Signal Processing
- International Conference on Computer Design (ICCD)
- Conferences on FPGAs: FPGA, FPL
- Conferences such as ISCA covering neural network architectures, memistors, and QCAs

Journals and Magazines

- **IEEE Transactions on Computers** (in particular, Special Issues on Computer Arithmetic)
- IEEE Journal on Solid State Circuits
- IEEE Transactions on VLSI
- IEEE Transactions on Nanotechnologies
- Journal of VLSI Signal Processing
- IEE Proceedings: Digital Techniques
- IEEE Micro
- IEEE Signal Processing Magazine

Web sites

- **UCLA Digital Arithmetic and Reconfigurable Architectures Lab - M. D. Ercegovac**. (needs updating - still a useful resource)
- **Stanford Architecture and Arithmetic Group - M. Flynn**.
- **Arithmetic Group, Ecole Normale Superieure, Lyon, France - J. M. Muller**.
- **IEEE Floating-Point Standard - W. Kahan**.
- **IEEE Floating-Point Standard - D. Hough**.
- **Arithmetic Group, IRISA-CAIRN - A.Tisserand**.

Tools, libraries, platforms, and simulators

- **Arithmetic Core Generator (FloPoCo) - F. DeDinechin**.
- **FloPoCo - Source Code**
- **SOAP - Structural Optimization of Arithmetic Programs - G. Constantinides**.
- **VHDL Libraries: adders with standard cells - R. Zimmermann**.
- **Arithmetic Module Generator - T. Aoki**.
- **Computer Arithmetic Algorithms Simulator - I. Koren**.