CSE565M: Acceleration of Algorithms in Reconfigurable Logic

Introduction

Anthony Cabrera

FL24::L01

Washington University in St. Louis

Table of contents

1. Administrative Stuff

2. Class Structure

3. Introductory Teaser

Administrative Stuff

Administrivia

- Instructor: Anthony Cabrera
  - Office: Virtual
  - Office Hours: TBD
- TAs: None
  - Your classmates are your own TAs! More on this later
- Please use Piazza over email for asking questions
  - Emails get lost in the pile, Piazza posts don't
  - Public post for general questions, private if needed
- Course Webpage: TBD
  - I haven't yet decided if I'm going to use my original course page or just stick to Canvas

Textbook

- Parallel Programming for FPGAs by Kastner, Matai, and Neuendorffer
  - This textbook is open-source (i.e., free!)
  - I will provide builds periodically if I decide to change anything or fix typos
  - To track updates in real-time, my fork of the textbook repo lives here

Class Logistics

- Lectures
  - Room Location: Hillman Hall 70
  - Planned recordings TBD
  - Some guest lectures planned
- My office hours
  - Definitely will hold at least one virtual office hour per week
- TA Office Hours
  - No TAs for now

Class Structure

Class Workload

- Programming Assignments/Labs
  - Aiming for 3-4 labs of programming/HW design this later)
  - Teams of 2-3 allowed, but more expected of them
  - Smaller studio assignments that should be completed alone
- Class Participation
- Student Presentations
  - Present once during semester
  - Subject is on an article – conference or journal – in the literature
- Final Project Proposal
  - Proposal due before spring break
- Final Project and Presentation
  - again, teams of 2-3

Grading Logistics

- Five elements
  - Programming Assignments/Labs
  - Class Participation
  - Paper Presentation
  - Final Project Proposal
  - Final Project
- Grading:

Assigments	40%
Class Participation	10%
Paper Presenation	10%
Final Project Proposal	10%
Final Project	30%

Programming Assignments/Labs

- Programming
  - We'll use C/C++ for both SW and HW (using HLS tools)

A note on building hardware

Be aware that the hardware development tools take take a long time to build
Could be on the order of hours!

- Programming Assignments
  - I plan on assigning small *studio*-flavored assignments to further introduce content covered in lecture/the textbook
- Labs
  - These will be larger assignments that aim to holistically synthesize multiple concepts from lecture/the textbook
  - Expect 3-4 of these in the semester

Compute Resources

- Builds will (hopefully) happen on New England Research Cloud
    - Otherwise, they'll happen on McKelvey Linux cluster
- Execution will happen on Open Cloud Testbed (shared resource in Boston, MA)
    - 24 AMD Alveo U280 FPGAs

Before our Next Class

Create an account on the Open Cloud Testbed following instructions 1.1 - 1.2 here: https://github.com/cabreraam/OCT-Tutorials/tree/master/cloudlab-setup

Create an account at NERC for the build resource following instructions here: https://github.com/cabreraam/OCT-Tutorials/blob/master/nercsetup/nerc-vm-guide.md

Class Participation

Discussions with your fellow classmates:

- I expect that you will be interacting with each other to help troubleshoot issues, whether that's in person, on Piazza, GitHub, or Canvas discussions.
    - For example, if you discover a neat HLS concept, programming trick, or productivity tool that someone might benefit, this would make for a great post on Canvas discussions.
- Since my availability is more restricted, I will be relying on y'all to help each other. More importantly though, **this type of collaboration is also what you can expect in a research-focused work environment** No individual has all of the answers but everyone has the same, limited hours in the day. Let your classmates help you troubleshoot your issues!
- For full marks, I will need to see a record of these interactions toward the end of the semester

Student Presentations

- Subject – related topic from the literature
    - Nominally 20-25 min in length
    - I will provide several candidate papers as options
    - You can choose one of your own
        - Journals – ACM TRETS
        - Conferences – FCCM, FPGA, FPL, ASAP
    - Talks about tools are also OK
    - Send me a (private?) note on Piazza to discuss options
- Each student will do *one* presentation
    - Before talk, submit slides

Signing up for Presentations

- First opportunity - September 9
- I will schedule slots for talks from September 23 through the end of the semester
  - Incentive to go early, I'll be gentler in assessment
- The goal for interleaving student presentations with my lectures is to hopefully help seed ideas for your final projects
- Volunteer via public Piazza post (TBD)
  - I'll make sure to nail down how I want to do this soon
  - OK to claim date and then identify subject later
  - must email me the paper first OK subject with me
  - Watch class webpage for claimed dates, subjects
  - Submit your preferred date due to me by September 30
    - I will assign a date if no date requested by then

Ask the Audience

- Quick poll: how many of y'all have written LaTeX?
  - In-class response: 7
- Quick poll: how many of y'all are comfortable with LaTeX?
  - i.e., it would not detract from learning the content in this class
  - In-class response: 3
- LaTeX is not required for this class, but if you're an undergrad/graduate student thinking about a research-flavored career that involves publishing conference/journal articles – e.g., industry labs, nat'l labs, academia – this might be a good opportunity to learn
  - That being said, this is not a LaTeXclass; the goal is to learn reconfigurable hardware design, *NOT* typesetting!

Project Proposal (in lieu of midterm)

- Propose a project that uses concepts from this class as core components
- Example ideas
  - accelerating a particular application/domain of applications
  - building tools to help facilitate reconfigurable hardware design [1]
  - performance/benchmarking [2]
  - formal methods for reconfigurable hardware design

Project Proposal (cont.)

- The proposal should include
  - Introduction/Motivation
  - Some background motivation
    - Related work need not be all-inclusive; I expect that as your classmates present papers, you'll likely have more literature to draw from
  - proposed work
    - elaborate on the scope of the work
    - what tools do you need?
    - what do you aim to do and how long will this take? (you'll encounter this at work, no doubt!)
    - Could be in the form of a Gantt chart but not required
- What's the point of doing this?
  - It's like a design document in industry
  - or a grant proposal for funding research

Class Bibliography

- Expect to contribute to a class bibliography via `.bib` file on GitHub
  - Even if you don't use LaTeX, it'll still be handy to have a bibliography of the papers covered in lecture, by your peers, and of interest to you.

Questions?

- Webpage: TBD

Final Project

- Code artifacts
  - I would encourage you to host this publically unless there's good reason not to, e.g., it involves work currently intended for submission to a conference, journal or the like
- Presentation
  - Expect to create a research poster
  - Each team will get an opportunity to give a 10 minute poster presentation (displayed on the projector) with 5 minutes for questions
- Report should read like mini-paper: coherent sentences, graphs, explanations of graphs, conclusions, etc.
  - Error bars, is Gaussian error reasonable assumption?, how many experiment replications?, etc.
  - Code snippets OK, no need to provide entire listing
  - Expectation of making code publicly available on GitHub

Academic Integrity

- Collaboration is encouraged!
- What is over the line?
  - Working in a group larger than 2-3
  - Showing your work to another group
  - Internet usage
    - Finding sources, ideas, examples, open source libraries - all OK
    - Copying text, ideas, code - not OK
    - I will allow ChatGPT, but not for the assignments/labs.

Introductory Teaser

What is Reconfigurable Logic?

- Field-Programmable Gate Array (FPGA)
  - Array of logic gates
  - Programmable in the field
  - Custom logic on a chip
- Enables hardware design
  - Custom data path
- Challenge is how to architect design
  - HW has many degrees of freedom

This Class

- FPGAs exist
    - but they frequently don't perform well
- Why is that?
    - need to understand performance drivers
- What can we do about it?
    - Need to be able to come up with "good" designs
    - Need to be able to tune performance

References i

- H. Ye, C. Hao, J. Cheng, H. Jeong, J. Huang, S. Neuendorffer, and D. Chen.
  Scalehls: A new scalable high-level synthesis framework on multi-level intermediate representation.
  In *2022 IEEE International Symposium on High-Performance Computer Architecture (HPCA)*, pages 741–755, 2022.
- H. R. Zohouri, N. Maruyama, A. Smith, M. Matsuda, and S. Matsuoka.
  Evaluating and optimizing opencl kernels for high performance computing with fpgas.
  In *SC '16: Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*, pages 409–420, Nov 2016.