Course Syllabus

ECE 5997 Hardware Accelerator Design and Automation

Fall 2021, October 15th – December 7th

---

1. Course Information

Lectures: Friday 04:40-05:55pm
OH/Tutorial: Tuesday 04:40-5:55pm
Instructor: Zhiru Zhang, zhiruz@cornell.edu
Credits: 1 Credit

Course Texts:

◦ R. Kastner, J. Matai, and S. Neuendorffer, Parallel Programming for FPGAs, arXiv, 2018.

2. Course Description

Targeted specialization of functionality in hardware has become arguably the best means to achieving improved compute performance and energy efficiency for a plethora of emerging applications. Unfortunately, it is a very unproductive practice to design and implement special-purpose accelerators using the conventional register transfer level (RTL) methodology. For this reason, both academia and industry are seeing an increasing use of high-level synthesis (HLS) to automatically generate hardware accelerators from software programs.

The course provides an introduction to the hardware accelerator design principles and the modern HLS design methodologies and tools, focusing on FPGA targets. Specific topics include C-based HLS design methods, hardware specialization, scheduling, pipelining, resource sharing, and case studies on deep learning acceleration. Commercial C-to-FPGA tools will be provided to the students to implement real-life image/video processing and machine learning applications on programmable system-on-chips that tightly integrate CPU and FPGA devices.

2.1. Prerequisites

This course assumes the student has a working knowledge of C/C++ and familiarity with basic concepts of digital logic and computer architecture, such as sequential circuits, timing analysis, pipelining, etc. A knowledge of basic algorithms and data structures is preferred. Experiences with RTL design for either ASICs or FPGAs would be helpful, although not required.

2.2. Target Audience and Learning Outcomes

This course is open to graduate students and senior undergraduates, who are interested in (1) learning application-/domain-specific hardware acceleration and (2) understanding the capabilities of current HLS tools and design methodologies. Upon completion of this course, students will be able to use HLS tools to design realistic hardware accelerators on FPGAs.

3. Course Organization

This course includes a combination of lectures (Friday) and a few TA-led tutorials (Tuesday) that cover the following topics.

1. Introduction
2. Hardware specialization
3. FPGAs
4. C-based synthesis
5. Front-end compilation
6. Scheduling and pipelining
7. Resource sharing
8. Deep learning acceleration

Students are expected to attend lectures and actively participate in various technical discussion during the lecture.

3.1. Assignments and Grading

There are three lab assignments that involve using C++ programming languages and HLS tools to design hardware accelerators. All assignments will be graded by completion. A student must at least complete two assignments to pass the course.

3.2. Academic Integrity

Each student in this course is expected to abide by the Cornell University Code of Academic Integrity. Any work submitted by a student in this course for academic credit will be the student's own work.
The term "group" in this section refers to yourself if you work alone or to you and your partner in case of a group (team of two) project. The work your group submits is expected to be the result of your group's effort only. The use of a computer in no way modifies the standards of academic integrity expected under the Cornell University Code of Academic Integrity. You are encouraged to study together and to discuss information and concepts covered in lecture with other students. You can give "consulting" help to or receive "consulting" help from such students. However, this cooperation should never involve one group having possession of a copy of all or part of the work done by some other group, including work from previous years. Should copying occur, both the student(s) who copied work from another student and the student(s) who gave material to be copied will automatically receive a zero on the work, and an extra penalty will be assessed, ranging anywhere from a deduction on the final grade to failure of the course and university disciplinary action. Please notice that this implies that at no time are you allowed to grant anyone but your group partner access to your computer files. Be sure to master the use of chmod and umask before starting to work on your projects. During exams, you must do your own work. Communication among students is not permitted during the exams, nor may you compare or borrow notes, copy from others, or collaborate in any way. You are strongly encouraged to read Cornell University's Code of Academic Integrity, available at https://blogs.cornell.edu/provost/files/2015/04/AcademicIntegrityPamphlet2015-p7hd3n.pdf.

3.3. Accommodations for Students with Disabilities

In compliance with the Cornell University policy and equal access laws, the instructor is available to discuss appropriate academic accommodations that may be required for students with disabilities. Requests for academic accommodations are to be made during the first three weeks of the semester, except for unusual circumstances, so arrangements can be made. Students are encouraged to register with Student Disability Services to verify their eligibility for appropriate accommodations.