CS 59200: Quantum Computer Software Stacks

Course Information

Instructor: Dr. Yuxiang Peng
Term: Spring 2026
Schedule: TuTh 9:00 am – 10:15 am
Location: GRIS 133

Course Description

This seminar explores the design, implementation, and formal verification of programming languages and compiler toolchains for quantum computers. Students will study emerging paradigms for expressing quantum computation, techniques for optimizing and verifying quantum programs and compilers, and the integration of quantum programming into cloud computing platforms. Topics include intermediate representations, circuit optimization, type systems for quantum data, and formal verification frameworks for quantum programs.

Lectures on the introduction to these topics will be given in the first half of the semester. Students will read and present seminal and recent research papers on these topics, finish small programming assignments, conduct a research-oriented final project, and write a final project report.

Prerequisites

Students should have graduate-level background in quantum computing. Prior courses in programming languages, compilers, and architecture are also recommended.

Learning Resources and Texts

Reading list includes but is not limited to:

1. Foundations of Quantum Programming, Mingsheng Ying.
2. Verified Quantum Computing, Robert Rand.
3. Quantum Computer Systems: Research for Noisy Intermediate-Scale Quantum Computers, Yongshan Ding, Frederic T. Chong.

4. Quantum Software: Aspects of Theory and System Design, Iaakov Exman, Ricardo Pérez-Castillo, Mario Piattini, Michael Felderer.

Learning Objectives

By the end of the course, students will:

1. Understand the core abstractions of quantum programming languages and how they differ from classical ones.
2. Gain familiarity with multiple quantum programming languages with different paradigms.
3. Learn basic compiler strategies for quantum circuit optimization, scheduling, and mapping to hardware.
4. Know how to formally verify the correctness of quantum programs and compilers.
5. Explore hardware–software co-design in quantum compilers.
6. Conduct research in quantum algorithm design and implementation, quantum programming languages and compilers.
7. Develop research insights into the next generation of quantum programming and compilation tools.

Assignment and Grading

Grades will be based on coding assignments (25%), topics reading reports (25%), in-class midterm exam (25%), and a research project report (25%).

- **Coding assignments (25%):** there will be 3 coding assignments, practicing students' skills in several quantum programming languages. Students will have chances to implement sophisticated quantum algorithms, compiler optimizations, and formal verifications for quantum programs.
- **Topics reading reports (25%):** students will form teams to survey literature focusing on one selected topic in quantum algorithm implementation, programming language design, compiler optimization, or software technology for specific quantum hardware platforms. A survey report will be submitted and evaluated.

- **In-class midterm exam (25%):** one “pen-and-paper” midterm exam on the basic understanding of quantum programming languages and compiler design and implementations.
- **Final project report (25%):** students will form teams to conduct research in areas related to this course. A final report summarizing the findings of the students will be submitted and evaluated.