EECE 528: PARALLEL AND RECONFIGURABLE COMPUTING

Instructor: Prof. Guy Lemieux

Description:

Semiconductor technology has reached the limits of clock frequency scaling, so the new focus is on providing increased amounts of parallelism at the chip level. This graduate level course will examine current and emerging parallel architectures and programming methods for exploiting large amounts of parallelism. This will include traditional approaches using multiple CPUs as well as newer approaches using reconfigurable computing, where a custom compute engine is built from one or more FPGAs. The course will examine the types of parallelism available, the limits of how much parallelism can be exploited, and ways to avoid such limits. Several programming assignments will be used to demonstrate the different techniques of expressing parallelism, and a final research-oriented project will be used to explore the differences of CPUs versus FPGAs in greater detail. The focus will be on uncovering advanced forms of parallelism.

Syllabus:

Parallel Computing

1. Concurrent Models of Computation
2. Flynn's Classification, Amdahl's Law, Compiler Optimizations
3. Benchmarking, Performance
4. Levels of Parallelism: Task, Thread, Data, Loop, Instruction, Bit
5. Limits of Parallelism (Wall paper)
6. Message Passing (eg, MPI or PVM)
7. Shared Memory (eg, OpenMP)
8. Memory Consistency, Cache Coherence, Transactional Memory
9. Interconnection Networks: Routing, Flow Control, Deadlock
10. Processor-in-Memory
11. Dataflow Processing
12. Vector/SIMD Processing

Reconfigurable Computing

13. Custom Instruction-Set Processing, ASIPs
14. High-level Synthesis
15. Compiling "C to Hardware"
16. Coarse-grain Arrays
17. Fine-grain Device Architectures
18. Non-conventional Programming Methods
19. Performance Issues: Pipelining, Retiming and C-slow Retiming

Grading (yes, this adds up to 110%):

60% Assignments (6 assignments)
10% Weekly Presentation (1 paper, 20 minutes)
10% Reading (1 page summary, each week)
20% Project
10% Participation/discussion (cue cards)

Course Structure:

Lectures: CEME1212 Wed 1300h-1500h Fri 1300h-1430h
(time and location may change to suit enrolled students)

6 Parallel Programming Assignments and 1 Project:

Sept 22	0. Compiler optimizations (gcc)
Oct 6	1. Message passing (OpenMPI)
Oct 13	2. Shared memory (pthreads, OpenMP)
Oct 20	3. GPUs (CUDA)
Nov 3	4. Vectors (VectorBlox MXP)
Nov 17	5. Choice: High-level Synthesis, OpenCL, Bluespec, VHDL/Verilog, or your suggestion.
Dec 1	6. PROJECT DUE, ABSOLUTELY NO EXTENSIONS

Workload: Assignment >20hrs, Project >40hrs

Presentation: Each student **TEACHES** 1 paper for 15 minutes and leads 5 min discussion. Emphasize the **key idea**. Signup list, first 3 speakers repeat to improve grade.

Reading: Each week, every student must read the assigned papers **in advance** and write a 1-page summary.

Project: Students will undertake a research and design project to solve a problem they propose (e.g., from their research). Compare performance of one parallel algorithm implemented using 2 different techniques. Students may work in pairs with higher expectations for quality and quantity of work.

Textbooks to consult:

- "Parallel Computer Architecture: A Hardware/Software Approach", Culler and Singh, Morgan Kaufmann/Elsevier, 1998
- "Reconfigurable Computing", Hauck and DeHon, Morgan Kaufmann/Elsvier, 2008