Cmpe221 - Advanced Microprocessor Design

Introduction to the latest advances in computer architecture. Focuses on processor core design. Topics include simultaneous multithreading, thread level speculation, trace caches, novel out-of-order mechanisms, and energy-efficient processor core designs. Final project is the modification/enhancement of an out-of-order processor on, an FPGA development system.

There is a big project associated with this class. It consists of enhancing an existing soft core (Leon based). A soft core is a VHDL/Verilog processor core that can be synthesized on a variety of FPGAs.

This course consists of two major blocks: Theory and lab. For the lab we use modelSim and several FPGA boards with the largest and most advanced chips: Stratix II or Virtex-4.

Requirements: To take this class you are required to be knowledgeable of on computer architecture (cmpe202 or equivalent), and to have some experience with Verilog or VHDL (cmpe125 or equivalent). Concurrent enrollment in the Lab class is required. No textbook is required.

Advanced Microprocessor Design Laboratory (3 credit) consists of weekly two-hour lab session, and 10-15 hours of independent work.

The grading is based on the project and the paper reviews. There is one big project due, and every week there are several papers that need to be reviewed. There are no exams or homeworks.

The indented audience of this class are graduate students that already know computer architecture and want to increase their knowledge in this area. At the end of the class, students will have learned the internals of current microprocessors, future research trends, and gained experience with soft cores.

Class Structure:

1st week: Introduction & fast review cmpe202 materials

Advanced pipelining, scheduling, RATs, ROBs, Instruction Window

2nd week: Leon platform

SPARC ISA, register window
AMBA Bus, Interrupts, exceptions
SCOORE processor

3rd week: Fetch Engine

Trace Cache & icaches
branch predictors, OBQ (Outstanding Branch Queue), future predictor

4th week: Execution Engine

RATs, ROBs, speculative execution

5th week: Memory Subsystem

Load/Store Queues & speculation
DRAMs, IRAM, FlexRAM

6th week: SMT & Cluster architectures

Simultaneous Multithreading
Clustered architectures

7th week: Thread Level Speculation

8th week: Power, Energy, and Temperature

Power, energy, and energy models
Optimizations

9th week: Multiprocessors

Coherence & directory based coherence
Consistency issues

10th week: Fault-Tolerance

11th week: Projects

Reading List

1st week

1. POWER4 System Microarchitecture
2. Alpha 21264/EV67 Microprocessor Hardware Reference Manual

2nd week:

1. Leon manuals
2. sparc V8
3. Complexity-Effective Superscalar Processors

3rd week:

1. Trace Cache Intel Patent
2. An Analysis of Correlation and Predictability: What Makes Two-Level Branch Predictors Work
3. Trace Cache: a Low Latency Approach to High Bandwidth Instruction Fetching
4. Trace Bases Instruction Caching Patent (Pentium 4 patent)

4th week:

1. The Design Space of Register Renaming Techniques
2. A Novel Renaming Scheme to Exploit Value Temporal Locality through Physical Register Reuse and Unification
3. Delaying Physical Register Allocation Through Virtual-Physical Registers
4. Organization and implementation of the register-renaming mapper for out-of-order IBM POWER4 processors

5th week:

1. FlexRAM Architecture Design Parameters
2. A high bandwidth memory pipeline for wide issue processors
3. Scalable Hardware Memory Disambiguation for High ILP Processors
4. Memory Ordering: A Value-Based Approach

6th week:

1. Simultaneous Multithreading: Maximizing On-Chip Parallelism
2. The raw microprocessor: A computational fabric for software circuits and general-purpose programs
3. Instruction Distribution Heuristics for Quad-Cluster, Dynamically-Scheduled, Superscalar Processors
4. Dynamic Cluster Assignment Mechanisms

7th week:

1. Multiscalar processors
2. Speculative Synchronization: Applying Thread-Level Speculation to Explicitly Parallel Applications
3. Cherry: Checkpointed Early Resouce Retirement
4. CAVA: Hiding L2 Misses with Checkpoint-Assisted Value Prediction

8th week:

1. Wattch
2. New methodology for early-stage, microarchitecture-level power–performance analysis of microprocessors
3. Energy-Efficient Thread level Speculation
4. Power Considerations in the Design of the Alpha 21264 Microprocessor

9th week:

1. Reducing Memory and Traffic Requirements for Scalable Directory-Based Cache Coherence Schemes
2. Performance Evaluation of Memory Consistency Models for Shared-Memory Multiprocessors
3. Shared Memory Consistency Models: A Tutorial
4. Two Techniques to Enhance the Performance of Memory Consistency Models

10^{th} week:

1. ReVive: Cost-Effective Architectural Support for Rollback Recovery in Shared-Memory Multiprocessors
2. Detailed Design and Evaluation of Redundant Multithreading Alternatives