Special Topics in Advanced Computing Technology: Programming and System Design

by

Dr. Lesley Shannon

Email: lshannon@ensc.sfu.ca

Course Website: http://www.ensc.sfu.ca/~lshannon/courses/ensc894
Slide Set Overview

• Course Motivation

• Logistical Details

• Course Components (handouts review)

• Lecture Topics
Computing Technology is everywhere & it’s one of the largest growing job markets
Personal Computers
Prescott Pentium 4

122 mm² with 125 million transistors
Intel Pentium 4 Prescott

Trace Cache Access, next Address Predict

Instruction Trace Cache

Execution Pipeline Start

Buffer Allocation & Register Rename

Instruction Queue (for less critical fields of the uOps)
General Instruction Address Queue & Memory Instruction Address Queue
(queues require entries and field of this uOps for scheduling)

uOp Schedulers

Parallel (Matrix) Scheduler for the two double pumped ALU's
General Floating Point and Slow Image Scheduler
(FPU dependency matrix)
FP Move Scheduler
(86 dependency matrix)
Load / Store Linear Address
Callout History Table
Load / Store uOp Scheduler
(86 dependency matrix)

FP, MMX, SSE1..3

Floating Point, MMX, SSE1..3
Renamed Register File
24 byte entries of 128 bit

Integer Execution Core

(1) uOp Dispatch unit & Dispatch Buffer
Deploys up to 6 uOps/cycle

(2) Integer Renamed Register File
256 entries of 32 bit (16 status flags)
12 read ports and six write ports

(3) Datapath control & Bypasses to and from the integer Register File

(4) Flags, Write Back

(5) Double Pumped ALU

(6) Double Pumped ALU

(7) Load Address Generator Unit

(8) Store Address Generator Unit

(9) Load Register (56 entries)

(10) Store Buffers (48 entries)

Instruction Fetch from L2 cache and Branch Prediction

Front Side Bus Interface, 533..800 MHz

512 kByte L2 Cache Block

L2 Phys Tags
L2 Data Tags

512 kByte L2 Cache Block

L2 Phys Tags
L2 Data Tags

April 19, 2003  www.chip-architect.com
Dual-Core Intel Itanium 2 processor (Montecito)

1.72 billion transistors, 27.72 mm x 21.5 mm
Core Duo
But what about …
Microsoft Kinect vs the ipad & iphone
We don’t even read the same ...
Outer-Space
Mars Rover

Xilinx

FPGAs

Here
Orbital Space Plane
Orbital Space Plane
Cars & Boats
Boats
Automotive Electronics

7-Series BMW: 63 Embedded Processors

Mercedes S-Class: 65 Embedded Processors

More than 80% of the innovation in autos is from innovations in electronics.
- Daimler-Chrysler

Automotive Semiconductor Market: US$13.1 billion / year
Inside you
Biomedical Applications

Diagnostic Applications: MRI, PET, FD-OCT…

Tele-health (diagnostics and surgery)
Cellular Applications
Cellular Applications

Cellphone Networks

&

Cellphones

ENSC 894: Lecture Set 0

22
Predicting the weather
Meteorological Applications

Doppler Weather Radar

National Doppler Radar Sites
Select radar location and click.
Requires Java/Javascript
Finding Oil
Seismic Imaging Applications

Let’s play: Find the petroleum!
Making movies
Special Effects and Computer Animated Films

Avatar: Petabytes worth of data

Lots of movie special effects done here in Vancouver
PLUS:

Music Mixing/Sound Effects
Environmental Monitoring
Production/Manufacturing Plants
Power Grid
Telecommunications Grid
Banking and Stock Markets
Hadron Collider
Quick Review: what should you know already?

• Von Neumann Architecture
• A software programming language (C/C++/java)
• Basic digital logic components and Synchronous Logic
  – Logic Gates and Registers
  – Counters, state machines
• Pipelining
  – Not for processors per se, but to make logic faster
Quick Review (cont’d)

• Numbers and Arithmetic
  – Base 2 and 16, floating point/fixed point, masking, etc

• I/O
  – Accessing via memory mapping, interrupts, buses, point-to-point, DMA

• Memory System
  – Distributed vs Shared
Quick Review (cont’d)

• Parallel Programming Concepts
  – Mutexes, semaphores, locks, barriers, message passing

• Assembly Language (what is it)?

• Context Switching

• Transistors (what the word means)
Quick Review (cont’d)

• CAD Flow or Compilation Flow
  – Basic understanding

• Operating Systems Basics
Why take this course?

• It’s cool!

• Gives you a great overview of how computing systems work from top to bottom (good introductory course)

• Gives you a practical understanding of how the different layers interact and impact performance

• Highlights the kind of considerations that need to be evaluated for Application-Specific Designs (HUGE MARKET)

*Note: Two in-class tests.*
What will you learn?

• What’s the cool research in computing system design!?!?

• Basic interactions between hierarchy layers
  – How they impact your design and performance

• Reading and analysing research papers:
  – Critical Analysis skills
About Me

• Microelectronics Group

• Computer Engineer:
  – I teach Digital and Embedded Systems Design courses
  – My research group is the Reconfigurable Computing Lab

• My interests include:
Logistics
My Contact Info

• Office:
  – ASB8819

• Email:
  – Ishannon@ensc.sfu.ca

• Homepage:
  – http://www.ensc.sfu.ca/~Ishannon/

• Office Hours:
  – Tuesdays 10:30-12:30pm & Wednesdays 2:30-3:30pm
Lectures

• Wednesdays 5:30-8:30pm
  – 5:30-7pm; 15 minute break; 7:15-8:15pm

• May schedule a Guest Lecture
  • If so ATTENDANCE IS MANDATORY

• Guest lecture at UBC recommended:
  – Dr. Philip Leong (more to follow; check the bulletin board)
Course Webpage

• Course Webpage:
• Course Handouts will be posted here.
• Lecture Slides will also be posted before each class.
• We will be using WebCT for bulletin board postings only
• A Course Mailing List has been created
  – It will only be used to post last minute notices.
  – All other information will be posted to WebCT
Please observe appropriate bulletin board etiquette (be respectful).
Please do not use these forums to post any material that is knowingly false and/or defamatory, abusive, vulgar, hateful, obscene, threatening, invasive of a person's privacy, or otherwise in violation of any law. Do not post any copyrighted material unless the copyright is owned by you. I reserve the right to remove any messages posted and to reveal your identity (or whatever information is known about you) in the event of a complaint or legal action arising from any message posted by you.

By posting your message, you agree to indemnify me, my employees, agents and representatives, and to hold them harmless from any and all claims and liabilities (including lawyers' fees) resulting from any material posted to these forums, or from any acts resulting from participants' use of these forums.
Class Surveys

- This is the FIRST offering of the course:
  - I’m developing/revising the lecture material
  - For this, I really would appreciate your honest input
  - I’m interested to find out what you think of the
    - Course Material
    - Pace and Depth of Presentation
    - Selected Readings
    - Approach
  - I’ll be giving a survey near the end of the course
    - However, if people have concerns/suggestions during the term, please let me know
Grade Breakdown

- 10% Reading Group Participation
- 15% 3 Paper Critiques*
- 25% Seminar Presentation
- 25% Research Review*
- 25% Application Analysis*

*Although there is no official course project, at the instructor’s discretion, students can choose to do a project in lieu of one more components of the course work (see handout)
Handouts
Remember: Pictures say a thousand words!
Lecture Topics
Lecture Topics

• January 26th: Programming Models – how we picture the problem.
  - Parallel Models: Dataflow, Queuing Models, Streams; Shared vs Distributed

• February 2nd: Programming Languages
  - Language Choice: VHDL, CUDA, OpenCL, LabView, Java, …
  - Algorithms and Data Structures

• February 9th: Analysing Application Performance, Debugging and Testing
  - Hardware Performance Counters, Gprof, vtune, Simulators, Testbeds, Code coverage, software engineering (program Design)
Lecture Topics

• February 16\textsuperscript{th}: Synthesis and Compilation
  – Just in time, Dynamic Recompiling
  – Optimization choices (area, power, speed)
• February 23\textsuperscript{rd}: Operating Systems
  – Key functionalities; Customization (Keeping what you need)
  – Scheduling
• March 2\textsuperscript{nd}: CLASS CANCELLED
Lecture Topics

• March 9\(^{th}/16\(^{th}\): Computing Architectures (SoC vs the Cloud)**
  – Heterogeneous, NoC, Embedded Computing, reconfigurable computing
  – Using the cloud, managing data in the cloud

• March 16\(^{th}/23\(^{rd}\): Computing System Design
  – Where’s the bottleneck? Memory hierarchy and memory Technology

• March 23\(^{rd}/30\(^{th}\): Reliability and DFT, Silicon Debug, Verification
  – Critical Systems
Lecture Topics

• March 30\textsuperscript{th}/April 6\textsuperscript{th}: Security
  – Is your application/data safe? (Drop boxes, …)

• April 6\textsuperscript{th}/13\textsuperscript{th}: Computing Technology
  – CMOS (how far can we go); molecular; quantum; ???
Lecture Topics: What does this look like?
Other Possible Lecture Topics

• Asynchronous Design
  – Data Synchronization Issues
  – Meta-stability
For Reference
Computing System Implementations

- ASIC
- Alternative Computing Architectures
- Von Neumann Architecture

Application-Specific Architecture vs Flexibility in Software
Computing System Implementations

- ASIC
- Alternative Computing Architectures
- Von Neumann Architecture

Flexible in Software
Application-Specific Architecture
Alternative Computing Architectures

• Systems-on-Chip
• Many-core heterogenous Systems-on-Chip
• Networks-on-Chip
• Embedded and Real Time Computing Systems
• Reconfigurable Computing
• Meta-Computing/the cloud
• .... and more
Recall: System-on-Chip

- System on Printed Circuit Board
- System-on-Chip (SoC)
Multi-Processor System-on-Chip

![Diagram showing a multi-processor system-on-chip with processors and memories connected through a shared bus.]
Network-on-Chip

Processing Unit
Processing Unit
Processing Unit
Processing Unit
Processing Unit
Processing Unit
Processing Unit
Processing Unit
Processing Unit
Embedded Computing Systems

- Design Constraints:
  - Area
  - Power
  - Performance
  - Cost

Diagram:
- Embedded Computing Systems
- Processor
- Dedicated Logic
Reconfigurable Computing

Chip Area:

Storm Cell Tracking

Precipitation Estimation

Unused

Dual Doppler Analysis

Storm Cell Identification
Meta-Computing/the Cloud