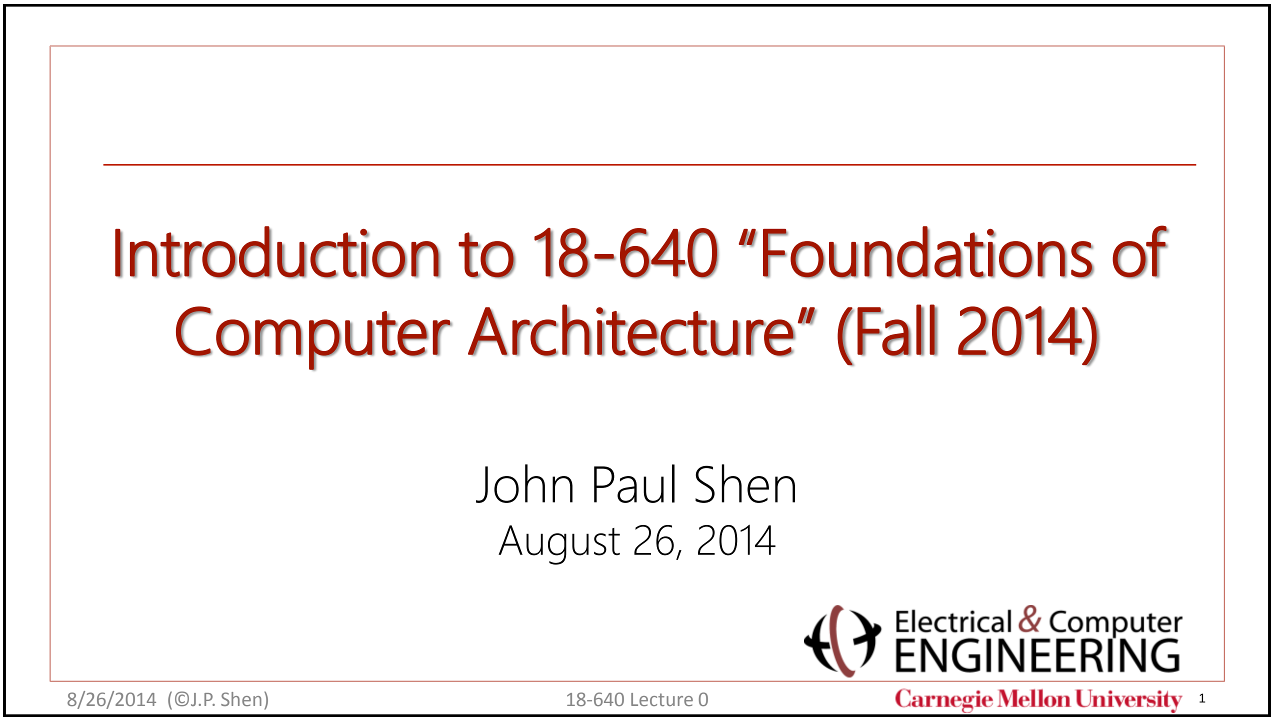
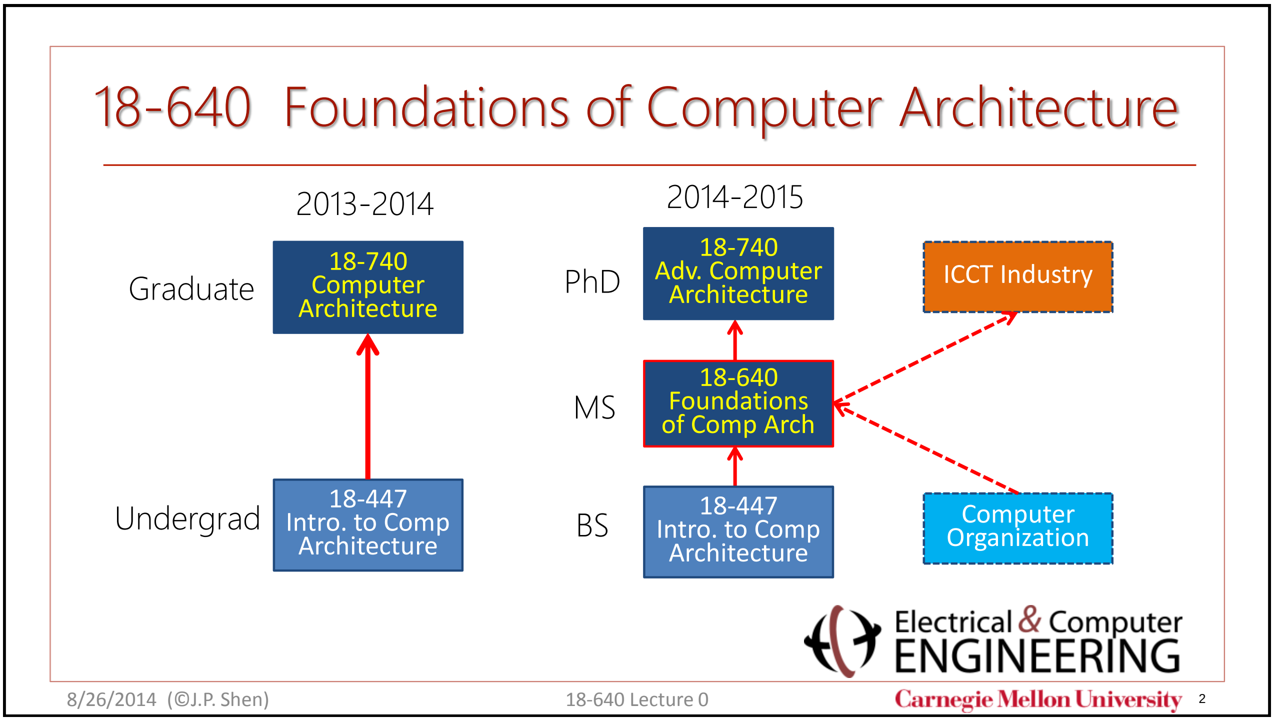


Introduction to 18-640 “Foundations of Computer Architecture” (Fall 2014)

John Paul Shen

August 26, 2014



18-640 Foundations of Computer Architecture

2013-2014

Graduate: 18-740 Computer Architecture

Undergrad: 18-447 Intro. to Comp Architecture

2014-2015

PhD: 18-740 Adv. Computer Architecture

MS: 18-640 Foundations of Comp Arch

BS: 18-447 Intro. to Comp Architecture

ICCT Industry

Computer Organization

18-640 Cast of Characters

- Instructor: John P. Shen (SV)
- Academic Services Assistants:
  - Chelsea Mastilak (Pgh)
  - Stephanie Scott (SV)
- Teaching Assistants:
  - Naman Jain (Pgh)
  - Guangshuo (Eric) Liu (Pgh)
    - Cover Pgh & GZ campuses
    - Primary TAs on Lab Projects
  - Maxim Kovalev (SV)
  - Mridula Srinivasa (SV)
    - Cover SV campus
    - Primary TAs on Homeworks
- SYSU Course Coordinator: Shuochen (Bill) Zhang (GZ)

My Personal Background:

- Academia
  - Carnegie Mellon University (1982-2000)
    - First Half → Computer Aided Design
      - Sabbatical at Stanford
    - Second Half → Computer Architecture
      - Sabbatical at Intel
- Industry
  - Intel, Research Lab (2001-2006)
    - Microarchitecture Lab → Microprocessor Design
  - Nokia, Research Center (2007-2014)
    - North America Lab → Mobile Computing System

Major lessons learned in my career …

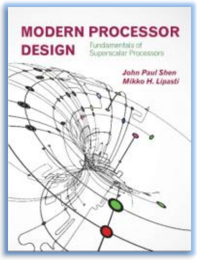
- CMU:
  - How to conquer the world with 10 PhD students.
  - How to succeed as an empire-builder builder.
- Intel:
  - How to conquer the world with 30 industry researchers.
  - How to deal with having a boss, business processes, and endless meetings.
- Nokia:
  - How to conquer the world with 90 diversely talented innovators.
  - How to help transform a 150 year old Finnish (not Japanese) company.

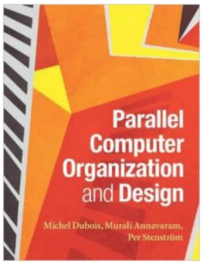
Major lessons learned in my life …

- Academia:
  - Teacher-student relationships → life-long friendships
  - Ideal for making impacts that last beyond our life time
- Industry:
  - All corporate business successes are only temporal
  - Two leadership styles: Advancement vs. Adventure
  - Two approaches to career: Ends driven vs. Means driven

18-640 Required Textbooks

- Two Required Textbooks:
  - *Modern Processor Design: Fundamentals of Superscalar Processors,* by John Paul Shen and Mikko H. Lipasti. Waveland Press.
  - *Parallel Computer Organization and Design,* by Michel Dubois, Murali Annavaram, Per Stenstrom, Cambridge University Press, 2012.

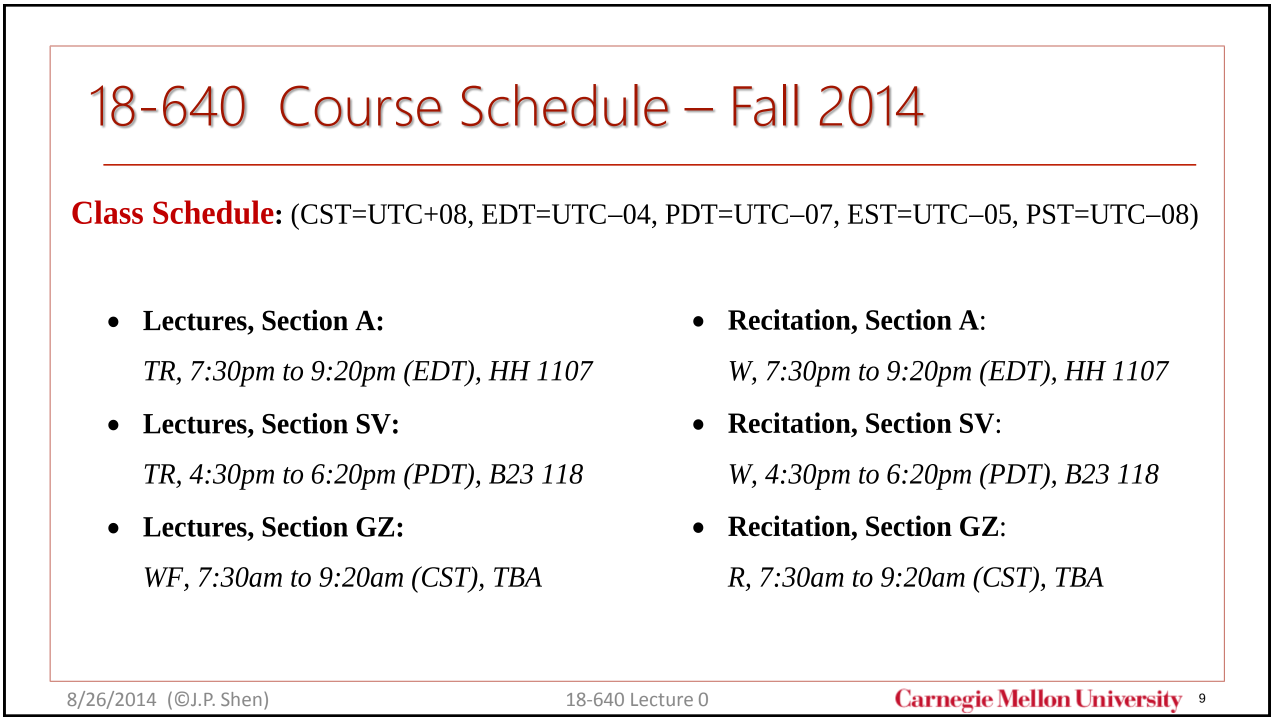



Parallel Computer Organization and Design
Michel Dubois, Murali Annavaram, Per Stenström

Electrical & Computer ENGINEERING

18-640 Assumptions and Expectations

- Who should take 18-640?
  - Graduate students (MS/IMB/PhD)
    - ○ computer architects to be (designing of & with computer systems)
    - ○ computing systems expertise and computer architects' mindset
- Assumed undergraduate background:
  - basic architecture or computer organization (18-347/447 or P&H)
  - basic OS and compilers (15-411, 15-412)
  - C/C++ programming
- Course expectations:
  - in-class discussion encouraged; focusing on foundational principles and key insights
  - emphasis on current cutting-edge issues as well as key historical advancements
  - hands-on team-based lab projects using Gem5 execution-driven simulator system
  - opportunities for extra credit for project teams: in-class presentation of project results



18-640 Course Schedule – Fall 2014

Class Schedule: (CST=UTC+08, EDT=UTC–04, PDT=UTC–07, EST=UTC–05, PST=UTC–08)

- **Lectures, Section A:**

  TR, 7:30pm to 9:20pm (EDT), HH 1107
- **Lectures, Section SV:**

  TR, 4:30pm to 6:20pm (PDT), B23 118
- **Lectures, Section GZ:**

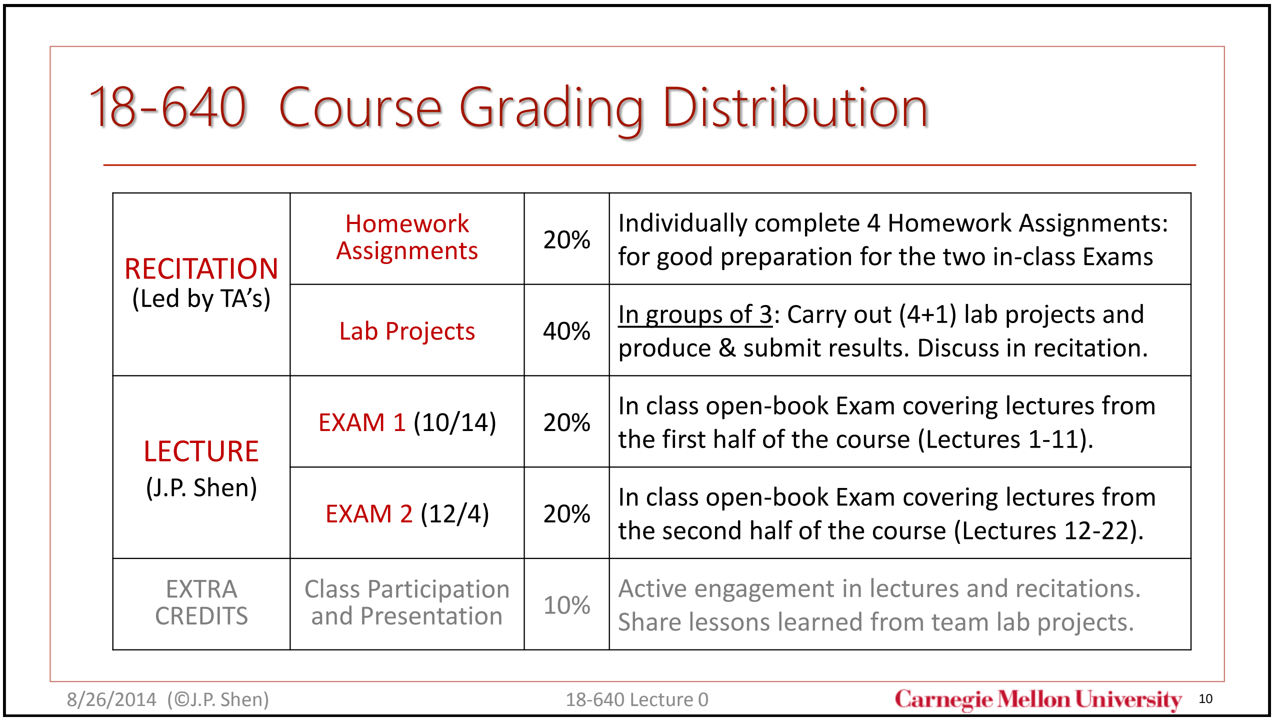
  WF, 7:30am to 9:20am (CST), TBA

- **Recitation, Section A**:

  W, 7:30pm to 9:20pm (EDT), HH 1107
- **Recitation, Section SV**:

  W, 4:30pm to 6:20pm (PDT), B23 118
- **Recitation, Section GZ**:

  R, 7:30am to 9:20am (CST), TBA



18-640 Course Grading Distribution

RECITATION (Led by TA's)	Homework Assignments	20%	Individually complete 4 Homework Assignments: for good preparation for the two in-class Exams
	Lab Projects	40%	In groups of 3: Carry out (4+1) lab projects and produce & submit results. Discuss in recitation.
LECTURE (J.P. Shen)	EXAM 1 (10/14)	20%	In class open-book Exam covering lectures from the first half of the course (Lectures 1-11).
	EXAM 2 (12/4)	20%	In class open-book Exam covering lectures from the second half of the course (Lectures 12-22).
EXTRA CREDITS	Class Participation and Presentation	10%	Active engagement in lectures and recitations. Share lessons learned from team lab projects.

18-640 Course Syllabus – Fall 2014

Superscalar Processor Design (SSP)
Instruction-Level Parallelism (ILP)

	Date	Day	Class Activity
	August		
Week 1	26	Tues.	Introduction to 18-640 "Foundations of Computer Architecture"
	27	Wed.	Lecture 1: Introduction to Computer Architecture
	28	Thurs.	Lecture 2: Review of Pipelined Processor Design
	September		
Week 2	2	Tues.	Lecture 3: From Pipelining to Superscalar (ILP)
	3	Wed.	Overview of Lab Projects (4+1) – (form teams of 3)
	4	Thurs.	Lecture 4: Instruction Flow Techniques
Week 3	9	Tues.	Lecture 5: Branch Prediction Techniques
	10	Wed.	**PROJECT #1 briefing**
	11	Thurs.	Lecture 6: Register Data Flow Techniques
Week 4	16	Tues.	Lecture 7: Dynamic Out-of-Order Execution
	17	Wed.	HOMEWORK #1
	18	Thurs.	Lecture 8: Review of Memory Hierarchy and Virtual Memory
Week 5	23	Tues.	Lecture 9: Memory Data Flow Techniques
	24	Wed.	**PROJECT #2 briefing**
	25	Thurs.	Lecture 10: Main Memory System Design (MLP)
	October		
Week 6	30	Tues.	Lecture 11: VLIW Architecture (ILP & LLP)
	1	Wed.	HOMEWORK #2
	2	Thurs.	Lecture 12: Dynamic Code Translation & Compilation
Week 7	7	Tues.	**Special Lecture**: "*Snap Dragon Multi-Core SOC*" [Bob Rychlik, Qualcomm]
	8	Wed.	(Presentation by Select Project Teams)
	9	Thurs.	REVIEW for EXAM 1
Week 8	14	Tues.	EXAM 1
	15	Wed.	**PROJECT #3 briefing**
	16	Thurs.	Lecture 13: Multithreading Processors (from ILP to TLP)
Week 9	21	Tues.	Lecture 14: Multi-Core Processor Architecture (ILP & TLP)
	22	Wed.	(Presentation by Select Project Teams)
	23	Thurs.	Lecture 15: Multi-Core Cache Coherence
Week 10	28	Tues.	Lecture 16: Multi-Core Parallel Programming
	29	Wed.	**PROJECT #4 briefing**
	30	Thurs.	**Special Lecture**: "*Web Based Computing Platforms*" [Antero Taivalsaari, Nokia]
	November		
Week 11	4	Tues.	Lecture 17: SIMD and Graphics Processors (DLP)
	5	Wed.	HOMEWORK #3
	6	Thurs.	Lecture 18: Performance and Power Iron Laws (the science part of comp arch)
Week 12	11	Tues.	Lecture 19: Power Reduction and Energy Efficiency (the new challenge)
	12	Wed.	**PROJECT #5 briefing**
	13	Thurs.	Lecture 20: Cluster Computing Systems
Week 13	18	Tues.	Lecture 21: Mobile and Cloud Computing
	19	Wed.	HOMEWORK #4
	20	Thurs.	**Special Lecture**: "*From Processor Design To NBA Technology*" [Quinn Jacobson]
Week 14	25	Tues.	Lecture 22: War of Mobile Platforms and Ecosystems (A Personal Retrospective)
	26	Wed.	NO CLASSES - THANKSGIVING
	27	Thurs.	NO CLASSES - THANKSGIVING
	December		
Week 15	2	Tues.	Review for EXAM 2
	3	Wed.	(Presentation by Select Project Teams)
	4	Thurs.	EXAM 2

Multi-Core Processor Design (MCP)
Thread-Level Parallelism (TLP)

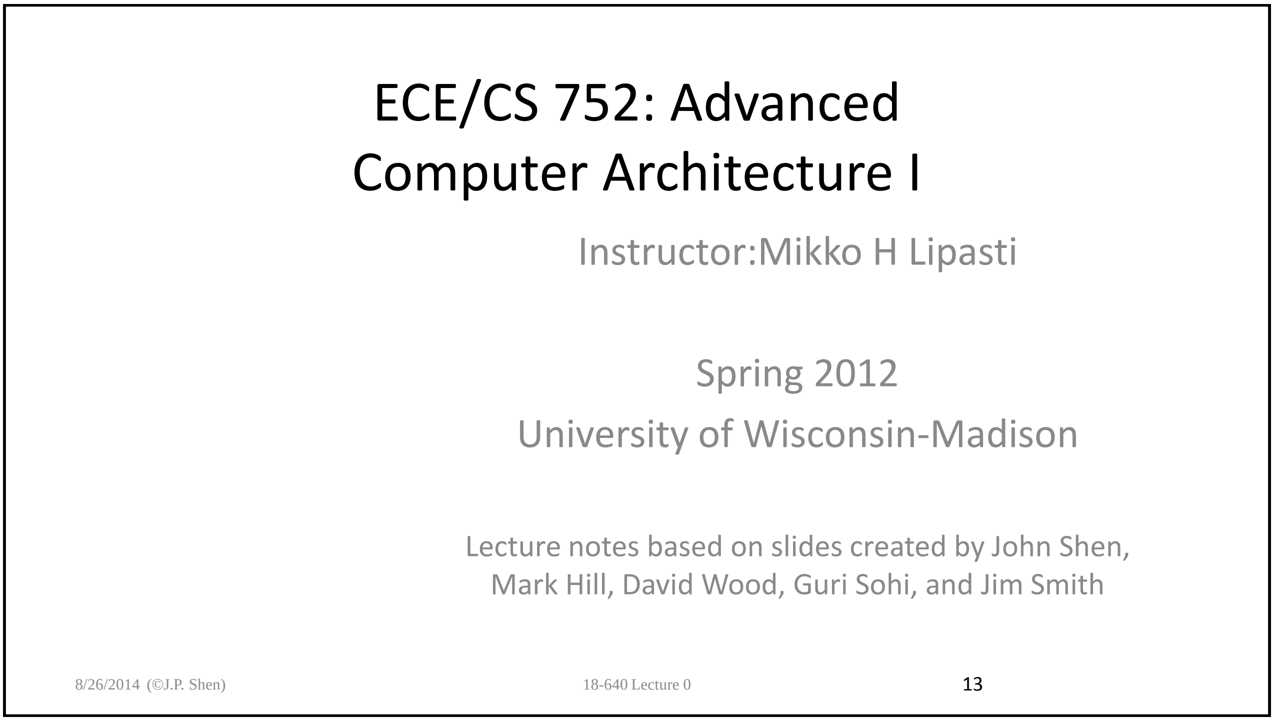
18-640 Similar Courses at Other Universities

Superscalar Processor Design (SSP) Instruction-Level Parallelism (ILP)

- ECE/CS 752 "Advanced Computer Architecture I" (University of Wisconsin)
- EE 382A "Advanced Processor Architecture" (Stanford University)

Multi-Core Processor Design (MCP) Thread-Level Parallelism (TLP)

- ECE 757 "Advanced Computer Architecture II" (University of Wisconsin)
- EE 557 "Parallel Computer Organization and Design" (University of S. California)



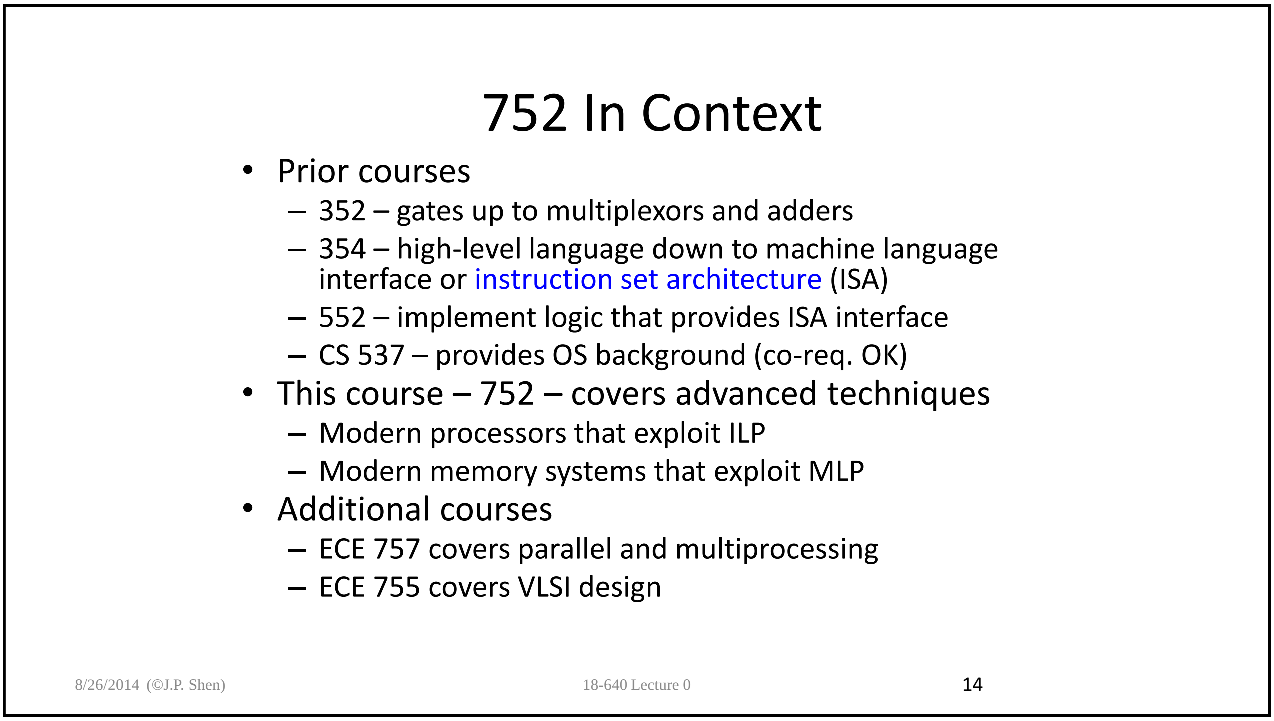
ECE/CS 752: Advanced Computer Architecture I

Instructor:Mikko H Lipasti

Spring 2012

University of Wisconsin-Madison

Lecture notes based on slides created by John Shen, Mark Hill, David Wood, Guri Sohi, and Jim Smith



752 In Context

- Prior courses
  - 352 – gates up to multiplexors and adders
  - 354 – high-level language down to machine language interface or instruction set architecture (ISA)
  - 552 – implement logic that provides ISA interface
  - CS 537 – provides OS background (co-req. OK)
- This course – 752 – covers advanced techniques
  - Modern processors that exploit ILP
  - Modern memory systems that exploit MLP
- Additional courses
  - ECE 757 covers parallel and multiprocessing
  - ECE 755 covers VLSI design

Why Take 752?

- To become a computer designer
  - Alumni of this class helped design your computer
- To learn what is *under the hood* of a computer
  - Innate curiosity
  - To better understand when things break
  - To write better code/applications
  - To write better system software (O/S, compiler, etc.)
- Because it is intellectually fascinating!
  - What is the most complex man-made single device?

Computer Architecture

- Exercise in engineering tradeoff analysis
  - Find the fastest/cheapest/power-efficient/etc. solution
  - Optimization problem with 100s of variables
- All the variables are changing
  - At non-uniform rates
  - With inflection points
  - Only one guarantee: Today's right answer will be wrong tomorrow
- Two high-level effects:
  - Technology push
  - Application Pull

ECE/CS 757: Advanced Computer Architecture II

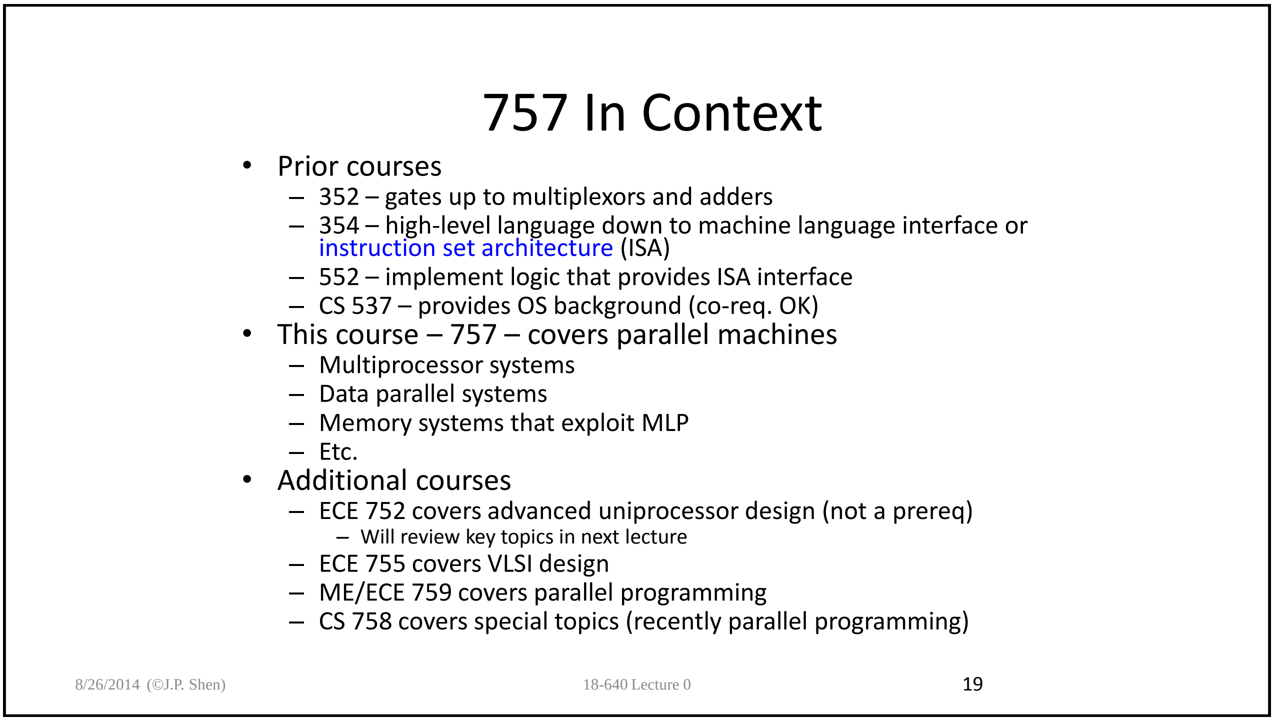
Instructor:Mikko H Lipasti

Spring 2013
University of Wisconsin-Madison

Lecture notes based on slides created by John Shen, Mark Hill, David Wood, Guri Sohi, Jim Smith, Natalie Enright Jerger, Michel Dubois, Murali Annavaram, Per Stenström and probably others

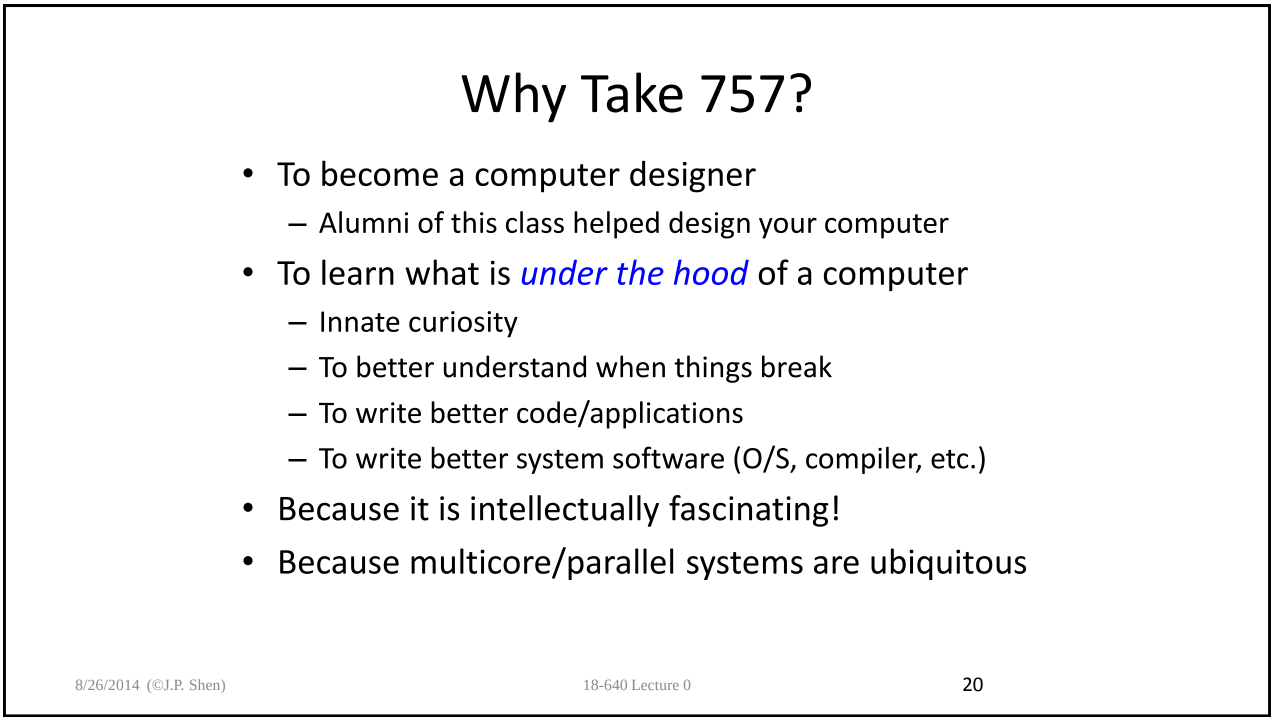
Computer Architecture

- Instruction Set Architecture (IBM 360)
  - *… the attributes of a [computing] system as seen by the programmer. I.e. the conceptual structure and functional behavior, as distinct from the organization of the data flows and controls, the logic design, and the physical implementation. -- Amdahl, Blaaw, & Brooks, 1964*
- Machine Organization (microarchitecture)
  - ALUS, Buses, Caches, Memories, etc.
- Machine Implementation (realization)
  - Gates, cells, transistors, wires



757 In Context

- Prior courses
  - 352 – gates up to multiplexors and adders
  - 354 – high-level language down to machine language interface or instruction set architecture (ISA)
  - 552 – implement logic that provides ISA interface
  - CS 537 – provides OS background (co-req. OK)
- This course – 757 – covers parallel machines
  - Multiprocessor systems
  - Data parallel systems
  - Memory systems that exploit MLP
  - Etc.
- Additional courses
  - ECE 752 covers advanced uniprocessor design (not a prereq)
    - Will review key topics in next lecture
  - ECE 755 covers VLSI design
  - ME/ECE 759 covers parallel programming
  - CS 758 covers special topics (recently parallel programming)



Why Take 757?

- To become a computer designer
  - Alumni of this class helped design your computer
- To learn what is *under the hood* of a computer
  - Innate curiosity
  - To better understand when things break
  - To write better code/applications
  - To write better system software (O/S, compiler, etc.)
- Because it is intellectually fascinating!
- Because multicore/parallel systems are ubiquitous

EE382a Advanced Processor Architecture

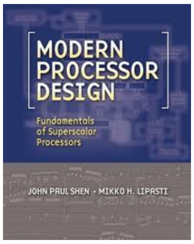
Christos Kozyrakis

http://ee382a.stanford.edu

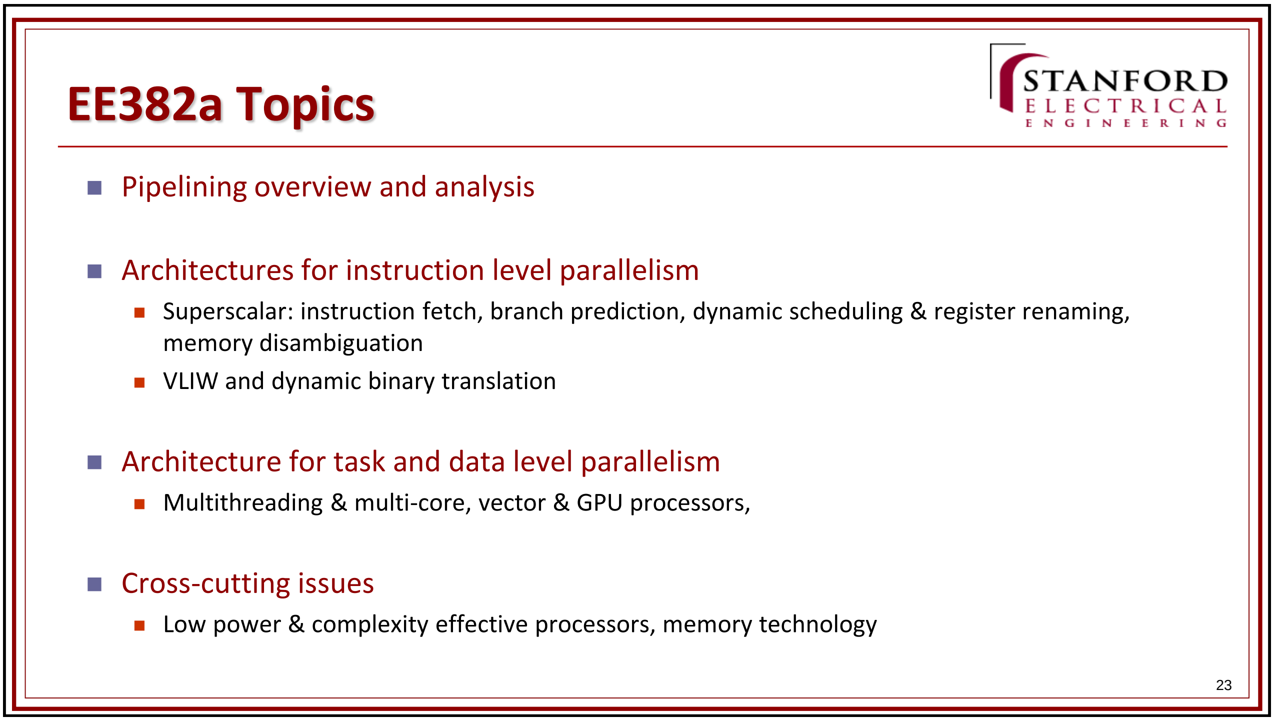
EE382a – Fall 2010 – Lecture 01

Textbooks and Papers

- Textbooks
  - Required: "Modern Processor Design: Fundamentals of Superscalar Processors", J.P. Shen and M. Lipasti, 1st edition, McGraw-Hill
  - Reference: “Computer Organization” and “Computer Architecture” books by D. Patterson & J. Hennessy, both 4th edition, Morgan Kaufmann

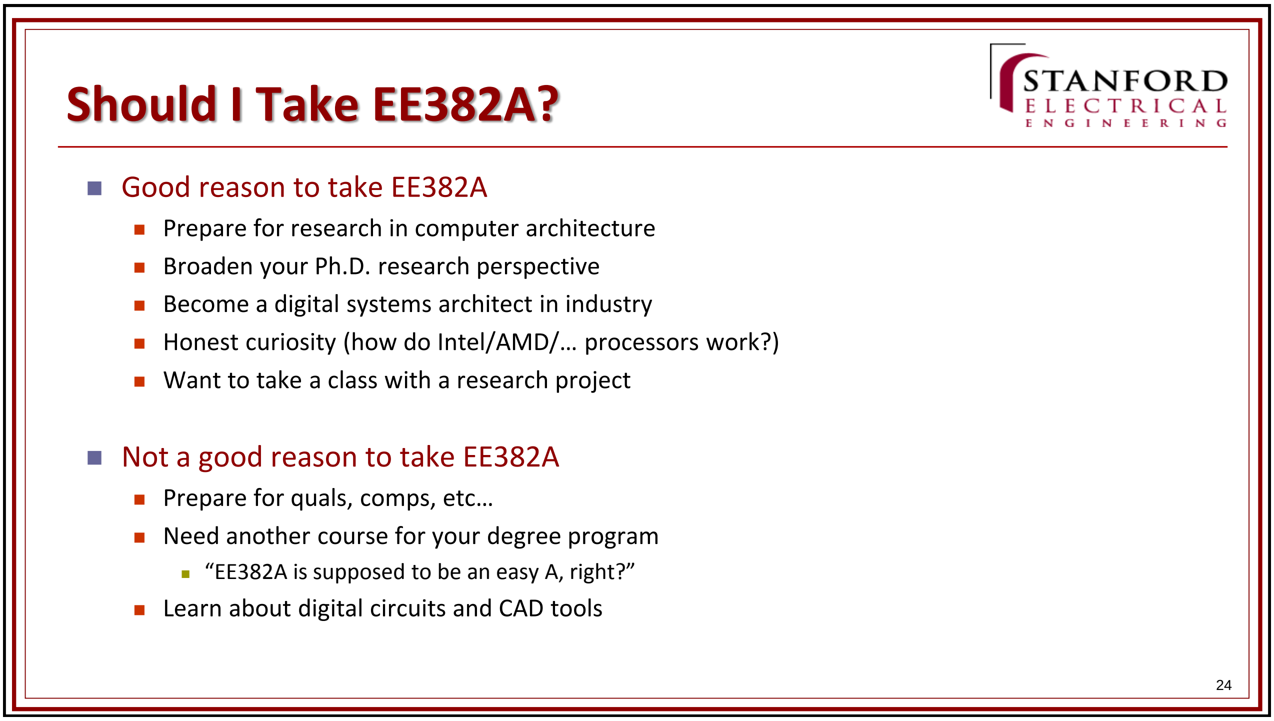

MODERN PROCESSOR DESIGN
Fundamentals of Superscalar Processors
JOHN PAUL SHEN • MIKKO H. LIPASTI

- Papers (check handouts link on the webpage)
  - A few required papers
    - These papers are included in the exam materials
    - Have to submit a 1-page paper summary by next lecture
  - Several optional papers
    - Further in-depth information, references for projects, …



EE382a Topics

- Pipelining overview and analysis
- Architectures for instruction level parallelism
  - Superscalar: instruction fetch, branch prediction, dynamic scheduling & register renaming, memory disambiguation
  - VLIW and dynamic binary translation
- Architecture for task and data level parallelism
  - Multithreading & multi-core, vector & GPU processors,
- Cross-cutting issues
  - Low power & complexity effective processors, memory technology



Should I Take EE382A?

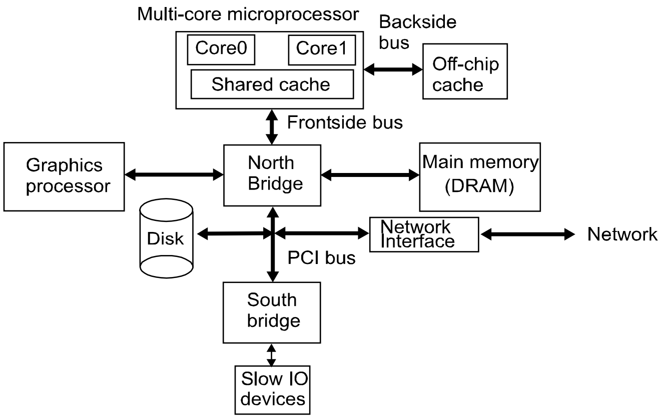
- Good reason to take EE382A
  - Prepare for research in computer architecture
  - Broaden your Ph.D. research perspective
  - Become a digital systems architect in industry
  - Honest curiosity (how do Intel/AMD/… processors work?)
  - Want to take a class with a research project
- Not a good reason to take EE382A
  - Prepare for quals, comps, etc…
  - Need another course for your degree program
    - “EE382A is supposed to be an easy A, right?”
  - Learn about digital circuits and CAD tools

PARALLEL COMPUTER ORGANIZATION AND DESIGN

Michel Dubois*, Murali Annavaram*, Per Stenström[+]
*Department of Electrical Engineering, University of Southern California.
[+]Chalmers University of Technology

COMPUTER ORGANIZATION

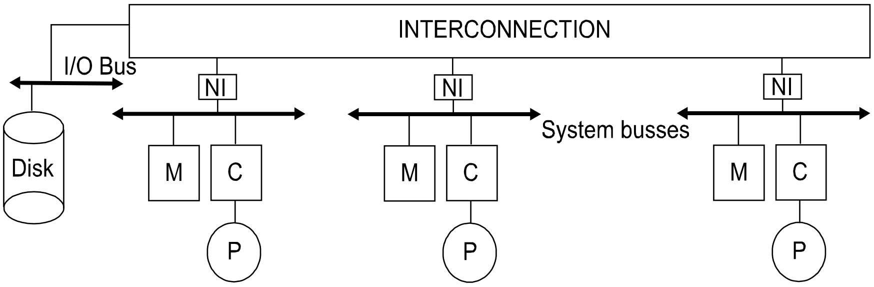
- **MODERN PC ARCHITECTURE**



Multi-core microprocessor
Core0
Core1
Shared cache
Backside bus
Off-chip cache
Frontside bus
Graphics processor
North Bridge
Main memory (DRAM)
Disk
Network Interface
Network
PCI bus
South bridge
Slow IO devices

COMPUTER ORGANIZATION

- **GENERIC HIGH-END PARALLEL SYSTEM:**



- **MAIN COMPONENTS: PROCESSOR, MEMORY SYSTEMS, I/O AND NETWORKS,**

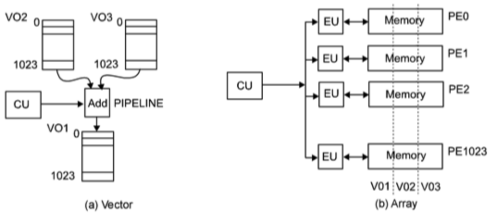
PARALLELISM IN ARCHITECTURES

- **THE MOST SUCCESSFUL MICROARCHITECTURE HAS BEEN THE SCALAR PROCESSOR**
  - A TYPICAL SCALAR INSTRUCTION OPERATES ON SCALAR OPERANDS
    ADD O1,O2,O3 /O2+O3=>O1
  - EXECUTE MULTIPLE SCALAR INSTRUCTIONS AT A TIME
    - PIPELINING
    - SUPERSCALAR
    - SUPERPIPELINING
    - TAKES ADVANTAGE OF **ILP**, I.E., INSTRUCTION-LEVEL PARALLELISM, THE PARALLELISM EXPOSED IN SINGLE THREAD OR SINGLE PROCESS EXECUTION
- **CMPs (CHIP MULTIPROCESSORS) EXPLOITS PARALLELISM EXPOSED BY DIFFERENT THREADS RUNNING IN PARALLEL**
  - THREAD LEVEL PARALLELISM OR TLP
  - CAN BE SEEN AS MULTIPLE SCALAR PROCESSORS RUNNING IN PARALLEL

PARALLELISM IN ARCHITECTURES

- VECTOR AND ARRAY PROCESSORS
  - A TYPICAL VECTOR INSTRUCTION EXECUTES DIRECTLY ON VECTOR OPERANDS

    VADD VO1,VO2,VO3 /VO2+VO3=>VO1
    - VOk IS A VECTOR OF SCALAR COMPONENTS
    - EQUIVALENT TO COMPUTING
      - VO2[i]+VO3[i]=>VO1[i], i=0,..,N
- VECTOR INSTRUCTIONS ARE EXECUTED BY PIPELINES OR PARALLEL ARRAYS



18-640 Foundations of Computer Architecture

Lecture 1:
"Introduction To Computer Architecture"

John Paul Shen
August 27, 2014

Next Time ...

- Required Reading Assignments:
  - **Chapters 1 and 2 of Shen and Lipasti (SnL).**
- Recommended References:
  - "Amdahl's and Gustafson's Laws Revisited" by Andrzej Karbowski. (2008)
  - "High Performance Reduced Instruction Set Processors" by Tilak Agerwala and John Cocke. (1987)

Electrical & Computer ENGINEERING
Carnegie Mellon University