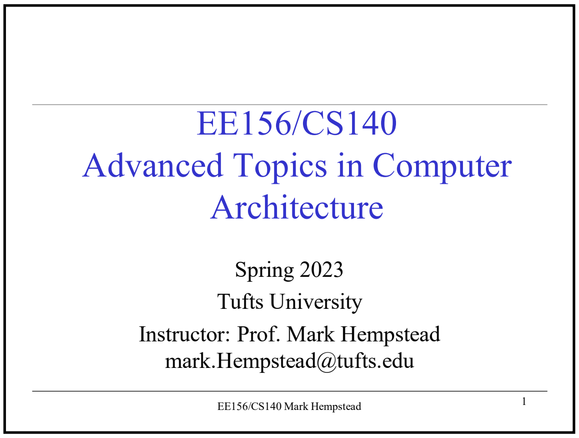
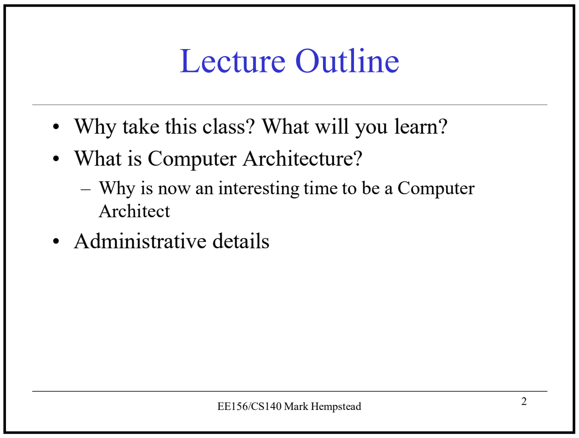


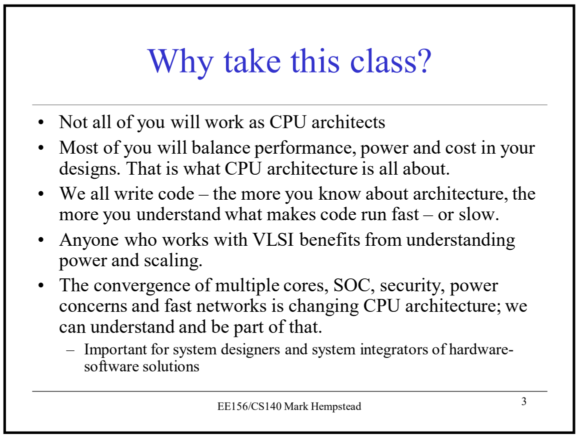
EE156/CS140 Advanced Topics in Computer Architecture

Spring 2023
Tufts University
Instructor: Prof. Mark Hempstead
mark.Hempstead@tufts.edu



Lecture Outline

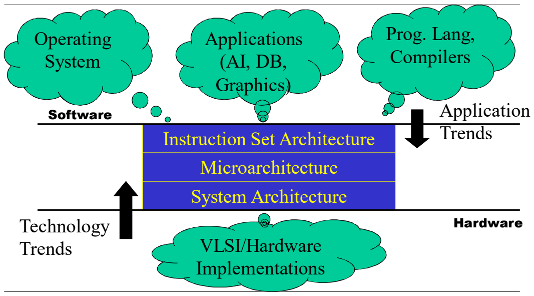
- Why take this class? What will you learn?
- What is Computer Architecture?
  - Why is now an interesting time to be a Computer Architect
- Administrative details



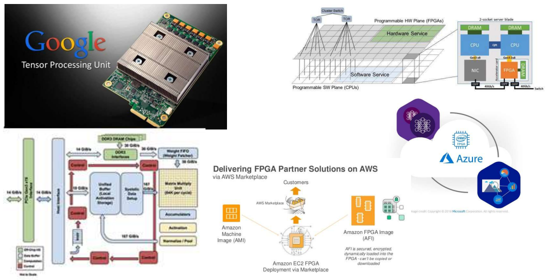
Why take this class?

- Not all of you will work as CPU architects
- Most of you will balance performance, power and cost in your designs. That is what CPU architecture is all about.
- We all write code – the more you know about architecture, the more you understand what makes code run fast – or slow.
- Anyone who works with VLSI benefits from understanding power and scaling.
- The convergence of multiple cores, SOC, security, power concerns and fast networks is changing CPU architecture; we can understand and be part of that.
  - Important for system designers and system integrators of hardware-software solutions

What is Computer Architecture?

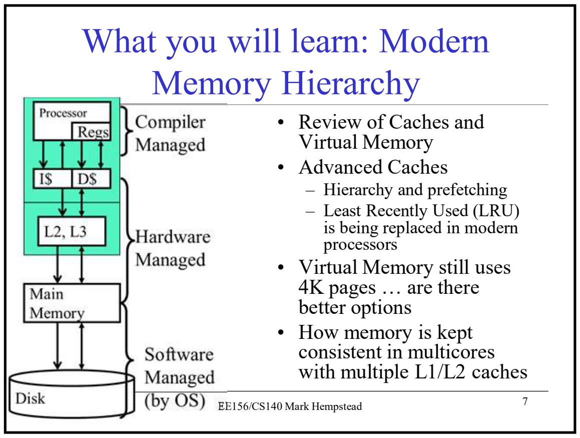


…wait I want to be a software developer why do I need to learn about hardware.



Topics 2022

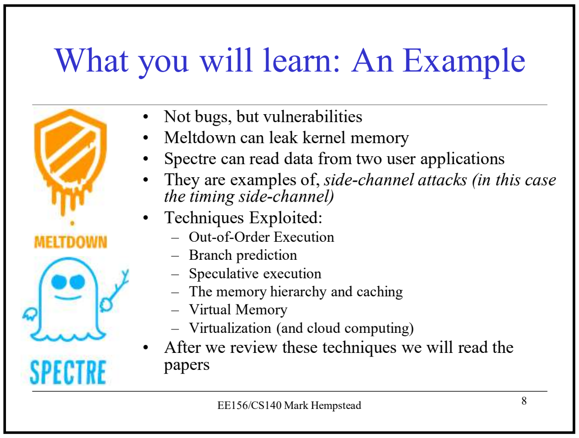
- *Unit 1: The Memory Hierarchy*
- *Unit 2: Microarchitecture and the Pipeline*
- *Unit 3: Security from the Hardware/Systems Perspective*
- *Unit 4: Multicore and Heterogeneous Systems*
- *Unit 5: Power and Energy*
- *Others TBD*


Processor
Regs
I$
D$
L2, L3
Main Memory
Disk
Compiler Managed
Hardware Managed
Software Managed (by OS)

What you will learn: Modern Memory Hierarchy

- Review of Caches and Virtual Memory
- Advanced Caches
  - Hierarchy and prefetching
  - Least Recently Used (LRU) is being replaced in modern processors
- Virtual Memory still uses 4K pages … are there better options
- How memory is kept consistent in multicores with multiple L1/L2 caches

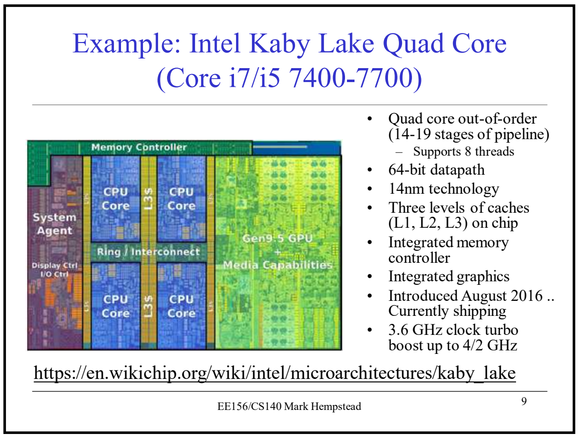
EE156/CS140 Mark Hempstead 7


MELTDOWN
SPECTRE

What you will learn: An Example

- Not bugs, but vulnerabilities
- Meltdown can leak kernel memory
- Spectre can read data from two user applications
- They are examples of, *side-channel attacks (in this case the timing side-channel)*
- Techniques Exploited:
  - Out-of-Order Execution
  - Branch prediction
  - Speculative execution
  - The memory hierarchy and caching
  - Virtual Memory
  - Virtualization (and cloud computing)
- After we review these techniques we will read the papers

EE156/CS140 Mark Hempstead 8


Memory Controller
System Agent
Display Ctrl I/O Ctrl
CPU Core
L3$
CPU Core
Ring / Interconnect
CPU Core
L3$
CPU Core
Gen9.5 GPU + Media Capabilities

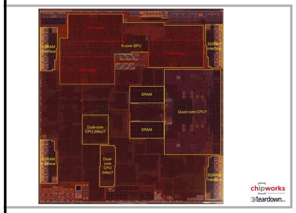
Example: Intel Kaby Lake Quad Core (Core i7/i5 7400-7700)

- Quad core out-of-order (14-19 stages of pipeline)
  - Supports 8 threads
- 64-bit datapath
- 14nm technology
- Three levels of caches (L1, L2, L3) on chip
- Integrated memory controller
- Integrated graphics
- Introduced August 2016 .. Currently shipping
- 3.6 GHz clock turbo boost up to 4/2 GHz

https://en.wikichip.org/wiki/intel/microarchitectures/kaby_lake

EE156/CS140 Mark Hempstead 9

Example Processor: Apple A10 Fusion



- Introduced 2016
  - iPhone 7
- 3.3 Billion Transistors
- 16 nm technology
- Integrated GPU
- 4 cores
  - 2 high power 2.34 GHz ARMv8-A cores
  - 2 Energy-efficient cores
- Significant number of specialized cores
  - Video, imaging, regEx, crypto

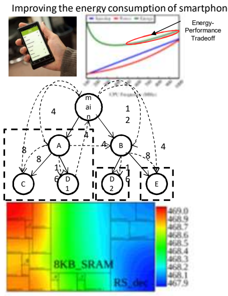
Instructor

- Instructor: Mark Hempstead (mark.hempstead@tufts.edu), CLIC 318 J (574 Boston Ave. New Office!!!)
- Canvas will be used as the course website
  - Lecture Slides, Assignments and Calendar
- My Background
  - BS in Computer Engineering from Tufts 2003
  - PhD at Harvard June 2009
  - Research Intern at Intel
  - Researcher ARM R&D in Cambridge UK
  - Recently spent a year at Facebook FAIR SysML Research Group
- Office Hours:
  - Mondays 1:30 – 3pm (Virtual and In-Person)
  - Wednesdays 11:45 – 1pm (In-Person walking from class)

Prof. Mark Hempstead
Associate Professor
Electrical and Computer Engineering

Tufts Computer Architecture Lab
https://sites.tufts.edu/tcal/

Improving the energy consumption of smartphones



- Security and Implications of Thermal/Power Side-Channels for Many Accelerator SoCs
- Microprocessor Modeling Hotspots
  - Building models for 14nm, 10nm, 7nm process tech
- Systems for Machine learning
  - Recommendation Systems
  - Near Memory Computing
- Mobile Computing
  - Power-Agile DVFS
  - Improving Smartphone Thermal Management
- Workload Characterization
  - SynchroTrace: Multi-Threaded Simulation
  - PRISIM (formally Sigil)
- Shared Accelerators
  - Early Identification of common hardware kernels
  - Using ASTs and then interface with HLS
- Next Generation Memory Systems
  - Cache replacement and cache partitioning for QoS
  - Characterization of different NVM technologies in LLC

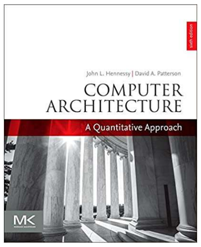
Tufts University | School of Engineering Electrical and Computer Engineering

Teaching Assistants

- Parnian Mokri Parnian.Mokri@tufts.edu
  - Thurdays XXXXX CLIC Room 310, 574 Boston Ave
- Robert Costa Robert_J.Costa@tufts.edu
  - Fridays 4-5pm on Zoom
- Bharat Kesari Bharat.Kesari@tufts.edu
  - Tuesdays 2-3 on Zoom
- If On Zoom (see Canvas for links)

Resources

- Text: “Computer Architecture: A Quantitative Approach,” **Sixth Edition**, Hennessy and Patterson

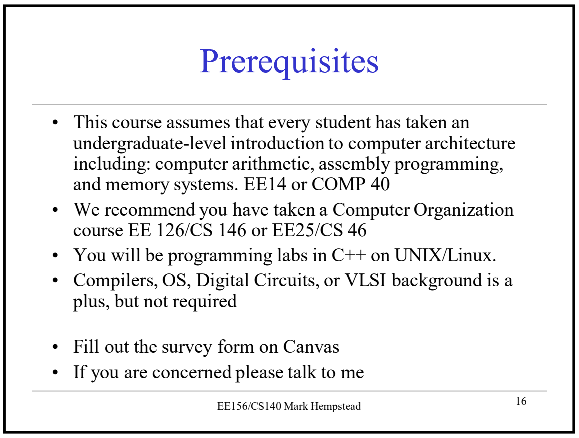


- Older editions are available in the library, but they use the MIPS ISA instead of RISC-V
- **Otherwise the key concepts are similar**

Resources: Modern Readings

- We will read research papers that serve as primary sources to the topics we are studying
  - Papers available on Canvas
  - General Approach: Something Old and Something New
- Because the textbook cannot go into each new area in detail we will also reference the computer architecture lecture series (free on Tufts VPN)
  - http://www.morganclaypool.com/toc/cac/1/1
  - One assignment for project groups will be to explain a new area to the class:

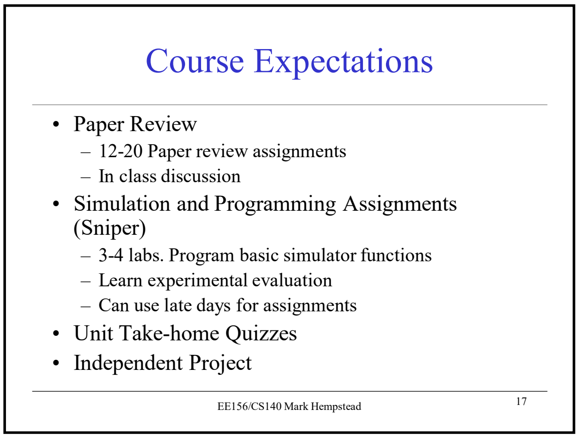
M&C Computer Architecture



Prerequisites

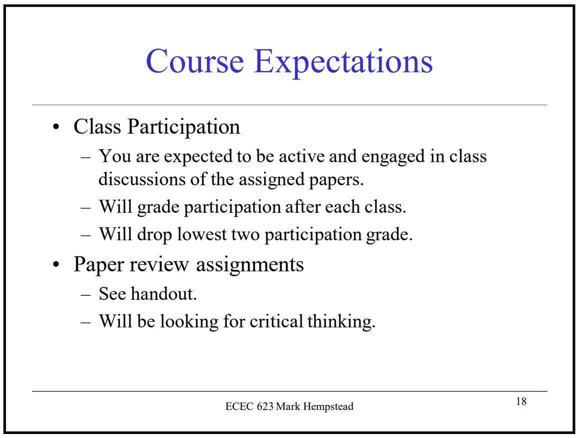
- This course assumes that every student has taken an undergraduate-level introduction to computer architecture including: computer arithmetic, assembly programming, and memory systems. EE14 or COMP 40
- We recommend you have taken a Computer Organization course EE 126/CS 146 or EE25/CS 46
- You will be programming labs in C++ on UNIX/Linux.
- Compilers, OS, Digital Circuits, or VLSI background is a plus, but not required

- Fill out the survey form on Canvas
- If you are concerned please talk to me



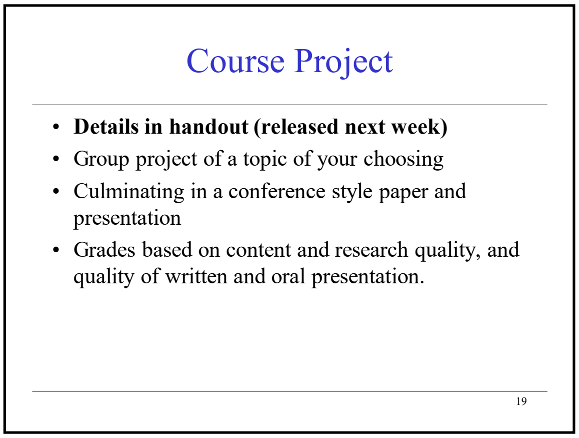
Course Expectations

- Paper Review
  - 12-20 Paper review assignments
  - In class discussion
- Simulation and Programming Assignments (Sniper)
  - 3-4 labs. Program basic simulator functions
  - Learn experimental evaluation
  - Can use late days for assignments
- Unit Take-home Quizzes
- Independent Project



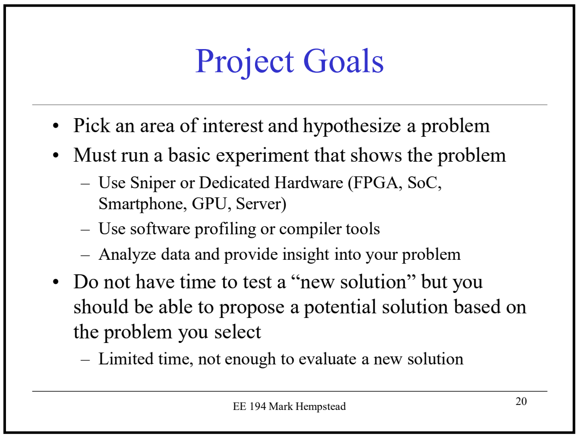
Course Expectations

- Class Participation
  - You are expected to be active and engaged in class discussions of the assigned papers.
  - Will grade participation after each class.
  - Will drop lowest two participation grade.
- Paper review assignments
  - See handout.
  - Will be looking for critical thinking.



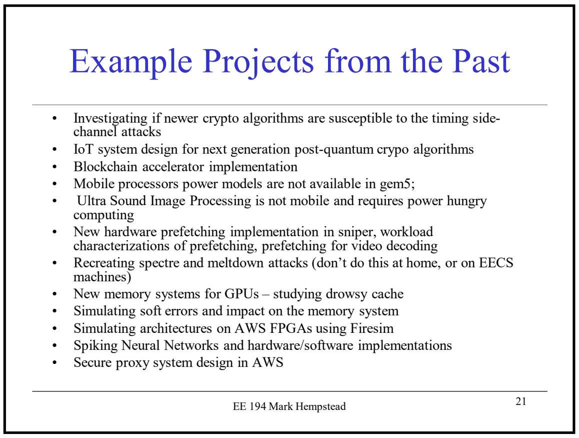
Course Project

- **Details in handout (released next week)**
- Group project of a topic of your choosing
- Culminating in a conference style paper and presentation
- Grades based on content and research quality, and quality of written and oral presentation.



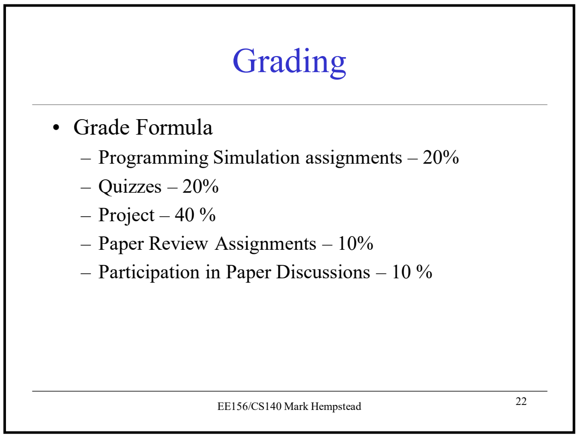
Project Goals

- Pick an area of interest and hypothesize a problem
- Must run a basic experiment that shows the problem
  - Use Sniper or Dedicated Hardware (FPGA, SoC, Smartphone, GPU, Server)
  - Use software profiling or compiler tools
  - Analyze data and provide insight into your problem
- Do not have time to test a "new solution" but you should be able to propose a potential solution based on the problem you select
  - Limited time, not enough to evaluate a new solution



Example Projects from the Past

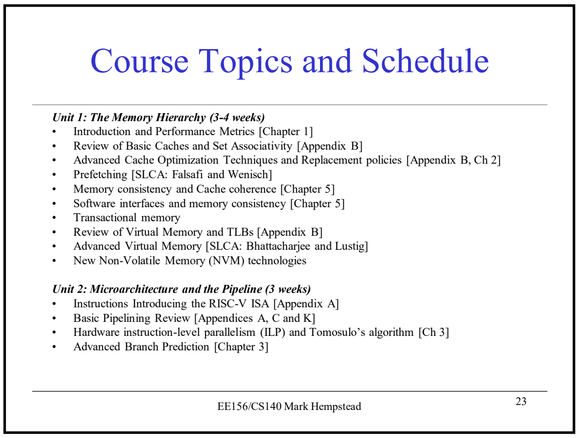
- Investigating if newer crypto algorithms are susceptible to the timing side-channel attacks
- IoT system design for next generation post-quantum crypo algorithms
- Blockchain accelerator implementation
- Mobile processors power models are not available in gem5;
- Ultra Sound Image Processing is not mobile and requires power hungry computing
- New hardware prefetching implementation in sniper, workload characterizations of prefetching, prefetching for video decoding
- Recreating spectre and meltdown attacks (don't do this at home, or on EECS machines)
- New memory systems for GPUs – studying drowsy cache
- Simulating soft errors and impact on the memory system
- Simulating architectures on AWS FPGAs using Firesim
- Spiking Neural Networks and hardware/software implementations
- Secure proxy system design in AWS



Grading

- Grade Formula
  - Programming Simulation assignments – 20%
  - Quizzes – 20%
  - Project – 40 %
  - Paper Review Assignments – 10%
  - Participation in Paper Discussions – 10 %

EE156/CS140 Mark Hempstead 22



Course Topics and Schedule

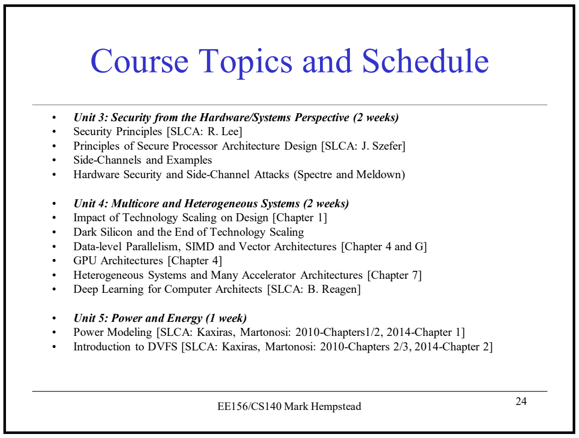
Unit 1: The Memory Hierarchy (3-4 weeks)

- Introduction and Performance Metrics [Chapter 1]
- Review of Basic Caches and Set Associativity [Appendix B]
- Advanced Cache Optimization Techniques and Replacement policies [Appendix B, Ch 2]
- Prefetching [SLCA: Falsafi and Wenisch]
- Memory consistency and Cache coherence [Chapter 5]
- Software interfaces and memory consistency [Chapter 5]
- Transactional memory
- Review of Virtual Memory and TLBs [Appendix B]
- Advanced Virtual Memory [SLCA: Bhattacharjee and Lustig]
- New Non-Volatile Memory (NVM) technologies

Unit 2: Microarchitecture and the Pipeline (3 weeks)

- Instructions Introducing the RISC-V ISA [Appendix A]
- Basic Pipelining Review [Appendices A, C and K]
- Hardware instruction-level parallelism (ILP) and Tomosulo's algorithm [Ch 3]
- Advanced Branch Prediction [Chapter 3]

EE156/CS140 Mark Hempstead 23



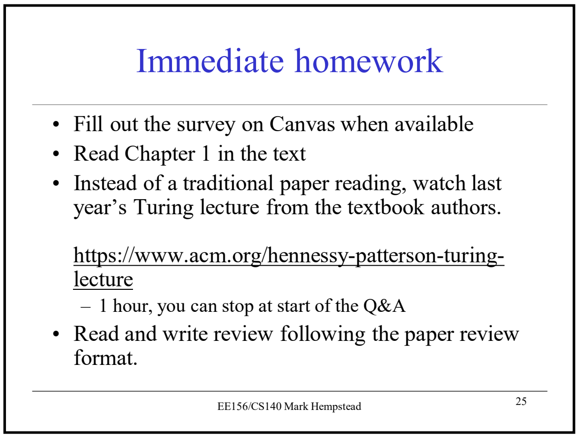
Course Topics and Schedule

- ***Unit 3: Security from the Hardware/Systems Perspective (2 weeks)***
- Security Principles [SLCA: R. Lee]
- Principles of Secure Processor Architecture Design [SLCA: J. Szefer]
- Side-Channels and Examples
- Hardware Security and Side-Channel Attacks (Spectre and Meldown)

- ***Unit 4: Multicore and Heterogeneous Systems (2 weeks)***
- Impact of Technology Scaling on Design [Chapter 1]
- Dark Silicon and the End of Technology Scaling
- Data-level Parallelism, SIMD and Vector Architectures [Chapter 4 and G]
- GPU Architectures [Chapter 4]
- Heterogeneous Systems and Many Accelerator Architectures [Chapter 7]
- Deep Learning for Computer Architects [SLCA: B. Reagen]

- ***Unit 5: Power and Energy (1 week)***
- Power Modeling [SLCA: Kaxiras, Martonosi: 2010-Chapters1/2, 2014-Chapter 1]
- Introduction to DVFS [SLCA: Kaxiras, Martonosi: 2010-Chapters 2/3, 2014-Chapter 2]

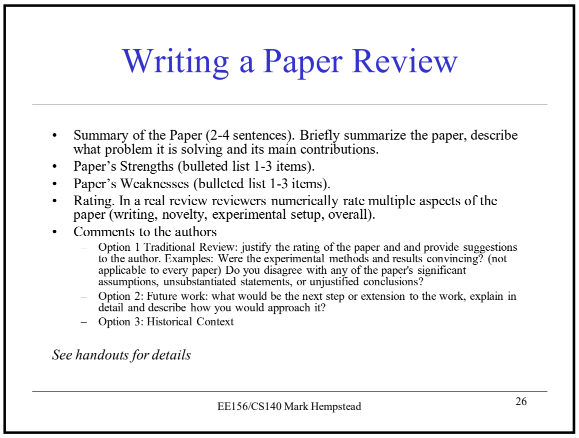
EE156/CS140 Mark Hempstead 24



Immediate homework

- Fill out the survey on Canvas when available
- Read Chapter 1 in the text
- Instead of a traditional paper reading, watch last year’s Turing lecture from the textbook authors.

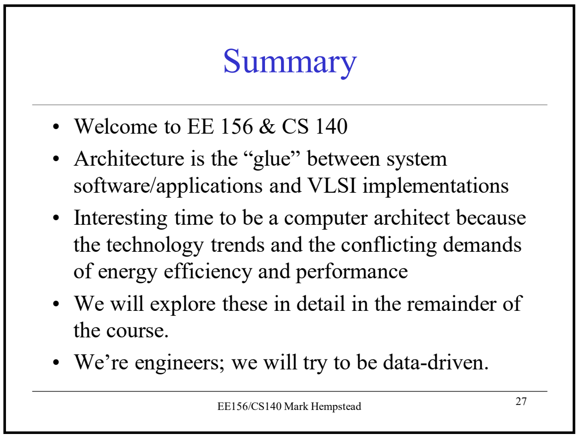
  https://www.acm.org/hennessy-patterson-turing-lecture
  - 1 hour, you can stop at start of the Q&A
- Read and write review following the paper review format.



Writing a Paper Review

- Summary of the Paper (2-4 sentences). Briefly summarize the paper, describe what problem it is solving and its main contributions.
- Paper’s Strengths (bulleted list 1-3 items).
- Paper’s Weaknesses (bulleted list 1-3 items).
- Rating. In a real review reviewers numerically rate multiple aspects of the paper (writing, novelty, experimental setup, overall).
- Comments to the authors
  - Option 1 Traditional Review: justify the rating of the paper and and provide suggestions to the author. Examples: Were the experimental methods and results convincing? (not applicable to every paper) Do you disagree with any of the paper's significant assumptions, unsubstantiated statements, or unjustified conclusions?
  - Option 2: Future work: what would be the next step or extension to the work, explain in detail and describe how you would approach it?
  - Option 3: Historical Context

See handouts for details



Summary

- Welcome to EE 156 & CS 140
- Architecture is the “glue” between system software/applications and VLSI implementations
- Interesting time to be a computer architect because the technology trends and the conflicting demands of energy efficiency and performance
- We will explore these in detail in the remainder of the course.
- We’re engineers; we will try to be data-driven.