



ECE4100/ECE6100/CS4290/CS6290 Advanced Computer Architecture Fall 2019

http://tusharkrishna.ece.gatech.edu/teaching/aca_f19/

Lecture 1: Introduction

Tushar Krishna

School of Electrical and Computer Engineering
Georgia Institute of Technology

tushar@ece.gatech.edu

Acknowledgment:

Lecture slides adapted from MIT EECS 6.823 (Arvind and J. Emer) and GT ECE 4100/6100 (M. Qureshi)



Background


- PhD from MIT in EECS (2013)
- Researcher at Intel (2014-15)
- Georgia Tech (2015 - present)

Office: Klaus 2318

Office Hours:

Tu & Th 4:15 – 5:00 PM after class

Fr: By Prior Appointment

Georgia Tech  **School of Electrical and Computer Engineering**
College of Engineering

TUSHAR KRISHNA
Assistant Professor

School of ECE
Georgia Institute of Technology
Atlanta, GA 30332, USA

EMAIL: tushar@ece.gatech.edu

WEB: <http://tusharkrishna.ece.gatech.edu>

Research Interests

- Computer Architecture
- Interconnection Networks
- Network-on-Chip
- Deep Learning Accelerators

Course Staff

3

■ Teaching Assistants (TAs)

- Abhinav Himanshu
- Email: ahimanshu3@gatech.edu
- Office Hours: TBA



Course Information and Updates

4

■ Course Website for Schedule

- http://tusharkrishna.ece.gatech.edu/teaching/aca_f19

■ Canvas

- Announcements
- Lecture notes
- Lab Assignments

■ Piazza

- Post common questions on Piazza instead of emailing TAs/me.
 - Related to the labs/lectures/homeworks.
- Encouraged for common questions
- Try to answer each other's posted questions, if you can
 - Otherwise TAs and/or I will respond (in some time)

■ Email

- Me or TAs directly **only** if you have any question that are not suited for public forum such as Piazza (E.g., why did I get only 20 pts?)

What is Computer Architecture?

5

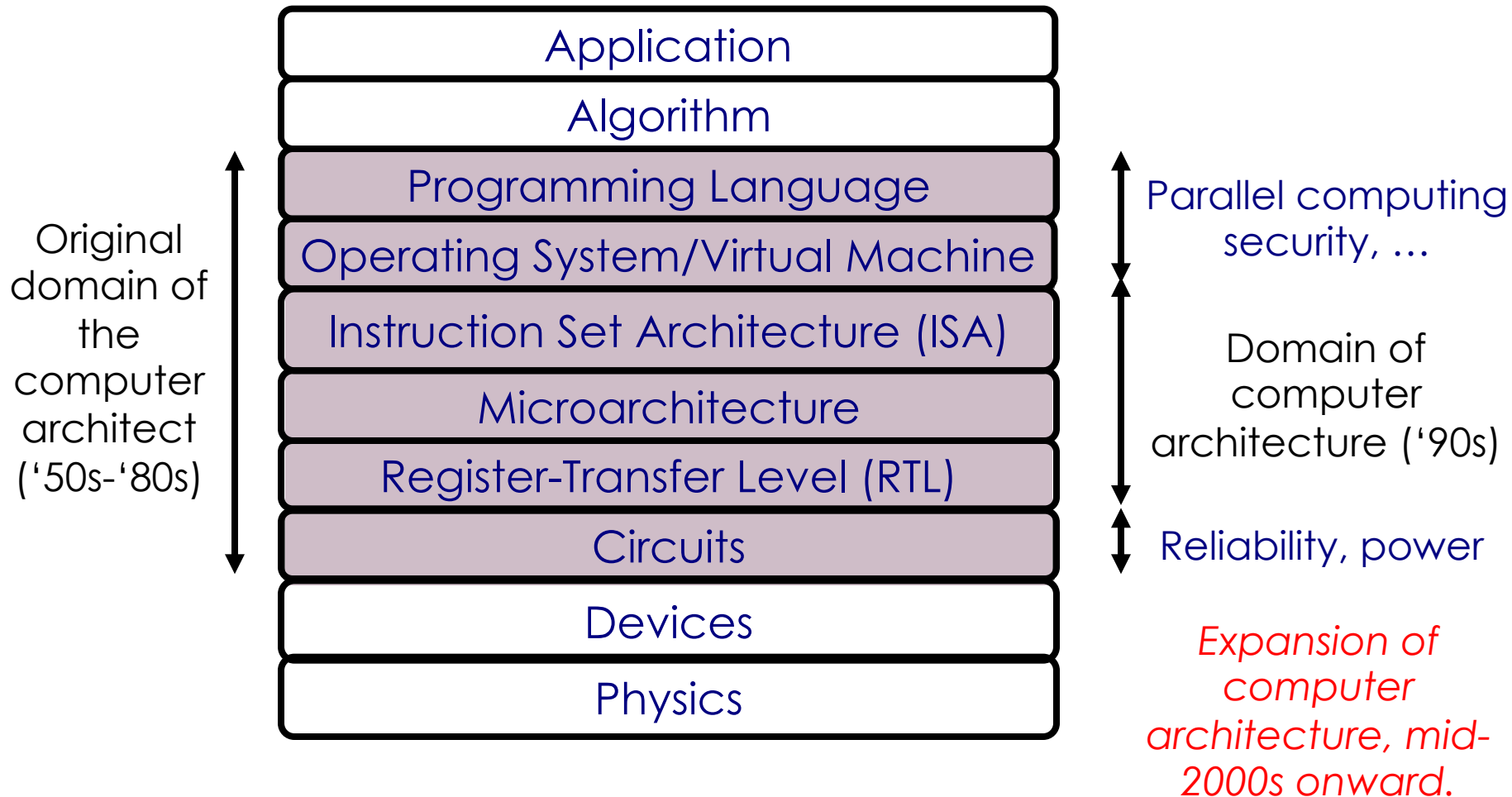
What is Computer Architecture?

6

- **Wide Dynamic Execution**
 - enables the delivery of more instructions per clock cycle
- **Hyper-Threading Technology.**
 - each core processes two application “threads” simultaneously
- **HD Boost.**
 - significant gains on the latest SSE4 instruction set.
- **Turbo Boost Technology.**
 - increases the processor’s frequency when needed
- **True quad-core**
 - enables cores to communicate at die level.
- **8 MB Shared Smart Cache.**
 - enabling multiple cores to dynamically share this space
- **Smart Memory Access**
 - increasing available data bandwidth
- **Intelligent Power Capability**
 - turning off portions of the processor when they aren't being used

**some marketing
buzz words**

Defining Computer Architecture



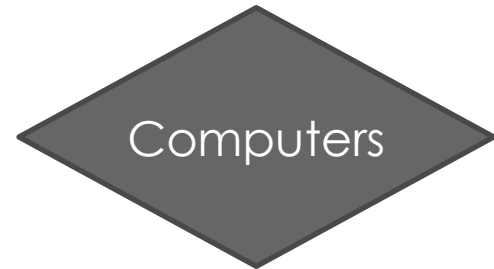
Computer Architecture is the design of abstraction layers

- What do abstraction layers provide?
 - Environmental stability within generation
 - Environmental stability across generations
 - Consistency across a large number of units

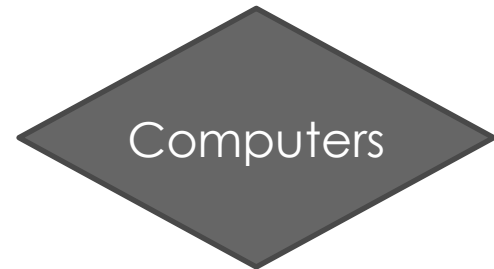
- What are the consequences?
 - *Encouragement to create reusable foundations:*
 - *Tool chains, operating systems, libraries*
 - Enticement for application innovation

Technology is the dominant factor in computer design

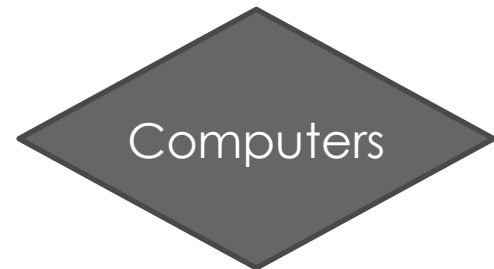
Technology
Transistors
Integrated circuits
VLSI (initially)
Flash memories, ...



Technology
Core memories
Magnetic tapes
Disks



Technology
ROMs, RAMs
VLSI
Packaging
Low Power

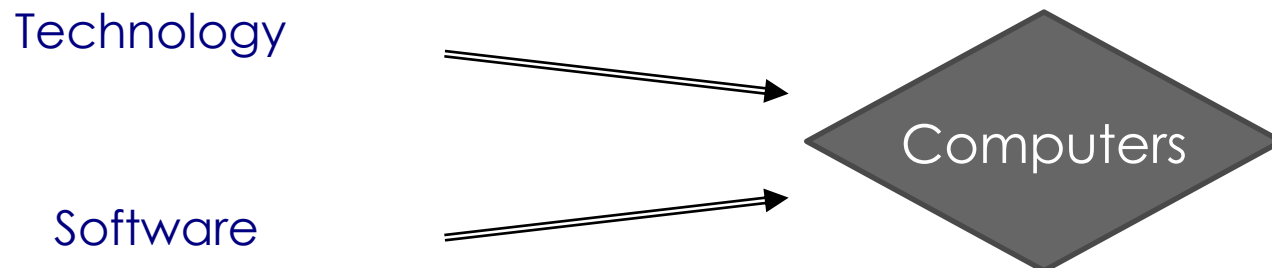


What about software?

As people write programs and use computers, our understanding of *programming* and *program behavior* improves.

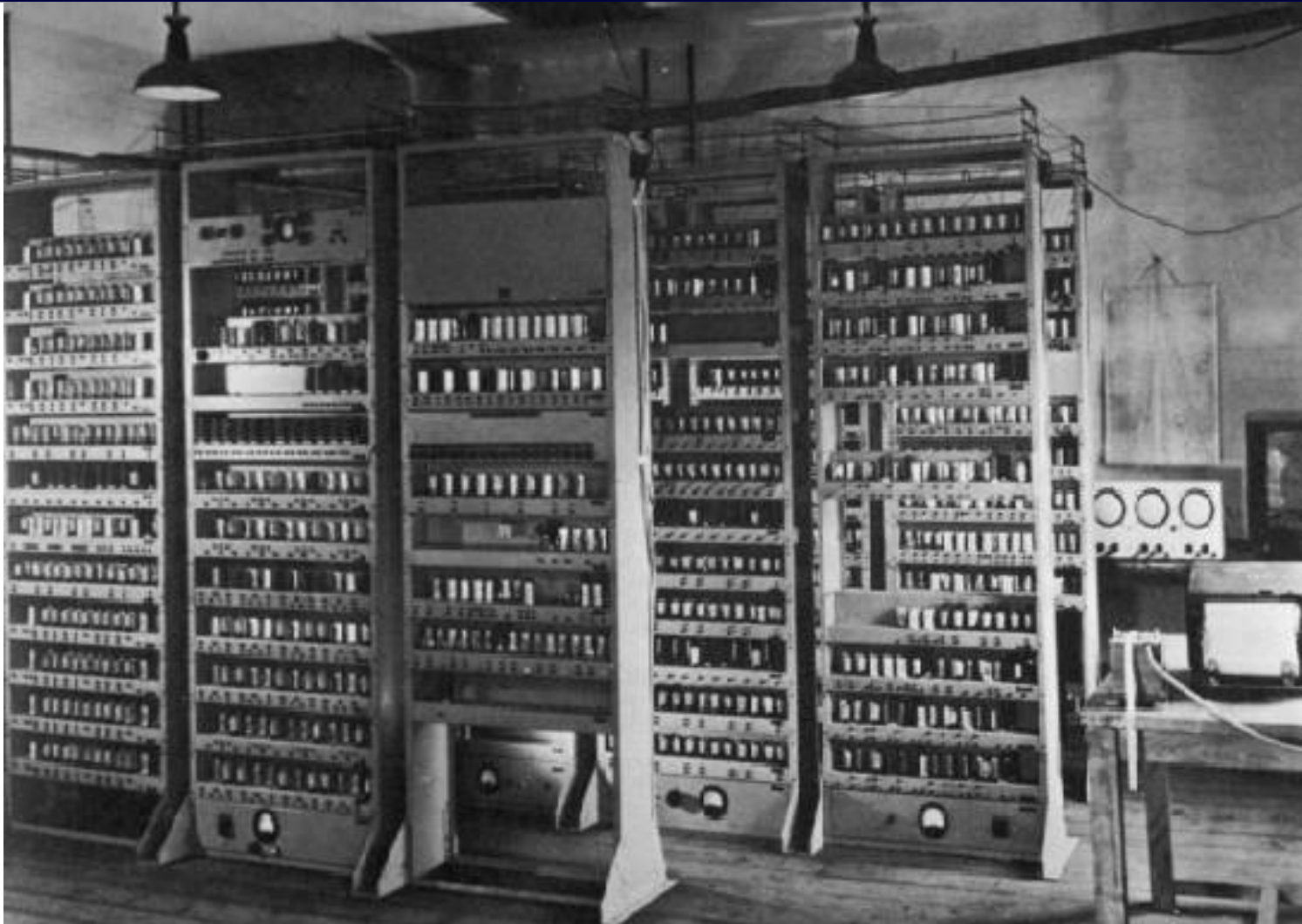
This has profound though slower impact on computer architecture

Modern architects cannot avoid paying attention to software and compilation issues.



Computing devices then...

11



Computing devices now

12



Classes of computers

- **Embedded**

- Price: \$10 - \$100K
- **Constraints:** price, energy, application-specific performance

- **Mobile** (smartphones/tablets)

- Price: \$100-\$1000
- **Constraints:** cost, energy, media performance, responsiveness

- **Desktop**

- Price: \$300-\$2500
- **Constraints:** price-performance, energy, graphics performance

- **Server**

- Price: \$5000 - \$10M
- **Constraints:** throughput, availability, scalability, energy

- **Warehouse-scale**

- Price: \$100K - \$200M
- **Constraints:** price-performance, throughput, energy

Architecture is engineering design under **constraints**

Factors to consider:

- **Performance** of whole system on target applications
 - Average case & worst case
- **Cost of manufacturing** chips and supporting system
- **Cost to design** chips (engineers, computers, CAD tools)
 - Becoming a limiting factor in many situations, fewer unique chips can be justified
- **Cost to develop applications** and system **software**
 - Often the dominant constraint for any programmable device
- **Power** to run system
 - Peak power & energy per operation
- **Reliability** of system
 - Soft errors & hard errors

At different times, and for different applications at the same point in time, the relative balance of these factors can result in widely varying architectural choices

- **“A New Golden Age for Computer Architecture”**
 - Hennessey and Patterson, Turing Award Lecture 2019
- **Challenges**
 - End of Transistor Scaling
 - Slowdown of Single-Thread Performance-scaling
 - High energy/power consumption
 - Low Reliability of Transistors
- **Opportunities**
 - Domain-specific Architectures
 - HW-SW Co-Design
 - Novel Compute and Memory Technologies

E.g., Computer Architecture for AI

The technology stack for artificial intelligence (AI) contains nine layers.

| Technology | Stack | Definition |
|------------|------------------------------|--|
| Services | Solution and use case | Integrated solutions that include training data, models, hardware, and other components (eg, voice-recognition systems) |
| Training | Data types | Data presented to AI systems for analysis |
| Platform | Methods | Techniques for optimizing weights given to model inputs |
| | Architecture | Structured approach to extract features from data (eg, convolutional or recurrent neural networks) |
| | Algorithm | A set of rules that gradually modifies the weights given to certain model inputs within the neural network during training to optimize inference |
| | Framework | Software packages to define architectures and invoke algorithms on the hardware through the interface |
| Interface | Interface systems | Systems within framework that determine and facilitate communication pathways between software and underlying hardware |
| Hardware | Head node | Hardware unit that orchestrates and coordinates computations among accelerators |
| | Accelerator | Silicon chip designed to perform highly parallel operations required by AI; also enables simultaneous computations |

Memory

- Electronic data repository for short-term storage during processing
- Memory typically consists of DRAM¹

Storage

- Electronic repository for long-term storage of large data sets
- Storage typically consists of NAND²

Logic

- Processor optimized to calculate neural network operations, ie, convolution and matrix multiplication
- Logic devices are typically CPU, GPU, FPGA, and/or ASIC³

Networking

- Switches, routers, and other equipment used to link servers in the cloud and to connect edge devices

Source: McKinsey, "Artificial-intelligence hardware: New opportunities for semiconductor companies"

A journey through this space

17

- Learn about the evolution of architectures, via historical examples
 - Prehistory: Babbage and Analytic Engine
 - Early days: ENIAC, EDVAC and EDSAC
 - Arrival of IBM 650 and then IBM 360
 - Seymour Cray – CDC 6600, Cray 1
 - Microprocessors and PCs
 - Multicores
 - Embedded Processors
 - Custom Hardware (e.g., for AI)
- Focus on ideas, mechanisms and principles, especially those that have withstood the test of time

■ **Module 1: Processors**

- Review – ISA, Simple Pipelining and Hazards
- Branch Prediction
- Superscalar
- Out of Order Execution
- Speculative Execution

■ **Module 2: Memory**

- Review - Caches
- DRAM
- Virtual Memory

■ **Module 3: Multiprocessors**

- Chip Multi-Processors
- Networks-on-Chip
- Cache Coherence
- Memory Consistency

■ **Module 4: Additional Optimizations**

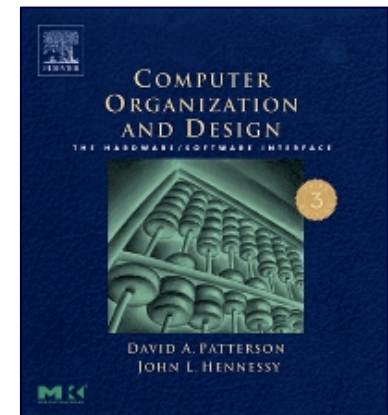
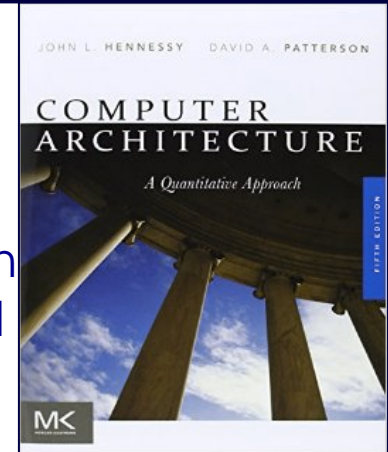
- Multi-threading
- Vector machines/GPUs
- Dataflow architectures/TPU
- Heterogeneous

Textbook and Readings

19

- “Computer Architecture: A Quantitative Approach”, Hennessy & Patterson, 5th Edition
 - Strongly Recommended (but not necessary)
 - **Course** website will list H&P reading material for each lecture, and optional readings for more background and in-depth coverage

- Prerequisite: ECE 3056 or equivalent
 - Appendix A, B, and C of “Computer Architecture: A Quantitative Approach”, H&P, 5th Edition cover required background



- **Digital Logic**
 - Finite State Machines
 - Combinational Logic vs. Sequential Logic
 - Operation of Muxes, Decoders, Encoders, ...
- **5-stage Processor Pipeline**
 - RAW, WAR, WAW Hazards
- **Caches**
 - Direct-Mapped
 - Set-Associative
 - Fully-Associative
- **We will not be covering these in class.**
 - If you do not have the right background, make sure you take the pre-requisite class first (ECE 3056)
 - If you have forgotten these topics, review them from Hennessy & Patterson

| Item | Percentage |
|-----------------|-------------|
| Lab1 | 5% |
| Lab2 | 10% |
| Lab3 | 10% |
| Lab4 | 10% |
| HW1 | 2% |
| HW2 | 2% |
| Participation | 1% |
| SubTotal | 40% |
| Midterm1 | 20% |
| Midterm2 | 20% |
| Final Exam | 20% |
| Total | 100% |

***Overall grading will be relative.
About 50% of you will get an A.***

Two Midterms and One Final

22

■ **Midterms**

- 1 hr 20 min
- In class, during lecture hours
- Material up to the end of the previous week will be covered
 - Midterm during Week N (Tu/Th) will cover material taught up to Week (N-1)
- Midterm 2 will not cover topics already covered by Midterm 1

■ **Final**

- In class during Finals Week
- 2hr 40 min
- Will cover topics from the entire semester

■ **There will be no make up exams!**

- Emergency situations with supporting and timely paperwork can be taken up on a case-by-case basis

■ Lab 1 already posted!

- Based on basic knowledge of instructions and CPI
- Tests ability to work with architecture simulators written in C/C++
- This is a touchstone for your pre-requisite
- **Due this Friday at 1pm** (so that you can drop the course if you find that you do not have the right background).
- **Get started early!**
 - You may not have the set up to do the work on the reference machine (ecelinsrv7.ece.gatech.edu)
 - No late submission accepted.

■ Labs 2 to 4 build a processor and memory system model

- Due on Fridays by 11:55 pm.
 - 3 hour grace period allowed (to account for Canvas issues).
 - One day late submission allowed at the cost of 2 points.
 - **Beyond Saturday 11:55 pm, no more submissions will be accepted!**

Three Homeworks

■ Homework 0: self test

- Questions covering basics of digital logic and computer architecture. Fill a table at the end
 - 0: Never seen this material before
 - 1: Used to know it
 - 2: Know it
- Will not be graded. No need to submit.
- We expect a 2 for all questions
 - Brush up background on topics where you have a zero.
 - If you have 6-7 zeros, talk to the TAs/me.

■ Homework 1 and 2: problem sets to prepare for midterms

- Due a few days before the midterm
 - You will get 2 points if you submit a reasonably attempted version
 - Solutions will also be posted after the due date
- No late submissions accepted!

- Slides will be posted on Canvas a day or two in advance
- For certain topics, supplementary reading materials (recent research papers etc) will also be posted

- TAs will hold **ONE** recitation session every week
 - **Cover material complementary to lectures**
 - walk-through examples of difficult concepts from lectures
 - review items that lot of students are discussing via Piazza/email
 - review common questions from the Lab assignments
 - model questions to help prepare for midterms and final
 - **Make sure you attend recitation before midterms!**
 - Topic of recitation will be emailed every week
 - Day, times and venue to be announced soon
- Recitations are **optional**, but highly **useful**
 - Individual engagement not possible during lecture
 - Safe space for specific questions and clarifications
 - Test your understanding vis-à-vis your classmates

Office Hours

27

- TAs will hold 1-hour office hours each every week, in addition to the recitation(s)
 - Use these for specific questions/concerns about the lectures or labs
- I will hold office hours after every class
 - Send me an email to setup an additional meeting time if required

Schedule (tentative)

| Week | Dates | Tuesday | Thursday | Due Dates [Fri] |
|------|-------------|-------------|--------------|----------------------|
| 1 | (Aug 20 -) | | | Lab #1 Due |
| 2 | (Aug 27 -) | | | |
| 3 | (Sep 3 -) | | | |
| 4 | (Sep 10 -) | | | |
| 5 | (Sep 17 -) | | | HW#1 Due, Lab #2 Due |
| 6 | (Sep 24 -) | | Midterm 1 | |
| 7 | (Oct 1 -) | | | |
| 8 | (Oct 8 -) | | | |
| 9 | (Oct 15 -) | Fall Recess | | |
| 10 | (Oct 22 -) | | | Lab #3 Due |
| 11 | (Oct 29 -) | Midterm 2 | | HW#2 Due |
| 12 | (Nov 5 -) | | | |
| 13 | (Nov 12 -) | | | |
| 14 | (Nov 19 -) | | Thanksgiving | |
| 15 | (Nov 26 -) | | | Lab#4 Due |
| 16 | (Dec 3 -) | | | |
| 17 | (Dec 10 -) | Final | | |

Warning!!

- This course requires heavy programming
- Don't take too many program/project heavy courses together!
- It is a 3-credit course but will feel like a 4-5 credit course
- The most ECE-like course in CS, the most CS-like course in ECE



Zero tolerance for cheating

30

- **ALL lab assignments are individual**
 - You can discuss ideas with other students
 - You CANNOT see (or show) other students code
 - We use MOSS to detect cases of substantial overlap
<http://theory.stanford.edu/~aiken/moss/>
- Zero tolerance towards violation of the GT honor code
 - If you are caught cheating: Zero on lab assignment + One grade drop + Report to dean (academic warning in file)



What is expected from you?

31

- Required background
 - Basic computer architecture (ECE 3056 or equivalent)
 - Basic programming (C/C++)
- Learn the material, understand it and analyze it
- Do the work & **work hard**
- Do the lab programming assignments
- **Ask questions, take notes, participate: If we are not discussing, then one way lecture will be boring (for both you and me)**
- We will have a “No Open Screens” policy in this class

“Electronic Etiquette Policy”

32

- No open Laptops, Tablets, Phone etc. in lectures!
 - Several studies [Princeton and U California] show Open Laptops are a hindrance to classroom learning: test scores of students with open Laptop substantially lower than the students with closed screens
<http://www.newyorker.com/tech/elements/the-case-for-banning-laptops-in-the-classroom> (6/6/2014)
 - The open screens affect
 - You (hard to be a part of the discussion if your attention is on your screen)
 - Your fellow students (who may get distracted by your videos of cute kittens)
 - Professor (more motivated if students are paying attention)
 - *If your screens/texting causes trouble for other students in class, I may have to ask you to leave and return after taking care of what you need to*

What is the difference between ECE 4100 and ECE 6100?

33

- The Lecture material remains the same
- The Lab assignments (Lab2 - Lab4) for the undergraduate section (4100) will have reduced requirements
 - Extra credit for doing the ECE6100 version of the assignments
- The midterm will be the same for both ECE4100 and ECE6100

What is the difference between sections A and B?

- Sections A and B are both cross-listed as ECE4100/ECE6100/ CS4290/CS6290
- Both sections will cover the same material by the end
 - Individual lectures may be different
 - The way we cover the material will be different
 - The digressions based on student questions will be different
 - The labs will be different
 - The midterm/final exams will be different
- You are welcome to attend both sections if you like to reinforce the material
- However, you are responsible for taking the midterm and final for your own section → Section B in our case

Resources to do well in this class

35

Lectures



Recitations



Piazza



TA Office Hours



Instructor Office Hours

Most of you can get
an A in this class.

Next Few Lectures

36

- History of Computers!
 - Difference Engine → ENIAC → IBM 360 → Modern ISAs
- Implementing an ISA
 - Non-pipelined
 - Pipelined
 - Hazards
- Extracting Instruction Level Parallelism (ILP)
 - Branch Prediction
 - Out of Order Execution
 - Speculative Execution

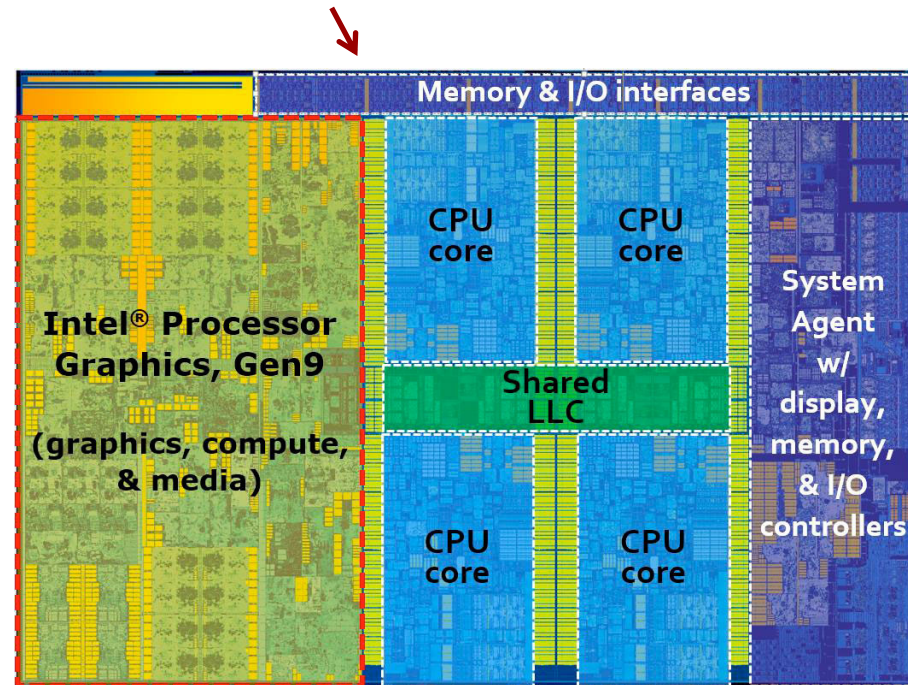
Takeaways from this class

37

The processor you built
in ECE3056/equivalent



What you'll understand and
can design after ECE6100



Die Photo of Intel Skylake, 2015

See you on Thursday!