Course Staff

- **Instructor**
  - Professor Tushar Krishna
  - Assistant Professor, School of ECE and CS
  - Background
    - PhD from MIT in EECS in 2013
    - Intel (MA) from 2014-15
  - Research Interests
    - Computer Architecture
    - **Networks-on-Chip (NoCs)**
    - Heterogeneous Architectures
- **Contact**
  - Email: tushar@ece.gatech.edu
  - Office: Klaus 2318
    - Office Hours
      - Tuesday and Thursday: 4:30 – 5 pm (after class)
      - By Appointment
  - Website: http://tusharkrishna.ece.gatech.edu
Course Staff

- **Teaching Assistants (TAs)**
  - **Kartikay Garg**
    - Email: kgarg40@gatech.edu
    - Office Hours to be announced via T-Square
  - **Anmol Gupta**
    - Email: anmol.gupta@gatech.edu
    - Office Hours to be announced via T-Square
Course Information and Updates

- Course Website for Schedule
  - [http://tusharkrishna.ece.gatech.edu/teaching/aca_f16/](http://tusharkrishna.ece.gatech.edu/teaching/aca_f16/)

- T-Square
  - 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?
What is Computer Architecture?

- **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
Defining Computer Architecture

Original domain of the computer architect (‘50s-'80s)

- Application
- Algorithm
- Programming Language
- Operating System/Virtual Machine
- Instruction Set Architecture (ISA)
- Microarchitecture
- Register-Transfer Level (RTL)
- Circuits
- Devices
- Physics

Parallel computing security, ...

Domain of computer architecture (‘90s)

Reliability, power

Expansion of computer architecture, mid-2000s onward.
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...
Computing devices now
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 journey through this space

- 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
  - Cell phones

- Focus on ideas, mechanisms and principles, especially those that have withstood the test of time
Syllabus

- **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
  - VLIW
  - Heterogeneous
Textbook and Readings

  - 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
Pre-Requisites

- **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
### Grading

<table>
<thead>
<tr>
<th>Item</th>
<th>Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lab1</td>
<td>5%</td>
</tr>
<tr>
<td>Lab2</td>
<td>10%</td>
</tr>
<tr>
<td>Lab3</td>
<td>10%</td>
</tr>
<tr>
<td>Lab4</td>
<td>10%</td>
</tr>
<tr>
<td>HW1</td>
<td>2%</td>
</tr>
<tr>
<td>HW2</td>
<td>2%</td>
</tr>
<tr>
<td>Participation</td>
<td>1%</td>
</tr>
<tr>
<td><strong>SubTotal</strong></td>
<td><strong>40%</strong></td>
</tr>
<tr>
<td>Midterm1</td>
<td>20%</td>
</tr>
<tr>
<td>Midterm2</td>
<td>20%</td>
</tr>
<tr>
<td><strong>Final Exam</strong></td>
<td><strong>20%</strong></td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td><strong>100%</strong></td>
</tr>
</tbody>
</table>

Overall grading will be relative. About 50% of you will get an A.
Two Midterms and One Final

**Midterms**
- 1hr 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
### Four Labs

- **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 T-square 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!
Lectures

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

- TAs will hold 1-2 recitation sessions every week
  - Cover material complementary to lectures
    - walk-through examples of difficult concepts
    - review items that lots of students are discussing via Piazza/email
    - model questions to help prepare for midterms and final
      - Make sure you attend recitation before midterms!
  - Topic of recitation will be emailed every week
  - Days, times and venues to be announced soon
    - You can attend one or all sessions every week

- 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

- Getting the most out of recitations
  - Prepare a list of questions
  - Communicate with recitation TA what you would like to see covered
  - Participate and ask questions
  - Follow up with TA/instructor through Piazza/email if more questions
Office Hours

- 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)

<table>
<thead>
<tr>
<th>Week</th>
<th>Dates</th>
<th>Tuesday</th>
<th>Thursday</th>
<th>Due Dates [Fri]</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>(Aug 22 - )</td>
<td></td>
<td></td>
<td>Lab #1 Due</td>
</tr>
<tr>
<td>2</td>
<td>(Aug 29 - )</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>(Sep 5 - )</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>(Sep 12 - )</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>(Sep 19 - )</td>
<td></td>
<td></td>
<td>HW#1 Due, Lab #2 Due</td>
</tr>
<tr>
<td>6</td>
<td>(Sep 26 - )</td>
<td></td>
<td></td>
<td>Midterm 1</td>
</tr>
<tr>
<td>7</td>
<td>(Oct 3 - )</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>(Oct 10 - )</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>(Oct 17 - )</td>
<td></td>
<td></td>
<td>Fall Recess</td>
</tr>
<tr>
<td>10</td>
<td>(Oct 24 - )</td>
<td></td>
<td></td>
<td>Lab #3 Due</td>
</tr>
<tr>
<td>11</td>
<td>(Oct 31 - )</td>
<td></td>
<td></td>
<td>HW#2 Due</td>
</tr>
<tr>
<td>12</td>
<td>(Nov 7 - )</td>
<td></td>
<td></td>
<td>Midterm 2</td>
</tr>
<tr>
<td>13</td>
<td>(Nov 14 - )</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>(Nov 21 - )</td>
<td></td>
<td></td>
<td>Thanksgiving</td>
</tr>
<tr>
<td>15</td>
<td>(Nov 28 - )</td>
<td></td>
<td></td>
<td>Lab#4 Due</td>
</tr>
<tr>
<td>16</td>
<td>(Dec 5 - )</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>(Dec 12 - )</td>
<td></td>
<td></td>
<td>Final</td>
</tr>
</tbody>
</table>
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 ECElike course in CS, the most CSlike course in ECE
Zero tolerance for cheating

- **ALL lab assignments are individual**
  - You can discuss ideas with other students
  - You CANNOT see (or show) other students code
    - We will use MOSS to detect cases of substantial overlap

- 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)

22 students caught in Fall 2012, 22 in Spring 2014, 2 in Fall 2014
What is expected from you?

- 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”

- 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
    

- 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?

- 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

Lectures

Recitations

Piazza

TA Office Hours

Instructor Office Hours

Most of you can get an A in this class.
Next Few Lectures

- 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

The processor you built in ECE3056/equivalent

What you’ll understand and can design after ECE6100

See you on Thursday!