Welcome to CS2630!

- Syllabus:
  - [http://homepage.cs.uiowa.edu/~bdmyers/cs2630_su17//syllabus/](http://homepage.cs.uiowa.edu/~bdmyers/cs2630_su17//syllabus/)
  - or linked from ICON > Syllabus

- 36 students but 72 seats!
- Please sit at the 7,6,4,5 tables
CS 2630
Computer Organization
Meeting 1: Introduction
Brandon Myers
University of Iowa
One of my research projects

Make programming this

...feel like programming this
What’s with the classroom?

• “TILE (Transform, Interact, Learn, Engage) is a nationally-recognized University of Iowa initiative that encourages team-based, inquiry-guided pedagogies.”
  • from “TILE at The University of Iowa”

• Less about the technology and more about a classroom design that supports student-centered learning
Think (write), pair, share

1. What do you think you will learn in CS2630?

2. What is one thing you are excited about in this course?

3. What is one thing you are not excited about in this course?

a) write answers to the three questions;
b) turn to 1 person next to you, introduce yourself;
c) share your answers
Why take 2630?

• Brandon’s esoteric answer: graduates of a Computer Science program should have an appreciation for how real computers work
• ACM and IEEE’s answer: Computer Organization and Architecture is a topic in Computer Science Curricula 2013 (https://www.acm.org/education/CS2013-final-report.pdf)
• But more concretely...
Why take 2630?

1. It will be up to you to **design our new computer systems (software AND hardware)**...computer architects have been panicking for nearly a decade and they are not calming down
Computer Architecture topics post

A chance for you to bring information to the course, perhaps including articles about new computer architectures!

(syllabus and ICON for more details)
Word cloud generated by Spring semester posts
2. at some point you will have to **measure a system you’ve built**: performance (latency & throughput), energy usage, reliability, ... To be able to measure/interpret/improve your system, it helps to understand how more of the computer works.

What metrics would you measure to know how good a car is?

What can the engineer do to change the values of those metrics?
The future of CS2630

• CS2630 moving to TILE classrooms in Fall-Summer
• Replacing some lectures with lab assignments
  • allow us to better support learning all the tools, get more time analyzing and designing programs and circuits
  • very speculatively: future opportunity for lab assistants (help students but do not grade work)
The **computing stack**

you’ll focus most on the shaded layers

each layer presents a convenient abstraction to the layer above
Recurring theme: The reality of our abstractions
Microprocessor Transistor Counts 1971-2011 & Moore's Law

curve shows transistor count doubling every two years
In pairs, generate 3 observations or questions about this plot; then as a table, decide on 3 to share with the class.
Why isn’t multicore working?

To create this plot, the experimenter:
1. found 4 different programs with different amounts of the program that could potentially be run in parallel (50%, 75%, 90%, 95%)
2. ran each program on 1 processor (aka core); let’s call the running time $T_1$
3. ran the same programs on $X$ processors, plotting $Y=T_1/T_x$ (speedup)

Here is some context to help:

Your task: in group of 3, come up with a description of what the plot is telling us!
TL;DR
Multicore will increase performance up to a point, but because of power issues as well as Amdahl’s Law, we can’t bet on the million-core chip. We need entirely new designs for the computers of the near future.

2011

Dark Silicon and the End of Multicore Scaling

Hadi Esmaeilzadeh$^1$  Emily Blum$^3$  Renée St. Amant$^4$  Karthikeyan Sankaralingam$^5$  Doug Burger$^6$

$^1$University of Washington  $^2$University of Wisconsin-Madison  $^3$The University of Texas at Austin  $^4$Microsoft Research

hadinezh@cs.washington.edu  blem@cs.wisc.edu  otsmamt@cs.utexas.edu  karu@cs.wisc.edu  dburger@microsoft.com

ABSTRACT
Since 2005, processor designers have increased core counts to exploit Moore’s Law scaling, rather than focusing on single-core performance. The failure of Dennard scaling, to which the shift to multicore parts is partially a response, may soon limit multicore scaling just as single-core scaling has been curtailed. This paper models multicore scaling limits by combining device scaling, single-core scaling, and multicore scaling to measure the speedup potential for a set of parallel workloads for the next five technology generations. For device scaling, we use both the ITRS projections and a set of parameters, and compiler advances, Moore’s Law, coupled with Dennard scaling [11], has resulted in commensurate exponential performance increases. The recent shift to multicore designs has aimed to increase the number of cores along with transistor count increases, and continue the proportional scaling of performance. As a result, architecture researchers have started focusing on 100-core and 1000-core chips and related research topics and called for changes to the undergraduate curriculum to solve the parallel programming challenge for multicore designs at these scales.

With the failure of Dennard scaling—and thus slowed supply volt-
Let’s discuss the syllabus

• ICON > 2630 Summer 2017 > Syllabus
• Or,
  http://homepage.cs.uiowa.edu/~bdmyers/cs2630_su17//syllabus/

• In triples, find 1 unique question to ask about the syllabus
  • try your best to find a question that is not directly answered by the syllabus
  • write your question on the board nearest to you
Course calendar

<table>
<thead>
<tr>
<th>Week</th>
<th>Date</th>
<th>In class</th>
<th>Reading</th>
<th>Notes</th>
<th>Homework / Projects</th>
<th>Quiz</th>
</tr>
</thead>
<tbody>
<tr>
<td>6/12</td>
<td></td>
<td>Introduction</td>
<td>Introduction</td>
<td></td>
<td>HW1: Bits, bytes, memory organization</td>
<td>Data, integers,</td>
</tr>
</tbody>
</table>
Approach of the course

• Review the content knowledge (resources on the website)
  • textbook or other readings
  • notes
  • videos and other resources
  • and finally, some live lecture (augments and doesn’t replace the above)

• Learn higher order skills: application, analysis, synthesis, evaluation
  • In-class activities and lab assignments
  • ICON quizzes (immediate feedback)
  • Homeworoks and projects
Be successful in CS2630

• Prepare for class; review the materials before and do pre-labs
• Attend class to do the activities and labs with your peers
• Help your peers on ICON and in class
• Attend Debug Your Brain and/or office hours
• Midterm in class (TBD)
• Keep on top of announcements in ICON
What to do now

• Review this week’s readings
• Take Quiz 1 until you get 100% (due Friday)
• Start on HW1 (due Friday)
• Prepare for Lab 1 (in-class Wednesday)
• Take the poll for agreeing on late policy