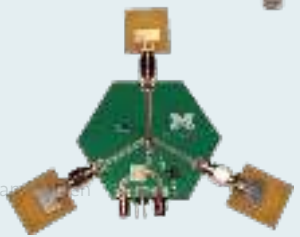
CSE 141: Introduction to Computer Architecture

Pat Pannuto, UC San Diego
ppannuto@ucsd.edu



What is Computer Architecture and where does it fit in Computer (Science) Engineering?

- One view: what is an Architect and how do they fit in the creation of buildings?

Zaha Hadid

Port Authority Building in Antwerp, designed by Zaha

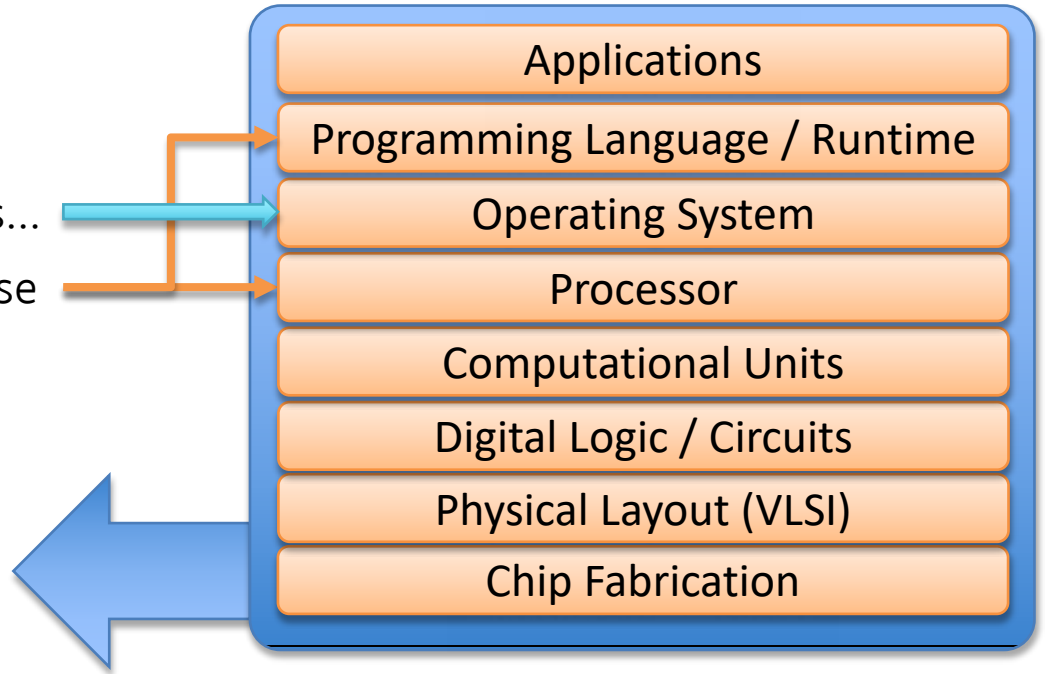
Graphics courtesy Wikipedia

Computer science is all about abstractions

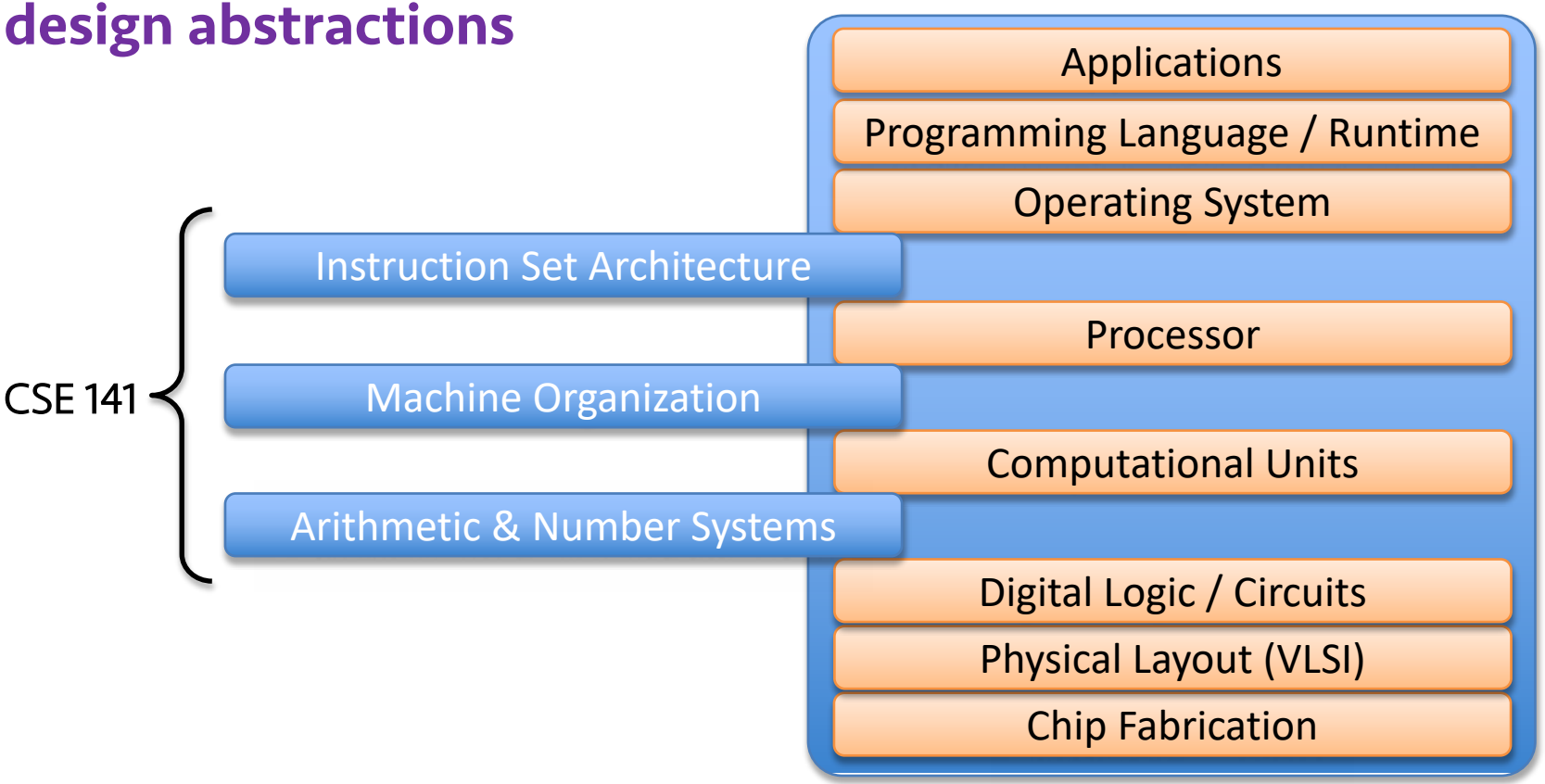
The person who designs this...

...only needs to worry about these

But who designs the big picture?



Computer architects look at the system as a whole and design abstractions

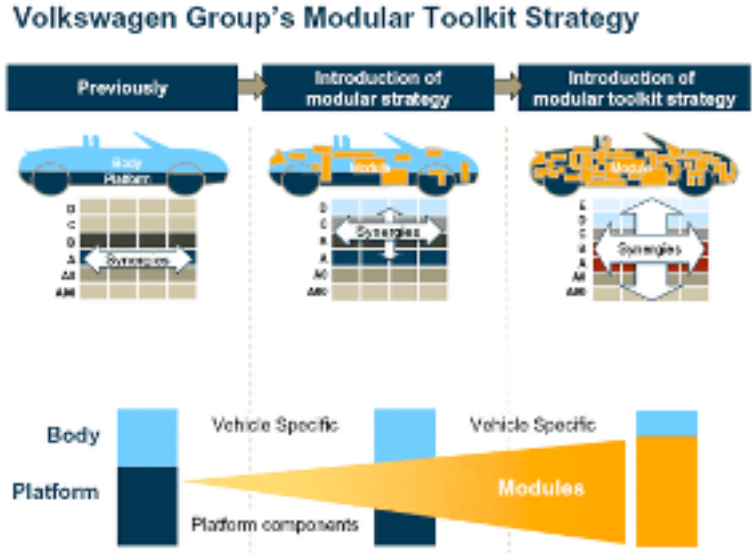


Good abstractions make it easier to focus on reasoning about one part of a large, complex system

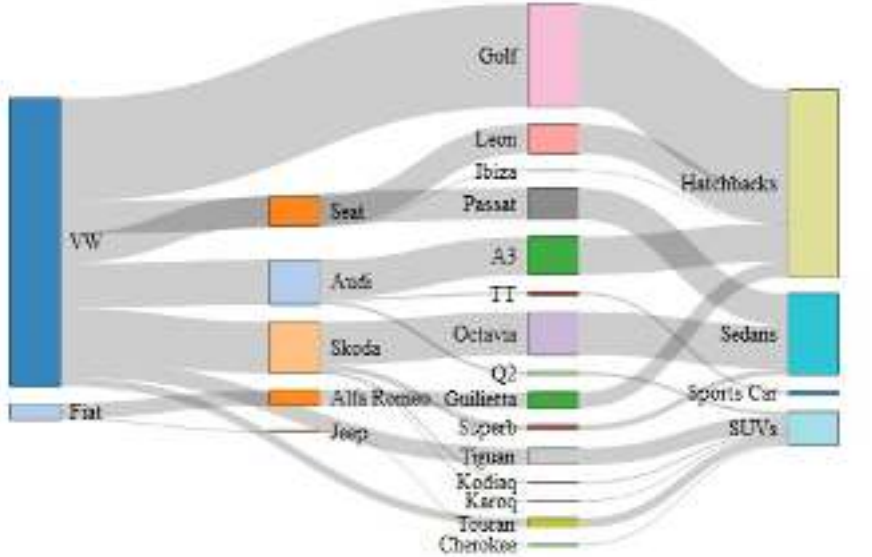
- Which of these maps is easier to use to plan a trolley trip?

Good abstractions make it easier to focus on reasoning about one part of a large, complex system

- Modularization is fundamental to design in many domains



Modular Car Body Design and Optimization by an Implicit Parameterization Technique via SFE CONCEPT
Fabien Duddeck, Hans Zimmer

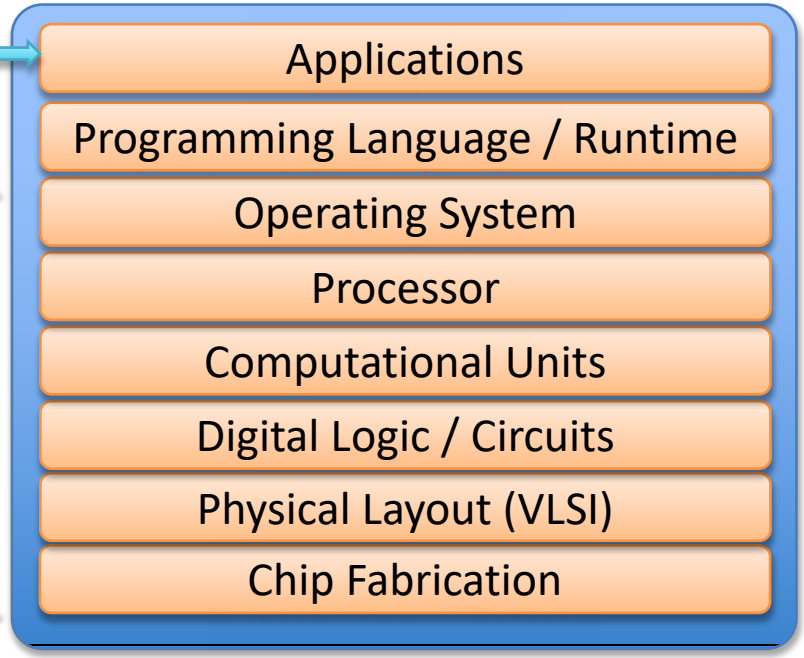


https://www.reddit.com/r/dataisbeautiful/comments/8m15g9/automobile_platform_sharing_work_in_progress/

But what if I'm not going to become a computer architect?

If I only want to build these…

…why do I need to know about any of this?



The real world is full of leaky abstractions

- **Goal:** Sum up all the entries of a two dimensional array
- Which of these implementations is faster?

```
int twoDarray[256][256];
int sum = 0;

for (int i=0; i<256; i++) {
  for (int j=0; j<256; j++) {
    sum += twoDarray[i][j];
  }
}
```

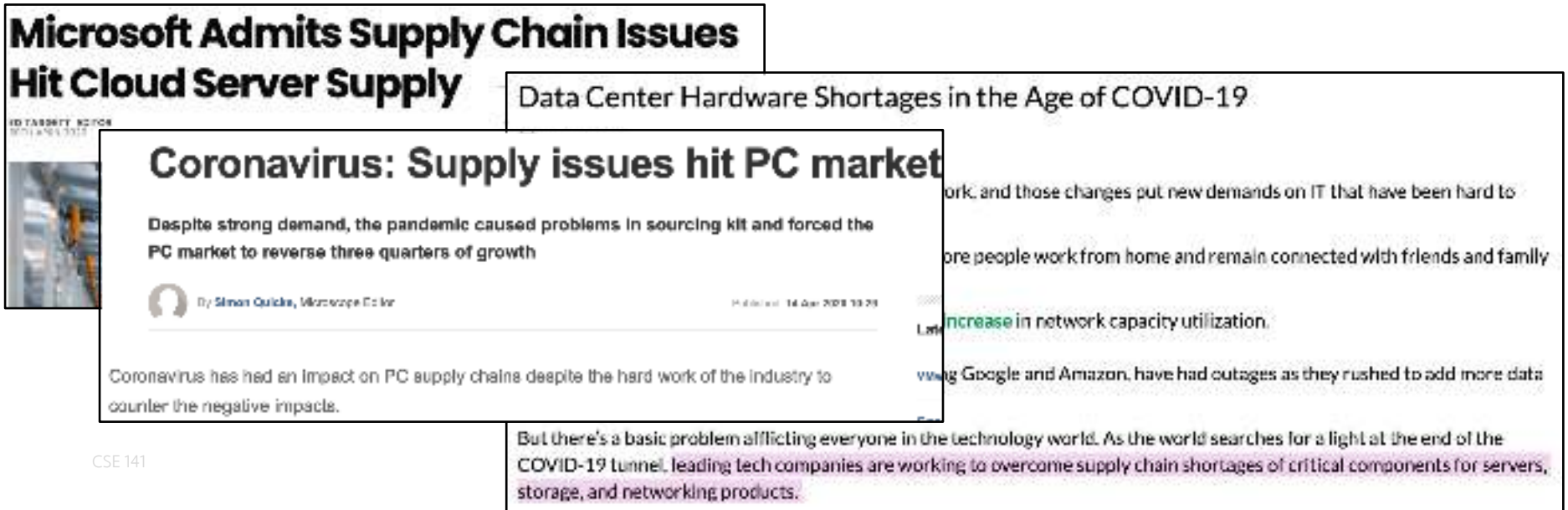
```
int twoDarray[256][256];
int sum = 0;

for (int i=0; i<256; i++) {
  for (int j=0; j<256; j++) {
    sum += twoDarray[j][i];
  }
}
```

Answer: "It depends"

Architects look across systems to find and improve inefficiencies – always valuable, sometimes critical…

- What happens when workload changes overnight, and there's no way to buy your way out of the problem?
  - Architects help to fix it



Course Administrivia

- Instructors
  - Sec A00: Pat Pannuto
  - Sec B00: Dean Tullsen
- The sections will run "in sync"
  - The lectures will cover the same content but in different ways
  - Assignments will be similar enough that folks should be able to work in groups across the sections, but may not be identical
  - Unified piazza, office hours
  - We will give exams at the same time
    - Note: The registrar assigned our final exam as Saturday, December 12 – Plan now!!

Course Staff

- Four amazing TAs:
  - Nitish Kulshrestha
  - Shanti Modi
  - Sumiran Shubhi
  - Kazem Taram
- Discussions
  - Sec A00: Wed from 11:00 to 11:50
  - Sec B00: Tue from 14:00 to 14:50
  - You may attend either, but A00 better when possible

Assessments & Workload

- Grading
  - Participation: 5%
    - Weekly **participation** quizzes of lecture material (10 weeks -> 0.5% / quiz!)
  - Homework: 20%
  - Midterm: 30%
  - Final Exam: 45%
    - (Inclusive final)
    - Our assigned final exam slot is SATURDAY, December 12 — Plan for this now!

Repeated, active engagement is key to effective learning

- Pre-class reading is your first exposure
  - 5 minutes before class is better than not at all, but 5+ hours before is much better
  - Read actively, try writing notes for yourself of what you understood from readings
- Watching video is not a passive activity
  - Ask (or write down) questions about what you do not understand!
  - Use checkpoints effectively
- Discussions, office hours, and exercises are not passive activities
  - Work through examples yourself and ask the questions you have
- Homework is designed to help you solidify your understanding
- Study for tests "honestly to yourself" – you must engage with questions

Class is not a competition

- My philosophy
  - I care whether you learn the material
  - The purpose of a grade is to assess how well you know the material in 141
  - The purpose of a grade is not "rank" students
  - I am most successful if everyone in class earns an A
- My goal is not to curve
  - (But I reserve the right to)
  - The midterm and final may be "internally" curved

Academic Integrity

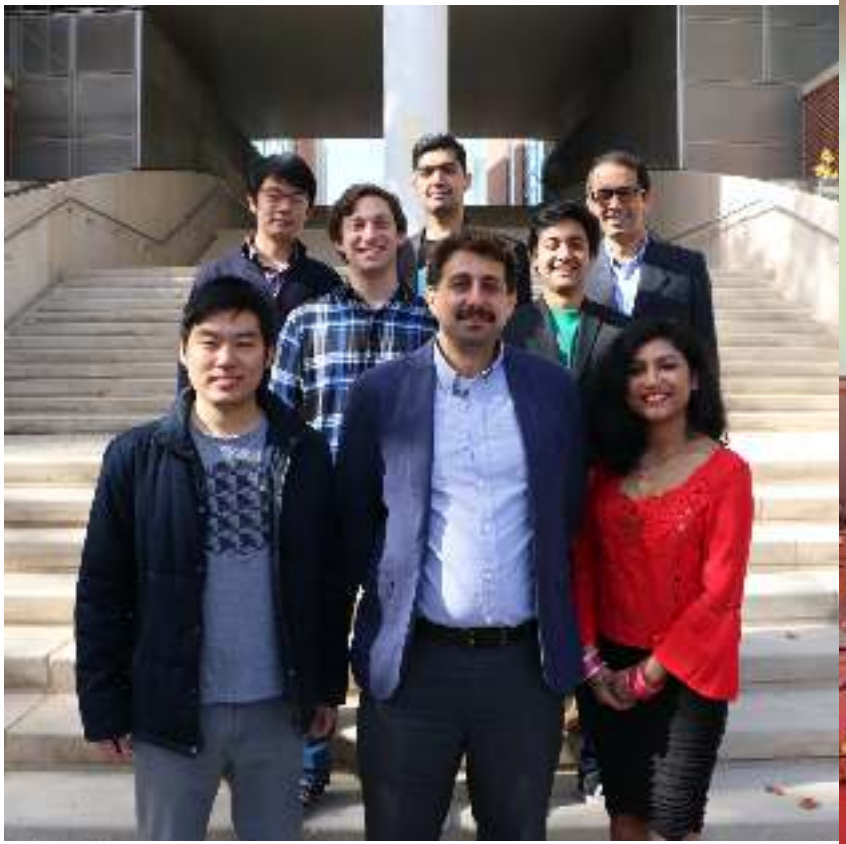
- Cheating will be taken very seriously
- Examples
  - Not cheating:
    - Discussing homework in groups, with **your own writeup, in your own words, done on your own, later**
  - Cheating:
    - Getting a walk-through from someone who has already done the homework
    - Looking at someone else's completed work (even "just to check")
    - Using solutions from the web, prior classes, or anywhere else
    - Receiving, providing, or soliciting assistance from another student during a test
- An experiment: Regret Policy

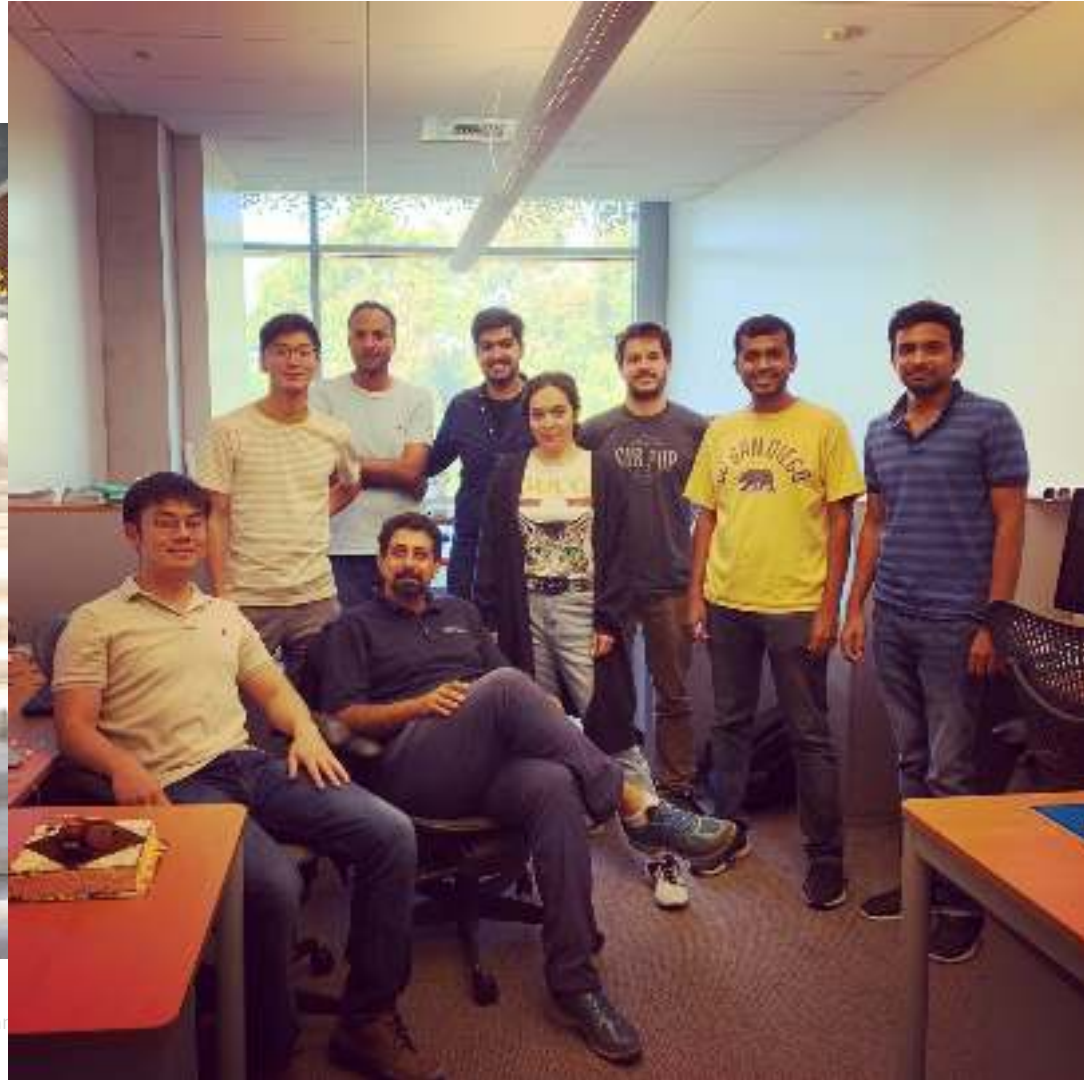
We'll take a short break here...

AND THEN SOME MODERN HIGHLIGHTS FROM HERE AT UCSD

But for the rest of today, I want to highlight the kinds of cool stuff that architects *do*

- UCSD has an amazing team of architecture faculty

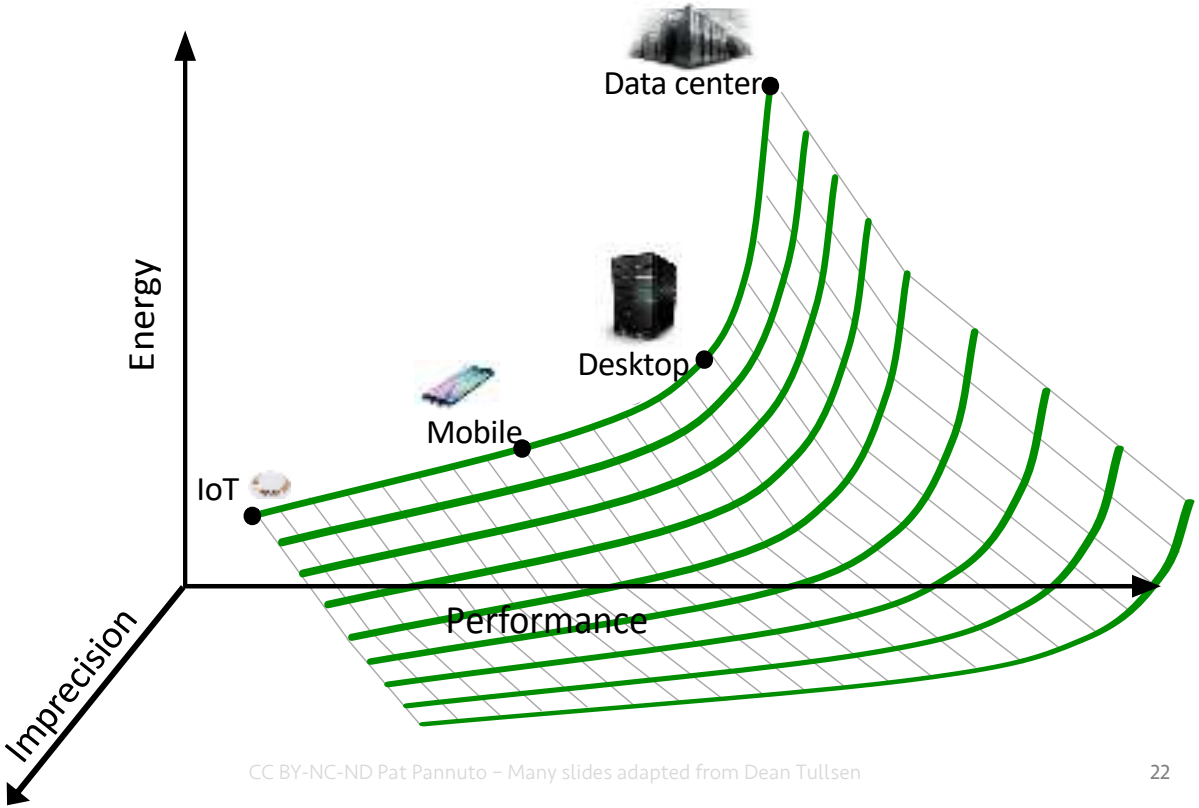




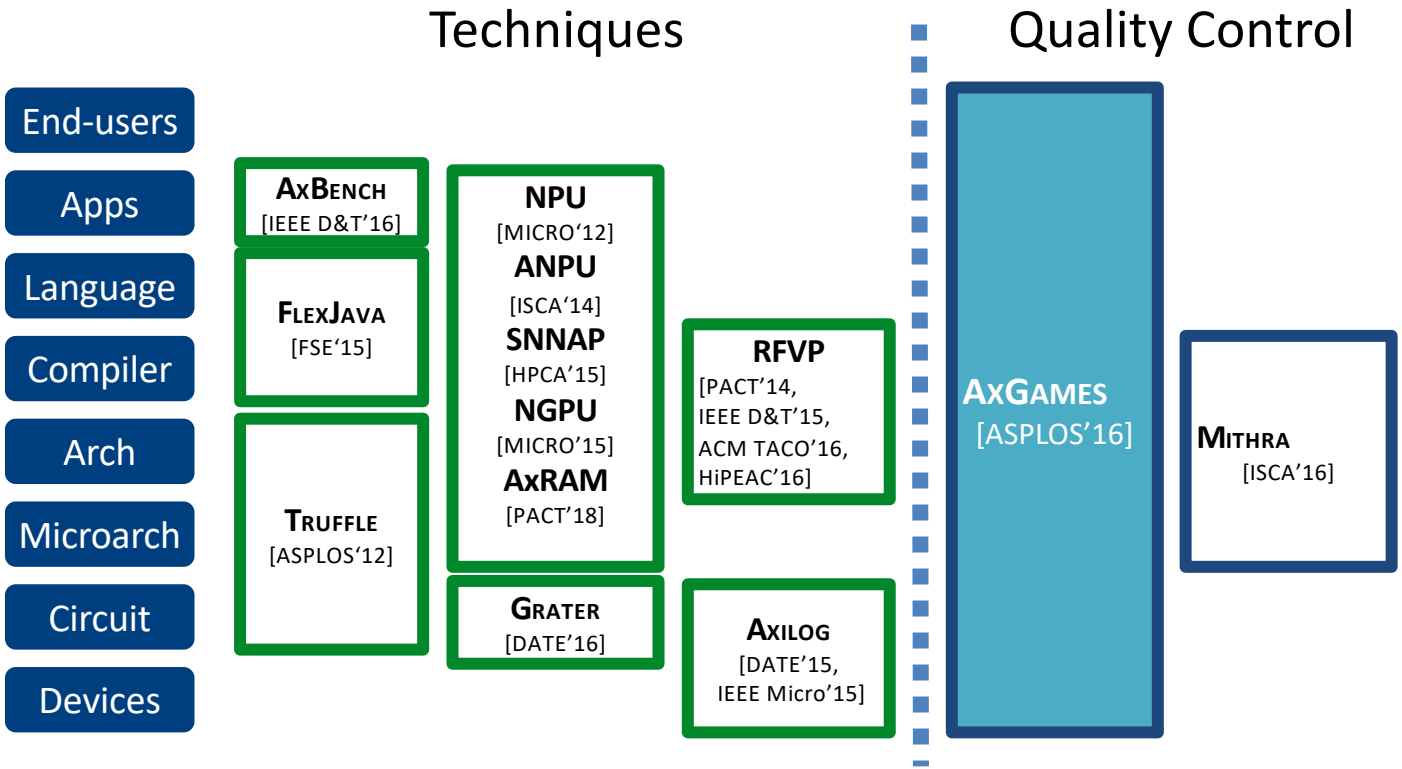
One wild idea: “Approximate Computing”

- Aka, what if 1 + 1 doesn’t *always* equal *exactly* 2?

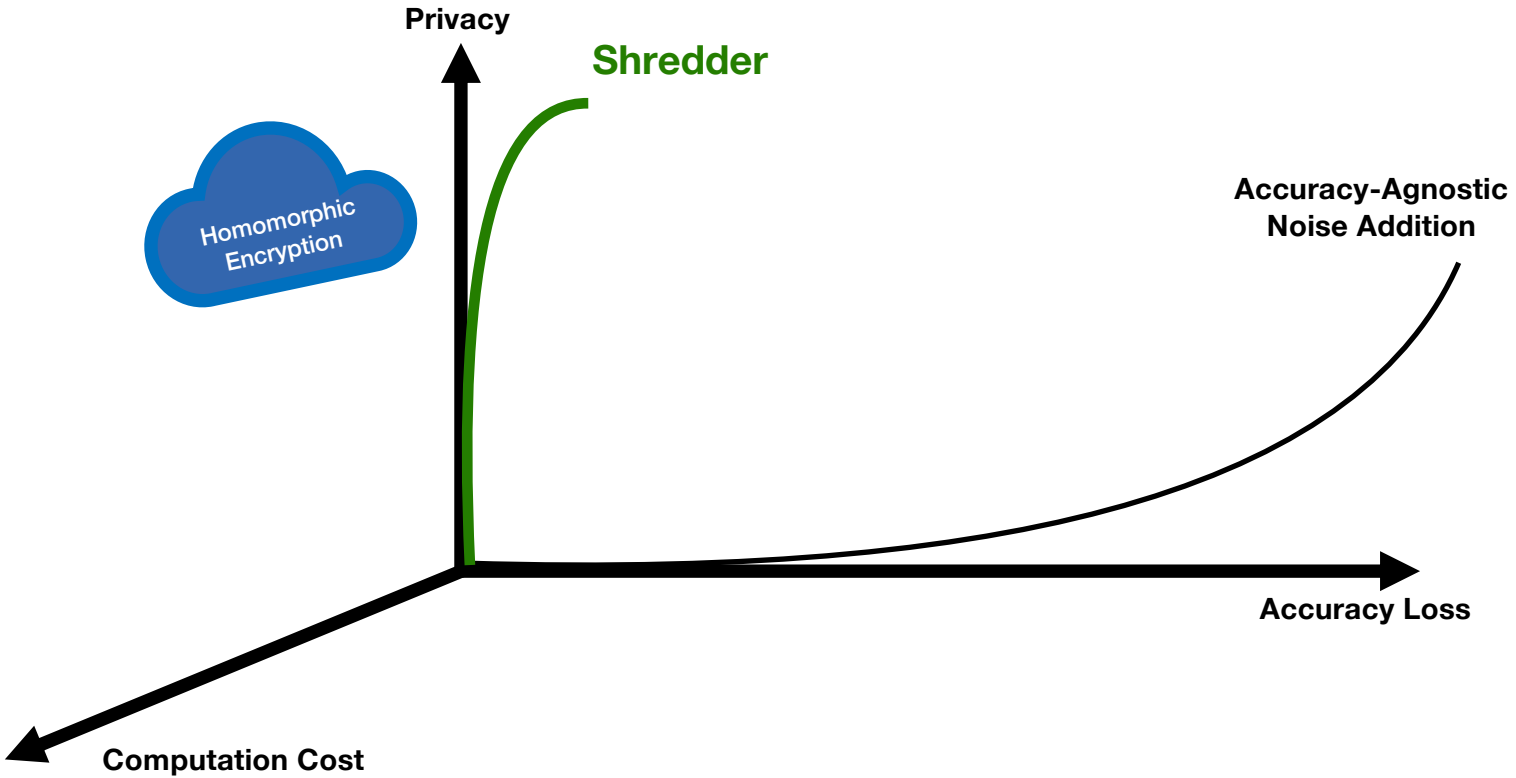
Embracing imprecision allows for major gains in performance and energy



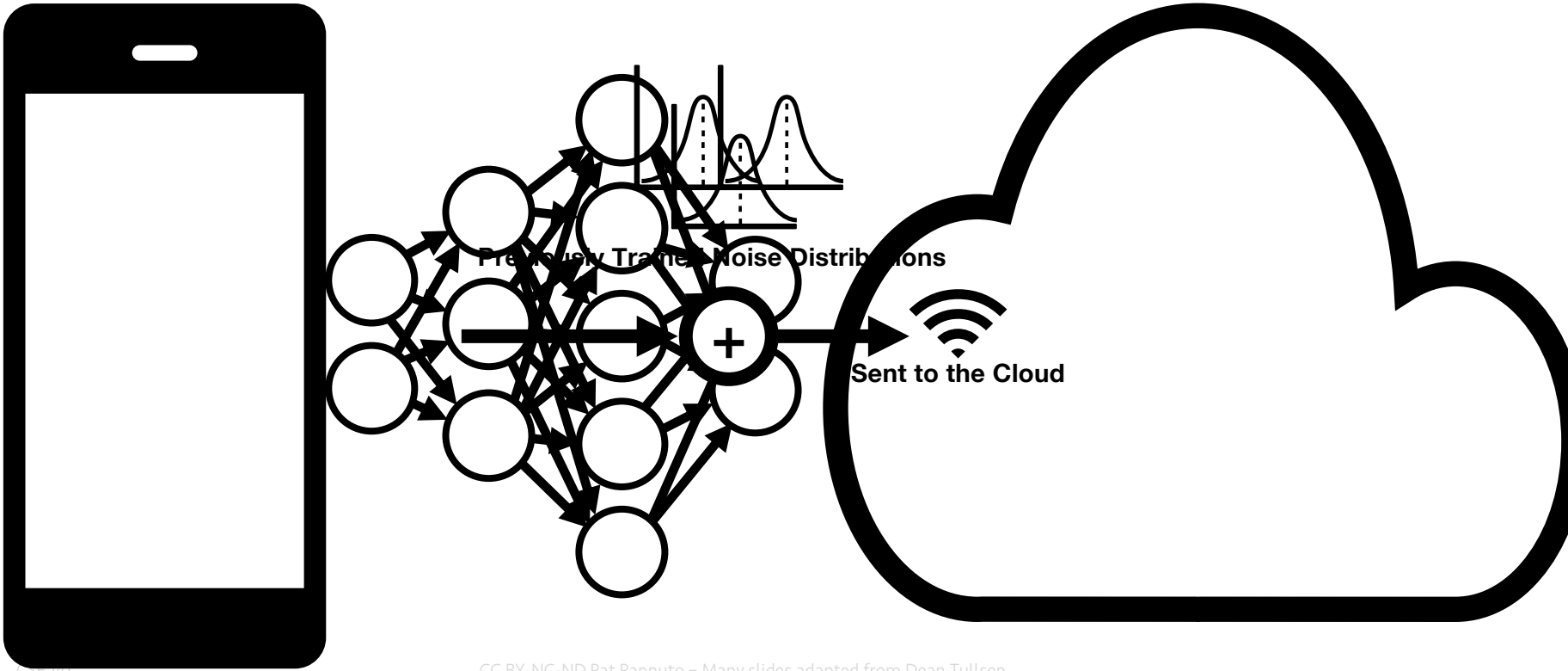
A cross-stack approach to enable approximation



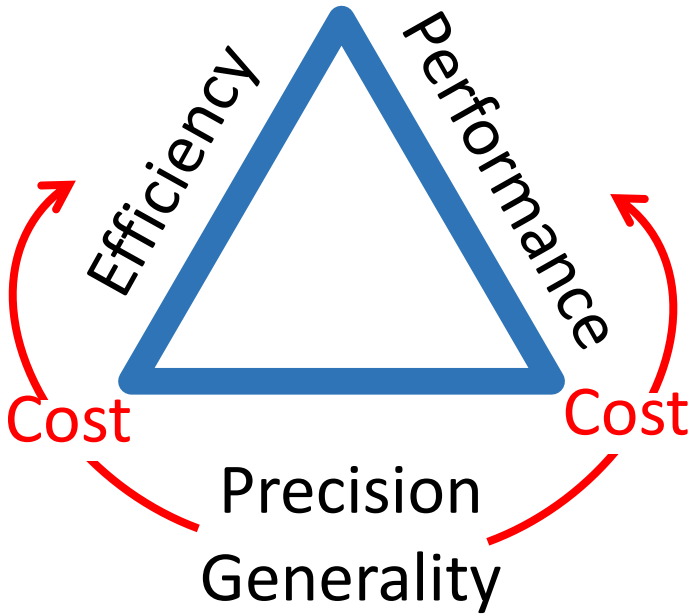
Privacy Preserving Techniques for Inference

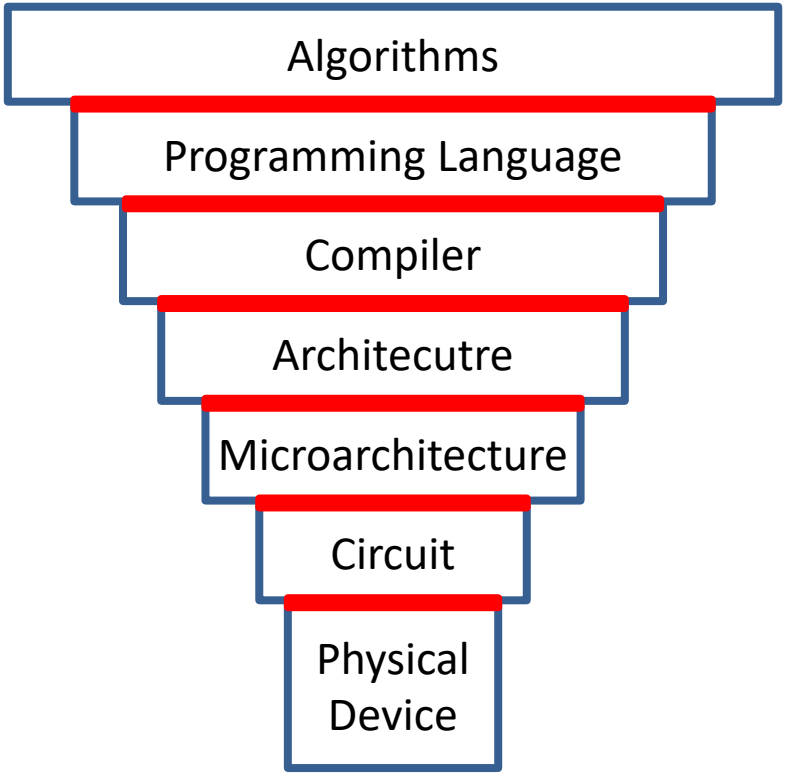


Execution Model



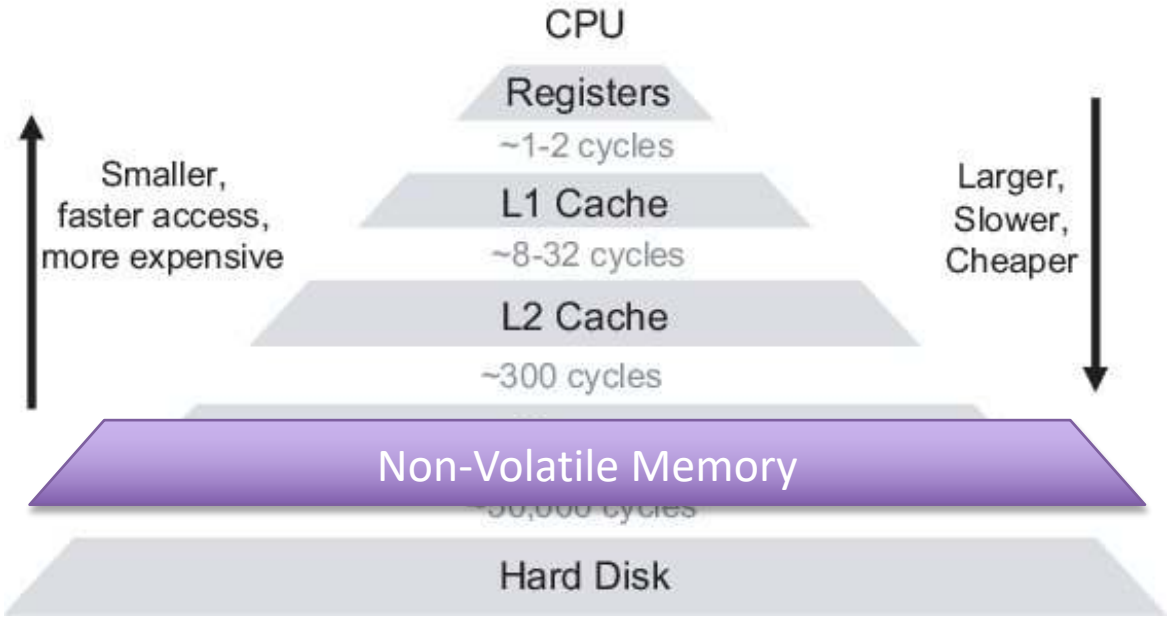
Rethinking the abstractions



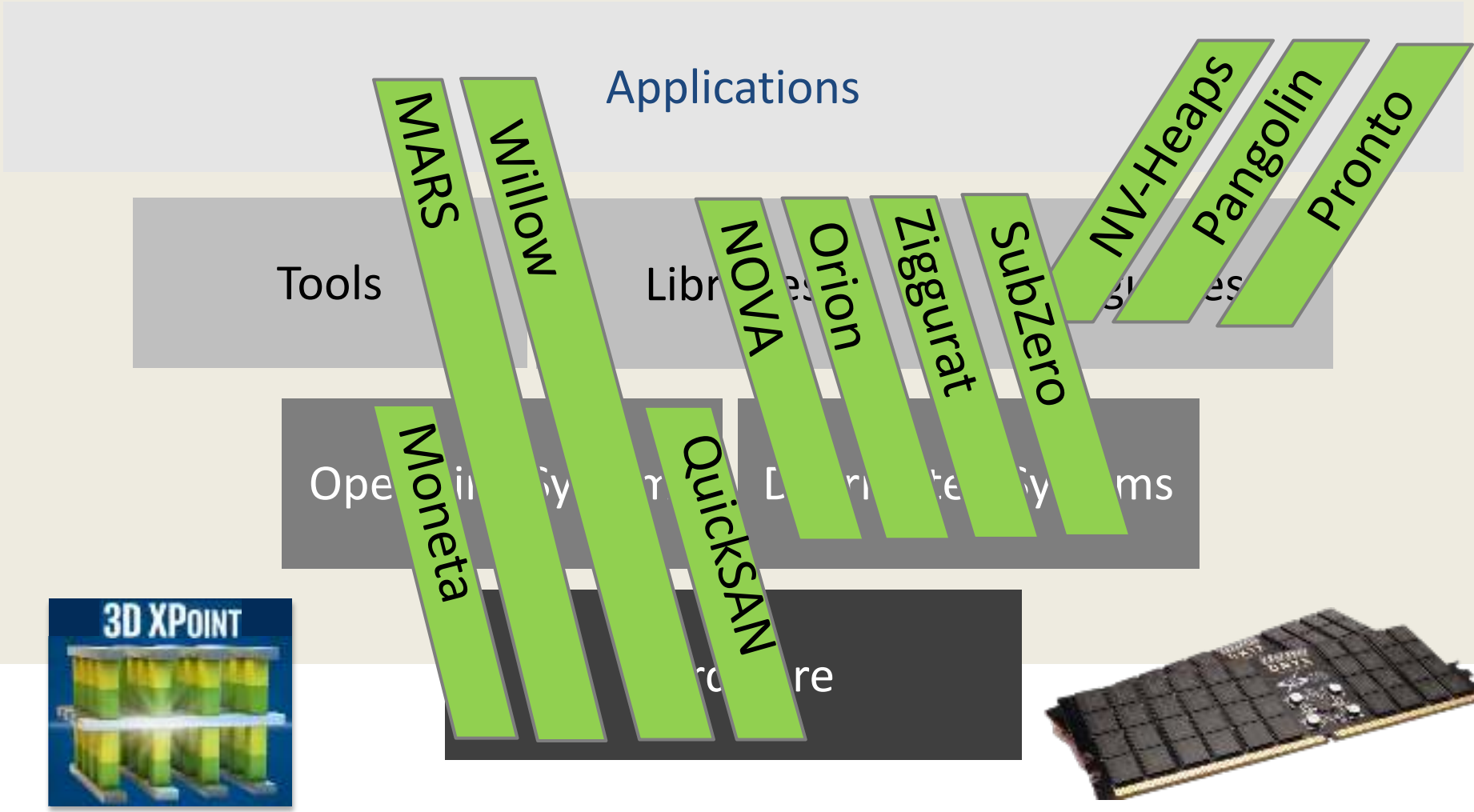


Memory, Storage, Software, and Architecture in the NVSL

This is a slide you will encounter in many CE/CSE classes…

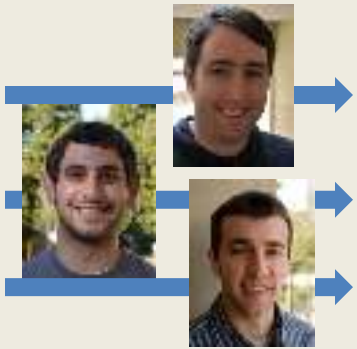


Chellappa, Srinivas & Franchetti, Franz & Püschel, Markus. (2007). How to Write Fast Numerical Code: A Small Introduction.


Applications
Tools
Libr
Ope
D
ms
re
MARS
Willow
Moneta
QuickSAN
NOVA
Orion
Ziggurat
SubZero
NV-Heaps
Pangolin
Pronto
3D XPOINT

NVSL Students Lead Industry

- We Built
  - Opt. SSD interface (2009)
  - Direct, remote SSD (2013)
  - First PCM SSD (2011)
  - PMEM prog. tools (2011)



- Industry Built
  - NVMe (2011)
  - NVMe over Fabrics (2016)
  - Optane (2016)
  - PMDK (~2014)

Mobilizing the Micro-Ops: Exploiting Context Sensitive Decoding for Security and Energy Efficiency

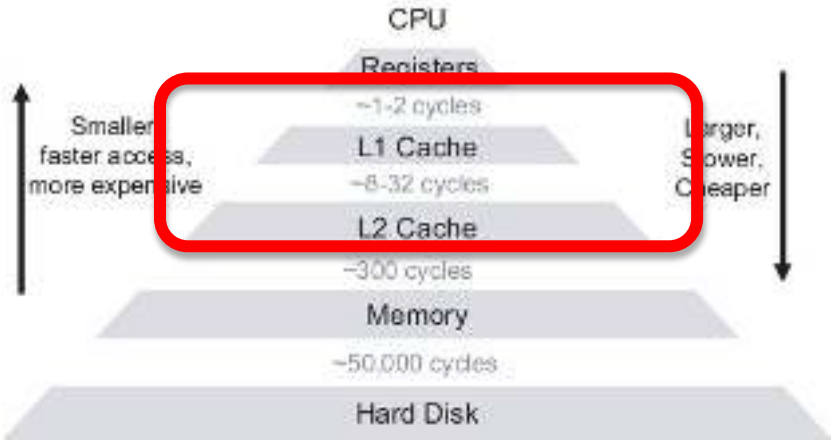
Leaky abstractions are not always just performance problems…

- This loop behaved differently because of how caches work

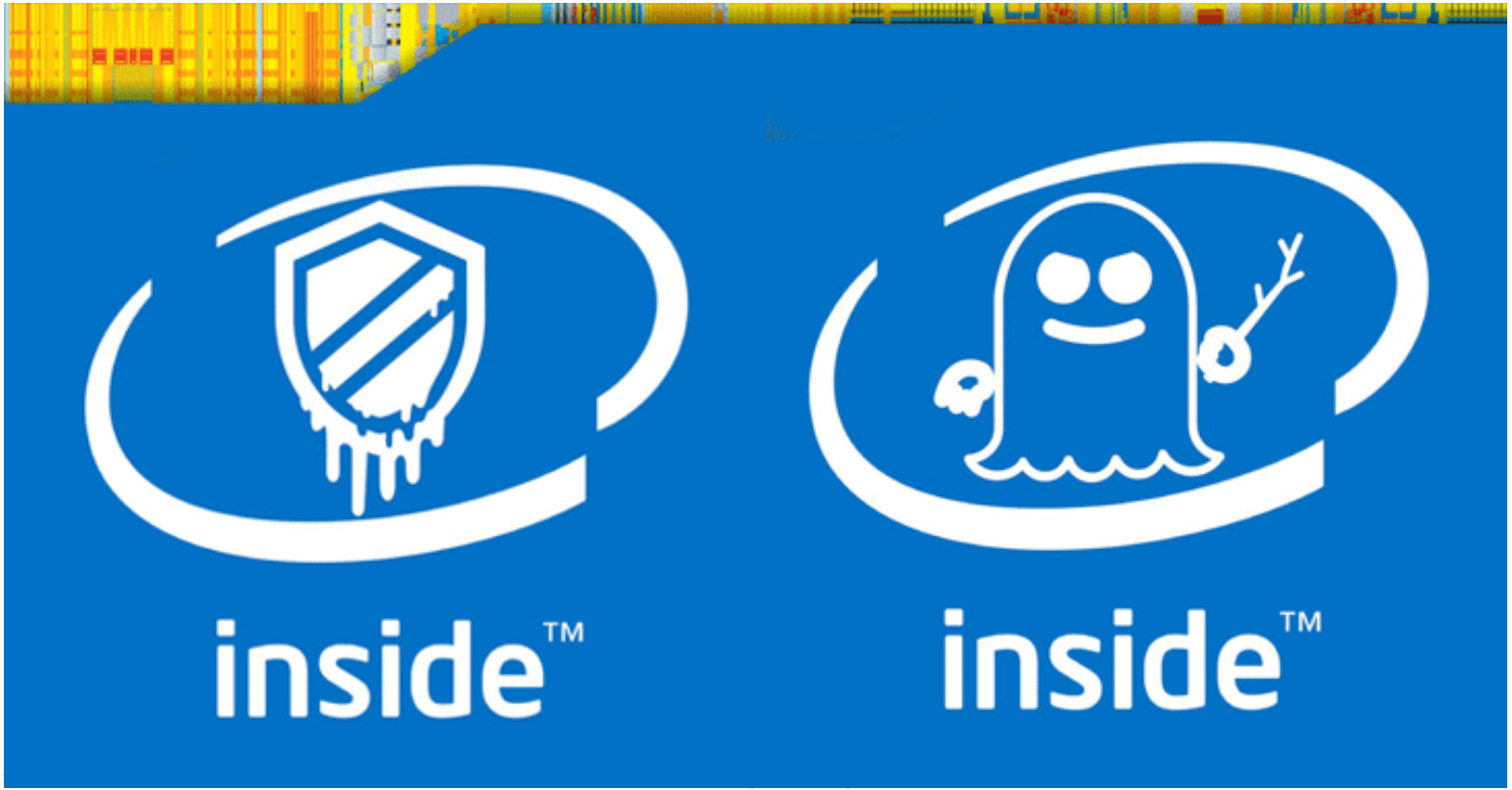
```
int twoDarray[256][256];
int sum = 0;

for (int i=0; i<256; i++) {
  for (int j=0; j<256; j++) {
    sum += twoDarray[i][j];
  }
}
```

Architects added "hidden" caches: faster, intermediate memories

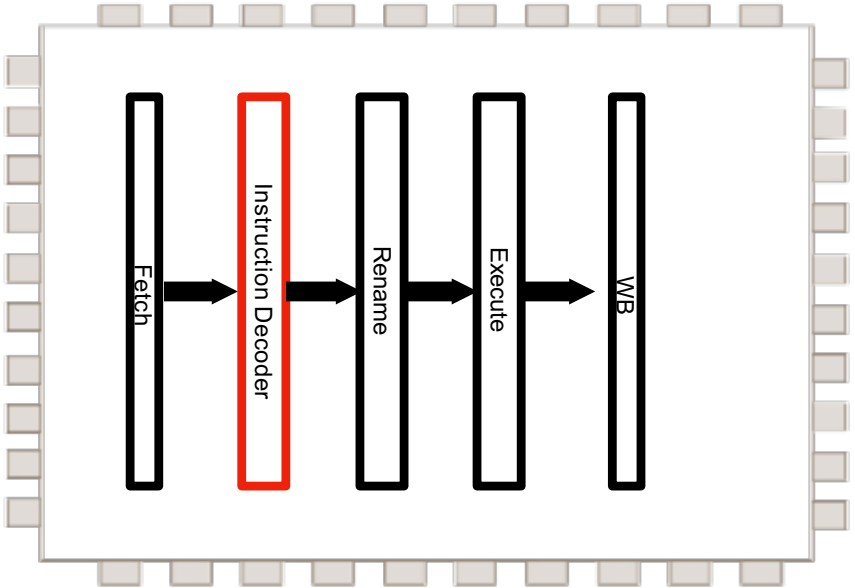


Leaky abstractions can be security threats!



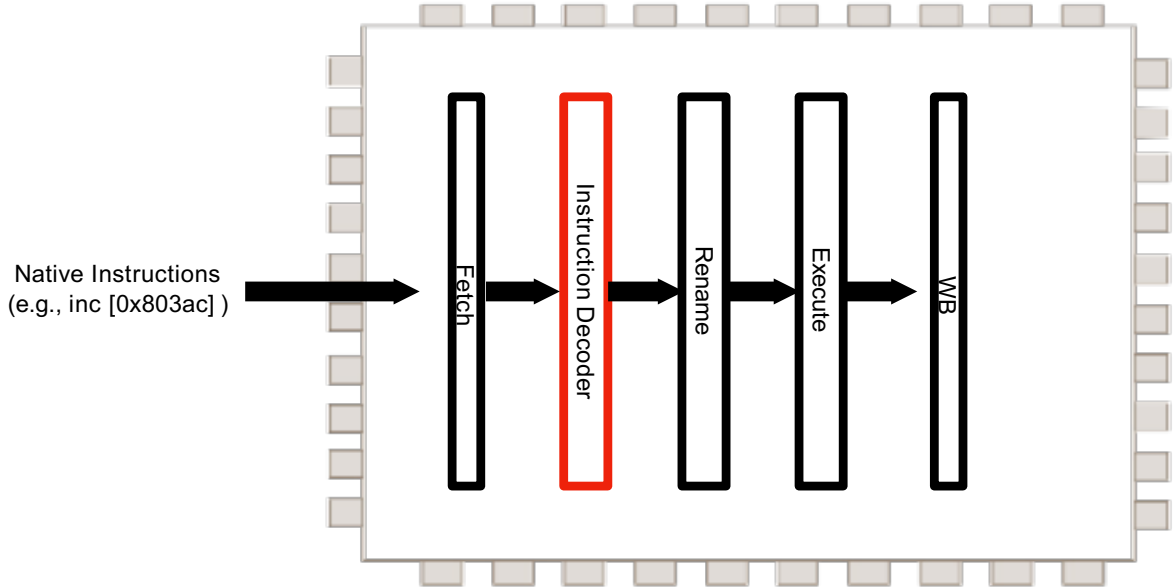
Mobilizing the Micro-Ops

Exploiting Translated ISAs



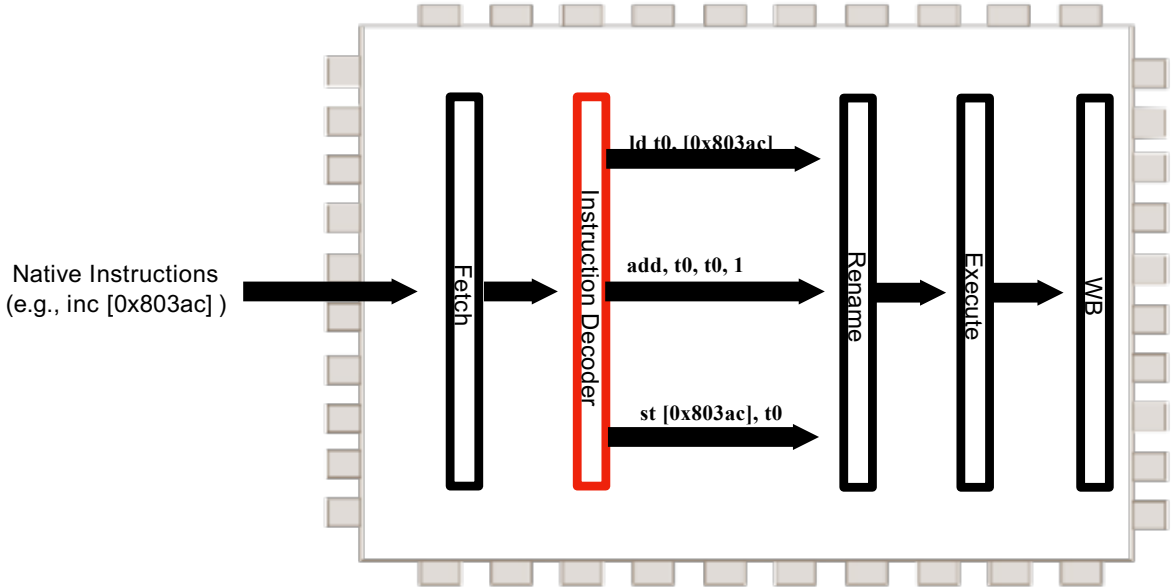
Mobilizing the Micro-Ops

Exploiting Translated ISAs



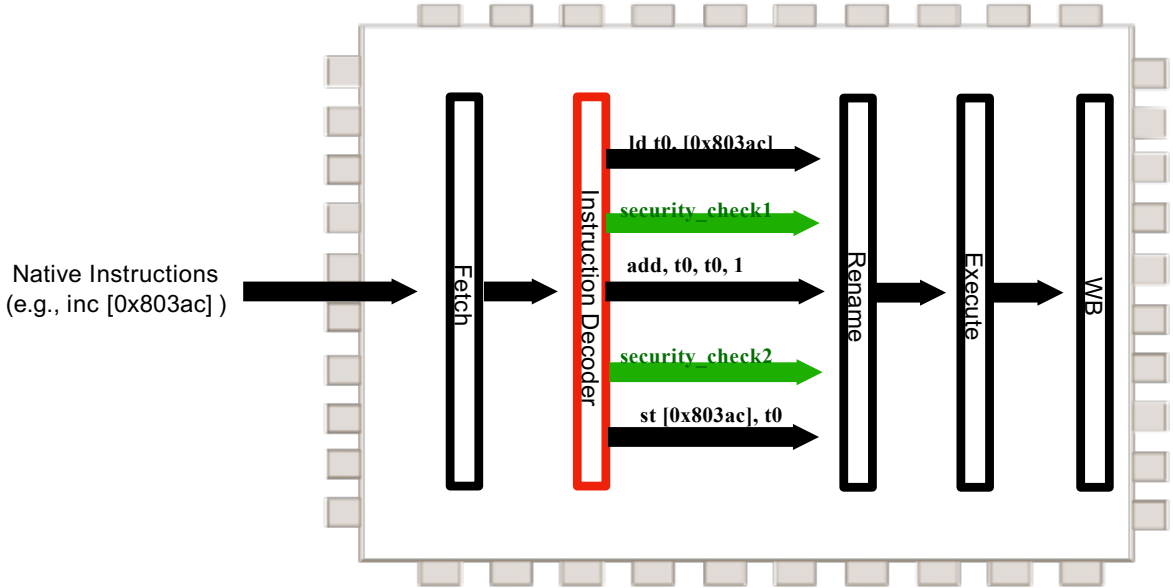
Mobilizing the Micro-Ops

Exploiting Translated ISAs



Mobilizing the Micro-Ops

Exploiting Translated ISAs



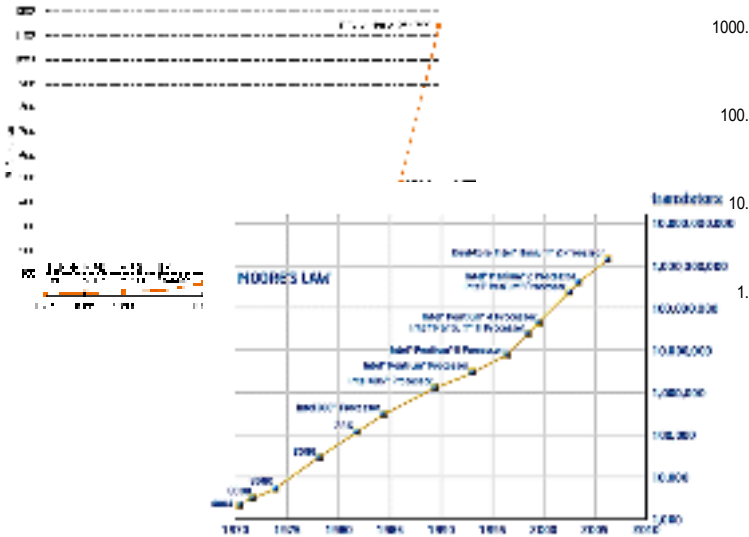
Context Sensitive Decoding fixes a leaky abstraction

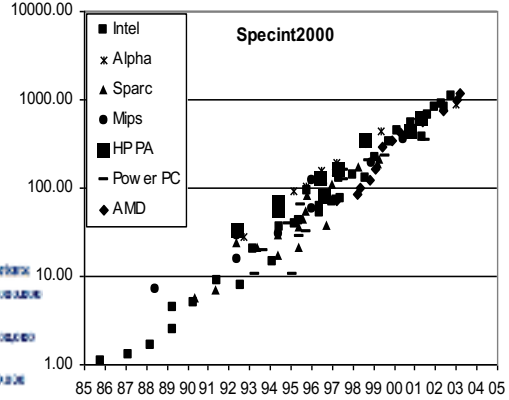
- Eliminating cache side channels via cache obfuscation
- Energy and Performance optimization via selective devectorization
  - ISCA 2018
  - IEEE Micro Top Picks in Computer Architecture

- Spectre mitigation via targeted insertion of fence micro-ops (Context Sensitive Fencing)
  - ASPLOS 2019

Performance was king, until we unplugged computers

- A lot of "classic" architecture research is makes sure graphs continue to go up and to the right

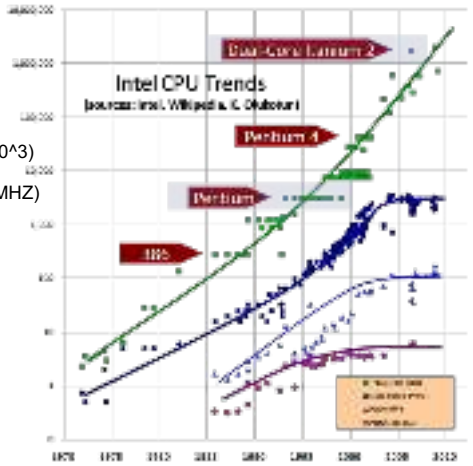




Processor Design Trends

- Transistors (*10^3)
- Clock Speed (MHZ)
- Power (W)
- ILP (IPC)

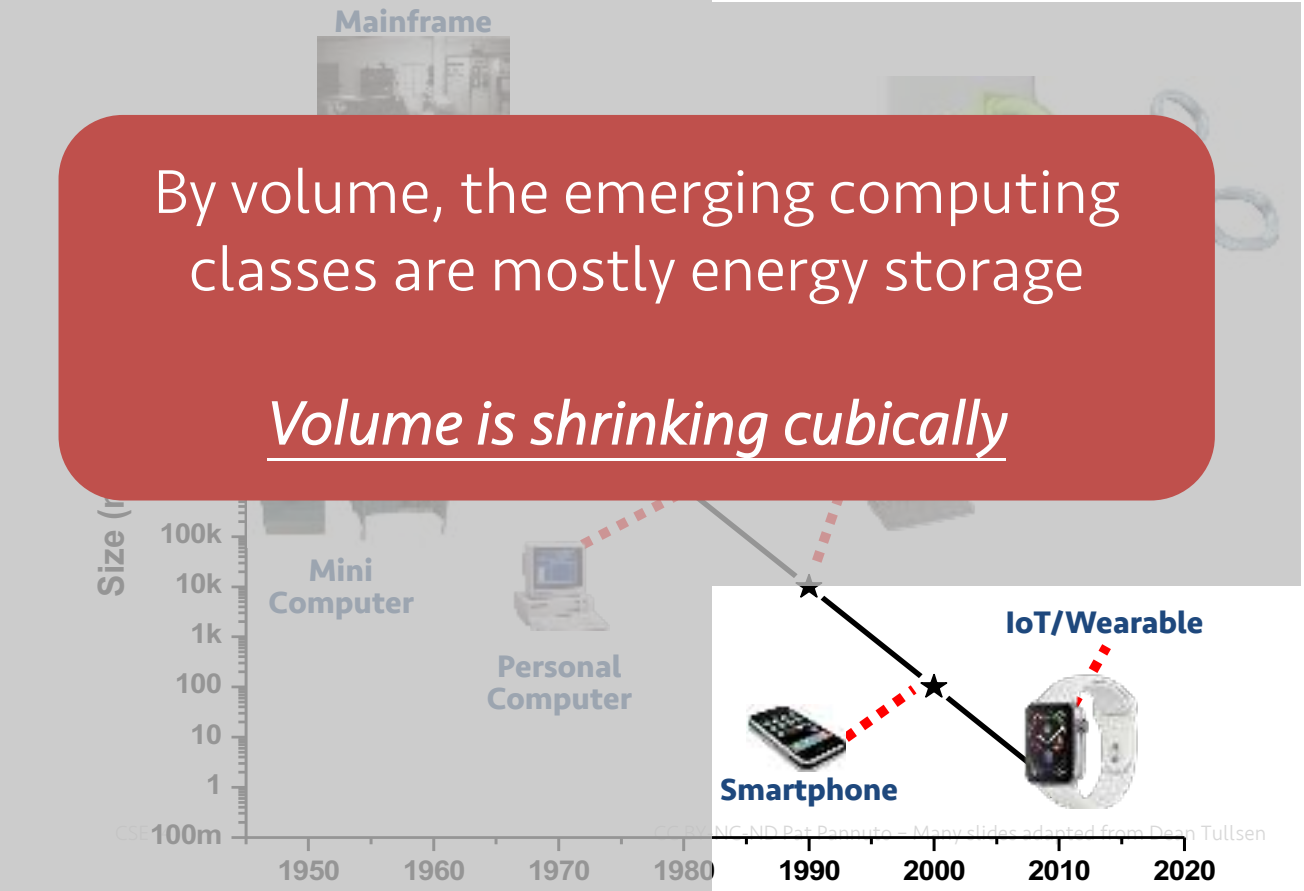
*From Herb Sutter, Dr. Dobbs Journal

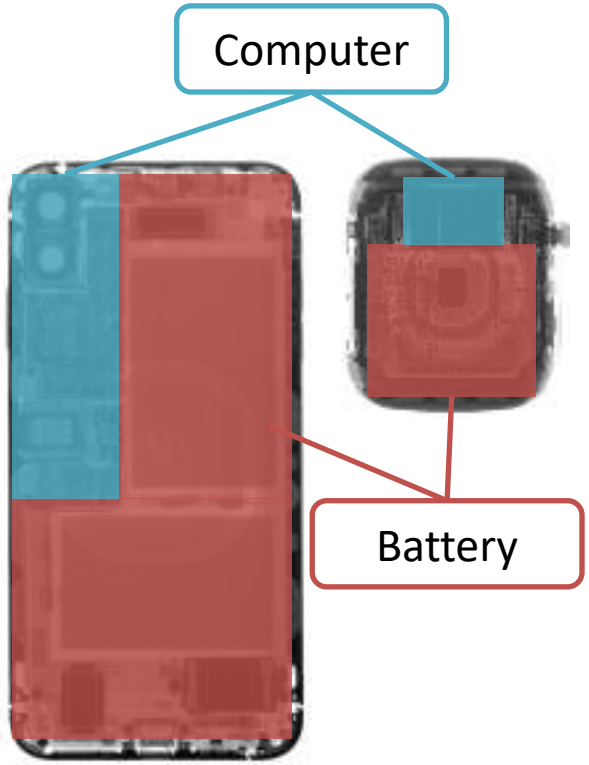


I spend my time on graphs that go down and to the right

By volume, the emerging computing classes are mostly energy storage

Volume is shrinking cubically



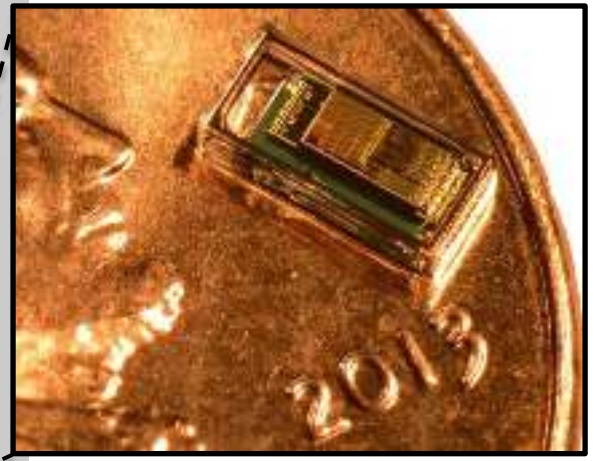


Computational platforms will continue to scale

The next generation of computing will only be a cubic millimeter in size

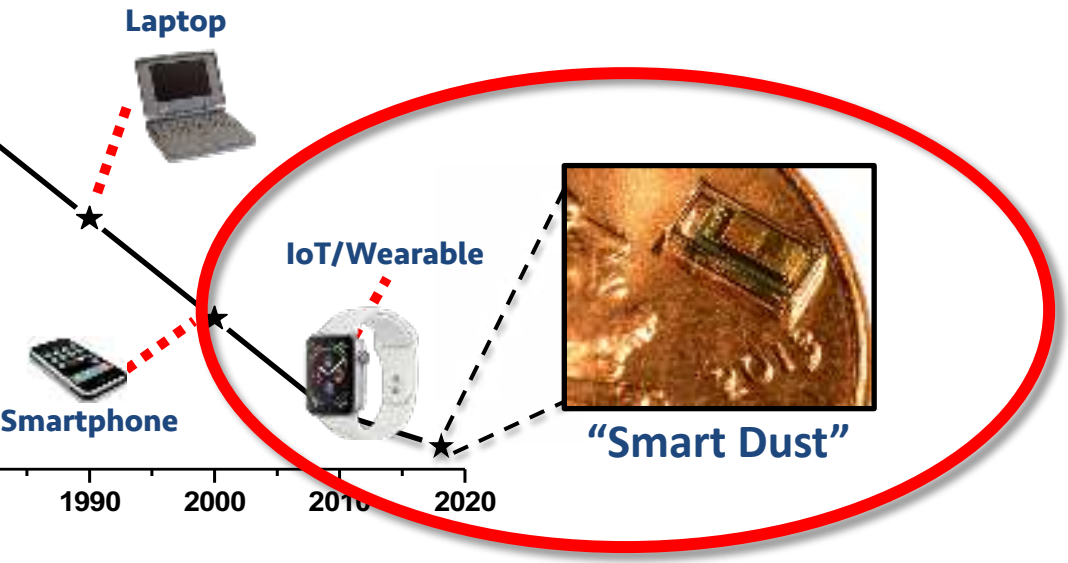
Millimeter-scale batteries have capacities around **5 µAh**

(would power an idle iPhone for 0.6 s)



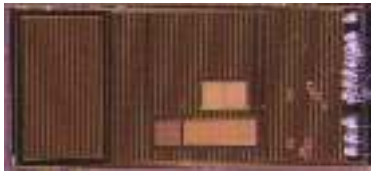
"Smart Dust"

Energy constraints will play a central role in the evolution of computing platforms



How must traditional paradigms change, adapt, or re-invent for the new computing classes?

One of the first challenges was re-thinking how we put together computers



Temperature Sensor
~10 pW standby, < 1 µW active

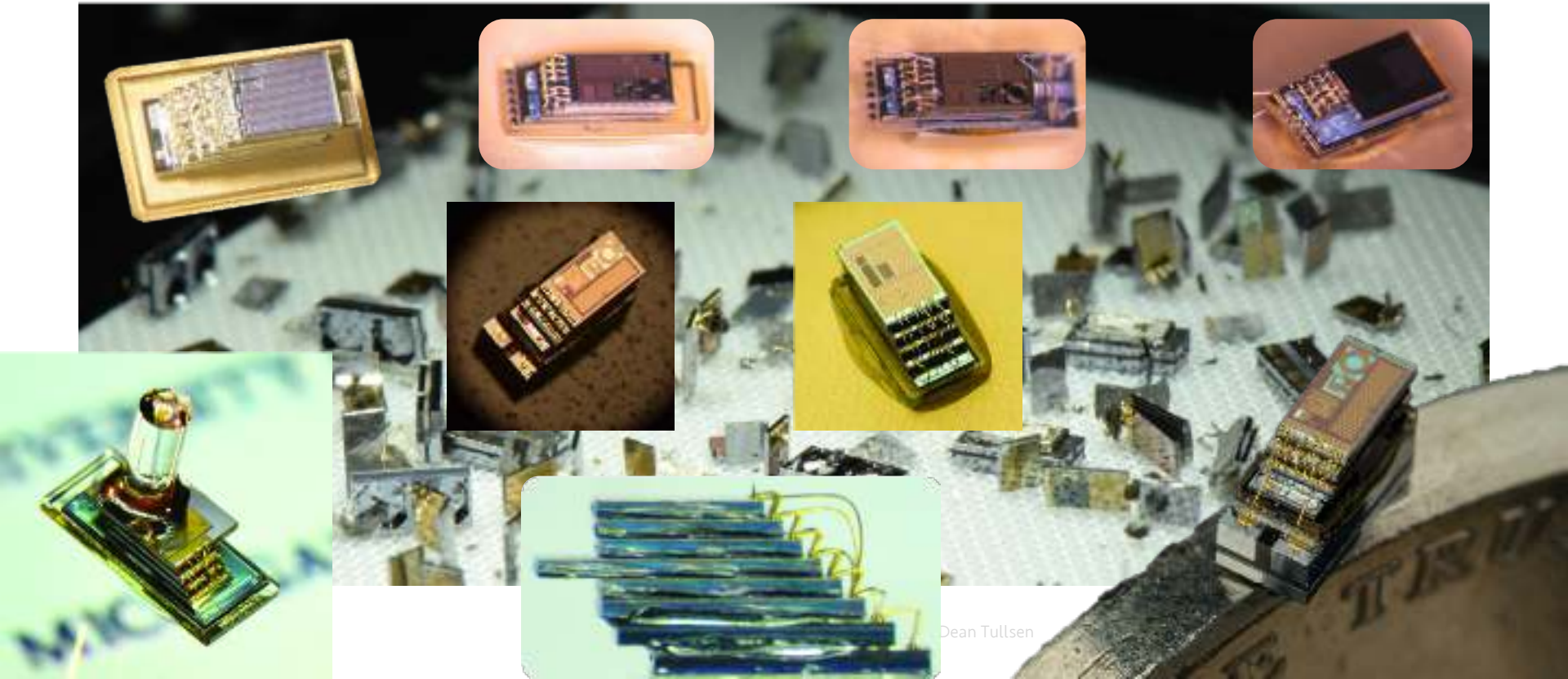
CPU
~1 nW standby, ~5 µW active

Radio
~10 pW standby, ~10 µW active

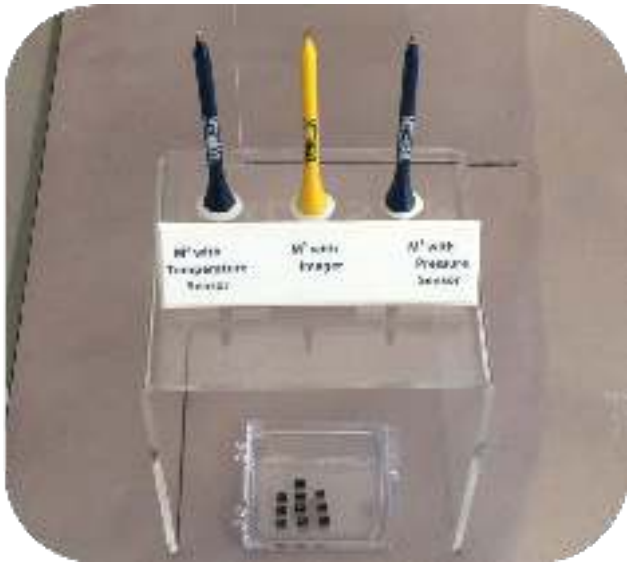
Energy Harvesting & Storage
1~10 nW indoors
2~10 µAh capacity

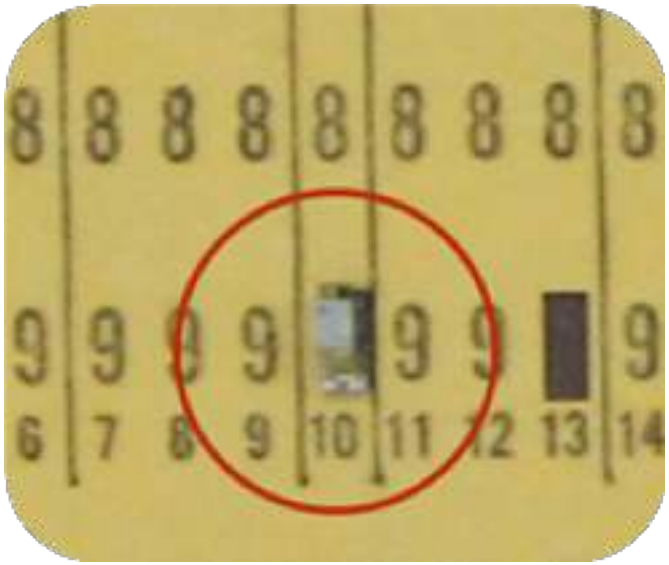
?

MBus enabled the of development of dozens of millimeter-scale motes as part of the Michigan Micro Mote (M3) project



Check out the “World’s Smallest Computer” exhibit at Silicon Valley’s Computer History Museum!





Next week: Instruction Set Architectures (ISAs)

- For Monday:
  - Skim 1.1 [7 pages]
  - Read 1.2, 1.3 [6.5 pages]