ENGRI 1210
Recent Trends in Computer Engineering

Christopher Batten

School of Electrical and Computer Engineering
Cornell University

The Complexity of Modern Computer Systems

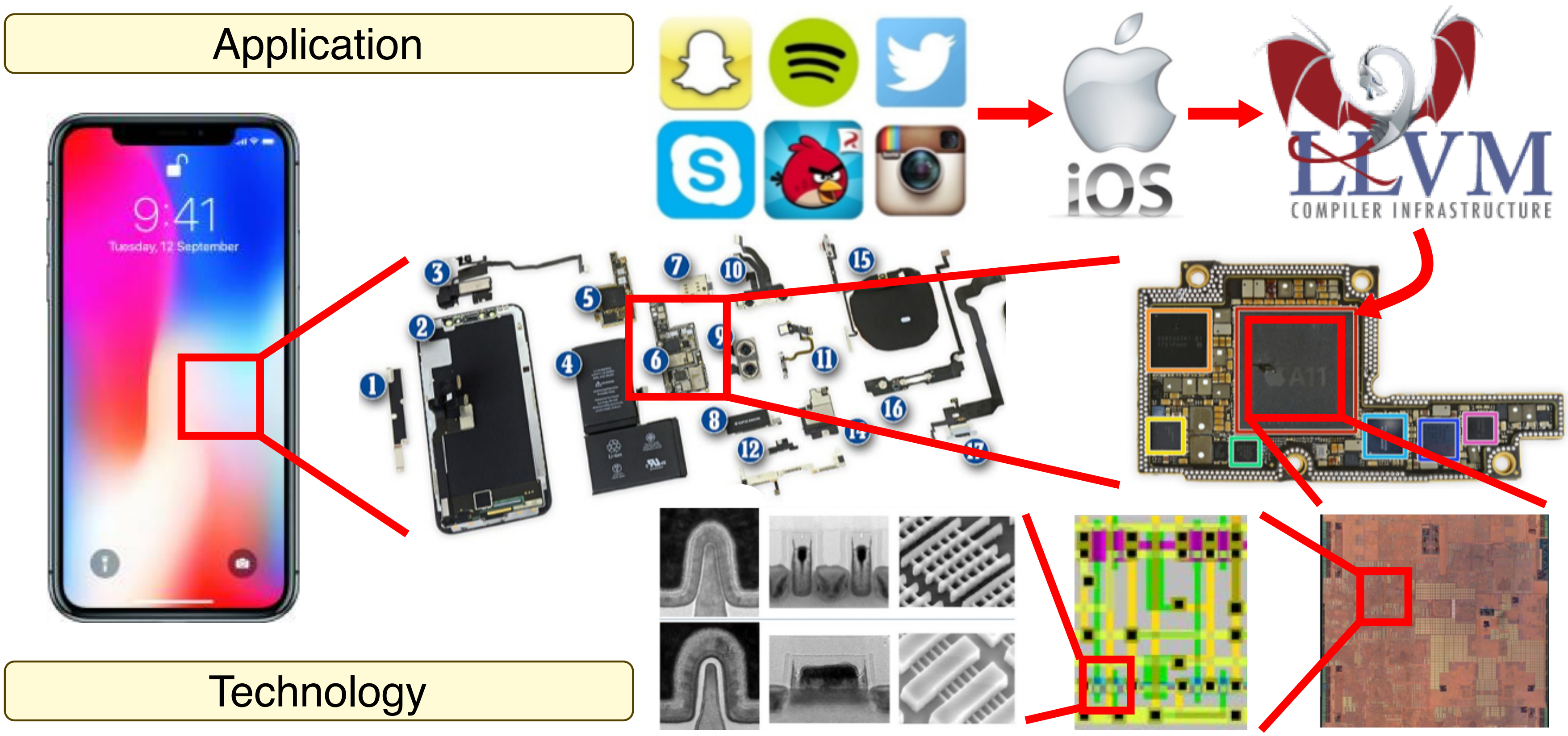
Application

Gap too large to bridge in one step
(but there are exceptions,
e.g., a magnetic compass)

Technology

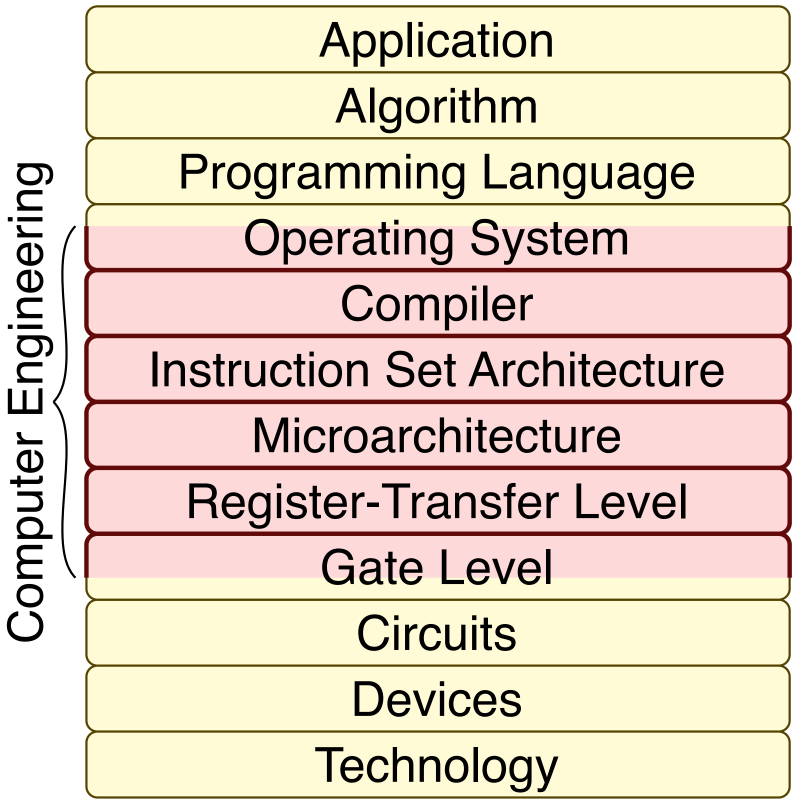
How can we manage all of this
hardware and software complexity?

The Complexity of Modern Computer Systems



How can we manage all of this hardware and software complexity?

The Computer Systems Stack



Sort an array of numbers

2,6,3,8,4,5 -> 2,3,4,5,6,8

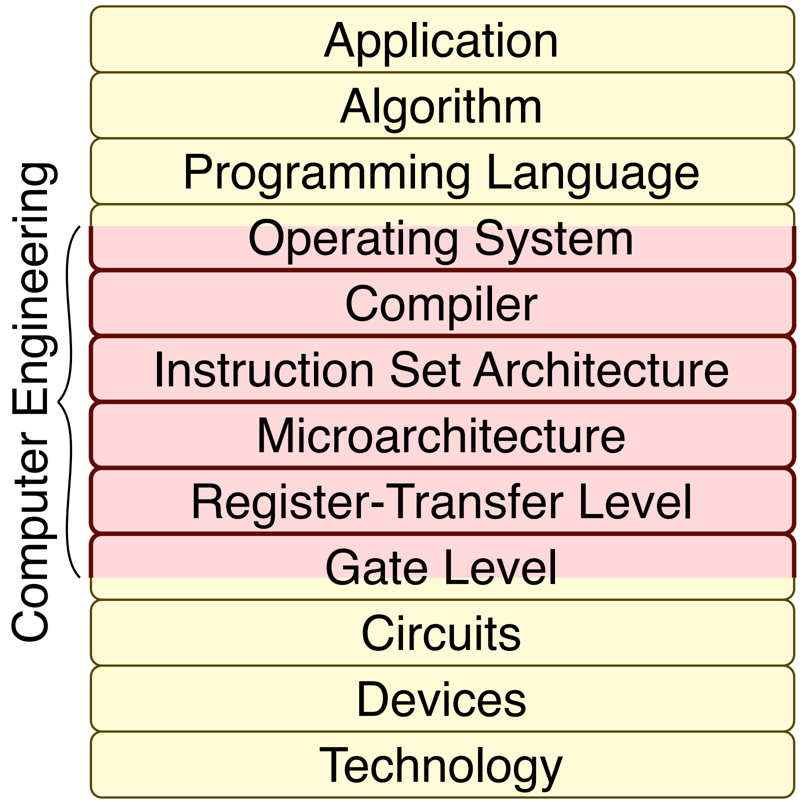
Out-of-place selection sort algorithm

1. Find minimum number in array
2. Move minimum number into output array
3. Repeat steps 1 and 2 until finished

C implementation of selection sort

```
void sort( int b[], int a[], int n ) {
  for ( int idx, k = 0; k < n; k++ ) {
    int min = 100;
    for ( int i = 0; i < n; i++ ) {
      if ( a[i] < min ) {
        min = a[i];
        idx = i;
      }
    }
    b[k]   = min;
    a[idx] = 100;
  }
}
```

The Computer Systems Stack



Mac OS X, Windows, Linux

Handles low-level hardware management

C Compiler

Transform programs into assembly

```
int a = b + c;      add $t0, $t1, $t2
A[i] = a;           sw  $t0, 0($t3)
```

RISC-V Instruction Set

Instructions that machine executes

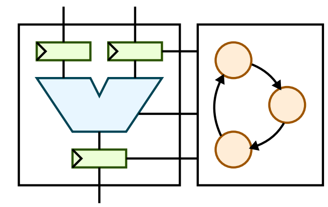
```
blez $a2, done
move $a7, $zero
li   $t4, 99
move $a4, $a1
li   $a3, 99
lw   $a5, 0($a4)
```

The Computer Systems Stack

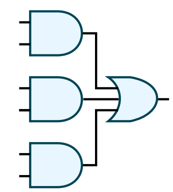
- Application
- Algorithm
- Programming Language
- Operating System
- Compiler
- Instruction Set Architecture
- Microarchitecture
- Register-Transfer Level
- Gate Level
- Circuits
- Devices
- Technology

Computer Engineering

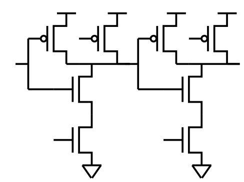
How data flows through system



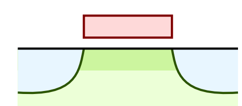
Boolean logic gates and functions



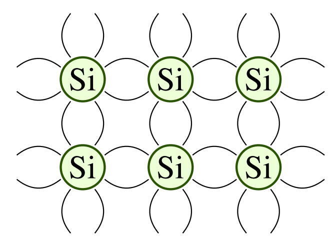
Combining devices to do useful work



Transistors and wires



Silicon process technology



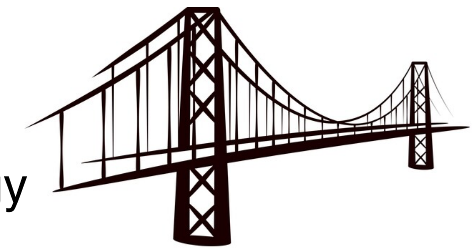
CS vs. Computer Engineering vs. EE

- Application
- Algorithm
- Programming Language
- Operating System
- Compiler
- Instruction Set Architecture
- Microarchitecture
- Register-Transfer Level
- Gate Level
- Circuits
- Devices
- Technology

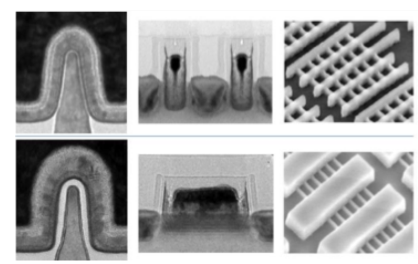
Computer Engineering

Traditional Computer Science

Computer Engineering is the bridge between applications & technology

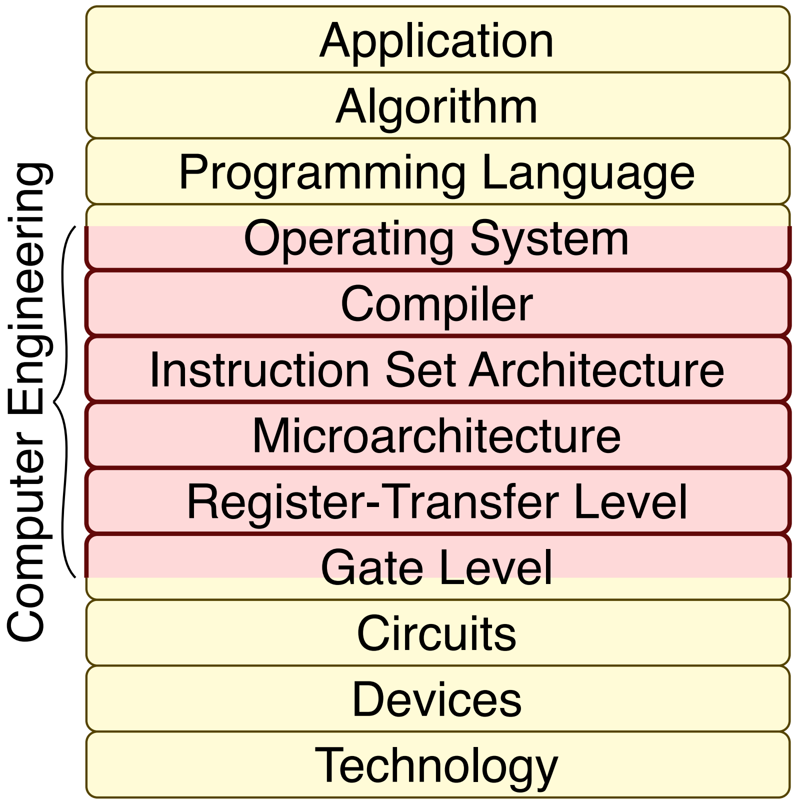


Traditional Electrical Engineering



In its broadest definition, computer engineering is the development of the abstraction/implementation layers that allow us to execute information processing applications efficiently using available manufacturing technologies

Cornell Computer Engineering Curriculum



CS 4820 Intro to Analysis of Algorithms
CS 3110 Data Struct & Functional Prog
CS 1110 / CS 1112 Intro to Computing
CS 2100 OO Programming & Data Struct
ECE 3140 Embedded Systems
CS 3410 Computer System Org & Prog
ECE 4760 Design with Microcontrollers
ECE 5760 Advanced SoC Design
ECE 4750 Computer Architecture
ECE 5745 Digital ASIC Design
ECE 2300 Digital Logic & Computer Org
ECE 3150 Microelectronics
ECE 4740 Digital VLSI
ECE 4360 Nanofabrication
ECE 4070 Physics of Semiconductors

- Application
- Algorithm
- PL
- OS
- Compiler
- ISA
- µArch
- RTL
- Gates
- Circuits
- Devices
- Technology

Agenda

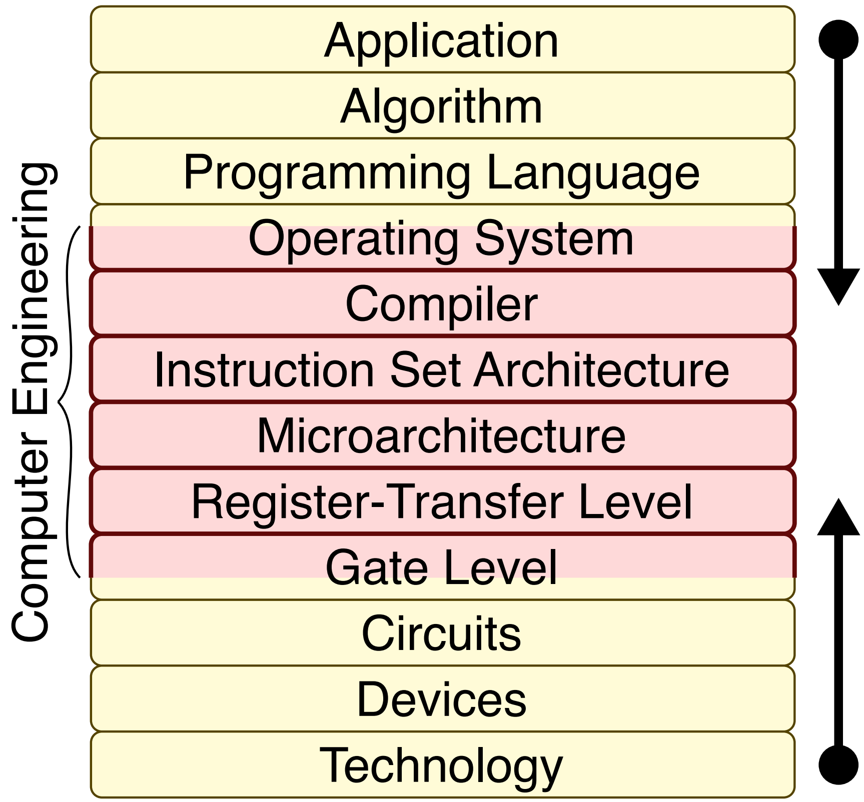
The Computer Systems Stack

Trends in Computer Engineering

- Trend #1: Bell's "Law"
- Trend #2: Moore's "Law"
- Trend #3: The Specialization Era

Cornell Custom Silicon Systems Project Team

Trends in Computer Engineering



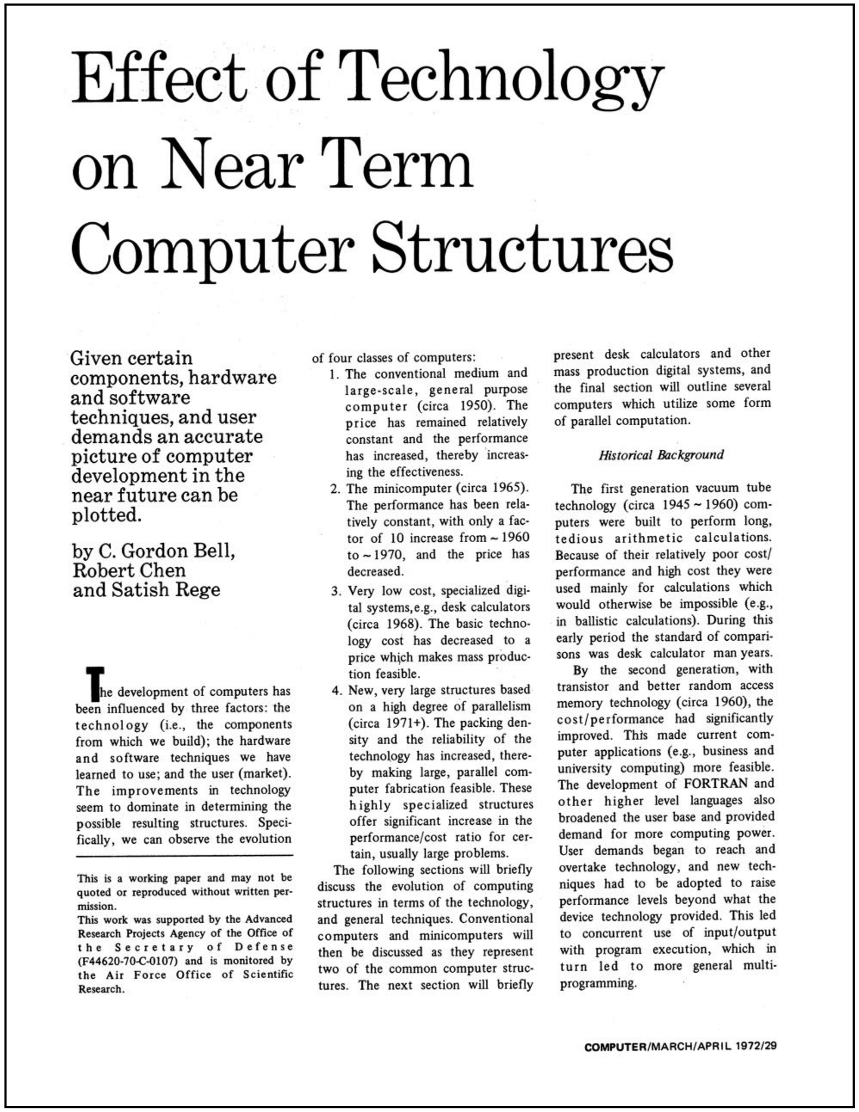
Bell's "Law"

How are application requirements changing?

Moore's "Law"

How are technology constraints changing?

Gordon Bell's "Law" of Computer Classes



Effect of Technology on Near Term Computer Structures

Given certain components, hardware and software techniques, and user demands an accurate picture of computer development in the near future can be plotted.

by C. Gordon Bell, Robert Chen and Satish Rege

The development of computers has been influenced by three factors: the technology (i.e., the components from which we build); the hardware and software techniques we have learned to use; and the user (market). The improvements in technology seem to dominate in determining the possible resulting structures. Specifically, we can observe the evolution of four classes of computers:

1. The conventional medium and large-scale, general purpose computer (circa 1950). The price has remained relatively constant and the performance has increased, thereby increasing the effectiveness.
2. The minicomputer (circa 1965). The performance has been relatively constant, with only a factor of 10 increase from ~1960 to ~1970, and the price has decreased.
3. Very low cost, specialized digital systems,e.g., desk calculators (circa 1968). The basic technology cost has decreased to a price which makes mass production feasible.
4. New, very large structures based on a high degree of parallelism (circa 1971+). The packing density and the reliability of the technology has increased, thereby making large, parallel computer fabrication feasible. These highly specialized structures offer significant increase in the performance/cost ratio for certain, usually large problems.

The following sections will briefly discuss the evolution of computing structures in terms of the technology, and general techniques. Conventional computers and minicomputers will then be discussed as they represent two of the common computer structures. The next section will briefly present desk calculators and other mass production digital systems, and the final section will outline several computers which utilize some form of parallel computation.

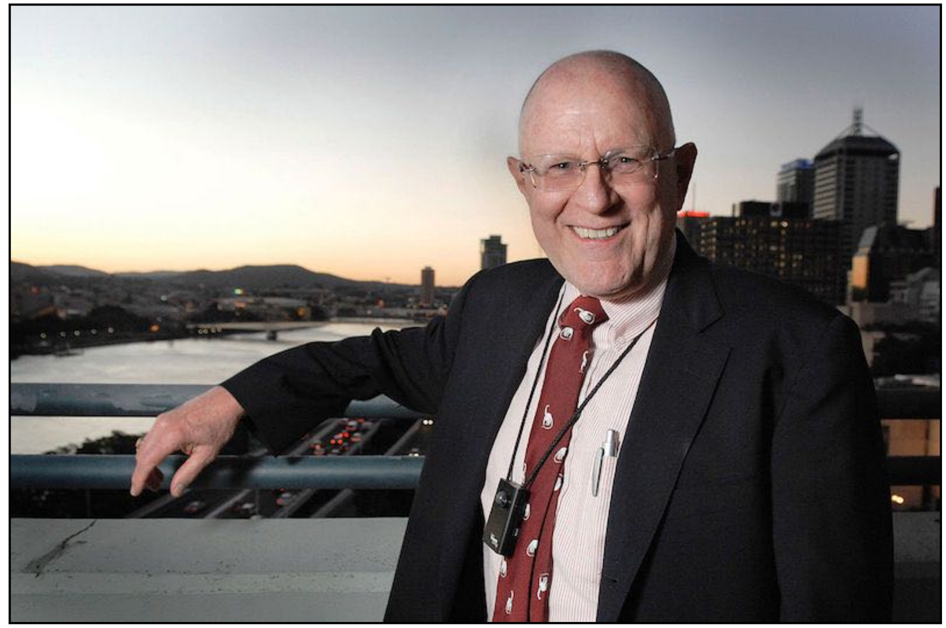
Historical Background

The first generation vacuum tube technology (circa 1945 ~ 1960) computers were built to perform long, tedious arithmetic calculations. Because of their relatively poor cost/performance and high cost they were used mainly for calculations which would otherwise be impossible (e.g., in ballistic calculations). During this early period the standard of comparisons was desk calculator man years.

By the second generation, with transistor and better random access memory technology (circa 1960), the cost/performance had significantly improved. This made current computer applications (e.g., business and university computing) more feasible. The development of FORTRAN and other higher level languages also broadened the user base and provided demand for more computing power. User demands began to reach and overtake technology, and new techniques had to be adopted to raise performance levels beyond what the device technology provided. This led to concurrent use of input/output with program execution, which in turn led to more general multiprogramming.

This is a working paper and may not be quoted or reproduced without written permission.

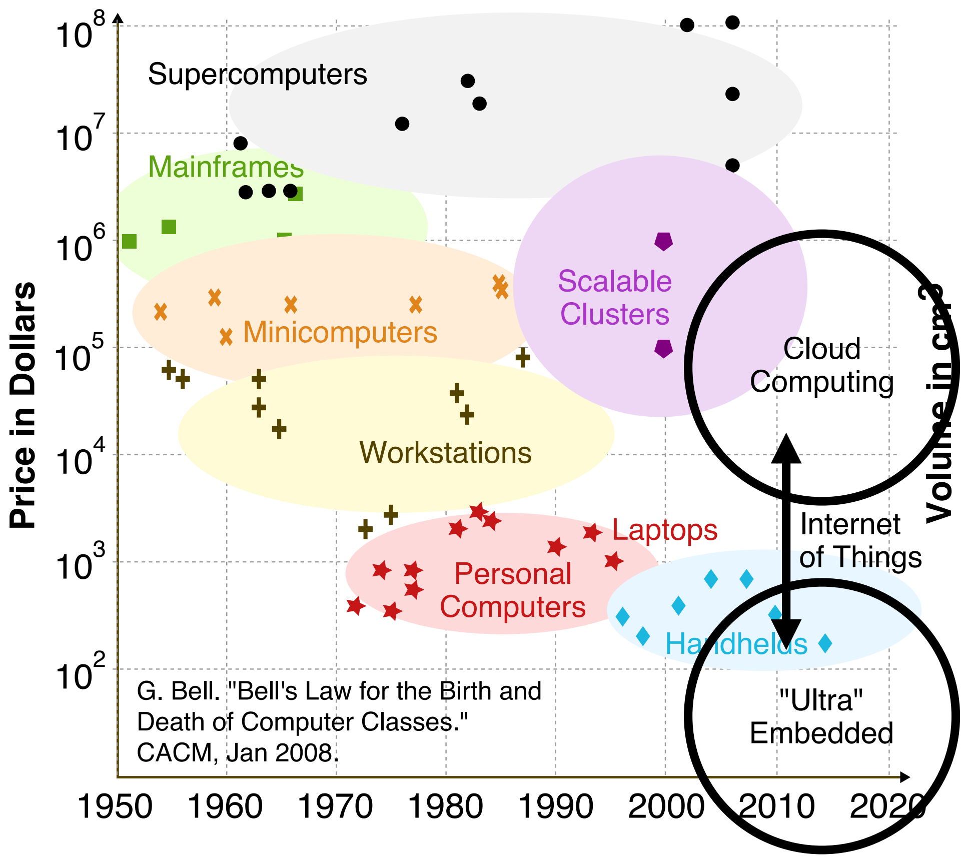
This work was supported by the Advanced Research Projects Agency of the Office of the Secretary of Defense (F44620-70-C-0107) and is monitored by the Air Force Office of Scientific Research.

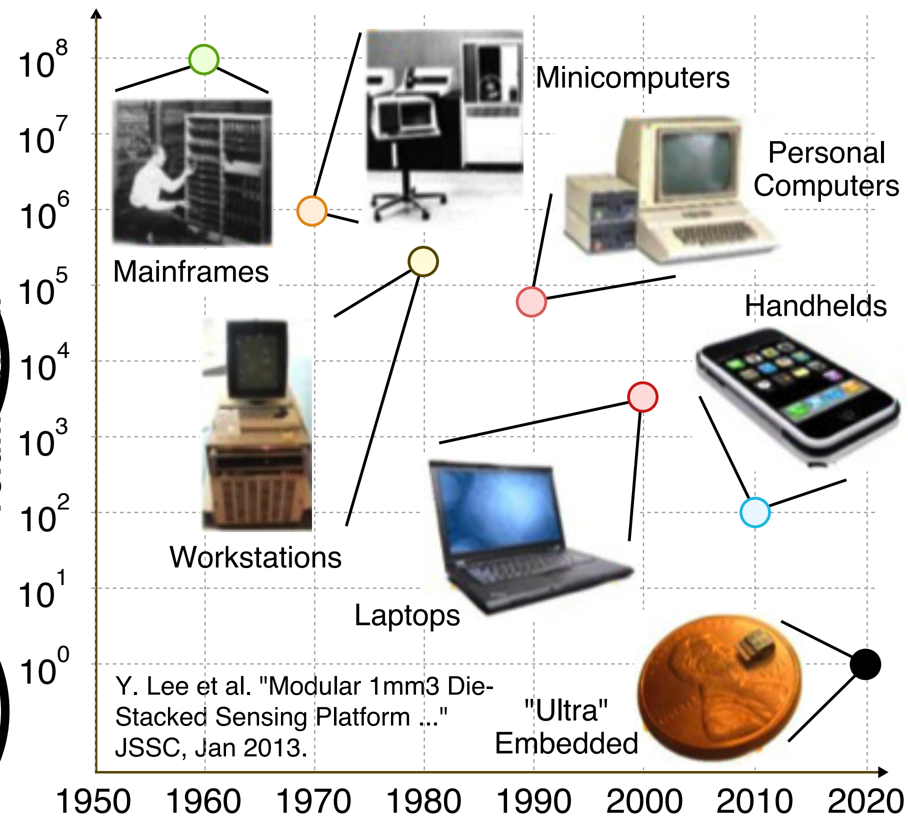


- Vice-President of Engineering at Digital Equipment Corporation
- Helped found Microsoft Research
- 1972 paper in IEEE Computer

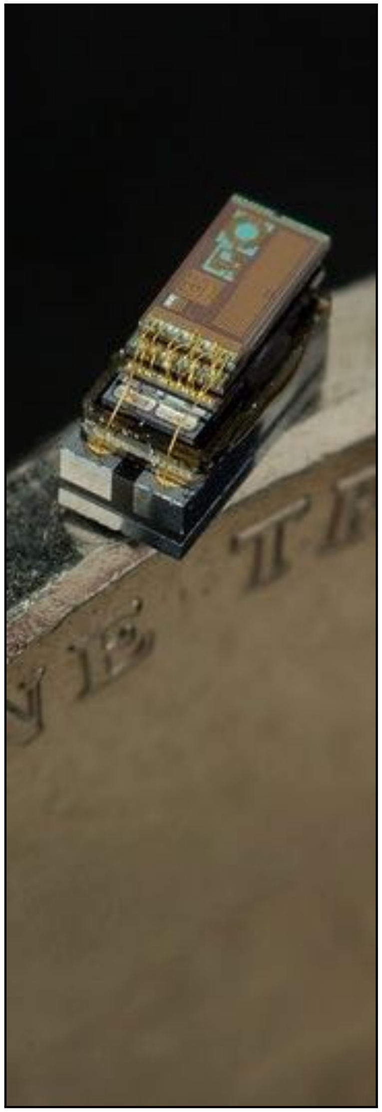
Trend #1: Bell's "Law"

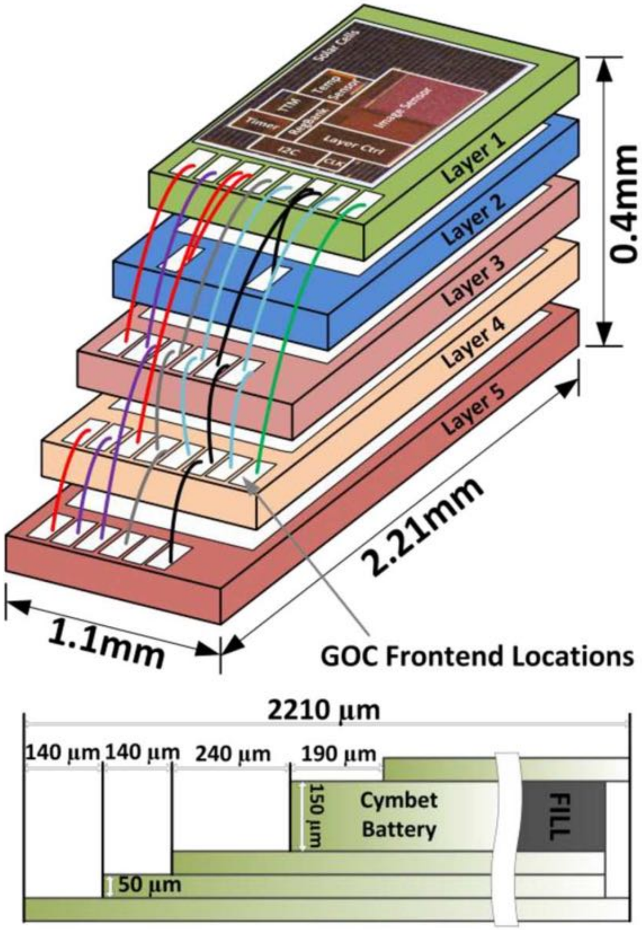
Roughly every decade a new, smaller, lower priced computer class forms based on a new programming platform resulting in entire new industries

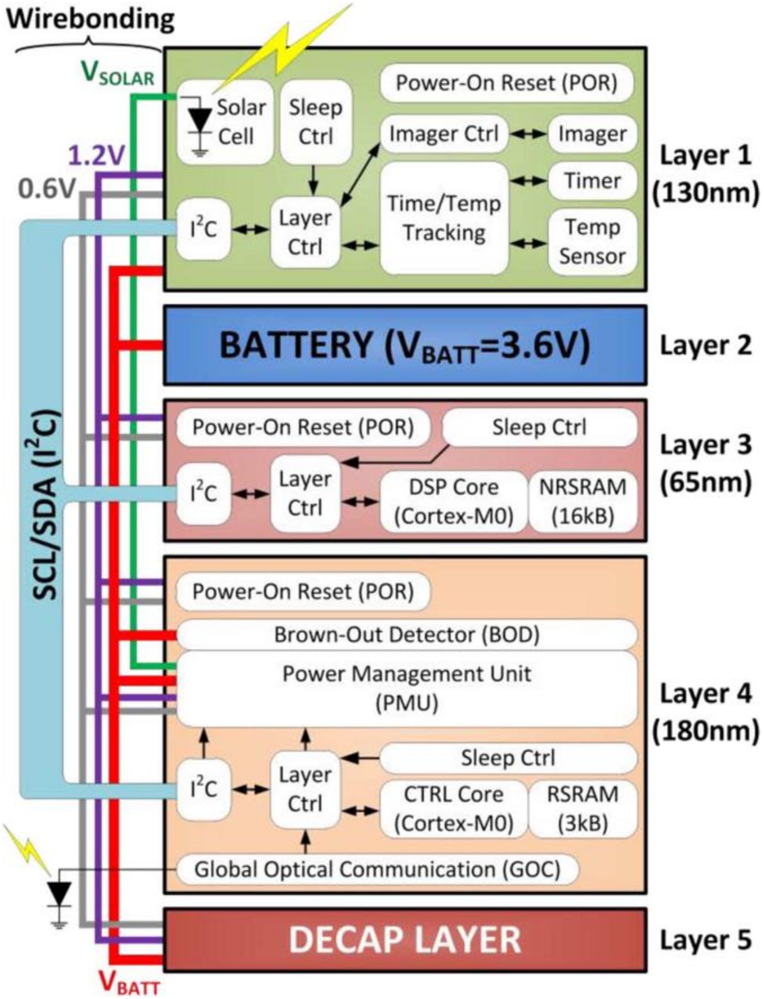




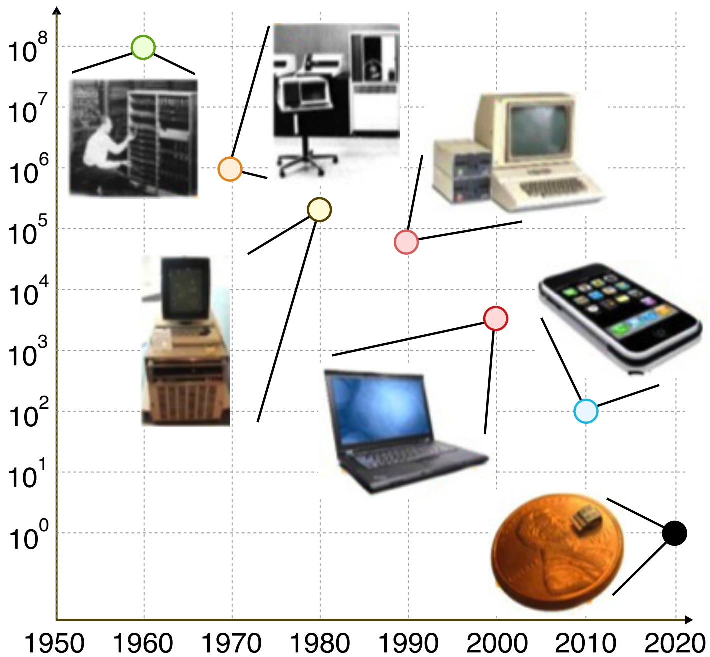
M3: Michigan Micro Mote







Adapted from Y. Lee et al., JSSC, 2013.



Trend #1: Bell's "Law"

Bell's "Law" predicts an Internet-of-Things and cloud computing which continuously demand more performance and better efficiency

Gordon Moore's "Law" of Technology Scaling

The experts look ahead

Cramming more components onto integrated circuits

With unit cost falling as the number of components per circuit rises, by 1975 economics may dictate squeezing as many as 65,000 components on a single silicon chip

By Gordon E. Moore

Director, Research and Development Laboratories, Fairchild Semiconductor division of Fairchild Camera and Instrument Corp.

The future of integrated electronics is the future of electronics itself. The advantages of integration will bring about a proliferation of electronics, pushing this science into many new areas.

Integrated circuits will lead to such wonders as home computers—or at least terminals connected to a central computer—automatic controls for automobiles, and personal portable communications equipment. The electronic wristwatch needs only a display to be feasible today.

But the biggest potential lies in the production of large systems. In telephone communications, integrated circuits in digital filters will separate channels on multiplex equipment. Integrated circuits will also switch telephone circuits and perform data processing.

Computers will be more powerful, and will be organized in completely different ways. For example, memories built of integrated electronics may be distributed throughout the machine instead of being concentrated in a central unit. In addition, the improved reliability made possible by integrated circuits will allow the construction of larger processing units. Machines similar to those in existence today will be built at lower costs and with faster turn-around.

Present and future

By integrated electronics, I mean all the various technologies which are referred to as microelectronics today as well as any additional ones that result in electronics functions supplied to the user as irreducible units. These technologies were first investigated in the late 1950's. The object was to miniaturize electronics equipment to include increasingly complex electronic functions in limited space with minimum weight. Several approaches evolved, including microassembly techniques for individual components, thin-film structures and semiconductor integrated circuits.

Each approach evolved rapidly and converged so that each borrowed techniques from another. Many researchers believe the way of the future to be a combination of the various approaches.

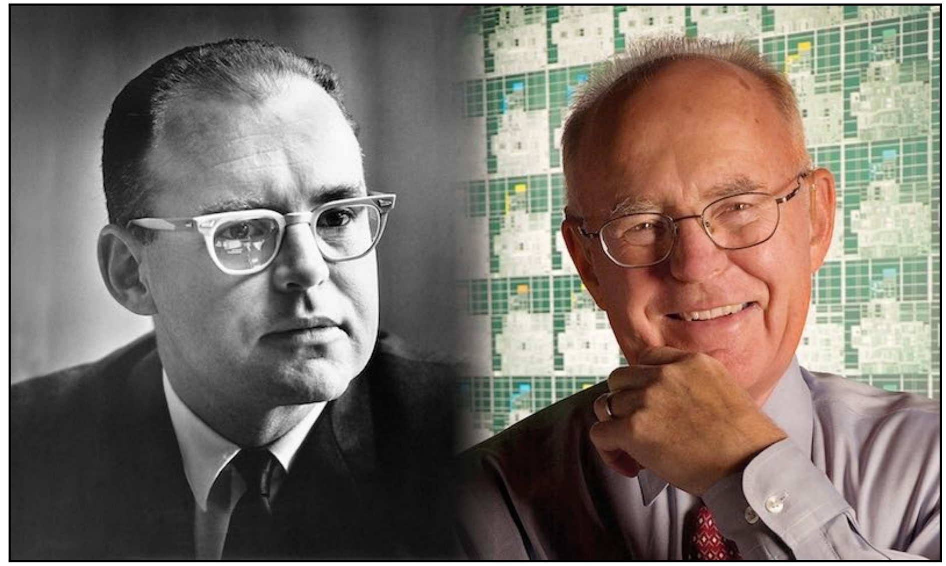
The advocates of semiconductor integrated circuitry are already using the improved characteristics of thin-film resistors by applying such films directly to an active semiconductor substrate. Those advocating a technology based upon films are developing sophisticated techniques for the attachment of active semiconductor devices to the passive film arrays.

Both approaches have worked well and are being used in equipment today.

The author

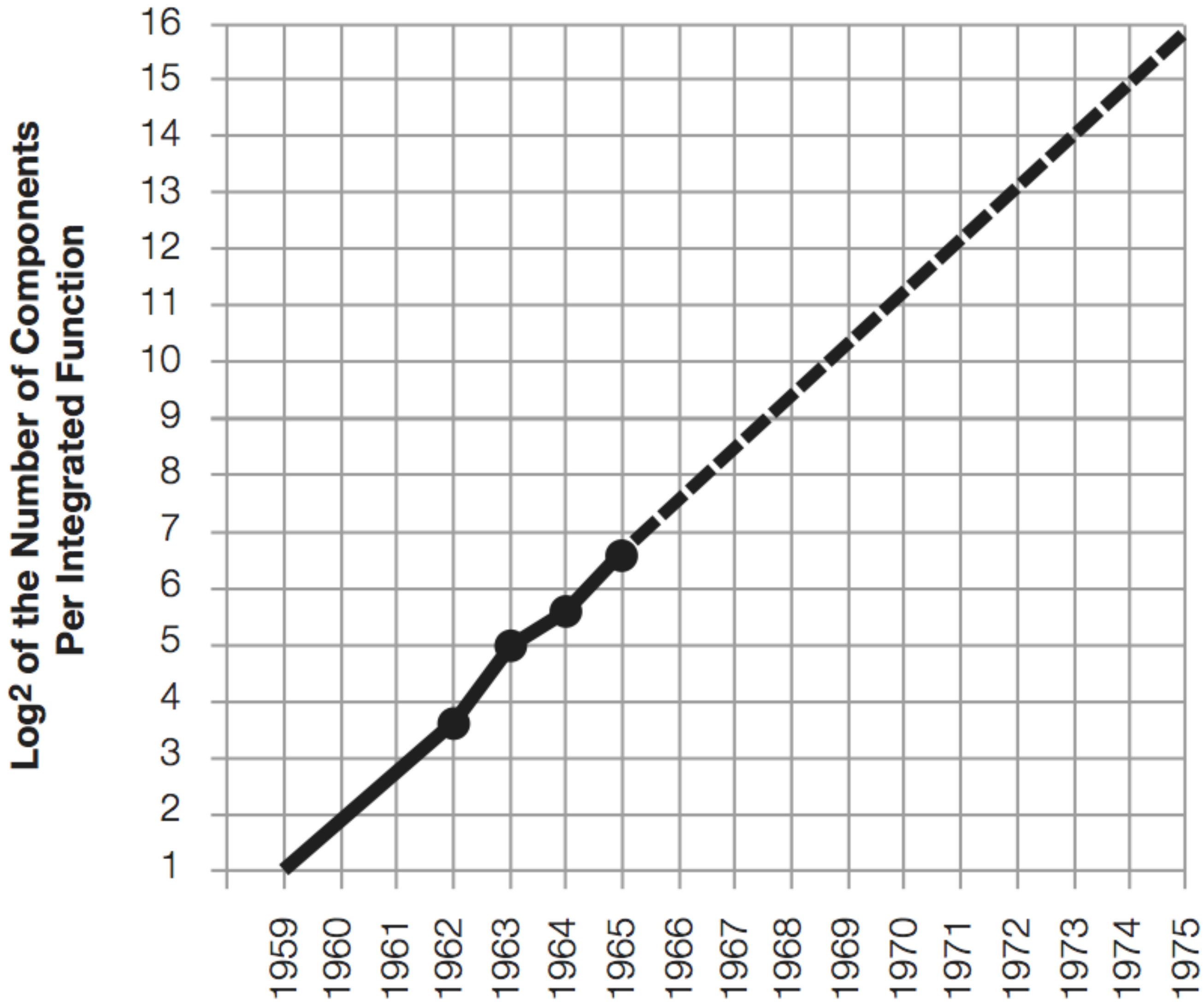
Dr. Gordon E. Moore is one of the new breed of electronic engineers, schooled in the physical sciences rather than in electronics. He earned a B.S. degree in chemistry from the University of California and a Ph.D. degree in physical chemistry from the California Institute of Technology. He was one of the founders of Fairchild Semiconductor and has been director of the research and development laboratories since 1959.

Electronics, Volume 38, Number 8, April 19, 1965



- Co-founder of Fairchild Semiconductor
- Co-founder of Intel Corp
- 1965 paper in Electronics Magazine

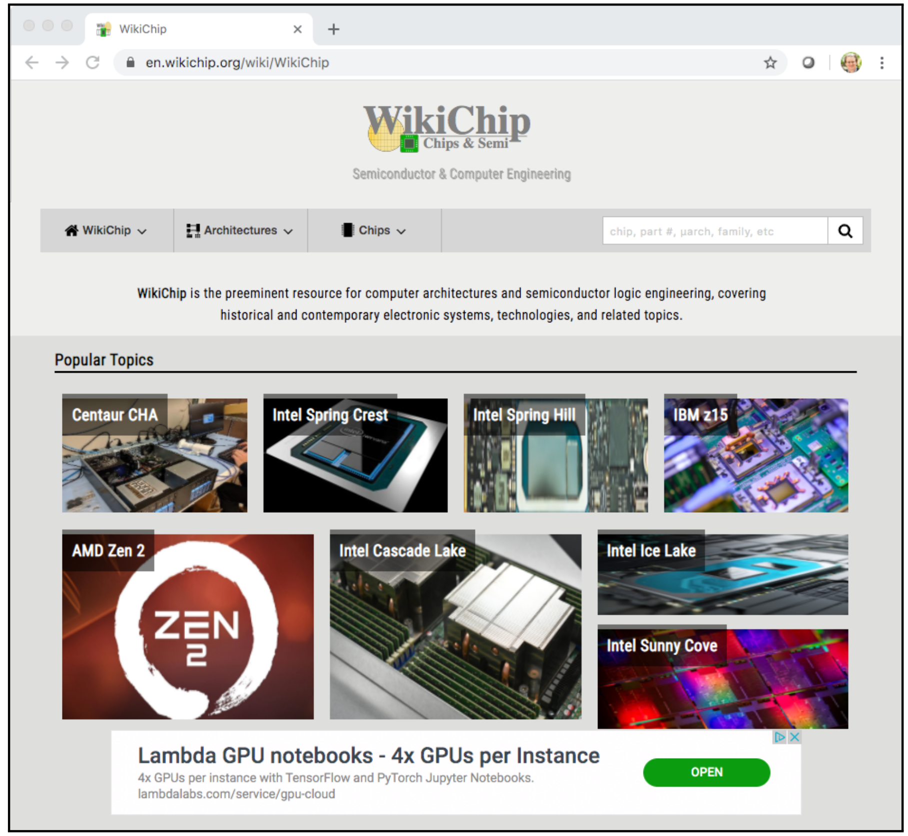
Gordon Moore's "Law" of Technology Scaling

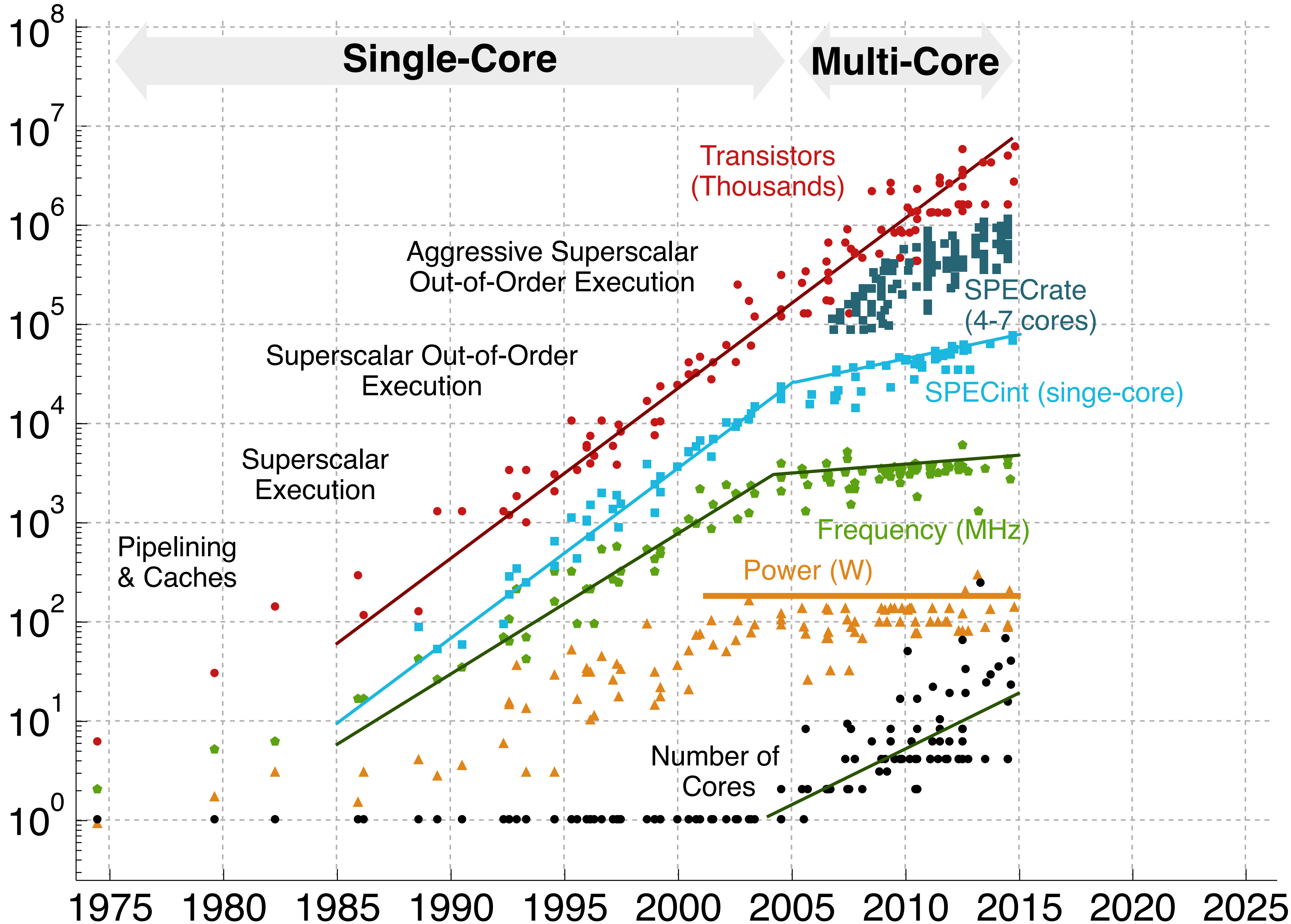


Activity: Specifications of Modern Processors

`http://tiny.cc/engri1210-2`

1. Breakout into groups of 3 students
2. Browse WikiChip or Intel Ark
3. Find a few processors
4. Enter year, frequency, core count, power in Google form



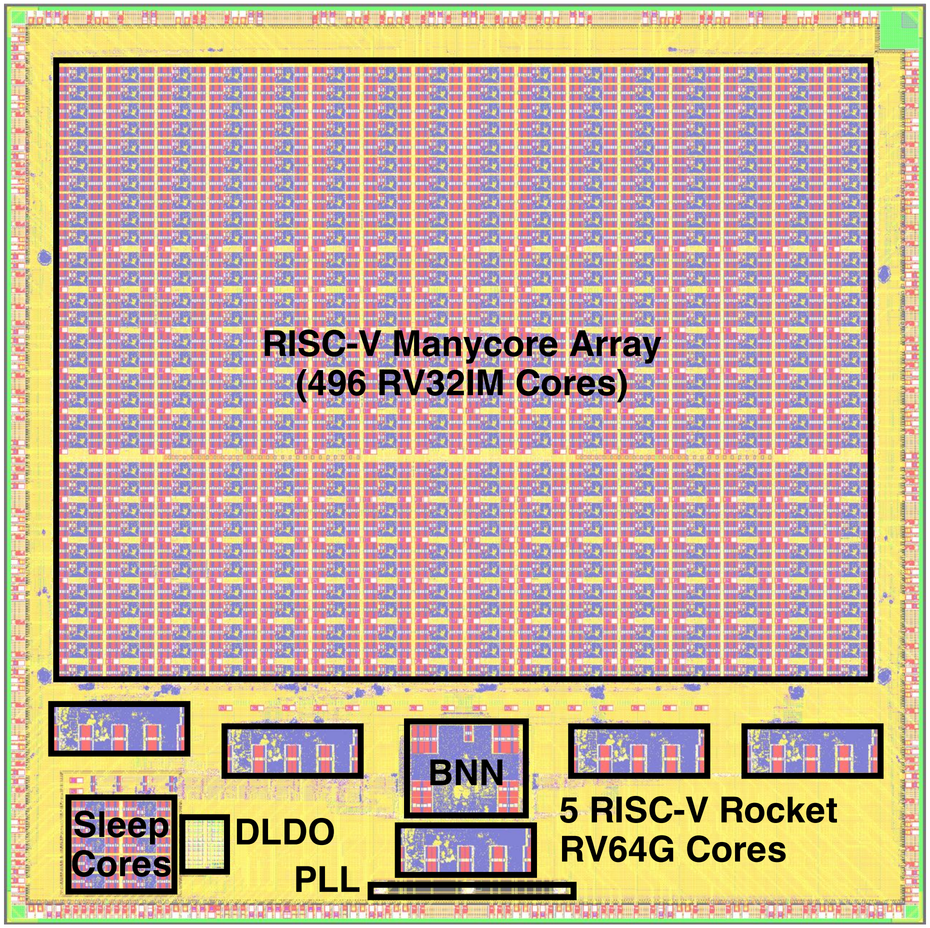


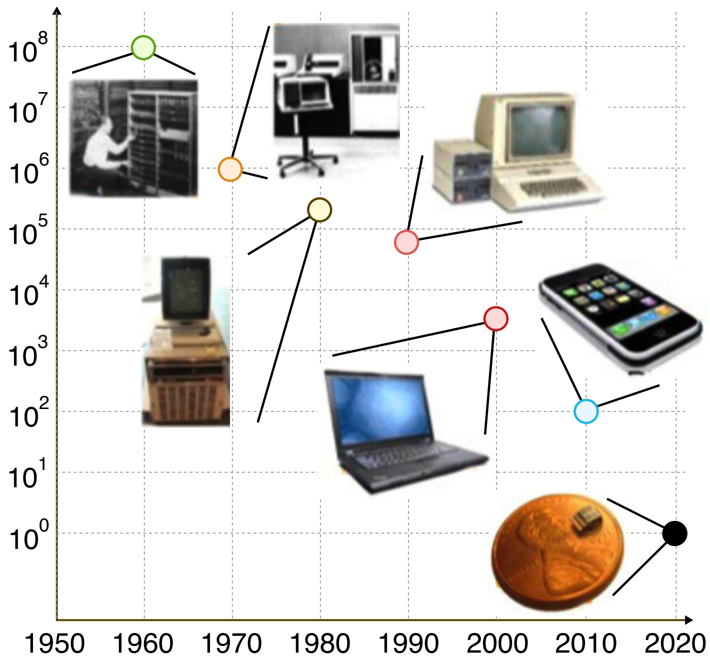
C. Batten, M. Horowitz, F. Labonte, O. Shacham, K. Olukotun, L. Hammond, K. Rupp & [Y. Shao, IEEE Micro'15] & [C. Leiserson, Science'20]

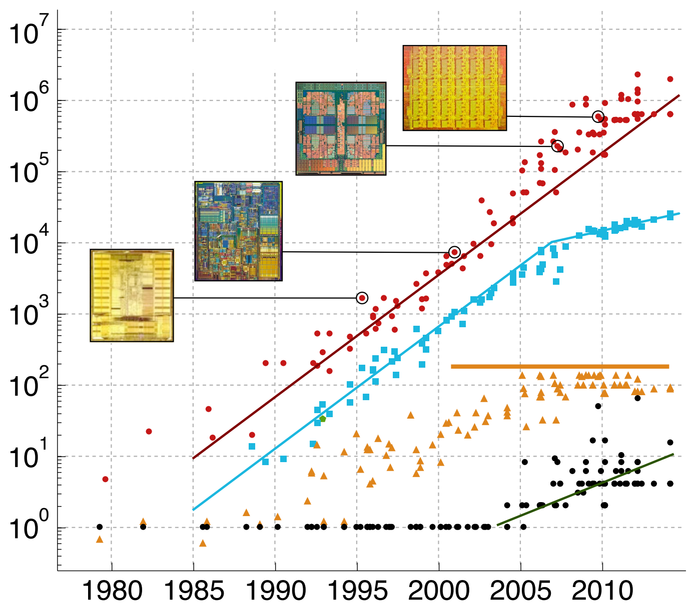
Celerity System-on-Chip

UCSD, Washington, Cornell, Michigan w/ DARPA CRAFT Program

- 5×5mm in TSMC 16 nm FFC
- 385 million transistors
- **511 RISC-V cores**
  - 5 Linux-capable Rocket cores
  - 496-core tiled manycore
  - 10-core low-voltage array
- 1 BNN accelerator
- 1 synthesizable PLL
- 1 synthesizable LDO Vreg
- 3 clock domains
- 672-pin flip chip BGA package
- 9-months from PDK access to tape-out







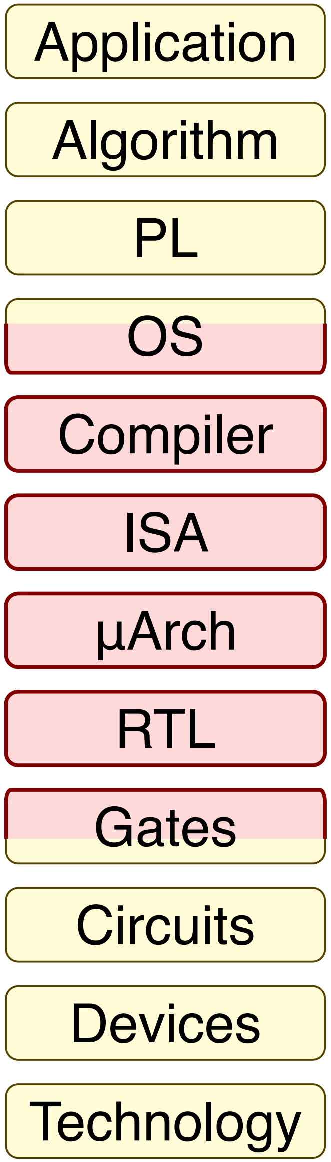
Trend #1: Bell's "Law"

Bell's "Law" predicts an Internet-of-Things and cloud computing which continuously demand more performance and better efficiency

Trend #2: Moore's "Law"

Moore's "Law" predicts an exponential increasing number of transistors per chip, but power limitations have motivated a move to multicore processors

Unfortunately, multicore processors are not enough. What else can we do to use more transistors to meet the needs of IoT and cloud?



Agenda

The Computer Systems Stack

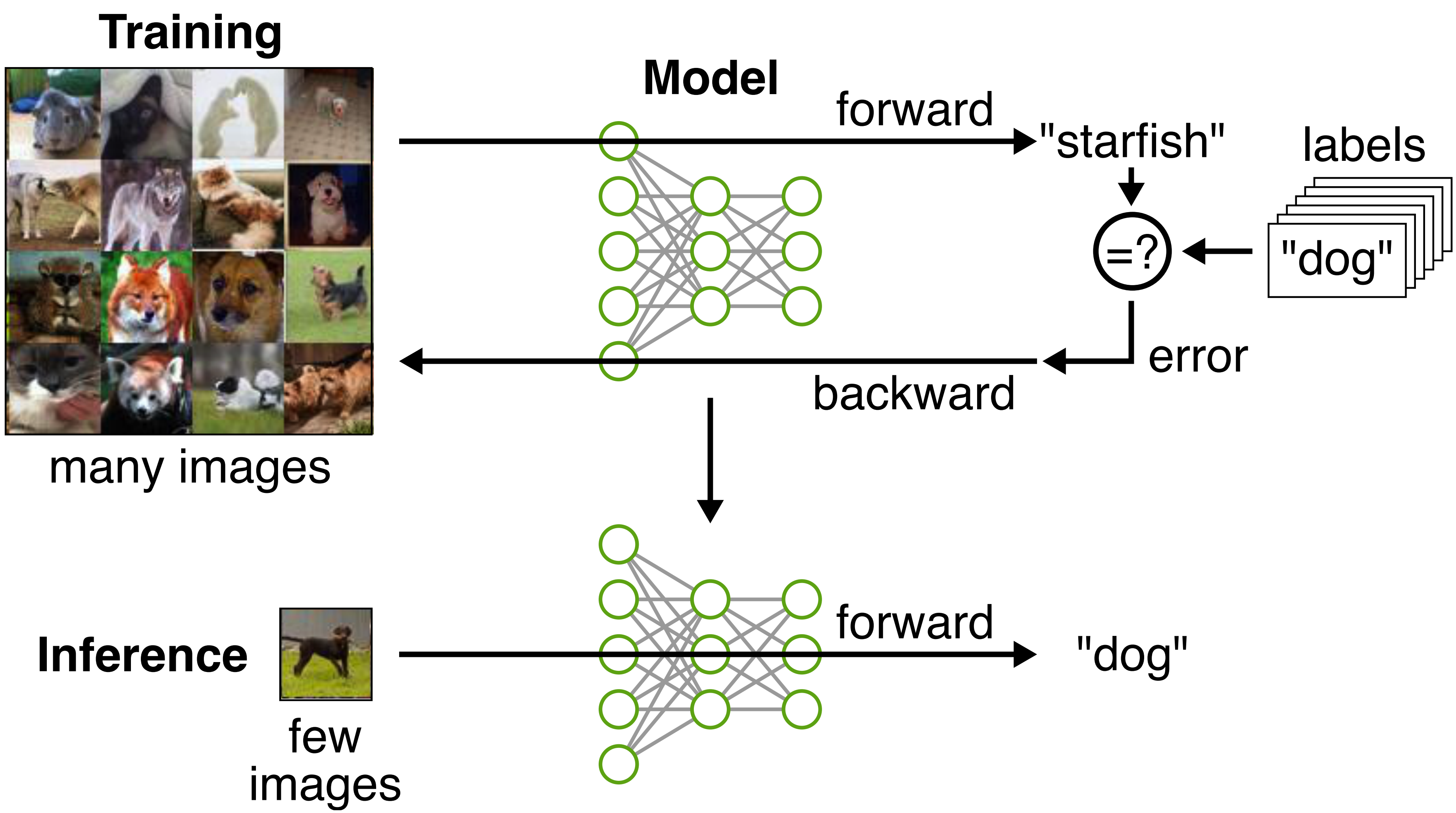
Trends in Computer Engineering

- Trend #1: Bell's "Law"
- Trend #2: Moore's "Law"
- Trend #3: The Specialization Era

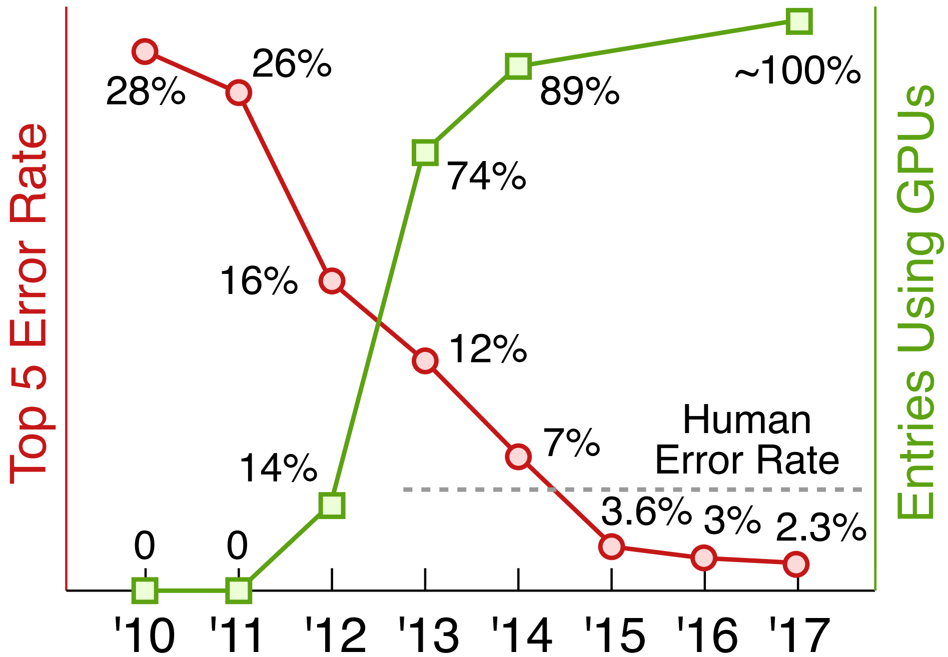
Cornell Custom Silicon Systems Project Team

Example Application Domain: Image Recognition

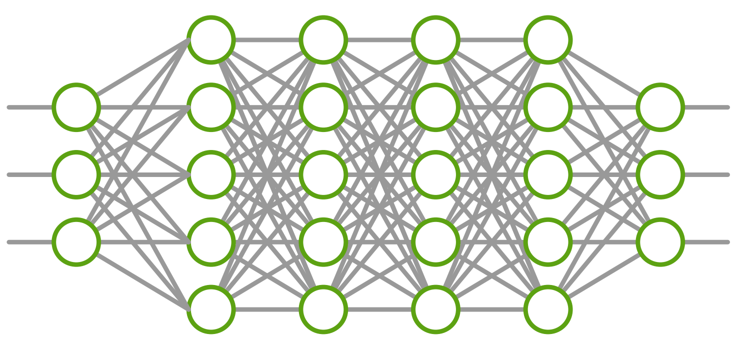
Machine Learning: Training vs. Inference



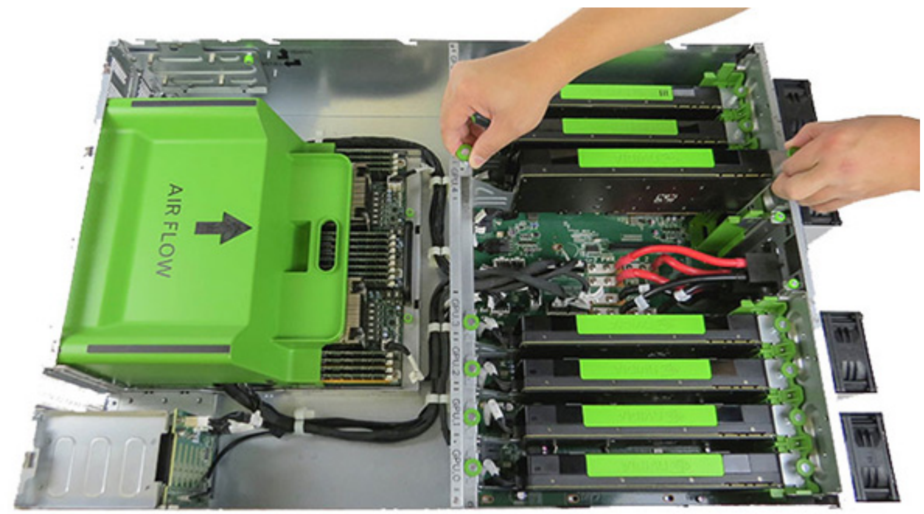
ImageNet Large-Scale Visual Recognition Challenge



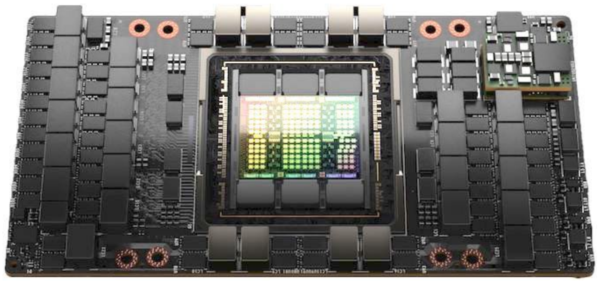
Hardware: Graphics Processing Units



Software: Deep Neural Networks

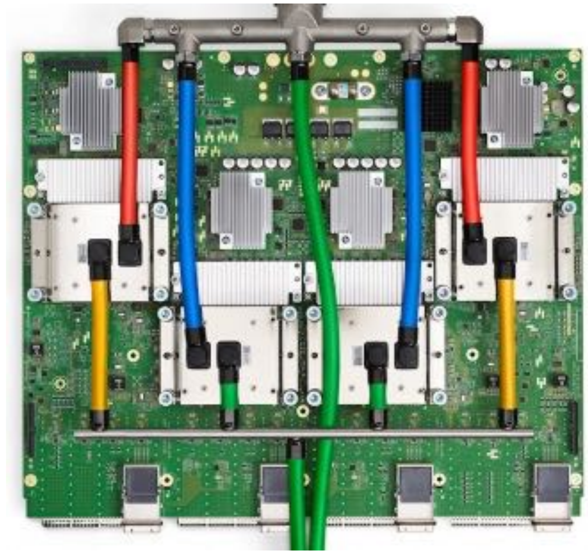


Accelerators for Machine Learning in the Cloud



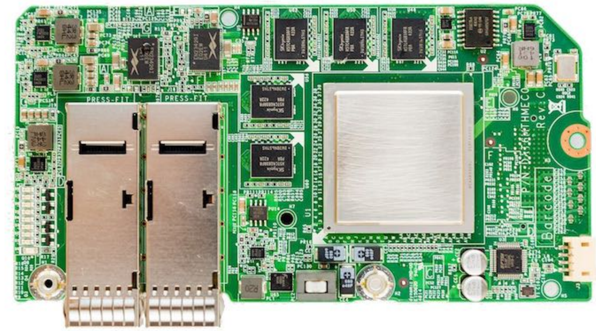
NVIDIA DGX Hopper

- Graphics processor specialized just for accelerating machine learning
- Available as part of a complete system with both the software and hardware designed by NVIDIA



Google TPU v4

- Custom chip specifically designed to accelerate Google's TensorFlow C++ library
- Tightly integrated into Google's data centers



Microsoft Catapult

- Custom FPGA board for accelerating Bing search and machine learning
- Accelerators developed with/by app developers
- Tightly integrated into Microsoft data center's and cloud computing platforms

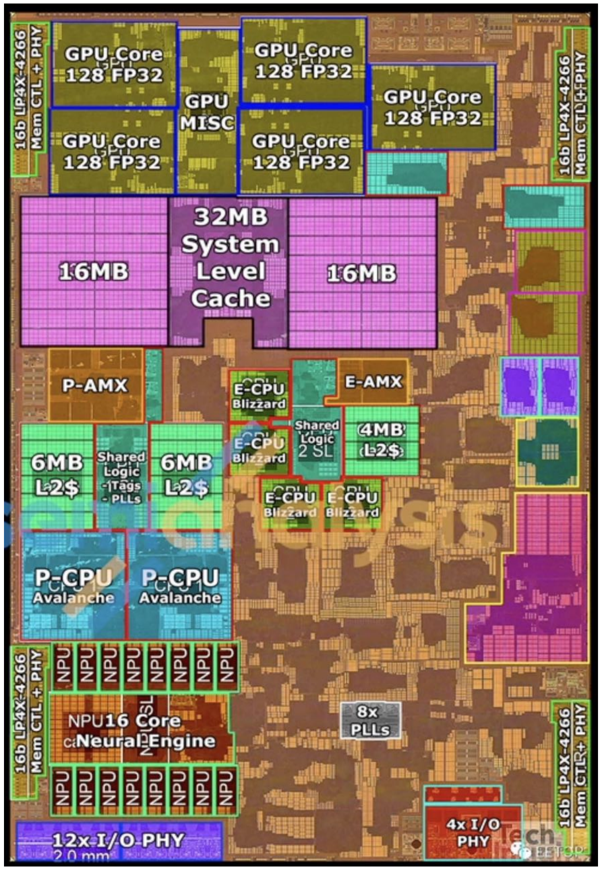
Accelerators for Machine Learning at the Edge

Amazon Echo

- Developing AI chips so Echo line can do more on-board processing
- Reduces need for round-trip to cloud
- Co-design the algorithms and the underlying hardware

Facebook Oculus

- Starting to design custom chips for Oculus VR headsets
- Significant performance demands under strict power requirements



Apple A15 Bionic

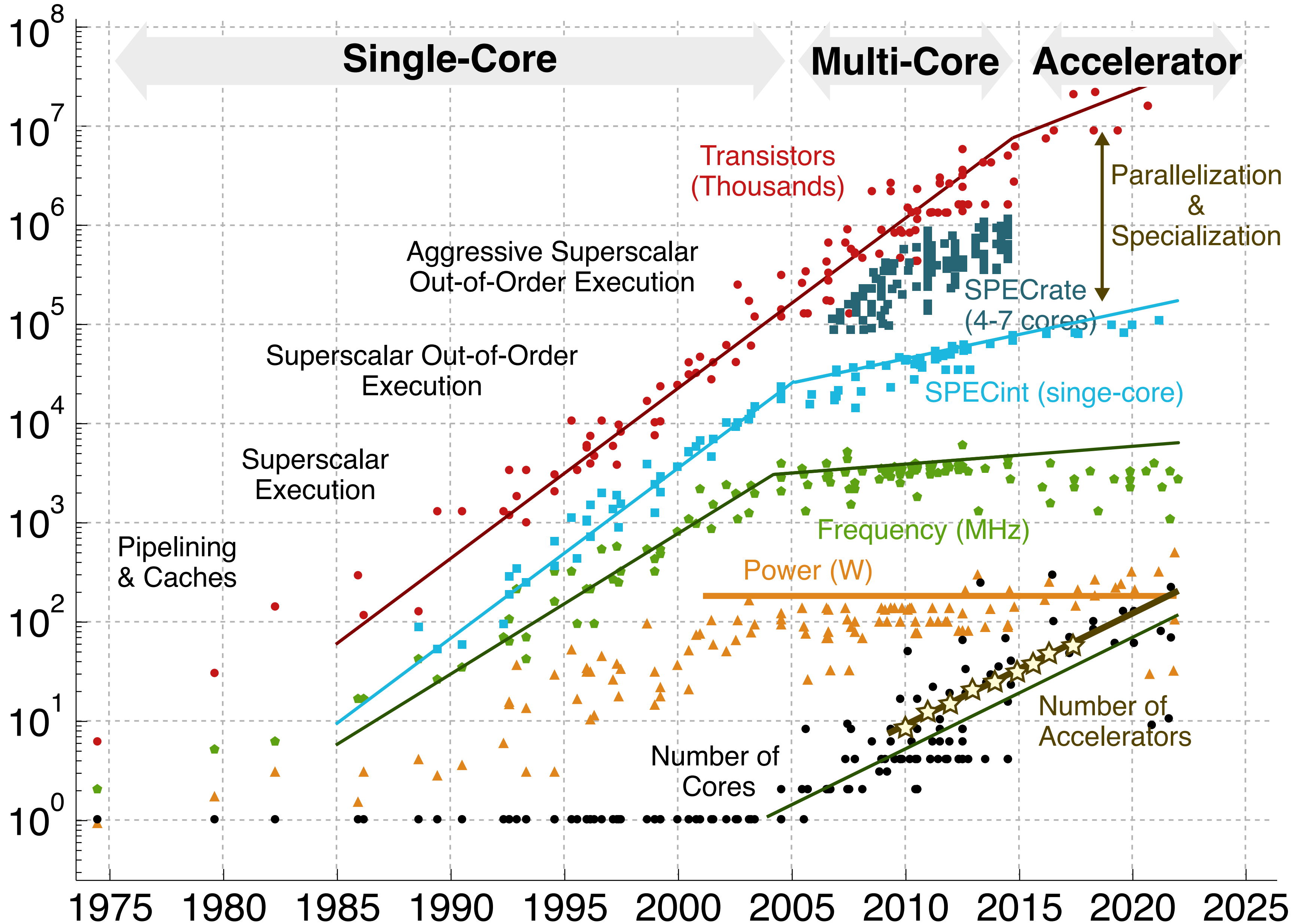
- 16-core Neural Engine

Top-five software companies are all building custom accelerators

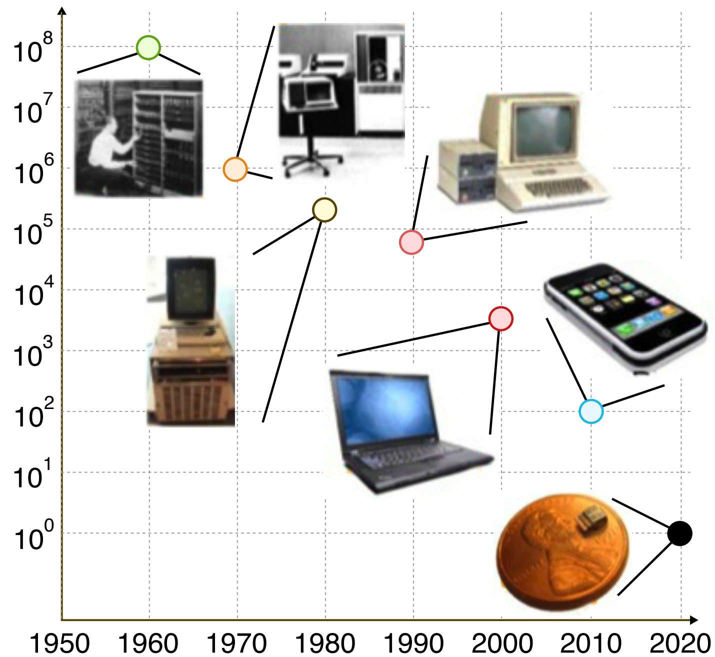
- **Facebook:** w/ Intel, in-house AI chips
- **Amazon:** Echo, Oculus, networking chips
- **Microsoft:** Hiring for AI chips
- **Google:** TPU, Pixel, convergence
- **Apple:** SoCs for phones and laptops

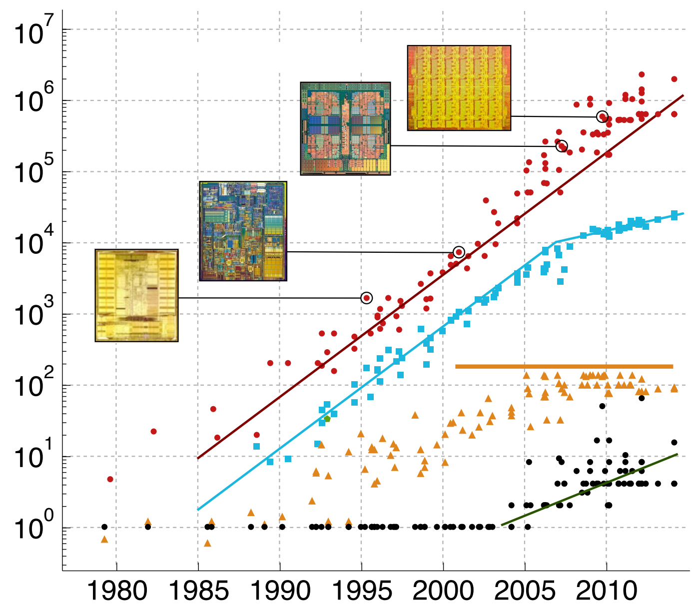
Chip startup ecosystem for machine learning accelerators is thriving!

- **Graphcore**
- **Nervana**
- **Cerebras**
- **Wave Computing**
- **Horizon Robotics**
- **Cambricon**
- **DeePhi**
- **Esperanto**
- **SambaNova**
- **Eyeriss**
- **Tenstorrent**
- **Mythic**
- **ThinkForce**
- **Groq**
- **Lightmatter**



C. Batten, M. Horowitz, F. Labonte, O. Shacham, K. Olukotun, L. Hammond, K. Rupp & [Y. Shao, IEEE Micro'15] & [C. Leiserson, Science'20]





Trend #1: Bell's "Law"

Bell's "Law" predicts an Internet-of-Things and cloud computing which continuously demand more performance and better efficiency

Trend #2: Moore's "Law"

Moore's "Law" predicts an exponential increasing number of transistors per chip, but power limitations have motivated a move to multicore processors

Trend #3: The Specialization Era

Hardware specialization can use wealth of transistors to meet needs of IoT and cloud

Application
Algorithm
PL
OS
Compiler
ISA
µArch
RTL
Gates
Circuits
Devices
Technology

Agenda

The Computer Systems Stack

Trends in Computer Engineering

- Trend #1: Bell's "Law"
- Trend #2: Moore's "Law"
- Trend #3: The Specialization Era

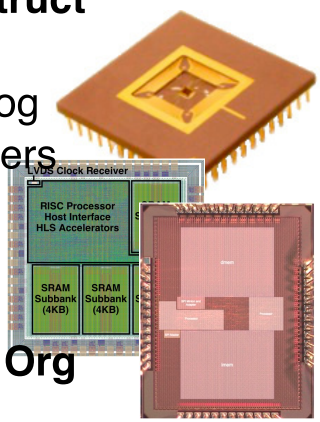
Cornell Custom Silicon Systems Project Team

Cornell Computer Engineering Curriculum

Computer Engineering (Operating System through Gate Level)

- Application
- Algorithm
- Programming Language
- Operating System
- Compiler
- Instruction Set Architecture
- Microarchitecture
- Register-Transfer Level
- Gate Level
- Circuits
- Devices
- Technology

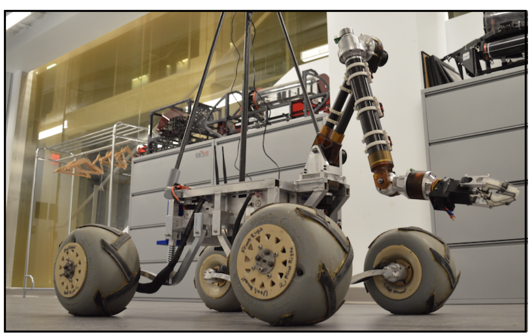
CS 4820 Intro to Analysis of Algorithms
CS 3110 Data Struct & Functional Prog
CS 1110 / CS 1112 Intro to Computing
CS 2100 OO Programming & Data Struct
ECE 3140 Embedded Systems
CS 3410 Computer System Org & Prog
ECE 4760 Design with Microcontrollers
ECE 5760 Advanced SoC Design
ECE 4750 Computer Architecture
ECE 5745 Digital ASIC Design
ECE 2300 Digital Logic & Computer Org
ECE 3150 Microelectronics
ECE 4740 Digital VLSI
ECE 4360 Nanofabrication
ECE 4070 Physics of Semiconductors



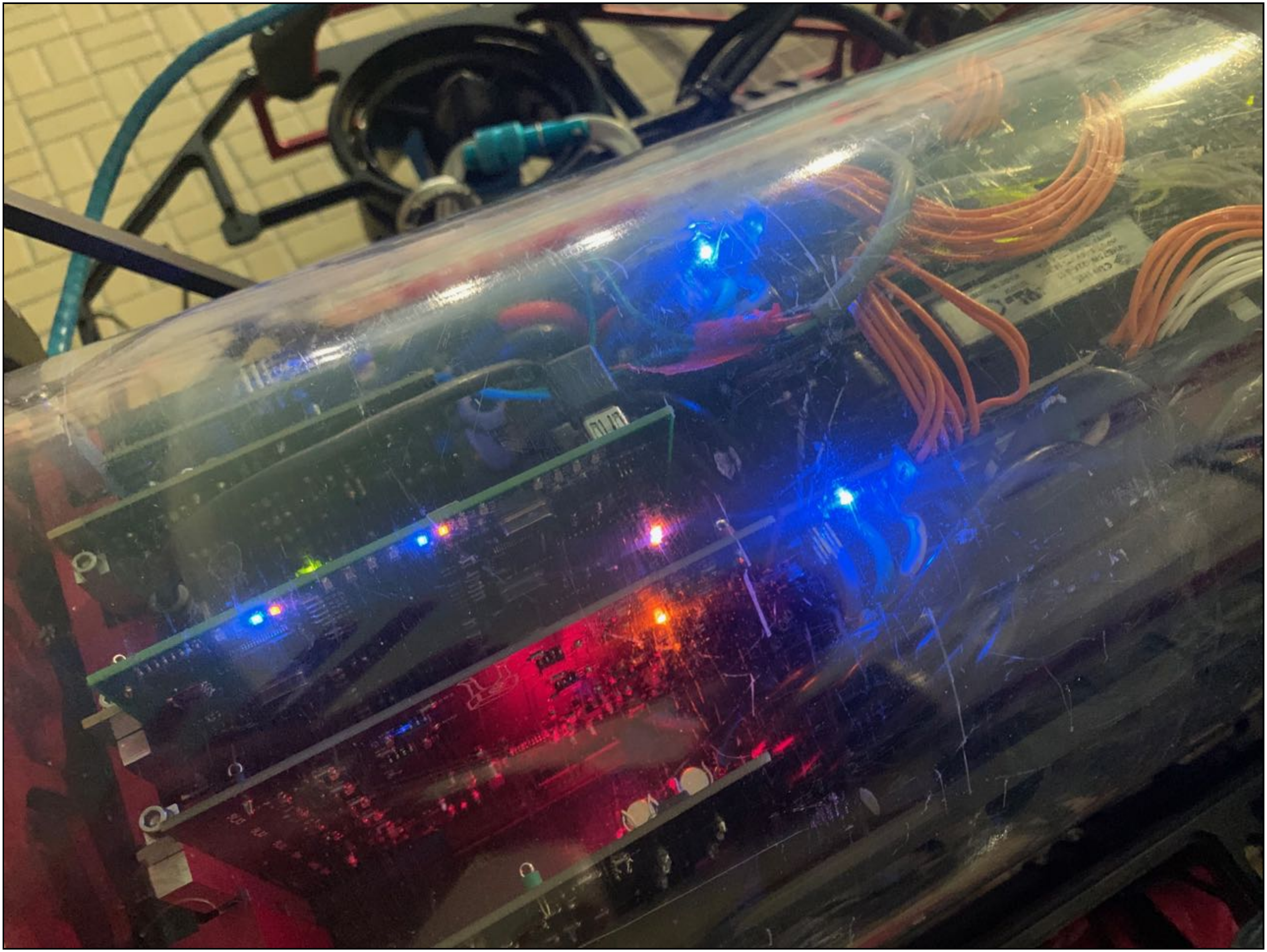
Conventional Wisdom

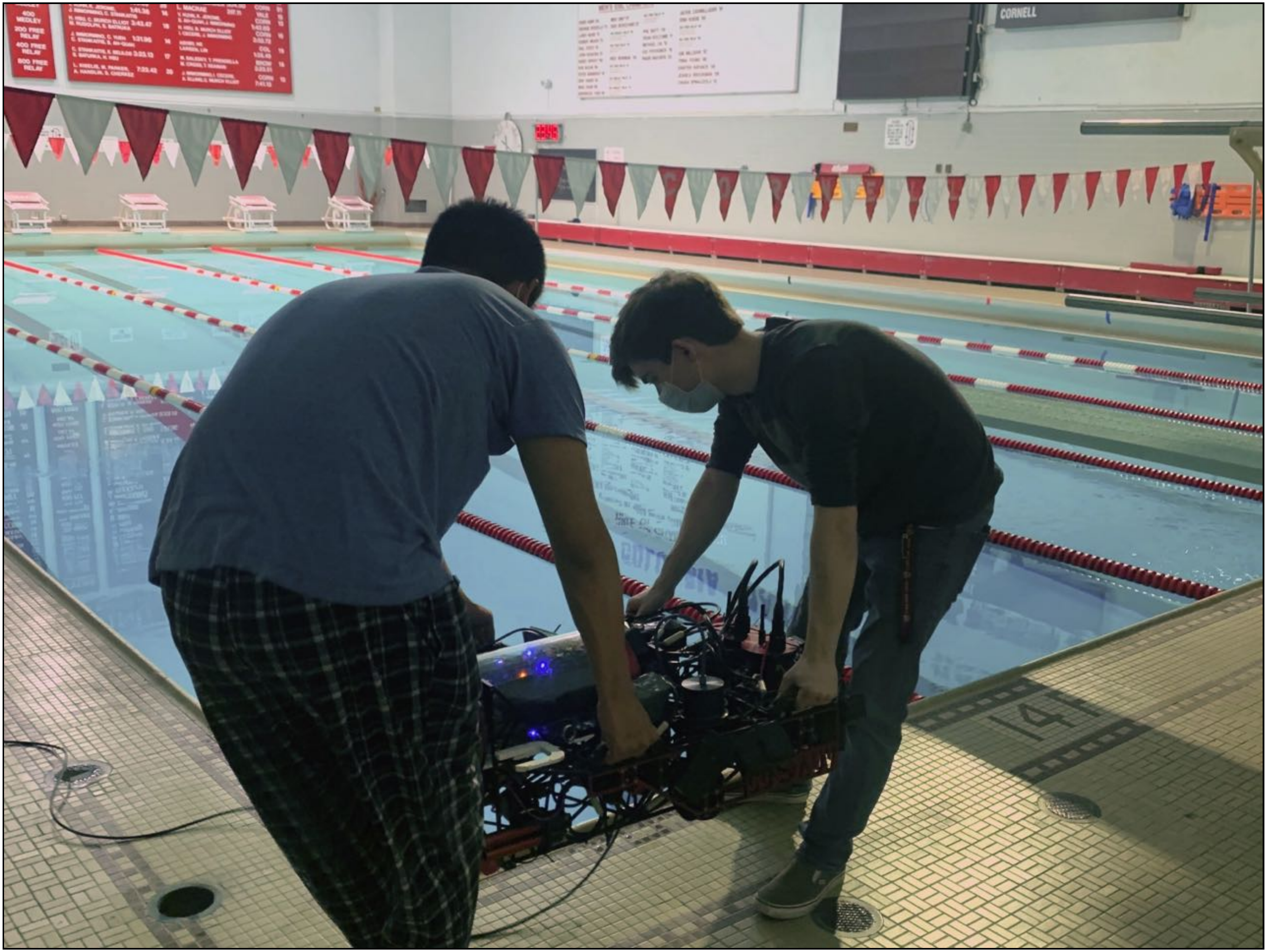
It is not possible for underclassmen to build custom computer chips
Must wait until senior or graduate level classes

Cornell Project Teams



1400 students on 30 teams
14 engineering majors




400 MEDLEY
200 FREE RELAY
400 FREE RELAY
800 FREE RELAY
CORNELL
14

Call for new engineering project team proposals — Exchange

Associate Dean for Undergraduate Programs
Call for new engineering project team proposals
To: ENGRFACULTY-L, ENGRFACULTYINCSANDBEE-L, ENGRACADEMICS-L, Cc: Lauren Stulgis,
Reply-To: Associate Dean for Undergraduate Programs

Archive - Exchange February 22, 2022 at 12:08 PM
Details

Colleagues,

Through a generous donor gift creating the **Shen Fund for Social Impact** we have the opportunity to fund multiple new engineering project teams. This program is designed to bring together new student teams under a faculty member's mentorship to address significant social challenges through novel and/or advanced engineering solutions. Falling under the Project Team Umbrella, the program will fund up to three new teams per year, with each supported for a three-year period at $30K/yr. The teams will also be provided space and support to design and implement these projects.

Proposals may be submitted by either faculty looking to guide a group of students, or by students who will engage with a faculty member to form the teams.

Attached to this e-mail are three documents:

- **Shen Fund FAQ Sp22.pdf:** More fully describes the nature of the projects and the goals of the program (also copied to the e-mail below).
- **Shen Fund Proposal Template Sp22.docx:** Short project proposal form.
- **Shen Funded Projects Summary_Sp22.pdf:** A summary document of a currently funded teams.

The ideal project will likely develop through discussions with Lauren Stulgis (as director of the project teams) and me. Feel free to reach out to us with rough ideas and concepts and we can help to try to develop a viable proposal.

Proposals will be considered as they arrive, with discussions to strengthen each within the program constraints. The initial application is a simple document identifying the primary goals, technical challenges and plans, timeline and budget, and currently engaged personnel.

Proposals must be uploaded directly to Box by email to: Proposa.zeuyhp9wqg5p8teo@u.box.com. The first round of decisions will be made based on submissions received by **11:59pm on Sunday, March 13, 2022.**

Again, please feel free to contact me or Lauren Stulgis with any questions or to discuss potential projects.

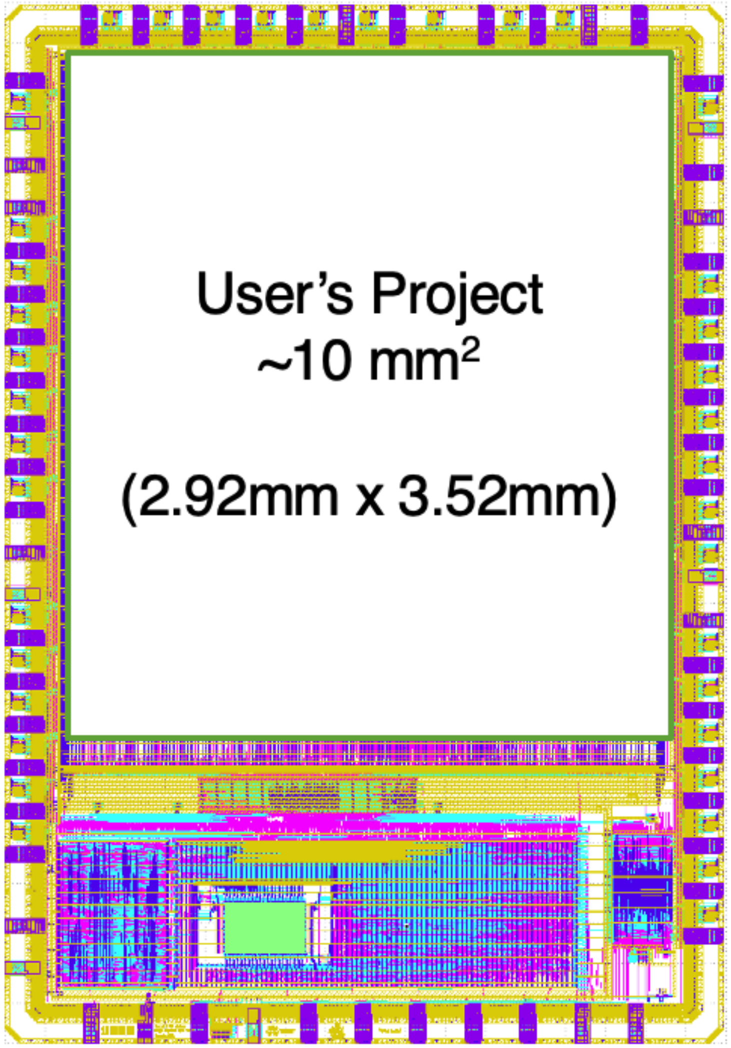
Prof. Alan Zehnder
Associate Dean for Undergraduate Programs
177 Rhodes Hall
Phone: (607) 255-9181
email: eng_ugdean@cornell.edu

C2S2: Cornell Custom Silicon Systems Project Team

Three-year student-led project team to tapeout multiple custom chips in SkyWater 130nm to implement a proof-of-concept system for a campus partner

- Open RISC-V instruction set
- Open-Source VexRISCV μcontoller
- Open-Source OpenROAD chip flow
- Open PDK for SkyWater 130nm
- OpenMPW + ChipIgnite w/ efabless

- Custom system-on-chip
- Custom evaluation board
- Custom software stack



C2S2 Recruiting

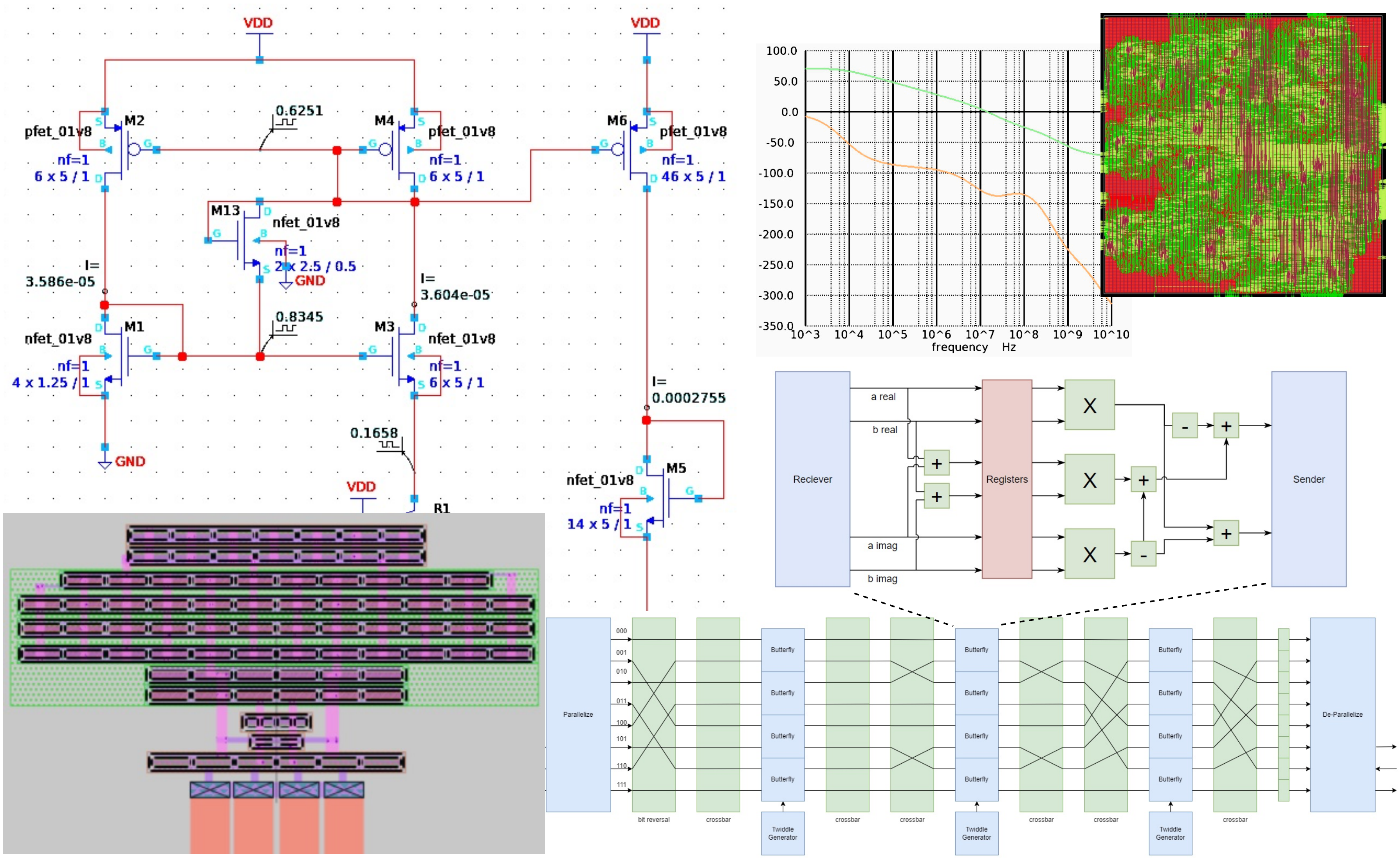
- 60 students attended virtual summer information session
- 40 students applied to be on the leadership team
- 7 students selected to lead the team including sophomores, juniors, seniors from ECE, CS, and Engineering Management
- Student-led recruiting in early fall with over 100 students applying to be on the team
- 25 students selected in fall with five students joining in spring

C2S2 Team

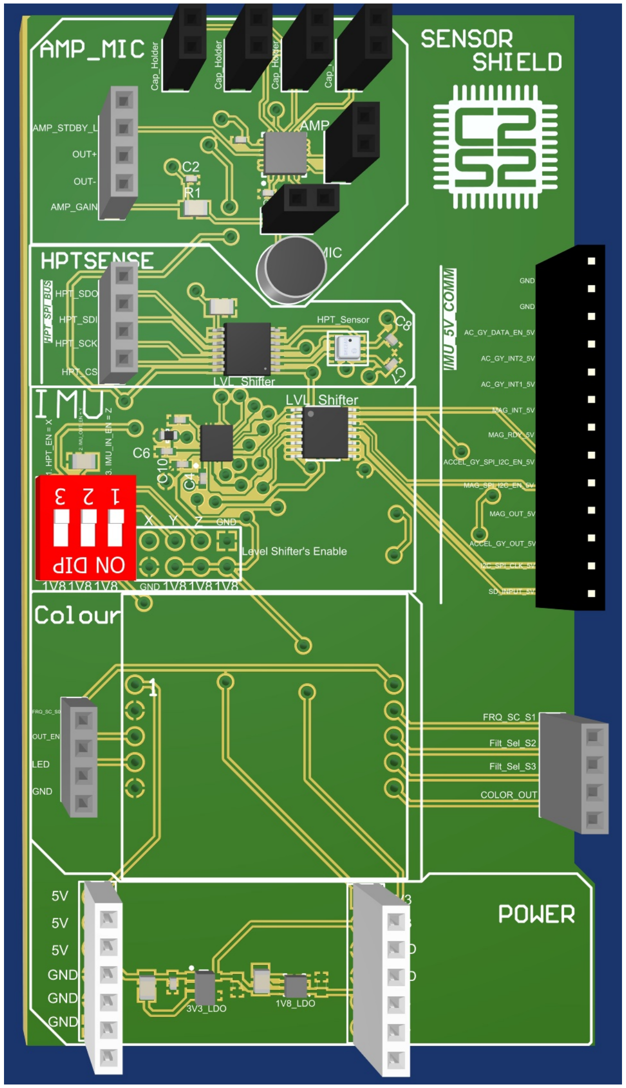
- Students self-organized into five subteams
  - Digital design subteam
  - Analog design subteam
  - Software subteam
  - System architecture subteam
  - Project management subteam

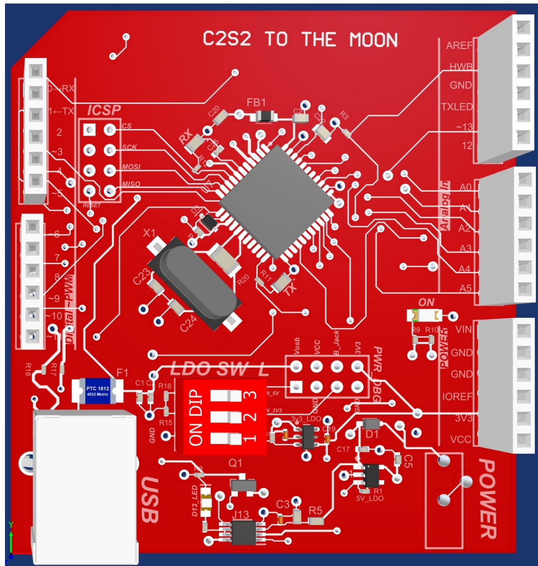
- Analog subteam was completely driven by student interest
  - Reached out to domain experts at Cornell and industry to help
  - Actively recruiting fellow students interested in analog design
  - Very few analog tapeouts on SkyWater 130nm using open-source EDA tools, pushing the frontier of tooling!
  - Planning on *both* an analog *and* digital tapeout

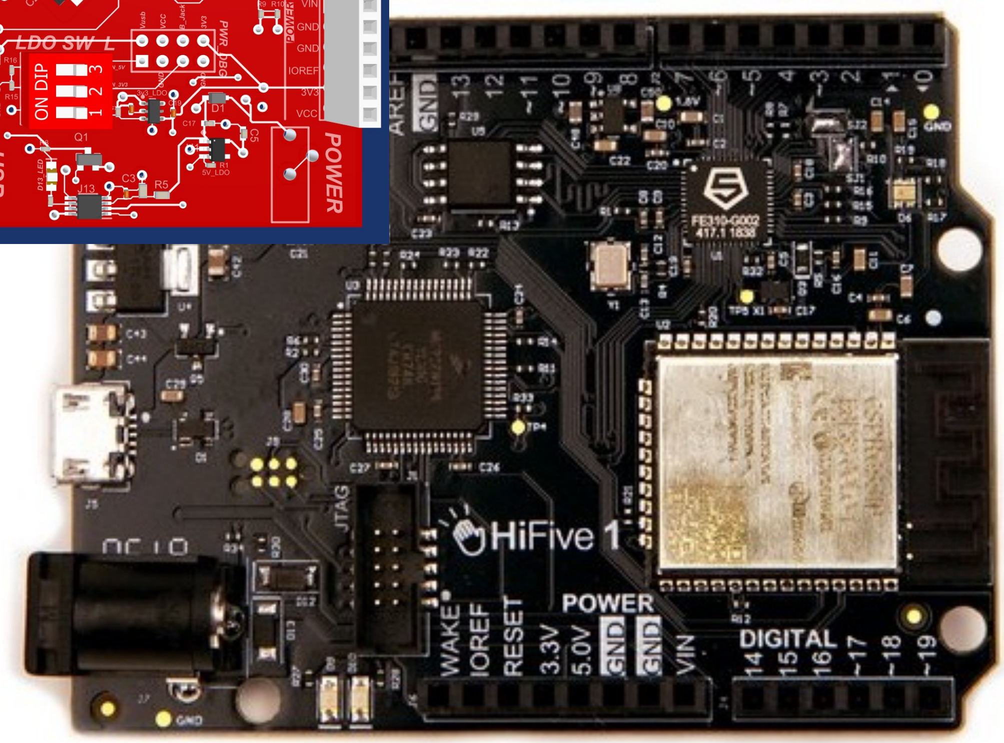
C2S2 Analog and Digital Subteams



C2S2 Software and System Architecture Subteams

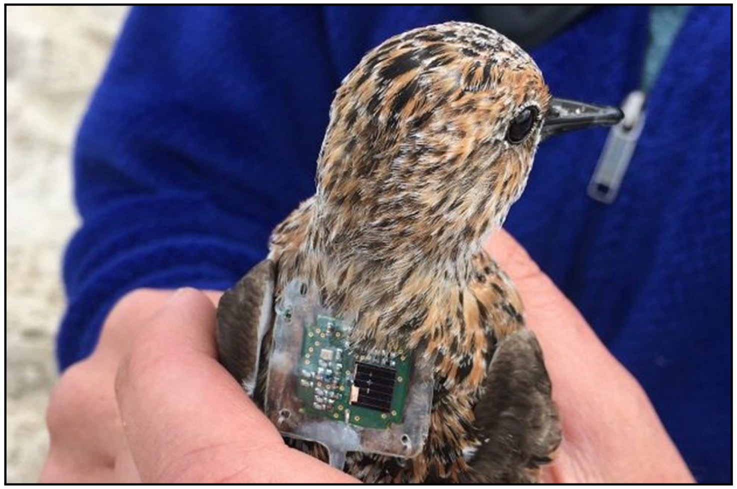


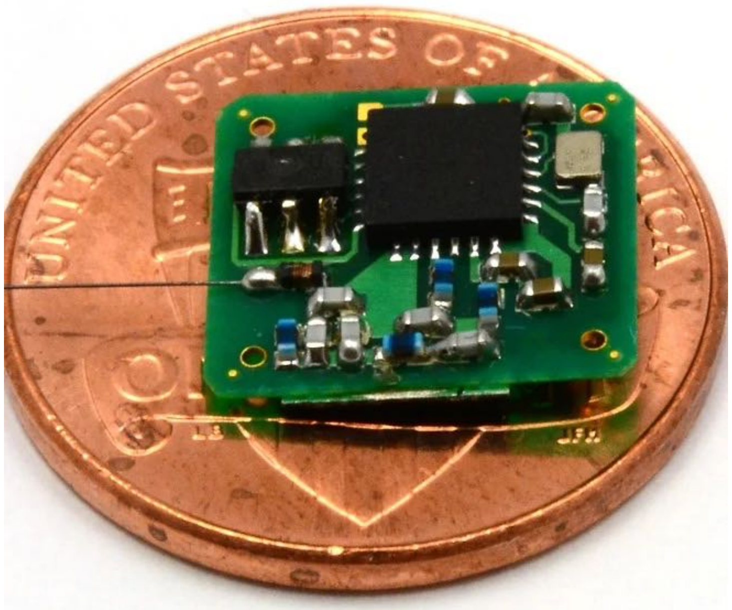




C2S2 Project Management Subteam

- Reached out to over 30 instructors, researchers, and faculty across Cornell to find a campus partner interested in working with the team
- Campus partner will provide problem specification that can potentially be addressed through a custom system-on-chip
- Team has decided to work with the Cornell Lab of Ornithology to explore meeting unique weight and power constraints of on-bird tracking systems

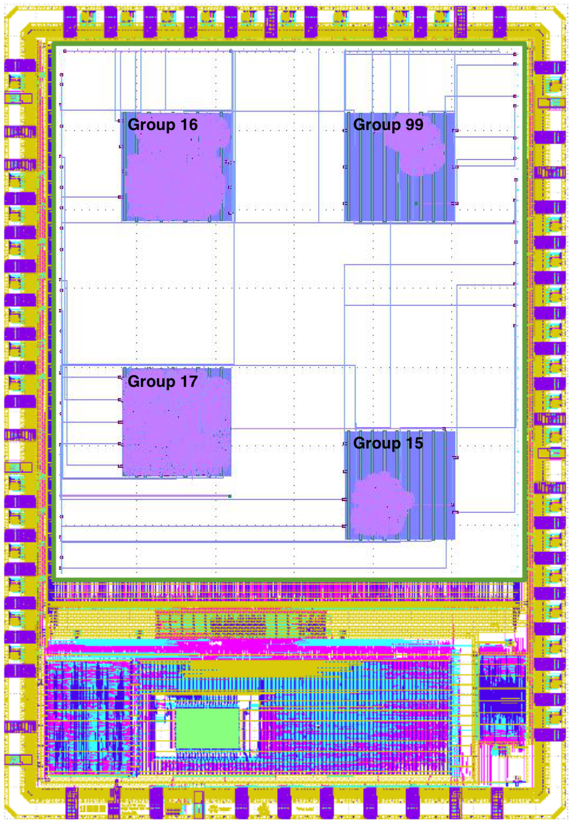




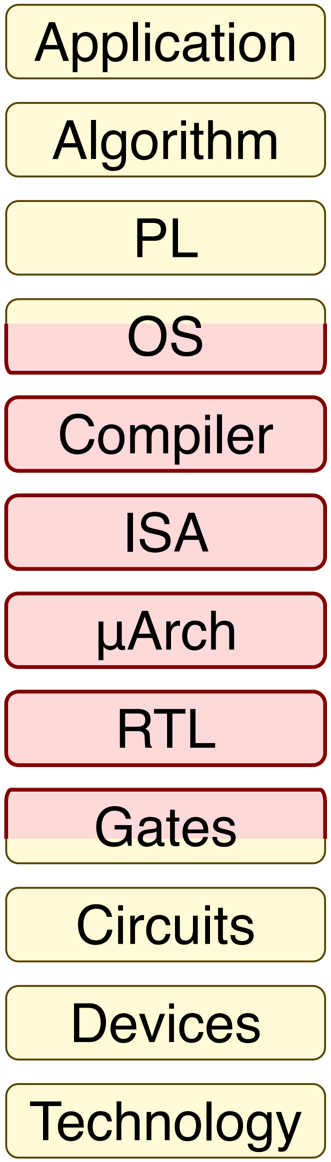
C2S2 Community

- Coding sprints and hack-a-thons
- Organized C2S2 swag including T-shits, sweatshirts, stickers
- Social events like bowling and movie night
- Recruiting with companies
- Day trip to the SUNY Poly Nanofabrication facility

Cornell Custom Silicon Systems Project Team



The C2S2 project team is unique across the country!
Email c2s2-leaders-l@cornell.edu for more information



Take-Away Points

- Computer engineering is the development of the abstraction/implementation layers that enable information processing applications to efficiently use available manufacturing technologies
- We are entering an exciting new specialization era which offers tremendous challenges and opportunities, which makes it a wonderful time to study and contribute to the field of computer engineering
- The Cornell Custom Silicon Systems project team is a unique way for underclassman to get practical hands-on experience across the entire computer systems stack (from apps to chips!)