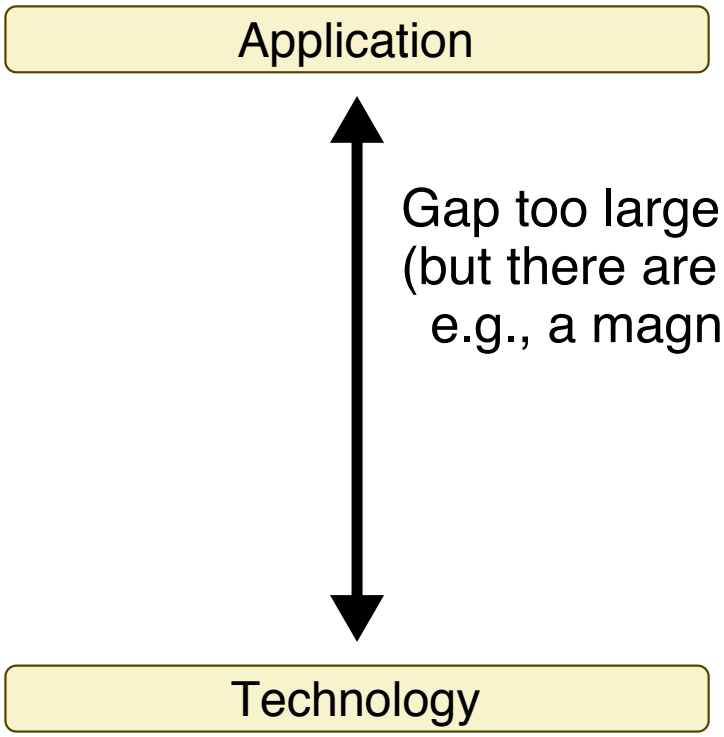
ENGRI 1210
Recent Trends in Computer Engineering

Christopher Batten

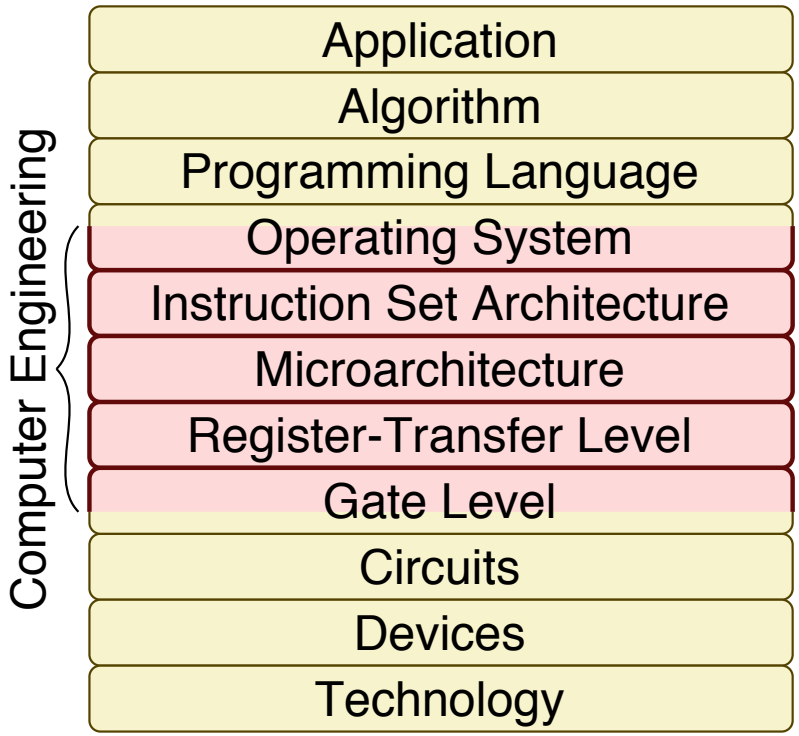
School of Electrical and Computer Engineering
Cornell University

The Computer Systems Stack



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

The Computer Systems Stack



Sort an array of numbers

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

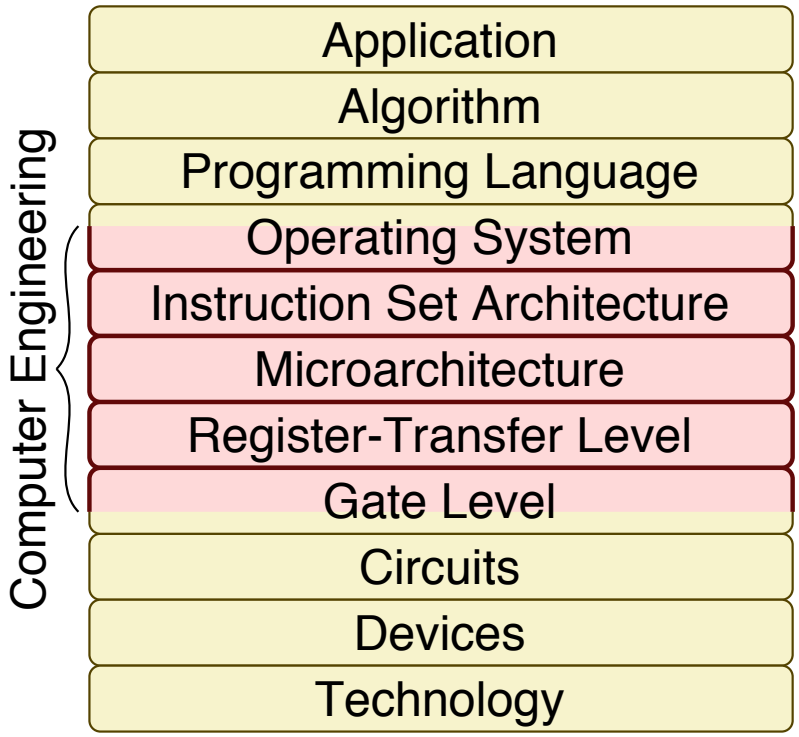
Out-of-place selection sort algorithm

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

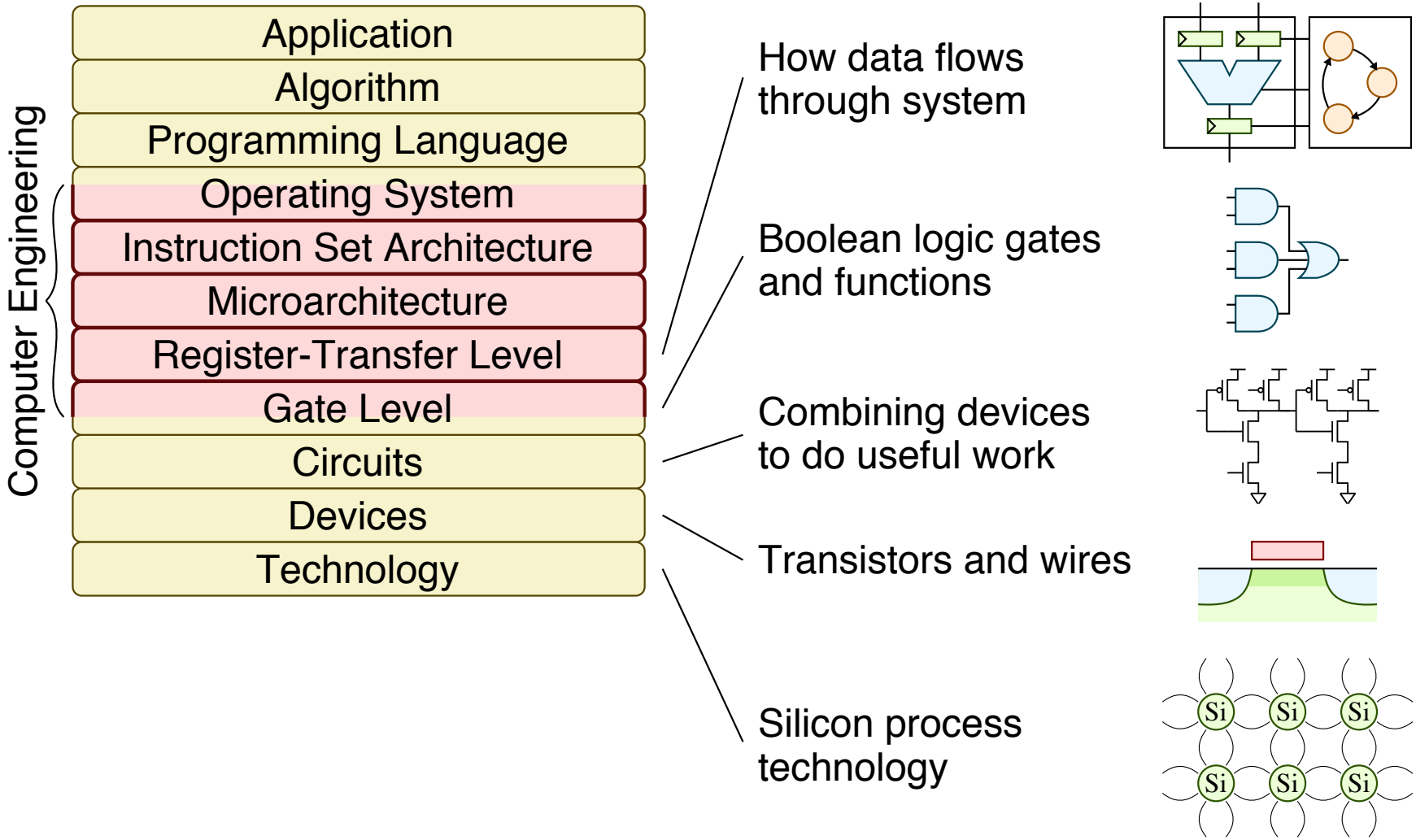


Mac OS X, Windows, Linux
Handles low-level hardware management

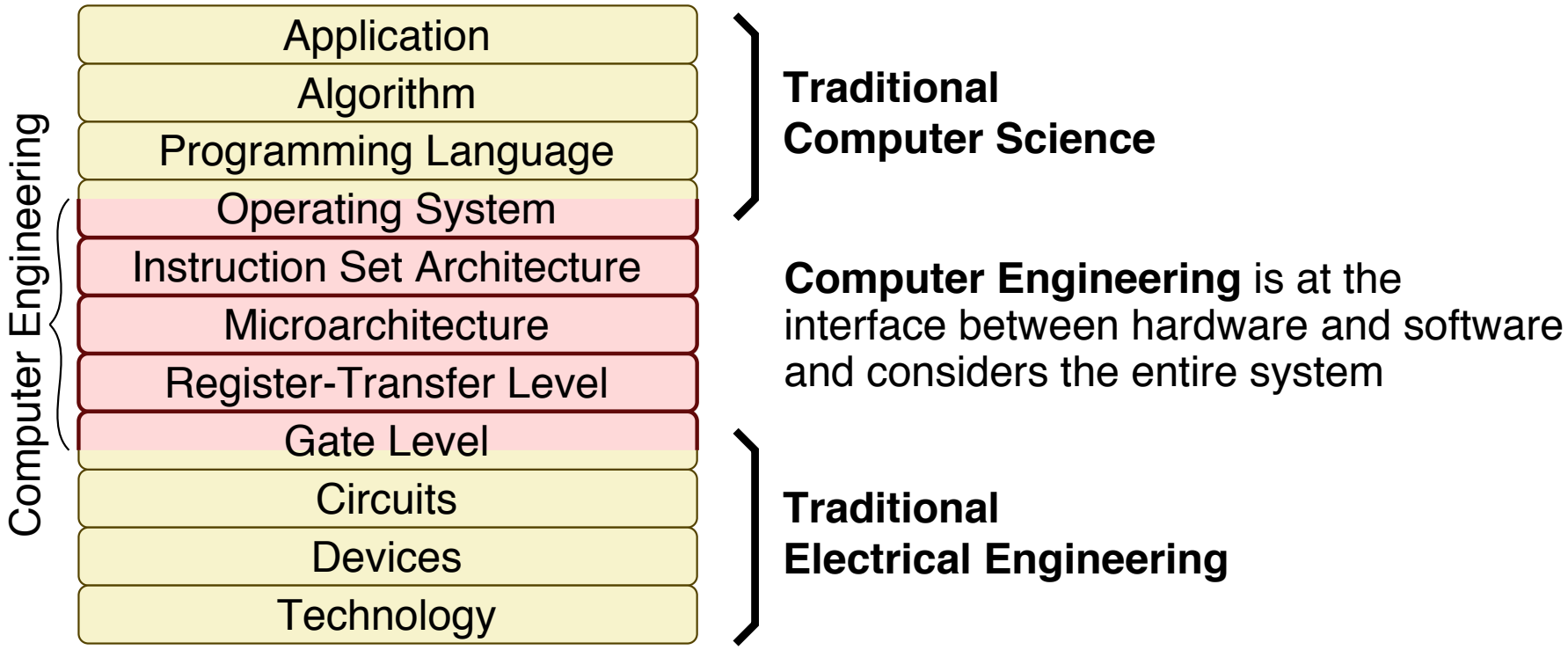
MIPS32 Instruction Set
Instructions that machine executes

```
blez  $a2, done
move  $a7, $zero
li    $t4, 99
move  $a4, $a1
move  $v1, $zero
li    $a3, 99
lw    $a5, 0($a4)
addiu $a4, $a4, 4
slt   $a6, $a5, $a3
movn  $v0, $v1, $a6
addiu $v1, $v1, 1
movn  $a3, $a5, $a6
```

The Computer Systems Stack

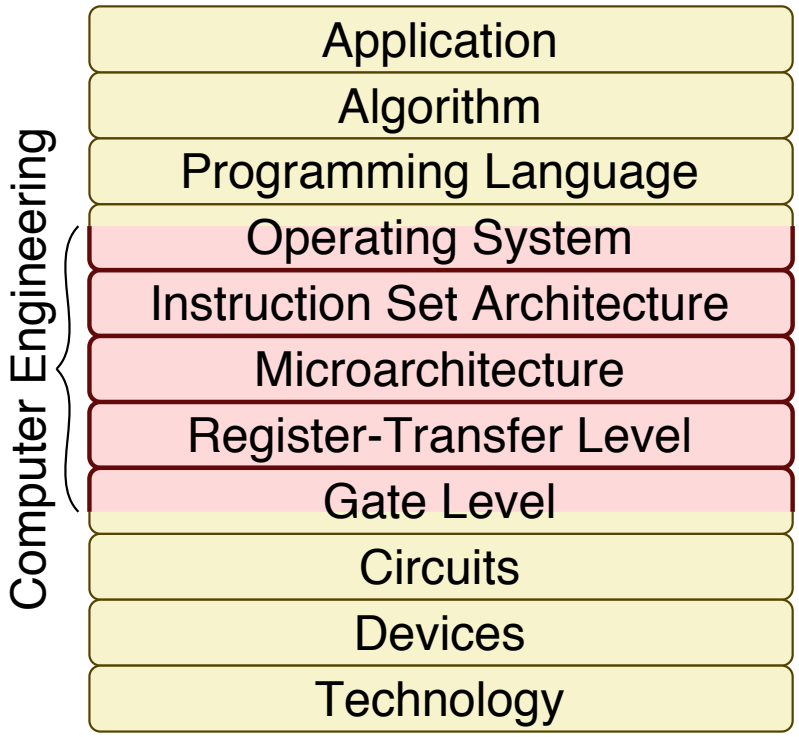


Electrical Engr vs. Comp Sci vs. Comp Engr



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

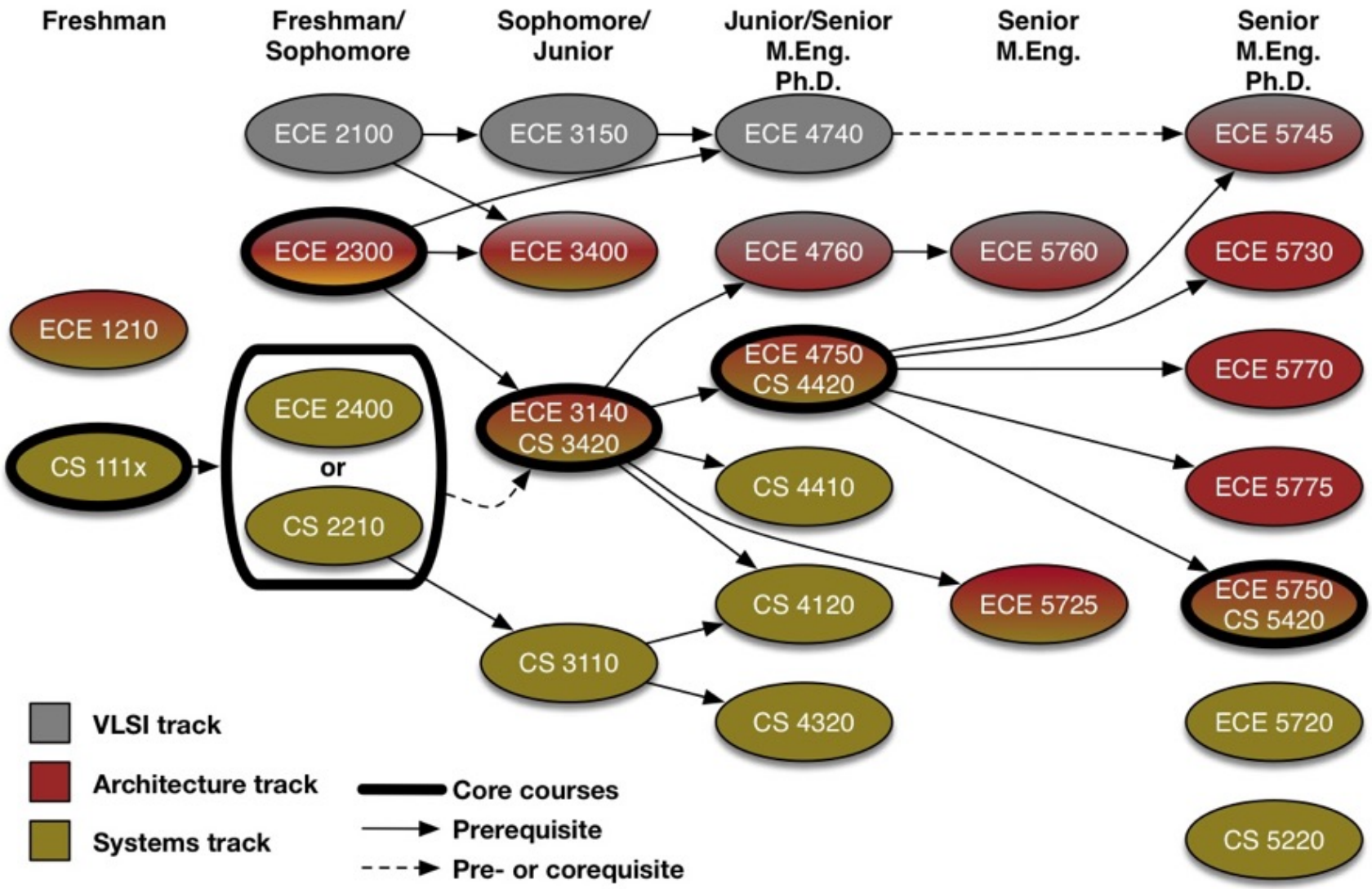


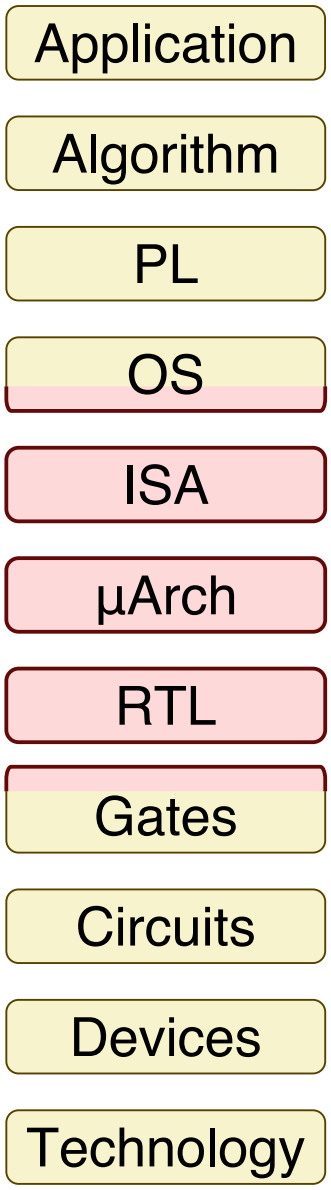
ECE 2400 Computer Systems Programming

- ECE 3140 Embedded Systems
  - ECE 4760 Design with Microcontrollers
  - ECE 4750 Computer Architecture

ECE 2300 Digital Logic & Computer Org

Cornell Computer Engineering Curriculum





Agenda

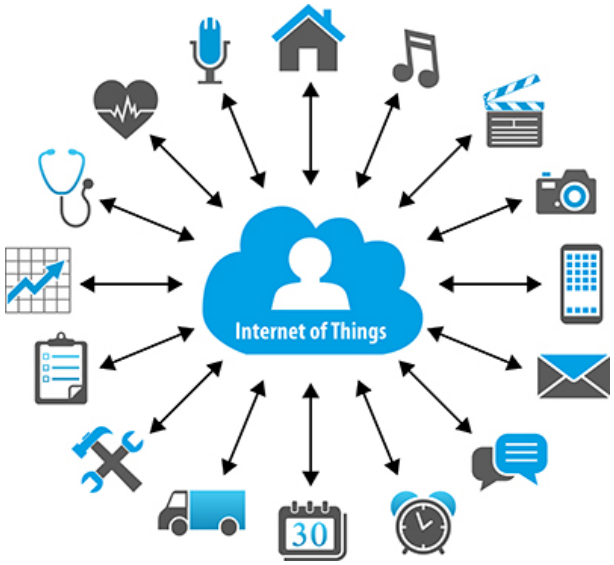
The Computer Systems Stack

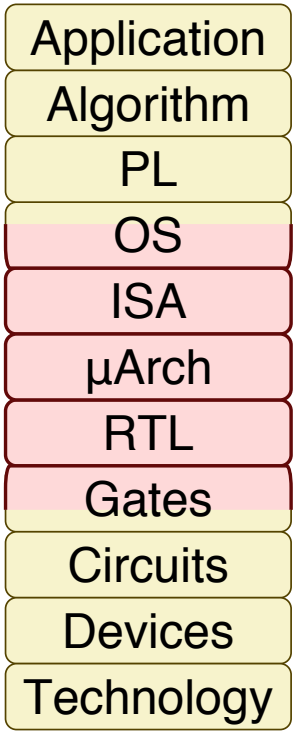
Trends in Computer Engineering

Hardware Acceleration for Deep Learning

Three Key Trends in Computer Engineering

Trend #1: Growing Diversity in Applications and Systems





Trend #2: Software/Arch Interface Changing Radically

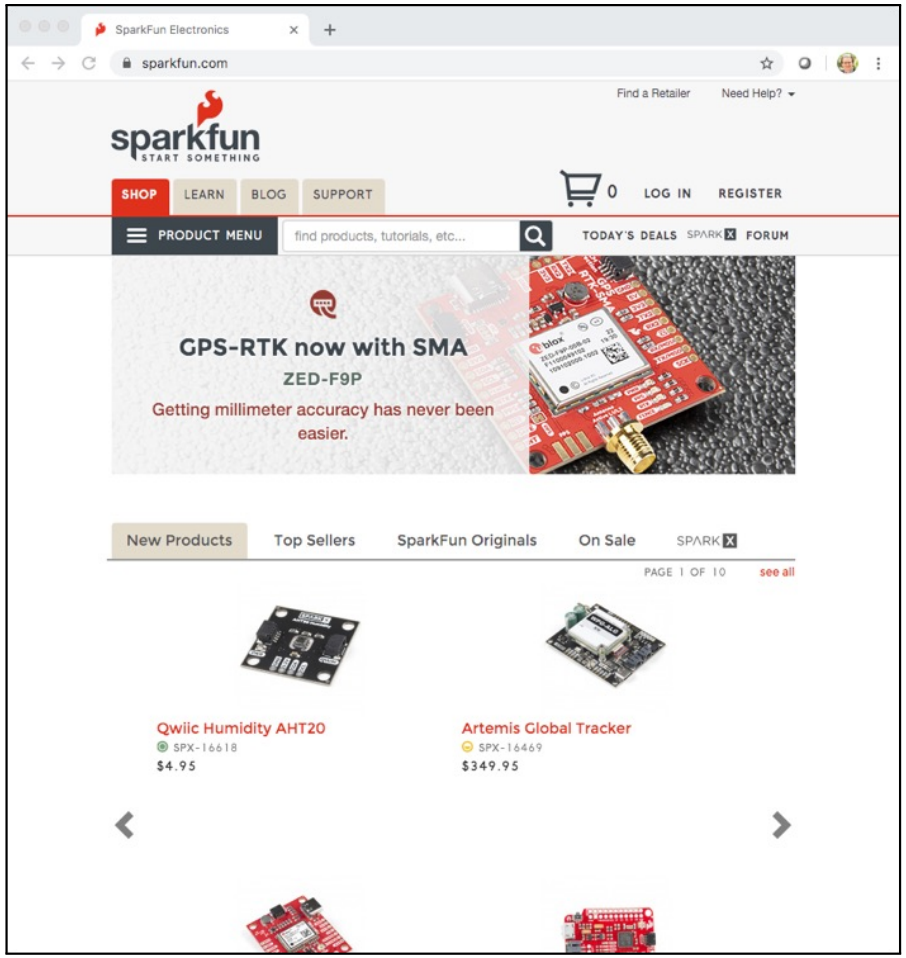
Trend #3: Technology/Arch Interface Changing Radically

Students entering the field of computer engineering have a unique opportunity to shape the future of computing and how it will impact society

Activity #1: Size and Cost of Modern Computers

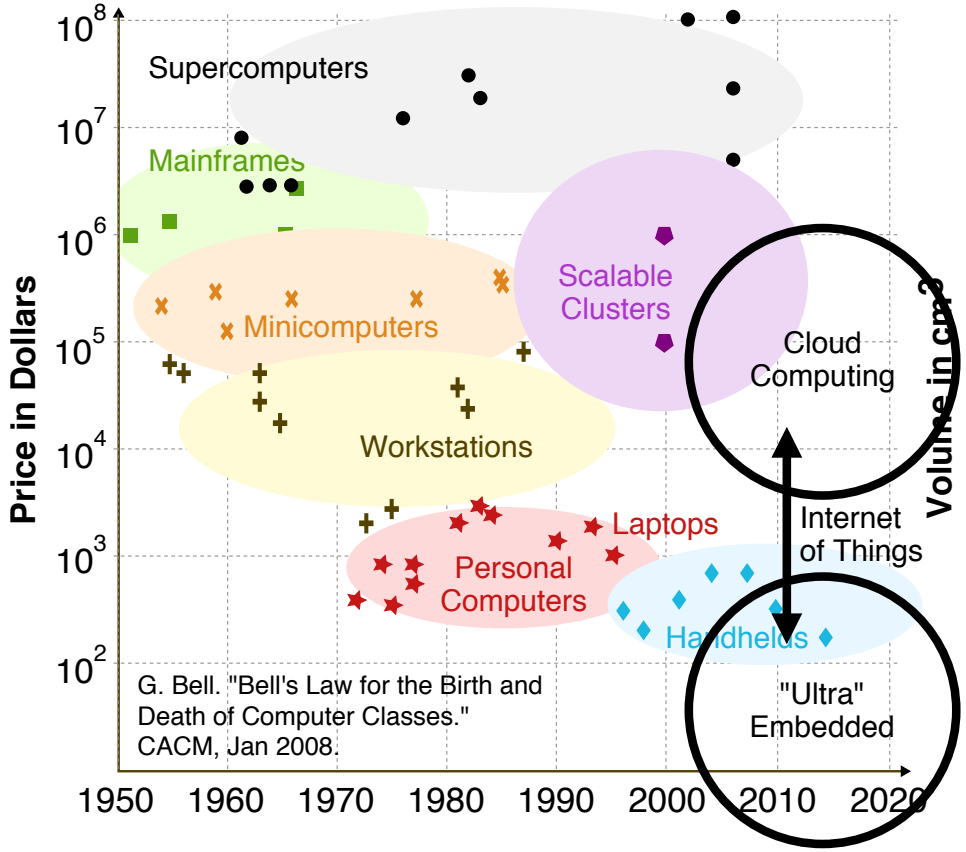
`http://tiny.cc/engri1210-1`

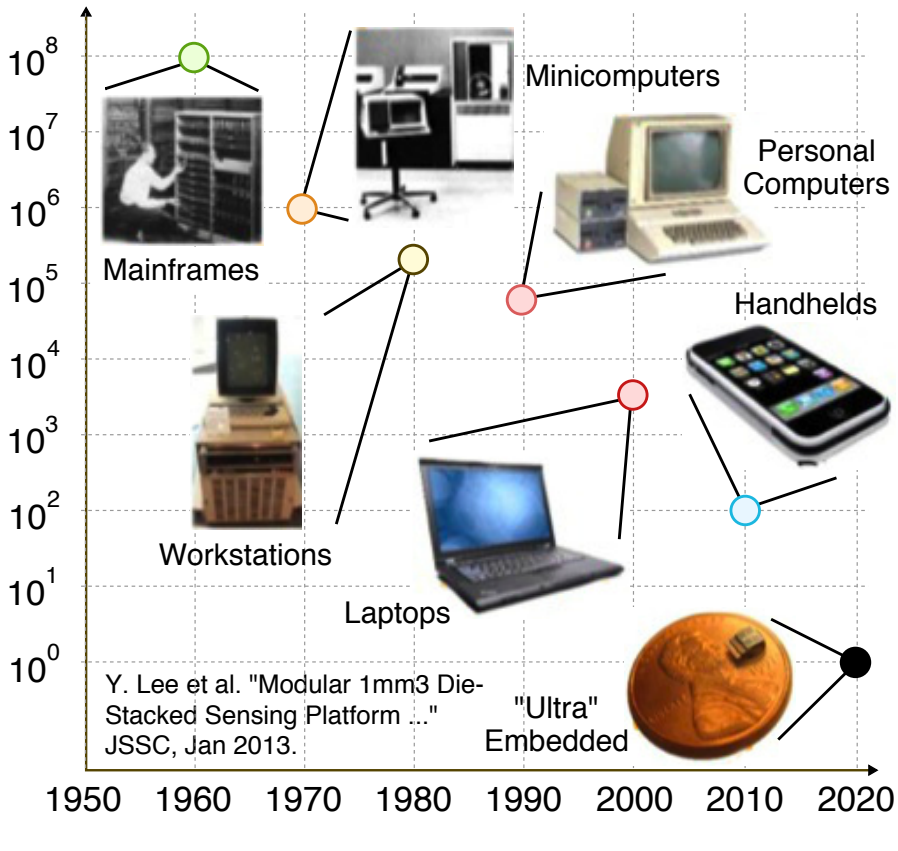
1. Breakout into groups of 3 students
2. Browse Digikey, Mouser, Sparkfun, Adafruit
3. Find a few “computers” (smaller is better)
4. Enter cost and estimated volume in Google form
5. Come back into main zoom room



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





Growing Diversity in Apps & Systems

Example: Internet of Things

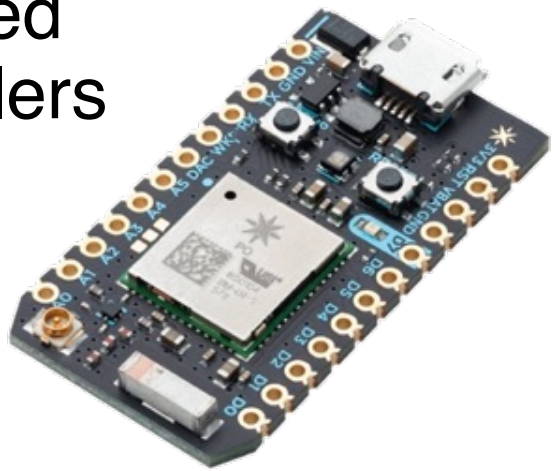
5.8 billion
Connected “things” in 2020
— Gartner

40 billion
Connected “things” in 2025
— IDC

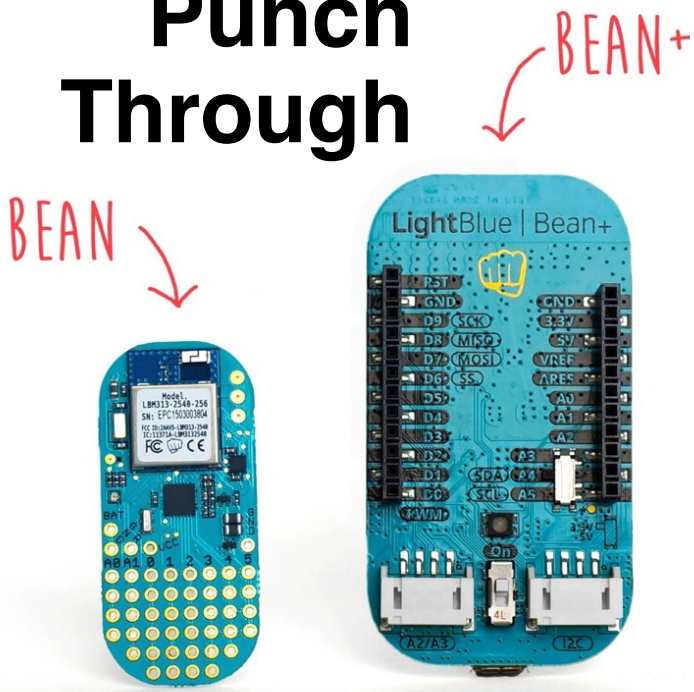
IoT Platform Startups

Particle: Photon

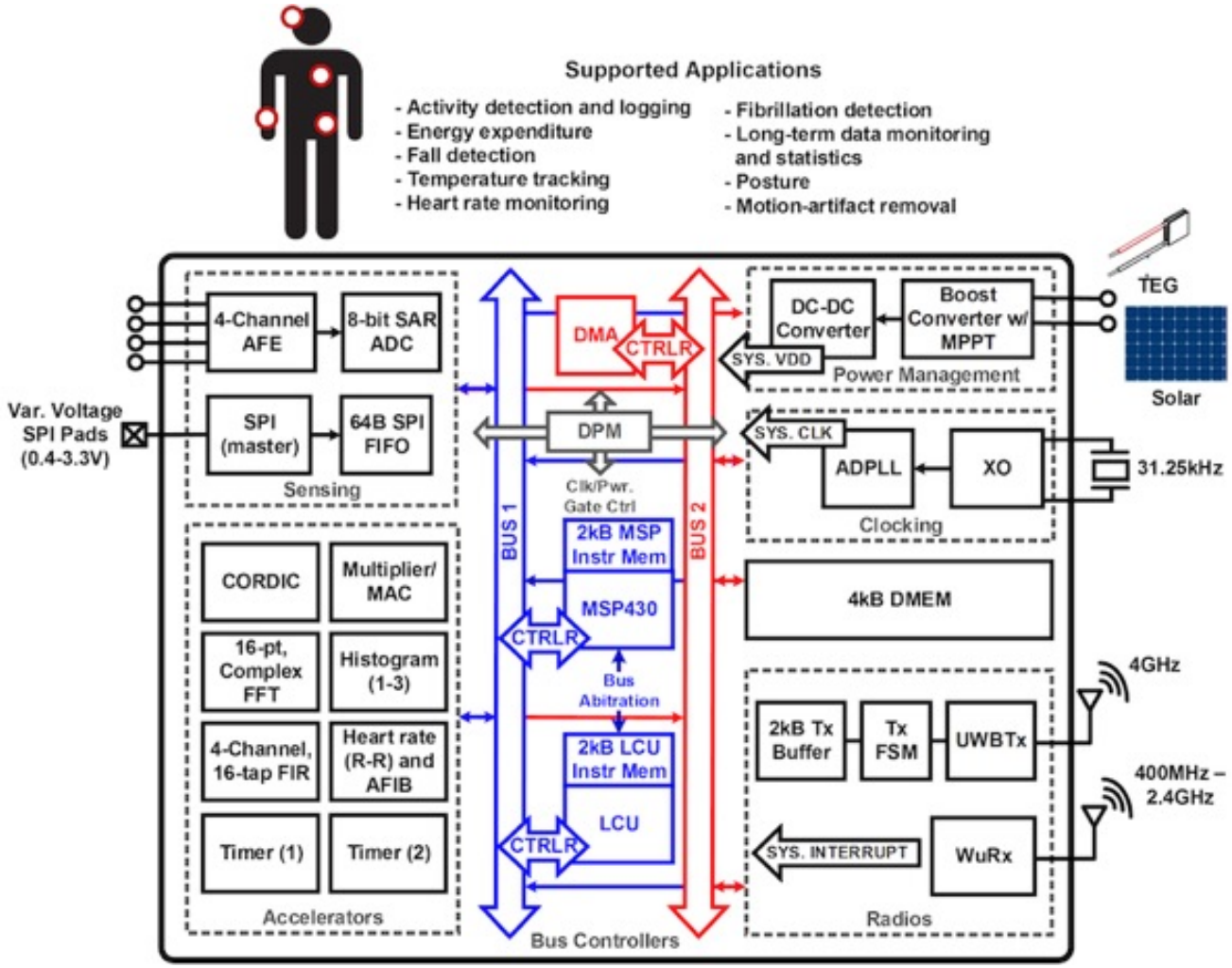
WiFi connected µcontrollers w/ Particle Cloud



Punch Through



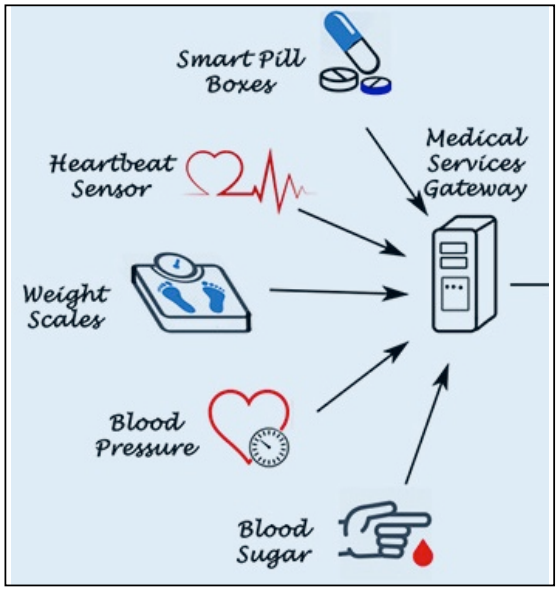
IoT Chip Startups

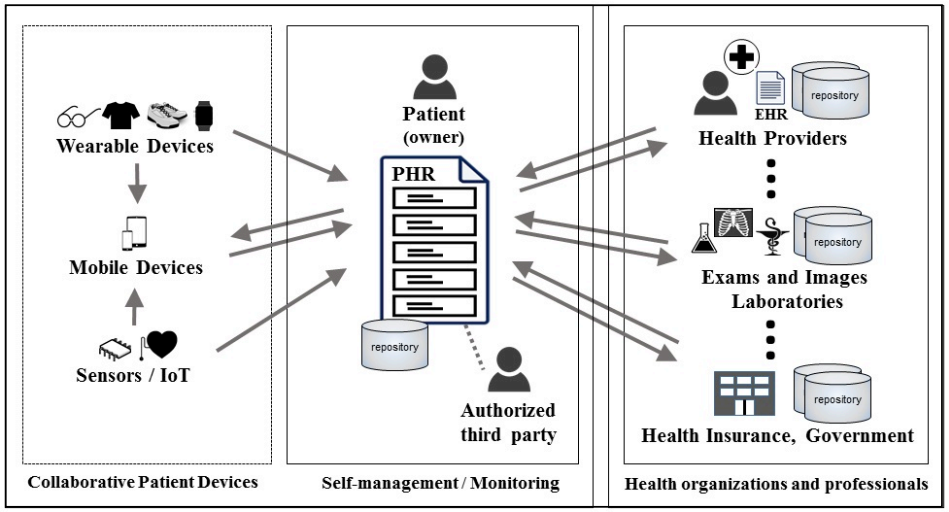


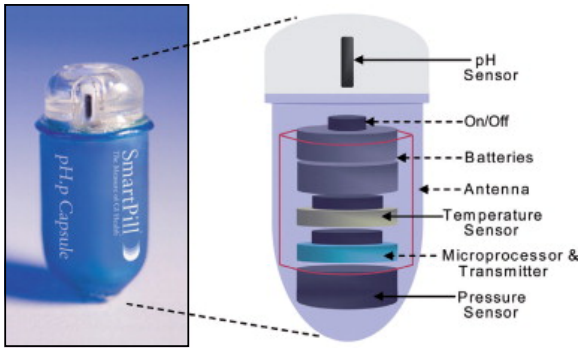
B. Calhoun, D. Wentzloff, et al.
Univ. of Virginia, Univ. of Michigan

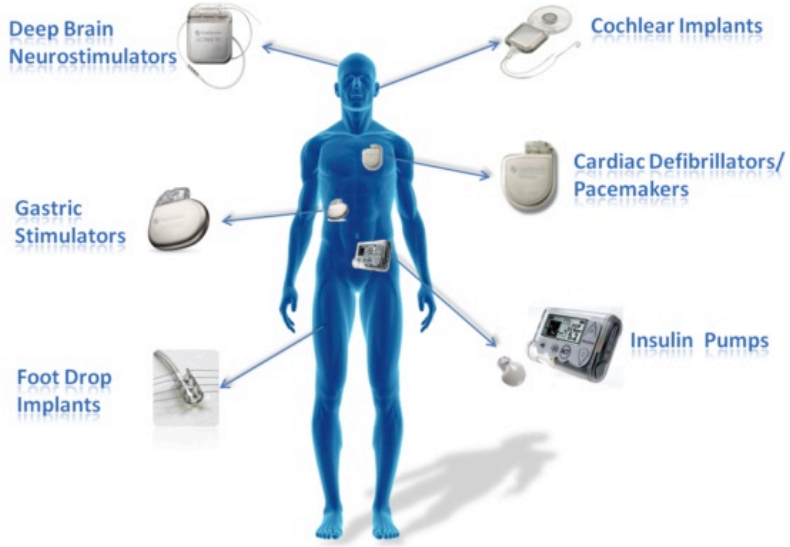
Chip startup founded in 2014 to use ultra-low-power circuits in energy harvesting IoT devices

IoT for Truly Personalized Medicine

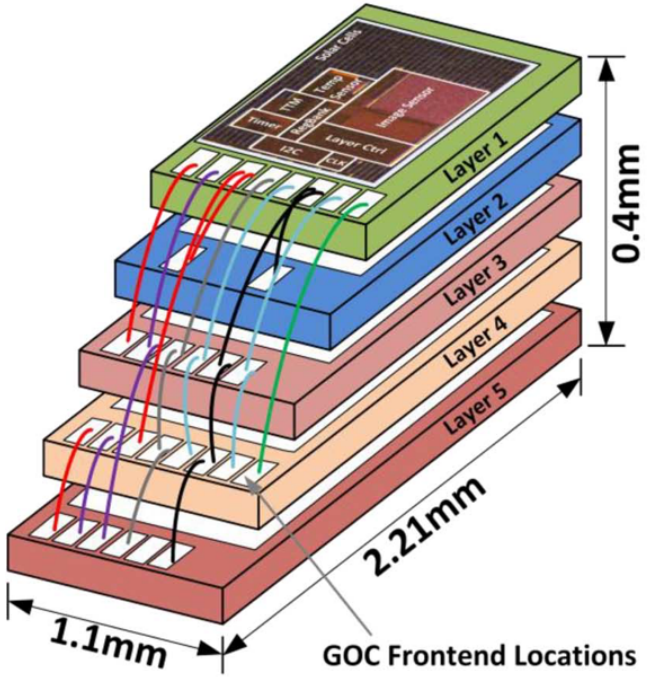


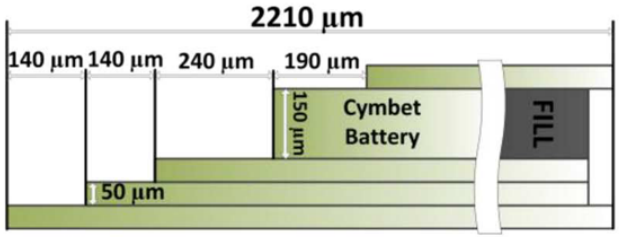


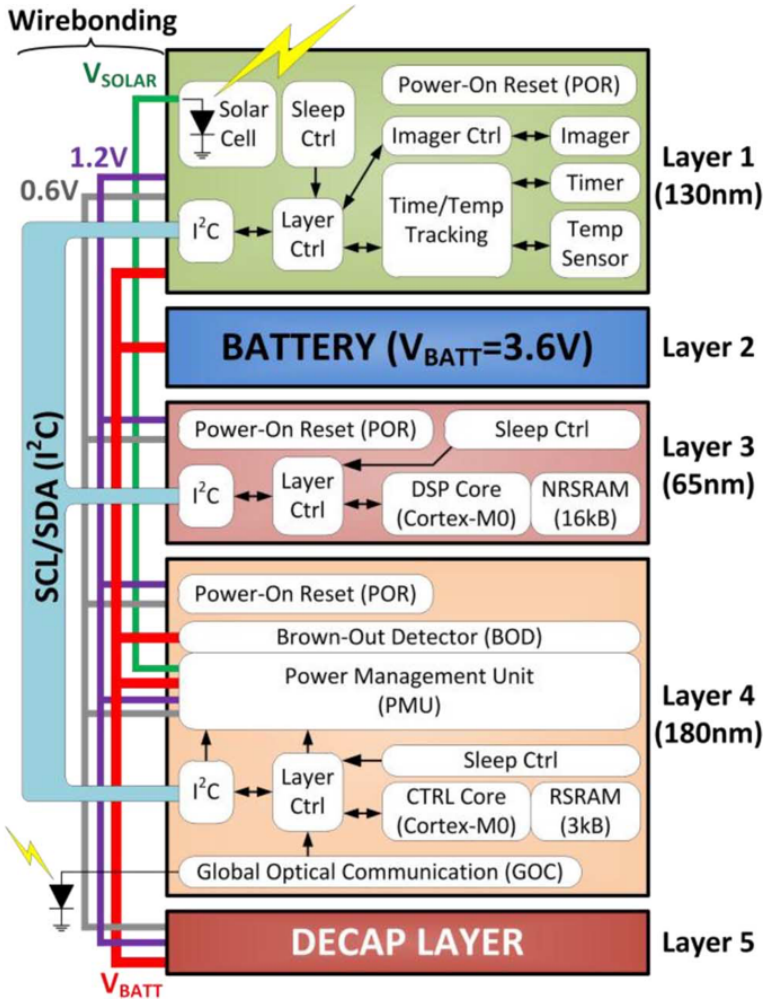




M3: Michigan Micro Mote



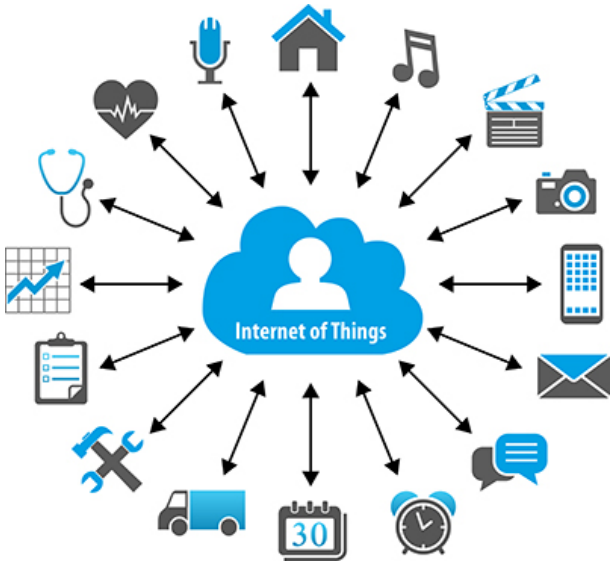


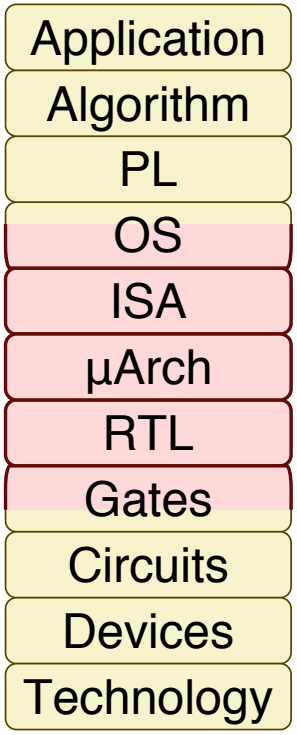


Adapted from Y. Lee et al., JSSC, 2013.

Three Key Trends in Computer Engineering

Trend #1: Growing Diversity in Applications and Systems





Trend #2: Software/Arch Interface Changing Radically

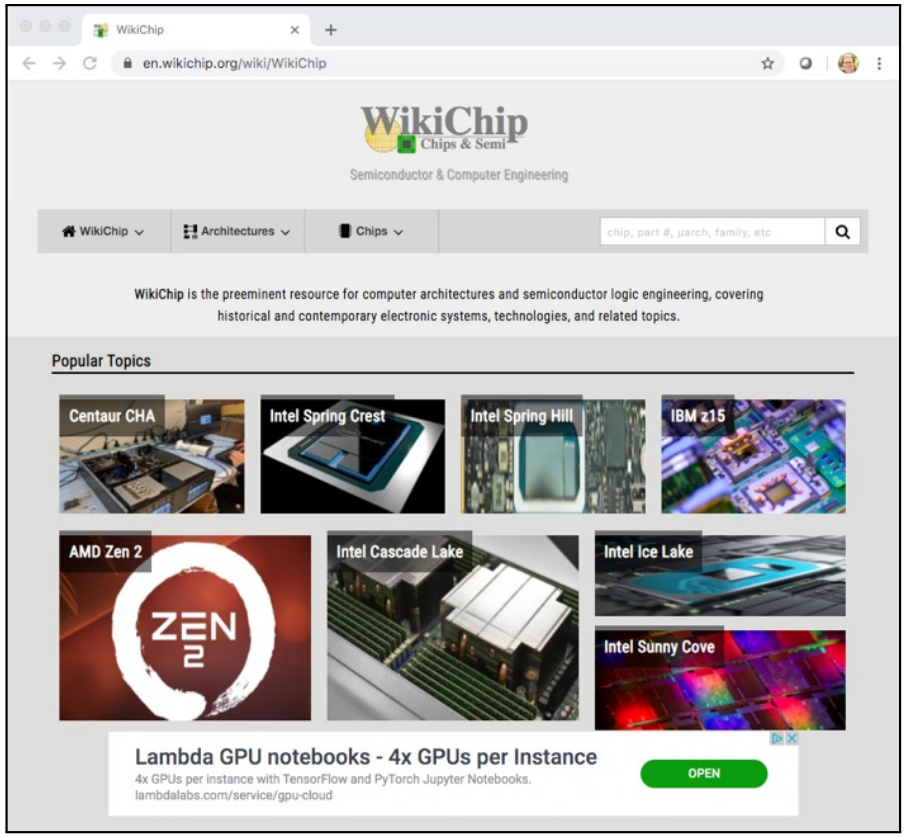
Trend #3: Technology/Arch Interface Changing Radically

Students entering the field of computer engineering have a unique opportunity to shape the future of computing and how it will impact society

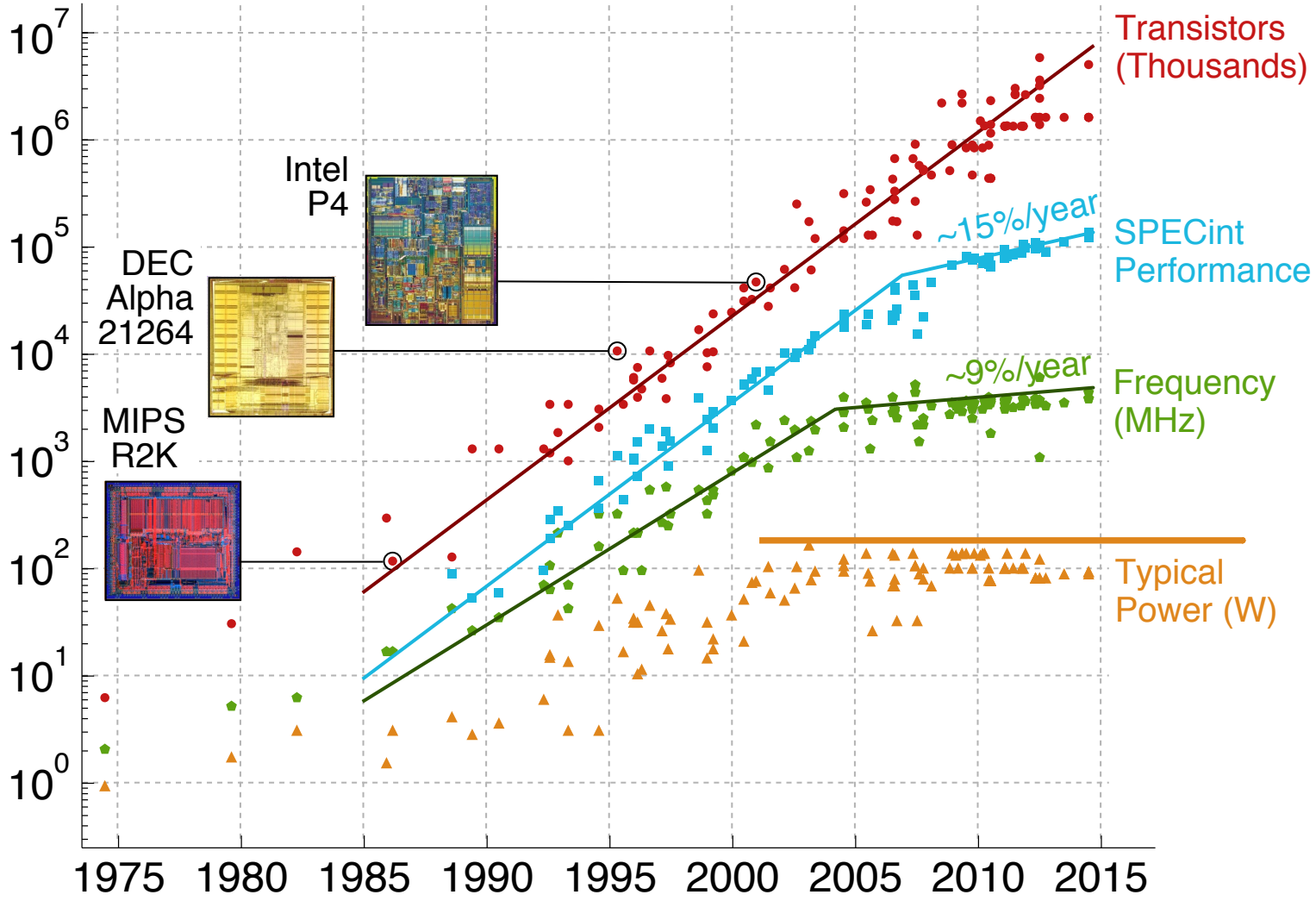
Activity #2: Specifications of Modern Processors

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

1. Breakout into groups of 3 students
2. Browse WikiChip
3. Find a few processors
4. Enter year, frequency, core count, power in Google form
5. Come back into main zoom room

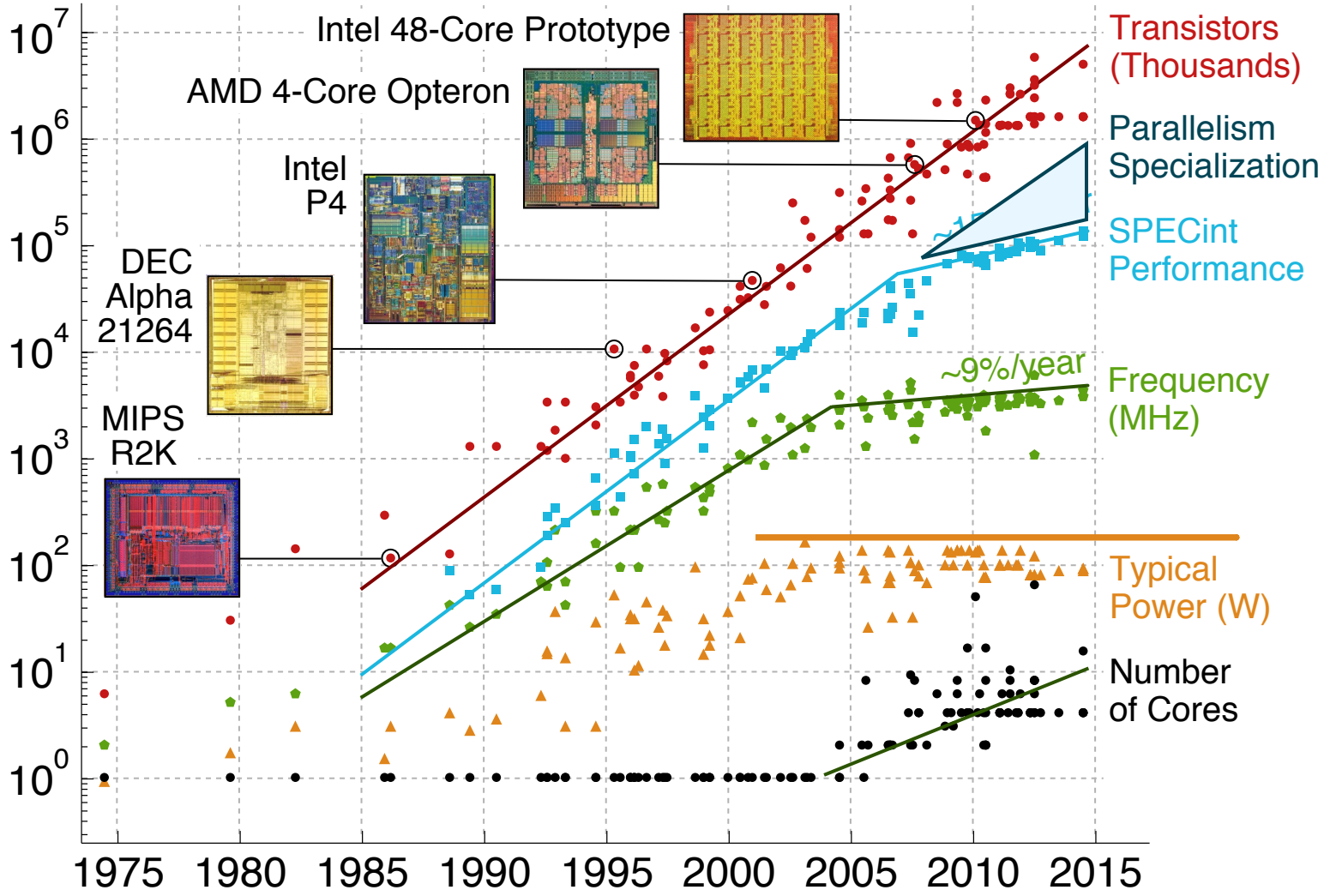


Trends in High-Performance Processors



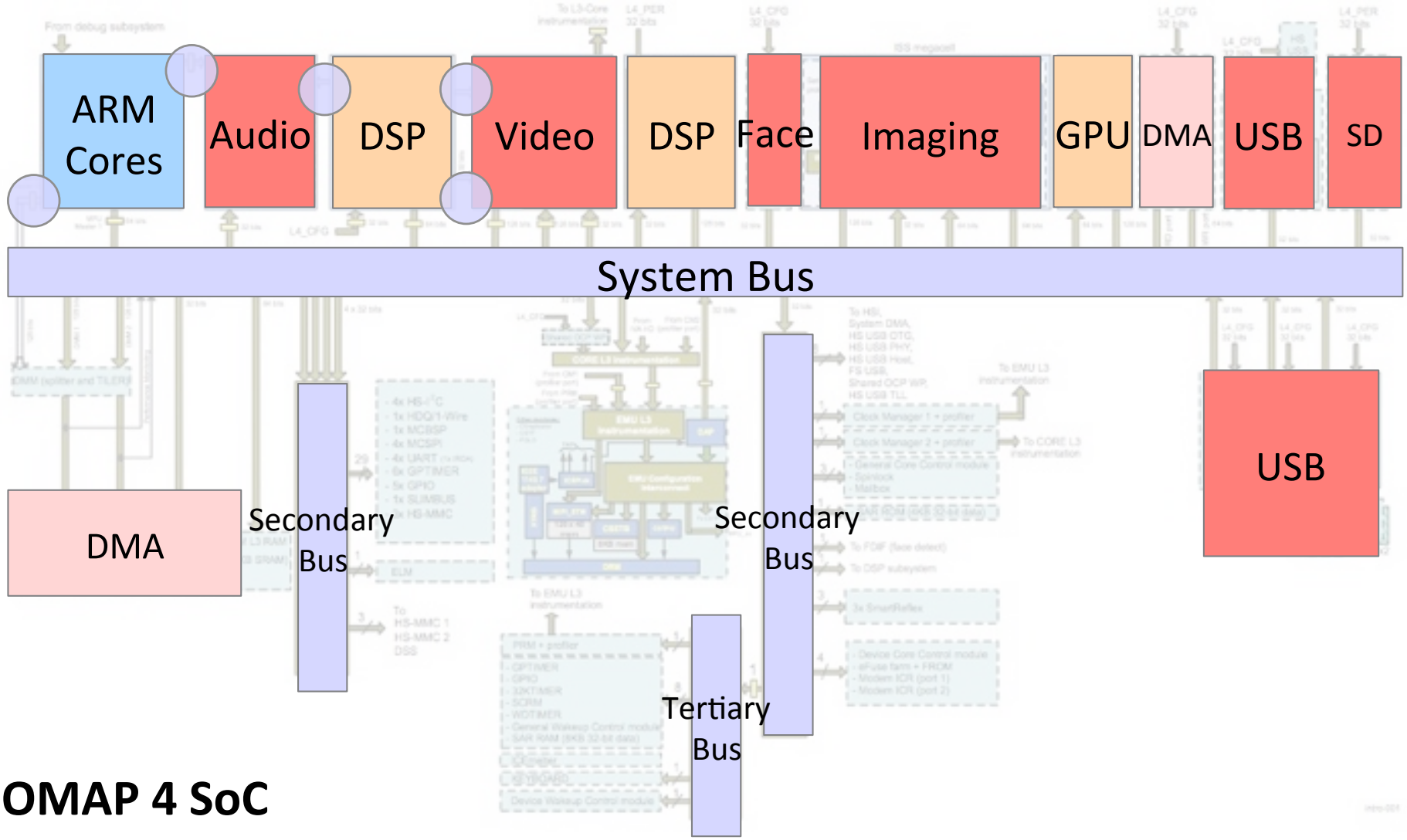
Data collected by M. Horowitz, F. Labonte, O. Shacham, K. Olukotun, L. Hammond, C. Batten

Parallelization & Specialization Are Now Critical



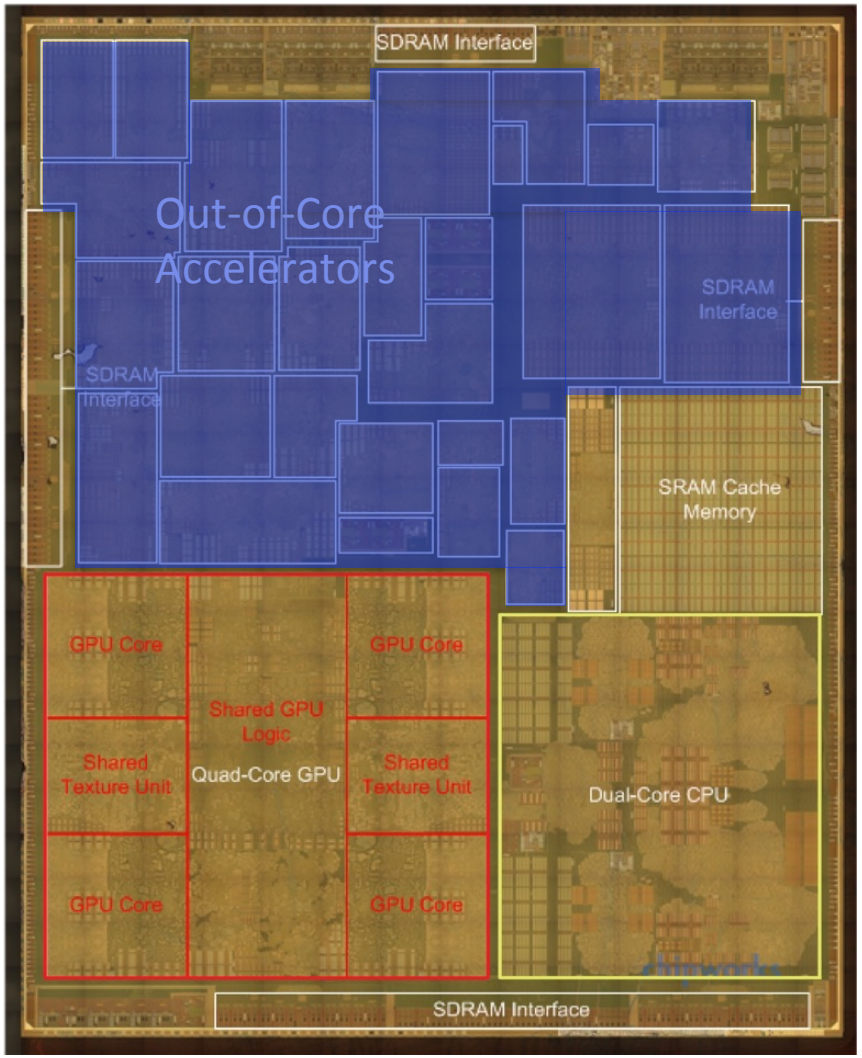
Data collected by M. Horowitz, F. Labonte, O. Shacham, K. Olukotun, L. Hammond, C. Batten

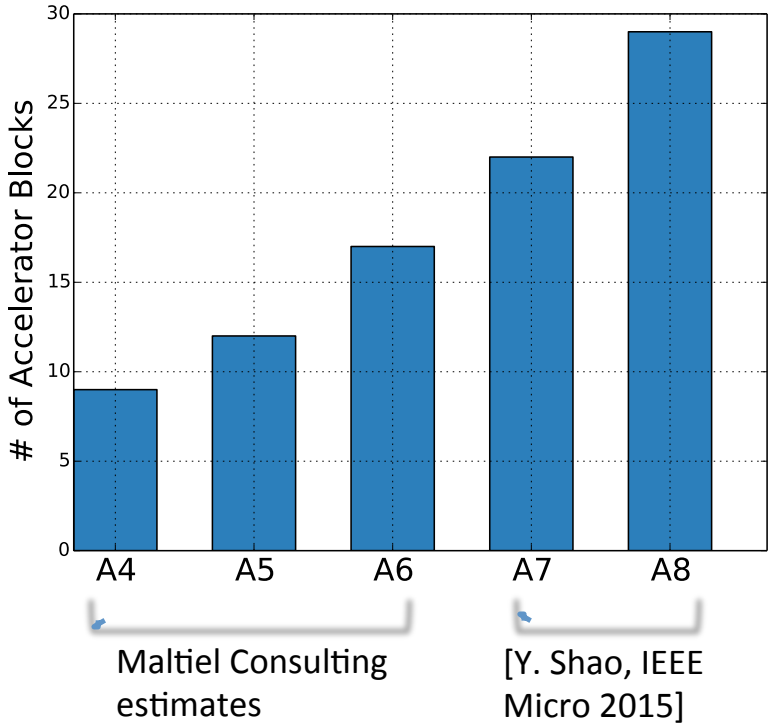
Heterogeneous Systems-on-Chip



Adapted from D. Brooks Keynote at NSF XPS Workshop, May 2015.

Heterogeneous Systems-on-Chip

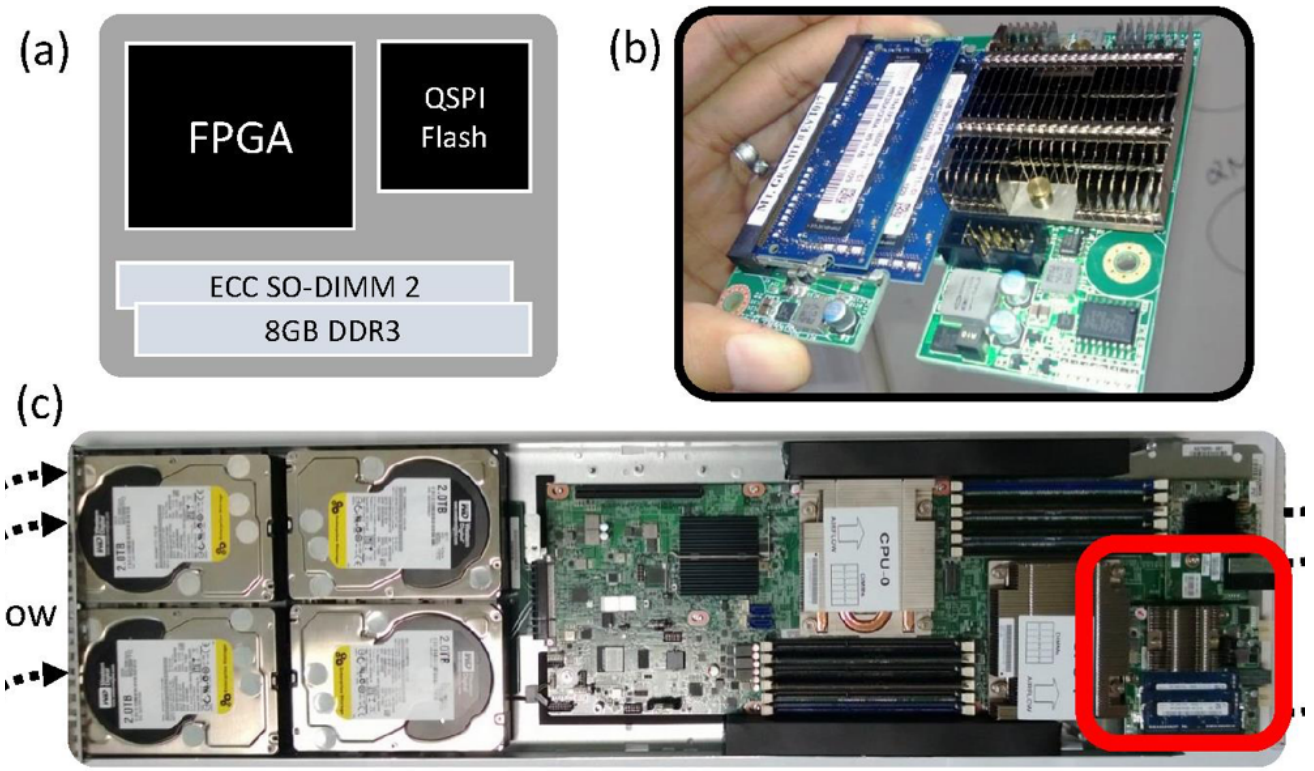




[www.anandtech.com/show/8562/chipworks-a8]

Adapted from D. Brooks Keynote at NSF XPS Workshop, May 2015.

Microsoft Catapult: FPGAs in the Data Center

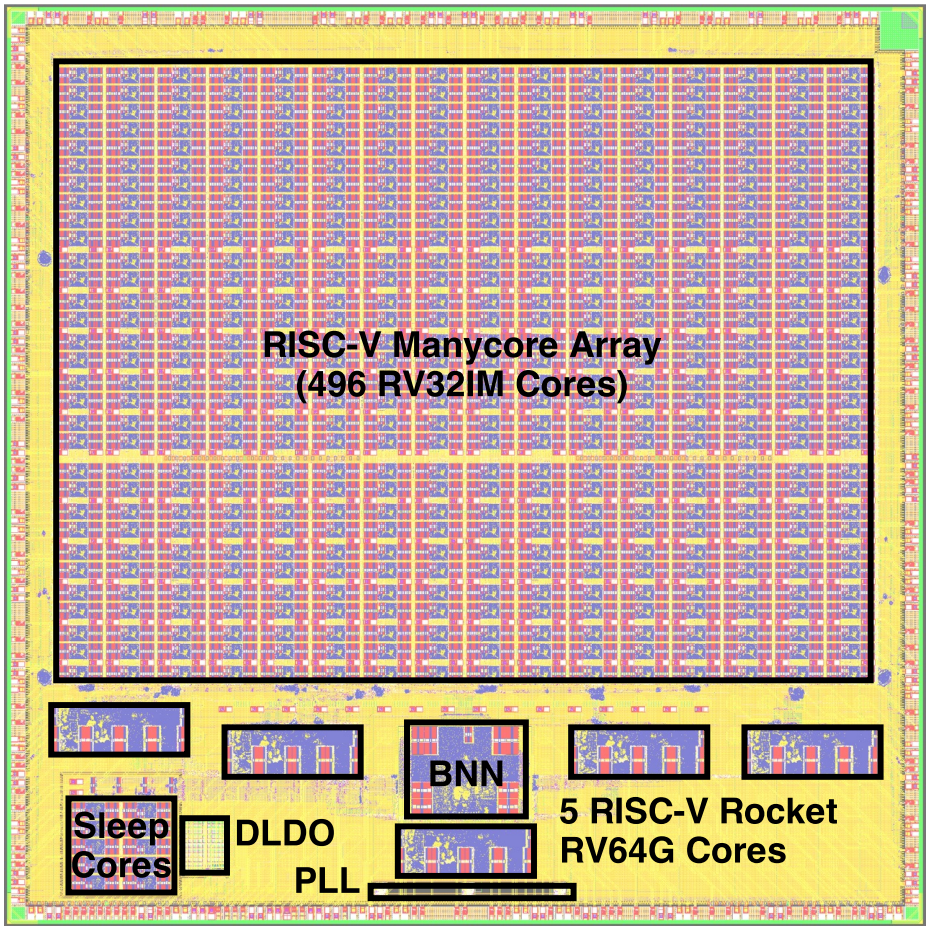


- Custom FPGA board for accelerating Bing search and other workloads
- Accelerators developed with/by app developers
- Tightly integrated into Microsoft data center's and cloud computing platforms, access gradually being given to outside developers

Celerity System-on-Chip

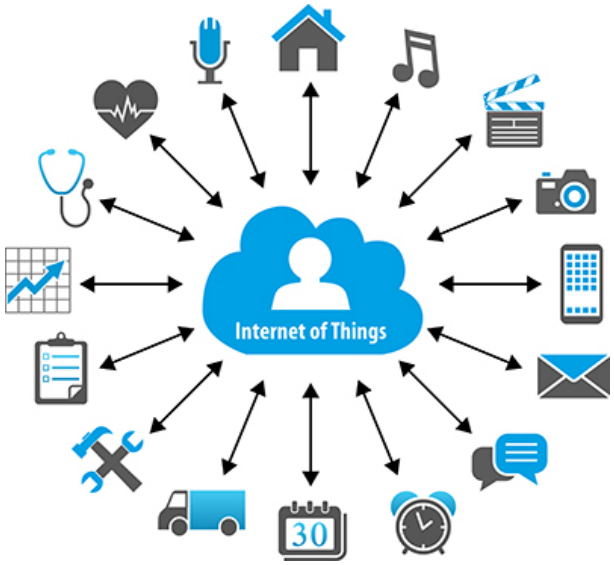
UCSD, Washington, Cornell, Michigan w/ DARPA CRAFT Program

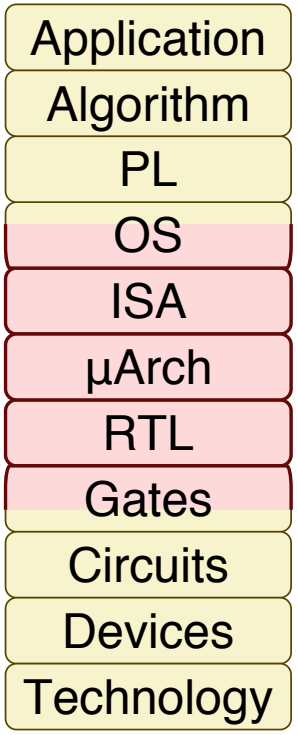
- 5 × 5mm in TSMC 16 nm FFC
- 385 million transistors
- 511 RISC-V cores
  - 5 Linux-capable Rocket cores
  - 496-core tiled manycore
  - 10-core low-voltage array
- 1 BNN accelerator
- 1 synthesizable PLL
- 1 synthesizable LDO Vreg
- 3 clock domains
- 672-pin flip chip BGA package
- 9-months from PDK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



Three Key Trends in Computer Engineering

Trend #1: Growing Diversity in Applications and Systems



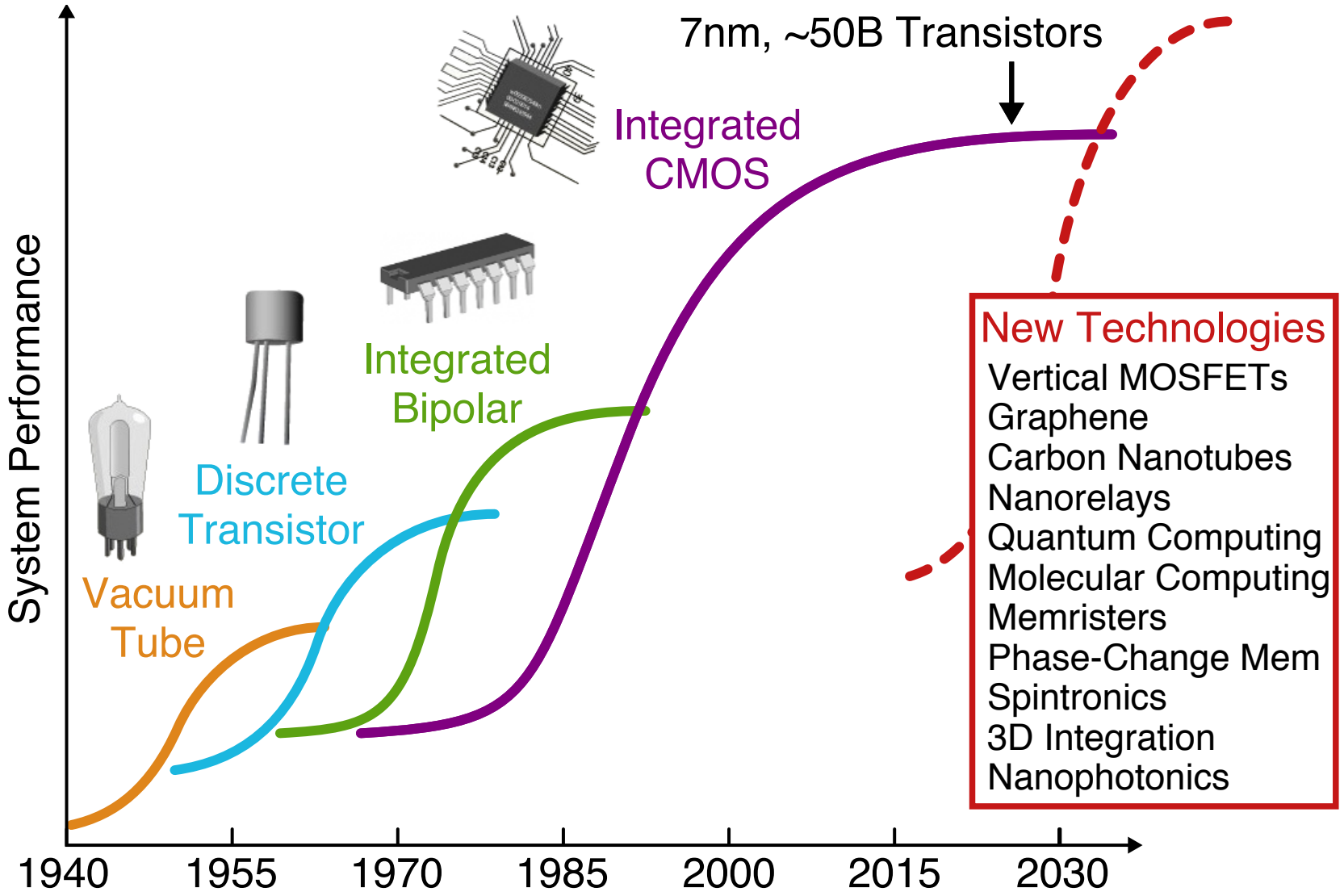


Trend #2: Software/Arch Interface Changing Radically

Trend #3: Technology/Arch Interface Changing Radically

Students entering the field of computer engineering have a unique opportunity to shape the future of computing and how it will impact society

Technology Scaling is Slowing

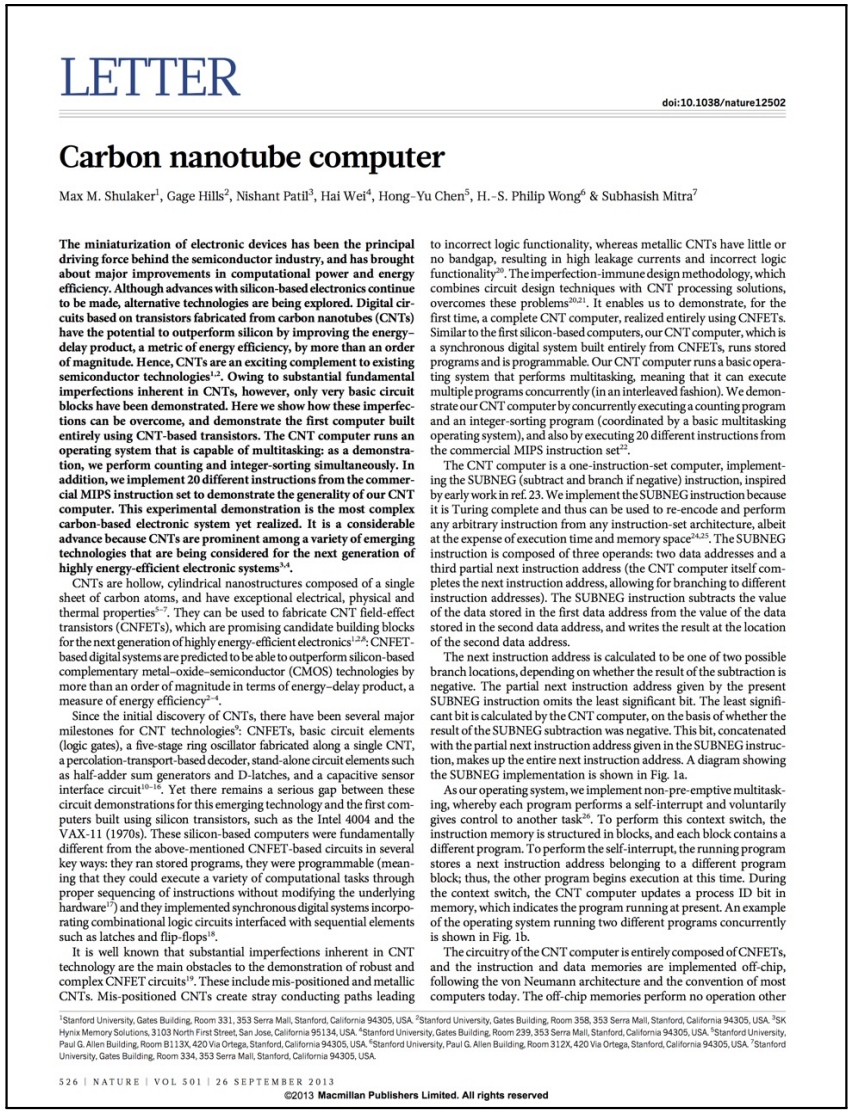


Adapted from D. Brooks Keynote at NSF XPS Workshop, May 2015.

Activity #3: A CNT Computer

`http://tiny.cc/engri1210-3`

1. Breakout into groups of 3 students
2. Skim Nature article on a CNT computer
3. Consider four questions
4. Use Google form to enter your answers
5. Come back into main zoom room



LETTER

doi:10.1038/nature12502

Carbon nanotube computer

Max M. Shulaker[1], Gage Hills[2], Nishant Patil[3], Hai Wei[4], Hong-Yu Chen[5], H.-S. Philip Wong[6] & Subhasish Mitra[7]

The miniaturization of electronic devices has been the principal driving force behind the semiconductor industry, and has brought about major improvements in computational power and energy efficiency. Although advances with silicon-based electronics continue to be made, alternative technologies are being explored. Digital circuits based on transistors fabricated from carbon nanotubes (CNTs) have the potential to outperform silicon by improving the energy–delay product, a metric of energy efficiency, by more than an order of magnitude. Hence, CNTs are an exciting complement to existing semiconductor technologies[1,2]. Owing to substantial fundamental imperfections inherent in CNTs, however, only very basic circuit blocks have been demonstrated. Here we show how these imperfections can be overcome, and demonstrate the first computer built entirely using CNT-based transistors. The CNT computer runs an operating system that is capable of multitasking: as a demonstration, we perform counting and integer-sorting simultaneously. In addition, we implement 20 different instructions from the commercial MIPS instruction set to demonstrate the generality of our CNT computer. This experimental demonstration is the most complex carbon-based electronic system yet realized. It is a considerable advance because CNTs are prominent among a variety of emerging technologies that are being considered for the next generation of highly energy-efficient electronic systems[3,4].

CNTs are hollow, cylindrical nanostructures composed of a single sheet of carbon atoms, and have exceptional electrical, physical and thermal properties[5–7]. They can be used to fabricate CNT field-effect transistors (CNFETs), which are promising candidate building blocks for the next generation of highly energy-efficient electronics[1,2,8]; CNFET-based digital systems are predicted to be able to outperform silicon-based complementary metal–oxide–semiconductor (CMOS) technologies by more than an order of magnitude in terms of energy–delay product, a measure of energy efficiency[2–4].

Since the initial discovery of CNTs, there have been several major milestones for CNT technologies[9]: CNFETs, basic circuit elements (logic gates), a five-stage ring oscillator fabricated along a single CNT, a percolation-transport-based decoder, stand-alone circuit elements such as half-adder sum generators and D-latches, and a capacitive sensor interface circuit[10–16]. Yet there remains a serious gap between these circuit demonstrations for this emerging technology and the first computers built using silicon transistors, such as the Intel 4004 and the VAX-11 (1970s). These silicon-based computers were fundamentally different from the above-mentioned CNFET-based circuits in several key ways: they ran stored programs, they were programmable (meaning that they could execute a variety of computational tasks through proper sequencing of instructions without modifying the underlying hardware[17]) and they implemented synchronous digital systems incorporating combinational logic circuits interfaced with sequential elements such as latches and flip-flops[18].

It is well known that substantial imperfections inherent in CNT technology are the main obstacles to the demonstration of robust and complex CNFET circuits[19]. These include mis-positioned and metallic CNTs. Mis-positioned CNTs create stray conducting paths leading to incorrect logic functionality, whereas metallic CNTs have little or no bandgap, resulting in high leakage currents and incorrect logic functionality[20]. The imperfection-immune design methodology, which combines circuit design techniques with CNT processing solutions, overcomes these problems[20,21]. It enables us to demonstrate, for the first time, a complete CNT computer, realized entirely using CNFETs. Similar to the first silicon-based computers, our CNT computer, which is a synchronous digital system built entirely from CNFETs, runs stored programs and is programmable. Our CNT computer runs a basic operating system that performs multitasking, meaning that it can execute multiple programs concurrently (in an interleaved fashion). We demonstrate our CNT computer by concurrently executing a counting program and an integer-sorting program (coordinated by a basic multitasking operating system), and also by executing 20 different instructions from the commercial MIPS instruction set[22].

The CNT computer is a one-instruction-set computer, implementing the SUBNEG (subtract and branch if negative) instruction, inspired by early work in ref. 23. We implement the SUBNEG instruction because it is Turing complete and thus can be used to re-encode and perform any arbitrary instruction from any instruction-set architecture, albeit at the expense of execution time and memory space[24,25]. The SUBNEG instruction is composed of three operands: two data addresses and a third partial next instruction address (the CNT computer itself completes the next instruction address, allowing for branching to different instruction addresses). The SUBNEG instruction subtracts the value of the data stored in the first data address from the value of the data stored in the second data address, and writes the result at the location of the second data address.

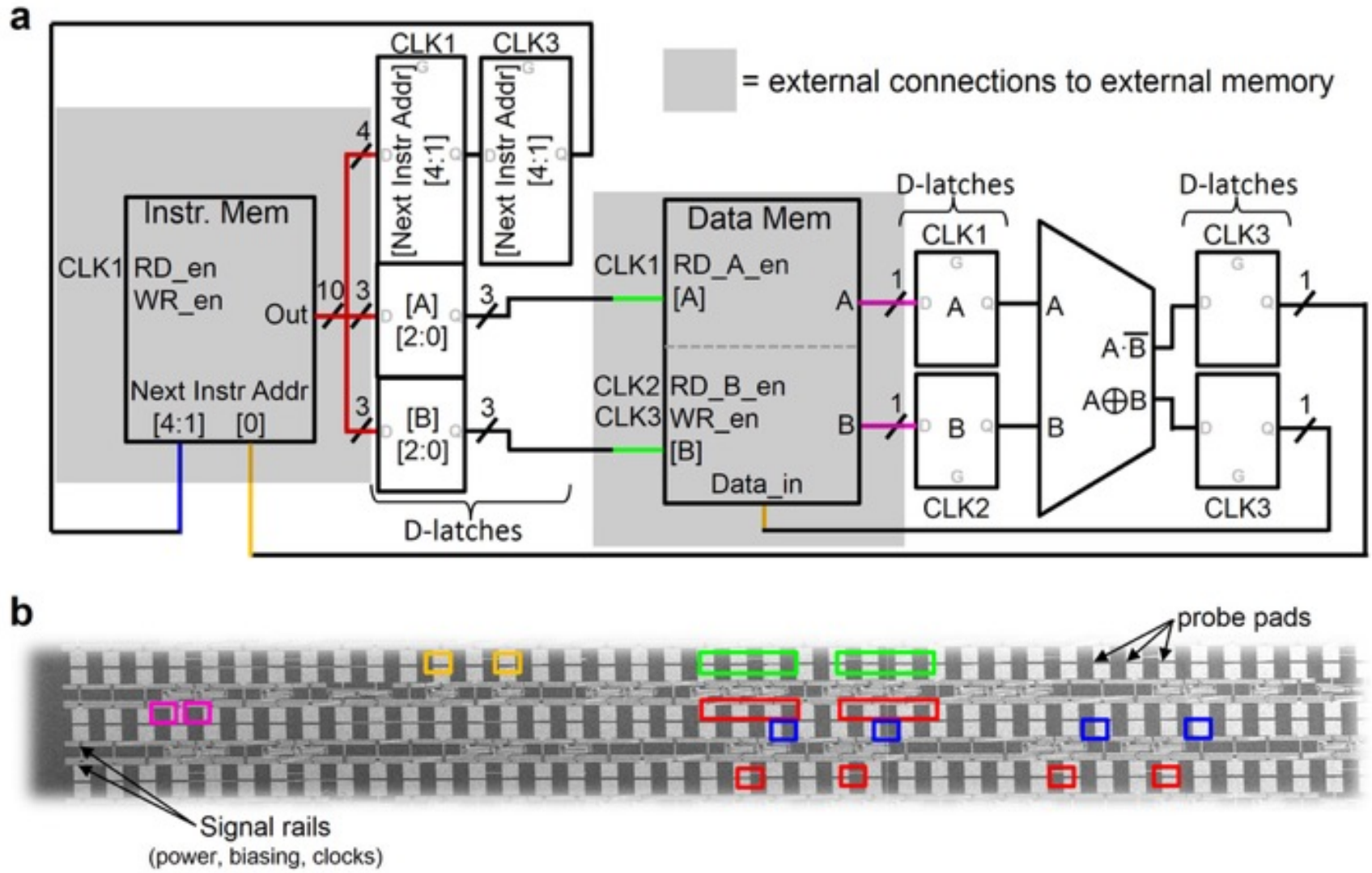
The next instruction address is calculated to be one of two possible branch locations, depending on whether the result of the subtraction is negative. The partial next instruction address given by the present SUBNEG instruction omits the least significant bit. The least significant bit is calculated by the CNT computer, on the basis of whether the result of the SUBNEG subtraction was negative. This bit, concatenated with the partial next instruction address given in the SUBNEG instruction, makes up the entire next instruction address. A diagram showing the SUBNEG implementation is shown in Fig. 1a.

As our operating system, we implement non-pre-emptive multitasking, whereby each program performs a self-interrupt and voluntarily gives control to another task[26]. To perform this context switch, the instruction memory is structured in blocks, and each block contains a different program. To perform the self-interrupt, the running program stores a next instruction address belonging to a different program block; thus, the other program begins execution at this time. During the context switch, the CNT computer updates a process ID bit in memory, which indicates the program running at present. An example of the operating system running two different programs concurrently is shown in Fig. 1b.

The circuitry of the CNT computer is entirely composed of CNFETs, and the instruction and data memories are implemented off-chip, following the von Neumann architecture and the convention of most computers today. The off-chip memories perform no operation other

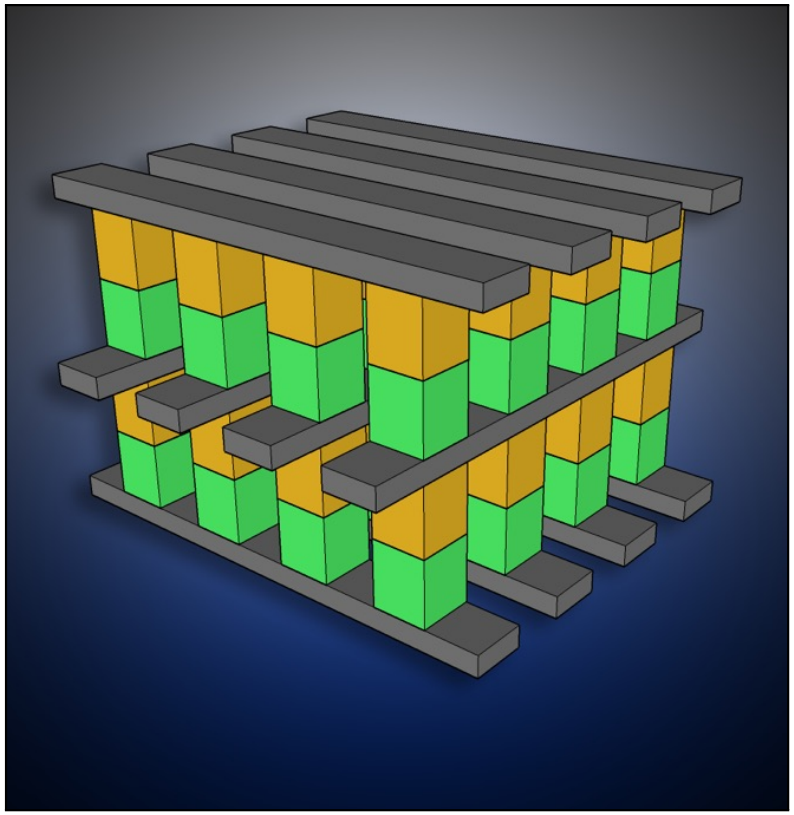
[1]Stanford University, Gates Building, Room 331, 353 Serra Mall, Stanford, California 94305, USA. [2]Stanford University, Gates Building, Room 358, 353 Serra Mall, Stanford, California 94305, USA. [3]SK Hynix Memory Solutions, 3103 North First Street, San Jose, California 95134, USA. [4]Stanford University, Gates Building, Room 239, 353 Serra Mall, Stanford, California 94305, USA. [5]Stanford University, Paul G. Allen Building, Room B113X, 420 Via Ortega, Stanford, California 94305, USA. [6]Stanford University, Paul G. Allen Building, Room 312X, 420 Via Ortega, Stanford, California 94305, USA. [7]Stanford University, Gates Building, Room 334, 353 Serra Mall, Stanford, California 94305, USA.

A Carbon Nanotube "Computer"



Adapted from M. Shulaker, et al., "Carbon Nanotube Computer," Nature, 2013.

Examples of Emerging Technologies

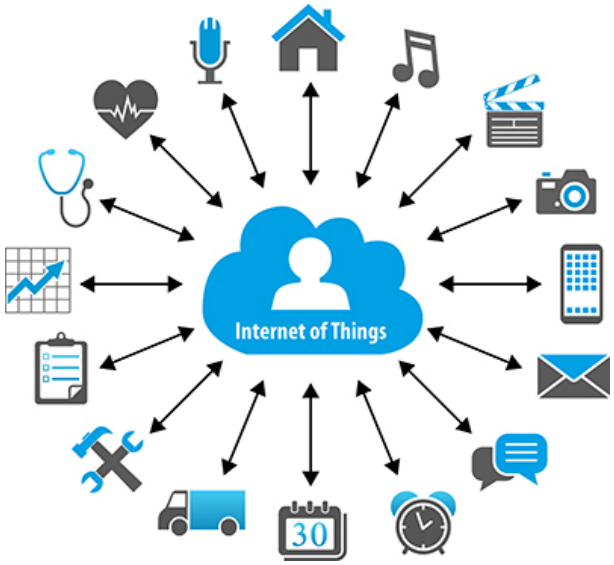


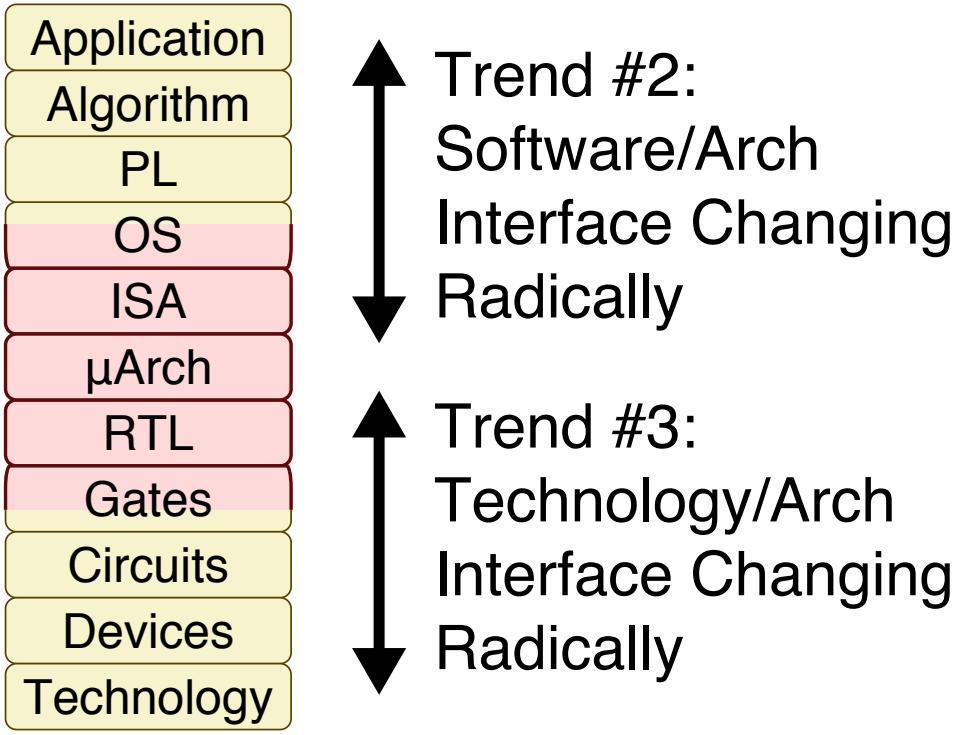
Intel 3D Crosspoint Memory
Resistive memory enables very high density, non-volatile storage with fast access times

D-Wave
Quantum annealing computer suitable for solving complex optimization problems

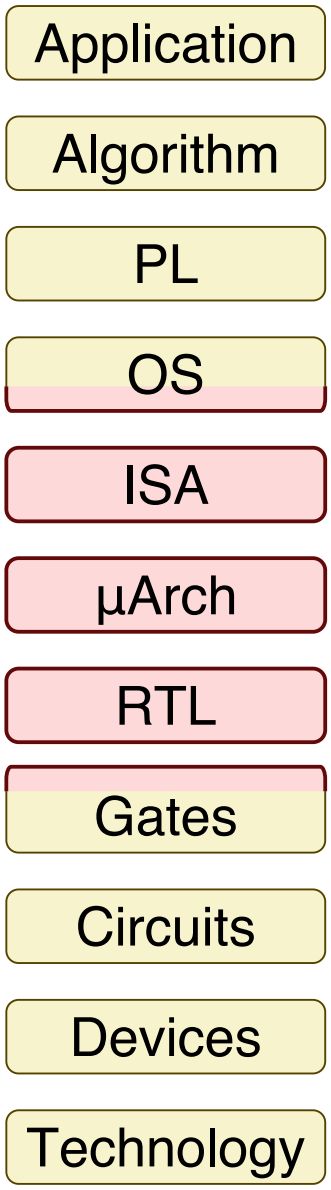
Three Key Trends in Computer Engineering

Trend #1: Growing Diversity in Applications and Systems





Students entering the field of computer engineering have a unique opportunity to shape the future of computing and how it will impact society



Agenda

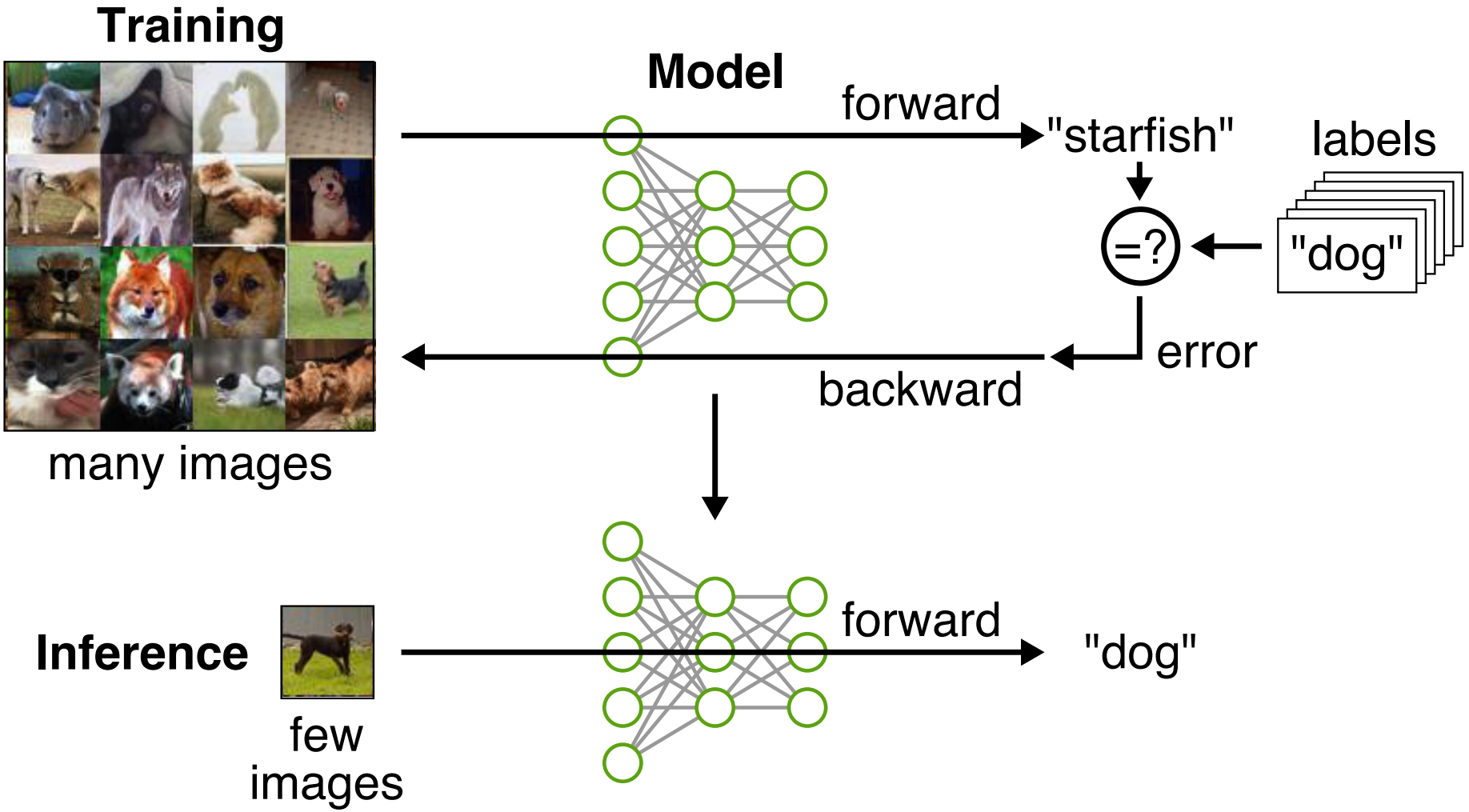
The Computer Systems Stack

Trends in Computer Engineering

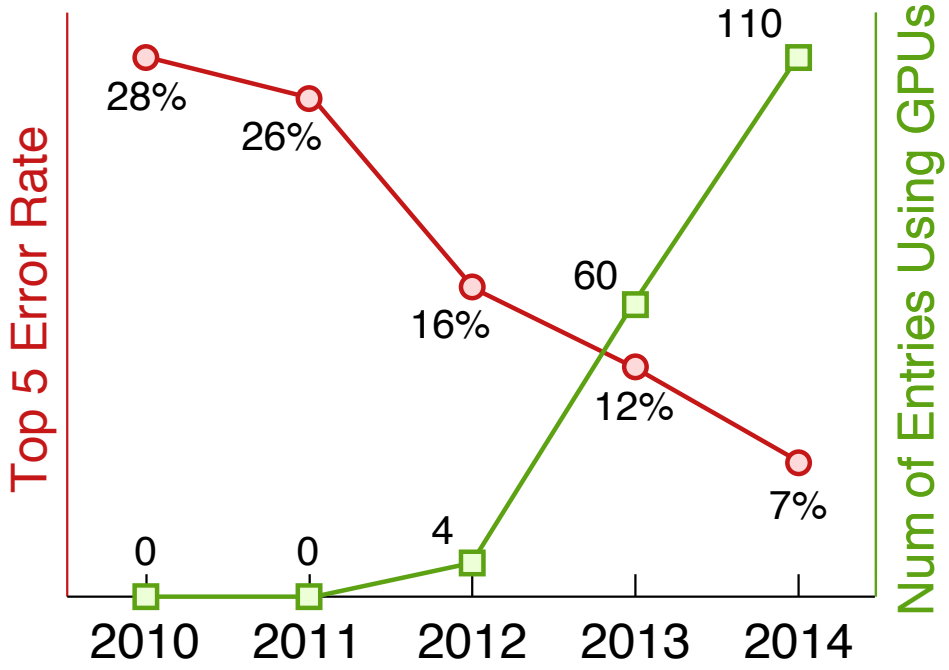
Hardware Acceleration for Deep Learning

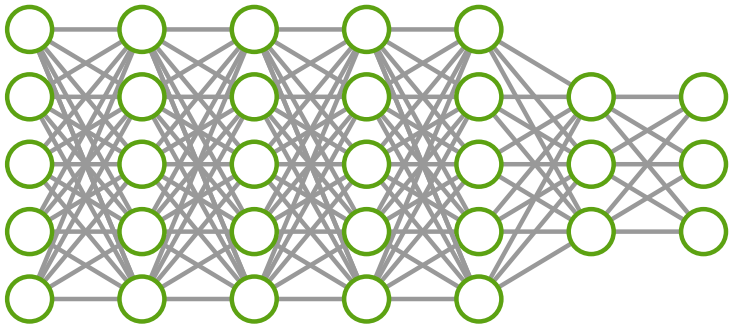
Image Recognition

Training vs. Inference



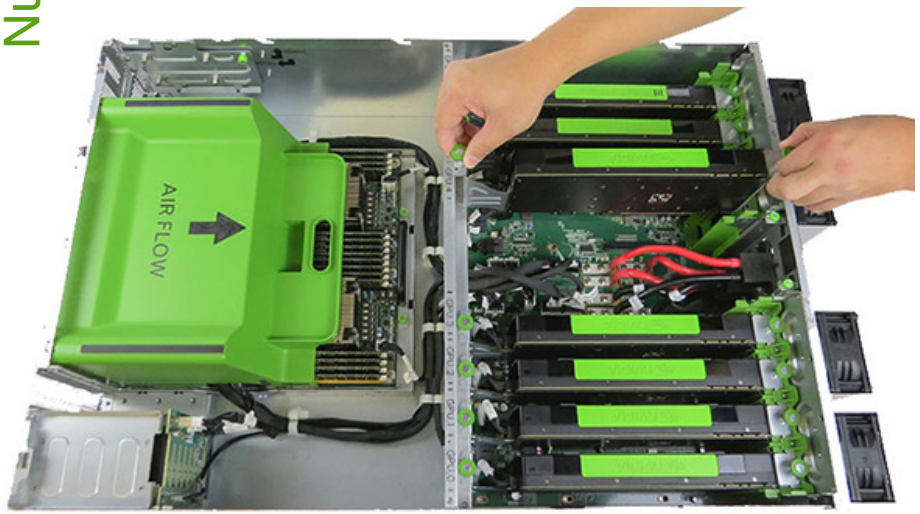
ImageNet Large-Scale Visual Recognition Challenge



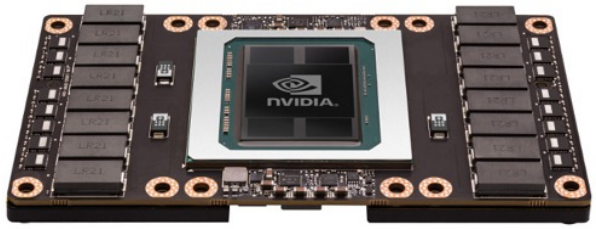


Software: Deep Neural Network

Hardware: Graphics Processing Units

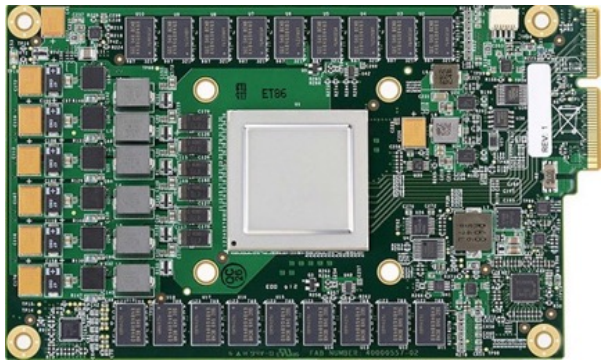


DL Hardware Acceleration in the Cloud



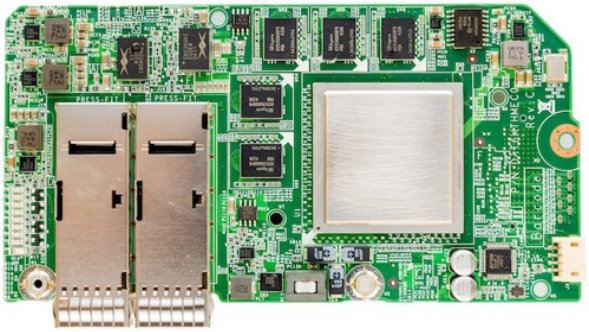
NVIDIA DGX-1

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



Google TPU

- Custom chip specifically designed to accelerate Google's TensorFlow C++ library
- Tightly integrated into Google's data centers
- 15–30× faster than contemporary CPU and GPUs



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

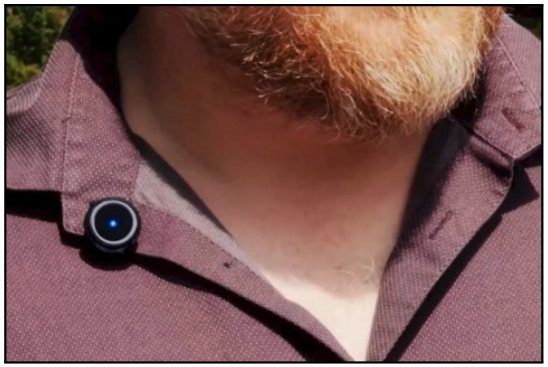
DL Hardware Acceleration for ML at the Edge

Amazon Echo

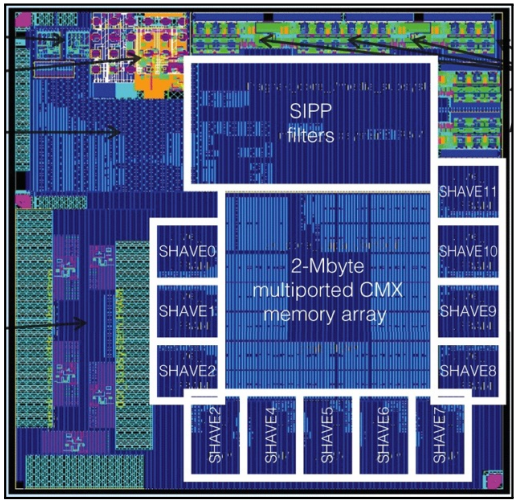
- Developing AI chips so Echo line can do more on-board processing
- Reduces need for round-trip to cloud
- Co-design the algorithms and the underlying hardware

Facebook Oculus

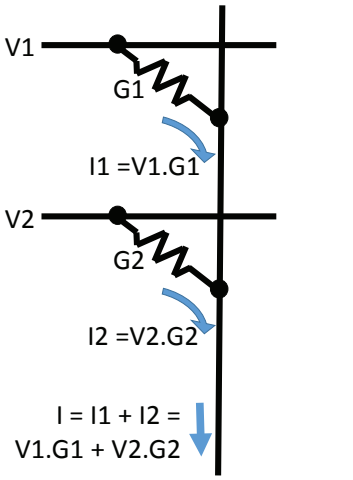
- Starting to design custom chips for Oculus VR headsets
- Significant performance demands under strict power requirements



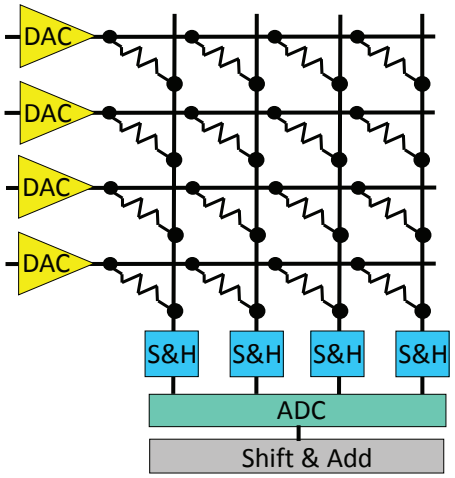
Movidius Myriad 2



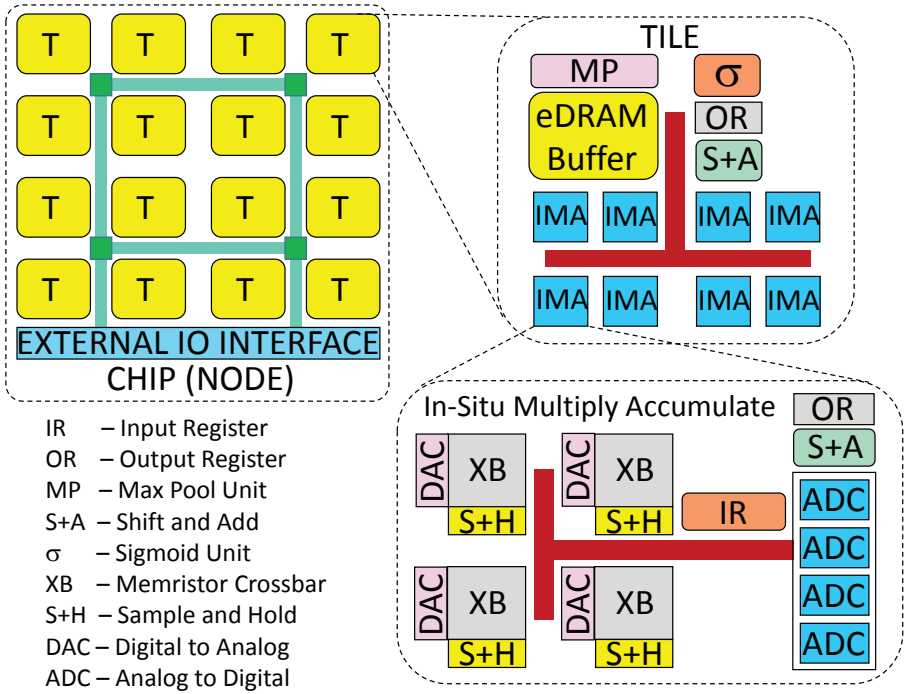
ML Acceleration Can Incorporate All Three Trends



(a) Multiply-Accumulate operation



(b) Vector-Matrix Multiplier



IR – Input Register
OR – Output Register
MP – Max Pool Unit
S+A – Shift and Add
σ – Sigmoid Unit
XB – Memristor Crossbar
S+H – Sample and Hold
DAC – Digital to Analog
ADC – Analog to Digital

- ISAAC: Convolutional neural network accelerator which uses in-situ analog arithmetic in crossbars of emerging resistive memory devices
- Captures all three trends
  - New applications and systems in ultra-low-power TinyML
  - New software/architecture interface for accelerator
  - New technology/architecture interface with non-traditional devices

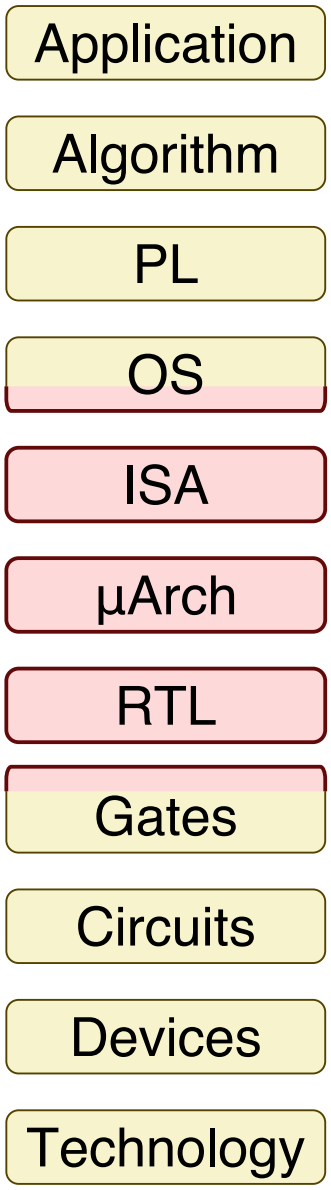
Adapted from A. Shafiee et al., ISCA, 2016.

Top-five software companies are all making chips

- **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, wireless chips

Chip startup ecosystem for machine learning is thriving!

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



Take-Away Points

- We are entering an exciting new era of computer engineering
  - Growing diversity in applications & systems
  - Radical rethinking of software/architecture interface
  - Radical rethinking of technology/architecture interface
- This era offers tremendous challenges and opportunities, which makes it a wonderful time to study and contribute to the field of computer engineering

ECE 2400 Computer Systems Programming

- **Part 1: Procedural Programming**
  - introduction to C, variables, expressions, functions, conditional & iteration statements, recursion, static types, pointers, arrays, dynamic allocation
- **Part 2: Basic Algorithms and Data Structures**
  - lists, vectors, complexity analysis, insertion sort, selection sort, merge sort, quick sort, hybrid sorts, stacks, queues, sets, maps
- **Part 3: Multi-Paradigm Programming**
  - transition to C++, namespaces, flexible function prototypes, references, exceptions, new/delete, *object oriented programming* (C++ classes and inheritance for dynamic polymorphism), *generic programming* (C++ templates for static polymorphism), *functional programming* (C++ functors and lambdas), *concurrent programming* (C++ threads and atomics)
- **Part 4: More Algorithms and Data Structures**
  - trees (binary trees, binary search trees), tables (lookup tables, hash tables), graphs (DFS, BFS, shortest path first, minimum spanning trees)

ECE 2400 Computer Systems Programming

- **PA1–3: Fundamentals**
  - PA1: Math functions
  - PA2: List and Vector Data Structures
  - PA3: Sorting Algorithms
- **PA4–5: Handwriting Recognition System**
  - PA5: Linear vs. Binary Searching
  - PA5: Trees vs. Tables
- **Every programming assignment involves**
  - C/C++ “agile” programming
  - State-of-the-art tools for build systems, version control, continuous integration, code coverage
  - Performance measurement
  - Short technical report

Application-Level Software

System-Level Software