A New Era of Highly Specialized Computing Systems

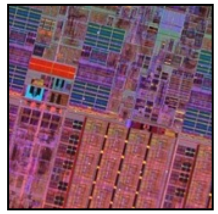
Christopher Batten

Associate Professor	@ ECE Department, Cornell University
Visiting Scholar	@ Computer Laboratory, University of Cambridge
Visiting Fellow	@ Clare Hall, University of Cambridge

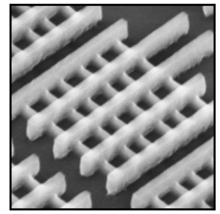
Clare Hall Colloquium
May 2018

Power Systems

Computer Engineering



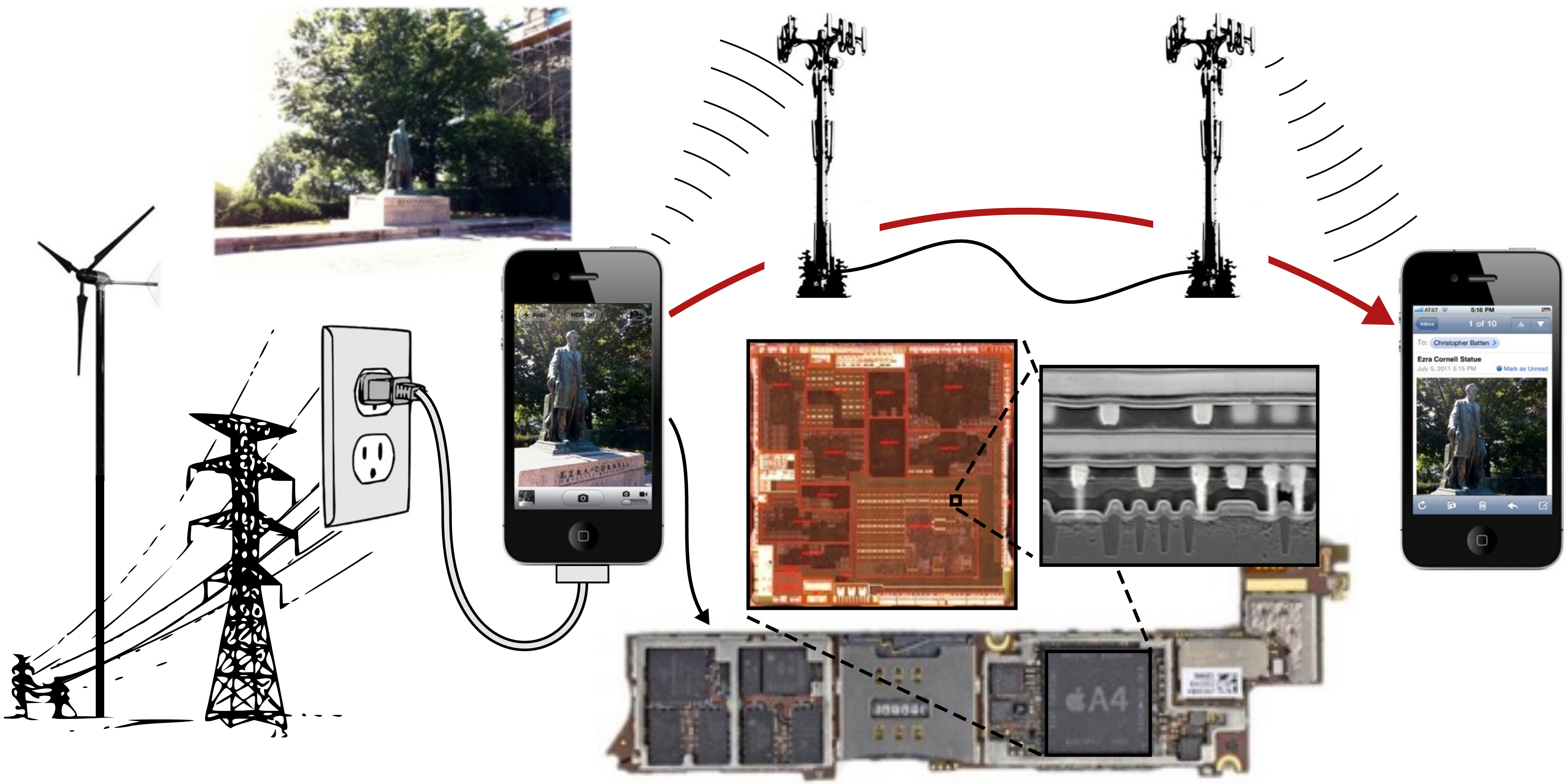
Electrical Circuits



Electrical Devices

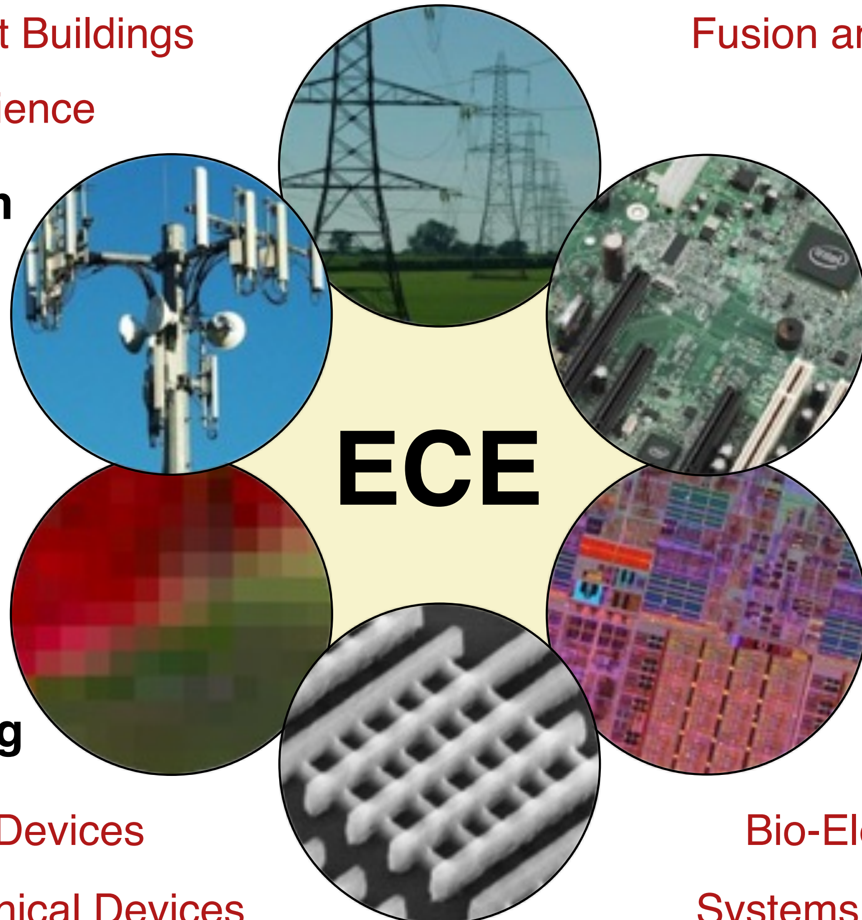
Signal Processing

Telecomm



ECE is the Study and Application of Electricity, Micro-Electronics, and Electro-Magnetism

Power Systems



Smart Grid and Smart Buildings

Atmospheric Science

Telecomm

Information Theory

Network Protocols and Optimization

Image, Audio, Video Processing

Signal Processing

Opto-Electrical Devices

Micro-Electro-Mechanical Devices

Fusion and Plasma Physics

Control Theory

Computer Engineering

Robotics

Computer-Aided Design

Analog and Digital Circuits

Electrical Circuits

Bio-Electrical Engineering

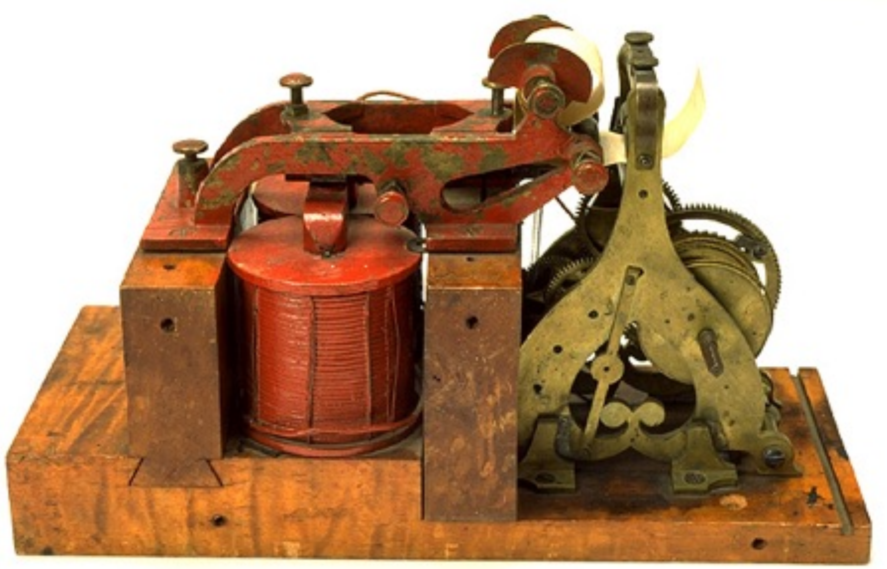
Systems and Synthetic Biology

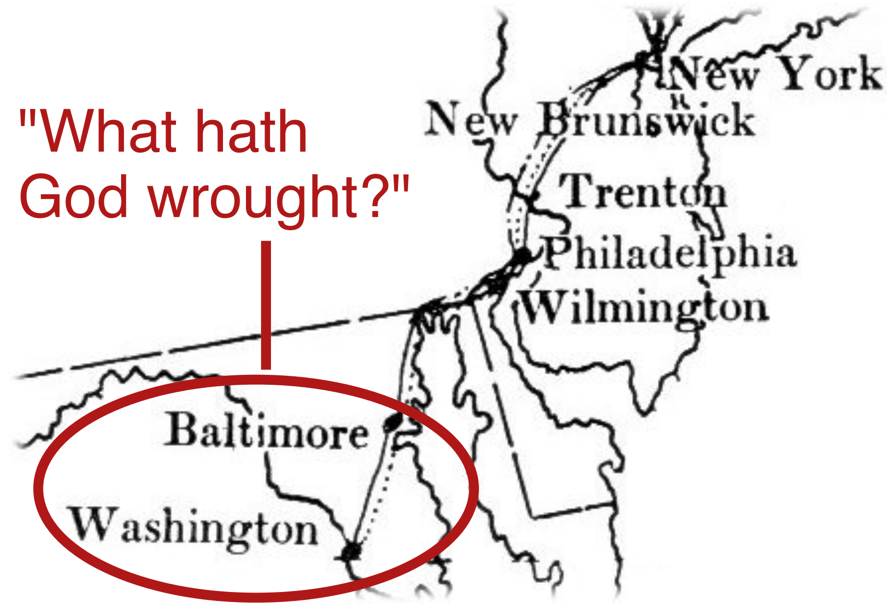
Electrical Devices

Cornell was founded because of ECE!

Samuel Morse invented the telegraph (a digital communication device), but needed help building the network

Ezra Cornell built the first telegraph line (the beginning of telecommunications), and invested in the Western Union Telegraph Co

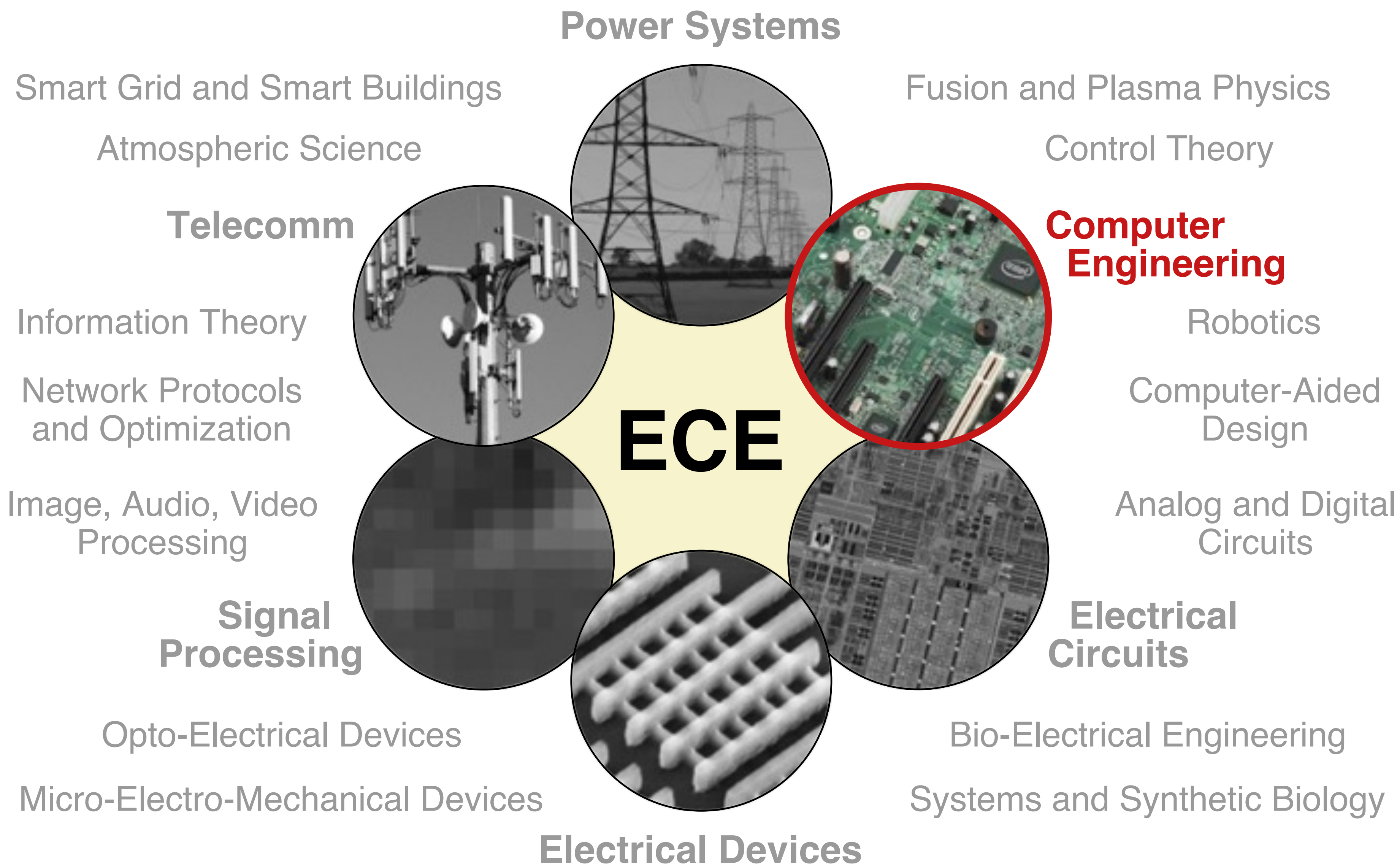




Ezra Cornell's investments created the fortune that eventually enabled the founding of Cornell University


EZRA CORNELL

Computer Engineering



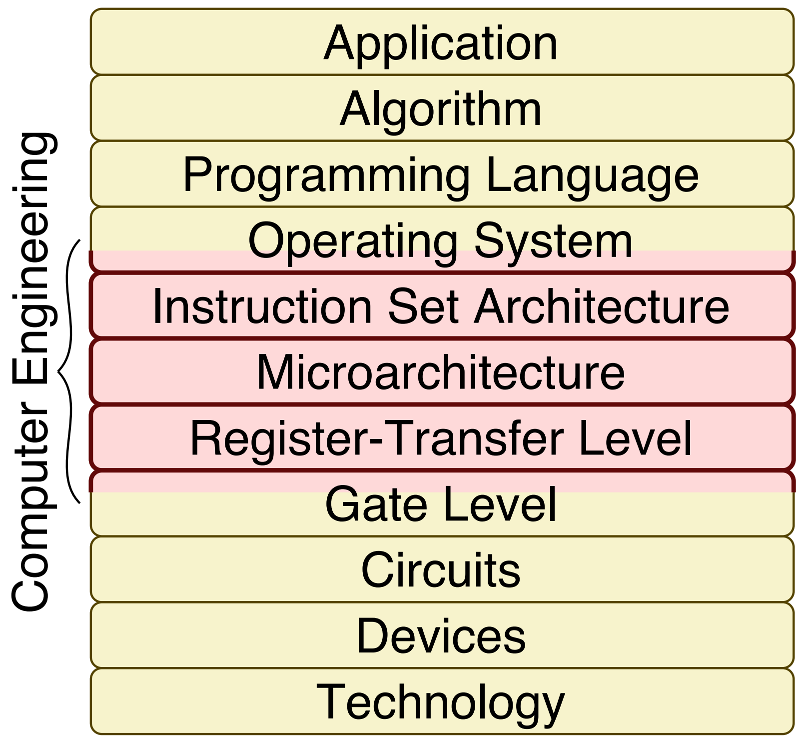
The Computer Systems Stack

Application

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

Technology

The Computer Systems Stack



Sort an array of numbers

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

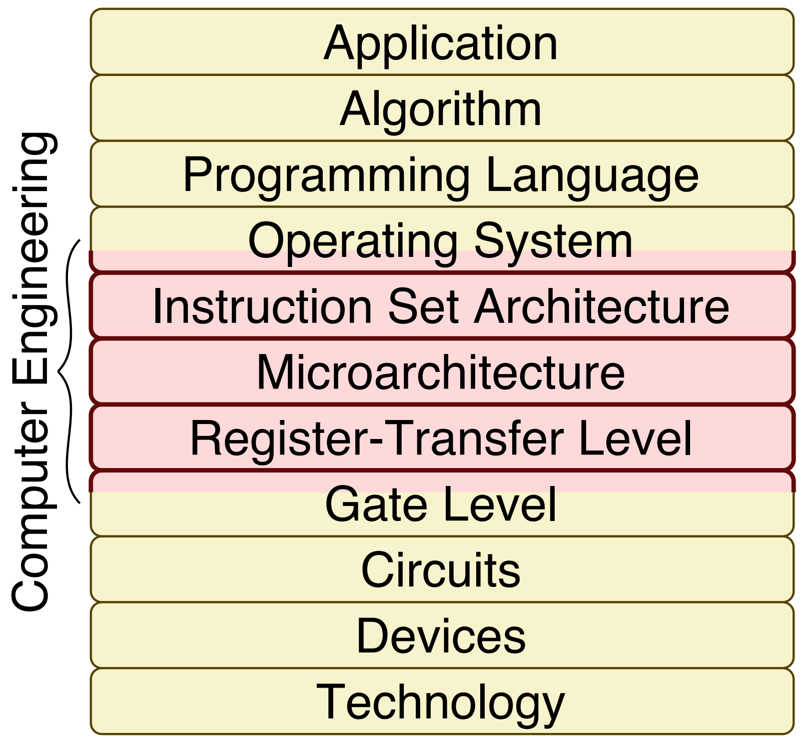
Out-of-place selection sort

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

C implementation of selection sort

```
void isort( 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

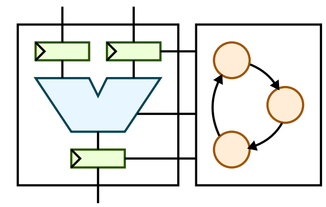
```
blez  $a2, done
move  $a7, $zero
li    $t4, 99
move  $a4, $a1
move  $v1, $zero
li    $a3, 99
lw    $a5, 0($a4)
addiu $a4, $a4, 4
slt   $a6, $a5, $a3
movn  $v0, $v1, $a6
addiu $v1, $v1, 1
movn  $a3, $a5, $a6
```

The Computer Systems Stack

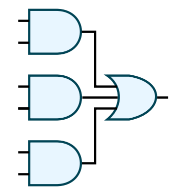
Computer Engineering

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

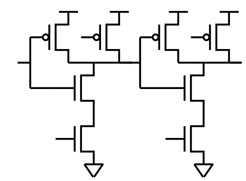
How data flows through system



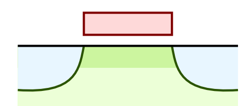
Boolean logic gates and functions



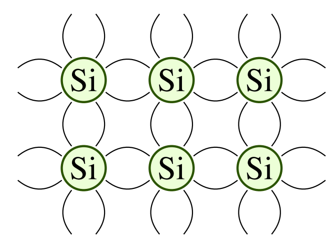
Combining devices to do useful work



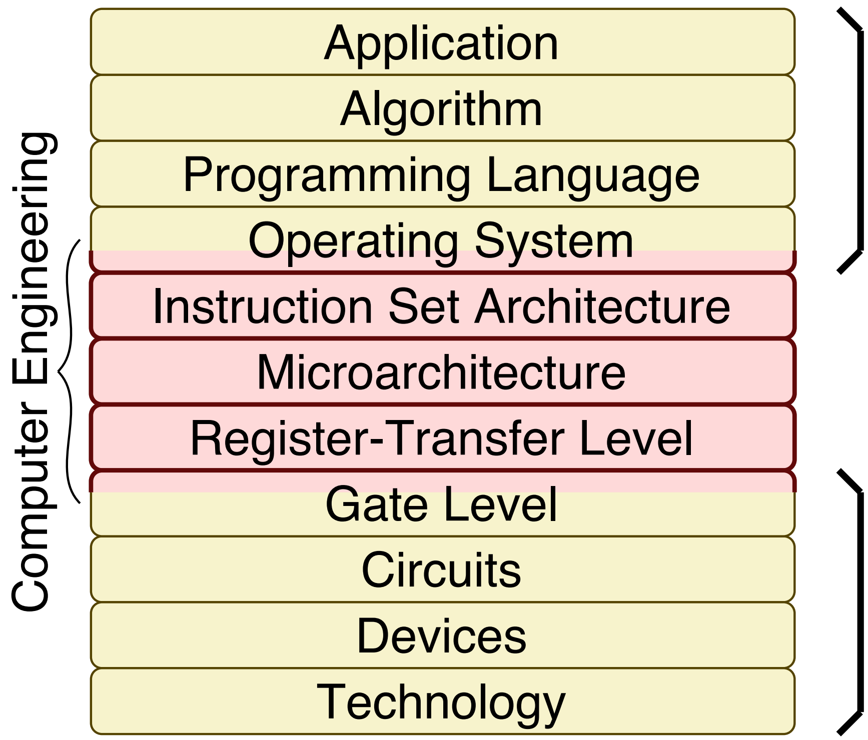
Transistors and wires



Silicon process technology



Computer Systems: CS vs. EE vs. CE



Traditional Computer Science

Computer Engineering is at the interface between hardware and software and considers the entire system

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

Application

Algorithm

PL

OS

ISA

µArch

RTL

Gates

Circuits

Devices

Technology

Agenda

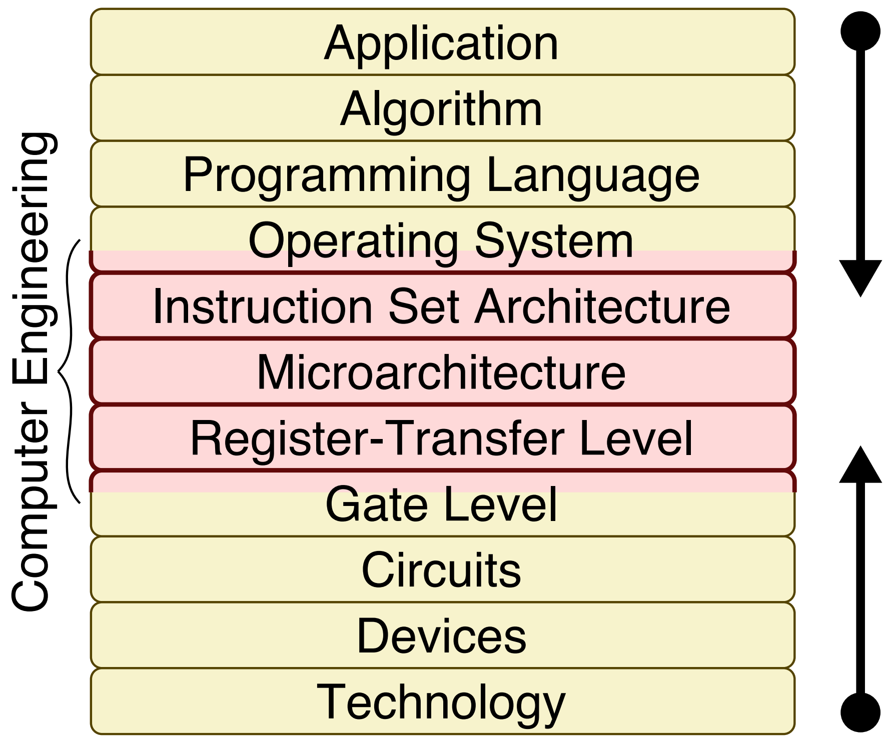
What is Computer Engineering?

Trends in Computer Engineering

- Trend #1: Bell's "Law"
- Trend #2: Moore's "Law"

Specialized Computer Systems

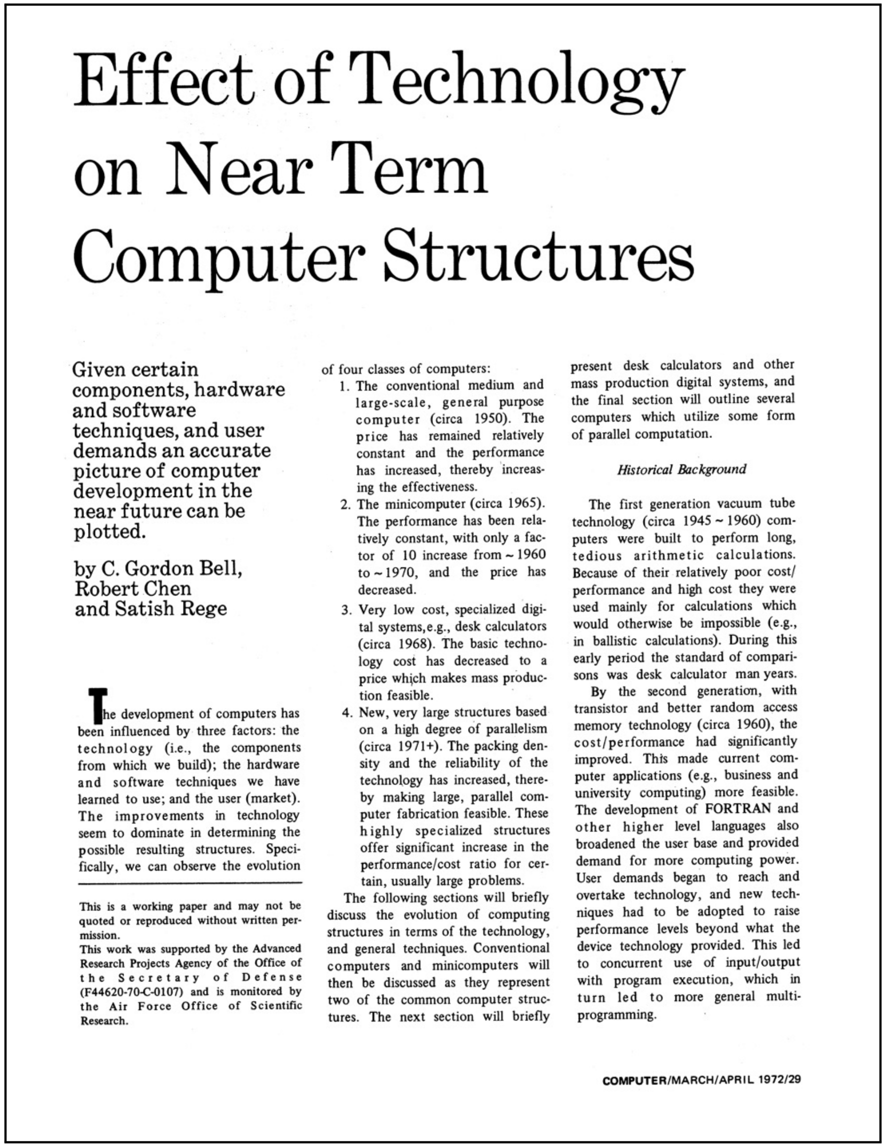
Trends in Computer Engineering



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.

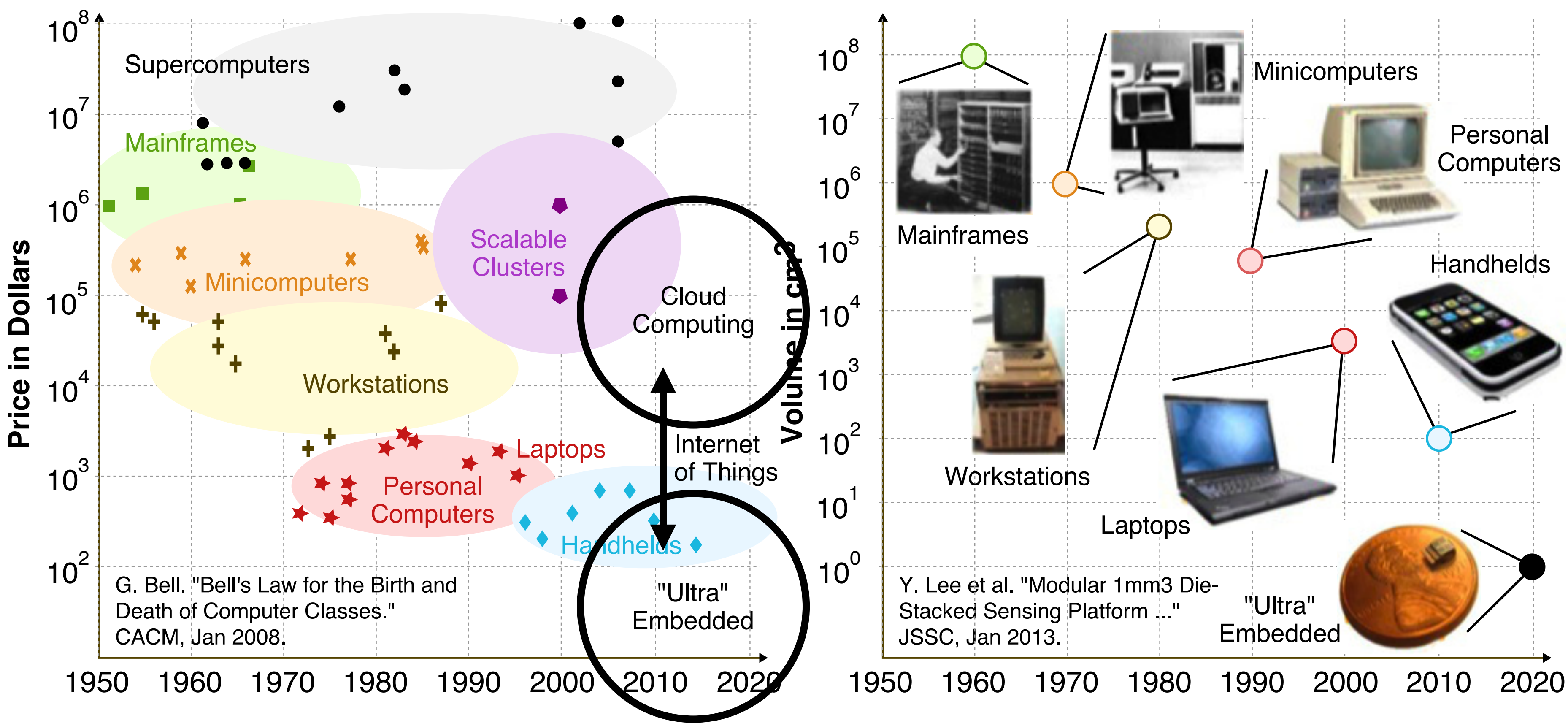
This is a working paper and may not be quoted or reproduced without written permission.
This work was supported by the Advanced Research Projects Agency of the Office of the Secretary of Defense (F44620-70-C-0107) and is monitored by the Air Force Office of Scientific Research.

COMPUTER/MARCH/APRIL 1972/29

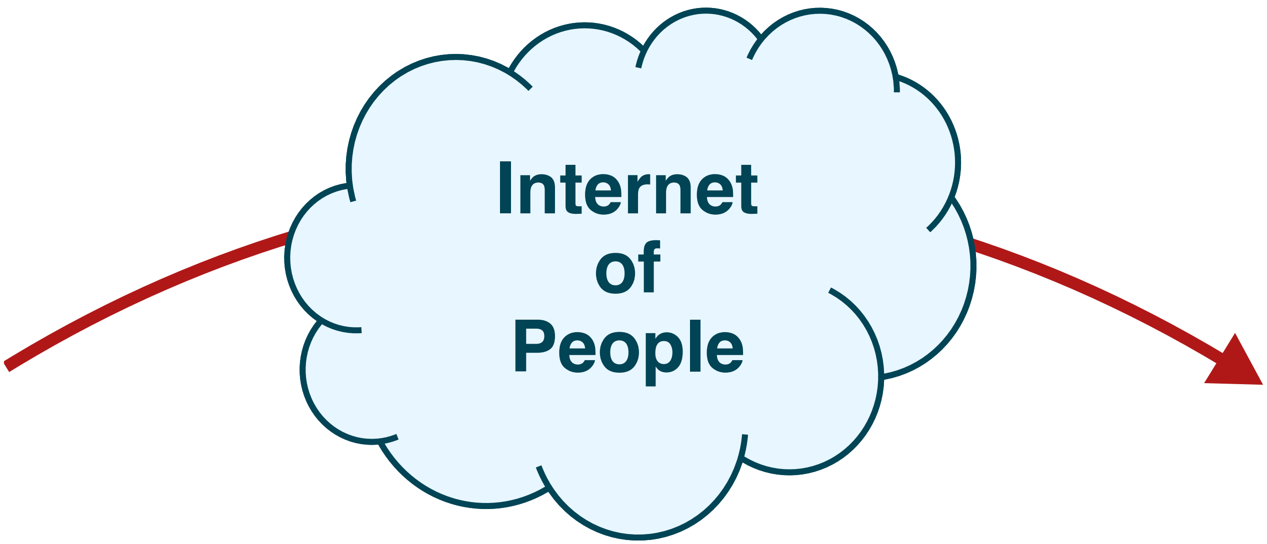
- Vice-President of Engineering at Digital Equipment Corporation
- Helped found Microsoft Research
- 1972 paper in IEEE Computer

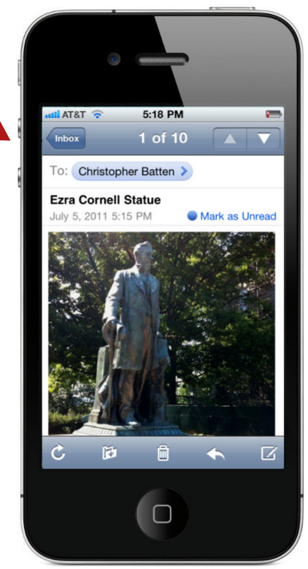
Trend #1: Bell's "Law"

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



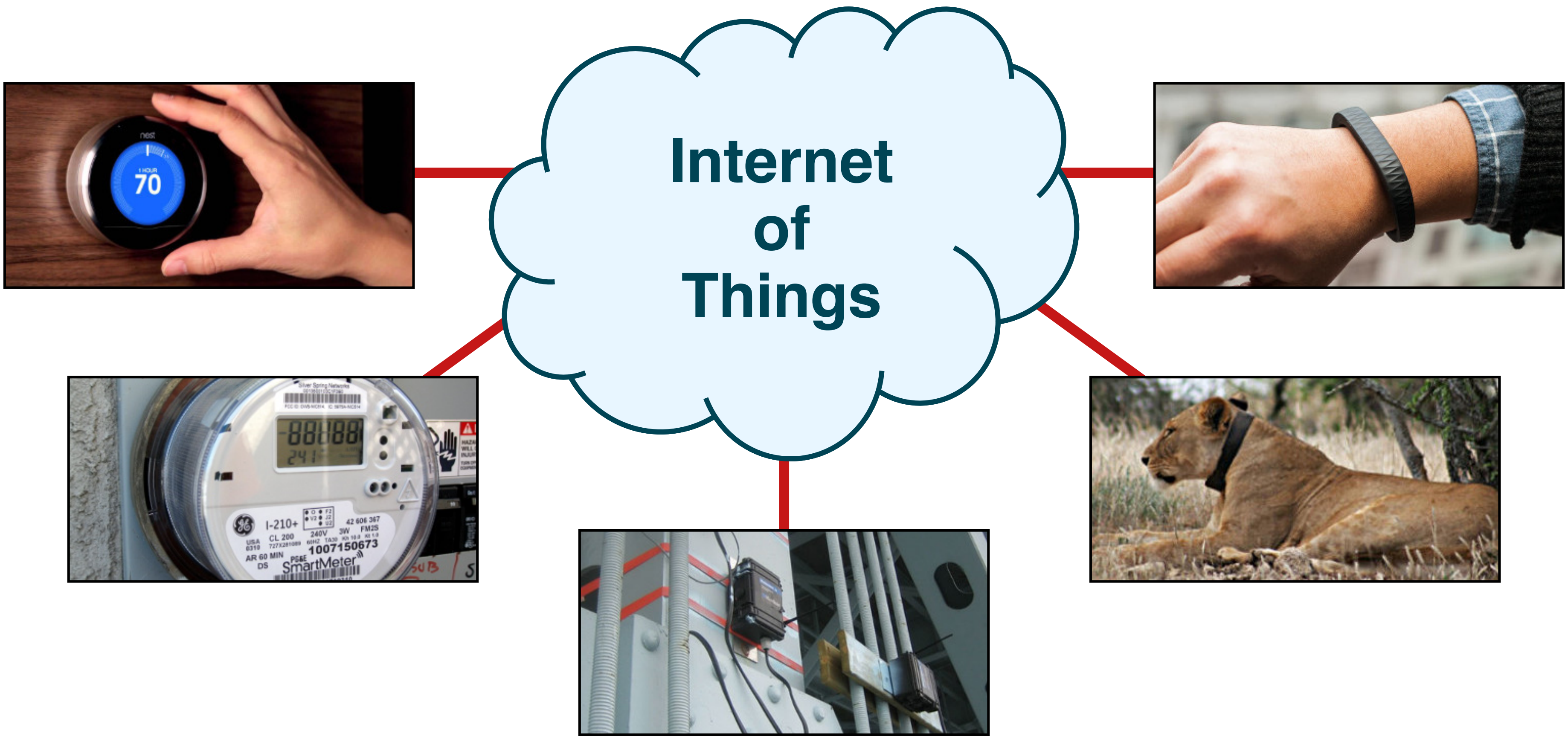
The “Traditional” Internet





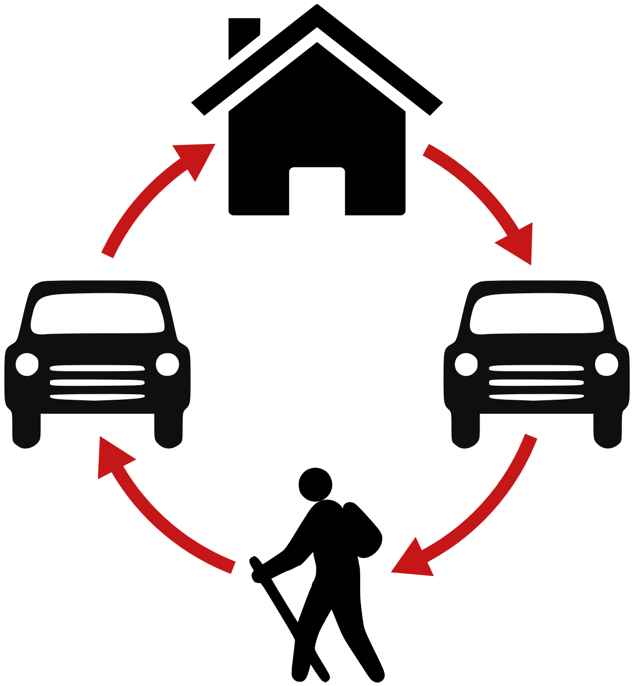
Human beings must
collect, enter, publish, and analyze
almost all of the information
that is transmitted over the Internet

Emerging Trend Towards an Internet of Things

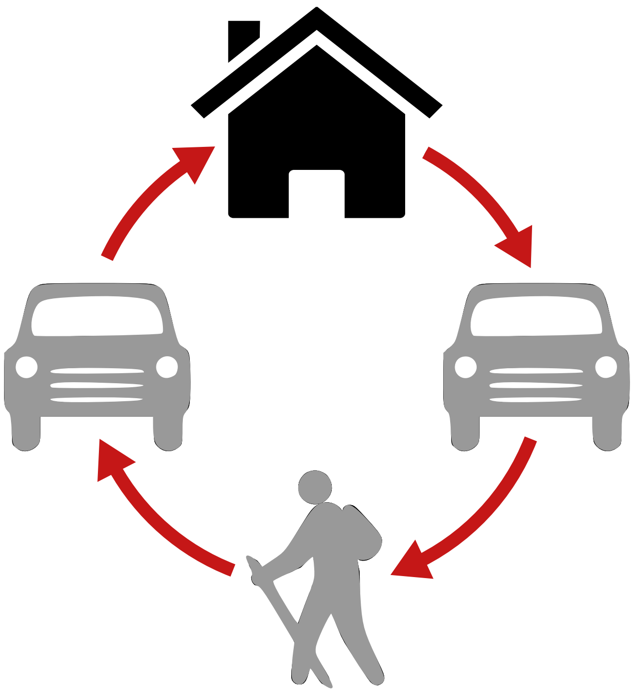


Interconnected "things" augmented with inexpensive embedded controllers, sensors, actuators to collect information and interact with the world

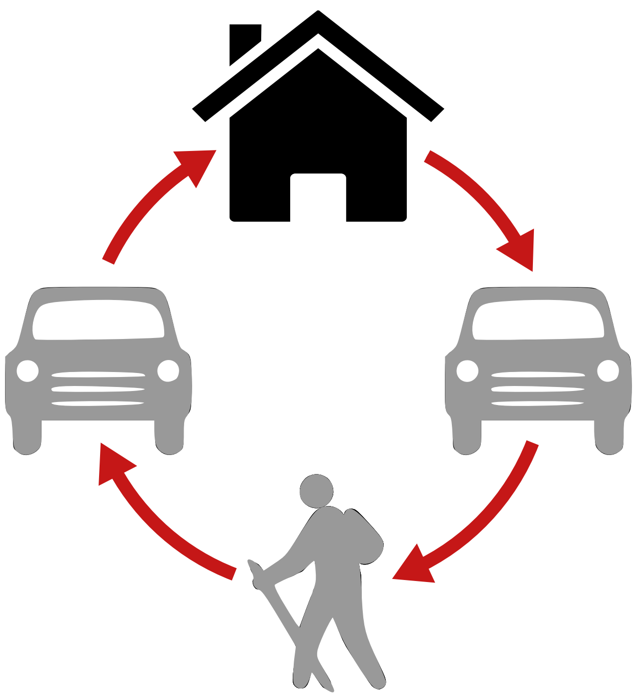
IoT Example: Spending the Day Hiking

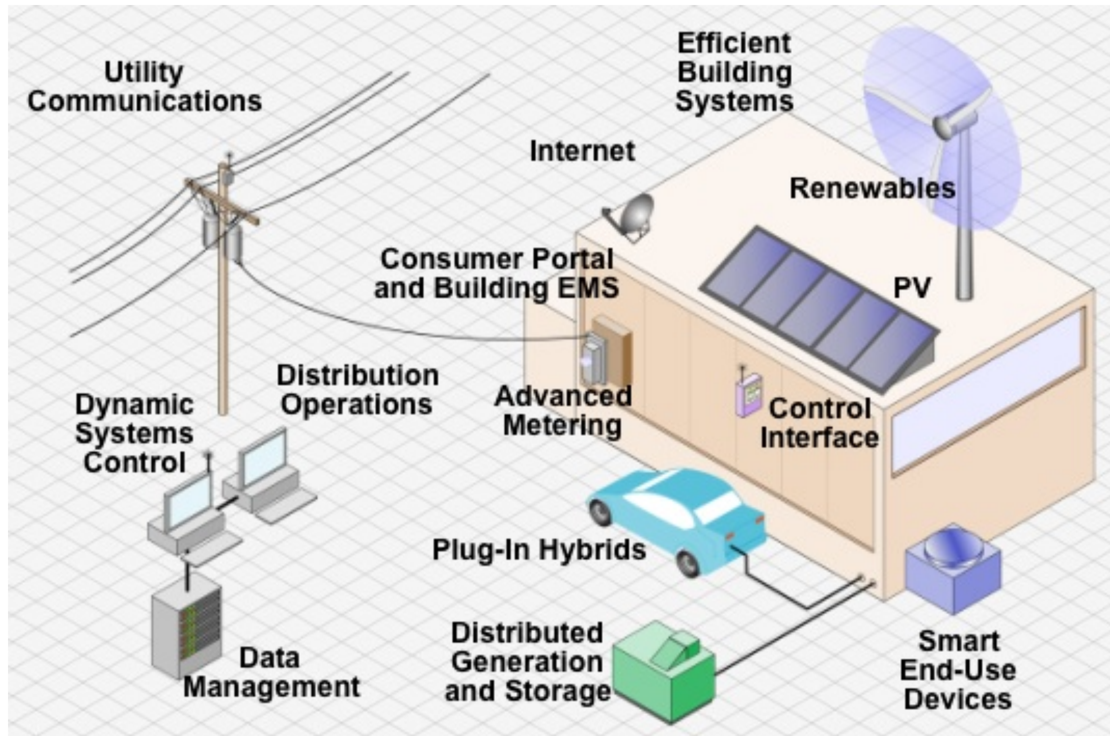


IoT Smart Home

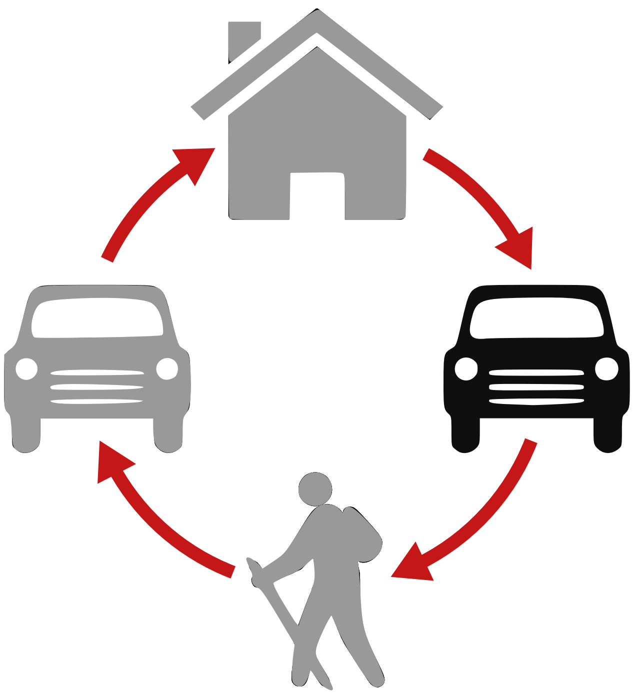


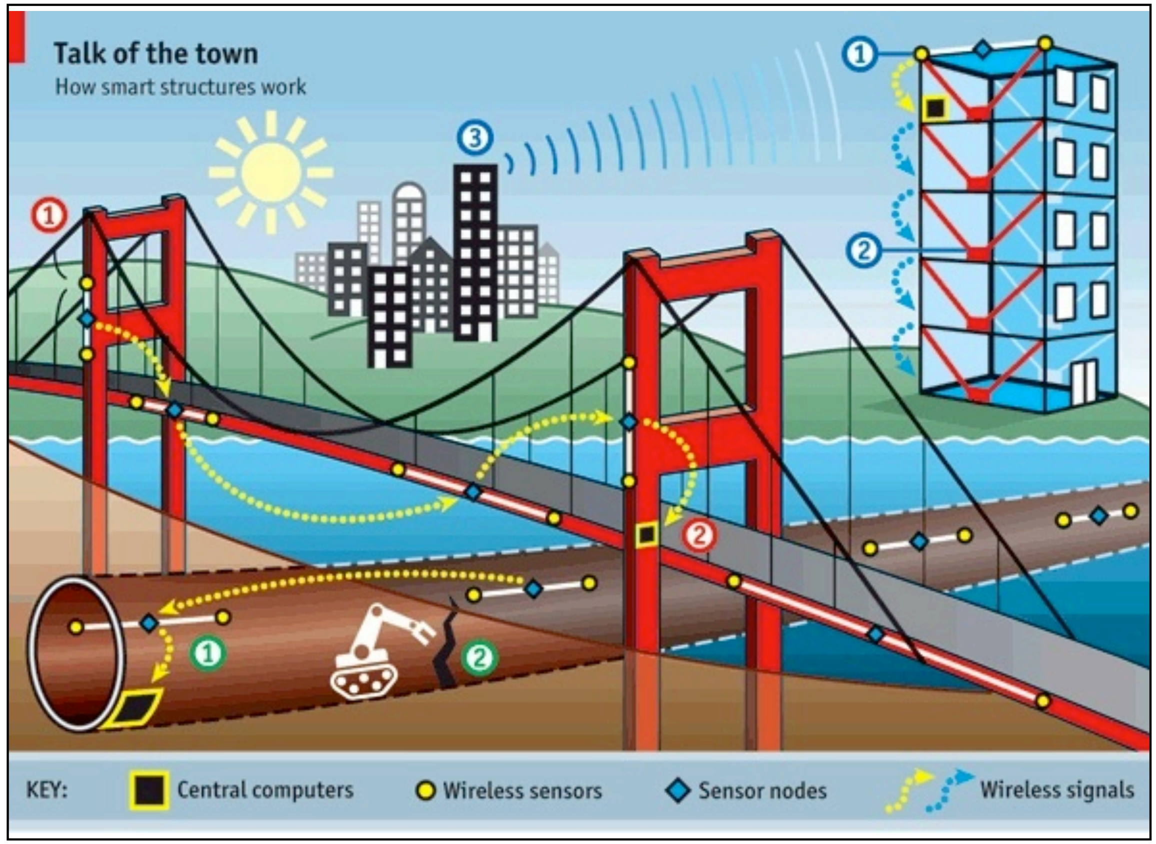
IoT Smart Power Distribution Grid



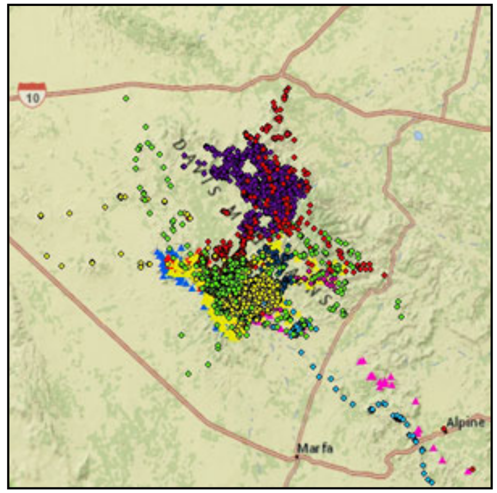


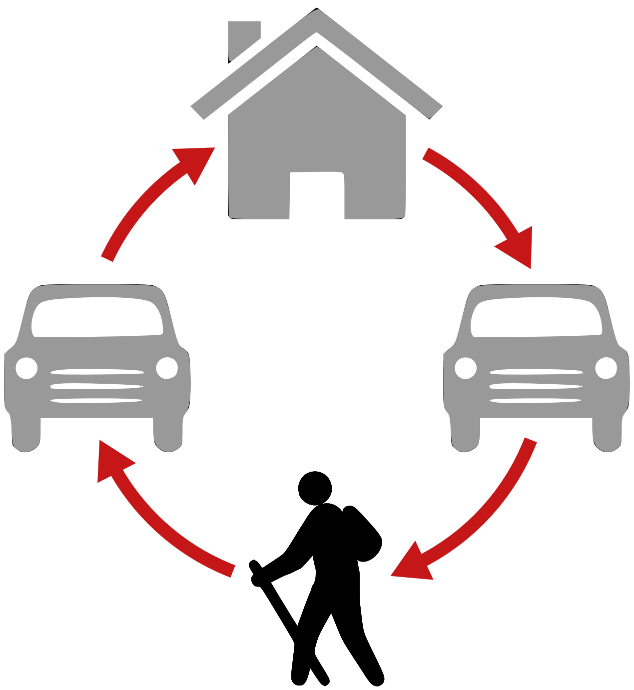
IoT Early Disaster Warning System



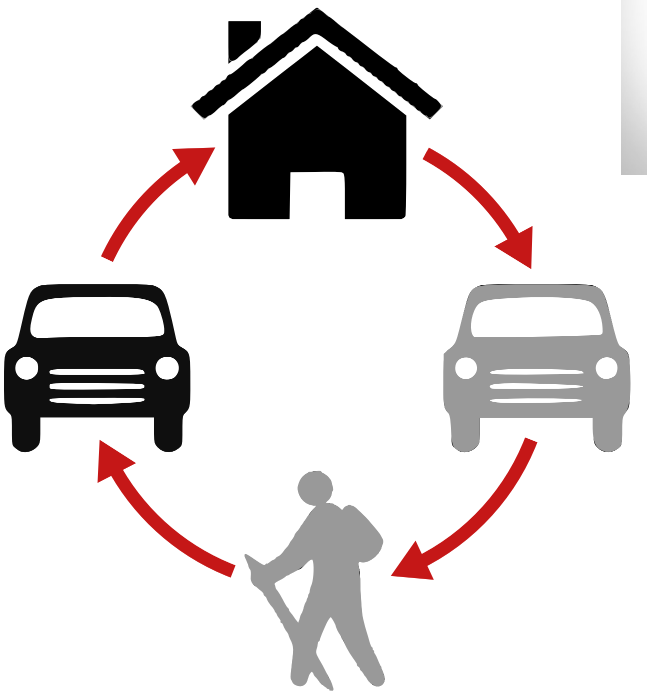


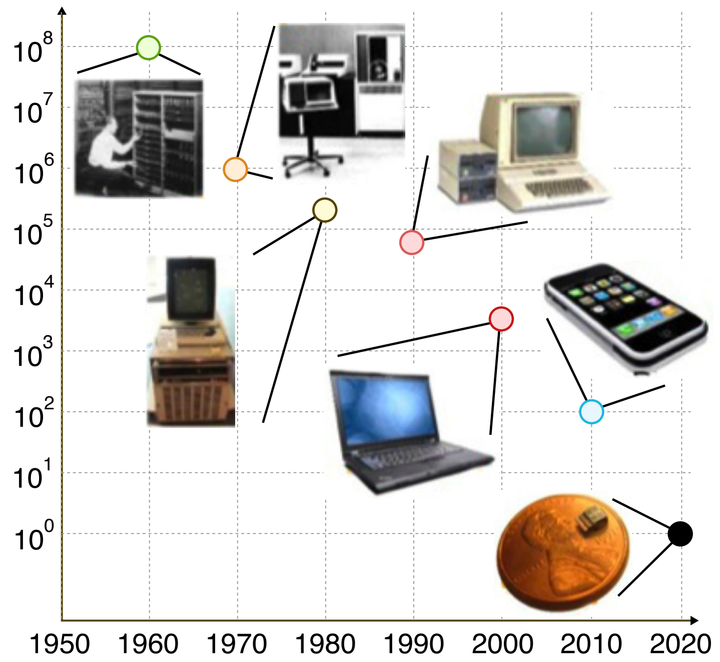
IoT Wildlife Tracking System





IoT Wearable Health Monitor





Trend #1: Bell's "Law"

Bell's "Law" predicts an Internet-of-Things, and IoT cloud and embedded devices are increasingly demanding 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.

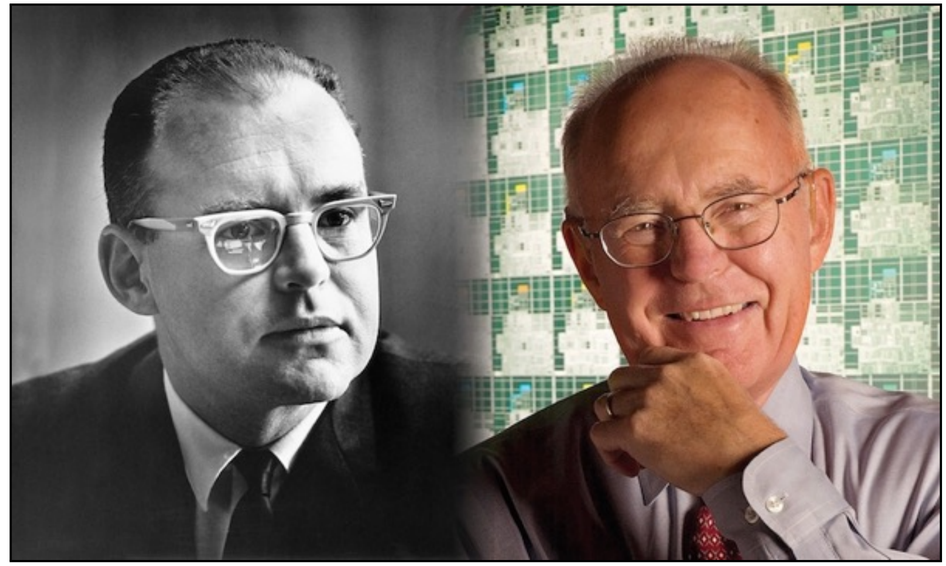
The advocates of semiconductor integrated circuitry are already using the improved characteristics of thin-film resistors by applying such films directly to an active semiconductor substrate. Those advocating a technology based upon films are developing sophisticated techniques for the attachment of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

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

The author

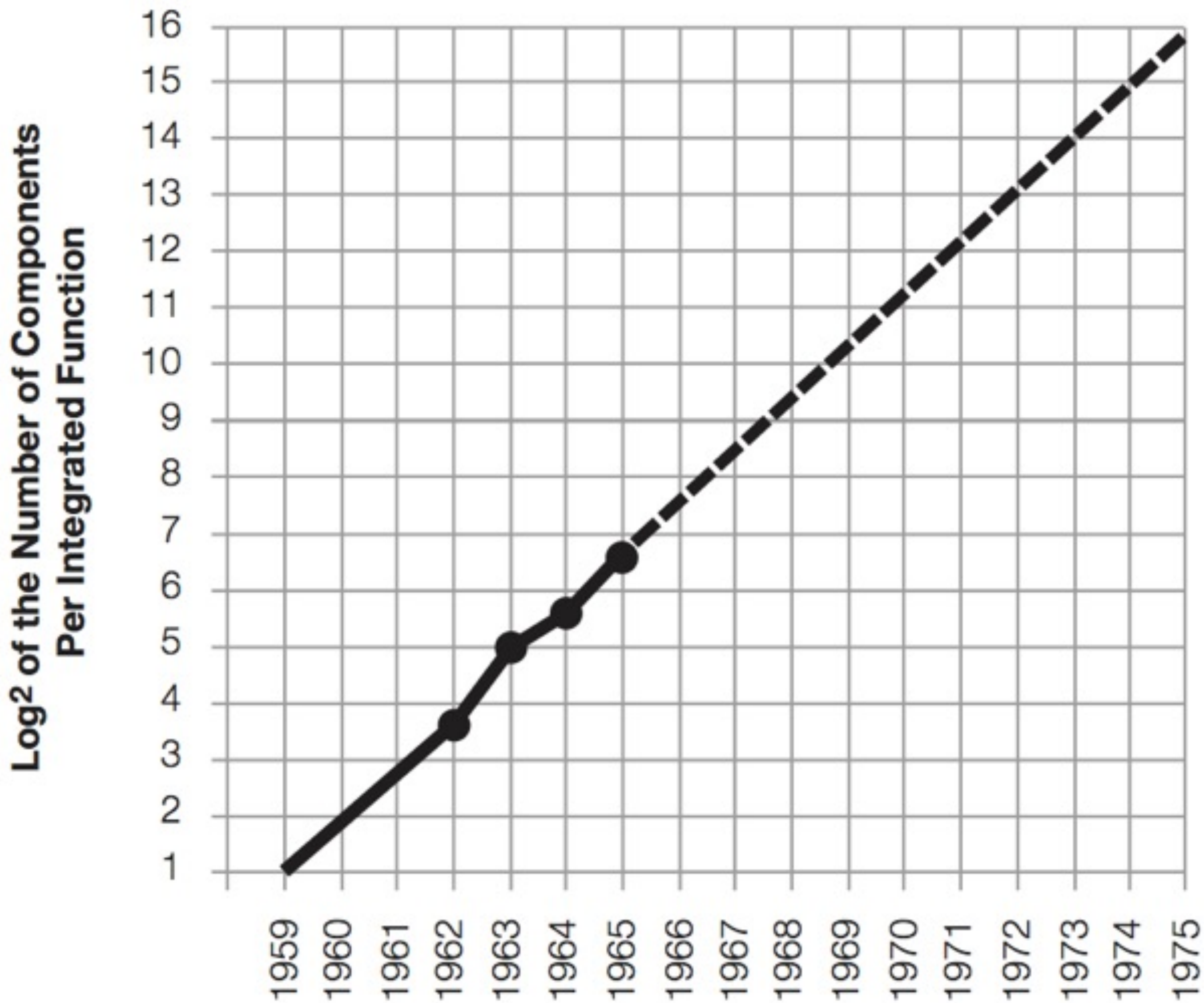
Dr. Gordon E. Moore is one of the new breed of electronic engineers, schooled in the physical sciences rather than in electronics. He earned a B.S. degree in chemistry from the University of California and a Ph.D. degree in physical chemistry from the California Institute of Technology. He was one of 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

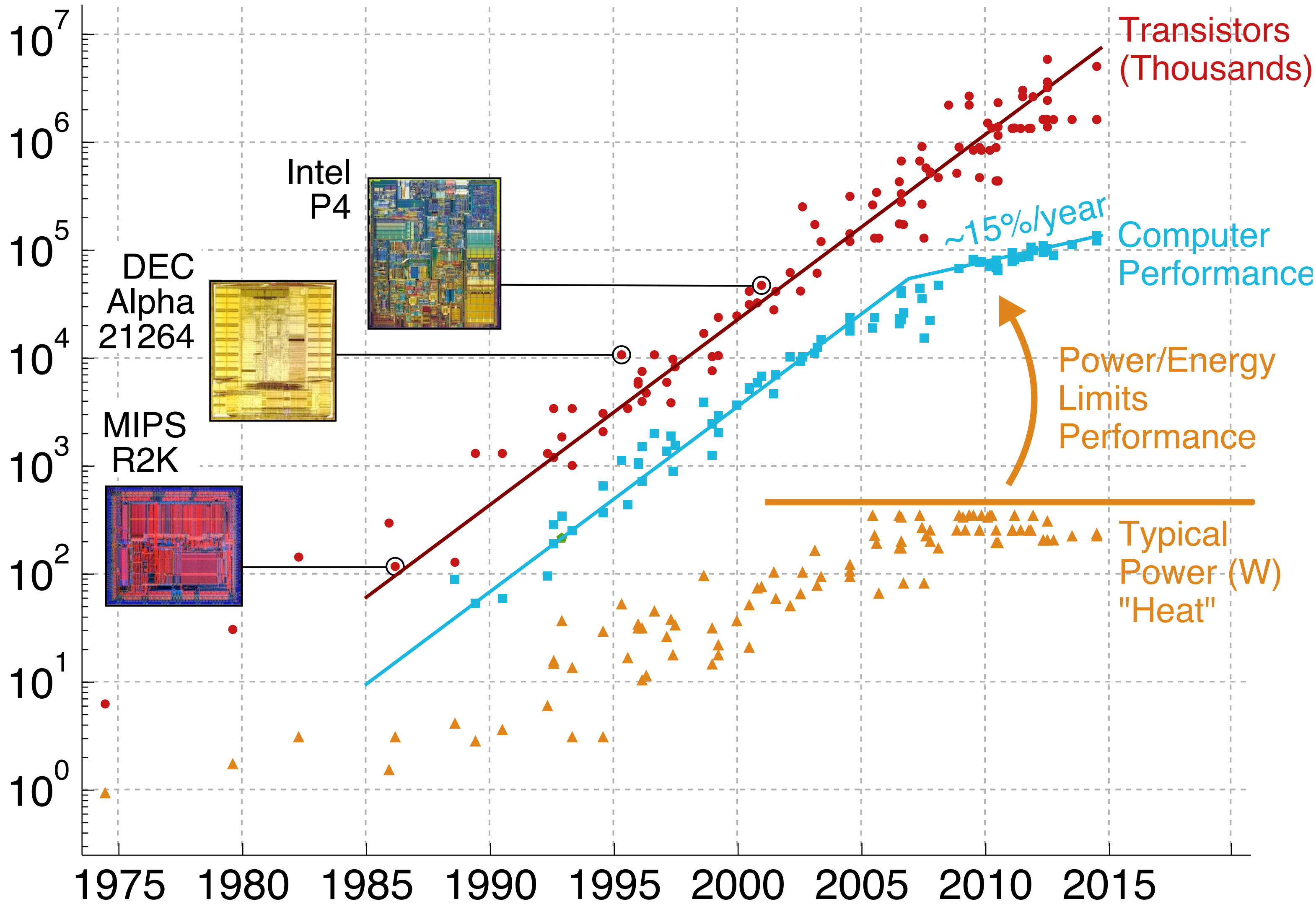


- Co-founder of Fairchild Semiconductor
- Co-founder of Intel Corp
- 1965 paper in Electronics Magazine

Gordon Moore's "Law" of Technology Scaling

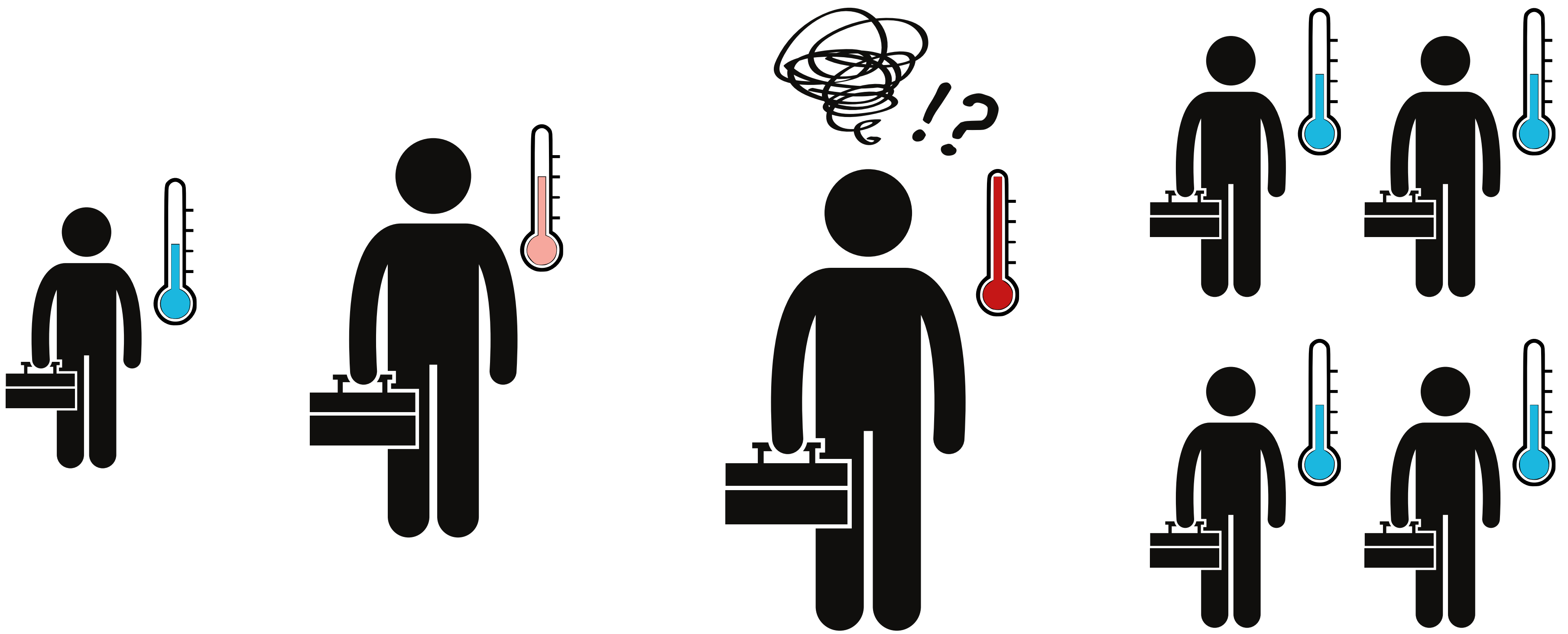


Trend #2: Moore's "Law"



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

One way to address the power challenge



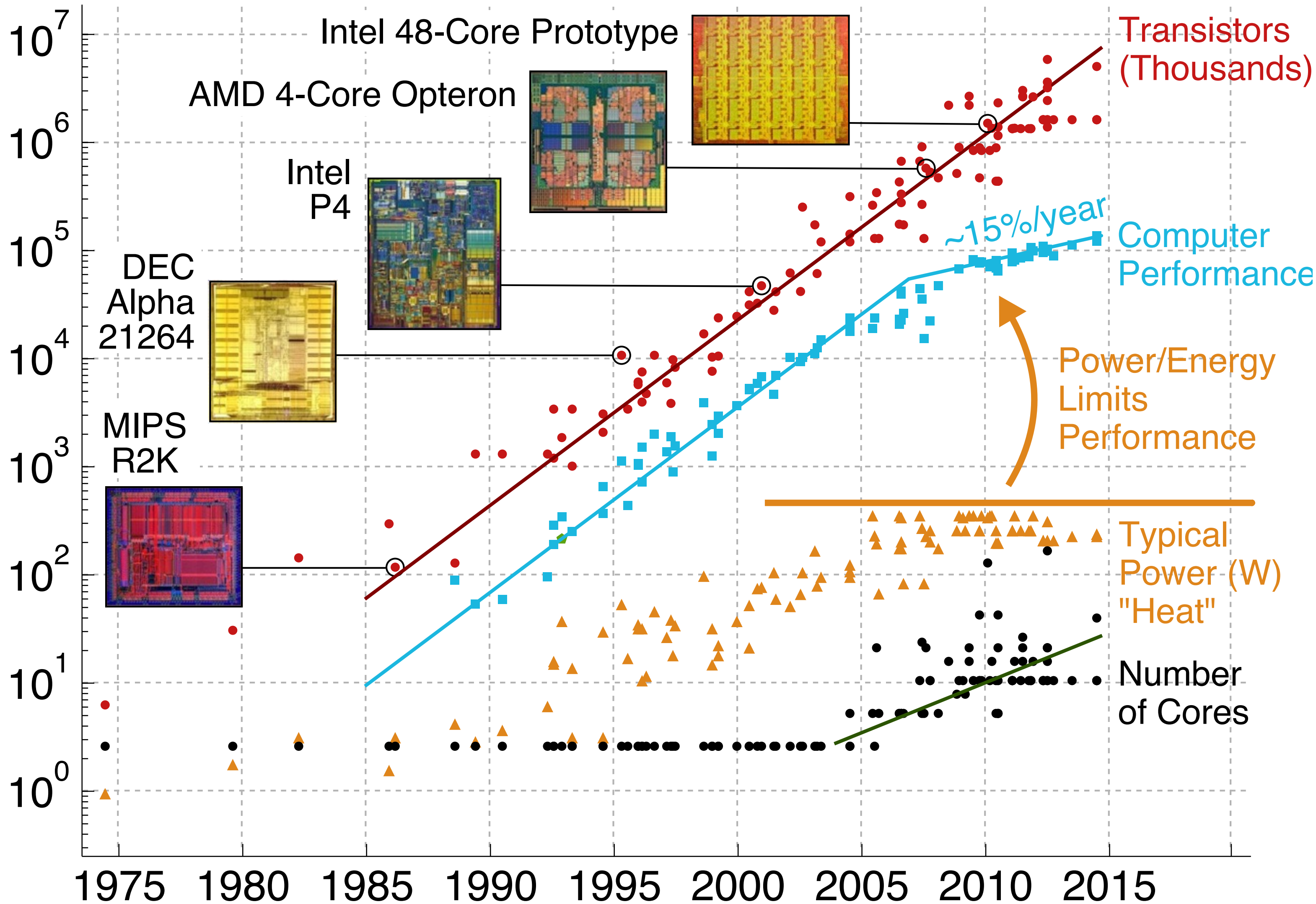
1980
single
simple
processor

1990

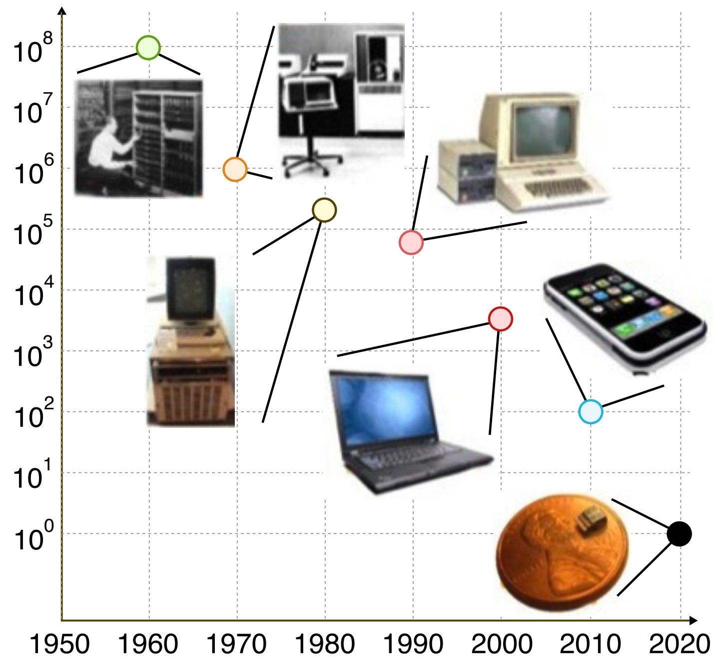
2000
single
complex
processor

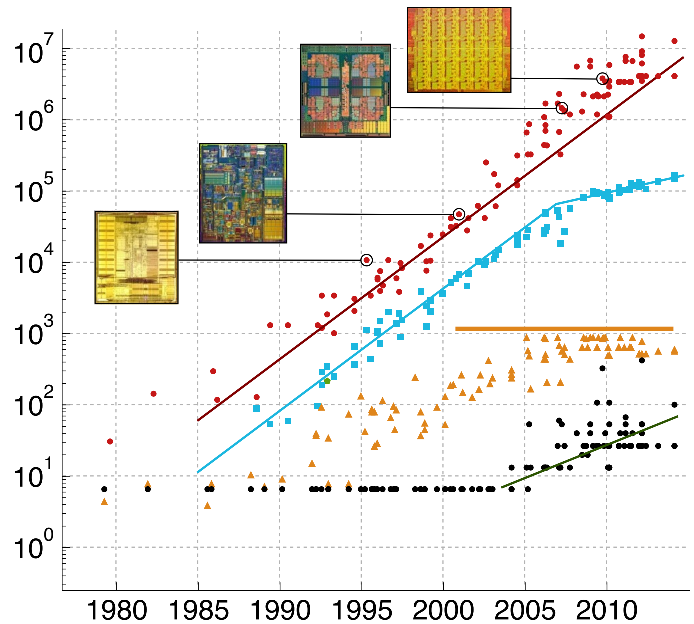
2010
multiple
simple
processors

Transition to Multicore



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





Trend #1: Bell's "Law"

Bell's "Law" predicts an Internet-of-Things, and IoT cloud and embedded devices are increasingly demanding 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 devices?

Application
Algorithm
PL
OS
ISA
µArch
RTL
Gates
Circuits
Devices
Technology

Agenda

What is Computer Engineering?

Trends in Computer Engineering

- Trend #1: Bell's "Law"
- Trend #2: Moore's "Law"

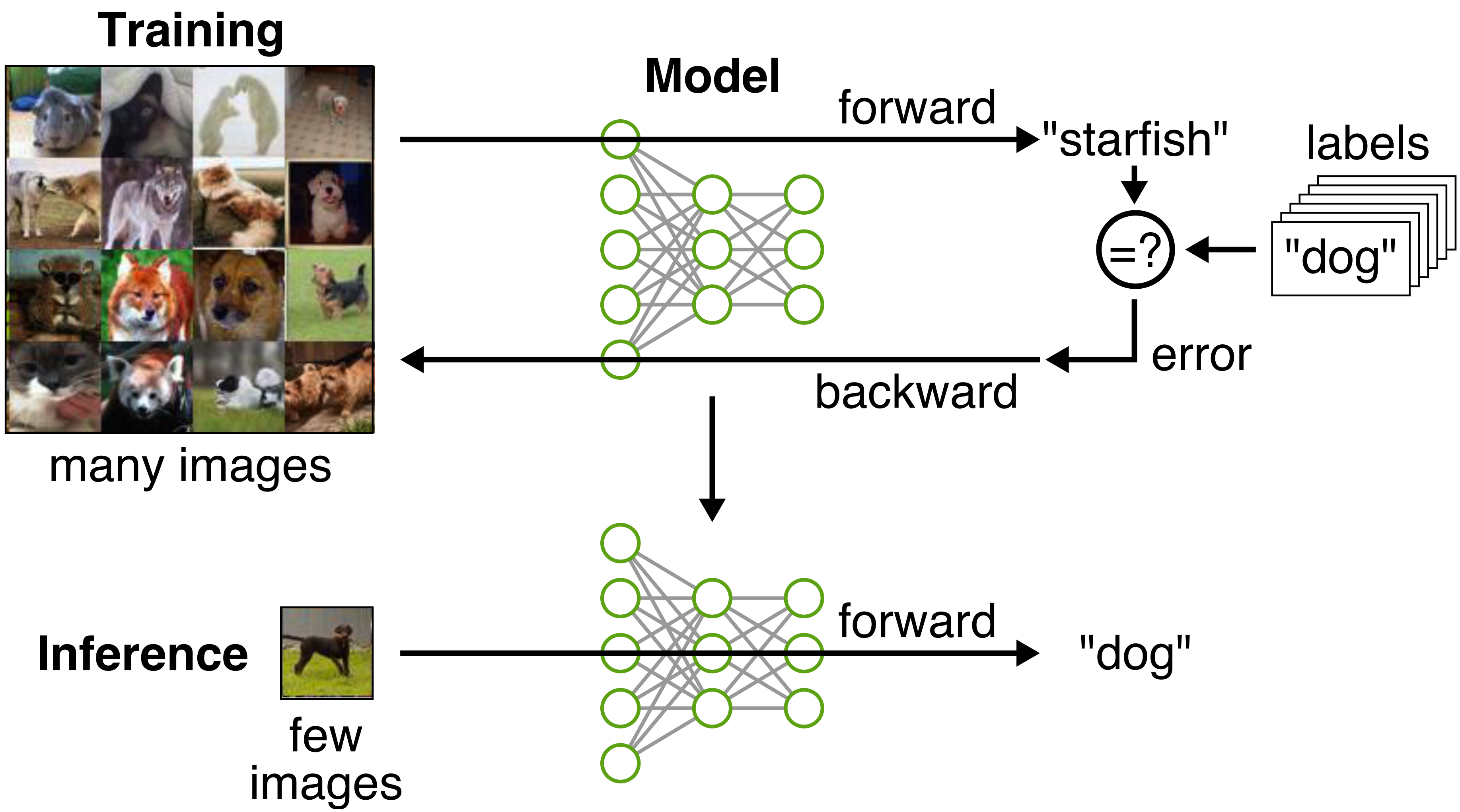
Specialized Computer Systems

General Purpose

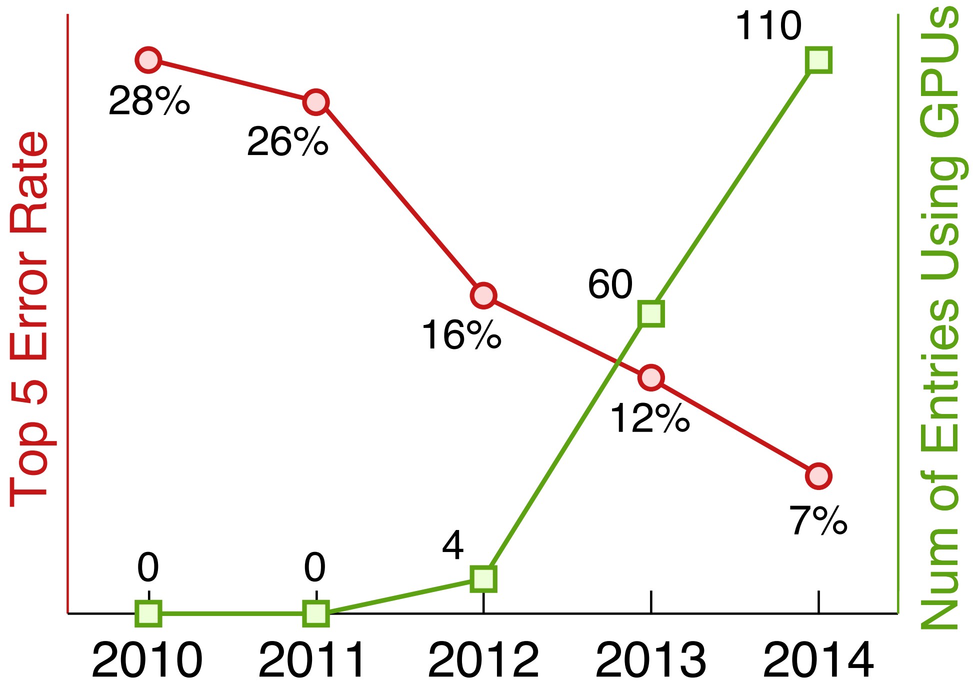
Specialized

Example Application Domain: Image Recognition

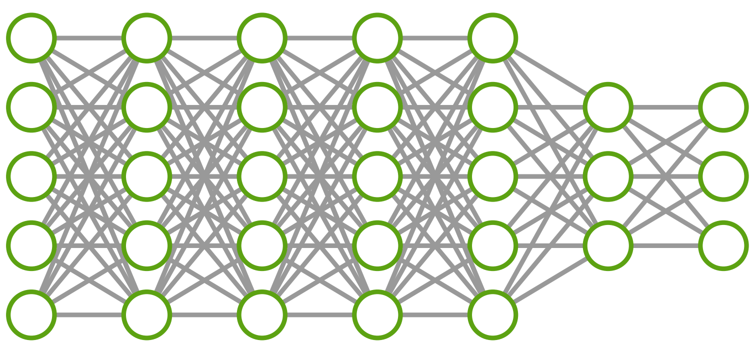
Machine Learning: Training vs. Inference



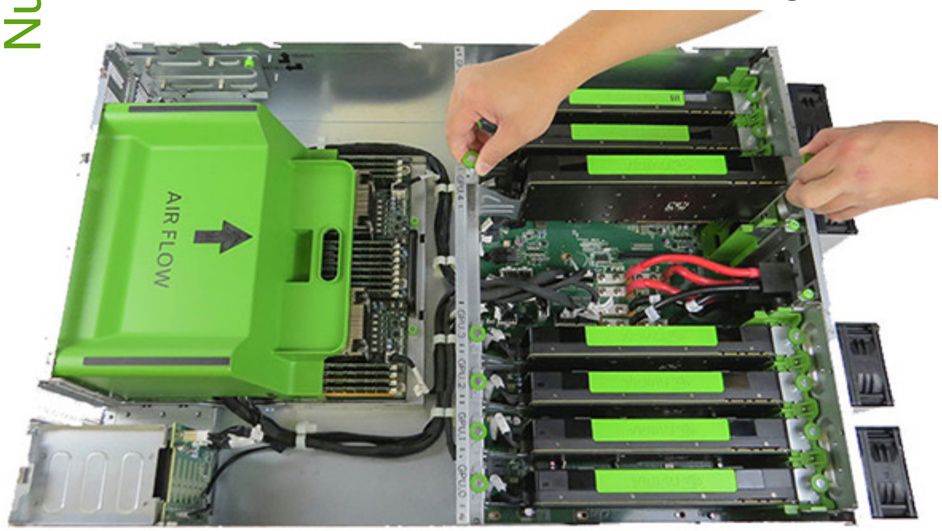
ImageNet Large-Scale Visual Recognition Challenge



Hardware: Graphics Processing Units

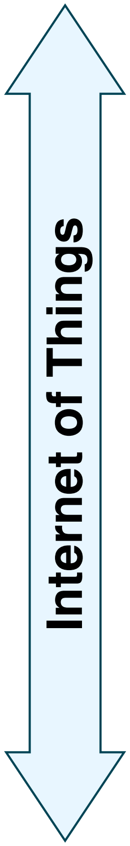


Software: Deep Neural Network

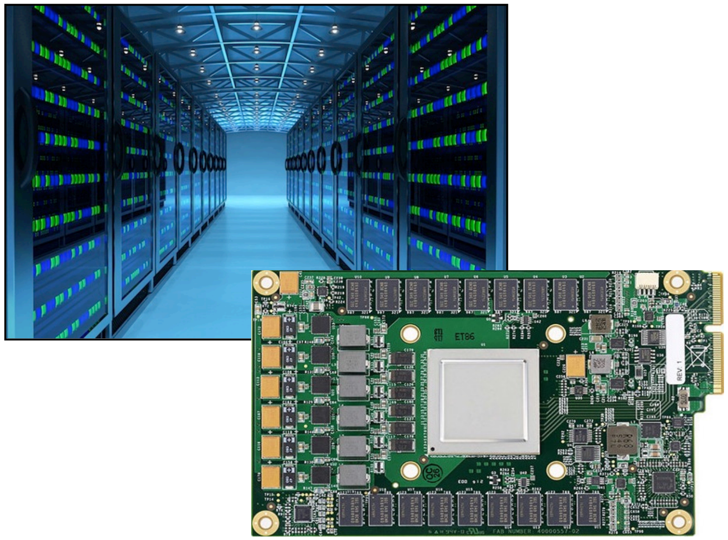


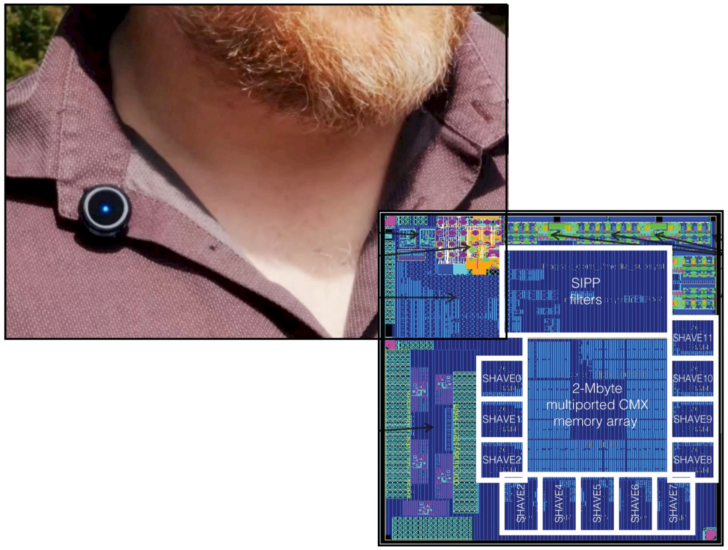
Hardware Specialization from Cloud to Embedded

Cloud Computing



Ultra-Embedded Computing





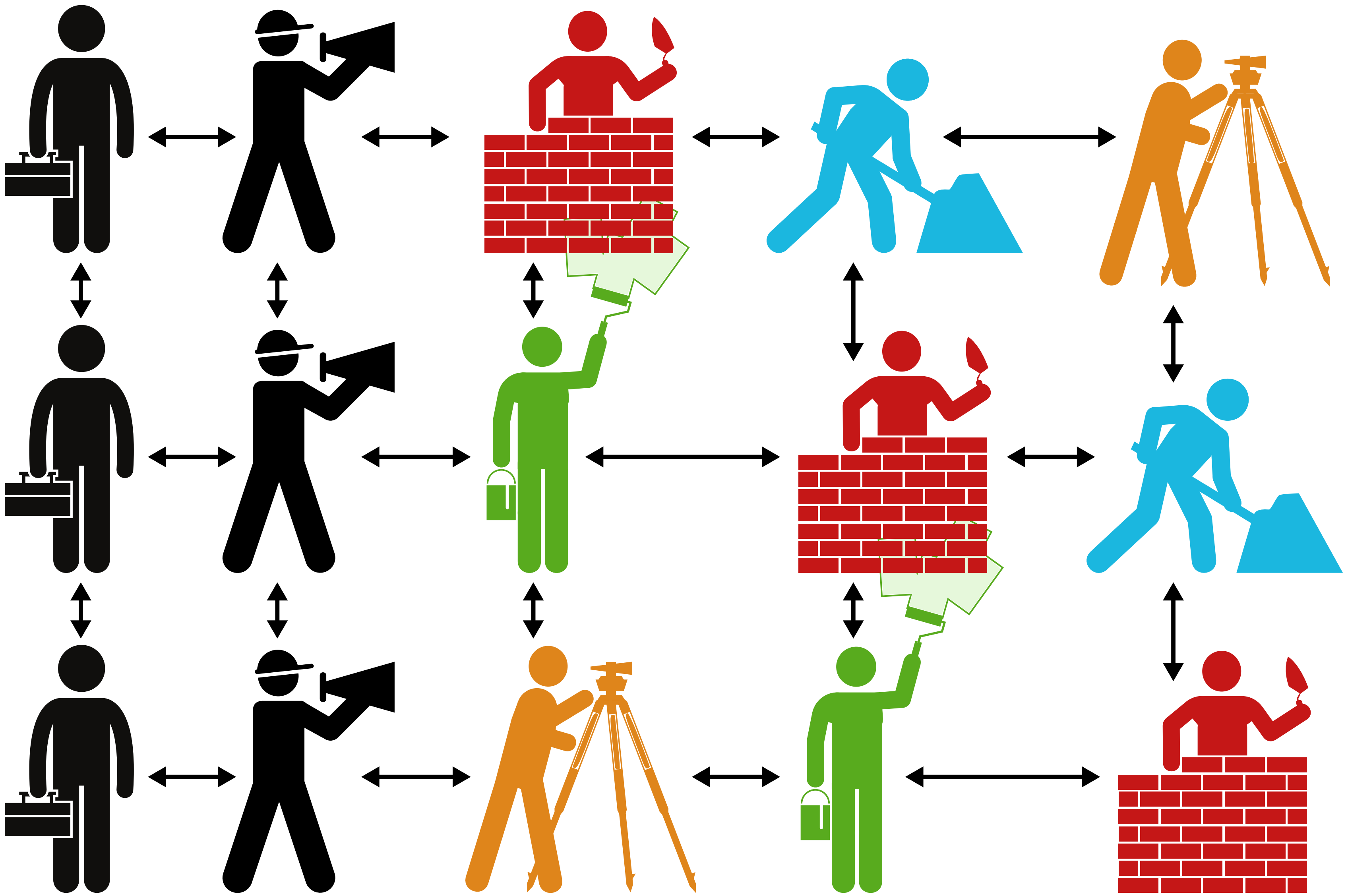
Google TPU

- Training can take weeks
- Inference has strict speed requirements
- Google TPU is custom chip to accelerate training and inference

Movidius Myriad 2

- Vision processing requires significant computation
- Can easily drain the battery of embedded IoT devices
- Myriad 2 is custom chip to accelerate machine learning

The Future of Computer Chips



The Celerity Open-Source 511-Core RISC-V Tiered Accelerator Fabric: Fast Architectures & Design Methodologies for Fast Chips

Scott Davidson, Shaolin Xie, Christopher Torng, Khalid Al-Hawaj
Austin Rovinski, Tutu Ajayi, Luis Vega, Chun Zhao, Ritchie Zhao
Steve Dai, Aporva Amarnath, Bandhav Veluri, Paul Gao, Anuj Rao
Gai Liu, Rajesh K. Gupta, Zhiru Zhang, Ronald G. Dreslinski
Christopher Batten, Michael B. Taylor.

IEEE Micro, 38(2):3041, Mar/Apr. 2018

CORNELL UNIVERSITY FOUNDED A.D. 1865

UCSD

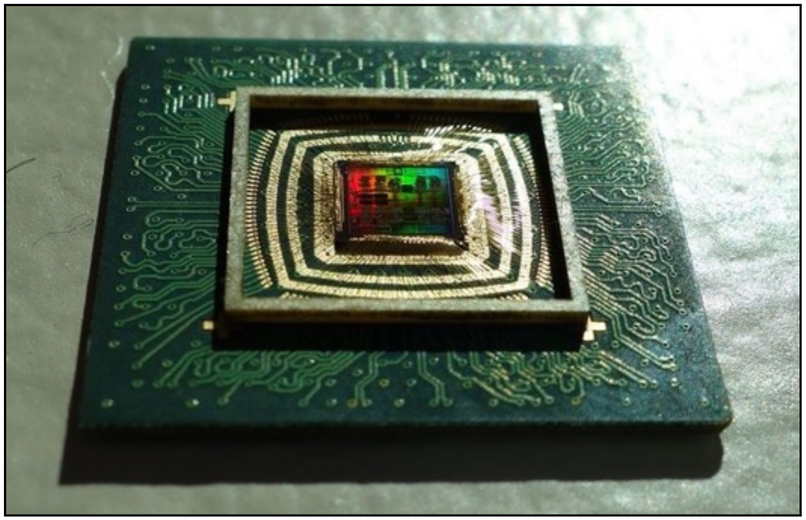
UNIVERSITY of WASHINGTON

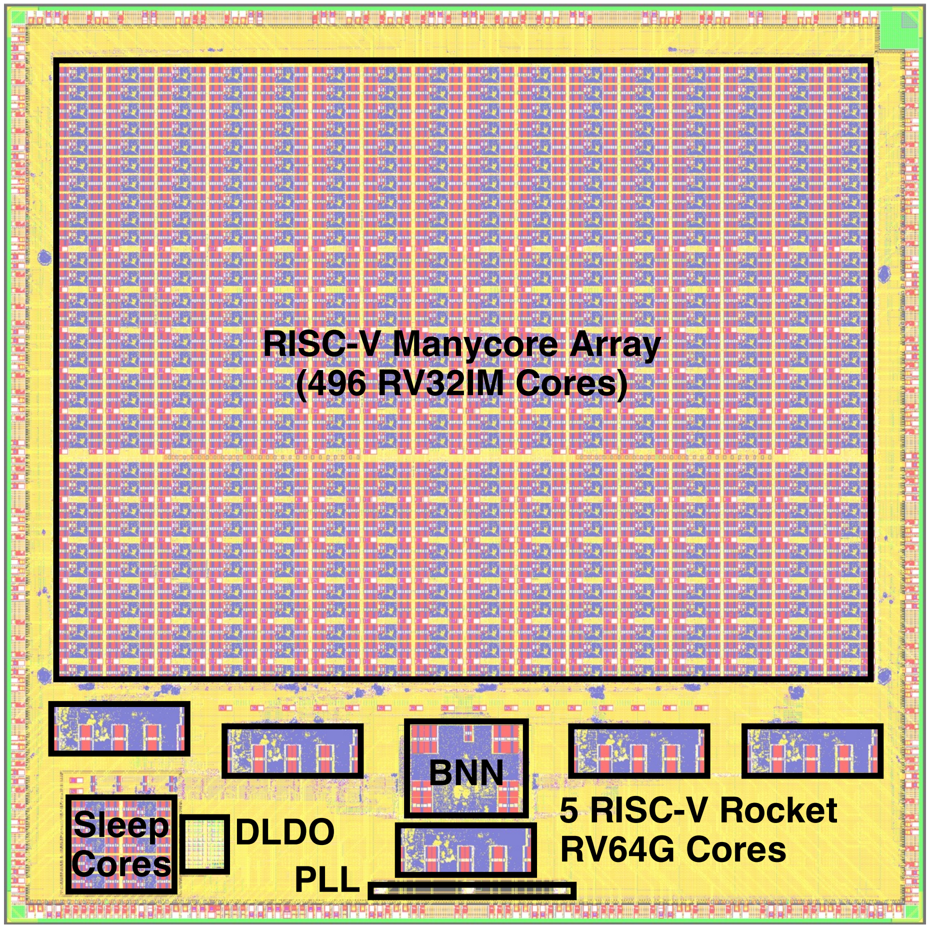
UNIVERSITY OF MICHIGAN

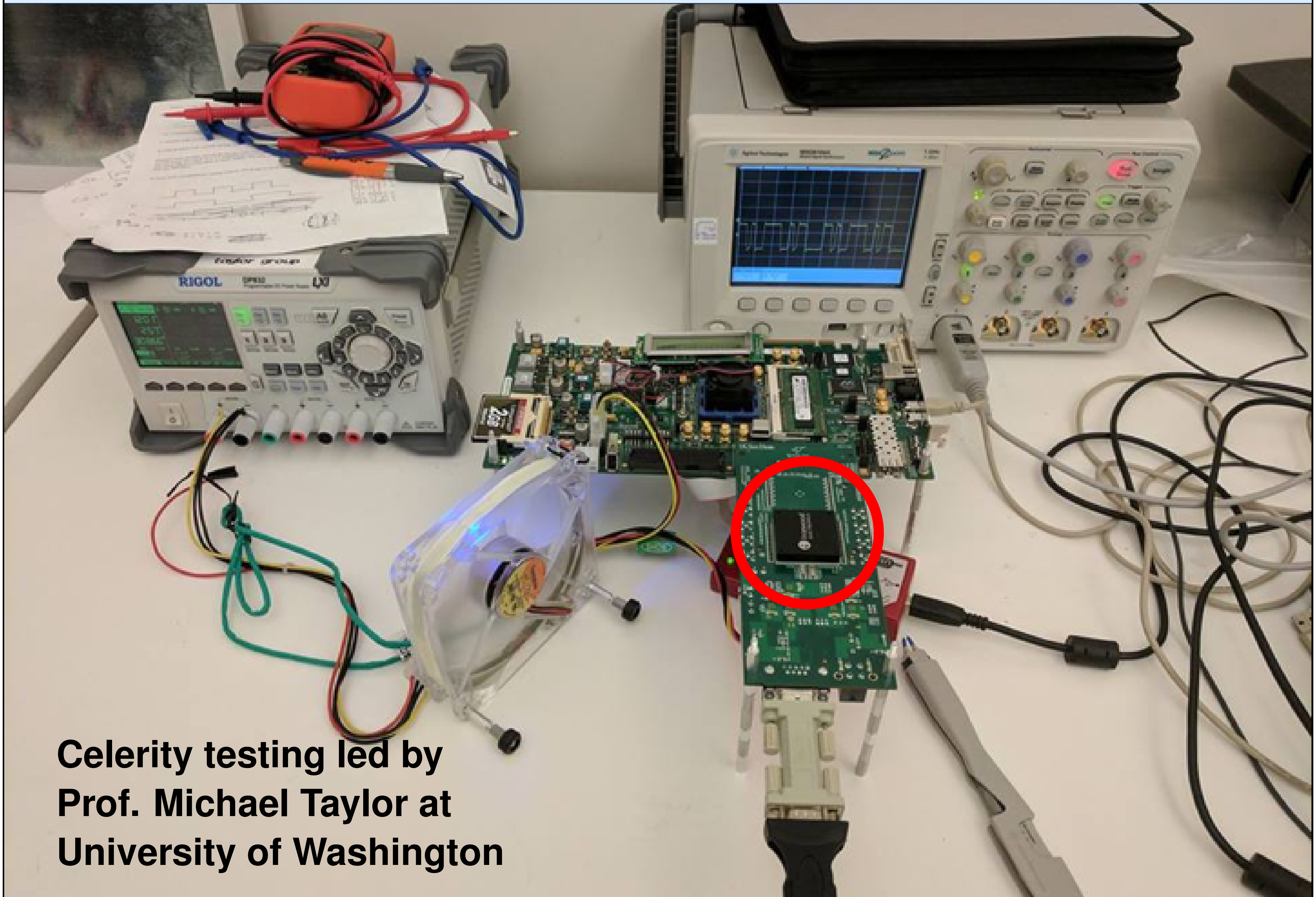
Celerity System-on-Chip Overview

Target Workload: High-Performance Embedded Computing

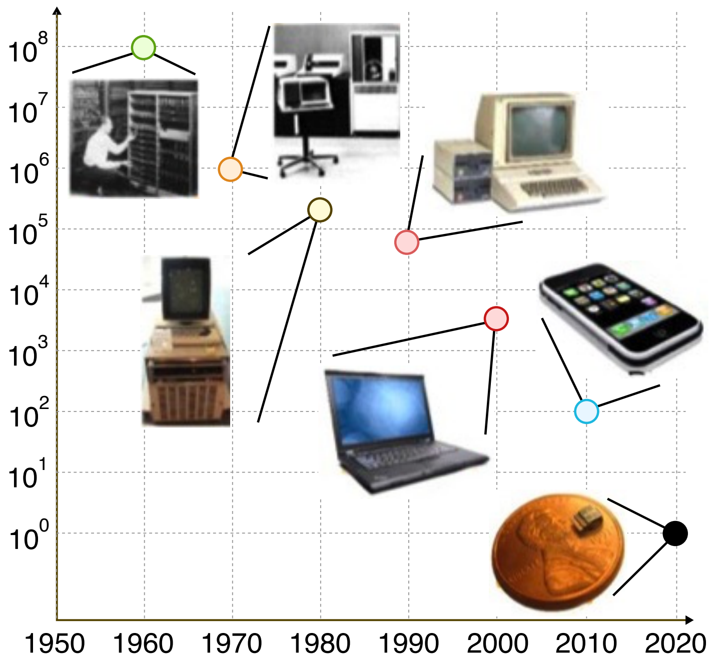
- 5×5mm in TSMC 16 nm
- 385 million transistors
- 5 "large" processing cores
- 496 "small" processing cores
- 1 neural network accelerator for machine learning

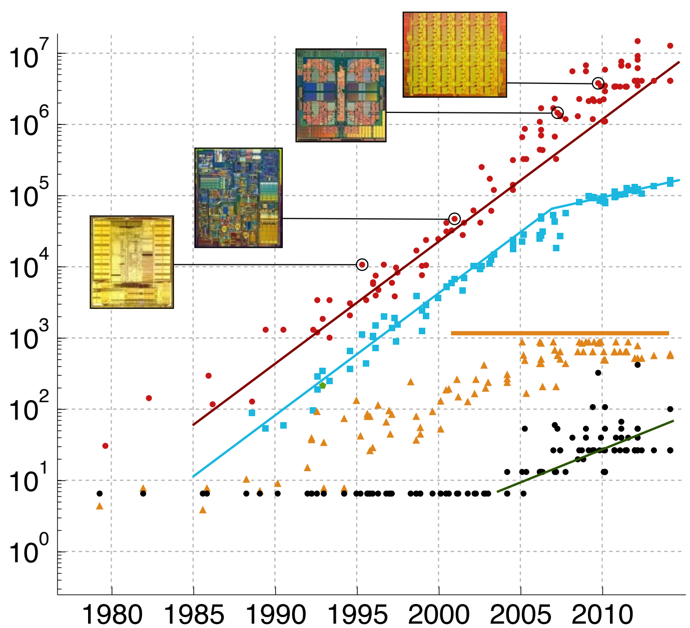






Celerity testing led by Prof. Michael Taylor at University of Washington





Trend #1: Bell's "Law"

Bell's "Law" predicts an Internet-of-Things, and IoT cloud and embedded devices are increasingly demanding 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

Specialized Computing Systems

Hardware specialization can use the wealth of transistors to meet the needs of IoT

Shreesha Srinath, Christopher Torng, Berkin Ilbeyi, Moyang Wang
Shunning Jiang, Khalid Al-Hawaj, Tuan Ta, Lin Cheng
and many M.S./B.S. students

Equipment, Tools, and IP
Intel, NVIDIA, Synopsys, Cadence, Xilinx, ARM