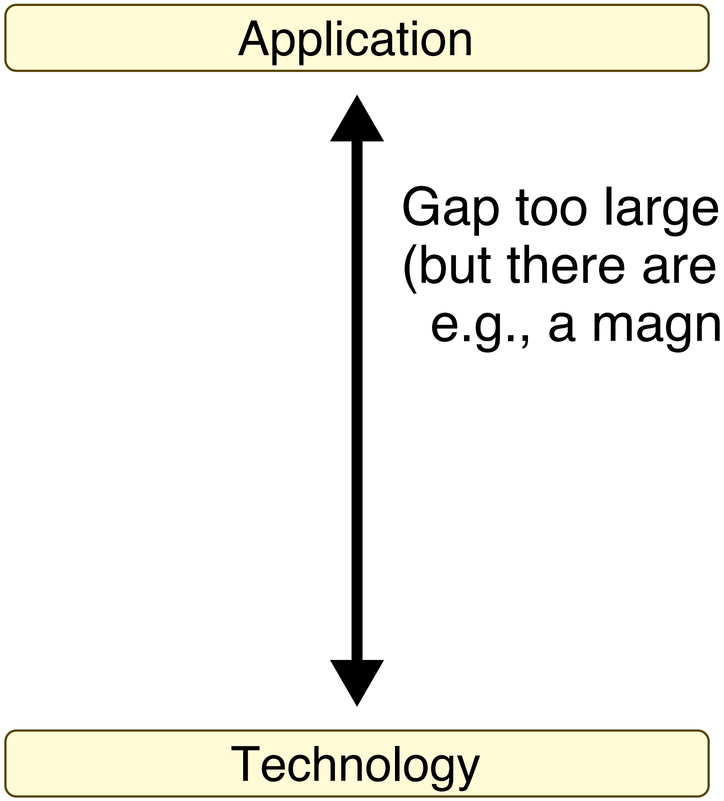
# ECE 4750 Computer Architecture
# Course Overview

Christopher Batten

School of Electrical and Computer Engineering
Cornell University

http://www.csl.cornell.edu/courses/ece4750

# The Computer Systems Stack

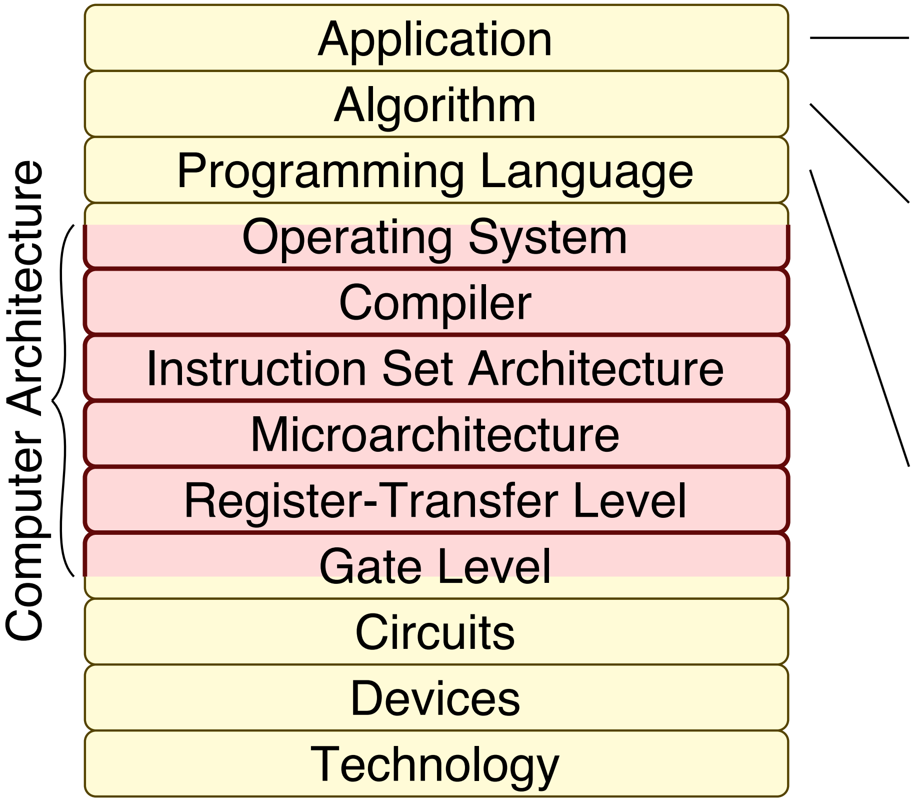


Application

Technology

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

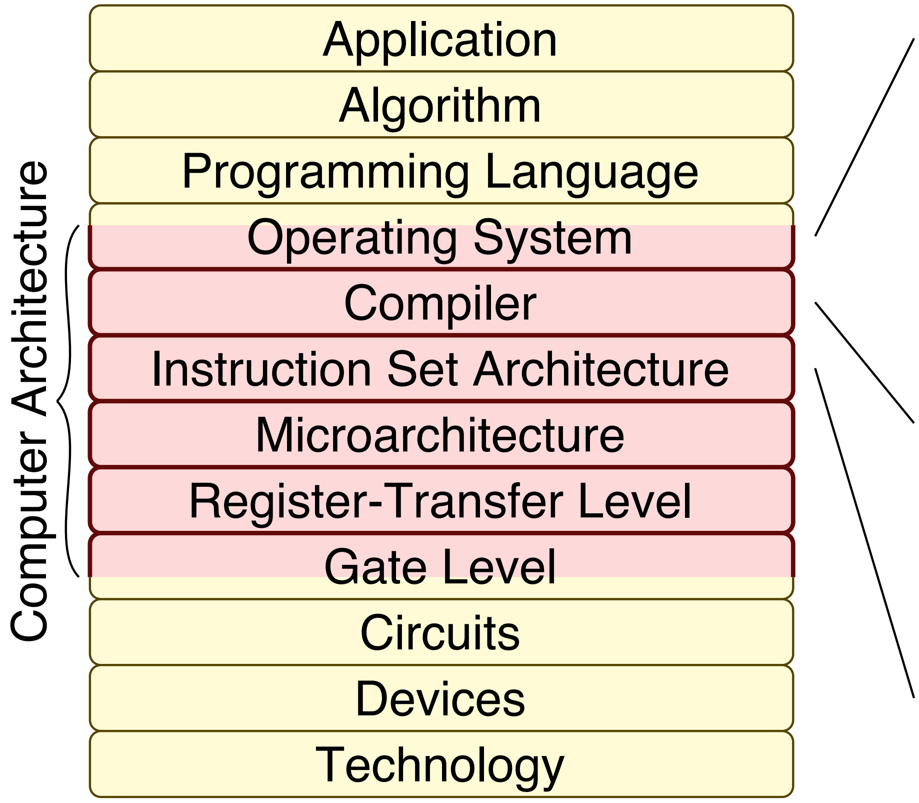
**Out-of-place selection sort algorithm**

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

**C Compiler**

Transform programs into assembly

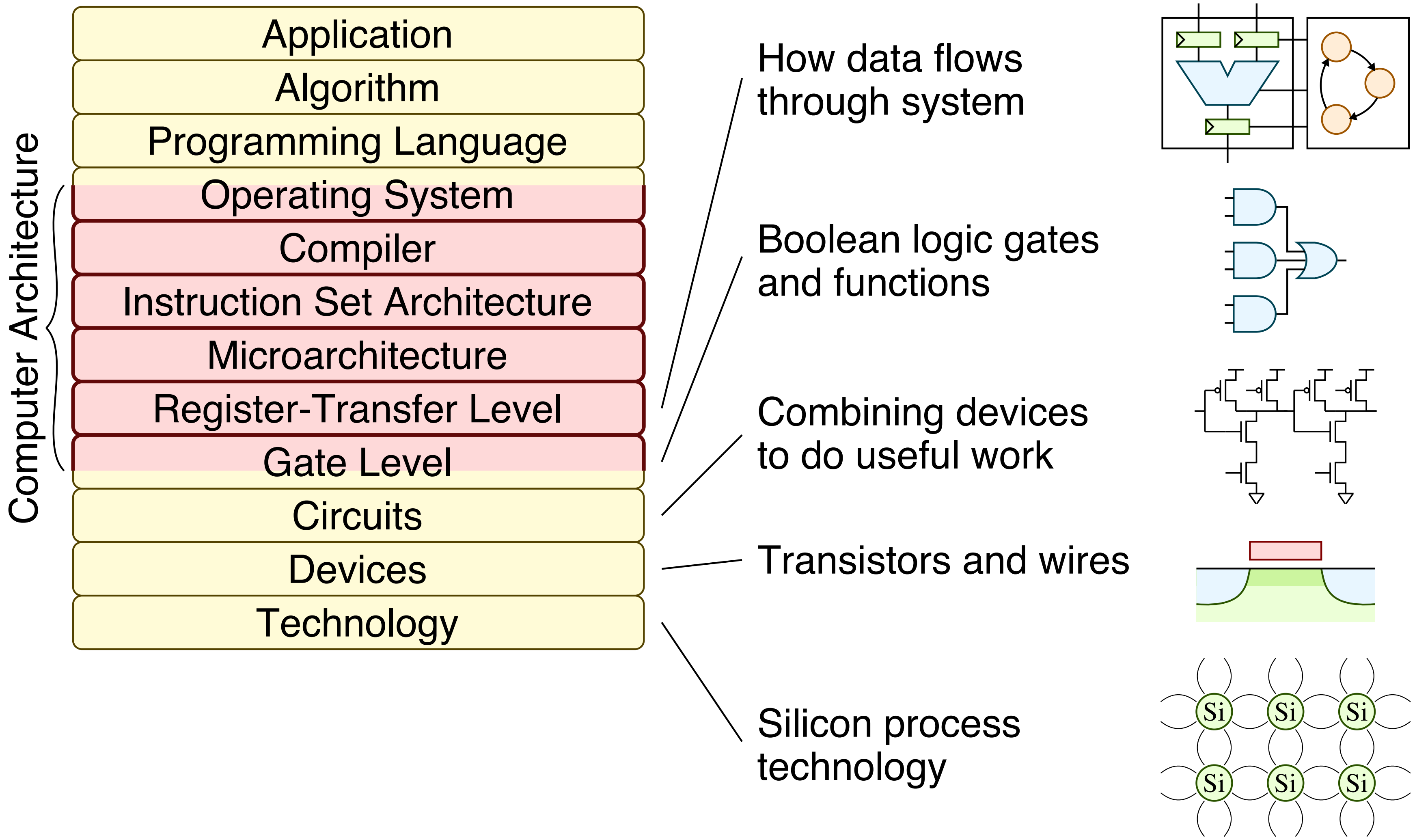
```
int a = b + c;   →   add r1, r2, r3
A[i] = a;            sw  r1, 0(r4)
```

**RISC-V Instruction Set**

Instructions that machine executes

```
blez r2, done
addi r7, r0, 0
addi r1, r0, 99
addi r4, r1, 0
addi r3, r0, 99
lw   r5, 0(r4)
```

# The Computer Systems Stack



Application

Algorithm

Programming Language

Operating System

Compiler

Instruction Set Architecture

Microarchitecture

Register-Transfer Level

Gate Level

Circuits

Devices

Technology

Computer Architecture

How data flows through system

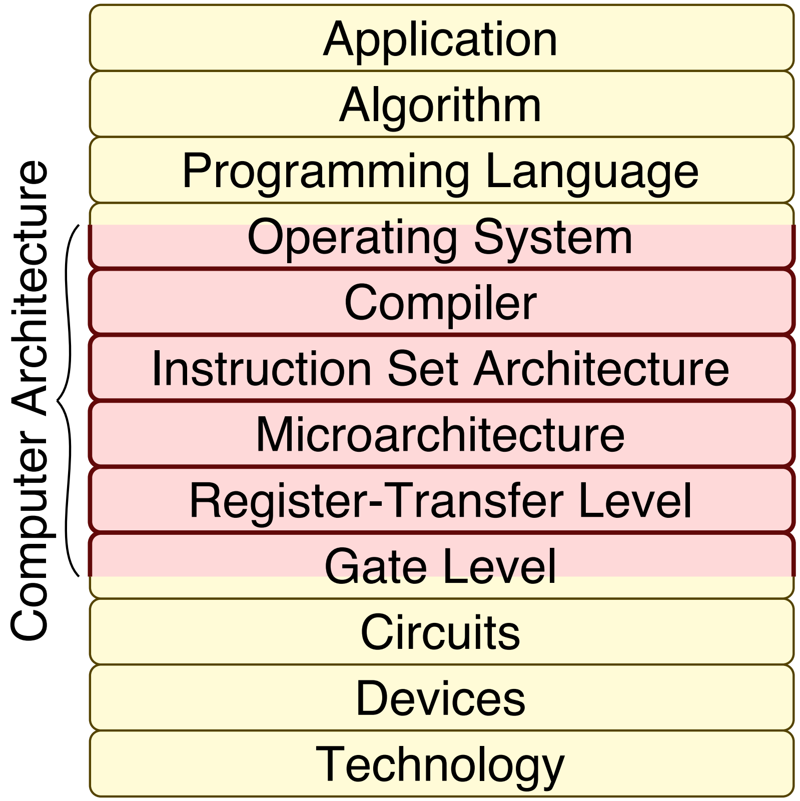
Boolean logic gates and functions

Combining devices to do useful work

Transistors and wires

Silicon process technology

# Application Requirements ↔ Technology Constraints



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

Computer Architecture

**Application Requirements**

- Provide motivation for building system
- SW/HW interface expressive yet productive

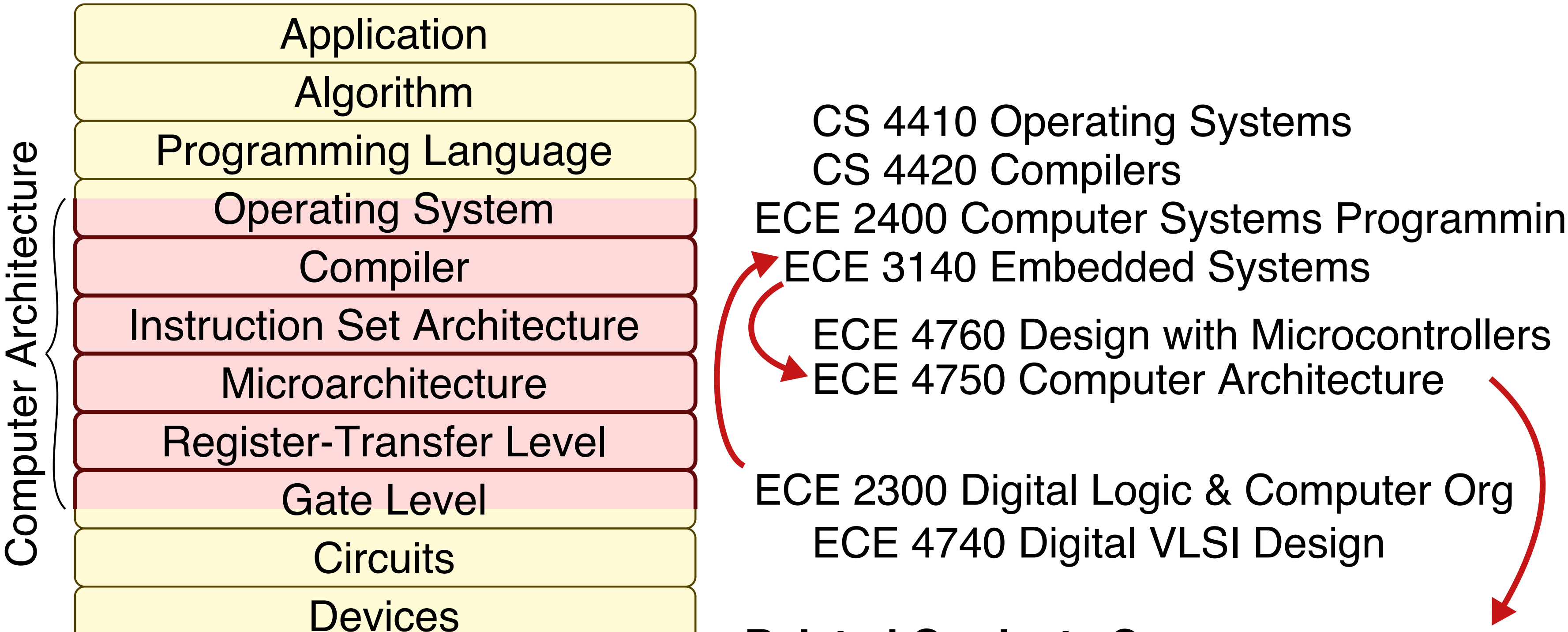
Computer architects provide feedback to guide application and technology research directions

**Technology Constraints**

- Restrict what can be done efficiently
- New technologies make new arch possible

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

# Computer Architecture in the ECE/CS Curriculum

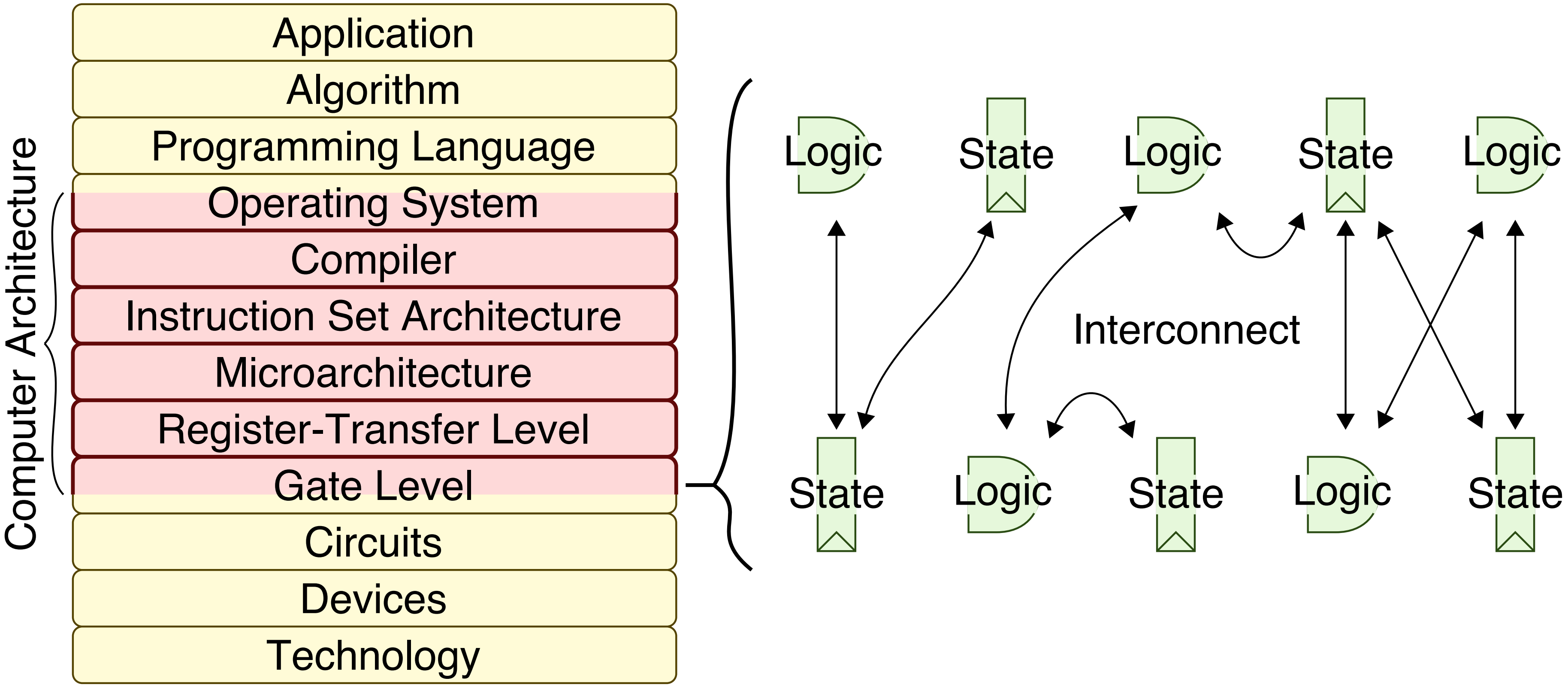


CS 4410 Operating Systems
CS 4420 Compilers
ECE 2400 Computer Systems Programming
ECE 3140 Embedded Systems
ECE 4760 Design with Microcontrollers
ECE 4750 Computer Architecture
ECE 2300 Digital Logic & Computer Org
ECE 4740 Digital VLSI Design

**Related Graduate Courses**

- ECE 5760 Advanced Microcontroller Design
- ECE 5750 Advanced Computer Architecture
- ECE 5745 Complex Digital ASIC Design
- ECE 5775 High-Level Design Automation

# Logic, State, and Interconnect



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

Computer Architecture

Logic State Logic State Logic

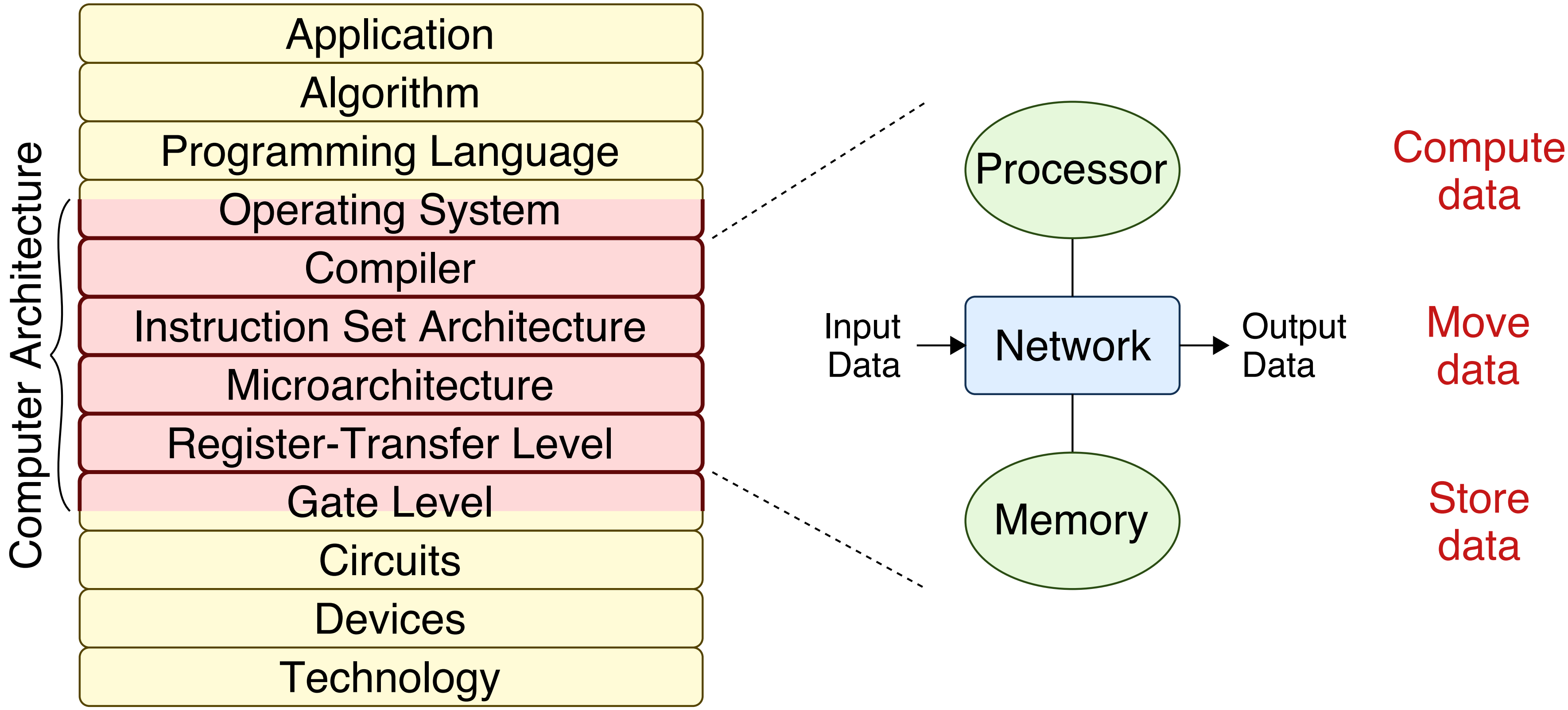
Interconnect

State Logic State Logic State

Digital logic basic building blocks

- Logic to process data
- State to store data
- Interconnect to move data

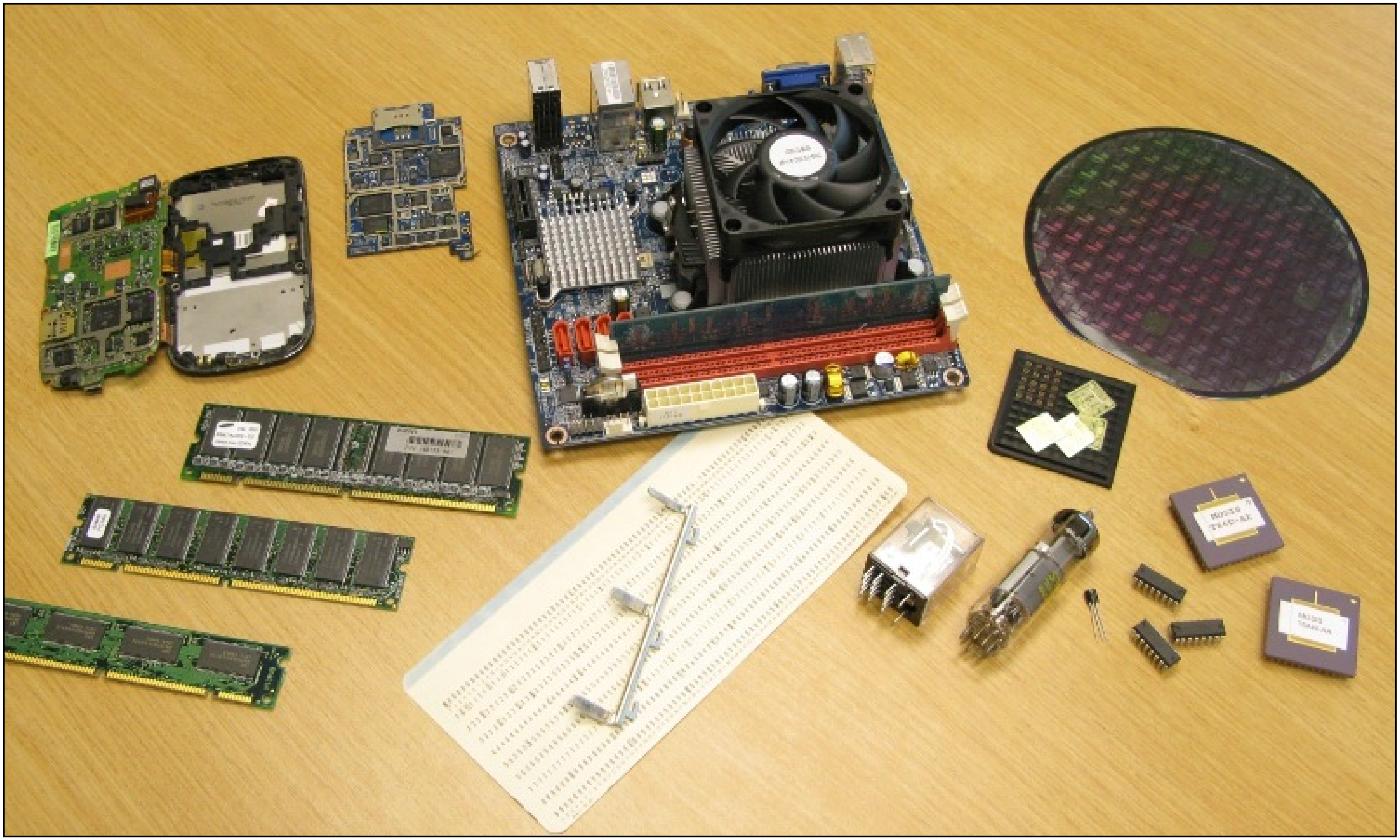
# Processors, Memories, and Networks



Computer architecture basic building blocks

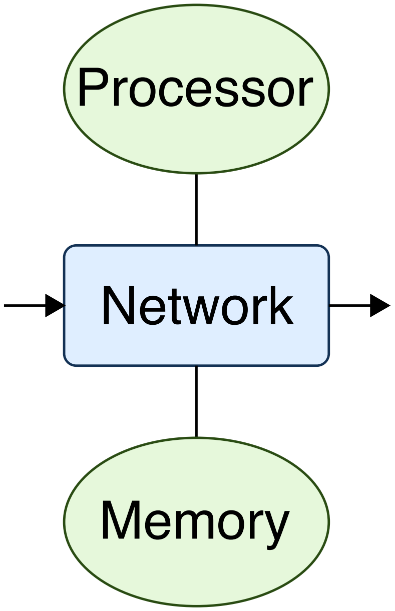
- Processors for computation
- Memories for storage
- Networks for communication

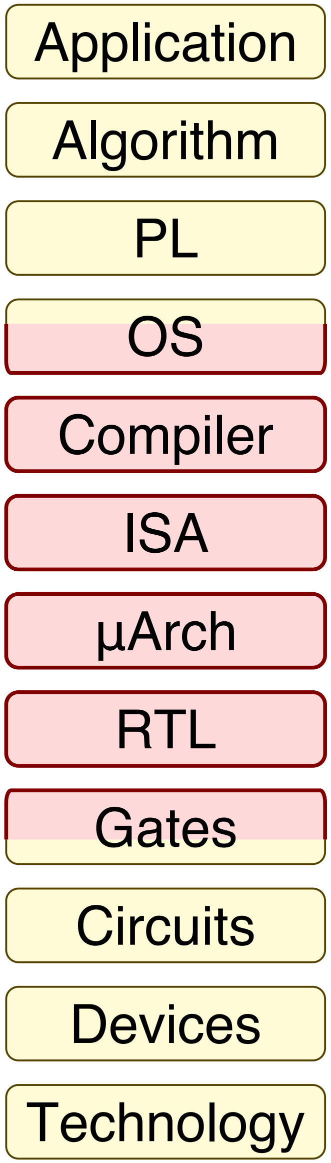
# Computer Architecture Artifacts



# Activity #1: Sorting with a Sequential Processor

- **Application:** Sort 32 numbers
- **Simulated Sequential Computing System**
  - Processor: You!
  - Memory: Worksheet, read input data, write output data
  - Network: Passing/collecting the worksheets
- **Activity Steps**
  - 1. Discuss strategy with neighbors
  - 2. When instructor starts timer, flip over worksheet
  - 3. Sort 32 numbers as fast as possible
  - 4. Lookup when completed and write time on worksheet
  - 5. Raise hand
  - 6. When everyone is finished, then analyze data





Application

Algorithm

PL

OS

Compiler

ISA

µArch

RTL

Gates

Circuits

Devices

Technology

# Agenda

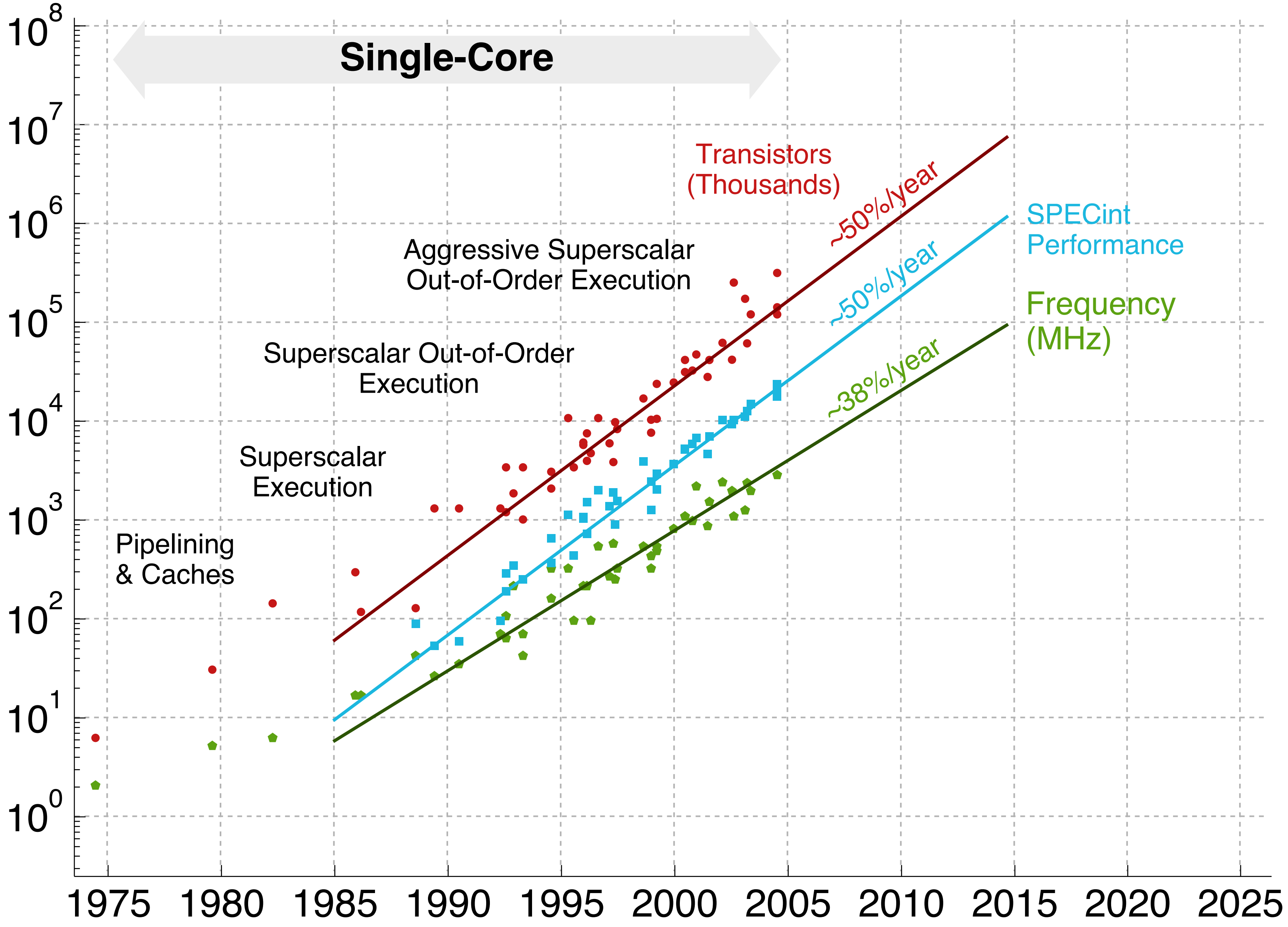
What is Computer Architecture?

Trends: Single-Core Era

Trends: Multicore-Core Era

Trends: Accelerator Era

Computer Architecture Design



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

# Energy and Power Constraints

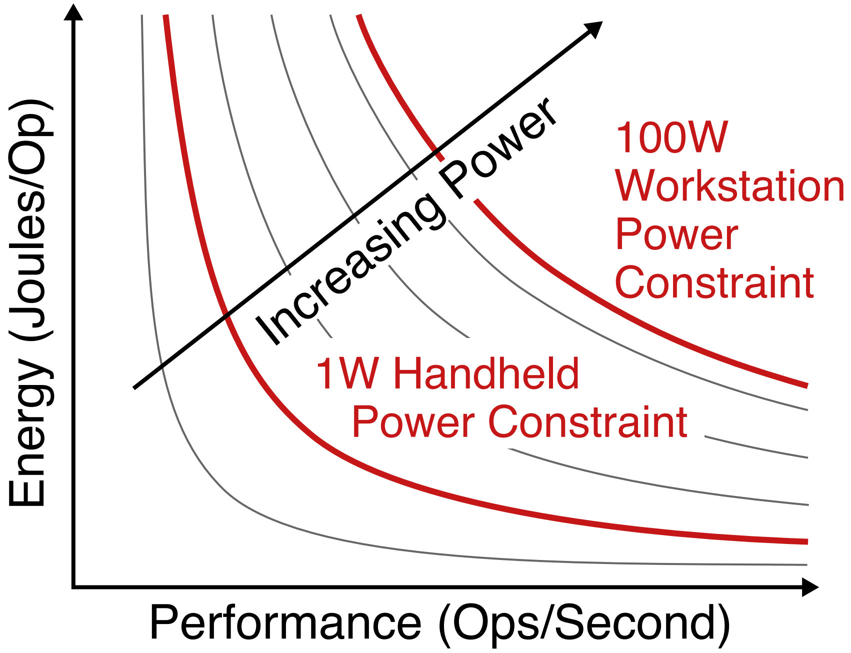
$$\text{Power} = \frac{\text{Energy}}{\text{Second}} = \frac{\text{Energy}}{\text{Op}} \times \frac{\text{Ops}}{\text{Second}}$$

**Power**

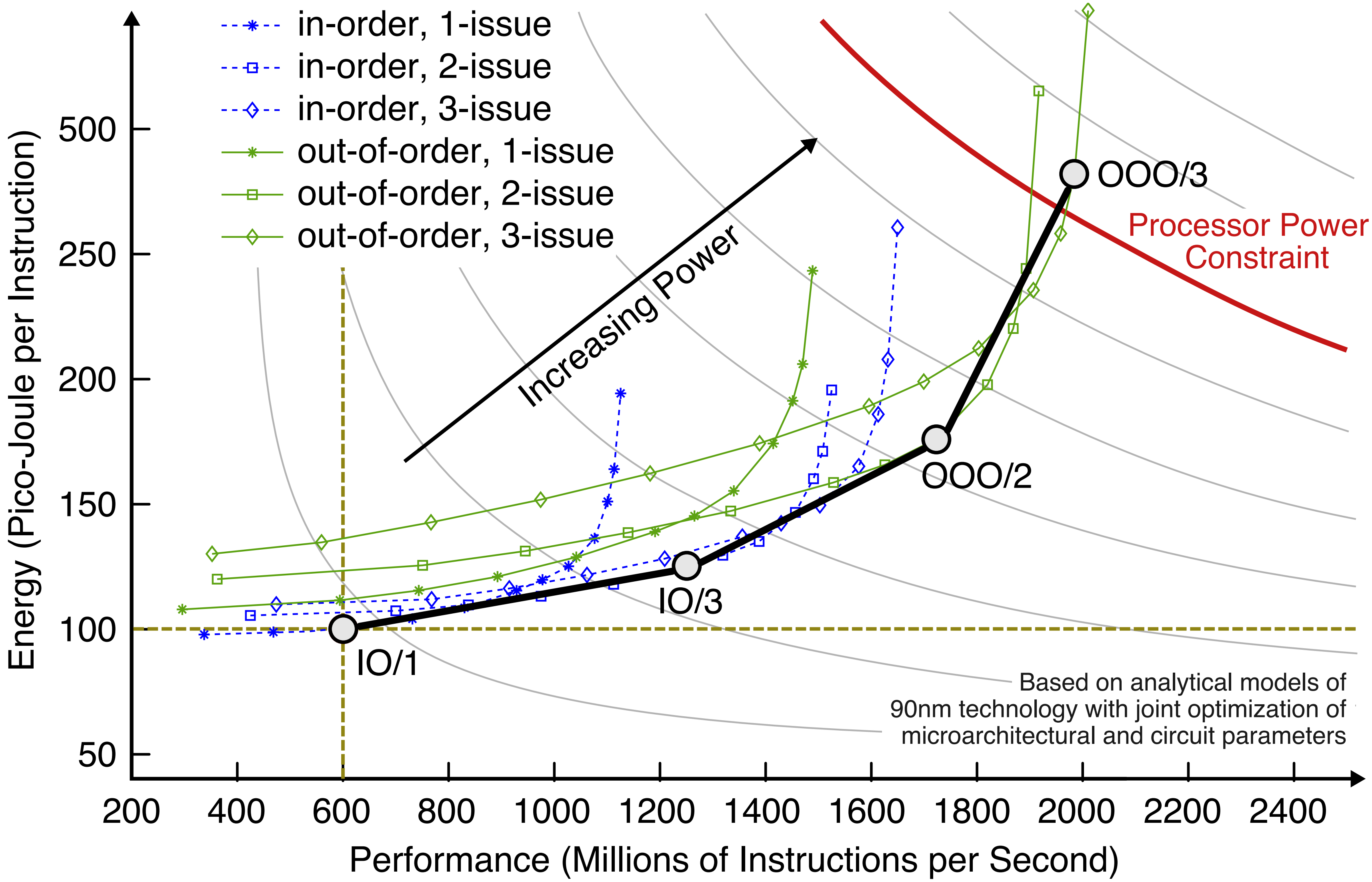
Chip Packaging
Chip Cooling
System Noise
Case Temperature
Data-Center Air Conditioning

**Energy**

Battery Life
Electricity Bill
Mobile Device Weight

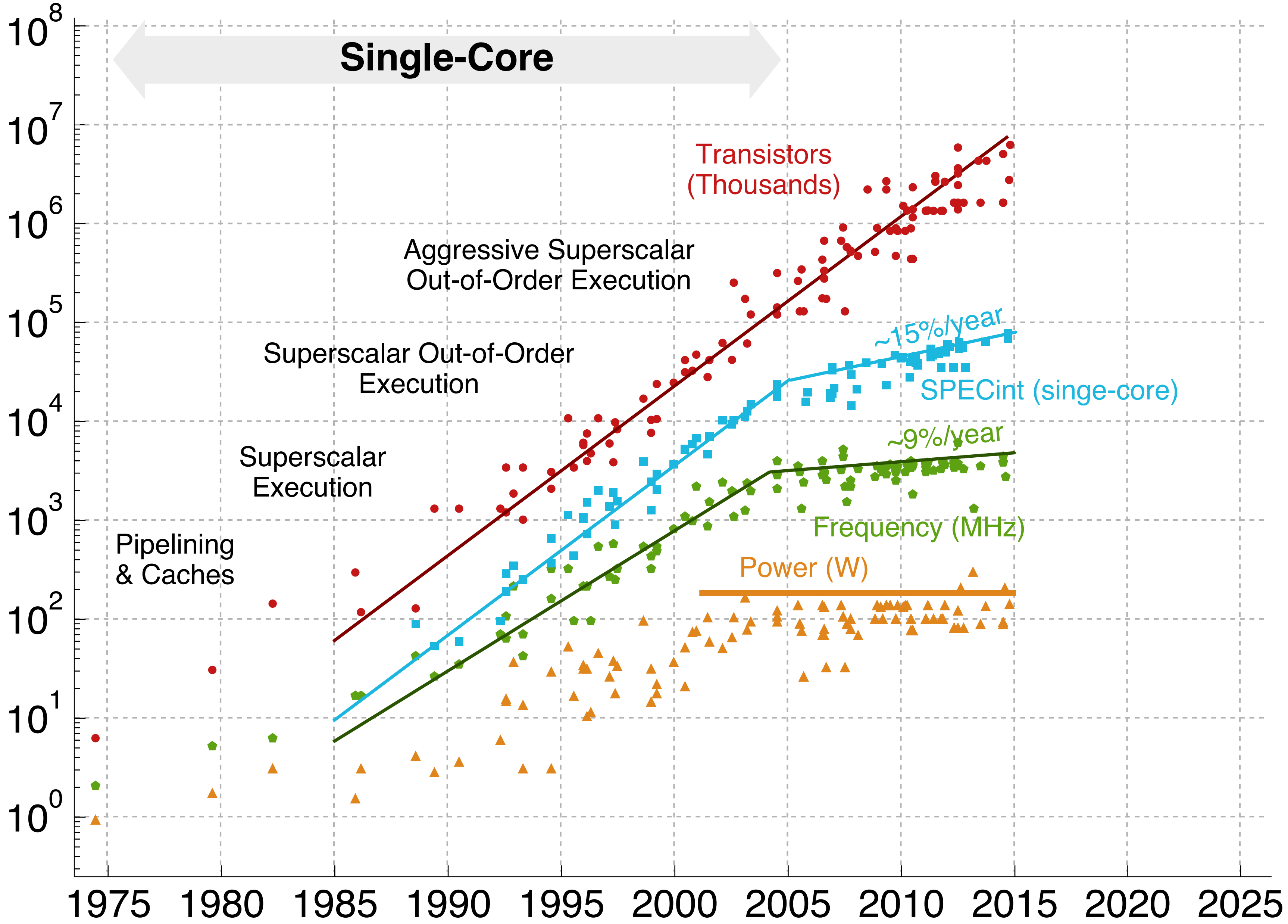


# Energy and Performance of Single-Core Processor

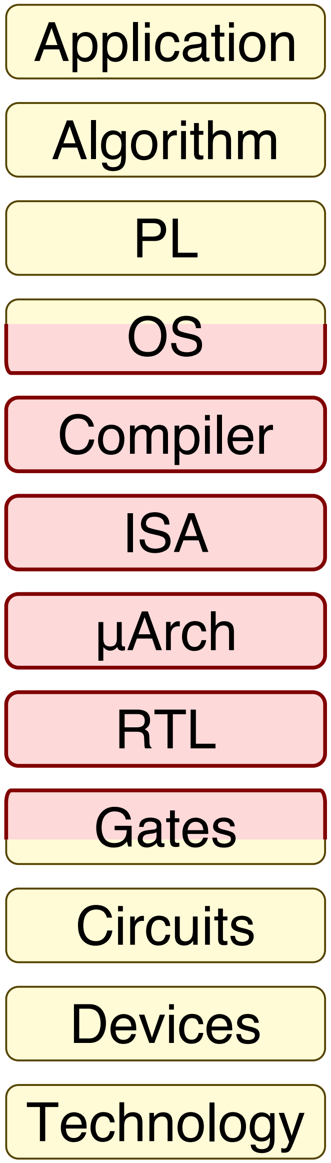


Based on analytical models of 90nm technology with joint optimization of microarchitectural and circuit parameters

Adpated from O. Azizi et al. “Energy-Performance Tradeoffs ...” ISCA, 2010.



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



# Agenda

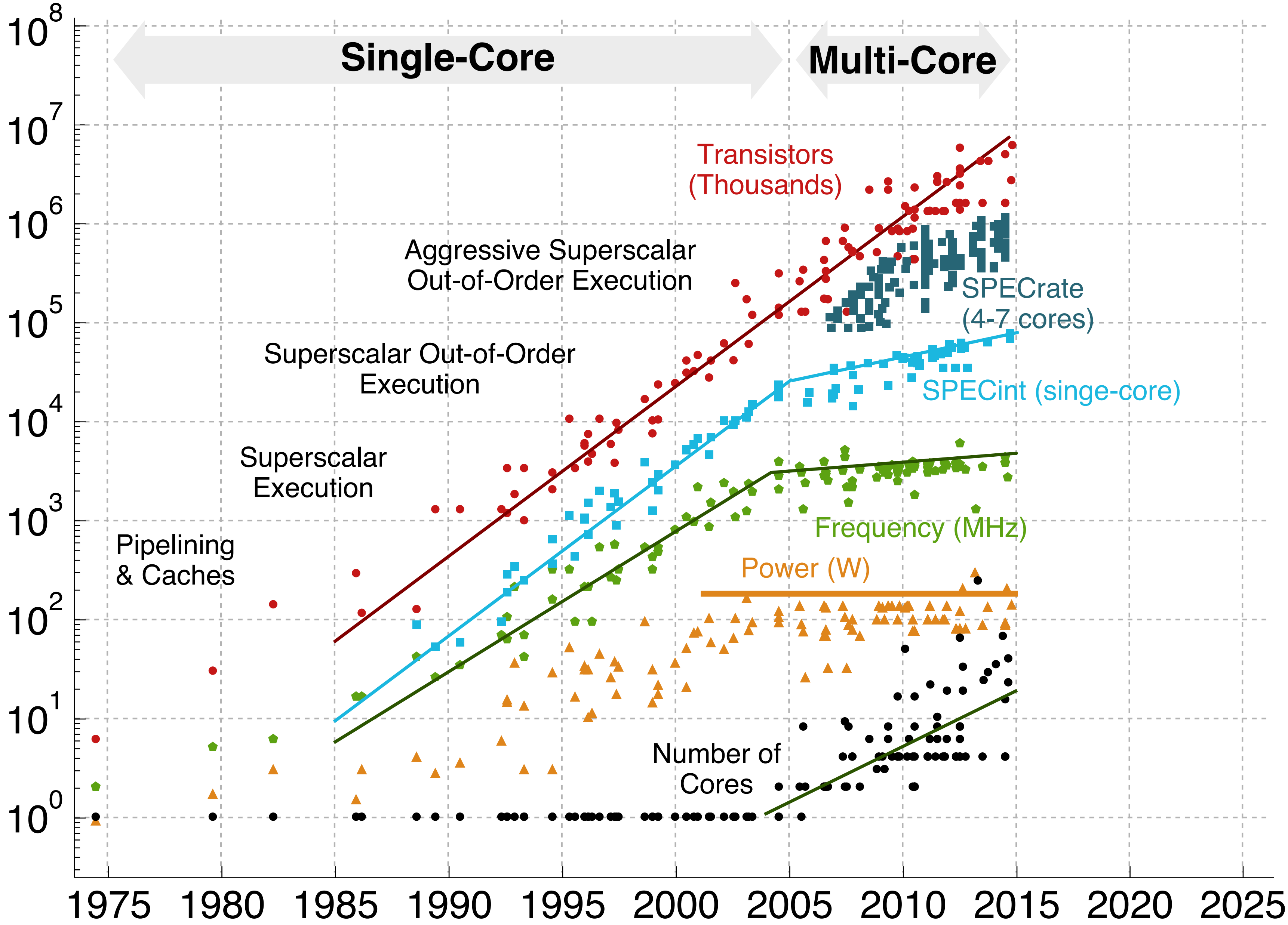
What is Computer Architecture?

Trends: Single-Core Era

**Trends: Multicore-Core Era**

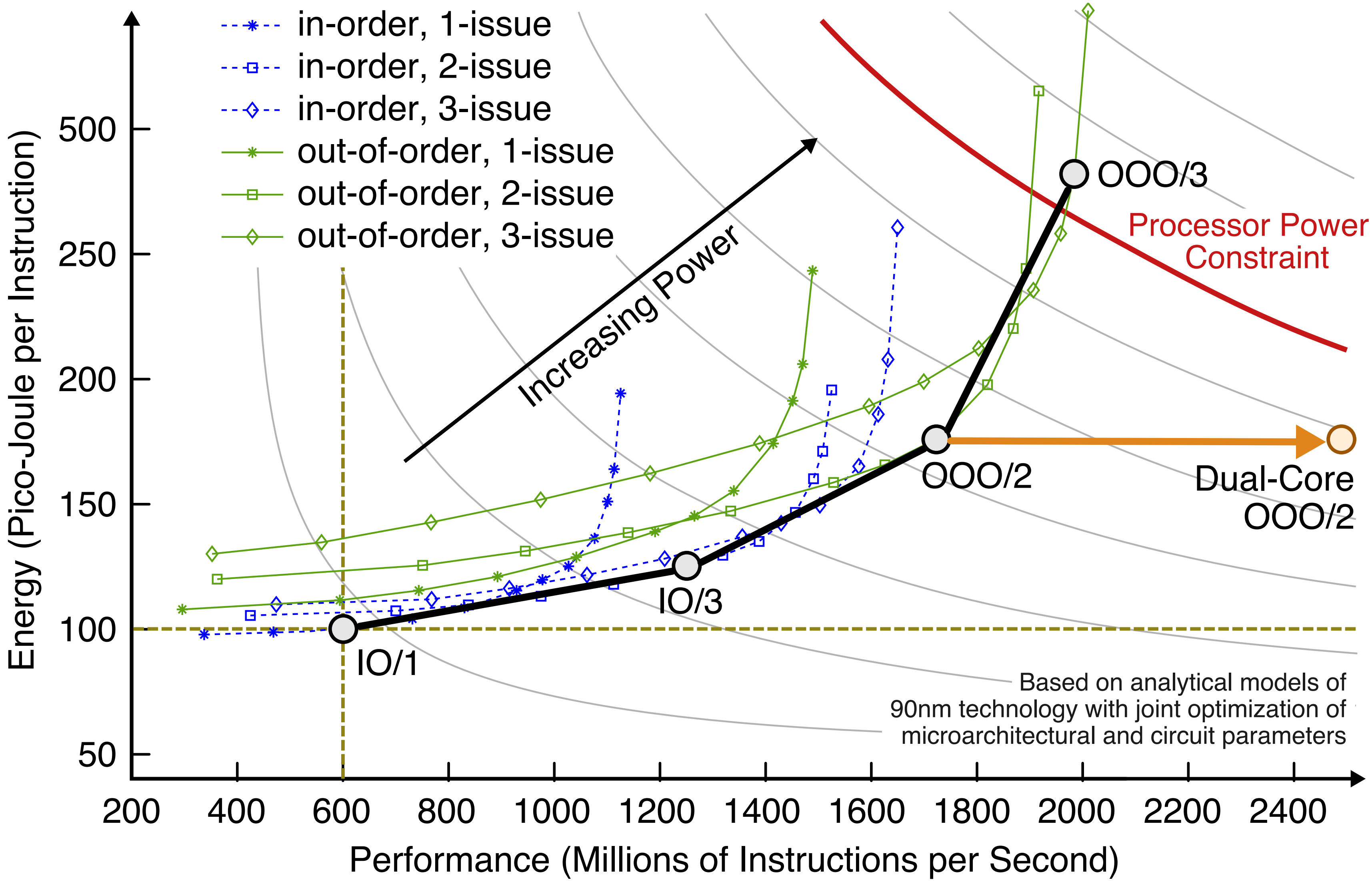
Trends: Accelerator Era

Computer Architecture Design



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

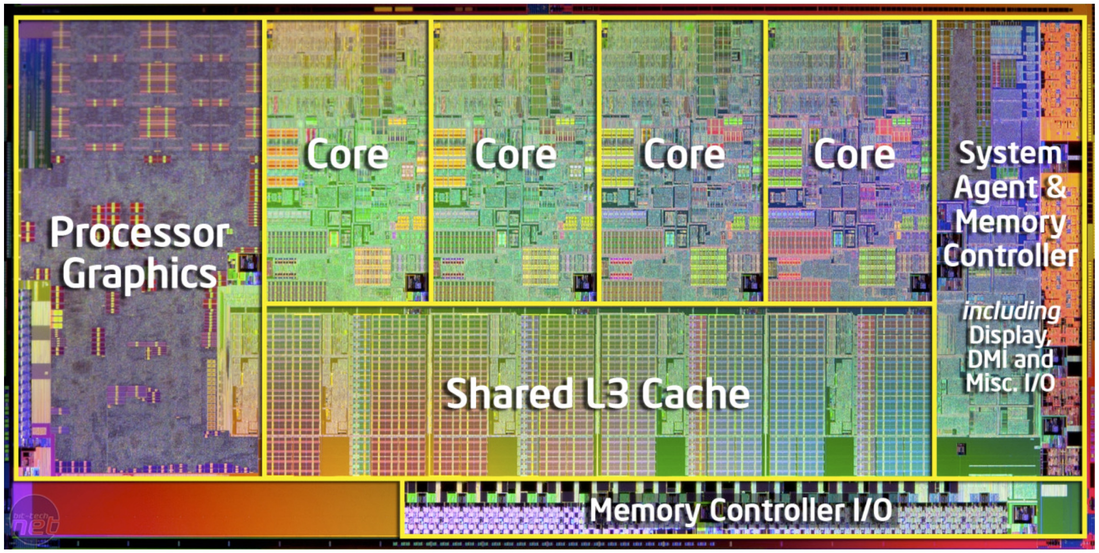
# Energy and Performance of Multi-Core Processor



Based on analytical models of 90nm technology with joint optimization of microarchitectural and circuit parameters

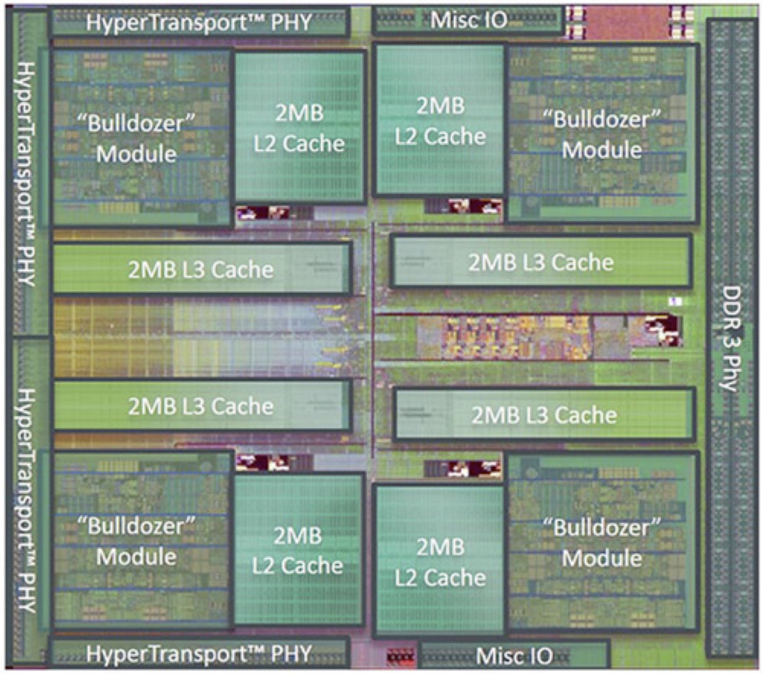
Adpated from O. Azizi et al. “Energy-Performance Tradeoffs ...” ISCA, 2010.

# Architectures in the Multi-Core Era



**Intel Sandy Bridge** (2011)

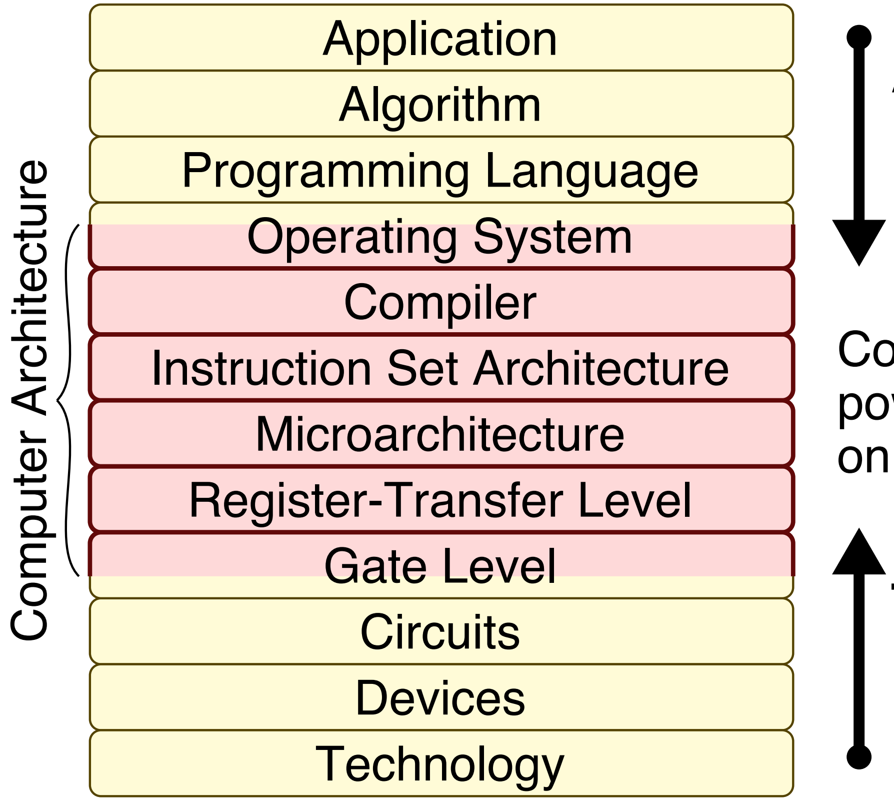
- 1B trans, 3.5GHz, 32nm
- Four superscalar out-of-order cores
- Multi-level cache hierarchy
- Ring network



**AMD Bulldozer** (2011)

- 1.2B trans, 3.6GHz, 32nm
- Four "two-core" clusters
- Multi-level cache hierarchy
- Crossbar network

# Application Requirements ↔ Technology Constraints in the Multi-Core Era



**Application Requirements**

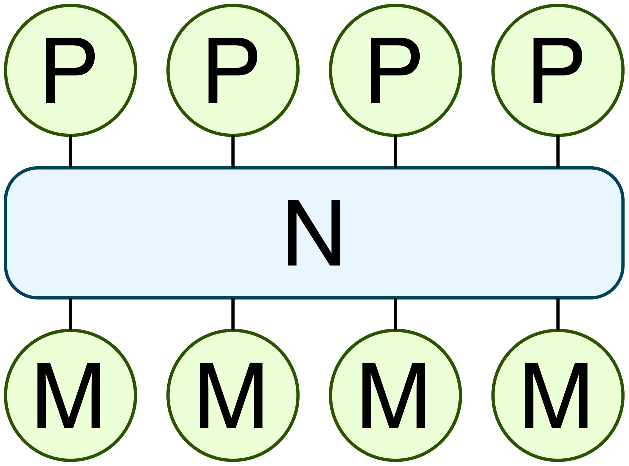
- Multi-core processors require programmers to parallelize their software to take advantage of the multiple cores

Computer architects address energy and power constraints by integrating multiple cores on a chip creating new software challenges

**Technology Constraints**

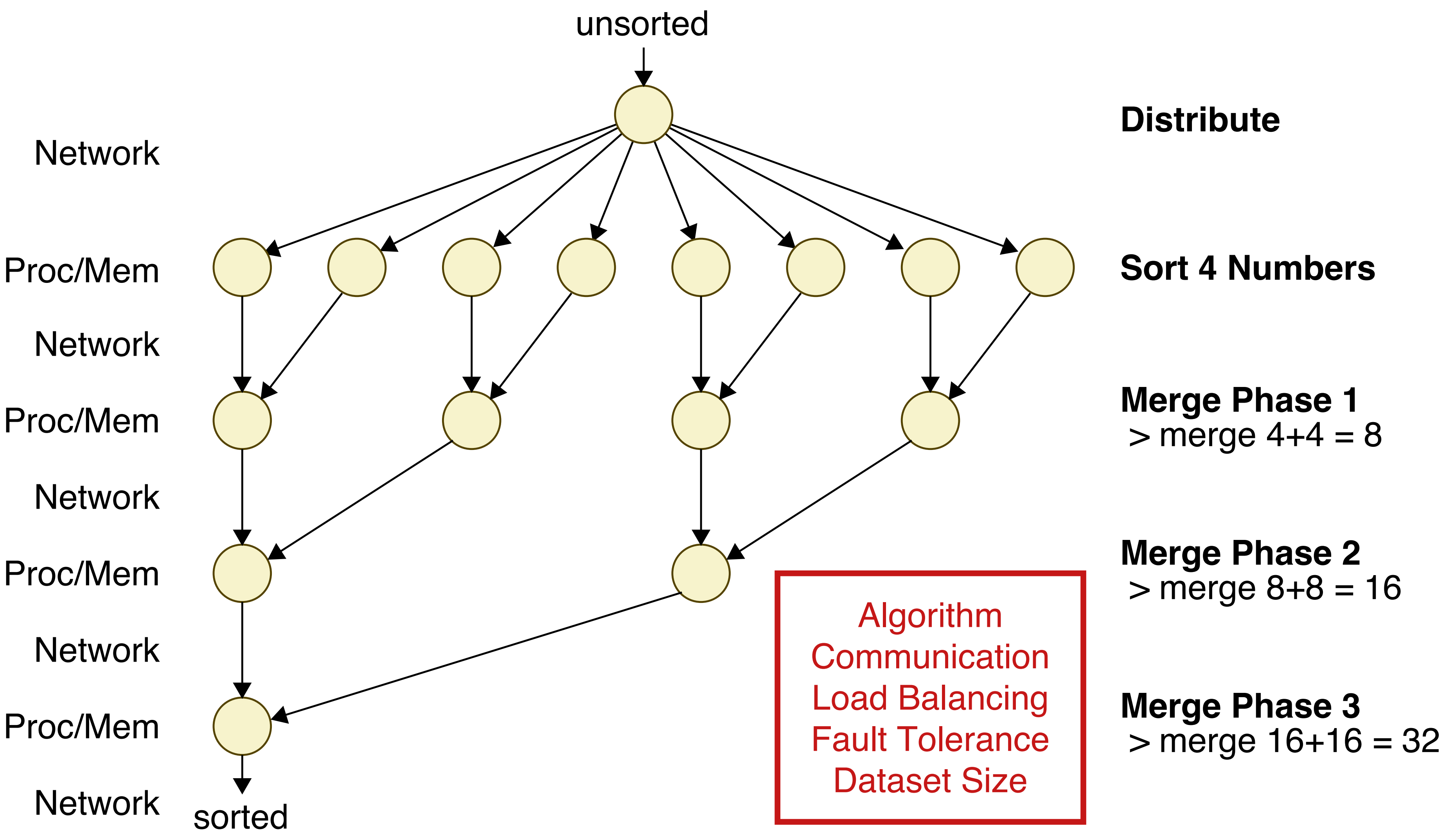
- Energy and power constraints limit processor clock frequencies and the complexity of each core

# Activity #2: Sorting with a Parallel Processor



- **Application:** Sort 32 numbers
- **Simulated Parallel Computing System**
  - Processor: Group of 2–8 students
  - Memory: Worksheet, scratch paper
  - Network: Communicating between students
- **Activity Steps**
  - 1. Discuss strategy with group
  - 2. When instructor starts timer, master processor flips over worksheet
  - 3. Sort 32 numbers as fast as possible
  - 4. Lookup when completed and write time on worksheet
  - 5. *Master processor only* raises hand
  - 6. When everyone is finished, then analyze data

# Activity #2: Discussion



unsorted

Network

Proc/Mem

Network

Proc/Mem

Network

Proc/Mem

Network

Proc/Mem

Network

sorted

**Distribute**

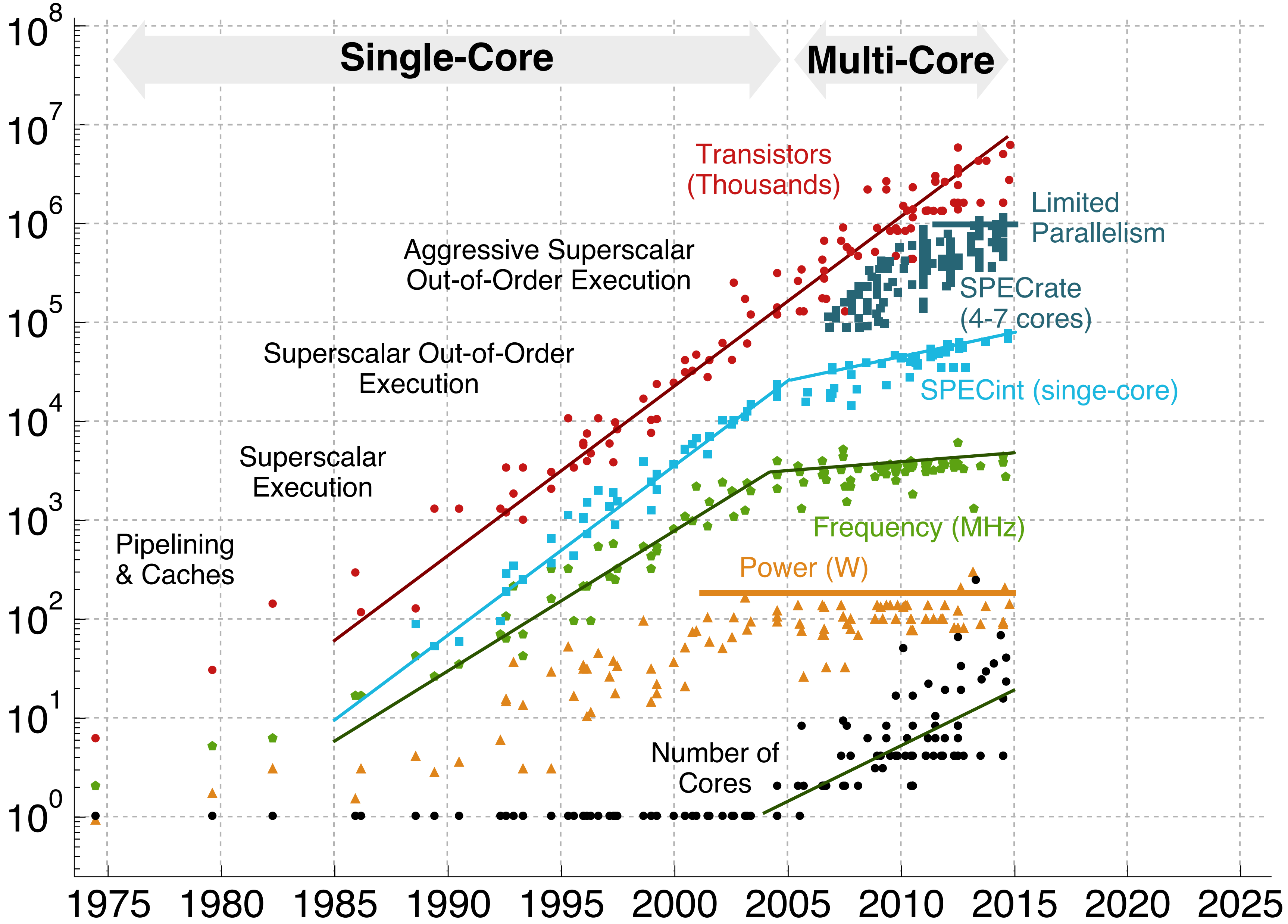
**Sort 4 Numbers**

**Merge Phase 1**
> merge 4+4 = 8

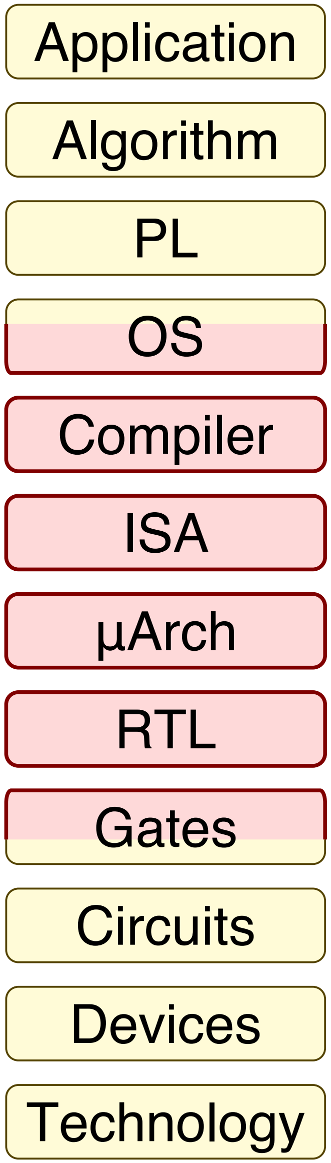
**Merge Phase 2**
> merge 8+8 = 16

**Merge Phase 3**
> merge 16+16 = 32

Algorithm
Communication
Load Balancing
Fault Tolerance
Dataset Size



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


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

# Agenda

What is Computer Architecture?

Trends: Single-Core Era

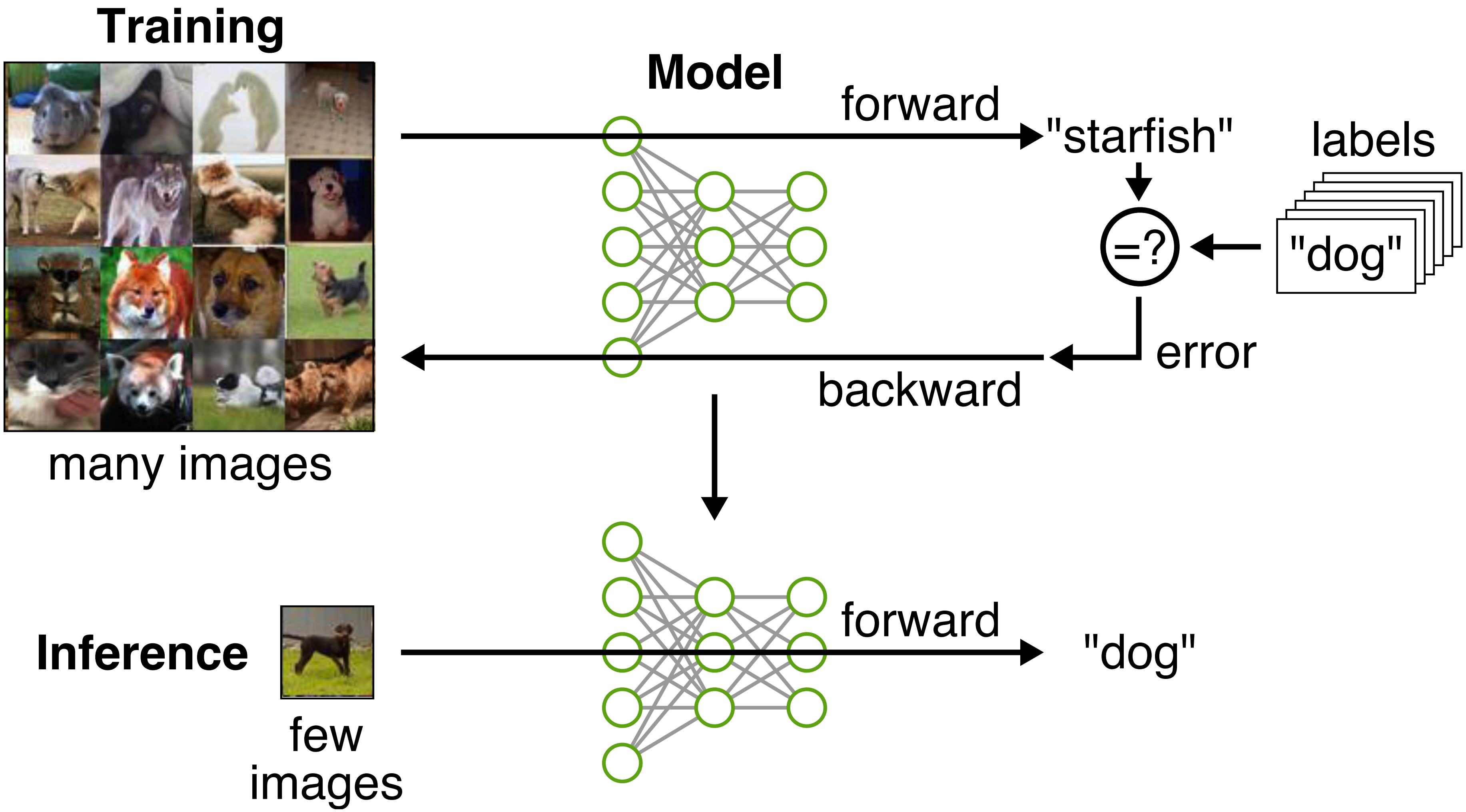
Trends: Multicore-Core Era

Trends: Accelerator Era

Computer Architecture Design

# Image Recognition

# Machine Learning (ML): Training vs. Inference



Training

Model

forward

"starfish"

labels

=?

"dog"

many images

error

backward

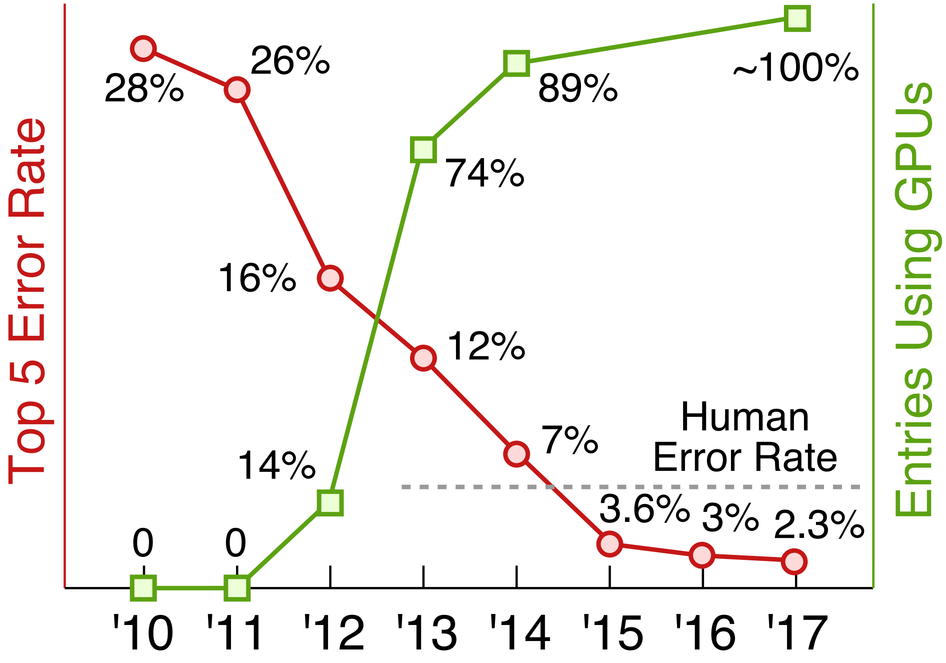
Inference

forward

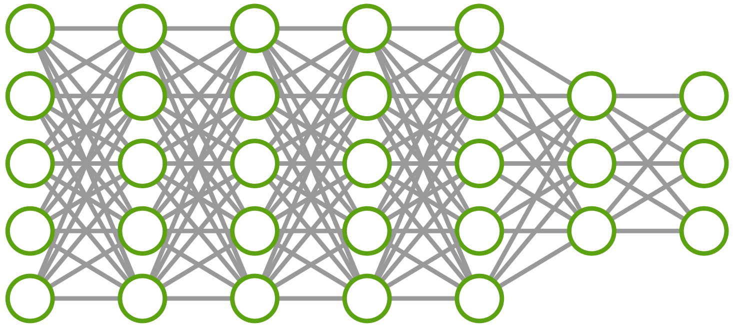
"dog"

few images

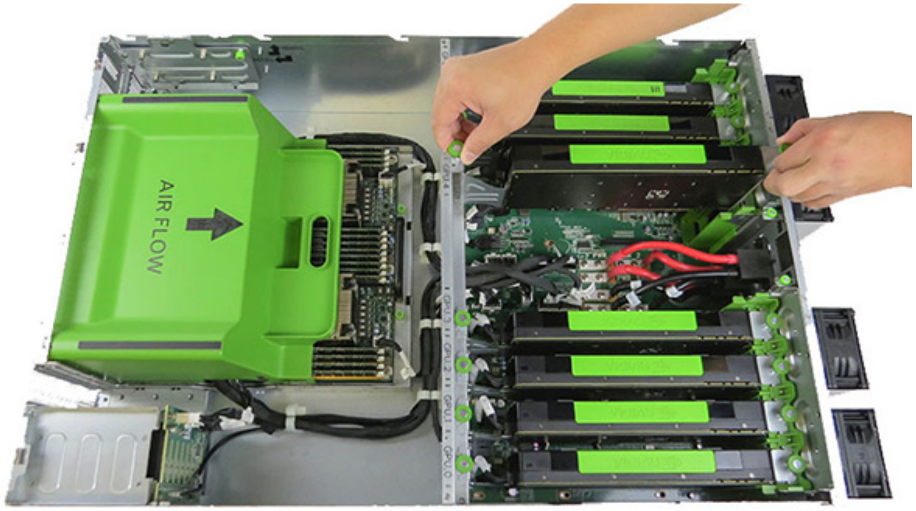
# ImageNet Large-Scale Visual Recognition Challenge



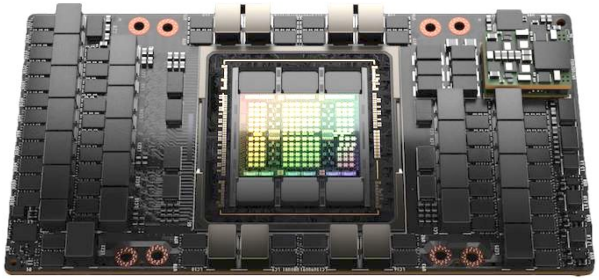
**Hardware:** Graphics Processing Units



**Software:** Deep Neural Network

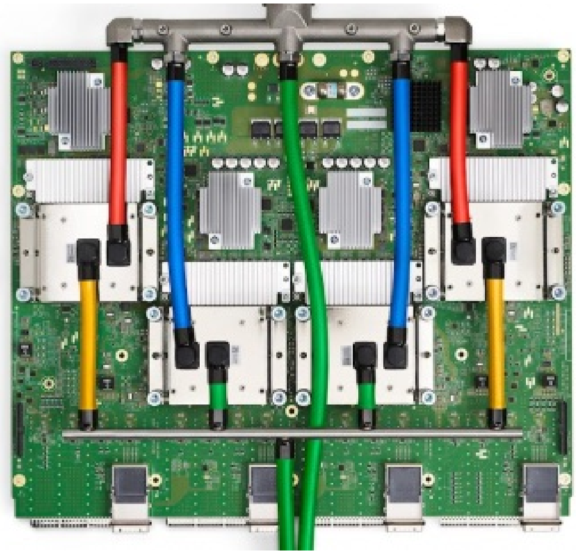


# Accelerators for Machine Learning in the Cloud



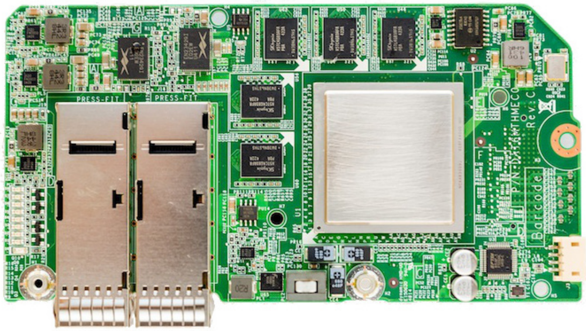
## NVIDIA DGX Hopper

- 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

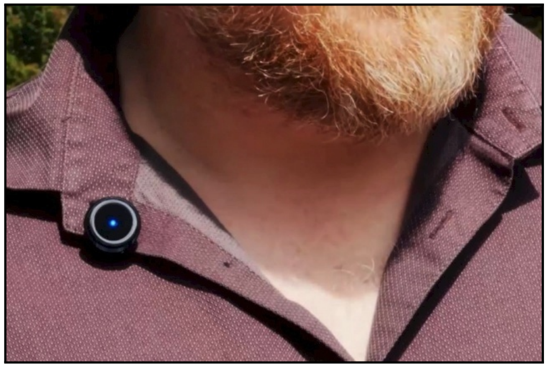
# Accelerators for Machine Learning at the Edge

## Amazon Echo

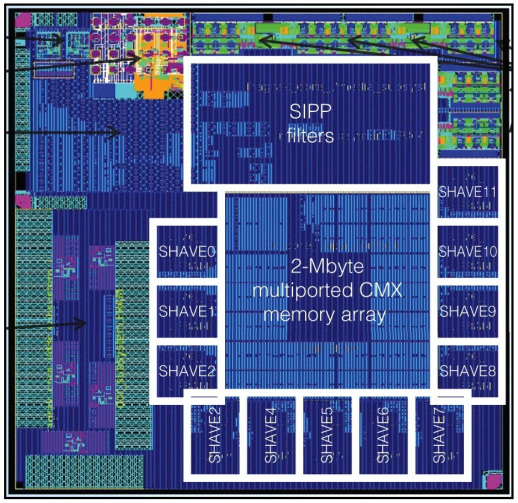
- Developing AI chips so Echo line can do more on-board processing
- Reduces need for round-trip to cloud
- Co-design the algorithms and the underlying hardware

## Facebook Oculus

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



## Movidius Myriad 2



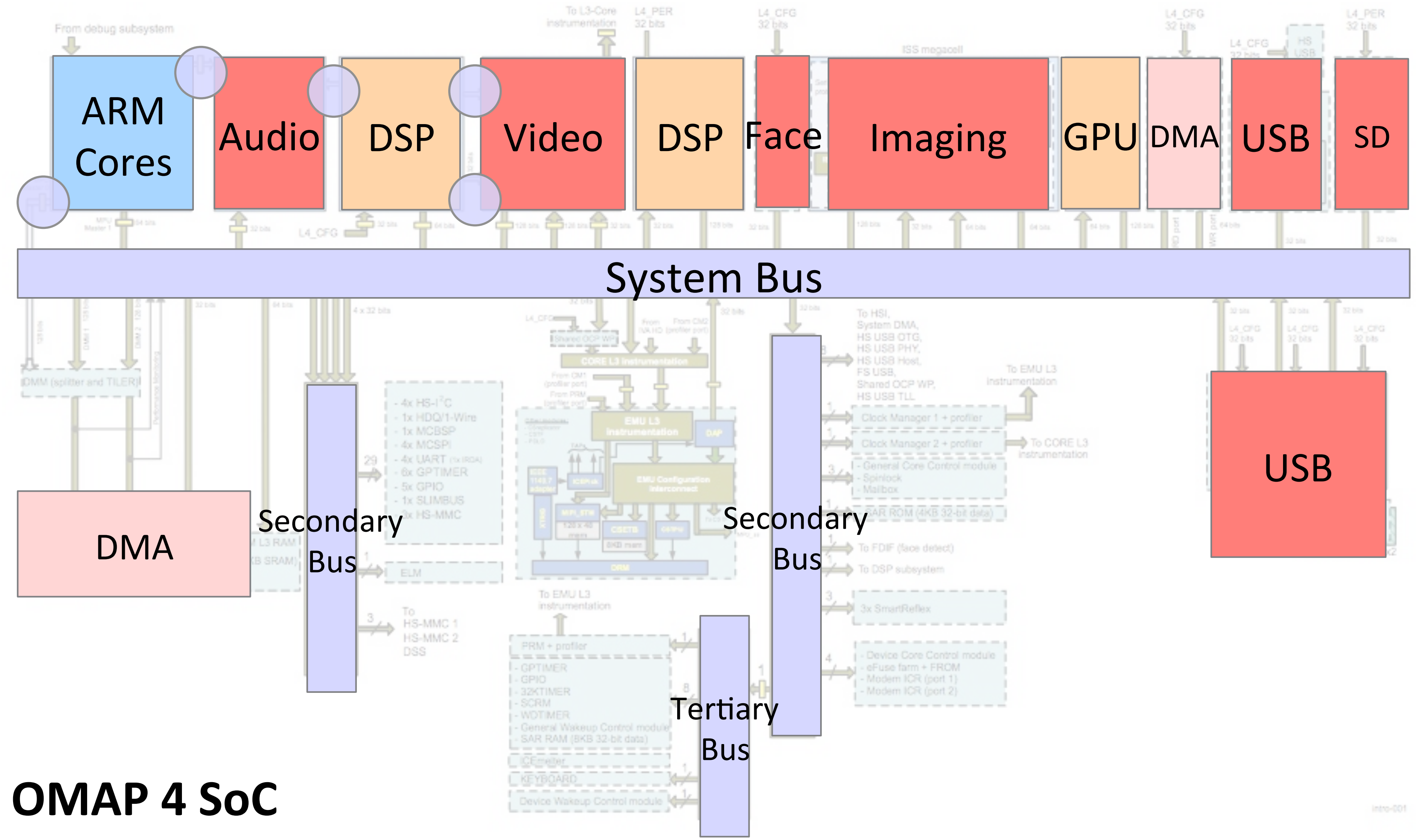
## Top-five software companies are all building custom accelerators

- **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**

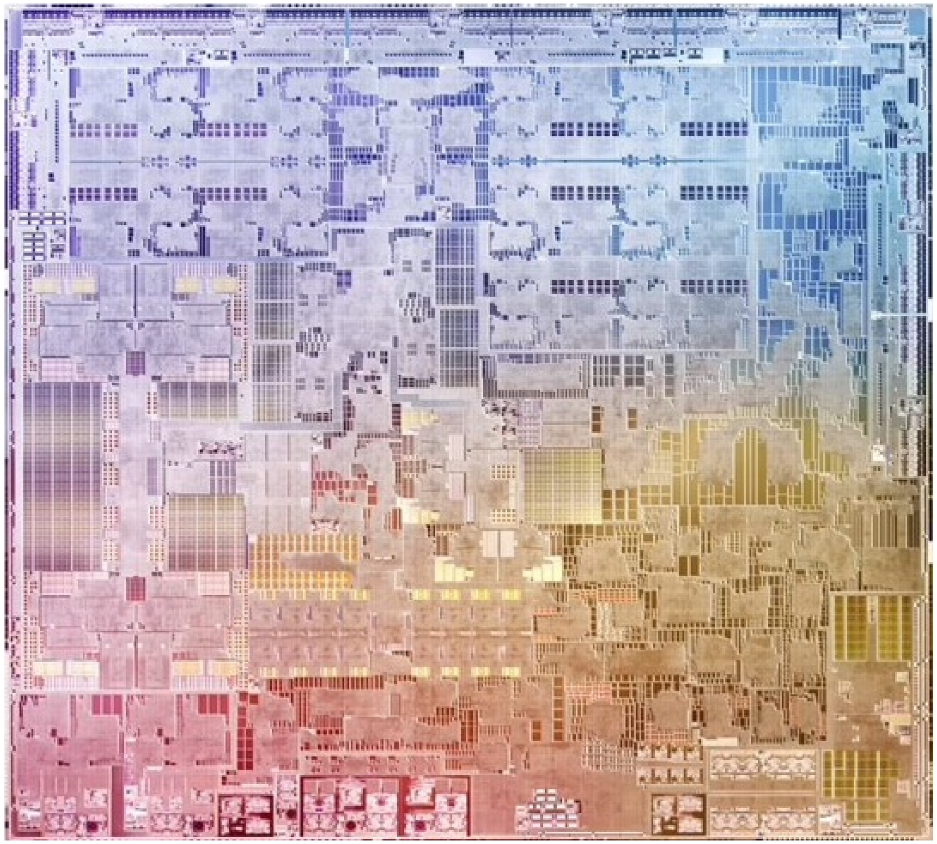
# Architectures in the Accelerator Era



**OMAP 4 SoC**

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

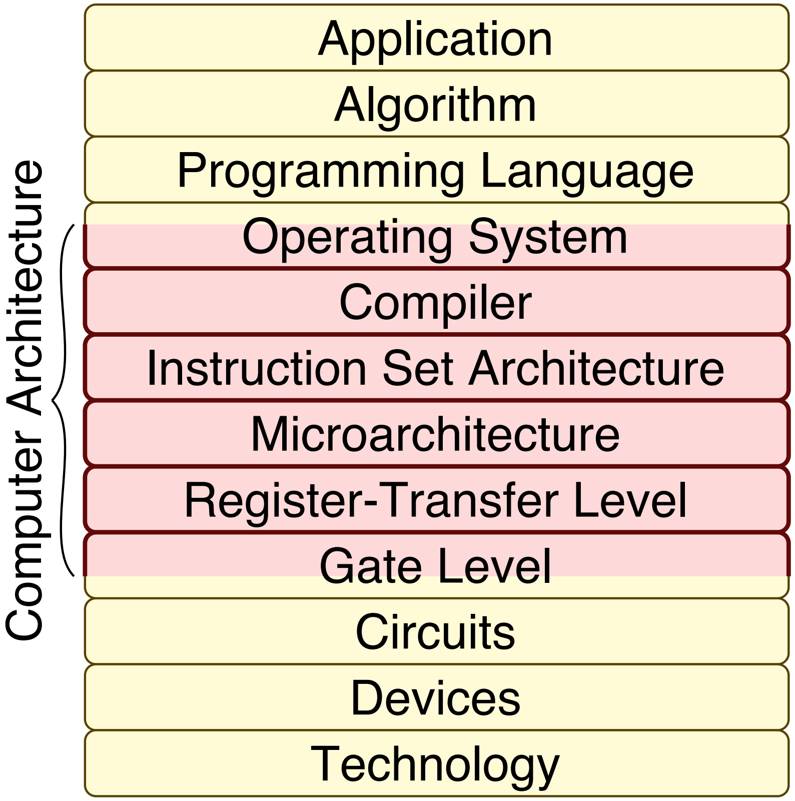
# Architectures in the Accelerator Era



**Apple M2 System-on-Chip** (2022)

- 20B trans, 3.5GHz, 5nm
- 8 superscalar out-of-order cores
- Multi-level cache hierarchy
- Crossbar network?
- NPU for accelerating ML
- GPU for accelerating graphics
- Media engine for accelerating video encode/decode

# Application Requirements ↔ Technology Constraints in the Accelerator Era



Application

Algorithm

Programming Language

Operating System

Compiler

Instruction Set Architecture

Microarchitecture

Register-Transfer Level

Gate Level

Circuits

Devices

Technology

Computer Architecture

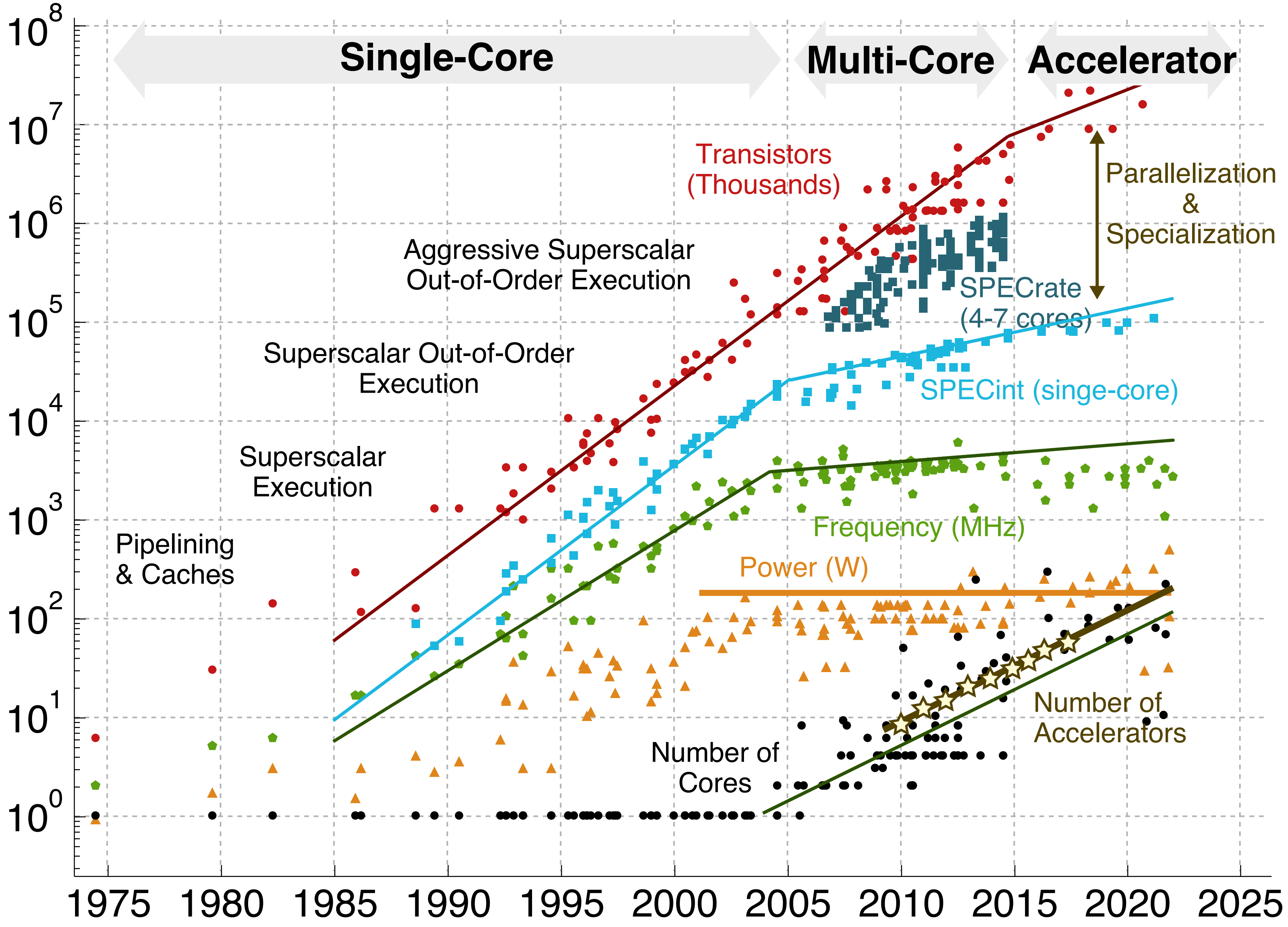
**Application Requirements**

- Accelerators require programmers to restructure their software to take advantage of the specialized hardaware

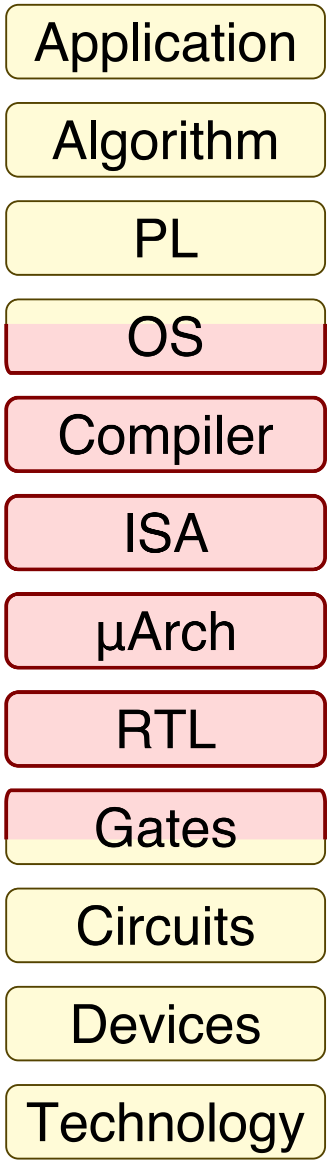
Computer architects address slower technology scaling by integrating multiple accelerators on a chip creating new software challenges

**Technology Constraints**

- The slowing of Moore's law means general-purpose processors no longer provide significant application benefit



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



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

# Agenda

What is Computer Architecture?

Trends: Single-Core Era

Trends: Multicore-Core Era

Trends: Accelerator Era

Computer Architecture Design

# What do computer architects actually do?

## General Science

Discover truths about nature

Ask question about nature

Construct hypothesis

Test with experiment

Analyze results and draw conclusions

## Computer Engineering

Explore design space for a new system

Design and model baseline system

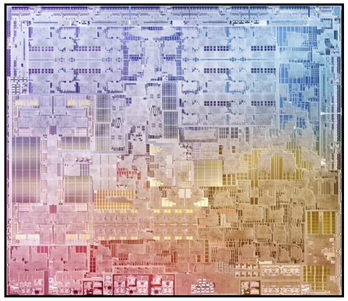
Ask question about system

Test with experiment

Analyze results and draw conclusions

Build prototype or real system

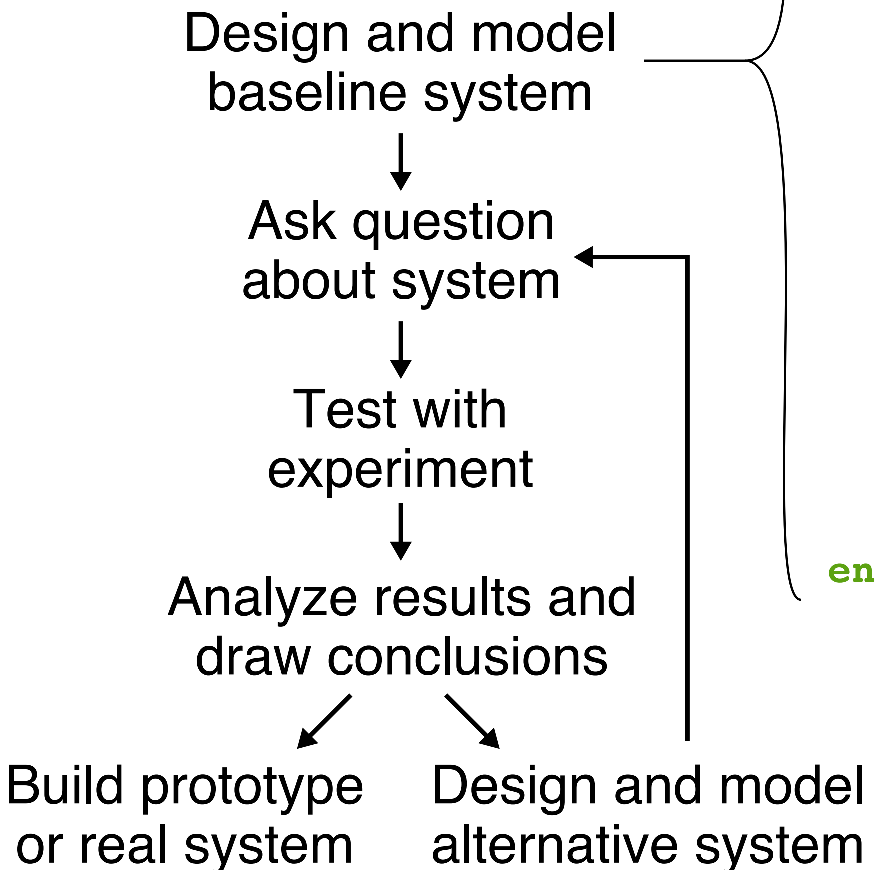
Design and model alternative system



# Modeling in Computer Architecture

**Computer Engineering**

Explore design space for a new system



Design and model baseline system

Ask question about system

Test with experiment

Analyze results and draw conclusions

Build prototype or real system

Design and model alternative system

```
// rdy is OR of the AND of reqs and grants
assign in_rdy = | (reqs & grants);

reg [2:0] reqs;
always @(*) begin
  if ( in_val ) begin

    // eject packet if it is for this tile
    if ( dest == p_router_id )
      reqs = 3'b010;

    // otherwise, just pass it along ring
    else
      reqs = 3'b001;

  end else begin
    // if !val, don't request any ports
    reqs = 3'b000;
  end
end
```

**Verilog • SystemVerilog • VHDL**

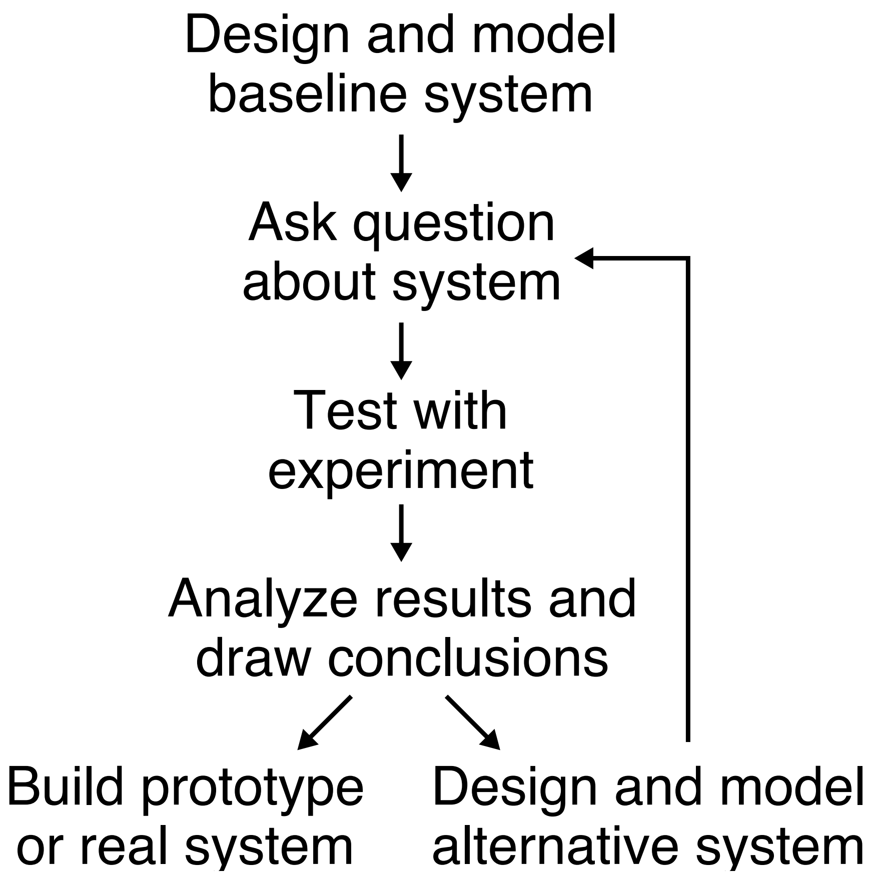
**C++ • SystemC**

**Bluespec • Chisel • Python**

# How do we design something so incredibly complex?

**Computer Engineering**

Explore design space for a new system



Design and model baseline system

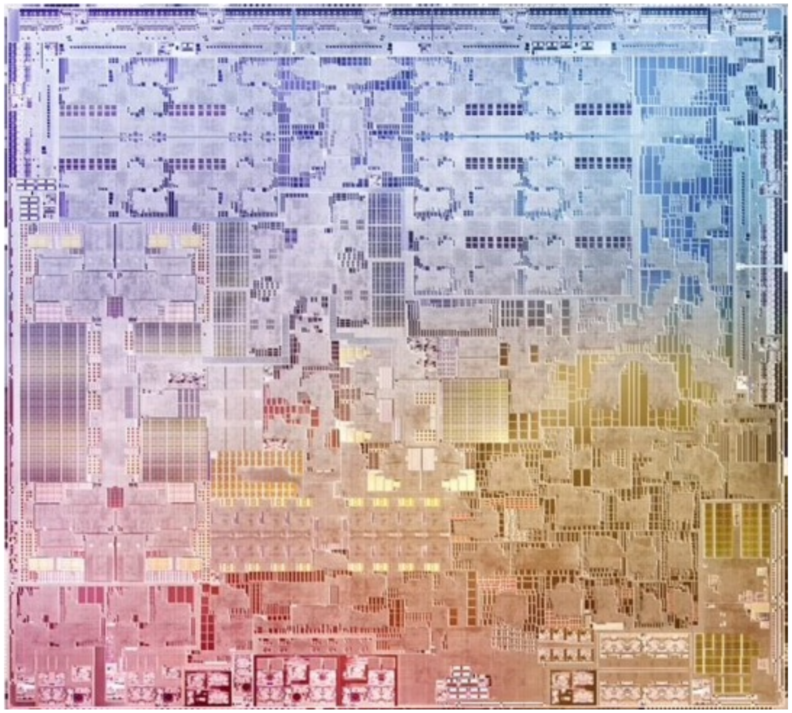
Ask question about system

Test with experiment

Analyze results and draw conclusions

Build prototype or real system

Design and model alternative system



**Fighter Airplane:** ~100,000 parts

**Apple M2 System-on-Chip**

20 billion transistors

## Design Principles

- Modularity – Decompose into components with well-defined interfaces
- Hierarchy – Recursively apply modularity principle
- Encapsulation – Hide implementation details from interfaces
- Regularity – Leverage structure at various levels of abstraction
- Extensibility – Include mechanisms/hooks to simplify future changes

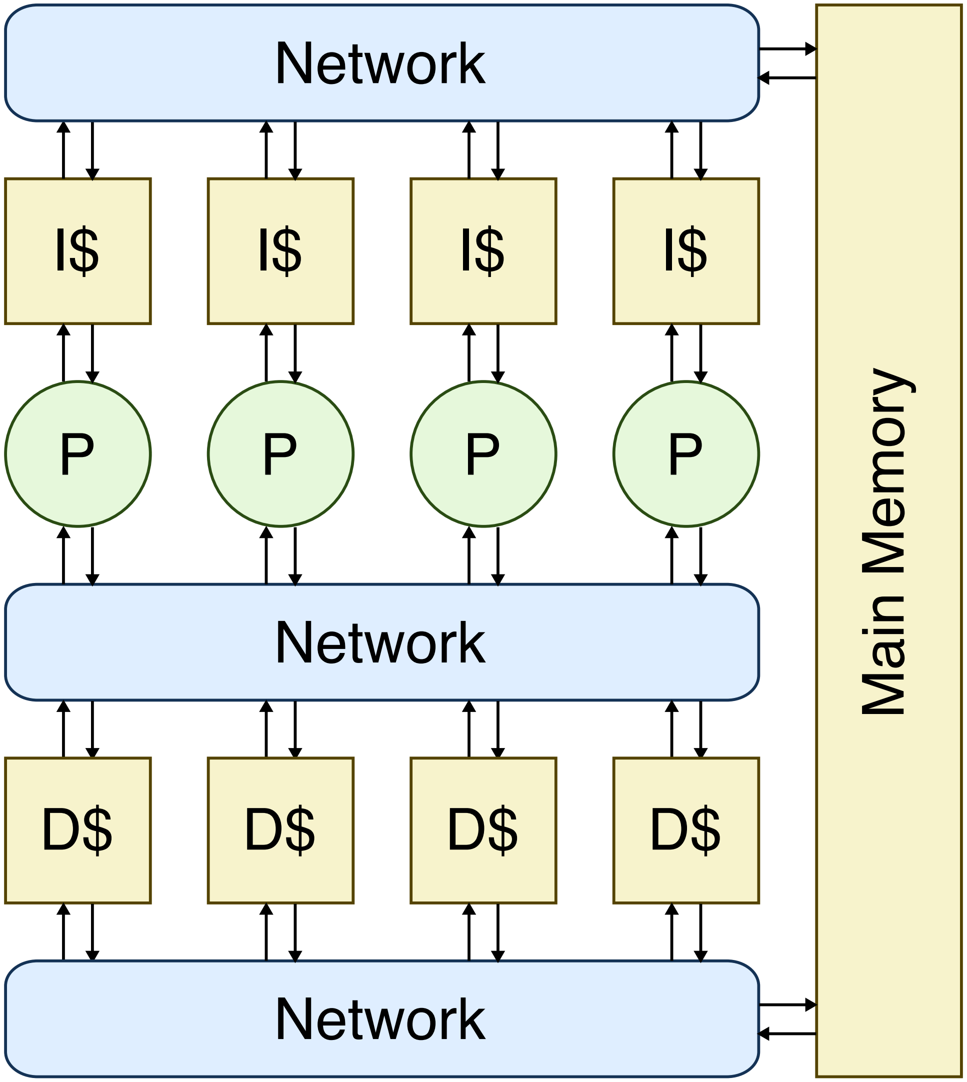
## Design Patterns

- Processors, Memories, Networks
- Control/Datapath Split
- Single-Cycle, FSM, Pipelined Control
- Raw Port, Message, Method Interfaces

## Design Methodologies

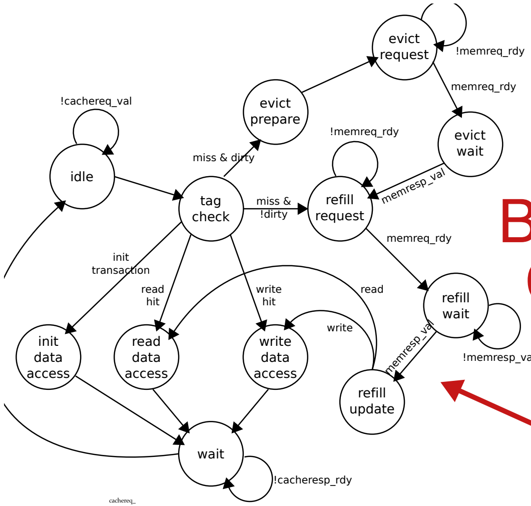
- Agile Hardware Development
- Test-driven Development
- Incremental Development

# Final Goal for Lab Assignments

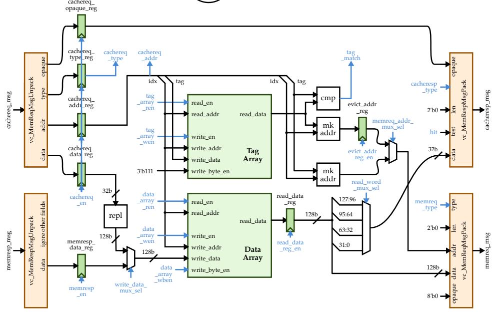


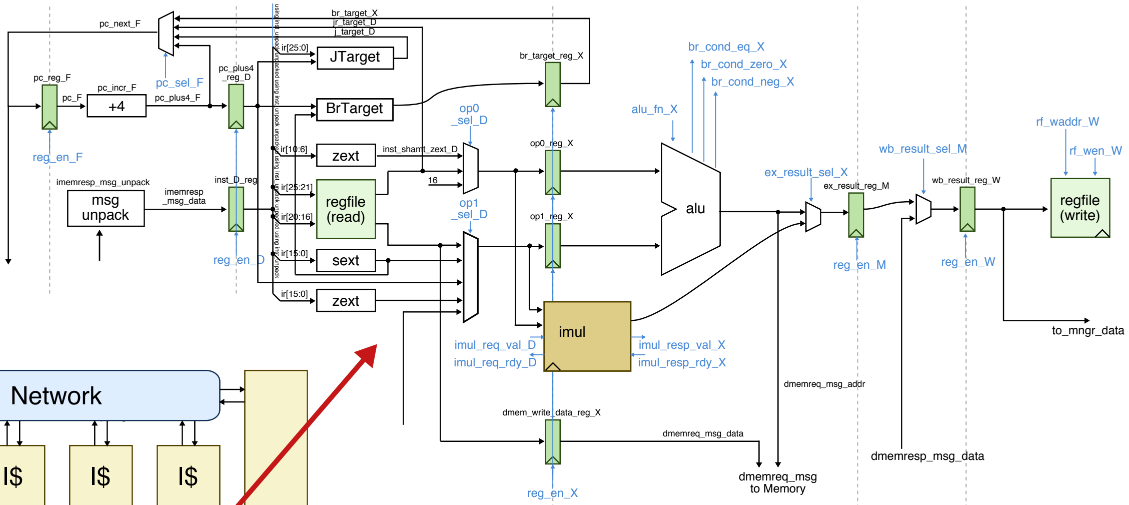
Quad-core processor with private L1 instruction caches and a shared, banked L1 data cache interconnected through various ring networks implemented at the register-transfer-level and capable running real parallel programs

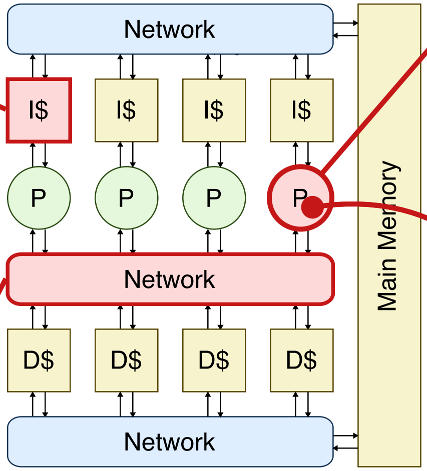
Lab assignments will use an agile hardware development methodology based on the Verilog hardware description language, a Python testing framework, the GitHub repository hosting site, and the GitHub Actions continuous integration service



# Lab 3: Blocking Cache

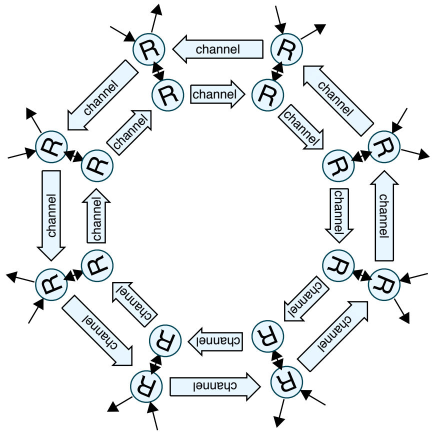


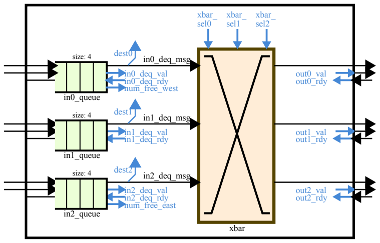


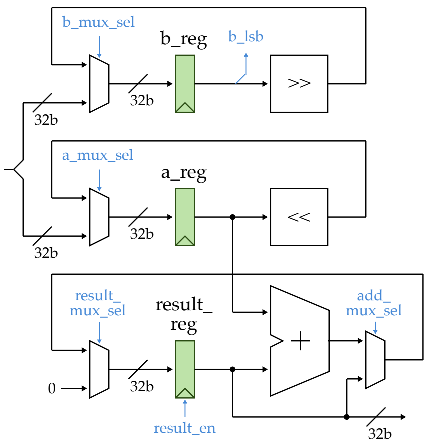


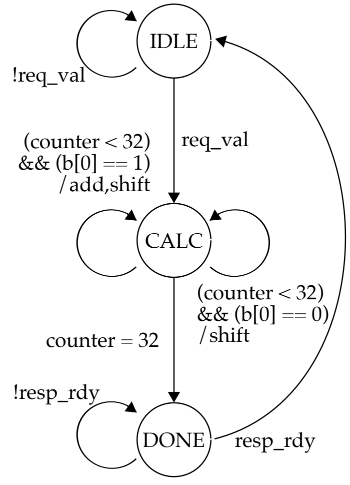
# Lab 2: Pipelined Processor

# Lab 4: Muliticore

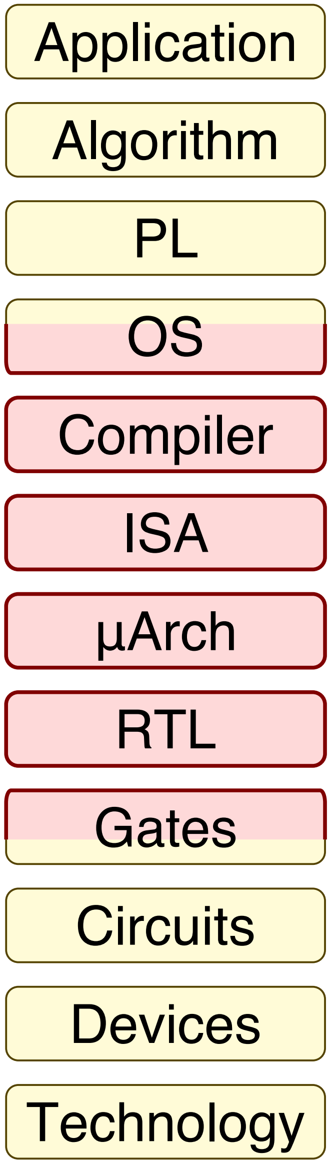







IDLE
!req_val
req_val
(counter < 32) && (b[0] == 1) / add,shift
CALC
(counter < 32) && (b[0] == 0) / shift
counter = 32
!resp_rdy
DONE
resp_rdy

# Lab 1: Iterative Multiplier



Application

Algorithm

PL

OS

Compiler

ISA

µArch

RTL

Gates

Circuits

Devices

Technology

# Take-Away Points

- Computer architecture is the process of building computing systems to meet given application requirements within physical technology constraints
- The field of computer architecture has recently evolved through the single-core era and multi-core era and is now in the accelerator era making it an exciting time to study computer architecture
- Computer architecture design involves a systematic design process based on design principles, patterns, and methodologies
- This course will provide a strong foundation in computer architecture principles and practice so that students can contribute to this new era!