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

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

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

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

```c
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

RISC-V Instruction Set
Instructions that machine executes

ECE 4750 Course Overview 2 / 40
The Computer Systems Stack

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

- Boolean logic gates and functions
- Combining devices to do useful work
- Transistors and wires
- Silicon process technology
- How data flows through system

- Technology: Silicon process
In its broadest definition, computer engineering is the development of the abstraction/implementaiton layers that allow us to execute information processing applications efficiently using available manufacturing technologies.
Computer Architecture in the ECE/CS Curriculum

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

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

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
Agenda

What is Computer Architecture?

Trends: Single-Core Era

Trends: Multicore-Core Era

Trends: Accelerator Era

Computer Architecture Design
What is Comp Arch?

- **Trends: Single-Core Era**
  - Trends: Multicore-Core Era
  - Trends: Accelerator Era
  - Comp Arch Design

### Transistors (Thousands)


<table>
<thead>
<tr>
<th>Year</th>
<th>Transistors</th>
</tr>
</thead>
<tbody>
<tr>
<td>1975</td>
<td>~10^5</td>
</tr>
<tr>
<td>1980</td>
<td>~10^6</td>
</tr>
<tr>
<td>1985</td>
<td>~10^7</td>
</tr>
<tr>
<td>1990</td>
<td>~10^8</td>
</tr>
<tr>
<td>1995</td>
<td>~10^9</td>
</tr>
<tr>
<td>2000</td>
<td>~10^10</td>
</tr>
<tr>
<td>2005</td>
<td>~10^11</td>
</tr>
<tr>
<td>2010</td>
<td>~10^12</td>
</tr>
<tr>
<td>2015</td>
<td>~10^13</td>
</tr>
<tr>
<td>2020</td>
<td>~10^14</td>
</tr>
<tr>
<td>2025</td>
<td>~10^15</td>
</tr>
</tbody>
</table>

### SPECint Performance

- ~50%/year
- ~38%/year

### Frequency (MHz)

- ~50%/year

### Single-Core Trends

- Pipelining & Caches
- Superscalar Execution
- Superscalar Out-of-Order Execution
- Aggressive Superscalar Out-of-Order Execution

Energy and Power Constraints

Power = Energy / Second = Energy / Op \times Ops / Second

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

- Energy
  - Battery Life
  - Electricity Bill
  - Mobile Device Weight

Increasing Power
100W Workstation Power Constraint
1W Handheld Power Constraint

Performance (Ops/Second) vs. Energy (Joules/Op)
What is Comp Arch?

- **Trends: Single-Core Era**
- Trends: Multicore-Core Era
- Trends: Accelerator Era
- Comp Arch Design

### Energy and Performance of Single-Core Processor

- **in-order, 1-issue**
- **in-order, 2-issue**
- **in-order, 3-issue**
- **out-of-order, 1-issue**
- **out-of-order, 2-issue**
- **out-of-order, 3-issue**

![Graph showing energy and performance tradeoffs for single-core processors](image)

- Increasing Power
- Processor Power Constraint

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

What is Comp Arch?

- **Trends: Single-Core Era**
- Trends: Multicore-Core Era
- Trends: Accelerator Era
- Comp Arch Design

Transistors (Thousands)


Agenda

What is Computer Architecture?

Trends: Single-Core Era

Trends: Multicore-Core Era

Trends: Accelerator Era

Computer Architecture Design
What is Comp Arch? Trends: Single-Core Era

• Trends: Multicore-Core Era

• Trends: Accelerator Era

Comp Arch Design

Energy and Performance of Multi-Core Processor

Performance (Millions of Instructions per Second)

Energy (Pico-Joule per Instruction)

- in-order, 1-issue
- in-order, 2-issue
- in-order, 3-issue
- out-of-order, 1-issue
- out-of-order, 2-issue
- out-of-order, 3-issue

Processor Power Constraint

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

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.

**Technology Constraints**
- Energy and power constraints limit processor clock frequencies and the complexity of each core.

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

- Application Requirements
- Technology Constraints
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

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
What is Comp Arch? Trends: Single-Core Era

- Trends: Multicore-Core Era
- Trends: Accelerator Era
Comp Arch Design

Transistors (Thousands)


Single-Core

Multi-Core

Limited Parallelism

SPECrate (4-7 cores)

SPECint (single-core)

Power (W)

Frequency (MHz)

Number of Cores


Pipelining & Caches

Superscalar Execution

Superscalar Out-of-Order Execution

Limited Parallelism

SPECrate (4-7 cores)

SPECint (single-core)

Power (W)

Frequency (MHz)

Number of Cores

Agenda

What is Computer Architecture?

Trends: Single-Core Era

Trends: Multicore-Core Era

Trends: Accelerator Era

Computer Architecture Design
Image Recognition

Starfish

Dog
Machine Learning (ML): Training vs. Inference

**Training**
- Many images
- Model: "dog"
- Labels: "dog"
- Error: backward
- Forward: "starfish"

**Inference**
- Few images
- Forward: "dog"
ImageNet Large-Scale Visual Recognition Challenge

Top 5 Error Rate

Software: Deep Neural Network

Hardware: Graphics Processing Units

Entries Using GPUs

Human Error Rate

3.6% 3% 2.3%

28% 26% 16% 12% 7%

'10 '11 '12 '13 '14 '15 '16 '17

74% 89% ~100%
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

ECE 4750 Course Overview 27 / 40
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 Requirements
  - Accelerators require programmers to restructure their software to take advantage of the specialized hardware

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

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

Computer architects address slower technology scaling by integrating multiple accelerators on a chip creating new software challenges.
What is Comp Arch? Trends: Single-Core Era

- Trends: Multicore-Core Era
  - Trends: Accelerator Era

Comp Arch Design

- Transistors (Thousands)

- Pipelining & Caches
- Superscalar Execution
- Aggressive Superscalar Out-of-Order Execution
- SPECint (single-core)
- SPECrate (4-7 cores)
- Power (W)
- Frequency (MHz)
- Parallelization & Specialization
- Number of Accelerators
- Number of Cores

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

---

What is Comp Arch?  Trends: Single-Core Era  Trends: Multicore-Core Era  Trends: Accelerator Era  • Comp Arch Design •
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.
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!