ECE 5745 Complex Digital ASIC Design
Course Overview

Christopher Batten

School of Electrical and Computer Engineering
Cornell University

http://www.csl.cornell.edu/courses/ece5745
Complex Digital ASIC Design

- Course goal, structure, motivation
  - What is the goal of the course?
  - Why should students want to take this course?
  - How is the course structured?

- Activity 1: Evaluation of Integer Multiplier

- Case Study: Scalar vs. Vector Processors
  - Example design-space exploration
  - Example real ASIC chips

- Activity 2: Brainstorming for Sorting Accelerator
In its broadest definition, computer architecture is the design of the abstraction/implementtion layers that allow us to execute information processing applications efficiently using available manufacturing technologies.
In its broadest definition, computer architecture is the design of the abstraction/implementation layers that allow us to execute information processing applications efficiently using available manufacturing technologies.
In its broadest definition, computer architecture is the **design of the abstraction/implementation layers** that allow us to execute information processing **applications** efficiently using available manufacturing **technologies**.
Key Metrics in Computer Architecture

- **Primary Metrics**
  - Execution time (cycles/task)
  - Energy (Joules/task)
  - Cycle time (ns/cycle)
  - Area ($\mu m^2$)

- **Secondary Metrics**
  - Performance (ns/task)
  - Average power (Watts)
  - Peak power (Watts)
  - Cost ($)
  - Design complexity
  - Reliability
  - Flexibility

Discuss qualitative first-order analysis from ECE 4750 on board
Unanswered Questions from ECE 4750

- How can we quantitatively evaluate area, cycle time, and energy?
- How do we actually implement processors, memories, and networks in a real chip?
- How should we implement/analyze application-specific accelerators?
  - Very loosely coupled memory-mapped accelerators
  - More tightly coupled co-processor accelerators
  - Specialized instructions and functional units
ASIC: Application-Specific Integrated Circuit

- **Complex Digital ASIC Design**

**Activity 1**
Case Study: Scalar vs. Vector Processors

**Activity 2**

**Network**

- Accelerated Instructions
- Superscalar w/ Deeper Pipelines
- Out-of-Order Superscalar Superpipelined
- Multicore

**Energy (Joules per Task)**

- Simple Proc
- Processor Power Constraint
- Specialized Accelerators

**Performance (Tasks per Second)**

- Superscalar
- Multicore
- Specialized Accelerators

**ECE 5745**
Course Overview
ASIC: Application-Specific Integrated Circuit

- Complex Digital ASIC Design
- Activity 1: Case Study: Scalar vs. Vector Processors
- Activity 2

**Design Performance Constraint**
- Embedded Architectures
- More Flexible Accelerator
- Less Flexible Accelerator
- Custom ASIC

**Energy Efficiency (Tasks per Joule)**
- Simple Processor
- High-Performance Architectures

**Performance (Tasks per Second)**
- Design Power Constraint

- Network
  - Accelerated Instructions
    - P
      - Xcel
        - P
  - D$
  - Network

- Performance vs. Flexibility vs. Specialization
Goal for ECE 5745 is to answer these questions!

- How can we quantitatively evaluate area, cycle time, and energy?
- How do we actually implement processors, memories, and networks in a real chip?
- How should we implement/analyze application-specific accelerators?
  - Very loosely coupled memory-mapped accelerators
  - More tightly coupled co-processor accelerators
  - Specialized instructions and functional units
Full Custom Design vs. Standard-Cell Design

- **Full-Custom Design (ECE 4740)**
  - Designer is free to do anything, anywhere; though team usually imposes some design discipline
  - Most time consuming design style; reserved for very high performance or very high volume chips (Intel microprocessors, RF power amps for cellphones)

- **Standard-Cell Design (ECE 5745)**
  - Fixed library of “standard cells” and SRAM memory generators
  - Register-transfer-level description is automatically mapped to this library of standard cells, then these cells are placed and routed automatically
  - Enables agile hardware design methodology
Standard-Cell Design Methodology

Cells arranged in rows

Generated memory arrays

Clock Rail (not typical)

Power Rails in M1

NAND2

Flip-flop

Cell I/O on M2

Well Contact under Power Rail

Ripple carry adder with carry chain highlighted

Mem 1

Mem 2

Mem

- Cells arranged in rows
- Generated memory arrays
- Clock Rail (not typical)
- Power Rails in M1
- NAND2
- Flip-flop
- Cell I/O on M2
- Well Contact under Power Rail
- Ripple carry adder with carry chain highlighted

Standard-Cell Design

Also called Cell-Based ICs (CBICs)

Fixed library of cells plus memory generators

Cells can be synthesized from HDL, or entered in schematics

Cells placed and routed automatically

Requires complete set of custom masks for each design

Currently most popular hard-wired ASIC type (6.884 will use this)
Standard-Cell Design Methodology

- Design in HDL
  - HDL Simulator
  - Switching Activity
  - Gate-Level Model
- Standard Cells
  - Synthesis
  - Place & Route
  - Layout
- Power Analysis

- Execution Time (cycles/task)
- Area ($\mu m^2$)
- Cycle Time (ns)
- Energy (J/task)
Example Standard-Cell Chip Plot

- Control Processor: 8.1%
- Vector Register File: 56.9%
- Vector Integer ALUs: 9.7%
- Vector FPUs: 9.4%
- Vector Memory Units: 7.6%
- Other: 8.3%
Complex digital ASIC design is the process of quantitatively exploring the area, cycle time, execution time, and energy trade-offs of various application-specific accelerators (and general-purpose proc+mem+net) using automated standard-cell CAD tools and then to transform the most promising design to layout ready for fabrication.
Complex Digital ASIC Design

- Course goal, structure, motivation
  - What is the goal of the course?
  - Why should students want to take this course?
  - How is the course structured?

- Activity 1: Evaluation of Integer Multiplier

- Case Study: Scalar vs. Vector Processors
  - Example design-space exploration
  - Example real ASIC chips

- Activity 2: Brainstorming for Sorting Accelerator
Technology Scaling is Slowing

System Performance

Vacuum Tube
Discrete Transistor
Integrated Bipolar
Integrated CMOS
7nm, ~50B Transistors
New Technology?
Technology Fallow Period?
OR
Golden Age of Chip Design?

ECE 5745 Course Overview 16 / 52
Example Application Domain: Image Recognition

Starfish

Dog
Machine Learning: Training vs. Inference

Training
- Many images
- Model:
  - Forward: "starfish"
  - Backward: error
  - Labels: "dog"

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

- **Top 5 Error Rate**
  - 2010: 28%
  - 2011: 26%
  - 2012: 16%
  - 2013: 12%
  - 2014: 7%

- **Software:** Deep Neural Network
- **Hardware:** Graphics Processing Units

- **Num of Entries Using GPUs**
  - 2010: 0
  - 2011: 4
  - 2012: 60
  - 2013: 110
ASIC Design for ML 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

**Intel Nervana**
- Custom ASIC specifically designed to accelerate machine learning
- Special hardware support for Winograd algorithms
- Collaboration between Intel and Facebook
- Maybe Facebook is also building their own in-house chips?

**Google TPU**
- Custom ASIC specifically designed to accelerate Google’s TensorFlow C++ library
- Tightly integrated into Google’s data centers
- 15–30× faster than contemporary CPU and GPUs
ASIC Design for ML at the Edge

Google Pixel
- Designed custom ASIC called the Pixel Visual Core for image processing
- PVC enables 5x performance improvement in high-dynamic-range imaging

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

Amazon Echo
- Developing AI ASICs so Echo line can do more on-board processing
- Reduces need for round-trip to cloud
- Co-design the algorithms and the underlying hardware
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
The field of complex digital ASIC design is experiencing a disruptive sea change and has a critical choice:

1. A technological fallow period
2. A golden age of ASIC design

This course will help you appreciate and possibly contribute to this golden age!
Course Motivation: Comp Arch Research Perspective

- Complex Digital ASIC Design
- Activity 1: Case Study: Scalar vs. Vector Processors
- Activity 2

ECE 5745 Course Overview

24 / 52
Cross-Layer Interaction is Critical

Need to quantitatively understand area, cycle time, and energy trade-offs to be able to create new revolutionary architectures that tackle the most challenging physical design issues.
Course Motivation: Circuits Research Perspective

Your Digital Circuit Here

Your Analog Circuit Here

Fighter Airplane
~100,000 parts

Intel Sandy Bridge E
2.27 Billion transistors
Cross-Layer Interaction is Critical

Circuit-level researchers need to appreciate the system-level context for their subsystems; cross-layer interaction can generate some of the most exciting research ideas.
Complex Digital ASIC Design

- Course goal, structure, motivation
  - What is the goal of the course?
  - Why should students want to take this course?
  - How is the course structured?

- Activity 1: Evaluation of Integer Multiplier
- Case Study: Scalar vs. Vector Processors
  - Example design-space exploration
  - Example real ASIC chips
- Activity 2: Brainstorming for Sorting Accelerator
Course Structure

Part 1
ASIC Design Overview

Part 2
Digital CMOS Circuits

Part 3
CAD Algorithms

Prereq
Computer Architecture

P P
M M
Part 1: ASIC Design Overview

- Topic 1: Hardware Description Languages
- Topic 2: CMOS Devices
- Topic 3: CMOS Circuits
- Topic 4: Full-Custom Design Methodology
- Topic 5: Automated Design Methodologies
- Topic 6: Closing the Gap
- Topic 7: Clocking, Power Distribution, Packaging, and I/O
- Topic 8: Testing and Verification

---

ECE 5745
Course Overview
30 / 52
Part 2: Digital CMOS Circuits
Part 3: CAD Algorithms

**Topic 12**
Synthesis Algorithms

- **RTL to Logic Synthesis**
  
  \[ x = a'bc + a'bc' \]
  \[ y = b'c' + ab' + ac \]

- **Technology Independent Synthesis**
  
  \[ x = a'b \]
  \[ y = b'c' + ac \]

- **Technology Dependent Synthesis**

**Topic 13**
Physical Design Automation

- **Placement**

- **Global Routing**

- **Detailed Routing**
Five-Week Design Project
Complex Digital ASIC Design

- Course goal, structure, motivation
  - What is the goal of the course?
  - Why should students want to take this course?
  - How is the course structured?

- Activity 1: Evaluation of Integer Multiplier

- Case Study: Scalar vs. Vector Processors
  - Example design-space exploration
  - Example real ASIC chips

- Activity 2: Brainstorming for Sorting Accelerator
Fixed-Latency Iterative Multiplier Datapath

- **b_reg**
- **b_mux_sel**
- **b_lsb**
- **a_reg**
- **a_mux_sel**
- **result_reg**
- **result_mux_sel**
- **result_en**
- **add_mux_sel**
- **resp_msg**
- **req_msg.a**
- **req_msg.b**
- **32b**
- **>>**
- **<<**

Diagram shows the flow of data through the multiplier datapath with multiplexers and shifters.
Complex Digital ASIC Design

- Course goal, structure, motivation
  - What is the goal of the course?
  - Why should students want to take this course?
  - How is the course structured?

- Activity 1: Evaluation of Integer Multiplier

- Activity 2: Brainstorming for Sorting Accelerator

ECE 5745
Course Overview
Scalar Processors with Multithreading

Programmer's Logical View

μT0 → μT1 → μT2 → μT3 → μT4 → μTi

Control Instructions
Memory Instructions
Execution Resources
Architectural Registers

Memory
Scalar Processors with Multithreading

Programmer's Logical View

Vector-Vector Add

```
loop:
  load a, a_ptr
  load b, b_ptr
  add c, a, b
  store c, c_ptr
  a_ptr++
  b_ptr++
  c_ptr++
  branch
```

Masked Filter

```
loop:
  load a, a_ptr
  a_ptr++
  branch a = 0
  op0
  op1
  ...
  branch
```
Scalar Processors with Multithreading

**Programmer's Logical View**

![Diagram of Processor Cores with Multithreading]

**Typical Core Micro-Architecture**

- Instr Memory
- Data Memory
- Multi-threaded Cores

**Energy vs. Performance**

- MIMD

---

ECE 5745 Course Overview 37 / 52
Vector-SIMD Processors

Programmer's Logical View

CT0

elm 0
elm 1
elm 2
elm 3

Memory

Vector-SIMD Arithmetic Instructions
Vector-SIMD Memory Instructions

Architectural Vector Register with 4 Elements

CTj

elem i
Vector-SIMD Processors

Programmer's Logical View

Vector-Vector Add

```
loop:
  vload A, a_ptr
  vload B, b_ptr
  vadd C, A, B
  vstore C, c_ptr
  a_ptr++
  b_ptr++
  c_ptr++
  branch
```

Masked Filter

```
loop:
  vload A, a_ptr
  vset F, A = 0
  vop0 under flag F
  vop1 under flag F
  ...
  a_ptr++
  branch
```
Vector-SIMD Processors

Programmer's Logical View

CT0

elm 0
elm 1
elm 2
elm 3

Memory

Typical Core Micro-Architecture

Instruction Memory

CP

VIU

VMU

Vector Lanes

Data Memory

Energy

MIMD

Vector-SIMD

Performance
Quantitative Area Evaluation

- Control Processor: 8.1%
- Vector Register File: 56.9%
- Vector Integer ALUs: 9.7%
- Vector FPUs: 9.4%
- Vector Memory Units: 7.6%
- Other: 8.3%

[Diagram showing various components and their areas where Vector Register File is highlighted in green.]
Quantitative Area Evaluation

Single-core multi-lane design reduces area by 15%
Multi-core single-lane design increases area by 20%
Quantitative Performance and Energy Evaluation

- **Case Study: Scalar vs. Vector Processors**

**Multithreaded Multicore** (increasing number of threads per core)

- **Single-Lane Vector** (increasing vlen)

Performance reduction with increasing threads due to increased cycle time and thread management overhead on fine-grain loops.

**Quad-Core w/ Vertical Multithreading**

**Vector-SIMD** (4 core w/ 1 lane)
Simple RISC Processor ASIC

- SP Datapath
- SP Regfile
- RAM Subbank (2KB)
- RAM Subbank (2KB)
- RAM Subbank (2KB)
- RAM Subbank (2KB)
- VCO
- AHIP Controller
Simple RISC Processor ASIC

- RISC processor w/ 8 KB SRAM
- TSMC 0.18 μm process
- 1.7 × 2.1 mm
- 610K Transistors
- 450 MHz at 1.8 V
Scale Vector-Thread Processor ASIC
## Scale Vector-Thread Processor ASIC

**TSMC 0.18µm • 7.14 Million Transistors • 16.6 mm² Core Area**

<table>
<thead>
<tr>
<th>XBAR</th>
<th>VMU</th>
<th>Cache Control</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Cache Tag CAMs</td>
</tr>
<tr>
<td></td>
<td></td>
<td>CP</td>
</tr>
<tr>
<td>Cache Data RAM Banks</td>
<td>Vector Lane 0</td>
<td></td>
</tr>
<tr>
<td>Cache Tag CAMs</td>
<td>Vector Lane 1</td>
<td></td>
</tr>
<tr>
<td>Cache Control</td>
<td>Vector Lane 2</td>
<td></td>
</tr>
<tr>
<td>CP</td>
<td>Vector Lane 3</td>
<td></td>
</tr>
</tbody>
</table>

---

**ECE 5745**

**Course Overview**
Scale Energy vs. Performance Results

Power consumption ranges from 0.4 to 1.1 Watts at 260 MHz
**Post-Silicon Evaluation Strategy**

The testing platform enables running small test programs on BRGTC1 to compare the performance and energy of pure-software kernels versus the HLS-generated sorting accelerator.

**Taped-out Layout for BRGTC1**

- 2x2mm 1.3M transistors in IBM 130nm RISC processor, 16KB SRAM
- HLS-generated accelerators
- Static Timing Analysis Freq. @ 246 MHz
Celerity System-on-Chip Overview (2017)

Target Workload: High-Performance Embedded Computing

- 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
Complex Digital ASIC Design

Activity 1

• Case Study: Scalar vs. Vector Processors •

Activity 2

BRG Test Chip 2 (2018)

Block Diagram
4xRV32IMAF cores with “smart” sharing L1$/LLFU, synthesizable PLL

Taped-out Layout for BRGTC2
2x2mm, 1.2M-trans, IBM 130nm
Static Timing Analysis Freq. @ 500MHz
Complex Digital ASIC Design

- Course goal, structure, motivation
  - What is the goal of the course?
  - Why should students want to take this course?
  - How is the course structured?

- Activity 1: Evaluation of Integer Multiplier

- Case Study: Scalar vs. Vector Processors
  - Example design-space exploration
  - Example real ASIC chips

- Activity 2: Brainstorming for Sorting Accelerator
Brainstorming for Sorting Accelerator

- **Software Baseline**
  - Insertion sort $O(n^2)$
  - 20% of dynamic instructions are lw/sw

- **Hardware Accelerator**
  - What is ideal speedup assuming we still use insertion sort?
  - What if we use a different algorithm?
  - Sketch hardware impl of a sorting accelerator
  - Where will accelerator be in the energy/perf space?
**Take-Away Points**

- Complex digital ASIC design is the process of quantitatively exploring the area, cycle time, execution time, and energy trade-offs of general-purpose and application-specific designs using automated standard-cell CAD tools and then to transform the most promising design to layout ready for fabrication.

- Course can provide useful experience with cross-layer interaction for students interested in pursuing research in computer architecture or circuits or students interested in pursuing a career in industry development of ASICS.