ECE 5775
High-Level Digital Design Automation
Fall 2018

Course Overview

Zhiru Zhang
School of Electrical and Computer Engineering
Agenda

- Important logistics
- Course motivation
- More logistics
Class Resources

- **Course website**
  - [http://www.csl.cornell.edu/courses/ece5775](http://www.csl.cornell.edu/courses/ece5775) (tiny.cc/ece5775)
  - Lectures slides, handouts, and other readings

- **Piazza**
  - Announcements and Q&A
  - Enrollment information to come

- **CMS: course management system**
  - Assignments and grades
  - Electronic submissions required
  - Enrollment information to come
Course Texts

- **SYNTHESIS AND OPTIMIZATION OF DIGITAL CIRCUITS** by Giovanni De Micheli
- **Parallel Programming for FPGAs** by Ryan Kastner, Janarbek Matai, and Stephen Neuendorffer
  - E-book available online
  - Draft copy available online
- Select papers, software manuals, & web articles
Seeking Help After Class

- **Piazza**
  - Questions on lectures, assignments, projects, etc.
  - Monitored by course staff

- **Staff email:** ece5775-staff@csl.cornell.edu
  - Grading related questions to instructor (& TA)

- **Instructor email**
  - Personal issues/appointment

- **Instructor office hours**
  - Thursday 5:00-6:00pm, Rhodes 320
Grading Scheme

▸ Class participation (4 pts)
  – Asking & answering questions during lectures
  – Contributions to Piazza forum

▸ Quizzes (8 pts)

▸ Student-led discussions (8 pts)

▸ Midterm exam (20 pts)

▸ Assignments (28 pts)

▸ Final project (32 pts)
This Course Concerns with Hardware/Software (HW/SW) Co-Design

- Specifying algorithms in SW programming languages

- Compiling SW descriptions into specialized HW architectures
  - Automatic compilation & synthesis techniques
  - Performance, area, power trade-offs

- Realizing SW & HW on reconfigurable system-on-chips
  - FPGA implementation
This Course Introduces EDA

EDA: A general methodology for refining a high-level description down to a detailed physical implementation for designs ranging from
- integrated circuits (including system-on-chips),
- printed circuit boards (PCBs),
- and electronic systems

Modeling, synthesis, and verification at every level of abstraction

[source: NSF’09 EDA Workshop]
“This incredible growth rate could not be achieved by hiring an exponentially growing number of design engineers. It was fulfilled by adopting new design methodologies and by introducing innovative design automation software at every processor generation.”
Custom vs. Synthesized Design

Future of Custom Design

Why do we do custom design?
1. Higher Frequency (Interconnect engineering), &
2. Higher density (smaller die)

In the future, frequency will not be limited by interconnect RC
Design rules have become complex, and becoming worse
Design mistake will be a lot more expensive

Claims:
Cutom design will not give frequency benefit
Custom design will not give die size benefit

Hard macros will be short lived!

[source: Shekhar Borkar’s CEDA distinguished lecture at DAC’ 2011]
Rise of the Machines

# of Customs over Time

>10x reduction over 5 generation

Milestone:
Digital Logic in 22nm server class Microprocessors 99% synthesized and signed-off by Gate Level signoff

E-D-A: My Other Interpretation

- **Exponential**
  - in complexity (or Extreme scale)

- **Diverse**
  - increasing system heterogeneity
  - multi-disciplinary

- **Algorithmic**
  - intrinsically computational
Exponential: Moore’s Law

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

[Figure credit: Christopher Batten, Cornell]
Era of Billion-Transistor Chips

- **Apple A11**
  - ~4B transistors (?)

- **Intel Haswell-EP Xeon E5**
  - ~7B transistors

- **IBM Power9**
  - ~8B transistors

- **Oracle SPARC M7**
  - ~10B transistors

- **NVIDIA V100 Pascal**
  - ~21B transistors

- **Intel/Altera Stratix 10**
  - ~30B transistors
Evolution of EDA and Design Abstraction

- Transistor-level entry
- Gate-level entry
- Register-Transfer-Level (RTL)
- McKinsey S-Curve
- EDA tool effort

[source: Kurt Keutzer, UCB]
Diverse Range of Integrated Functionalities

Samsung Galaxy S II, Note, Tab 7.0 Plus, 7.7, ...
End of Dennard Scaling: Power becomes the Key Constraint

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

[Figure credit: Christopher Batten, Cornell]
Power-Constrained Modern Computers

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

- Energy efficiency must improve!
- Limitations of general-purpose multicore scaling
  - Amdahl’s law
  - Dark silicon

<<1W/chip  ~1W/chip  ~15W/chip  ~50W/chip  ~100W/chip  >100W/chip
Inefficiency of General-Purpose Computing

- Typical energy overhead for every 10pJ arithmetic operations
  - 70pJ on instruction supply
  - 47pJ on data supply

- Only 59% of the instructions are arithmetic

[Embedded Processor Energy Breakdown]

- Arithmetic: 70%
- Clock and control: 28%
- Data supply: 24%
- Instruction supply: 6%

[source: Dally et al. Efficient Embedded Computing, IEEE’08]
Ending of Moore’s Law:
A Reality Check of Density Scaling

[Figure credit: David Brooks, Harvard]
Advance of Civilization

▸ For humans, Moore’s Law scaling of the brain has ended a long time ago
  – Number of neurons and their firing rate did not change significantly

▸ Remarkable advancement of civilization via specialization
Computers are Following the Same Path

Modern SoCs integrate a rich set of specialized accelerators
- Speed up critical tasks
- Reduce power consumption and cost
- Increase energy efficiency

[source: Samsung]
Hardware Specialization for Higher Efficiency

[source: Bob Broderson, Berkeley Wireless group]
Best of Times for Specialized Computing

- **Pressing demand** to efficiently accelerate a growing array of datacenter & embedded workloads
  - Multicore performance scaling significantly slowed
  - Pervasive hardware specialization is inevitable?

Microsoft Project Catapult

Amazon EC2 F1 instances
Best of Times for Specialized Computing

Blue Chips
Apple
Google
Intel
Microsoft
...

Startups
Cambricon
Cerebras
Deephi (→ Xilinx)
Graphcore
...

Academia
EIE/ESE [Han ISCA’16, FPGA’17]
Eyeriss [Chen ISCA’16]
FINN [Umuroglu FPGA’17]
Minerva [Reagen ISCA’16]
...

Deep Learning is causing a revolution AI and computer hardware industry
“Worst of Times” Also?

- Target of specialization is constantly & rapidly moving!
  - In particular, machine learning (ML) algorithms and software frameworks are rapidly evolving

Exponential Growth of ML arXiv Papers

Development Effort of Hardware Accelerators

- Comparative study on Monte Carlo option pricing:
  - Data for a single option pricing, using 524,288 simulation paths

<table>
<thead>
<tr>
<th>Platform</th>
<th>Normalized Speed-Up</th>
<th>Normalized Performance/Watt</th>
<th>Development Time in Days</th>
</tr>
</thead>
<tbody>
<tr>
<td>FPGA</td>
<td>545:1</td>
<td>1090:1</td>
<td>60</td>
</tr>
<tr>
<td>GPU</td>
<td>50:1</td>
<td>21:1</td>
<td>3</td>
</tr>
<tr>
<td>GPP</td>
<td>1:1</td>
<td>1:1</td>
<td>1</td>
</tr>
</tbody>
</table>


Conventional hardware design practice requires extensive hand-coding in RTL and manual tuning
**RTL Verilog vs. Untimed C**

**RTL Verilog**

```verilog
module dut(rst, clk, q);
    input rst;
    input clk;
    output q;
    reg [7:0] c;

    always @ (posedge clk)
    begin
        if (rst == 1b'1)
            c <= 8'b00000000;
        else
            c <= c + 1;
    end

    assign q = c;
endmodule
```

**C**

```c
uint8 dut() {
    static uint8 c;
    c += 1;
}
```

High-Level Synthesis

An 8-bit counter
High-Level Design Automation to Manage Design Complexity

- Significant code size reduction
- Shorter simulation/verification cycle

[Diagram showing the process of high-level design automation with code size reduction and simulation speed improvement, along with the logic synthesizer, high-level synthesis, and RTL-based synthesis stages.]

[source: Wakabayashi, DAC’05 tutorial]
Algorithms Drive Automation

Key Algorithms in EDA
[source: Andreas Kuehlmann, Synopsys Inc.]
Course Organization

- Refer to syllabus (pdf) for course organization details

Course Syllabus
ECE 5775 High-Level Digital Design Automation
Fall 2018, Tuesday and Thursday 11:40am-12:55pm, Phillips 403

1. Course Information

Lectures: TuTh 11:40am-12:55pm, 403 Phillips Hall
Website: http://www.csl.cornell.edu/courses/ece5775
CMS: https://cmsx.cs.cornell.edu
Piazza: http://piazza.com/cornell/fall2018/ece5775/home

Instructor: Zhiru Zhang, zhiruz@cornell.edu
Office Hours: Thursday 5:00-6:00pm, 320 Rhodes Hall
Staff Email: ece5775-staff@csl.cornell.edu

Course Texts:
- Lecture slides/notes on course website

Supplementary Materials:
- Additional reference papers will be posted as a course reader.
Preferred Background

- Working knowledge of the following at undergraduate level
  - C/C++
  - Digital logic and basic computer architecture concepts (e.g., adders, clock, registers, pipelining)

- Experiences with the following would increase appreciation & productivity
  - Algorithms and data structures
  - RTL design for FPGA or ASIC
Learning Outcomes

▸ High-level digital design methodologies
  – Designing above register transfer level (RTL)
  – Building application-specific accelerators with **C-based design** flow

▸ High-level automation algorithms
  – Fundamentals of **high-level synthesis**
    • Scheduling, resource sharing, pipelining, etc.
  – Useful **combinatorial optimization** techniques
    • Graph algorithms, dynamic programming, greedy algorithms, integer linear programming, etc.
Course Roadmap

▸ Lecture and paper discussion sessions
  – **Background**
    • Introduction
    • Specialized computing
    • Algorithm review
  – **High-level synthesis**
    • C-based synthesis
    • Compiler IR and LLVM
    • Scheduling
    • Resource sharing
    • Pipelining
  – **More advanced topics**
    • Deep learning acceleration
    • Domain-specific programming

▸ Midterm
▸ Project meetings
▸ Final presentation
Exams and Quizzes – 28%

▶ In-class midterm (20%)
  – Open notes & open book
  – When: Thursday Oct 18th
  – No sit-down final

▶ Quizzes (8%)
  – You will need to answer pop quiz questions in most lectures (using paper & pencil)
  – TWO lowest scores will be dropped
Student-Led Discussions – 8%

▸ Each student is expected to lead the discussion of a course-related topic
  – 2-3 sessions in total
    • First session is on October 11th
    • Up to 4 talks per session
    • Audience are expected to read book/papers and ask questions
  – 3-4 students / talk, depending on class size
  – Popular presentations will earn bonus points
Assignments – 28%

- Two problem sets (8%)

- Four lab assignments (20%)
  - Design & programming assignments leveraging high-level synthesis tools and software compilers
  - Experiments to be conducted on ecelinux servers
    - `% ssh -X <netid>@ecelinux-01.ece.cornell.edu`
    - Necessary tools will be installed in common directories
Final Project – 32%

- In-depth exploration of a research topic
  - Designing new applications with high-level design automation tools
  - OR
    - Devising new automation methods/algorithms
  - 3-4 students / team, depending on class size

- Timeline
  - Proposal due after midterm
  - Weekly meeting with the instructor to track progress
  - Presentation before the final week
  - Final project report due by the final exam date
Past Projects: 3 Major Themes

- **App**: New hardware accelerators of compute-intensive applications
- **Arch**: New architectural mechanisms for hardware specialization
- **Tool**: New high-level design automation techniques

Example from Fall’17 (App):
Fast Super-Resolution Convolutional Neural Net (FSRCNN)
High-Level Synthesis Tool

In-class tutorial on Xilinx Vivado HLS on Tuesday 8/28 (Led by TA)
Local FPGA Cluster

- For labs and project, we will use Zynq-based FPGA development boards (ZedBoard and ZC-706)
  - FPGA + Dual-core ARM Cortex-A9
  - Boot Linux
  - Support audio and 1080p video
Cloud FPGA Platforms

- For final project, students can also choose to explore cloud FPGA platforms using the AWS F1 instances.
Summary

- Exponential growth in silicon capacity calls for higher level of **design abstraction**

- End of Dennard scaling leads to increasing **specialization** to sustain improvement in hardware performance and energy efficiency

- EDA tools are fueled by highly sophisticated and yet scalable CAD **algorithms**

  In this class we aim to understand and tackle many of these technological issues and challenges
Before Next Class

- **Actions**
  - Check out the course website
  - **Read through the course syllabus**
  - **Verify your login on ece-linux**
    - `ssh -X <netid>@ecelinux-01.ece.cornell.edu`

- **Next topic: Vivado HLS Tutorial**
  - **Bring your laptop!**
Acknowledgements

- These slides contain/adapt materials developed by
  - Prof. Jason Cong (UCLA)