Welcome to the FPGA World!

Cong (Callie) Hao
callie.hao@ece.gatech.edu

Assistant Professor
ECE, Georgia Institute of Technology

Sharc-lab @ Georgia Tech https://sharclab.ece.gatech.edu/
Introduction

- **Instructor: Cong (Callie) Hao**
  - Ph.D. in Japan, Waseda University (2017)
  - Postdoc in UIUC (2018~2020)
  - Postdoc in GT (2021~2022 waiting for my visa...)
  - Assistant Professor (Jan. 2020)

  Email: callie.hao@ece.gatech.edu
  Homepage: [http://callie.ece.gatech.edu](http://callie.ece.gatech.edu)

- **Sharc-Lab**
  - FPGAs
  - Domain specific accelerators
  - Hardware-aware machine learning
  - Electronic design automation
  - Graph and graph neural network
Course Information

• Course Info
  o Location: Van Leer C457
  o Time: 3:30 pm - 4:20 pm, Tue & Thu.
  o Office Hours:
    - Tuesday 4:30 – 5:30 PM after class
    - In my office, Klaus 2306

• TA Info
  o Lakshmi Sathidevi
  o Email: lsathidevi3@gatech.edu

• Course Schedule
  o https://sharclab.ece.gatech.edu/teaching/2022-spring-fpga/
  o Not finalized: may change based on our progress

• Canvas
  o Announcements, lecture notes, lab assignments and submission

• Piazza
  o Post common questions on Piazza first before emailing TAs/me.
    - Related to labs and lectures, especially bugs you encountered in your project
    - Highly encouraged
    - I probably won’t have enough time to help you debug – sorry 😞
  o Try to answer each other’s posted questions
    - Highly encouraged and appreciated!
Why do we need this course?

- **Specialized** High-Efficiency Computing!
- Why specialization?
  - Power constraint of modern computers

<< 1W ~ 1W ~ 15W ~ 50W ~ 100W ~ 100W

[Image credit]: Prof. Zhiru Zhang @ Cornell
Why do we need this course?

- **Specialized** High-Efficiency Computing!
- Why specialization?
  - Power constraint of modern computers
  - Inefficiency of general–purpose computing

---

Embedded Processor Energy Breakdown

- Arithmetic: 24%
- Clock and control: 6%
- Data supply: 70%
- Instruction supply: 28%
- 42%

*Source: Dally et al. Efficient Embedded Computing, IEEE'08*
Why do we need this course?

- **Specialized High-Efficiency Computing!**
- Why specialization?
  - Power constraint of modern computers
  - Inefficiency of general–purpose computing
  - Data and computation explosion (big data, AI)

---

Compute Demands During Training

[Bianco, IEEE Access 2018]

https://openai.com/blog/ai-and-compute/
Why do we need this course?

• **Specialized** High-Efficiency Computing!
• **Why specialization?**
  o Power constraint of modern computers
  o In-efficiency of general–purpose computing
  o Data and computation explosion (big data, AI)
  o **Real-time processing requirement**
  o …

[Bianco, IEEE Access 2018]  Images per second [FPS]
Why do we need this course?

- **Specialized** High-Efficiency Computing!
- Why specialization?
  - Power constraint of modern computers
  - Inefficiency of general-purpose computing
  - Data and computation explosion (big data, AI)
  - Real-time processing requirement
  - ...

- **FPGA** is the perfect device for specialization!
- But...?

Verilog is scary...
Why do we need this course?

- **Specialized** High-Efficiency Computing!
- Why specialization?
  - Power constraint of modern computers
  - In-efficiency of general-purpose computing
  - Data and computation explosion (big data, AI)
  - Real-time processing requirement
  - ...

- **FPGA** is the perfect device for specialization!
- But...?

- But!

  C
  C++

  Verilog is scary...

High-Level Synthesis (HLS) saves the world!
What to expect from this course?

- What is FPGA, and how to use and program (a real) FPGA
- What is High-Level Synthesis (HLS), and how to use HLS to program FPGA
- What is domain-specific accelerator, and how to design a good accelerator
- Get a flavor of state-of-the-art research problems
What to expect from this course?

- **What is FPGA, and how to use and program (a real) FPGA**
  - But we don’t have enough boards (sorry... 😞) so you’ll have to share or use remote ones
  - You may also buy one (Pynq-Z2) if you’re really want to touch it

*Figure source: embeddedrelated.com*
What to expect from this course?

- **What is High-Level Synthesis (HLS), and how to use HLS to program FPGA**
  - No Verilog required – although we will have one lecture to briefly introduce Verilog
  - HLS is C/C++ based

---

### What to expect from this course?

For (i=1; i<=c;)

```
    a = a++;
    b = x*2-a;
    a = y+b/3;
```

---

**HLS Tools**

- **C / C++, Chisel, …**
- **Verilog, VHDL, …**
What to expect from this course?

- **What is domain-specific accelerator, and how to design a good accelerator**
  - Principles and best practices of designing a good accelerator
  - Use HLS to accelerate something fun: deep neural network, image processing, etc.

---

Efficiency (high performance)

Flexibility (easy programming)

Intel’s 12th Gen “Alder Lake” 10nm Desktop CPU

NVIDIA RTX A6000 Workstation Graphics Card (in my lab)

Xilinx Alveo U280 Data Center Accelerator Card

Google’s Tensor Processing Unit

Callie Hao | Sharc-lab @ Georgia Institute of Technology

https://sharclab.ece.gatech.edu/
What to expect from this course?

- What is domain-specific accelerator, and how to design a good accelerator
  - Principles and best practices of designing a good accelerator
  - Use HLS to accelerate something fun: deep neural network, image processing, etc.

Deep Neural Network (DNNs)  
Graph Neural Networks (GNNs)  
Signal Processing, Communication, Networking  
Scientific Computing, Simulation
What to expect from this course?

• Get a flavor of state-of-the-art research problems
  o Software/hardware co-design
    - Neural Architecture Search (NAS), auto-ML
  o More advanced applications to accelerate

Manually designed DNN (GoogleNet)  
Searched by NAS

Grading

- **Labs: 30 pt**
  - Lab 1: 10 pt: HLS 101
  - Lab 2: 10 pt: More advanced HLS techniques
  - Lab 3: 10 pt: Program your FPGA!

- **Paper presentation: 10 pt**

- **Final project: 60 pt (almost all about project)**
  - Project proposal: 10 pt
  - Mid-term report: 5 pt
  - Source code: 15 pt
  - Project presentation: 10 pt
  - Final report: 20 pt
    - Clearly state everyone’s contribution

- **A: > 90 pt; B: 80 ~ 90 pt; C: < 80pt; D: fail to submit final project**
Overview of Labs

• Three labs (10 pt each – 30 pt) – all individual assignments
  o Lab submission will include:
    - Your source code
    - Executable file on board (if applicable)
    - A report
  o Lab 1: Getting to know HLS fundamentals
    - Basic usage and practice of Vitis HLS coding
      • Implement an HLS accelerator for a matrix multiplication with basic pipelining, loop unrolling, and array partition
  o Lab 2: Getting things done on FPGA
    - Learn C/RTL co-simulation
    - Basic usage of Pynq/Ultra96 FPGA boards
      • Run your FPGA kernel on board and control it from Jupyter Notebook
  o Lab 3: Getting creative and have fun
    - Advanced HLS optimization for machine learning kernels
      • Implement an HLS accelerator for a convolution network
        • You may choose to implement one layer, or two consecutive layers
        • Implement the whole DNN is a plus
        • Learn to use more advanced techniques such as streaming, line buffer, and anything else you can think of
Overview of Final Projects

• **Final Project (60 pt) – work in groups: up to 3 people each group**
  
  o **Project proposal: 10 pt**
    - Project proposal: describe what topic you want to work on and explain why it worth doing
  
  o **Mid-term report: 5 pt**
    - Should discuss your progress, novel techniques, problems you run into, and future plans
  
  o **Source code: 15 pt**
    - Submit your final source code
    - Must be executable on-board (depending whether the board is accessible)
  
  o **Project presentation: 10 pt**
    - Up to 15 minutes
  
  o **Final report: 20 pt**
    - Should be in a conference paper format

• **Paper presentation: state-of-the-art work related to your selected topic**
  
  o Up to 15 minutes each group
Possible Project Topics

- **Acceleration for ResNet + Faster RCNN (what we are doing right now)**
  - Need to understand the structure of ResNet and Faster RCNN model for object detection
  - Lead: Akshay and Priyal
- **Graph Neural Network (GNN) Acceleration (what we are doing right now)**
  - Need to understand GNN model structure and its computation in PyTorch
  - Lead: Rishov and Stefan
- **Homomorphic encryption**
  - Need to know basic algorithms for homomorphic encryption and how they are implemented
  - Lead: Jianming
- **Robotics**
  - Perception, localization, motion planning, and control algorithms in robotics
  - Lead: Zishen
- **Tensor Network**
  - Tensor singular value decomposition (tensor SVD) and other tensor algorithms
- **More…**

We will introduce these projects in detail soon in a later lecture
Possible Project Topics

• More choices on your own! Please be creative and bold!
  o Any algorithm/application you would like to accelerate (robotics, aerospace, communication, networking, etc.)
    - But must be worth accelerating and novel

• Submit a project proposal first so we can discuss as early as possible
  o Start to think as early as possible!
  o Talk to: Akshay, Priyal, Rishov, Stefan, Jianming, and Zishen

• Hopefully every group can submit a good paper! 😊
  o FPGA, FCCM, FPL, DAC, ICCAD, ASAP, DATE, …
Some Warnings...

- **Programming-heavy**
  - You may run into **extremely annoying bugs** that are hard to debug
    - We’ll all have desperate moments – Why HLS crashed? Why is FPGA not working? Why my accelerator output is wrong? Why my optimization results in slowdown? Etc.
  - The FPGA boards are limited because of chip shortage (sorry again) so you have to share boards or remotely access
    - I know this adds extra difficulty...

- **Project-oriented**
  - You will spend most of your time working on “research-like” projects
    - Read many papers
    - Define your problem
    - Try to solve it and read more papers
    - Struggle with your project, many trial-and-errors
  - You will also need to write a nice “paper-like” reports and do “conference-like” presentations

- **Paper presenting**
Course Schedule & Syllabus

• Flexible and adjustable – depending on what you are interested
  o Let me know in advance, and I can try to include in a later lecture!

• Less lecture, more practice (on your own)
  o But me and TA would love to help 😊

• Involves some paper reading and project presenting
<table>
<thead>
<tr>
<th>Content</th>
<th>Labs</th>
</tr>
</thead>
<tbody>
<tr>
<td>Jan. 11  Course Introduction</td>
<td></td>
</tr>
<tr>
<td>Jan. 13  Overview of domain-specific accelerators</td>
<td></td>
</tr>
<tr>
<td>Jan. 18  Introduction to FPGA</td>
<td></td>
</tr>
<tr>
<td>Jan. 20  Introduction to Verilog</td>
<td></td>
</tr>
<tr>
<td>Jan. 25  Introduction to HLS</td>
<td>Lab 1 release</td>
</tr>
<tr>
<td>Jan. 27  Vitis HLS Tutorial (bring your laptop)</td>
<td></td>
</tr>
<tr>
<td>Feb. 1   Data precision and quantization</td>
<td></td>
</tr>
<tr>
<td>Feb. 3   Loop optimizations</td>
<td></td>
</tr>
<tr>
<td>Feb. 8   Loop and memory optimizations</td>
<td>Lab 1 due</td>
</tr>
<tr>
<td>Feb. 10  Machine learning overview (DNN and GNN)</td>
<td>Lab 2 release</td>
</tr>
<tr>
<td>Date</td>
<td>Content</td>
</tr>
<tr>
<td>------------</td>
<td>---------------------------------------------------</td>
</tr>
<tr>
<td>Feb. 15</td>
<td>Convolution and GEMM</td>
</tr>
<tr>
<td>Feb. 17</td>
<td>Data streaming and C/RTL co-simulation</td>
</tr>
<tr>
<td>Feb. 22</td>
<td>More advanced DSP techniques</td>
</tr>
<tr>
<td>Feb. 24</td>
<td>Sorting algorithms on FPGA</td>
</tr>
<tr>
<td>Mar. 1</td>
<td>Winograd, FFT, and systolic array</td>
</tr>
<tr>
<td>Mar. 3</td>
<td>Project Introduction (with guests)</td>
</tr>
<tr>
<td>Mar. 8</td>
<td>FPGA accelerator for CNNs</td>
</tr>
<tr>
<td>Mar. 10</td>
<td>FPGA accelerators for GNNs</td>
</tr>
<tr>
<td>Mar. 15</td>
<td>Proposal Presentation</td>
</tr>
<tr>
<td>Mar. 17</td>
<td>Software-hardware Co-design</td>
</tr>
<tr>
<td>Mar. 22 (spring break)</td>
<td>--</td>
</tr>
<tr>
<td>Mar. 24 (spring break)</td>
<td>--</td>
</tr>
</tbody>
</table>
## Course Schedule & Syllabus

<table>
<thead>
<tr>
<th>Date</th>
<th>Content</th>
<th>Labs</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mar. 29</td>
<td>Paper Presentation (3-4 groups)</td>
<td>Lab 3 due</td>
</tr>
<tr>
<td>Mar. 31</td>
<td>Paper Presentation (3-4 groups)</td>
<td></td>
</tr>
<tr>
<td>Apr. 5</td>
<td>Paper presentation (3-4 groups)</td>
<td></td>
</tr>
<tr>
<td>Apr. 7</td>
<td>Project time...</td>
<td></td>
</tr>
<tr>
<td>Apr. 12</td>
<td>Project time...</td>
<td>Mid-term report due</td>
</tr>
<tr>
<td>Apr. 14</td>
<td>Project time...</td>
<td></td>
</tr>
<tr>
<td>Apr. 19</td>
<td>Project time...</td>
<td></td>
</tr>
<tr>
<td>Apr. 21</td>
<td>Project time...</td>
<td></td>
</tr>
<tr>
<td>Apr. 26</td>
<td>Project time...</td>
<td></td>
</tr>
<tr>
<td>Apr. 28</td>
<td>Project time...</td>
<td></td>
</tr>
<tr>
<td>May. 3</td>
<td>Final project presentations (2:40 - 5:30 PM)</td>
<td>Final report and code submission due</td>
</tr>
</tbody>
</table>
Recommended Materials

• A lot of useful materials can be found here:
  o https://sharclab.ece.gatech.edu/materials/

• Textbook:
  o Parallel Programming for FPGAs

• Vitis (Vivado) HLS Written Tutorials:
  o 256 pages but very reader friendly. No need to read from the beginning but can check it whenever needed: https://www.xilinx.com/support/documentation/sw_manuals/xilinx2020_1/ug871-vivado-high-level-synthesis-tutorial.pdf

• Vitis Examples:
  o https://github.com/Xilinx/Vitis-HLS-Introductory-Examples
**Prerequisites**

- A little bit Verilog would help, but not mandatory

- **Good C programming**
  - Especially required for your final project

- **Familiar with PyTorch is a plus**
  - A plus for lab 2 and lab 3 if you could verify your code functionality by cross-checking with PyTorch model (e.g., for CNNs)
  - Also good for your final project if you’re implementing a real-world algorithm
    - Usually the original implementation is in PyTorch or Python
    - You need to translate into C first in order to use HLS