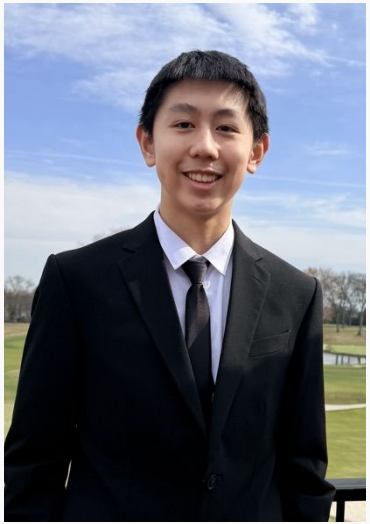
Using CaST for FPGA Design

Charles Yu, Abraham Klein, Rishabh Ramaswamy, Dhruv Ramaswamy

Meet the Team

Rishabh Ramaswamy



Charles Yu

Abraham Klein

Dhruv Ramaswamy

Project Goals

- Run neural network on FPGA to determine inferencing capabilities
- Show CaST proof of concept

01

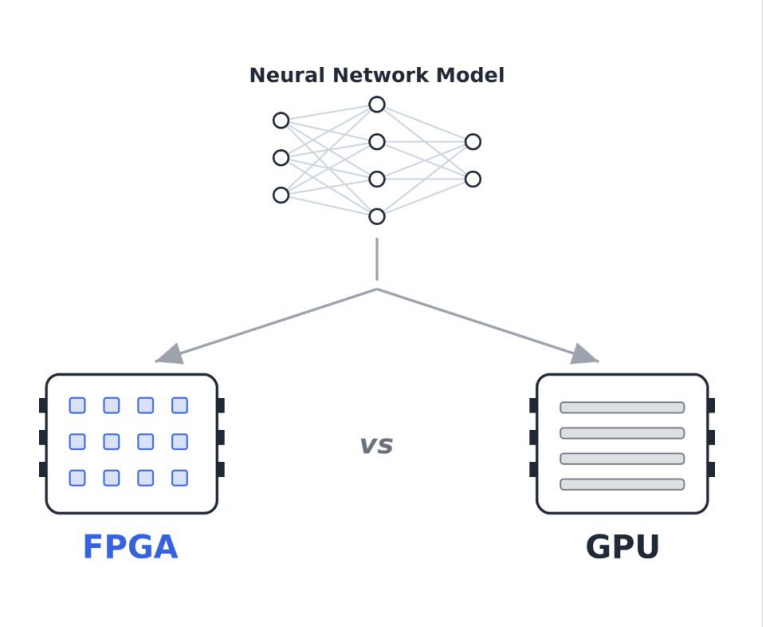
FPGA Introduction

What is an FPGA?

- Type of Computer Chip
- Highly Customizable
- Deterministic (constant) latency
- Energy efficient

What we are using FPGAs for

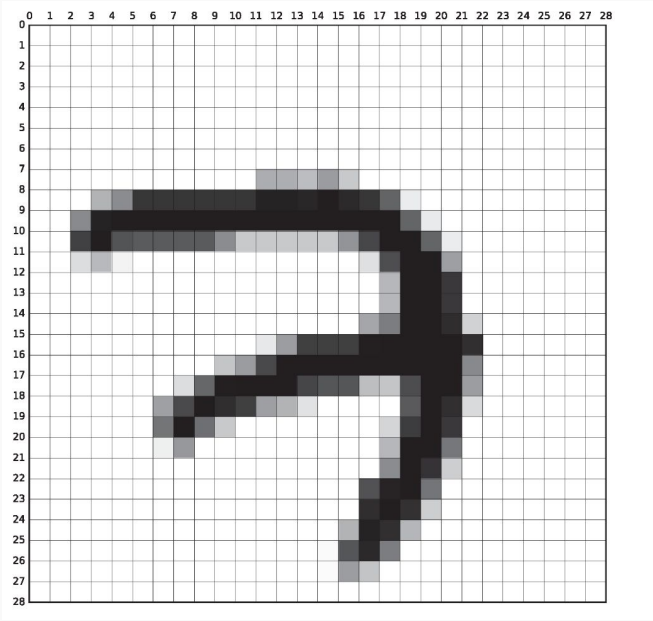
- FPGA vs traditional hardware
- Testing using Neural Network Models
  - MNIST
  - WFNET



02

Our Neural Network Models

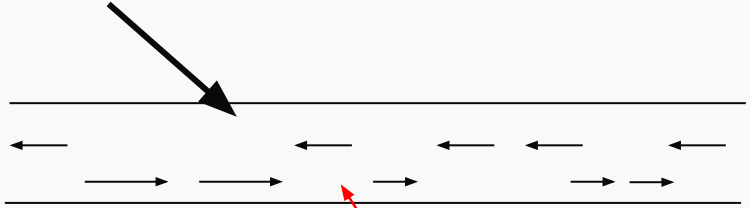
MNIST

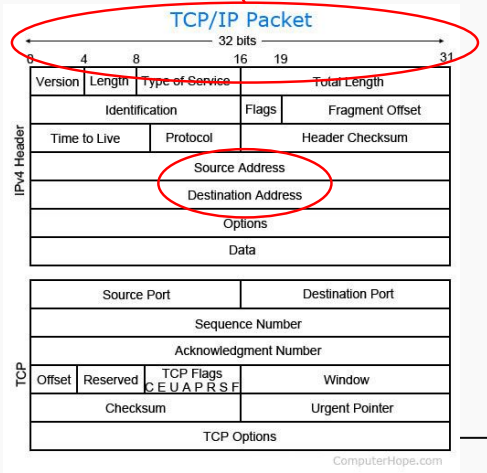


7

WFNET CNN

Trained on Packets



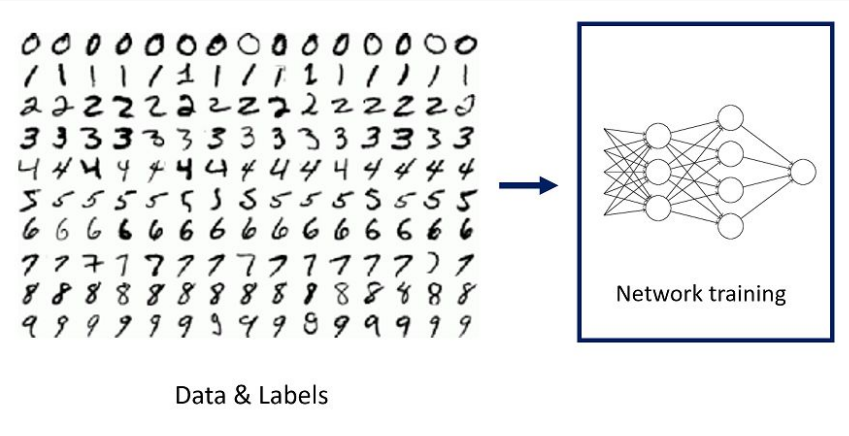


03

Training Our Models

MNIST

- Training goal: get accurate weights + biases
- Training data = MNIST database
- Think of it like tuning a guitar



WFNET (v1 and v2)

V1

- Trained on 2022 data for 20 websites
- Limitations:
  - Old data → websites change over time, accuracy lowers
  - Biggest layer has 84% of weights, hard to port to hardware

V2

- Inspired by v1, built from scratch for 8 websites
- Tries to fix limitations
  - Newly collected data
  - Smaller model, easier to port to FPGA

04a

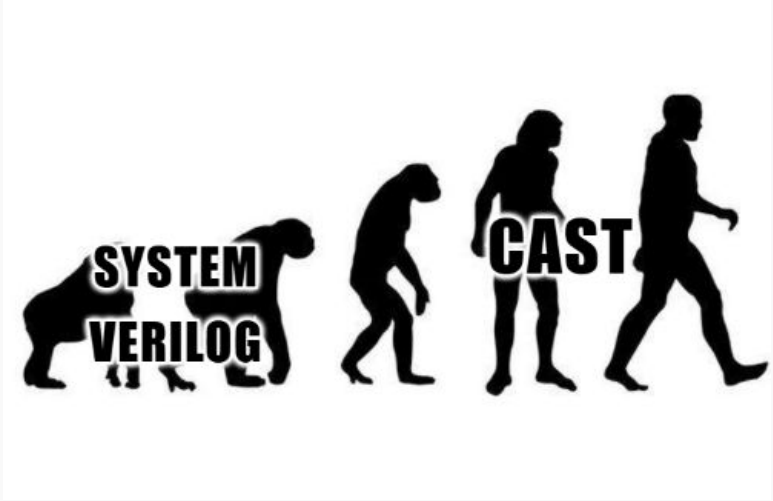
Running our Models - MNIST

System Verilog

- HDL (Hardware Description Language)
- Can be difficult to use
  - Testbench is required
  - Driven by a clock
  - No loops
- Used Verilator to simulate programs

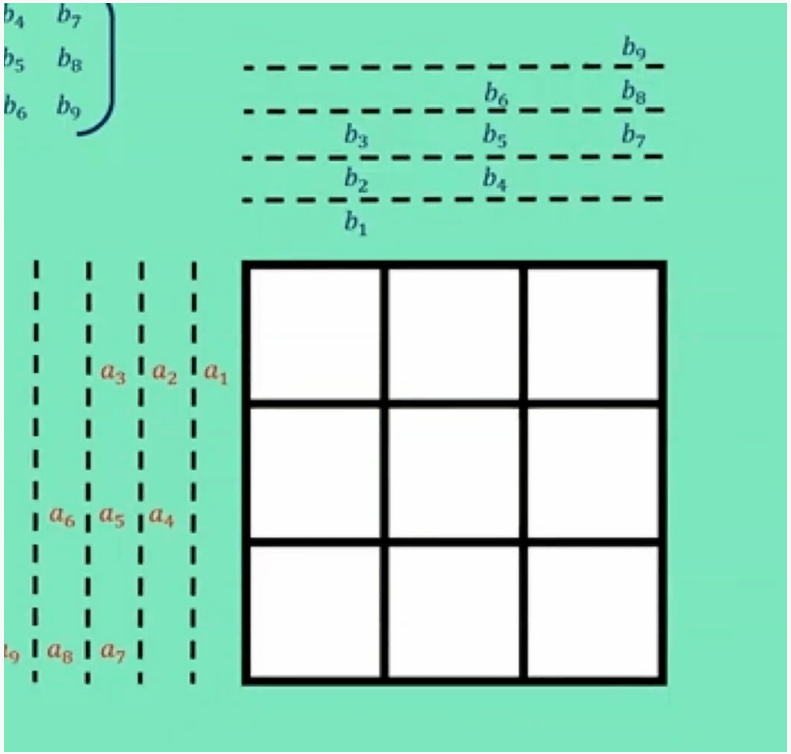
CaST (Channels and State)

- System Verilog: old standard
- CaST: new, high-level alternative
  - Better readability
  - Easier to learn
- CaST transpiler:
  CaST -> System Verilog



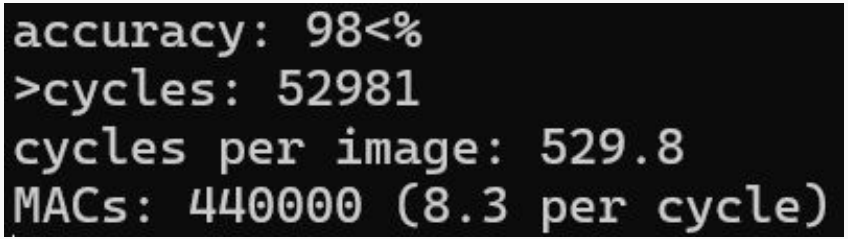
Systolic Matrix Multiplication

- Neural networks are built from giant matrix multiplications
- Much faster and energy efficient
- Solves data movement problem



Results for MNIST Model

- Accuracy of 98% could be improved if we switched to CNN
- Cycles per image is constant → benefit of FPGA
- Proved CaST viability



04b

Running our Models - WFNET

Process of putting Model on FPGA

- Not as easy as running on GPU, the standard approach
- Putting neural network on FPGA is very tedious
  - System verilog, DMA (direct memory address), ton of tools
  - Compilation takes forever

Results

- Were able to put WFNETv1 on FPGA
  - CPU inference (M5 chip) wins
- Design flaws
  - Need to be more parallel
  - Massive layer translates poor to hardware
- With proper optimization, should be about the same inference time

U200 inference

```
kernel time      : 1171.598 ms for 1200 images
throughput       : 1024.2 inferences/s
latency/image    : 976.3 us (batch-amortised)

accuracy         : 82.42% (989 / 1200)
```

Macbook M5 CPU inference

config	1000 inferences	ms/inference	inferences/sec
batch 1, 1 thread	0.117 s	0.117	8,581

05

Future Work

Future Work

- Integrate CaST into more complex models
- Expand WFNETv2 (more data)

Questions?