# **DIGITAL CHIP FABRICATION**

### SILICON PROVEN LEAKY INTEGRATE & FIRE NEURON

TEAM: SDMAY23-28

CLIENT & ADVISOR: DR HENRY DUWE

### **Our Team**



Katherine Gisi	Fulai Zhu	Tyler Green	William Zogg	Aaron Sledge
EE	EE	CprE	CprE	EE
Team Management System Organization Design Backup	PCB Research Bring-up Plan	Software Tools Environment SNN Design / Implementation	SNN Design / Implementation	Software Tool Research Caravel Tutorial Document

# INTRODUCTION

### **Project Overview**

### **Problem Description & Location**

At ISU, it is rare to find undergraduate students in computer or electrical engineering that have created, fabricated, and brought-up a digital ASIC.

• Traditionally large barrier to entry

#### **Client's Goals**

Dr. Duwe is investigating opportunities for a co-curricular where students fabricate and physically implement digital chip designs.

### **Our Project Objectives**

Design and silicon-prove an ASIC.

Compose a bring-up plan for our design.

Develop documentation for future students.



### **Background & Hardware**



### Manufacturing EFABLESS

- Doped Silicon Composes Transistors
- Transistors Compose Logic
- Efabless Organizes all the components

#### Process Design Kit SKYWATER

- Logic Combinations create Standard Cells
- Standard Cells are Spec-ed for Fabrication
- Specs Distributed as Process Design Kit (PDK)



#### https://efabless.com/open\_shuttle\_program

### **Background & Hardware**



#### Hardware Description CARAVEL HARNESS / USER AREA

- Verilog Defines User Functionality
- Wrapper to Standardize Designs





### ASIC Flow OPENLANE

- Map Verilog to Netlist
- Place Cells, Connect Power/Clock
- Define Fabrication Masks

https://efabless.com/open\_shuttle\_program

# ABOUT OUR PROJECT

### **SNN Model Context**

### Machine Learning in Medical Imaging

- Identify diseases, injuries, tumors, etc...
- Speed up a lengthy process
- "ML (...) will have a greater influence in the future" [1]

### The SNN

- "Bridge gap between Neuroscience and ML"<sup>[2]</sup>
- Closer to biology
- Typically, lowerpower
- Discrete electrical signals (digital)
- Accessible and scalable with MNIST data set

Use Case 2. Functionality as an ASIC design Wow this handwriting recognition software is to stars. Present Images Interface with Onboard CPU Grabs data assign 8 ZZ a a La

1 Wieglet 1 Connection

Exam

t0

at

EASIC E

More ASIC

Designs

Soft ware

Functionality as an Example for Future design

Sr. Design 202\*

For

SNN Accelerator Design Context

HARDWAR

4

Use Case 2.

community

EDde REPO

our project outcomes

Outout

al is a d

Document

### Spiking Neural Network Description

Neuron Model Leaky Integrate and Fire

### **Network Model**

snnTorch

- Allows easy network design and testing
- Allows us to export weights

# snnTorch



"Training Spiking Neural Networks Using Lessons From Deep Learning", snntorch.readthedocs.io

### Application

### **Classifying Handwritten Digits**

#### **MNIST Dataset**

Training Set 60,000 images (28x28) - labels (0-9) Testing Set 10,000 images

Reason for Using Handwritten Digits Simple application Size constraints

3 3 3 3 4A ч SS Π Ð 

https://www.researchgate.net/figure/A-sample-of-the-MNISThandwritten-digits\_fig3\_339204564

### **Spiking Neural Network**



Rate coding

Weights

# SYSTEM ARCHITECTURE

### **Design Overview**

#### Generate

- Rate-encoded spikes
- Repeat for all pixels

#### Load Spikes/Weights

- Data movement States
- Send to and from Neuron Unit

### Integrate/Leak/Fire

- Build up Membrane voltage
- Repeat for all pixels
- Leak each time step
- Fire according to Vmem at neuron ID
- Repeat for all Membrane Voltages

### Wait/Time Check

- Controls Beginning and End of process
- Wait is default state



#### Wishbone Feed

- Pixel values <- MNIST</li>
- Weights <pretrained snnTorch network
- Addresses <- CPU</li>

#### Randomizer

- Pseudo random no. generator
- Seed with image values

#### Spike Queue

 FIFO holder for Spikes



Spiking Neuron

Neuron

• LIF hardware model



**Spiking Neuron** 

Neuron

LIF hardware model



### **Neuron Unit**

Equations Describing Neuron Behavior  $U[T] > Uthr \Rightarrow S[T+1] = 1$   $U[t+1] = \beta U[t] + Iin [t+1] - RUthr$  $U[t+1] = \beta U[t] + Isyn [t+1] - R(\beta U[t] + Iin [t+1])$ 

I/O Diagram





# DESIGN OUTCOMES

Work Accomplishments

### **Goal 1 – SNN Submission**

### **Client Request** To silicon prove a chip design.

#### **Product to be Delivered**

Silicon-proven neuron for a spiking neural network.

#### **Contents of Product**

Silicon-proven SNN neuron ready for fabrication.

Renderings of the SNN Control Logic & Prechecks Passed ->



No DRC Violations found {{Klayout Pin Label Purposes Overlap {{STEP UPDATE}} Executing Check 14 o in CUSTOM klayout\_gds\_drc\_check run: klayout -b -r /home/tjgreen/mpw \_project\_wrapper -rd report=/home/tj /git/sdmay23-28/precheck\_results/29\_ 29/logs/klayout\_zeroarea\_check.log No DRC Violations found {{Klayout ZeroArea CHECK PASSED}} Th {{FINISH} Executing Finished, the f {{SUCCESS} All Checks Passed !!! Generating LALR tables WARNING: 183 shift/reduce conflicts 05:54:22 Sat Apr 29 tjgreen@tg-arch \$



### **Goal 1 – Final Results**

### **Simulation of Full Design**

- 9 x 9 image
- Simulated as signals sent by CPU
- 83% snnTorch accuracy translates to 77% accuracy
  - Less precision (floating point to 8-bit fixed point)

∽sst	Signals	Waves		
∽ 📩 snn_simple_integration_te	s Time	JS 40	0 us 50	00 us 600
- ∨ <b>du</b> t	clk=0			
📫 image	curr_timestep[9:0]=064	<u>)+))+++++++++++++++++++++++++++++++++</u>		+()+)(+)+()()()()+))+()(64
	output_spike_count_0[7:0] =10	5 6	7	8 9 10
	output_spike_count_1[7:0] =3	1 2	3	
	output_spike_count_2[7:0] =1	1		
т spike_gen_rand	output_spike_count_3[7:0] =1	1		
🐜 weights	output_spike_count_4[7:0] =4	2 3		4
	output_spike_count_5[7:0] =0	0		
	output_spike_count_6[7:0] =3	1	2	3
	output_spike_count_7[7:0] =7	5 6		7
	output_spike_count_8[7:0] =4	1 2		3 4
	output_spike_count_9[7:0]=6	2 3 4 5		6
Type Signals	output_vmem_0[8:0] =33			33
wire output_spike_count_7[	7			
wire output_spike_count_8[	7			
wire output spike count 90	7			



integer i;

end end initial begin

initial begin
 clock = 0:

#5 clock = ~clock;

dut.image.mem[0] = 8'h00;

dut.image.mem[1] = 8'h00;

dut.image.mem[2] = 8'h00;

dut.image.mem[3] = 8'h00;

dut.image.mem[4] = 8'h01;

dut.image.mem[5] = 8'h00;

dut.image.mem[6] = 8'h00; dut.image.mem[7] = 8'h00;

dut.image.mem[8] = 8'h00;

\$dumpfile("snn\_simple\_integration\_test.vcd"); \$dumpvars(0, snn\_simple\_integration\_test\_tb);

https://pythoncourse.eu/machinelearning/training-andtesting-with-mnist.php

> dut.image.mem[9] = 8'h00; dut.image.mem[10] = 8'h00;dut.image.mem[11] = 8'h00;dut.image.mem[12] = 8'h08;dut.image.mem[13] = 8'h3e; dut.image.mem[14] = 8'h18; dut.image.mem[15] = 8'h00; dut.image.mem[16] = 8'h00; dut.image.mem[17] = 8'h00; dut.image.mem[18] = 8'h00; dut.image.mem[19] = 8'h00; dut.image.mem[20] = 8'h05; dut.image.mem[21] = 8'h44; dut.image.mem[22] = 8'h74; dut.image.mem[23] = 8'h4f; dut.image.mem[24] = 8'h12; 58485 5848 5848 58492 584956 584986 Reading 10 - 3

Waveform of Spikes Accumulating

-----

### Goal 2 - Bring Up Plan

#### **Client Request:**

Develop a bring up plan for future students to use to bring up our design on the Caravel Infrastructure.

### **Product to be Delivered:**

Bring up plan for our chip design

### **Contents of Product:**

- 1. Background setup and assembly
- 2. Hardware Test
- 3. Software Test
- 4. SNN Chip Test
- 5. The test and simulation code

### **Physical Description**

Caravel Nucleo Board - Provides Interface

Caravel Nucleo Hat - Mounting

Caravel Breakout - Holds Design



Efabless Caravel Board<sup>[6]</sup>



NUCLEO-F746ZG<sup>[5]</sup>

Caravel Nucleo Hat<sup>[6]</sup>



Caravel Breakout Boards with chip<sup>[6]</sup>

### **Goal 3 – Caravel Tutorial Document**

#### **Client Request:**

Create a tutorial document to streamline the process of understanding all Open-Source tools for the chip design process.

**Product to be Delivered:** Caravel Tutorial Document

#### **Contents of Product:**

- 1. 40 pages long.
- 2. Contains 8 sections of required information.
- 3. Contains 2 additional sections of supplementary information.

Tutorial Table of Con	itents ->
-----------------------	-----------

1. Overview of the Efabless Process Steps	4
2. Windows Subsystem for Linux Installation (WSL2) on Windows 11	6
2.1 Opening Power Shell	6
2.2 Installing WSL2	7
3. Setting Up Dependencies for the Caravel Repository	8
3.1 Updating Ubuntu to latest version	8
3.2 Installing Docker in Ubuntu	10
3.3 Troubleshooting	12
3.4 Installing Python 3 with PIP	13
3.5 Installing klayout (Optional)	13
4. Getting Started with the Caravel Repo	14
4.1 Cloning your Project Repository into Linux	14
4.2 Setting up Local Caravel Environment	14
4.3 Before hardening user project	15
4.4 Hardening user project to test open lane setup: Example - User adder	15
4.5 Before Hardening the user project wrapper	16
4.6 Creating user project wrapper	17
4.7 Running test bench simulations on your design	18
4.8 Example	18
4.9 Running final checks on your project before submission	19
5. Getting started with the Wishbone Bus	21
5.1 What is the wishbone bus	21
5.2 Using the wishbone bus from the Management SoC	21
5.3 Writing to the user area	21
5.4 Reading from the user area	22
6. Getting started with the Logic Analyzer	24
6.1 What is the logic analyzer	24
6.2 How to use the logic analyzer in the user area	24
6.3 How to use the logic analyzer from the management SoC	24
7. How to use interrupts	25
7.1 What is an interrupt	25
7.2 How to trigger an interrupt from the user area	25
7.3 How to receive an interrupt in the SoC	25
8. How to use klayout to view GDS files	27
9. How to submit your project design to Efabless	30
10. Caravel PCB daughter & carrier boards you receive from Efabless	33
10.1. Introduction	33
10.2.0. Overview of caravel_board	33
10.2.1. Power management section	33
10.2.2. FPGA section	33
10.2.3. GPIO section	33
10.2.4. JTAG section	33
10.3.0. Overview of caravel_Nucleo	34
10.3.1. ST-Link debugger	34
10.3.2. USB interface	34
10.3.3. Buttons and LEDs	34
10.4.0 How the parts on the board connect to each other?	34
10.4.1. FPGA connections	34
10.4.2. Power management connections	34
10.4.3. GPIO connections	35
10.4.4. JTAG connections	35
10.4.5. ST-Link connections	35
10.4.6. Button and LED connections	35
10.5. How the caravel_board connects to the user area and how to test it?	35
10.6. Conclusion	36
10.7. Full form of all abbreviations	37
11. References	37

# CONCLUSION

### **Challenges and Solutions**

#### **MPW Submission**

MPW9 not yet open for submission and future of funding unclear - Wait for next shuttle with project ready

### **Unclear / Outdated Information**

Lots of conflicting information

- Asking questions in Slack, creating a document with all knowledge

#### Efabless program changes

Only 2 of 8 shuttles were returned with changes to the bring-up each time

- Efabless provides a lot of test and simulation code
- Ask question to Slack

### **Future Work**

#### **Next Steps:**

Submission of our chip design to the next MPW Submission. Bring up of our SNN chip design in the Caravel Infrastructure.

#### Improvements to be Made:

More in-depth analysis and testing of the Caravel Tutorial Document.

a. Only internal testing of the document has been performed

#### Performance Improvements for the SNN

- a. Larger data-width for better precision and more accurate inferences
- b. Minimize sources of BW bottlenecking to improve inference speed
- c. Find ways to add more SRAM for higher resolution images

### Conclusion

- Our project our goals:
  - Silicon prove a neuron model.
  - Create a bring up plan for our chip design.
  - Create a tutorial document for the Open-Source software tools used.
- We have accomplished all our objectives to a reasonable degree.
- Still further steps to take as well as improvements to be made.

### Bibliography

[1] Yamazaki, K., Vo-Ho, V. K., Bulsara, D., & Le, N. (2022). Spiking Neural Networks and Their Applications: A Review. Brain sciences, 12(7), 863. <u>https://doi.org/10.3390/brainsci12070863</u>

[2] Erickson, B. J., Korfiatis, P., Akkus, Z., & Kline, T. L. (2017). Machine Learning for Medical Imaging. Radiographics : a review publication of the Radiological Society of North America, Inc, 37(2), 505–515. <u>https://doi.org/10.1148/rg.2017160130</u>

[3] Jason K. Eshraghian, Max Ward, Emre Neftci, Xinxin Wang, Gregor Lenz, Girish Dwivedi, Mohammed Bennamoun, Doo Seok Jeong, and Wei D. Lu "Training Spiking Neural Networks Using Lessons From Deep Learning". arXiv preprint arXiv:2109.12894, September 2021.

[4] Hashlake Blockchain,

https://www.hashlake.io/?msclkid=75a76a1a43a21dd6d9dc0e5a16821f6c&utm\_source=bing&utm\_medium=cpc&utm\_campaign=Web+search+2023&utm\_term=asic+miner+values&utm\_content=Asic+Miner

[5] STMicroelectronics.(2014). STM32 microcontrollers: UM1974 <u>https://www.st.com/resource/en/user\_manual/um1974-stm32-nucleo144-boards-mb1137-stmicroelectronics.pdf</u>

[6] Efabless. (n.d.). *Efabless/caravel\_board*. GitHub. Retrieved April 17, 2023, from <u>https://github.com/efabless/caravel\_board</u>

# SUPPLEMENTAL SLIDES

#### Wishbone Feed

- Pixel values <- MNIST</li>
- Weights <- pretrained snnTorch network
- Addresses <- CPU</li>

#### Randomizer

- Pseudo random no.
  generator
- Seed with image values

#### Spike Queue

• FIFO holder for Spikes

#### Neuron

LIF hardware model



### **Goal 1 - Challenges and Solutions**

#### **MPW Submission**

MPW9 not yet open for submission Unclear future of funding

- Wait for next shuttle with project ready

### **Unclear / Outdated Information**

Lots of conflicting information Information exists in random Slack threads and outdated forums - Asking questions in Slack, creating a document with all knowledge

### Space

Minimal space creates memory constraints - Reducing network size and data precision

### **Goal 2 - Challenges and Solutions**

#### Efabless program changes

- Only 2 of 9 shuttles returned with changes to bring-up each time
- Efabless provides a lot of test and simulation code, but some of them have issues need to fix
- o Ask question to Slack

#### Hard to try our program and debug the code

• Due to unavailable the physical board, it is hard for us to test our code and give expected output image.

#### **Testing setup is complicated**

- There are a lot of steps details, equipment and software tools we need to use. Easy to miss a step and need redo from the beginning.
- Make the bring up plan in more details.

### **Challenges and Solutions**

Limited documentation and conflicting/outdated information:

- Tested each documents information case by case.
- Communicated with Efabless personel through Slack
- Accumulated all the up-to-date info & links in the Caravel Tutorial Document.



### **Physical Configuration**

The components in our Efabless project:

- NUCLEO-F746ZG or NUCLEO-F413ZH
- Caravel Nucleo Hat
- Caravel breakout boards with a Caravel part installed
- One or more chips (The SNN Chip we design)
- Two jumpers for J8 & J9
- USB micro-B to USB-A cable





Caravel Breakout Boards with chip<sup>[6]</sup>



Caravel Nucleo Hat<sup>[6]</sup>

### **Katherine Gisi**

#### **Team Direction and Communication**

- Worked at organizing tasks, people, agendas, and times
- Kept tabs on various components

#### Diagram Drawing and System Organization

- Created system, block, process, and data level drawings/diagrams
- Was a sounding board for design

#### Research

- Kept up with Efabless updates on shuttle status and slack channel announcements
- Gained working understanding of SNNs, Digital Neurons, Test Benches, and ASIC flow

### Fulai Zhu

- PCB Research
  - provides an introduction and overview of the efabless caravel\_board and caravel\_Nucleo
  - Show all the components in the caravel\_Nucleo and how each board connect
- Bring up Test Plan
  - Write the document about how to test our Efabless project
  - Hardware and software test plan and setting
  - Simulation setting



https://github.com/efables s/caravel\_board/blob/mai n/firmware\_vex/nucleo/do cs/caravel+nucleo.jpg

### **Tyler Green**

- Tool Research and Documentation
  - Wishbone and Logic Analyzer
  - Interrupts
  - Simulation tools
- RTL implementation
  - SNN control logic
  - Spike queue
  - LFSR random number generator



https://github.com/gtkwa ve/gtkwave/blob/master/ gtkwave3/share/icons/gtk wave\_256x256x32.png



https://snntorch.readthedocs.io/en/latest/tutorials/tutorial\_1.html



### William Zogg

- SNN Architecture Design
  - SNN Control Logic
  - Inter-module connections
- RTL Implementation
  - SNN Control Logic
  - Inter-module connections
- RTL Verification
  - SNN Control datapath
  - Wishbone data



https://hackster.imgix.net/uploads/attachments/140 6996/image\_ra4sBaLmFG.png?auto=compress%2Cf ormat&w=740&h=555&fit=max



https://cdn-media-1.freecodecamp.org /images/0\*msRB3BVp VkGVgEOd.

### **Aaron Sledge**

- Creation of the Caravel Tutorial Document
  - Over thirty pages in total
  - Created a majority of the sections in the document
  - Details how to acquire and how to use essential Efabless tools
  - Details how to acquire and use supplemental, but not essential, open-source software tools

1.	Overview of the Efabless Process Steps
1.	Windows Subsystem for Linux Installation (WSL2) on Windows 11
	2.1 Opening Power Shell
	2.2 Installing WSL2
2.	Setting Up Dependencies for the Caravel Repository
	3.2 Undating Ubuntu to latest version
	3 3 Installing Docker in Libuntu
	3.4 Troubleshooting
	3.5 Installing Dython 3 with DID
	3.6 Installing klavout (Ontional)
4	Gatting Started with the Caraval Rapa
4.	4.1 Claning your Droject Paperitory into Linux
	4.1 Cloning your Project Repository into Linux
	4.2 Setting up Local Caraver Environment
	4.3 manuelling user project to test open rare setup. Example - Oser_auter
	4.4 Creating user project wrapper
	4.5 Kunning test bench simulations on your design
	4.0 Example
	4.7 Running That checks on your project before submission
5.	Getting started with the Wishbone Bus
	5.1 what is the wishbone bus
	5.2 Using the wishbone bus from the Management SoC
	5.3 Writing to the user area
	5.4 Reading from the user area
6.	Getting started with the Logic Analyzer
	6.1 What is the logic analyzer
	6.2 How to use the logic analyzer in the user area
	6.3 How to use the logic analyzer from the management SoC
7.	How to use interrupts
	7.1 What is an interrupt
	7.2 How to trigger an interrupt from the user area
	7.3 How to receive an interrupt in the SoC
8.	How to use klayout to view GDS files
9.	How to submit your project design to Efabless
10	D. Caravel PCB daughter & carrier boards you receive from Efabless Efabless
	10.1. Introduction
	10.2.0. Overview of caravel_board
	10.2.1. Power management section
	10.2.2. FPGA section
	10.2.3. GPIO section
	10.2.4. JTAG section
	10.3.0. Overview of caravel_Nucleo
	10.3.1. ST-Link debugger
	10.3.2. USB interface
	10.3.3. Buttons and LEDs
	10.4.0 How the parts on the board connect to each other?
	10.4.1. FPGA connections
	10.4.2. Power management connections
	10.4.3. GPIO connections
	10.4.4. JTAG connections
	10.4.5. ST-Link connections
	10.4.6. Button and LED connections
	10.5. How the caravel_board connects to the user area and how to test it?
	10.6. Conclusion
	10.7. Full form of all abbreviations
11	. References

### 1. William Zogg SNN Architecture Design, RTL Implementation

### 2. Tyler Green

Software Tool Research, Documentation, RTL Implementation

### 3. Katherine Gisi

Team Direction/Communication, Diagram Drawing & System Organization, Research

### 4. Aaron Sledge

Software Tool Research & Caravel Tutorial Document

### 5. Fulai Zhu

PCB Research & Bring up Test Plan

### IDEAS FOR INDUSTRY REVIEW PANEL PRESENTATION

- MAYBE INCLUDE A PICTURE OF OUR GIT REPOSITORY
- KLAYOUT VIEW OF OUR PROJECT DESIGN IN THE USER AREA OF THE CARAVEL HARNESS
- PICTURES OF DIAGRAM ALONG WITH THE TEST BENCH RESULTS TO DEMONSTRATE FUNCTIONALITY OF DESIGN
- PICTURES OF THE INTEGRATION OF MODULES ALONG WITH TEST BENCH RESULTS OF THE INTEGRATION TESTING OF THE MODULES
- PICTURES OF OVERALL SYSTEM ALONG WITH RESULTS OF SENDING AN IMAGE THROUGH IT
  - GIFF OF HOW THE NEURONS IN OUR DESIGN SHOULD ACT, SOMETHING SIMPLE AND CONSUMABLE THAT WILL BE EASILY UNDERSTOOD BY THOSE WHO DON'T QUITE UNDERSTAND CHIP DESIGN/NEURON MODELS
- PICTURE/QUICK VIDEO DEMONSTRATING THE TUTORIAL DOCUMENT
- PICTURES OF THE PCB OUR CHIP WOULD BE RETURNED ON FOR MPW9
- POSSIBLY INCLUDE A PICTURE OF OUR BRING UP PLAN IF WE GET AROUND TO IT IN TIME