Digital Chip Fabrication

Team: sdmay23-28

Client & Advisor: Dr. Henry Duwe

12-07-22

Our Team

Katherine Gisi

EE

Spokeswoman
SNN Design

Fulai Zhu

EE

Documentation

Tyler Green

CprE

Software Tools
Environment

William Zogg

CprE

SNN Design

Aaron Sledge

EE

Software Tools
Documentation

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 design.

Client's Goals

Dr. Duwe is looking to create a co-curricular at ISU for students interested in this area

Our Project

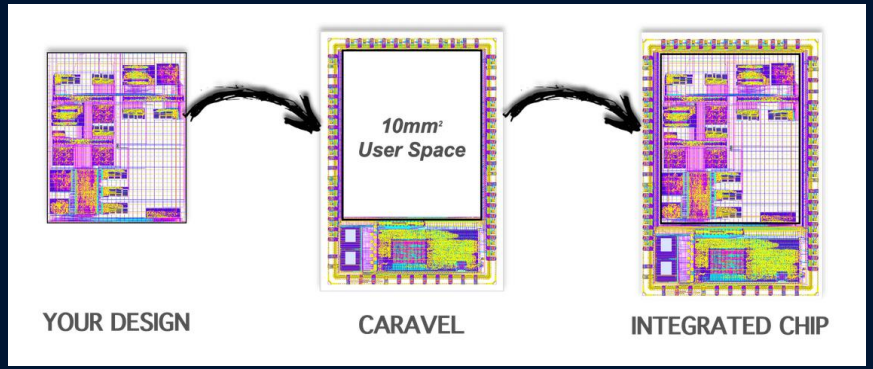
Silicon-prove a digital spiking neural network
Learn Efabless tools and flow
Develop documentation to help others learn the required software tools

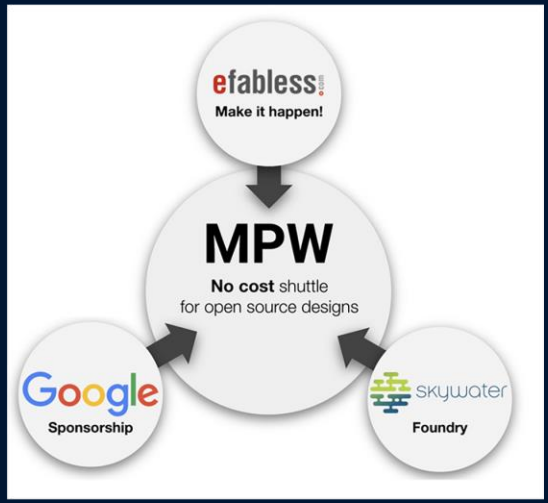
ASIC Fabrication

Who
Efabless + Google + Skywater = Open MPW Shuttles

What
ASIC Fabrication at no cost to developers

Why
Open Source ASICs





Introduction

Users
Requirements
Constraints
Standards

User Needs

1

Dr. Duwe

Silicon Proven Design
Software + Hardware
Documentation
Communication

2

Future student

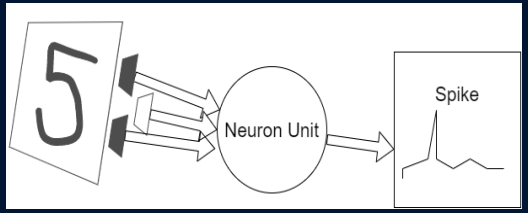
Examples
Resources
Handoff

3

Open Source community

Open Source IP
Documentation
Functionality

Project Description



Hardware Neuron to implement a spiking neural network (SNN) and surrounding infrastructure to identify digits from the MNIST data set.

Reason for project development
So that it might be used as an example project for a future chip design co-curricular at ISU.

Tasks from Client
Functional digital chip design
Bring-up plan for our chip
Documentation of software tools used

Project Requirements

1. Design & Functionality Requirements

- Our chip design must fit within a 10 square millimeter area
- Logic represents at least one neuron in a Leaky Integrate and Fire Spiking Neural Network
- Design uses a digital configuration
- Design must successfully pass the Open MPW precheck tool

2. Documentation Requirements

- Repository should have a README file
- Detailed bring up plan for our chip design
- Write up a tutorial document to explain how to use all of the necessary Efabless software tools

Project Requirements (continued)

3. Environment Requirements

- The project must be posted on a git-compatible repo and be publicly accessible and contain the following items:
  - A makefile with targets to compress/uncompress, and clean the project
  - LICENSE files for the top-level project as well as each of the third-party components used
- Project is designed utilizing open-source tools (constraint)
- Project should pass checks provided in caravel harness GitHub repository before submitting to Efabless (constraint)

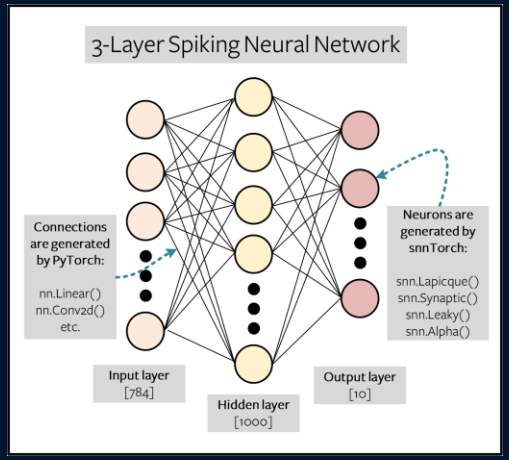
Design

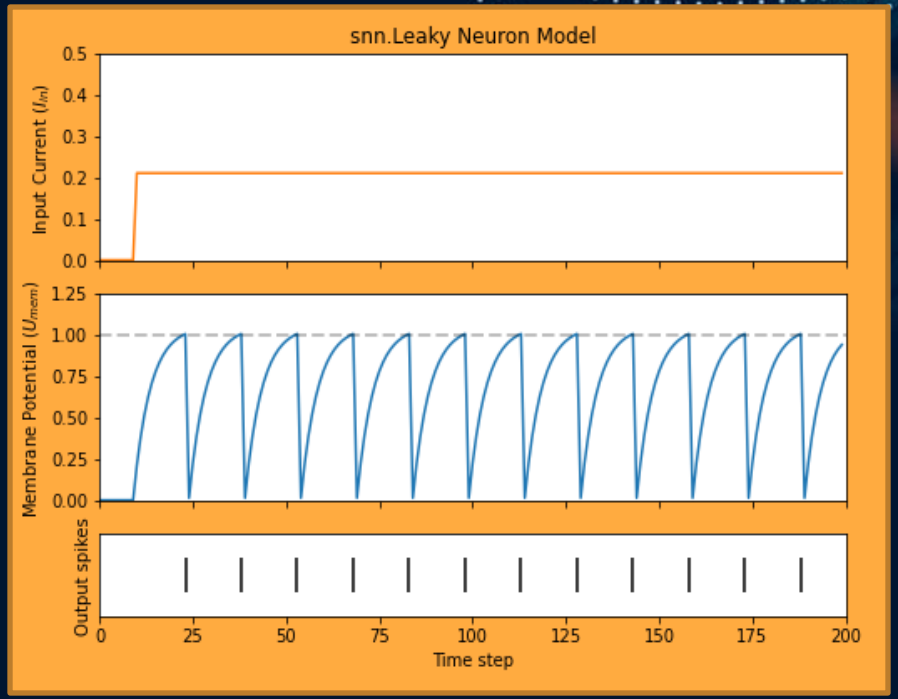
Context
Exploration
Proposal
Considerations

SNN Model

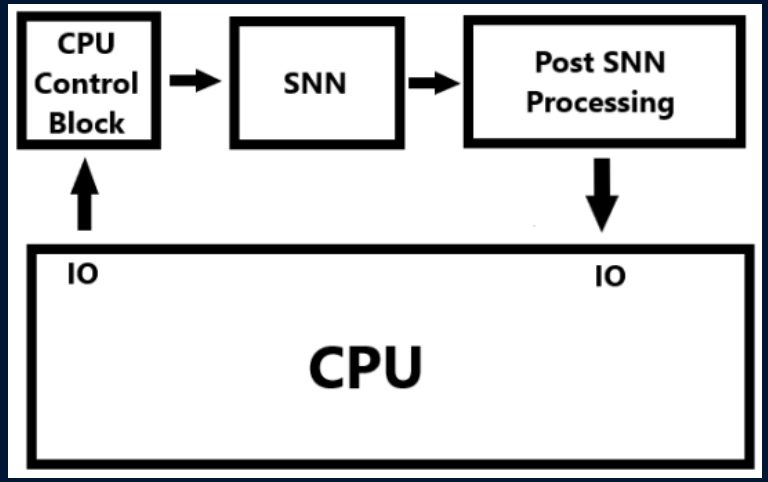
Neuron Model
Leaky Integrate and Fire

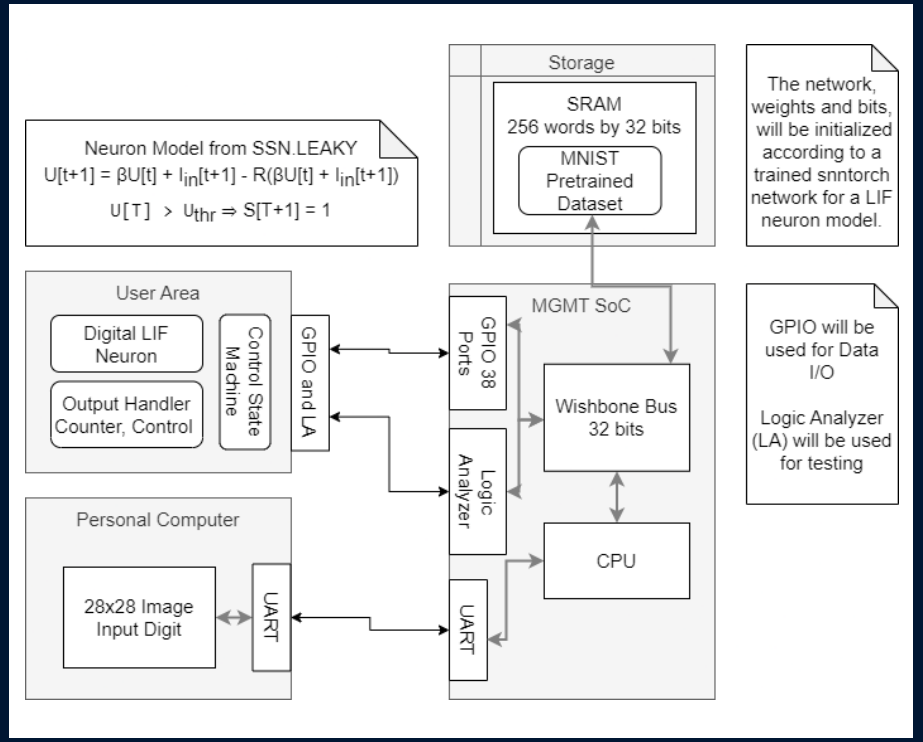
Network Model
snnTorch
Provides a pre-trained network





High Level Sketch



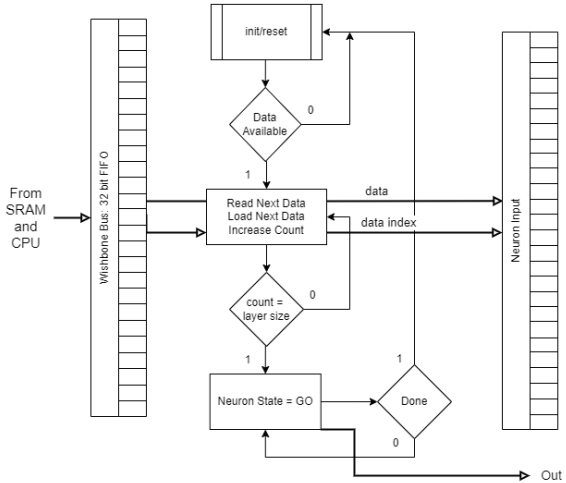


Block Processes

CPU Control

Inputs: Data Available, Data, Done

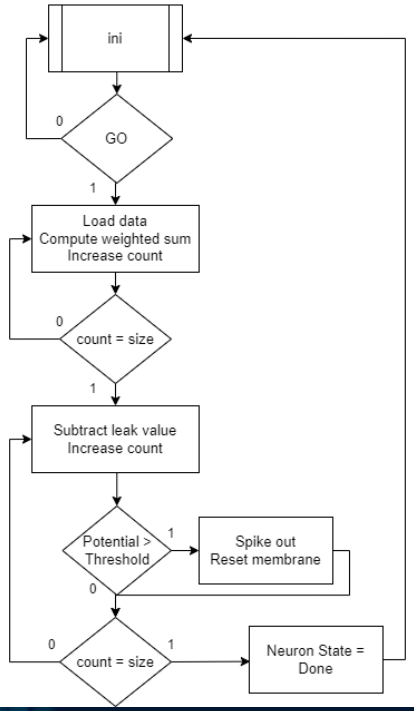
Outputs: Data Index, Data, Neuron State



SNN Unit

Inputs: Neuron State, Data (weights, potentials), size

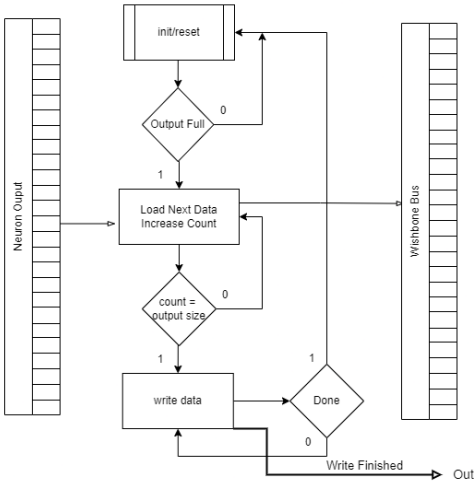
Outputs: Neuron Data (spikes, potentials), Output size, Neuron State



Post Process Control

Inputs: Neuron State, Neuron Output Data, Output Size

Outputs: Write Data, Write Done



Testing

Types and Plan

Testing Methods and Tools

Unit – Individual components (integrator)
- Testbenches and waveforms

Integration – Grouped components (control unit, neuron)
- Testbenches and waveforms

System – Full user and management area
- Testbenches and waveforms
- Prechecks
- Logic analyzer

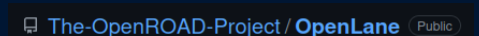
GTKWave
Waveform
analysis

Testing Methods and Tools (cont.)

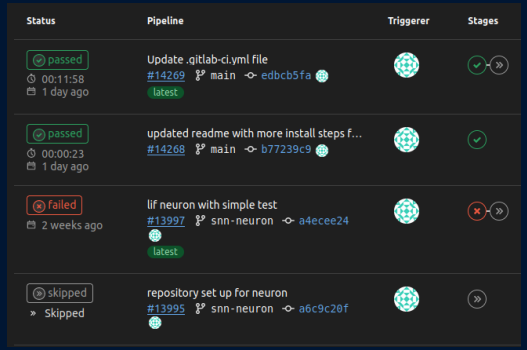
Regression
- Version control (GitLab)
- Continuous Integration

Acceptance
- Prechecks
- Open MPW approval

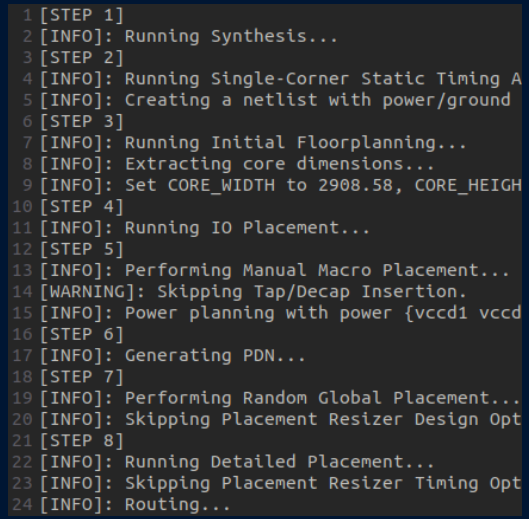
Additional Testing
- Feedback on documentation



OpenLane
RTL to GDSII



GitLab
Source Control
Continuous Integration



```
[STEP 1]
[INFO]: Running Synthesis...
[STEP 2]
[INFO]: Running Single-Corner Static Timing A
[INFO]: Creating a netlist with power/ground
[STEP 3]
[INFO]: Running Initial Floorplanning...
[INFO]: Extracting core dimensions...
[INFO]: Set CORE_WIDTH to 2908.58, CORE_HEIGH
[STEP 4]
[INFO]: Running IO Placement...
[STEP 5]
[INFO]: Performing Manual Macro Placement...
[WARNING]: Skipping Tap/Decap Insertion.
[INFO]: Power planning with power {vccd1 vccd
[STEP 6]
[INFO]: Generating PDN...
[STEP 7]
[INFO]: Performing Random Global Placement...
[INFO]: Skipping Placement Resizer Design Opt
[STEP 8]
[INFO]: Running Detailed Placement...
[INFO]: Skipping Placement Resizer Timing Opt
[INFO]: Routing...
```

MPW Precheck
Synthesis
Placement
Layout violation

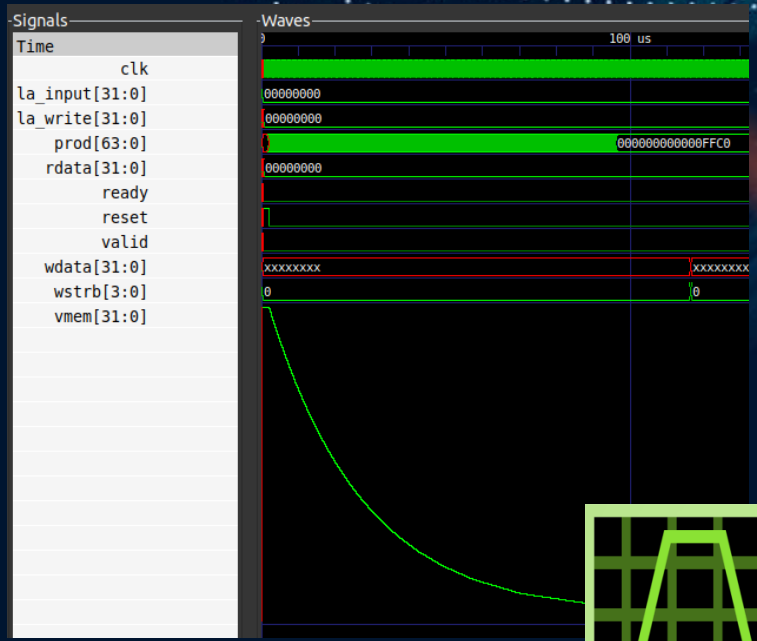
Experience with Testing and Tools

What we did:

- Experimented with caravel example project
- Hardened adder from sddec22-17
- Started new simple project for XOR
- Developed simple but robust testbenches
- Hardened our own XOR project
- Analyzed testbench results (GTKWave)
- Entire process again for neuron

What we learned:

- Caravel harness
- Testbenches
- Waveform analysis (GTKWave)
- Hardening (MPW precheck, OpenLane)



Implementation

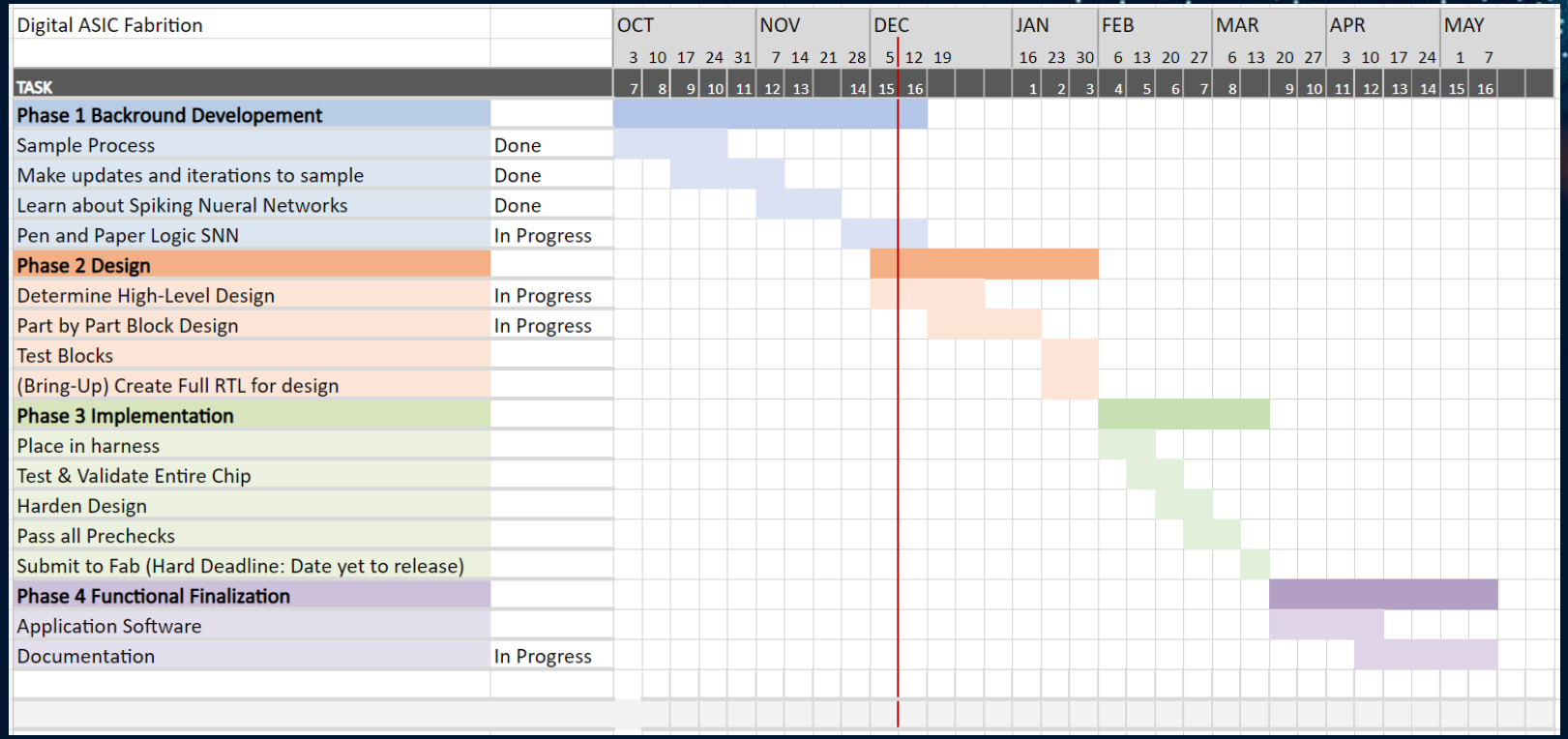
Spring Semester 2023

Project Plan

- Broken Into Phases
  - Background
  - Design
  - Implementation
  - Documentation
- Adjustments as we proceed (AGILE)
  - Well-scoped design allows flexibility
  - Number of neuron modules

Task Decomposition	
Task	Goal Date
Phase 1 Background Developement	**11/7/2022**
Sample Process	10/24/2022
Completer User_Adder Exmple	
Every Member Have Design Environment Set-Up	
Make updates and iterations to sample	10/31/2022
Do something outside of the sample with the adder	
Possibly do layout visualization	
Learn about Spiking Nueral Networks	10/24/2022
Deterime what our design will do	
Know theorehtical Input and Output	
Pen and Paper Logic SNN	11/7/2022
Create Initial Design Sketch	
Create Design Flowchart	
Phase 2 Design	**12/12/2022**
Determine High-Level Design	11/14/2022
Divide into Memory/Logic Parts	
Deterime ideas for how to test each piece	
Part by Part Block Design	11/28/2022
Create RTL for wishbone bus	
Create Verilog (x number of pieces)	
Test Blocks	
Finalize Tests	12/5/2022
Test and Reiterate	
(Bring-Up) Create Full RTL for design	12/12/2022
Generate Netlist	
Phase 3 Implementation	**2/20/2023**
Place in harness	1/23/2023
Test & Validate Entire Chip	1/30/2023
Harden Design	2/6/2023
Pass all Prechecks	2/13/2023
Submit to Fab (Hard Deadline: Date yet to release)	2/17/2023
Phase 4 Functional Finalization	**5/1/2023**
Application Software	3/6/2023
Documentation	5/1/2023

Project Schedule



Project Status

Summary:
Environment Usage, Tool Learning, Simple Designs and Iterations, Defined design idea and components (SNN, snntorch, LIF, MNIST, etc...)

Ongoing SRAM/Storage solution investigation
Our Plan:
- Use slack community once we have specs
- 2KB macro available

Neuron size determining area investigation
Our Plan:
- Start with one Neuron
  - Use tools to determine area used
  - See how many to fit
- Other ways use area with data storage

Next Steps

1. Continue implementation of design/tests in RTL
2. Continue documentation for chip design curriculum and caravel
3. Our goal shuttle, MPW-9, will be ~3 months following the MPW-8 submission date of Dec 31st. Exact date not announced.
    - Bring-up process hand off
4. Prove design (Hardware and Software interaction) in simulation


Thank You
Questions

Supplementary Slides

Definitions

ASIC

Application Specific Integrated Circuit

MPW

Multi Project Wafer

Shuttle Run

Round of Efabless wafer fabrication

Harness

Carvel provided infrastructure

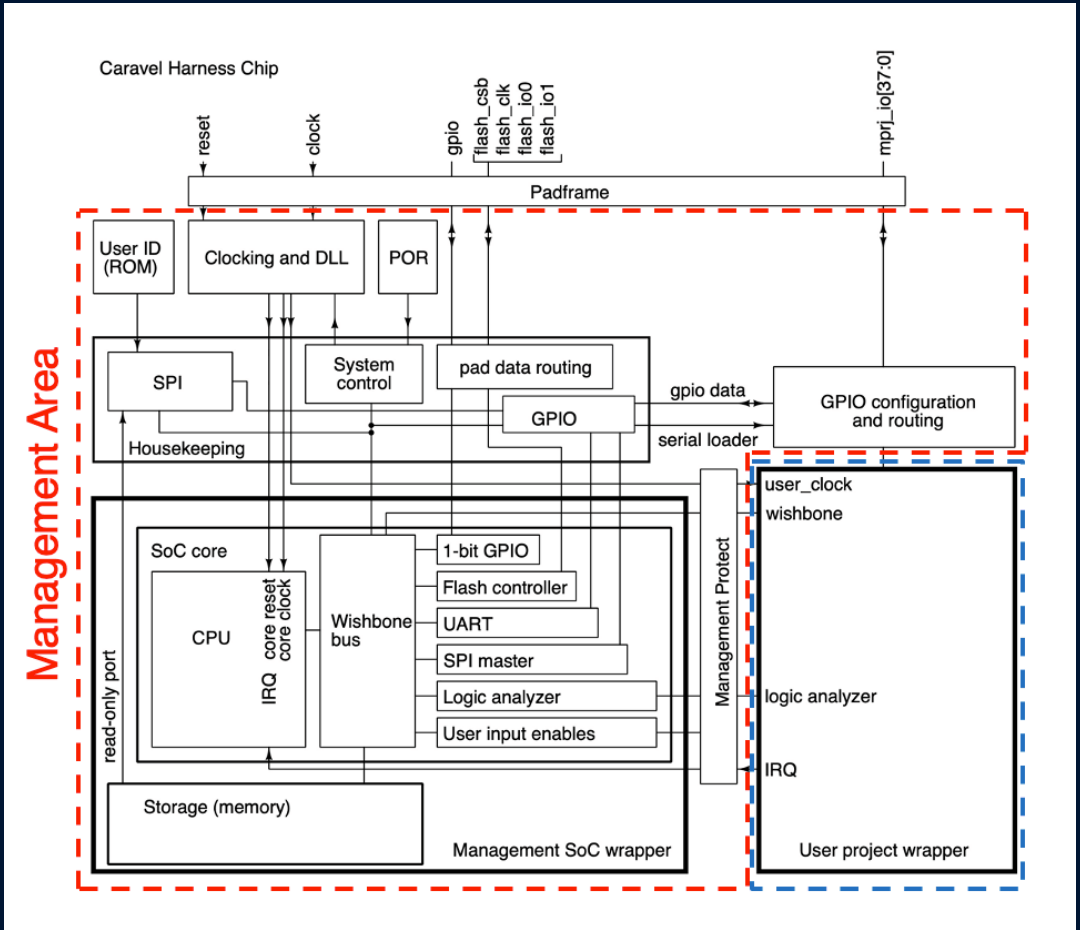
MNIST Database

Database of handwritten digits, used for image processing

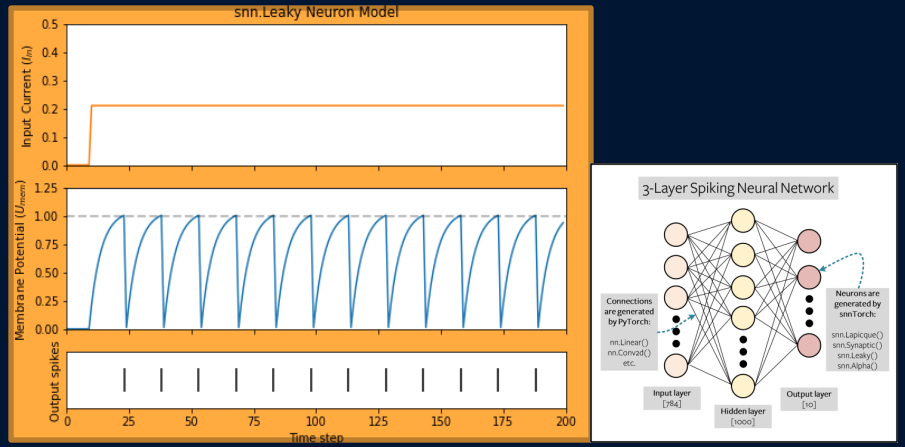
ASIC Flow

Process steps from design specification to design tape-out

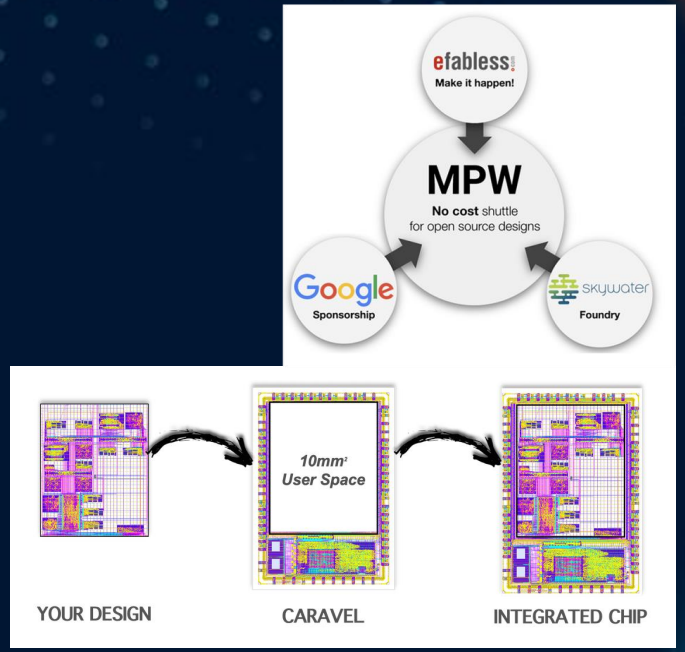
Harness Diagram



References



https://snntorch.readthedocs.io/en/latest/tutorials/tutorial_3.html



https://efabless.com/open_shuttle_program

References

The efabless Caravel project---Chip design for the software-oriented | COSCUP x RubyConf TW 2021 - YouTube

https://github.com/gtkwave/gtkwave/blob/master/gtkwave3/share/icons/gtkwave_256x256x32.png