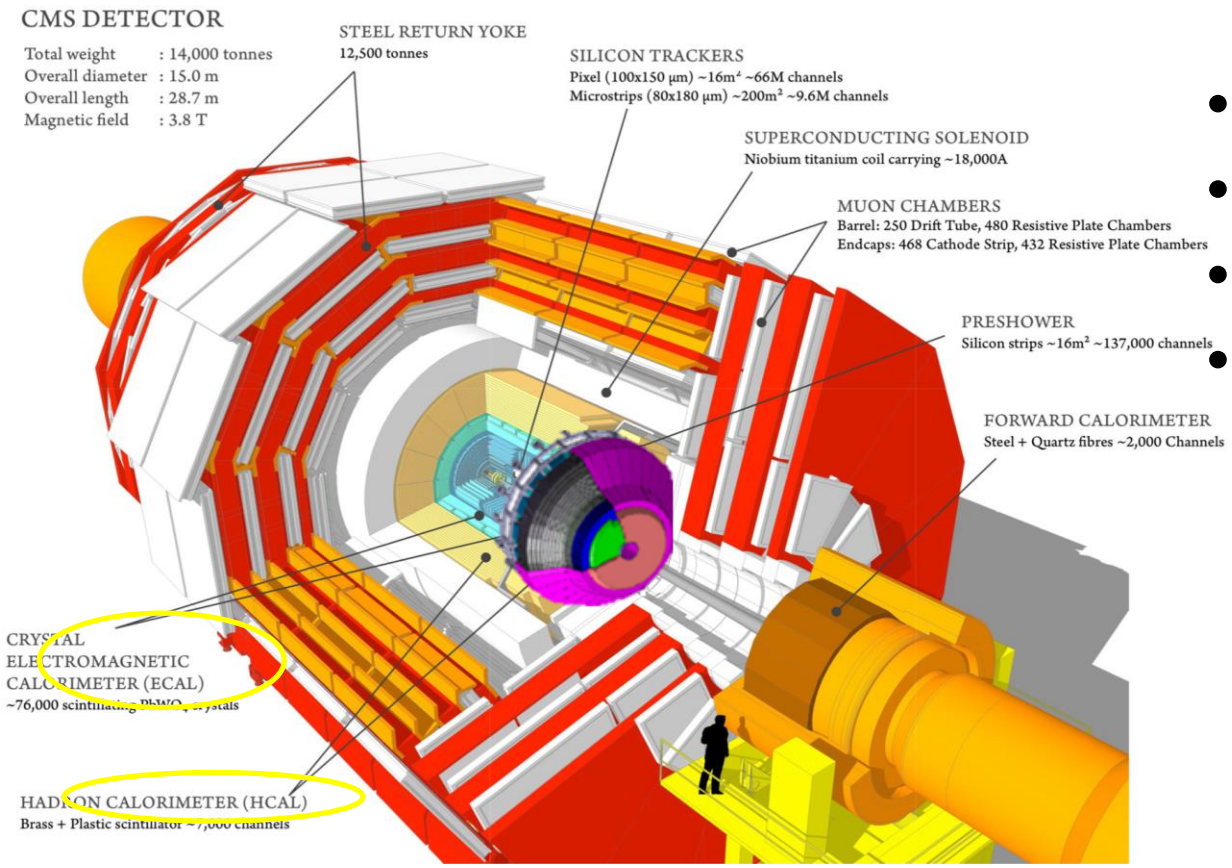
# HGCAL Project: ECONT and ECOND Testing

Mentee: Casandra Saxon (Southeastern Louisiana Univ.)
Mentor: Jim Hirschauer (Fermilab)
Team: Danny Noonan, Grace Cummings, Alex Campbell, Yulun Miao, Erdem Ertorer, and Ramneet Kaur

# Overview

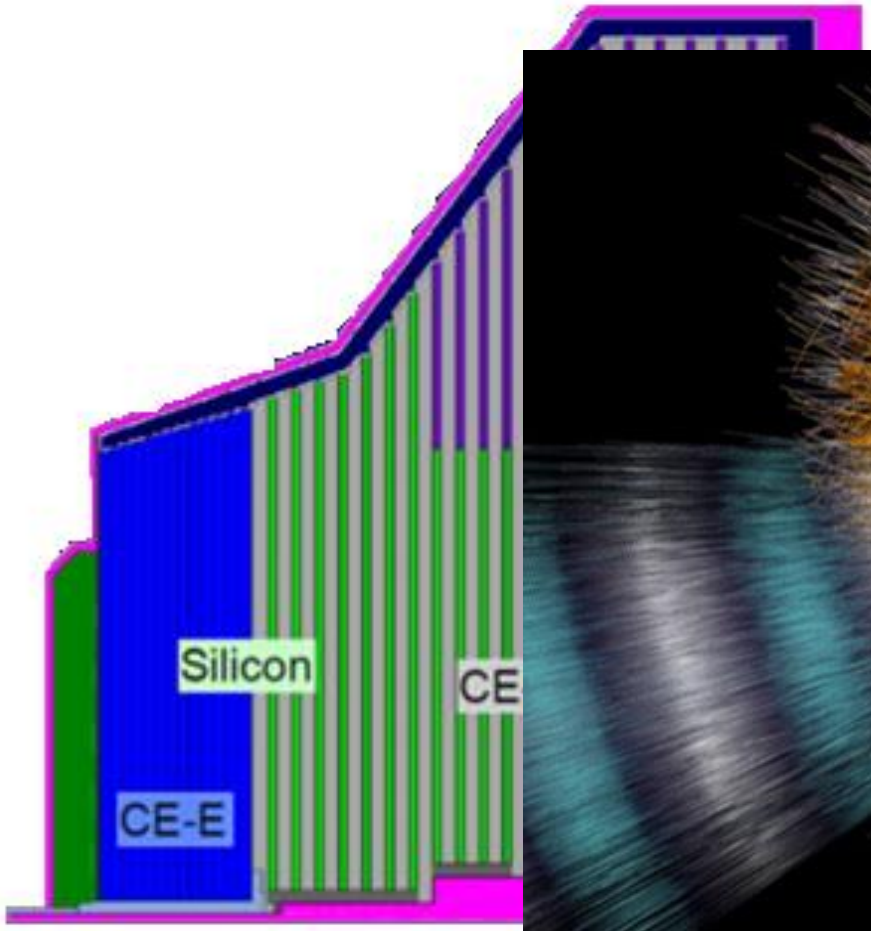
1. Introduction:
    - HGCAL Project
    - ECON Project
2. Test System of ECONs
3. Familiarizions
4. Different Tests
5. Conclusion
6. Awknowledgement

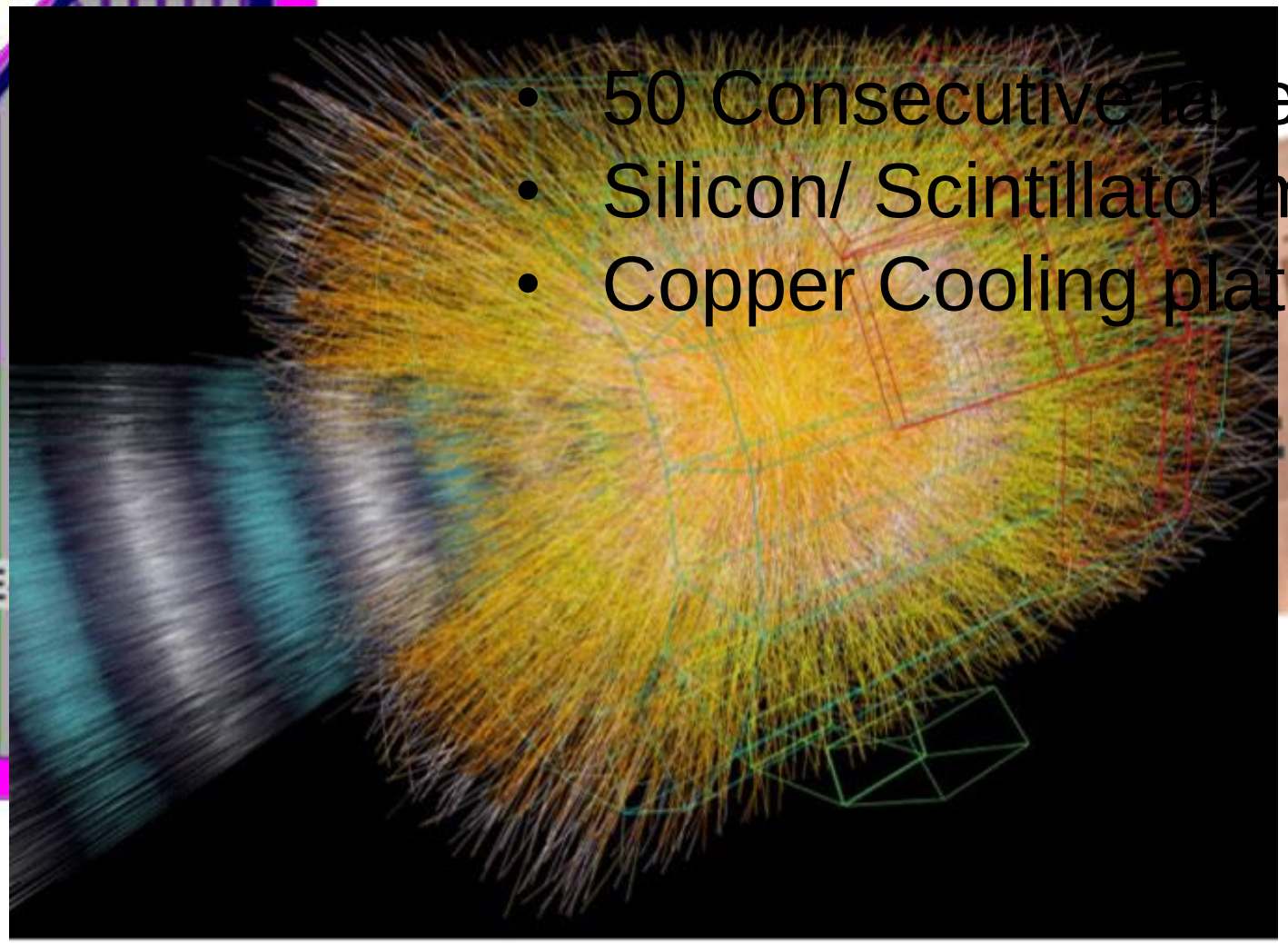
# Introduction: HGCAL Project




- Update goal: High Luminosity
- - more data/ radiation
- HGCAL: High Granularity Calorimeter
- End Cap replacement:
  ↑ radiation ↓ light production
  - cannot handle large data amounts
  - finer sensors, smaller detectors

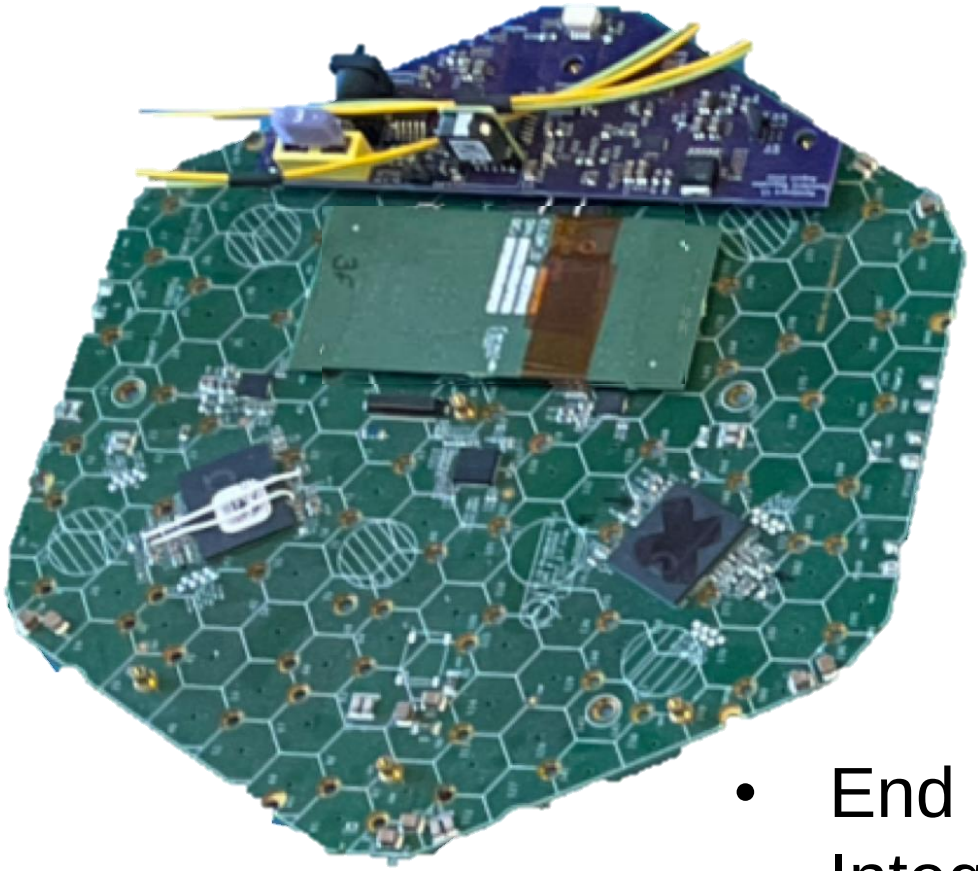
# Introduction: HGCAL Project

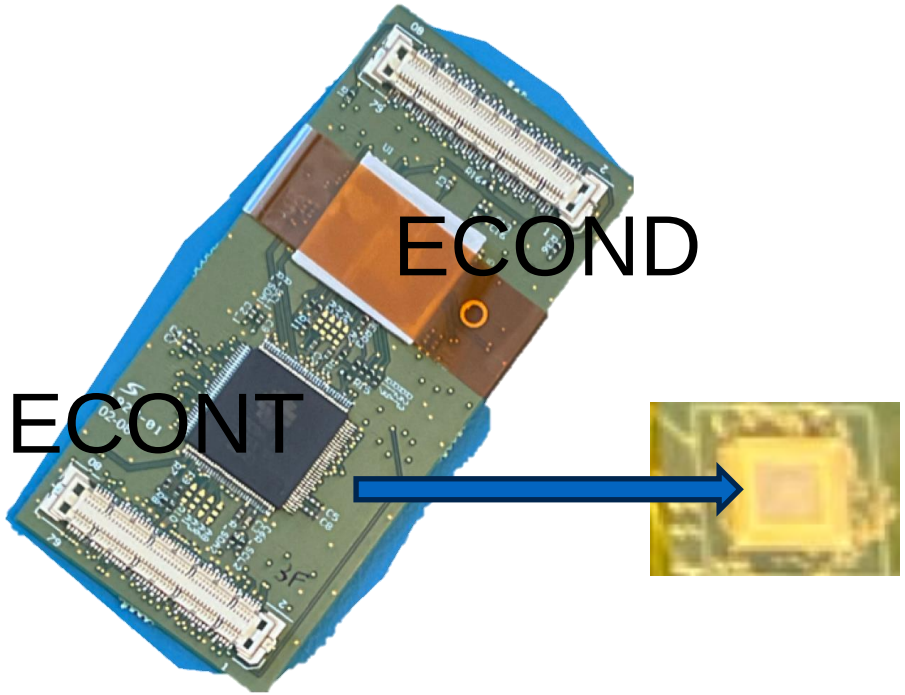


- 50 Consecutive layers
- Silicon/ Scintillator modules
- Copper Cooling plate

# Introduction: ECON Project

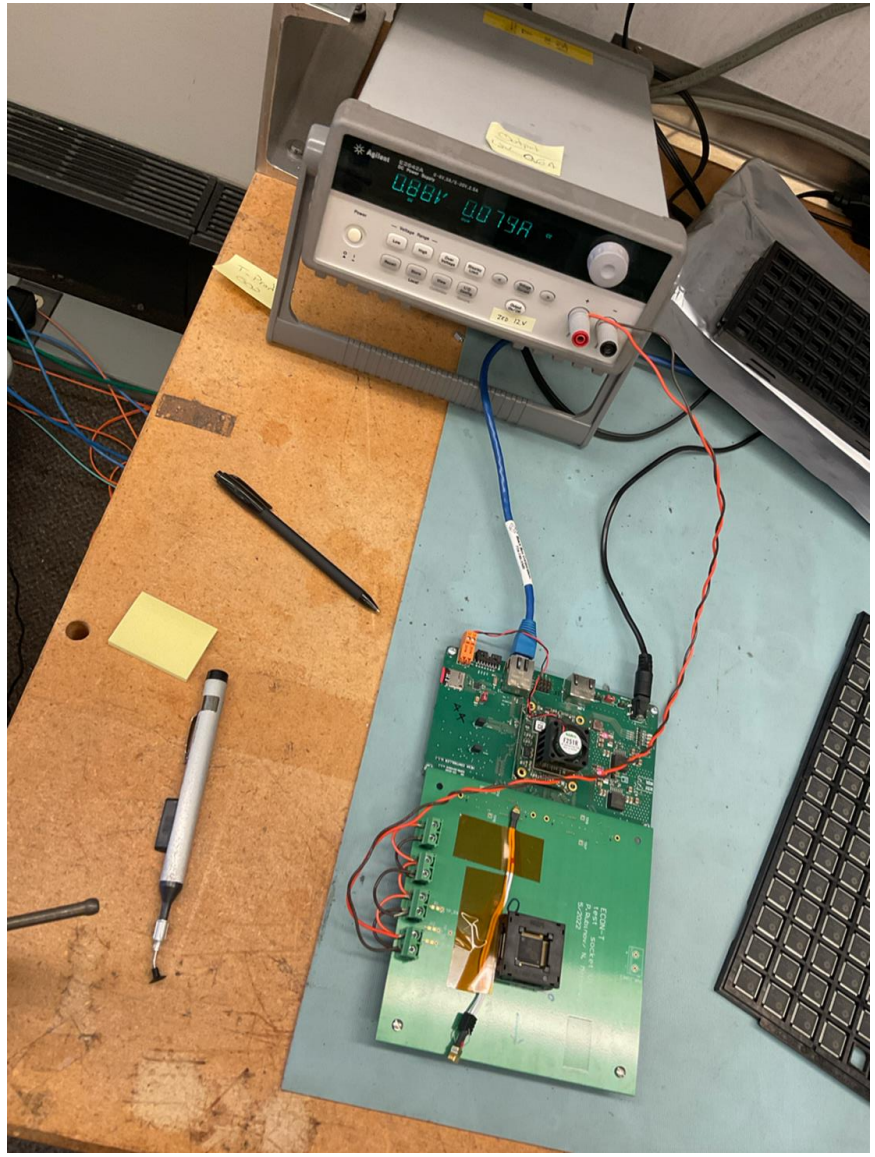





- End Cap Concentrators (Application Specific Integrated Circuit)
- ECONT: Trigger System
- ECOND: Data Acquisition System

# Test System of ECONs

- ~70,000 ECON ASICs
- In June we received the 1st 2000 chips
- The goal is to learn the basics of how the chips should operate to finalize the test scripts
- Test Firmware loaded on Field Programmable Gate Array for each chip type
- FPGA sends data to chip and confirms the expected chip behavior
- ~100 tests targeting different functionalities



# Familiarization: Test Data Format

Select pursue@L-SWL-132151:~/CasandraSaxon/ECON/EconDChip/July/COB_July_10th2-5


```
{
  "created": 1720628560.457216,
  "duration": 127.61141681671143,
  "exitcode": 1,
  "root": "/home/HGCAL_dev/CSaxon/econd-sw/test_bench",
  "environment": {
    "Python": "3.6.8",
    "Platform": "Linux-4.19.0-xilinx-v2019.2-aarch64-with-centos-7.9.2009-AltArch",
    "Packages": {
      "pytest": "7.0.1",
      "py": "1.11.0",
      "pluggy": "1.0.0"
    },
    "Plugins": {
      "order": "1.1.0",
      "json-report": "1.5.0",
      "anyio": "3.3.1",
      "ordering": "0.6",
      "metadata": "1.11.0",
      "repeat": "0.9.1"
    }
  },
  "summary": {
    "passed": 68,
    "failed": 5,
    "skipped": 32,
    "total": 105,
    "collected": 105
  },
  "collectors": [
    {
      "nodeid": "",
      "outcome": "passed",
      "result": [
        {
          "nodeid": "test_algorithm.py",
          "type": "Module"
        },
        {
          "nodeid": "test_bist_threshold.py",
```

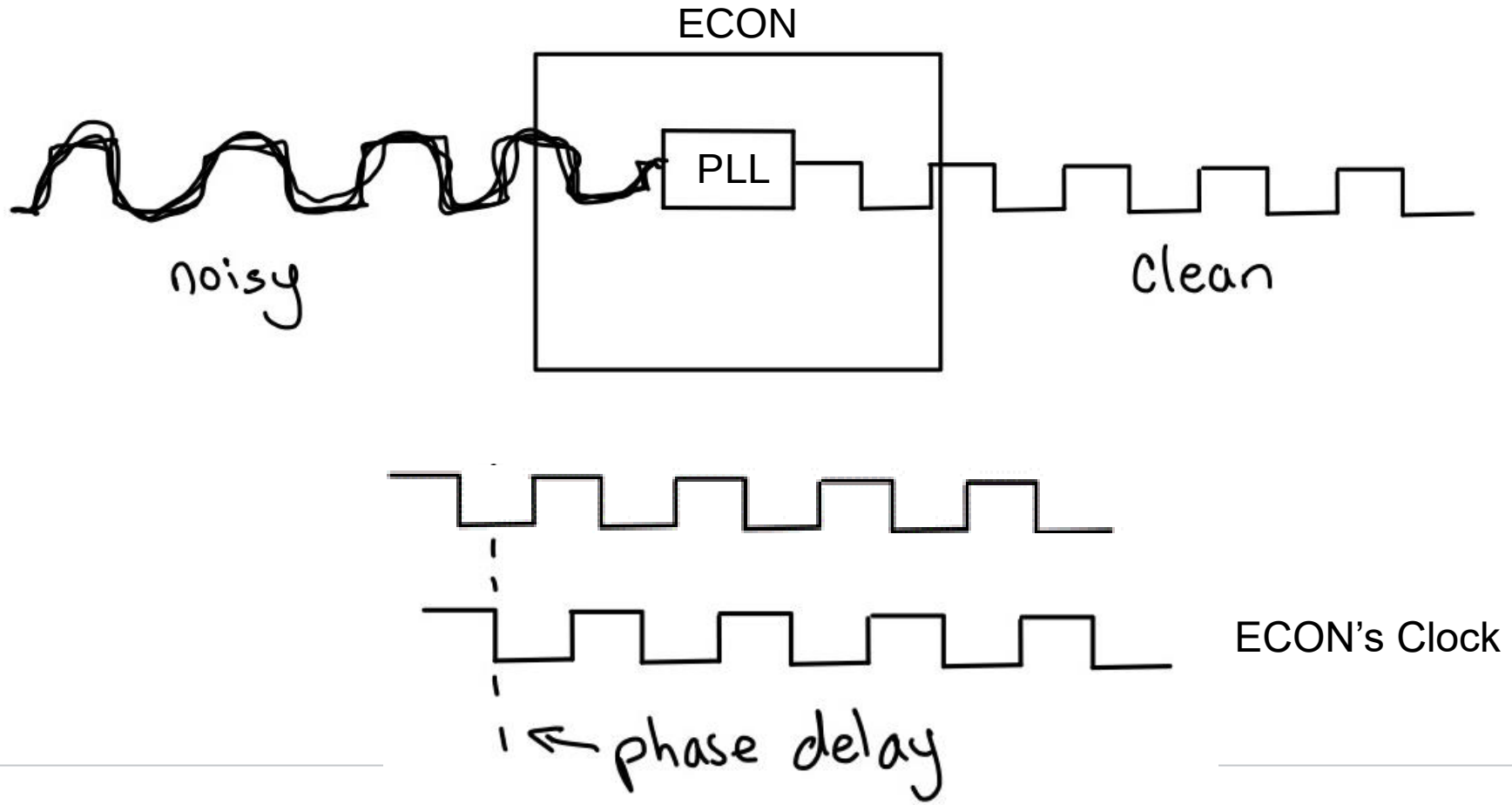

Learn to open and navigate through hundreds of json files through the terminal:

```
def create_json_dict(folder_path): #function for creating json_dict
files on a certain path
    json_dict = {}

    if not folder_path.endswith('/'):
        folder_path += '/'

    files = os.listdir(folder_path)

    for file_name in files:
        if file_name.endswith('.json'):
            file_path = os.path.join(folder_path, file_name)
            chip_name = file_path.split("_")[9:12] #change to match
naming
            chip_name = '_'.join(chip_name)
            json_dict[chip_name] = file_name
            json_dict = dict(sorted(json_dict.items()))

    return json_dict
```

```
folder_path =
#specific folder path of /json file destination
json_dict = create_json_dict(folder_path)
                    for i, test in enumerate(data['tests']):
                        if          in test['nodeid']:
                            line = test['nodeid']
                            line_number = i
                            break

                freq = data['tests'][line_number]['metadata']['frequencies_used']
                pusm = data['tests'][line_number]['metadata']['pusm_states']
```

# PLL's Functionality

The HGCAL project has thousands of ECON chips on it to filter the data. For all the chips to synchronize a Reference Clock (320 MHz) is sent to the ECON chip and the Phase Locked Loop inside the ECON chip modifies the Reference Clock to synchronize to the ECON's clock inside the chip. By adjusting the voltage and capacitance, the PLL can lock in the frequency range of 285 – 355 MHz.



# PUSM “Booting Up”

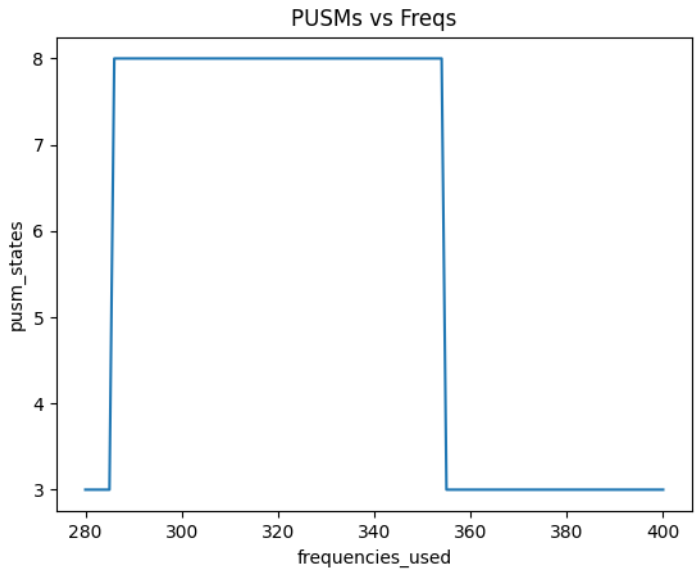
Once the clock is synchronized the ECON is “booted up” and ready to start filtering data. The Power Up State Machine is a sequence of numbers that tells us when the ECON is “booted up

- 3: Not Ready for data
- 8: Ready for data
- Supposed to stay at 8, but once the frequency range is outside 355 MHz, it reverts back to 3

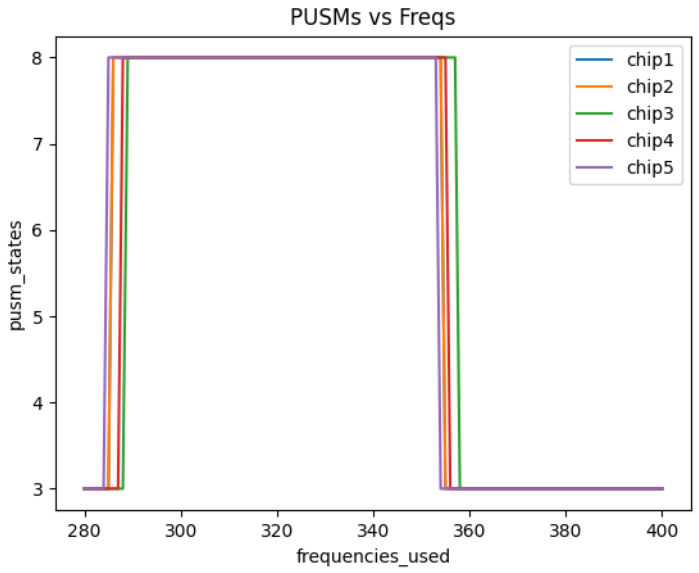
The PLL Lock pytest script collects the PUSM States and the Frequencies to see if the ECON is “booted up.”

# Familiarization: Plotting Raw Data

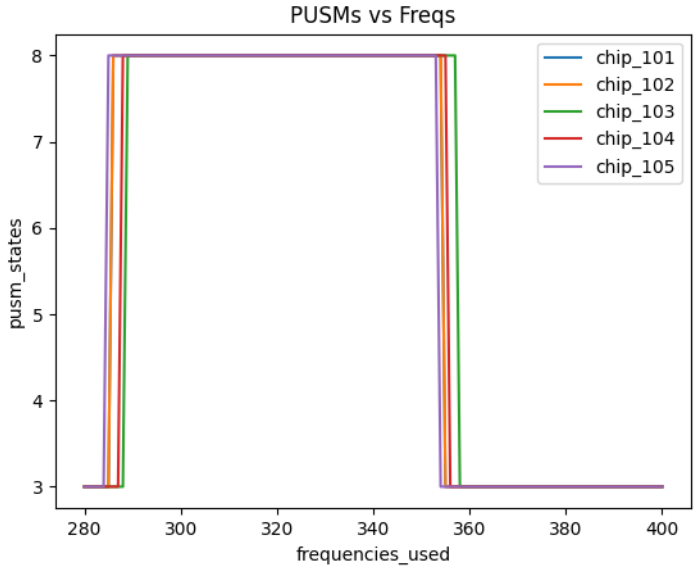
Plot from a given list of information:




Plot from one to multiple json files:



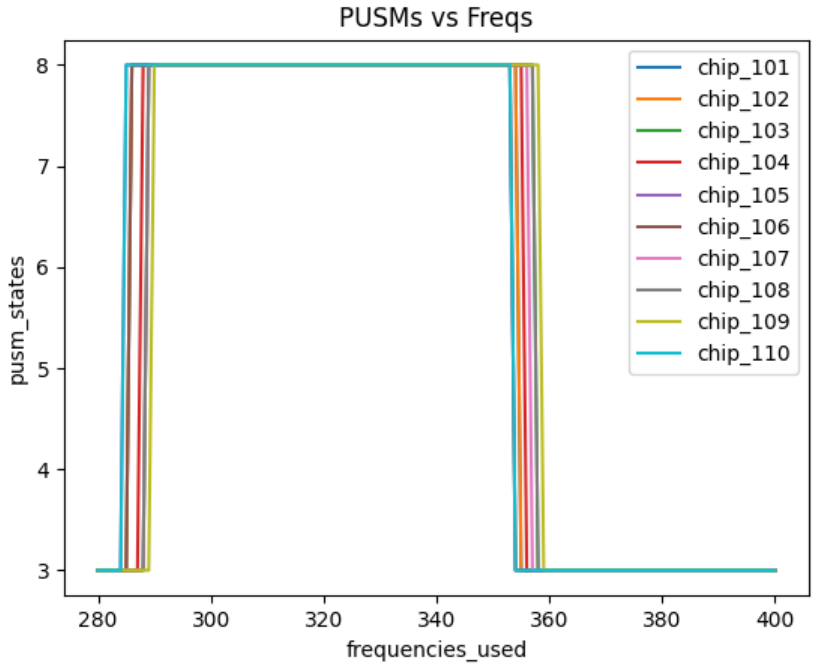

Plot from a directory:

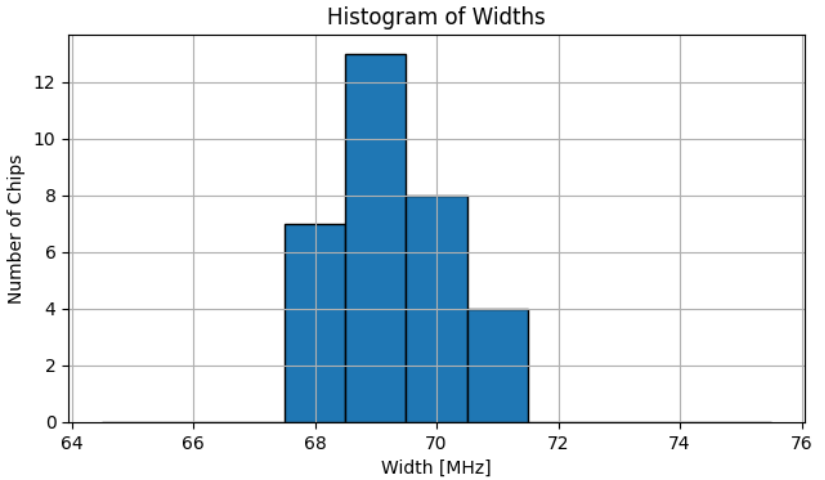


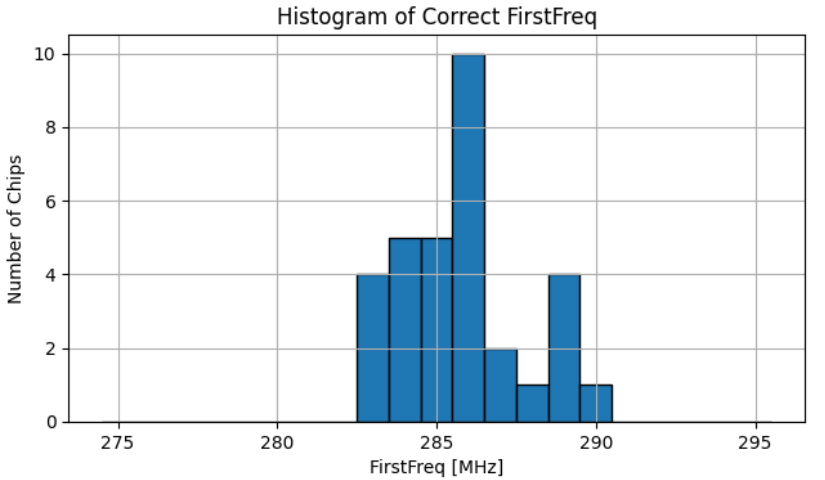

ECONT June (65 chips)

# Familiarization: Analyzing with Histograms

Plotting the Frequency Widths vs Chip number:




ECONT June (65 chips)

# Complete PLL Lock Graphing Script

Personal Coding assignment :

- Searchable Error Analysis Document
- Graphs
  - Overall graph for PUSM vs Freq
  - Individual Failed graphs
  - Sets of 5 Passed Graphs
- Summary printout
- Multiple Voltages 1.08, 1.2, 1.32

Histogram # of chips vs widths:

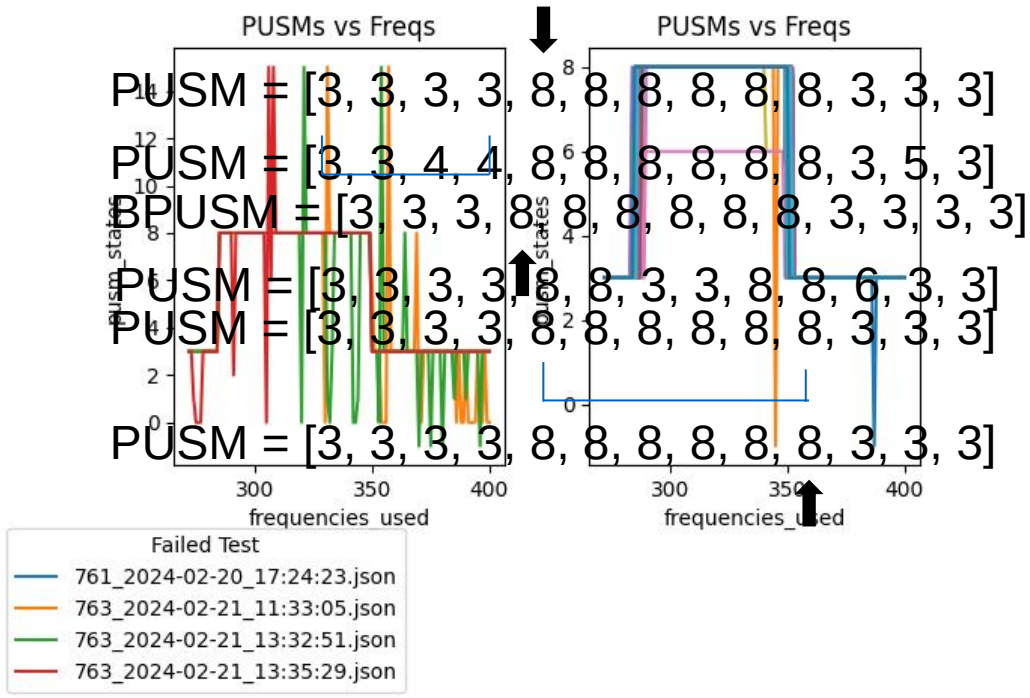
Histogram # of chips vs first Freq:

2D Histogram first Freq. vs widths:

- Weird vs Correct PUSM
- PUSM Check Document

Graphs completed for each sub-assignment:

- June ECONT chips (65)
- February ECOND chips (411)
- July ECONT chips (90)




PUSM = [3, 3, 3, 3, 8, 8, 8, 8, 8, 8, 3, 3, 3]

PUSM = [3, 3, 4, 4, 8, 8, 8, 8, 8, 8, 3, 5, 3]

BPUSM = [3, 3, 3, 8, 8, 8, 8, 8, 8, 3, 3, 3, 3]

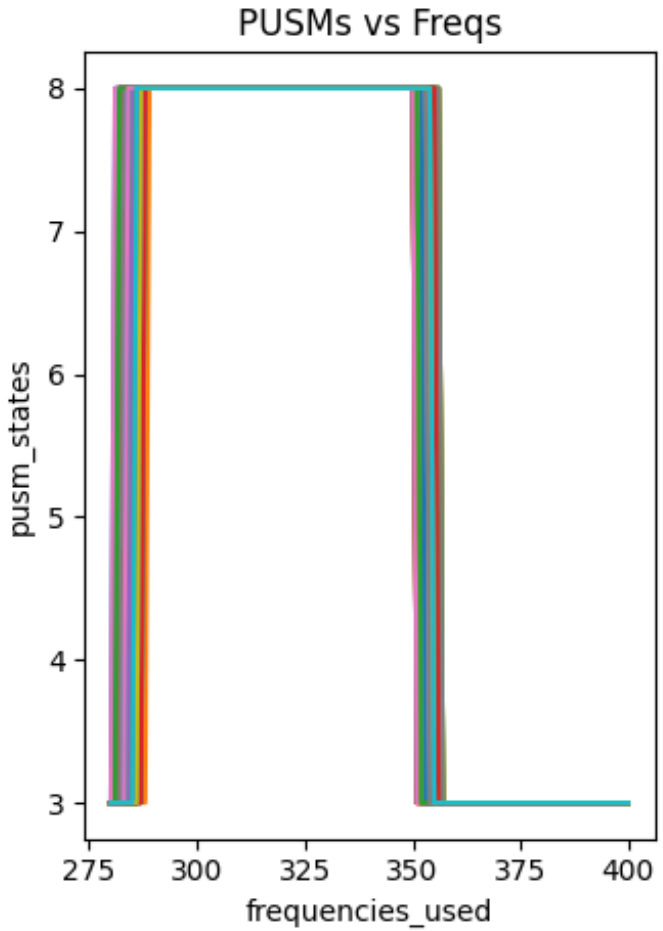
PUSM = [3, 3, 3, 3, 8, 8, 3, 3, 8, 8, 6, 3, 3]

PUSM = [3, 3, 3, 3, 8, 8, 8, 8, 8, 8, 3, 3, 3]

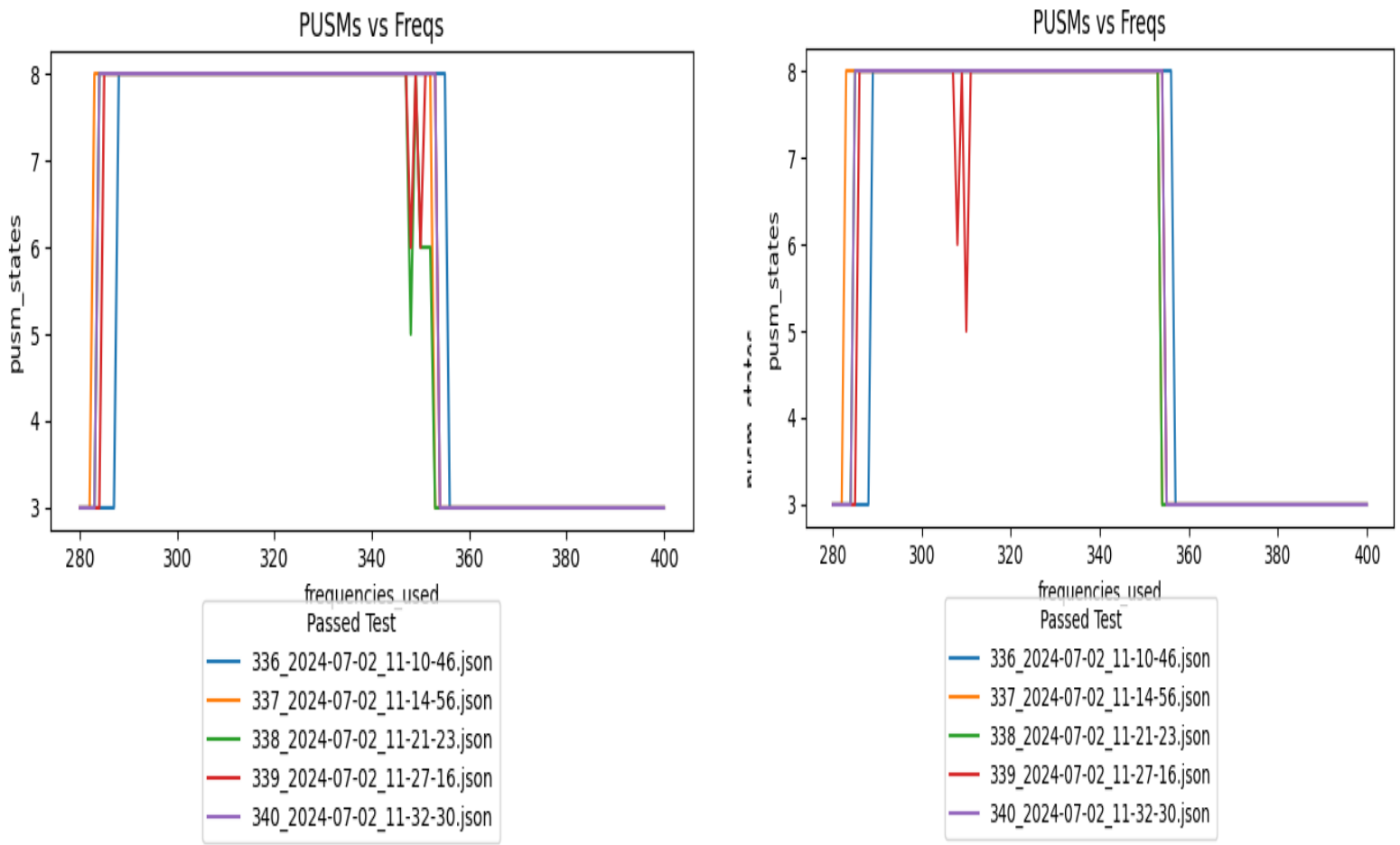
PUSM = [3, 3, 3, 3, 8, 8, 8, 8, 8, 8, 3, 3, 3]

# Complete PLL Lock Graphing Script

Overall Graph: ECONT July (90 chips)




Autolock 1.32



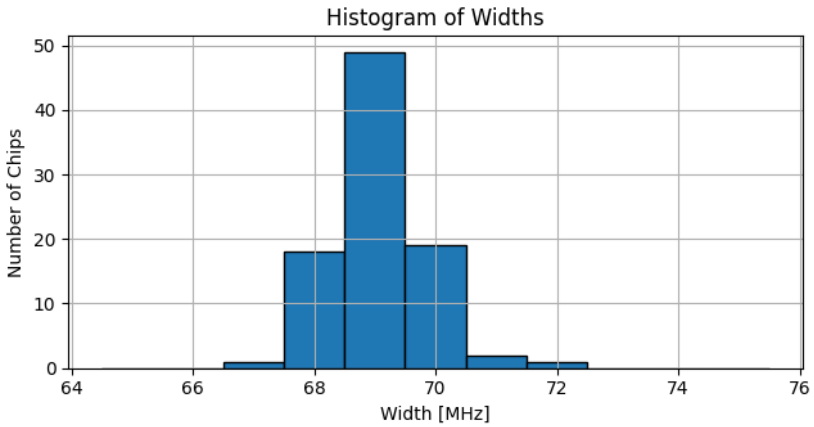

Autolock 1.2

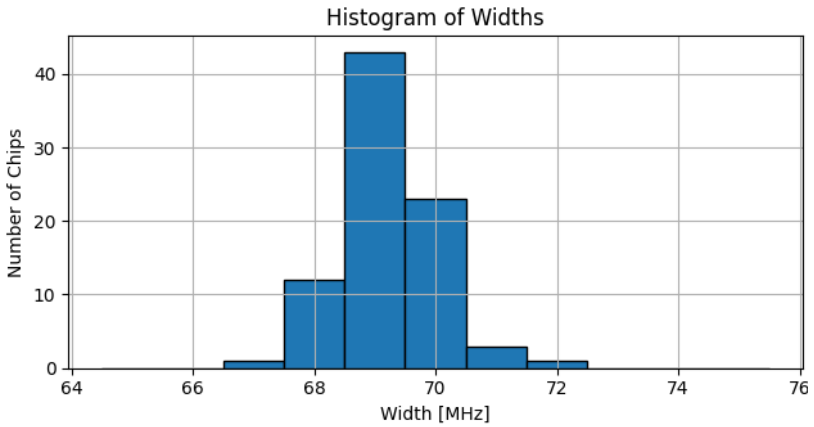


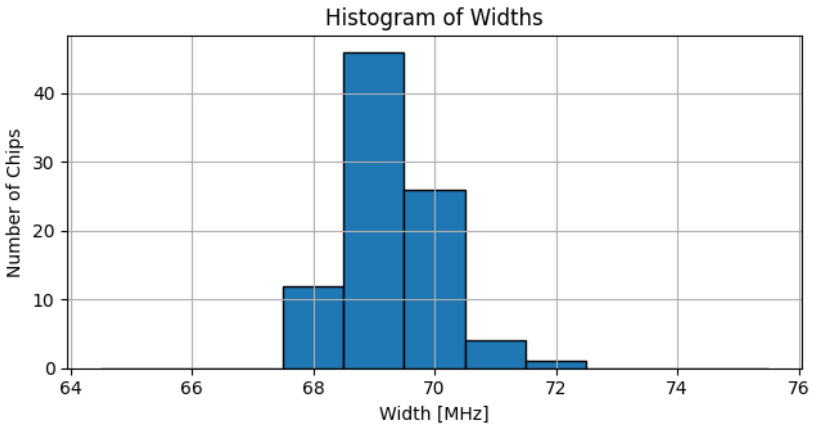

Autolock 1.08

# Complete PLL Lock Graphing Script

Widths Histogram: ECONT July (90 chips)



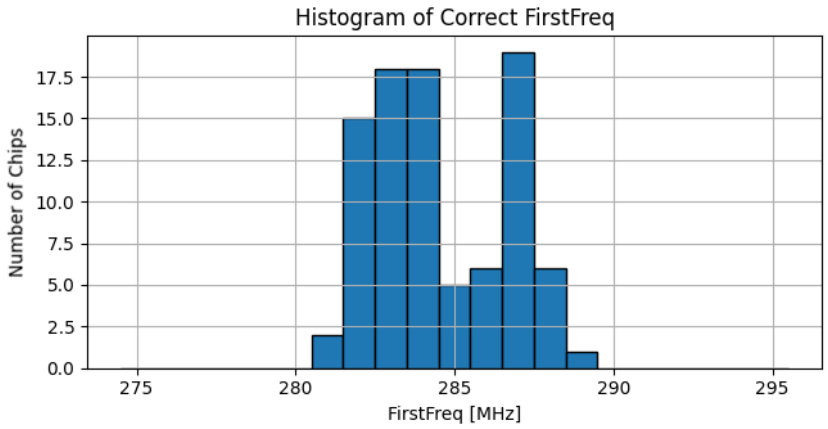

Autolock 1.32

Autolock 1.2

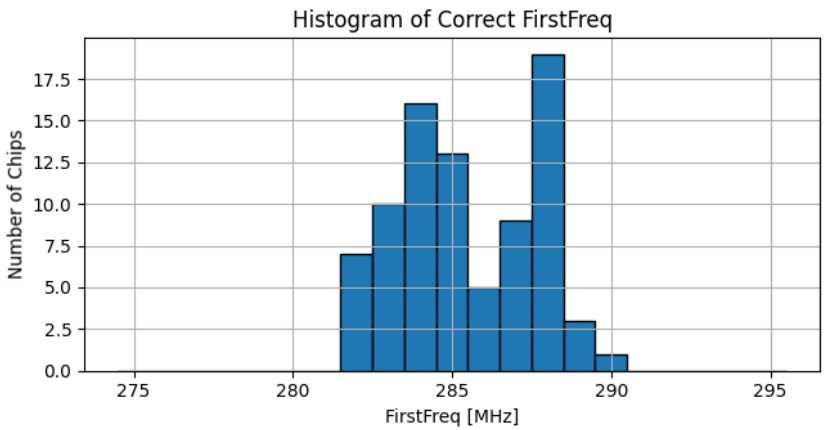
Autolock 1.08

# Complete PLL Lock Graphing Script

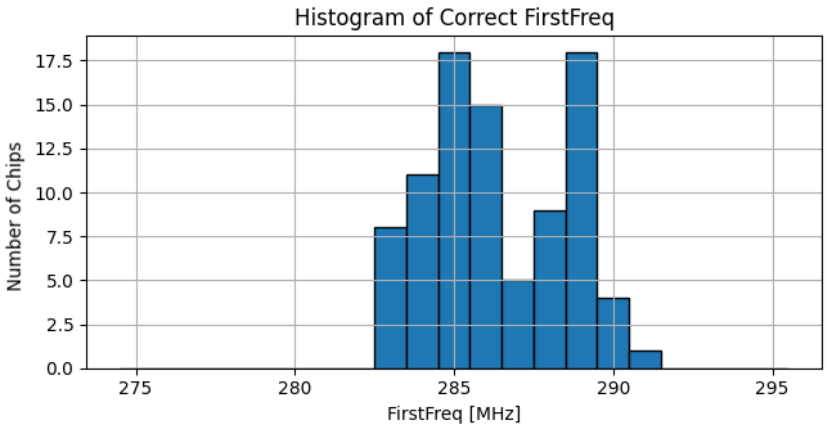
Freq. Histogram: ECONT July (90 chips)




Autolock 1.32



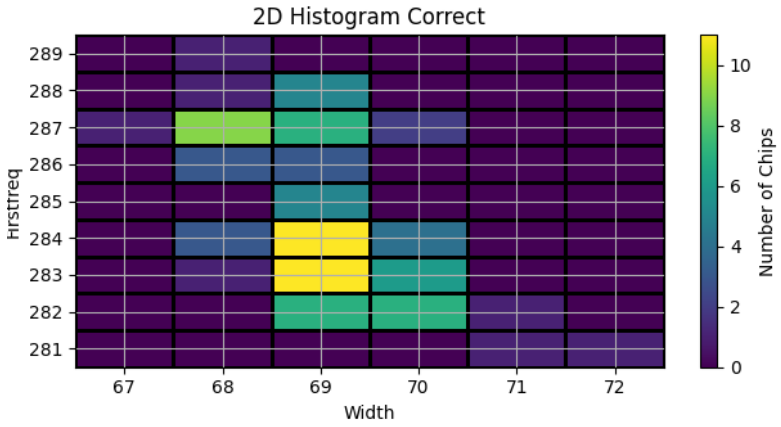

Autolock 1.2



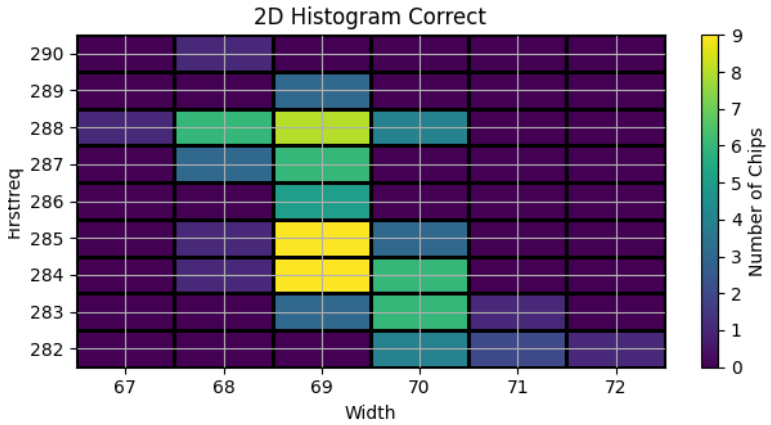

Autolock 1.08

# Complete PLL Lock Graphing Script

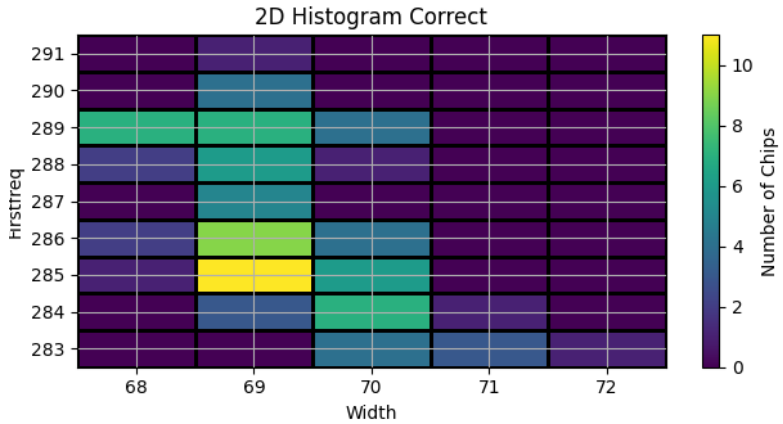
Freq. vs Width 2D Histogram: ECONT June (65 chips)




Autolock 1.32




Autolock 1.2

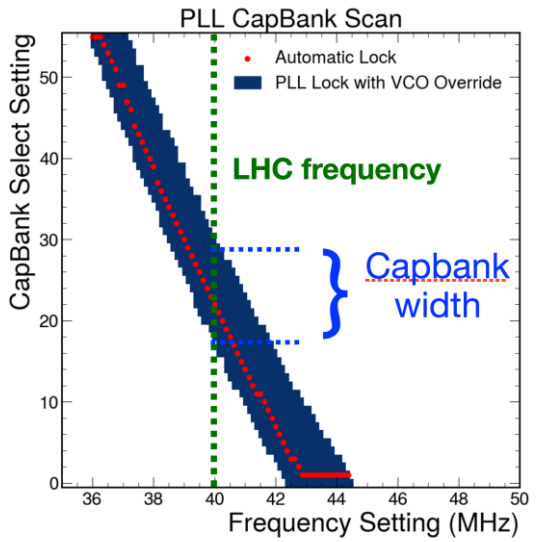


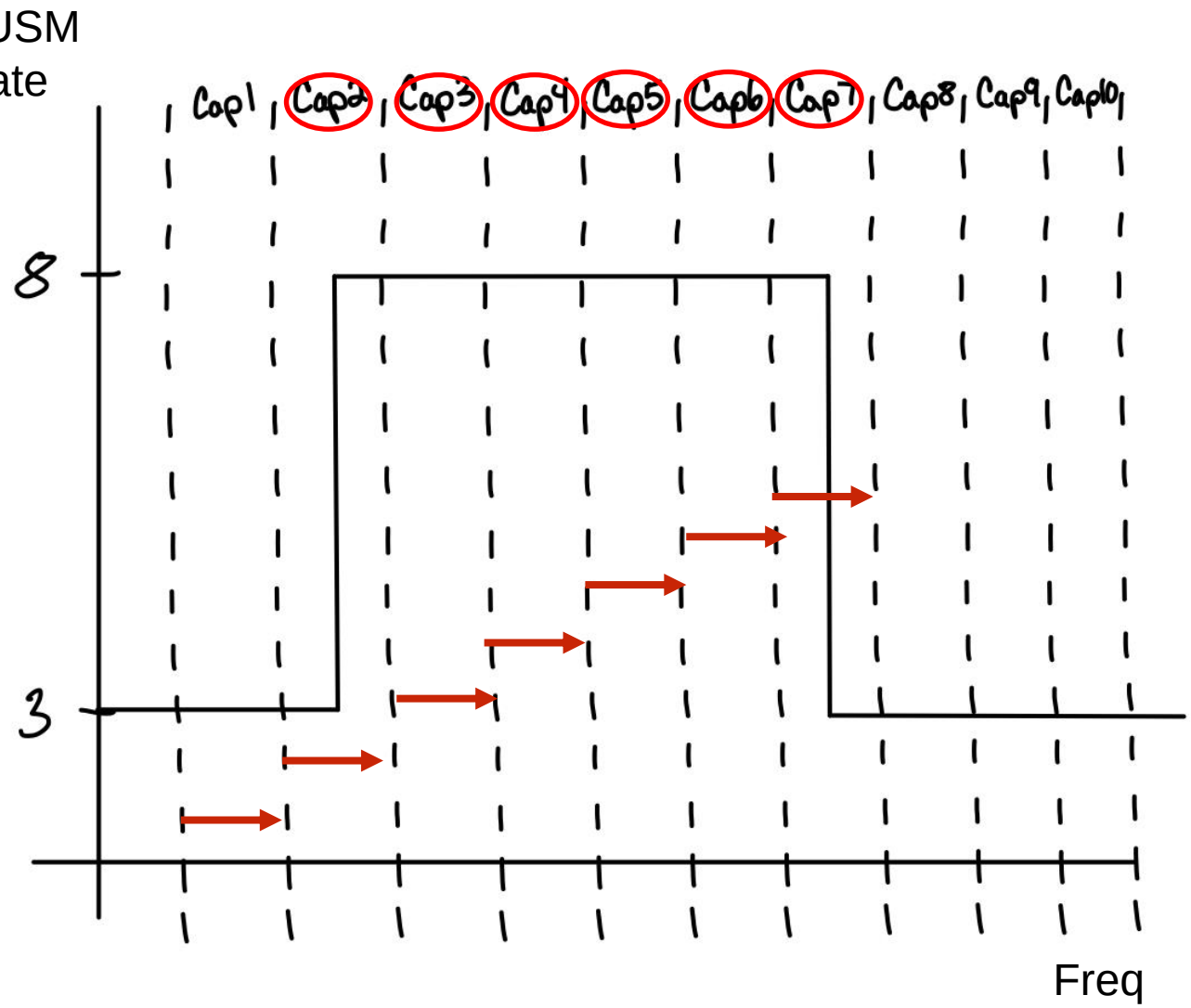

Autolock 1.08

# PLL Capacitance Range Test

# PLL Capacitance Range Test

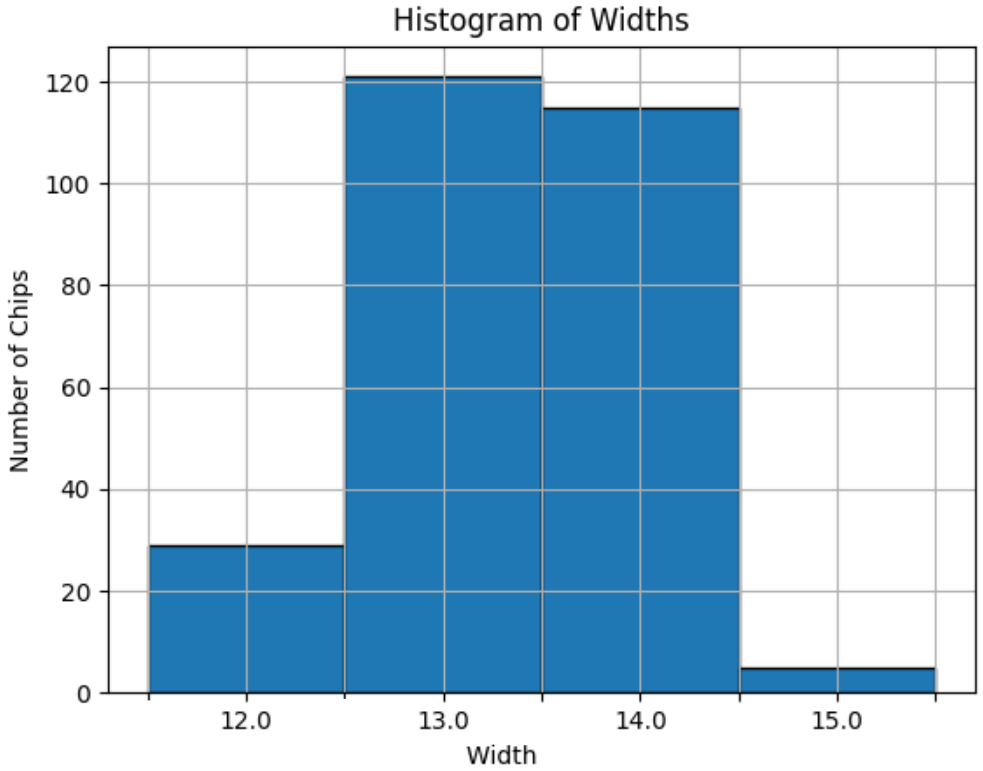
- adjusts the voltage and capacitance, the PLL can lock in the frequency range
- Capacitor Bank : “Capbank”
- the test counts how many capacitors between the PLL Lock
- this is performed at different voltages (1.08, 1.2, and 1.32)
- instead of 40 it would be 8*40=320 MHz



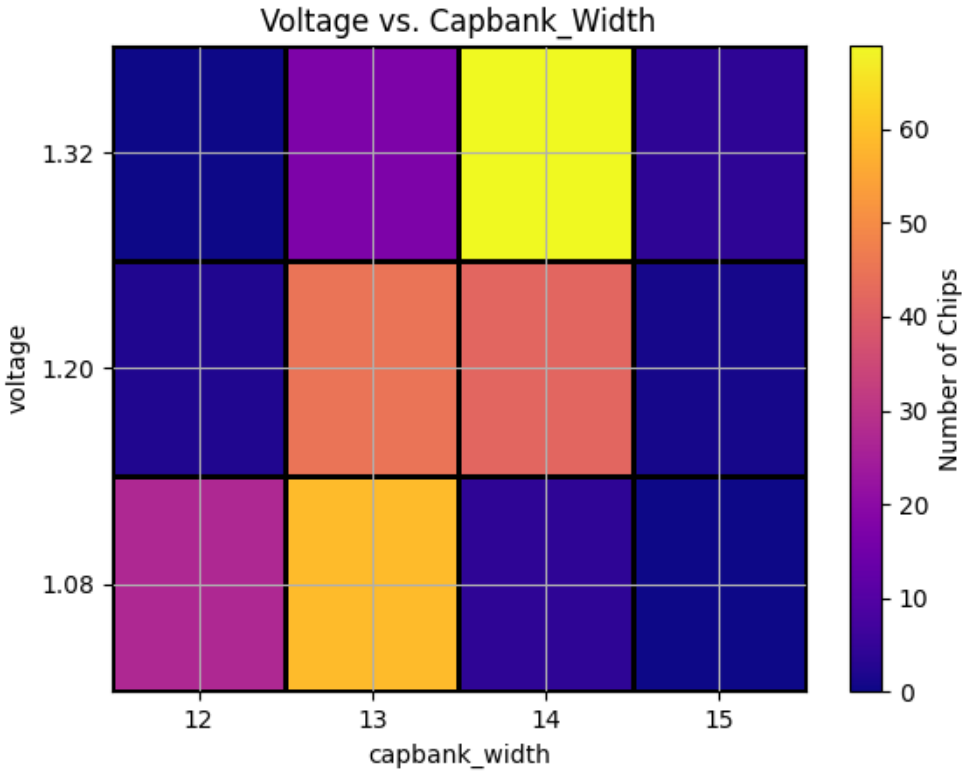



# “Capbank” Width Graphing Script

ECONT July (90 chips)




Capbank Width Histogram




Voltage vs Capbank Width 2D Histogram

# One Chip vs Multiple Chips

Purpose: Is each individual chip operating as it says it is or is it random?

Not Removed vs Removed from socket

Is there human error in placement of the chips?

Is there a difference in the individual graphs completed through different tests?

Methods:

ECOND 261 (965 times) Not Removed “Good”: 4 errors

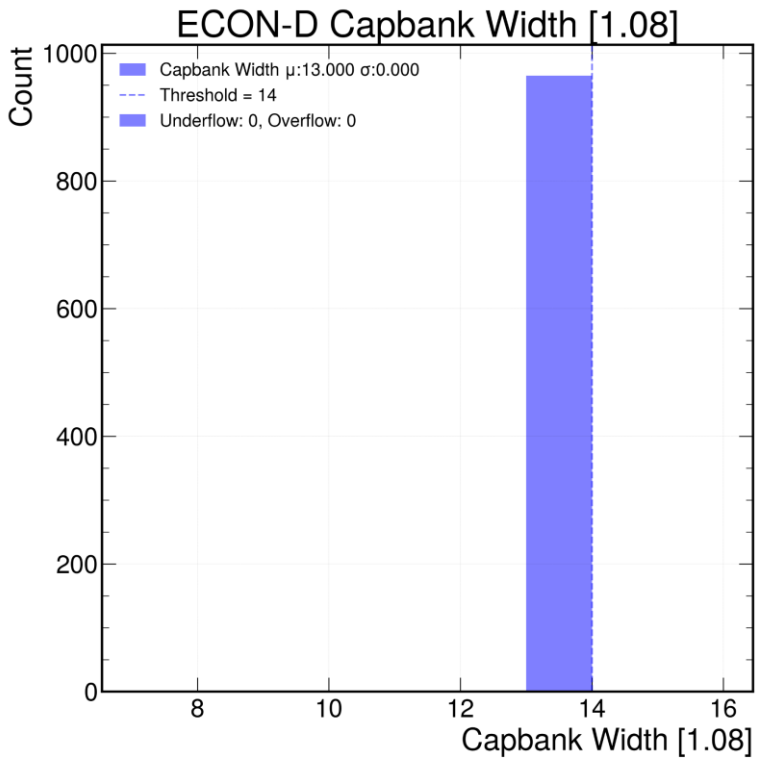
ECOND 261 (100 times) Removed “Good”

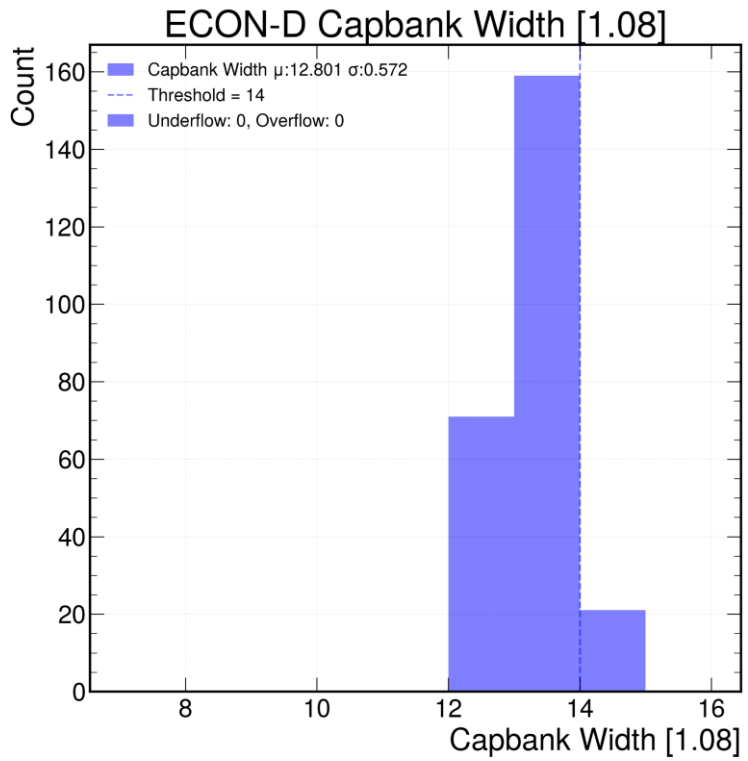
ECOND Multiple (257 different chips) Different Firmwares

Each set of json files are run through all the graphing codes for comparison

# One Chip vs Multiple Chips

ECOND 261 (965 times) Not Removed “Good chip”




ECOND Multiple (257 different chips) Different Firm wares

# Conclusion:

What I have learned:
- Particle Physics Basics
- HGCAL/ ECON projects
- Skills: presenting, critical thinking, problem solving
- software, hardware, data analysis, navigating further research
- independent learning, team collaboration, contributing to a team project,
- career building and path finding

Summer Contributions:
- Individual Chip Analysis Document
- Testing Chips on Board for CERN
- Provided valuable insight for ECON chip learning
- Results from one chip vs multiple chips
- ideas for the finalized test pyscripts: Notice to needed changes to the “passed” parameters and PLL Lock Test changes

Potential Future Contributions:
- Remote ECON testing, writing pyscripts, and plots to continue contributing to the ECON chip
- ECONT prototype Neuro network that uses machine learning: small group learning to train and evaluate how well it compresses data.

# Acknowledgement

*"This work was supported by the DOE grant RENEW-HEP: U.S. CMS SPRINT - "A Scholars Program for Research INTernship" at Tougaloo College, MS (DE-SC0023681), Brown University (DE-SC0023651), RI, University of Puerto Rico-Mayaguez, PR (DE-SC0023680), and University of Wisconsin – Madison, WI (DE-SC0023643) and U.S. CMS Operations at the Large Hadron Collider (NSF-2121686)"*

# Any Questions?

Thank you!

PURSUE internship and colaborators
Fermilab and everyone involved
My mentor and the team