FERMILAB-SLIDES-25-0122-ETD

Fermilab

FERMI NATIONAL ACCELERATOR LABORATORY

June 21, 2025

An Open-Source Framework for Rapid Validation of Scientific ASICs (Spacely)

Adam Quinn
ASIC Development Engineer

U.S. DEPARTMENT of ENERGY

Fermi National Accelerator Laboratory is managed by FermiForward for the U.S. Department of Energy Office of Science

This manuscript has been authored by FermiForward Discovery Group, LLC under Contract No. 89243024CSC000002 with the U.S. Department of Energy, Office of Science, Office of High Energy Physics.

Agenda

Today

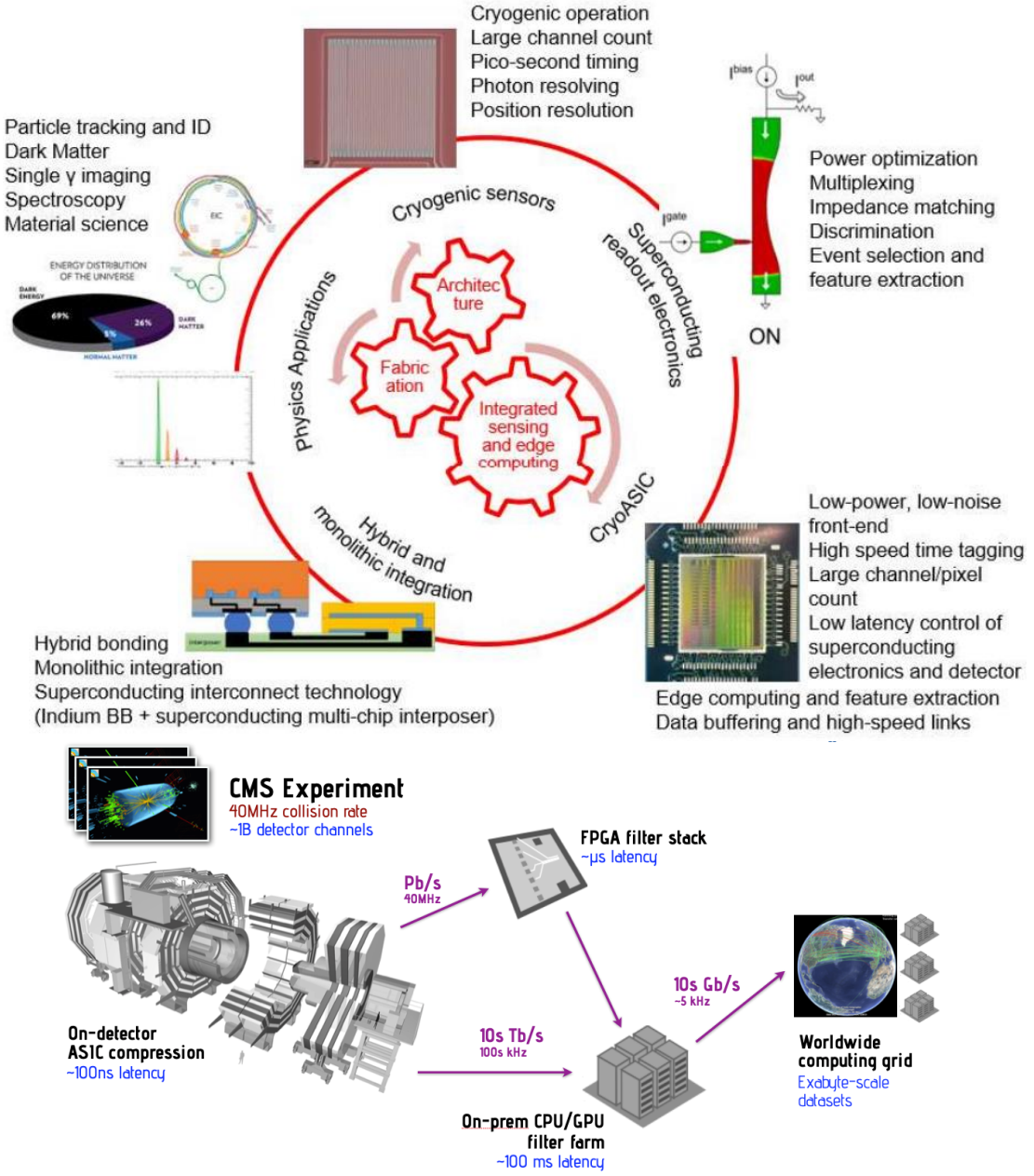
- ✓ Background
- ✓ Spacely
- ✓ Spacely Workflows & Integrations
- ✓ ASICs Tested with Spacely

Fermilab Microelectronics Division

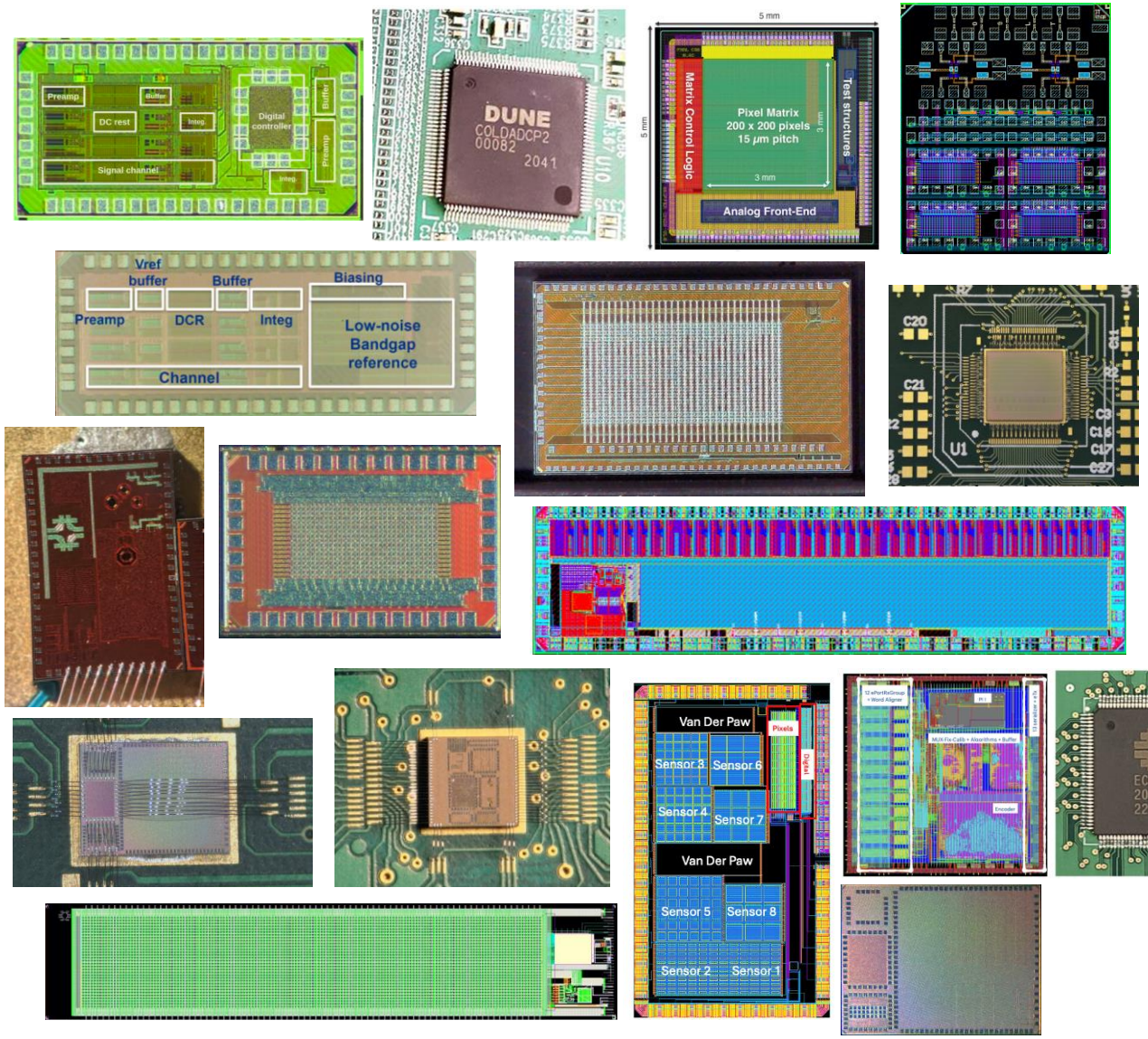
Mission: Leverage novel microelectronics to solve the mysteries of matter, energy, space, and time for the benefit of all.

Portfolio including:

- Detectors for collider experiments (> **1Grad ionizing radiation**, 1000x higher than outer space)
- Deep cryogenic electronics for **dark matter + Quantum Information Science** (SNSPDs, Skipper CCDs, TES, etc.)
- **Supeconducting** (TWPAs, JPAs), integrated **photonics** (cryo-MRMs)
- Low-volume **hybrid integration** + co-design
- **AI-on-chip**, edge compression / data sparsification.

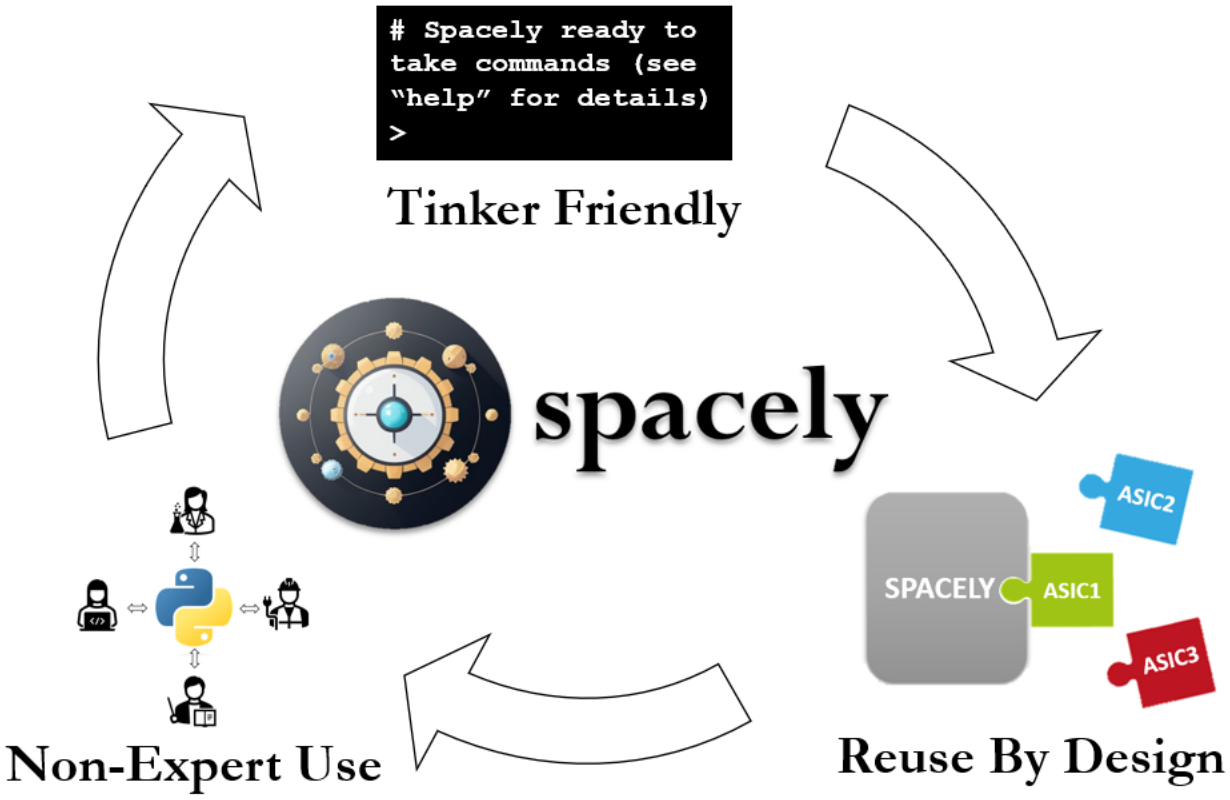


Problem Statement



- We design a lot of chips.
- These are not incremental revisions: different technologies, operating conditions, and specs.
- Limited time and resources for test.
- Primary stakeholders (scientists) are not experts in chip design *or* test.
- To get chips tested on time:
  - (1) Multiply small pool of expert resources.
  - (2) Pull in as much non-expert help as possible.
  - (3) Reuse everything.

What is Spacely?



Spacely is:

- An **open-source, Python-based test automation framework**

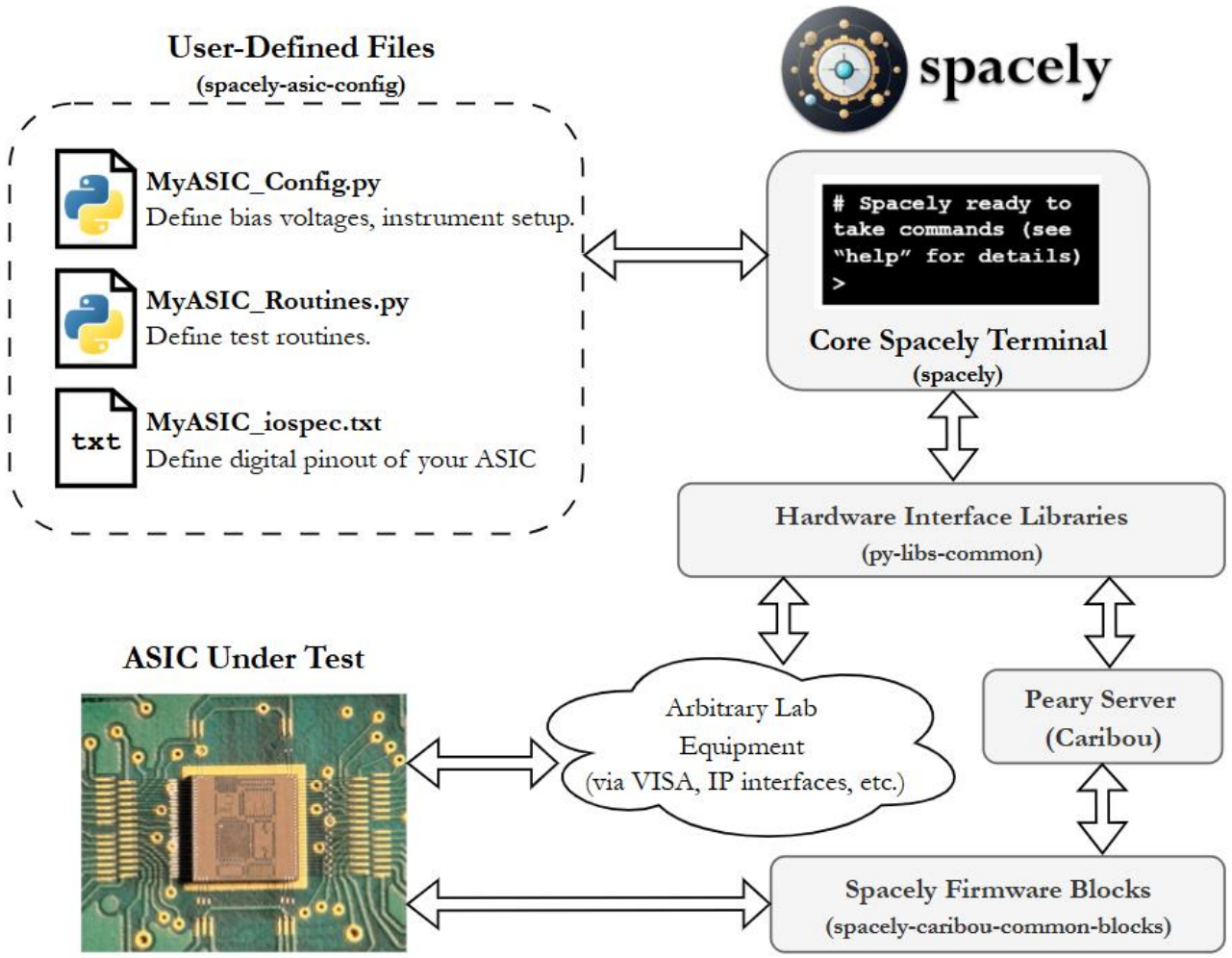
Spacely contains:

- A core **Python Github repository**
- **Instrument Libraries**
- **Reusable Firmware**
- Recommended **design flows**

Spacely is for:

- Small ASIC design teams in **research / academia**
- Non-expert (i.e. scientific) users

What is Spacely?



- Users write a config file which is used by Spacely to initialize benchtop instruments.
- Test routines are recorded as native Python routines.
  - Documents the procedure to obtain a result.
- Spacely idioms allow quick control of the DUT via the terminal.
- Spacely interfaces to benchtop instruments via VISA/GPIB/etc in addition to FPGA targets (more on that later).

Why Spacely?

Bespoke in-house test systems

- × Need in-house expertise to build + maintain.
- × Can't be reproduced elsewhere.
- × Only work for one specific ASIC or architecture.

Commercial Products

- × Expensive upfront + license costs.
- × Locked into vendor offerings, may not be easy to network with other tools.

spacely

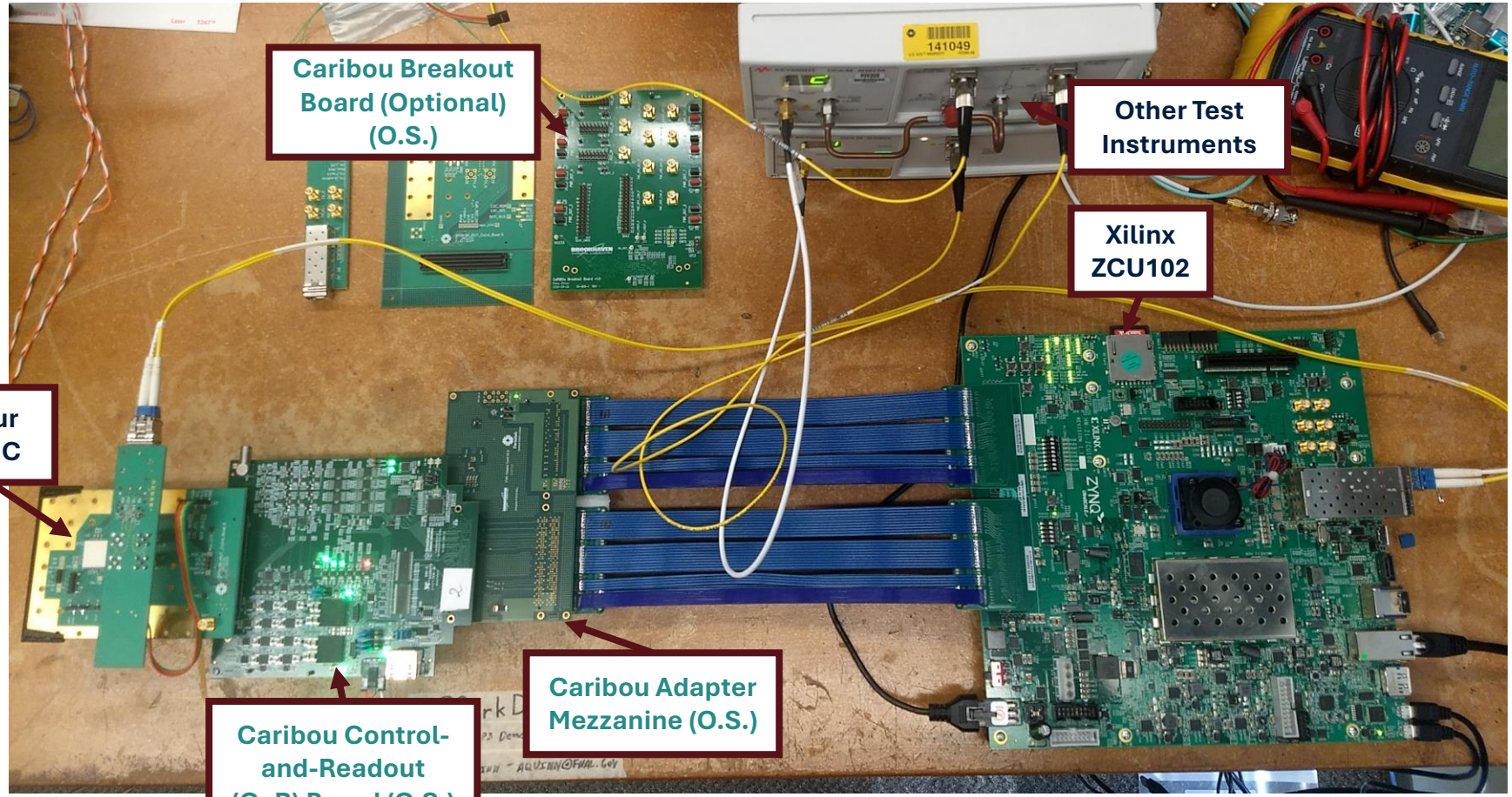
- ✓ Open-source software + hardware
- ✓ Python dev environment + FPGA performance.
- ✓ Standard process flows + customization with scripts.
- ✓ Share results – and scripts – with non-expert collaborators.

Spacely + Caribou

Open-source Python-based test automation framework.

spacely +

Open-source test hardware and firmware.



Curious about Caribou?

Caribou is a versatile data acquisition system for silicon pixel detectors including custom open-source hardware, firmware and software.

https://caribou-project.docs.cern.ch/

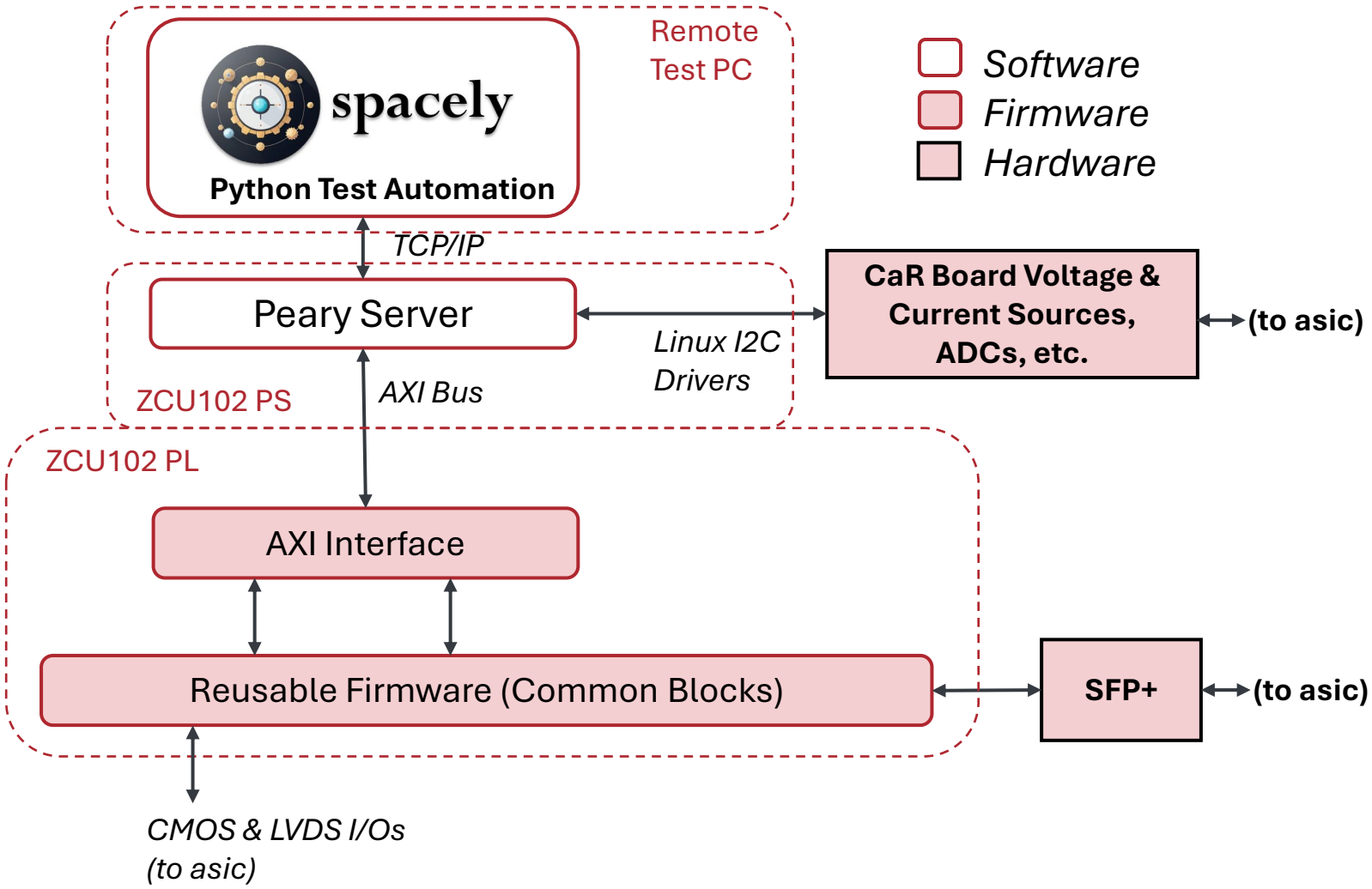
Developed By:

Brookhaven National Laboratory · Oak Ridge National Laboratory · DESY. · CERN · Carleton University · AIDA innova · EP R&D · RD50

FERMI NATIONAL ACCELERATOR LABORATORY

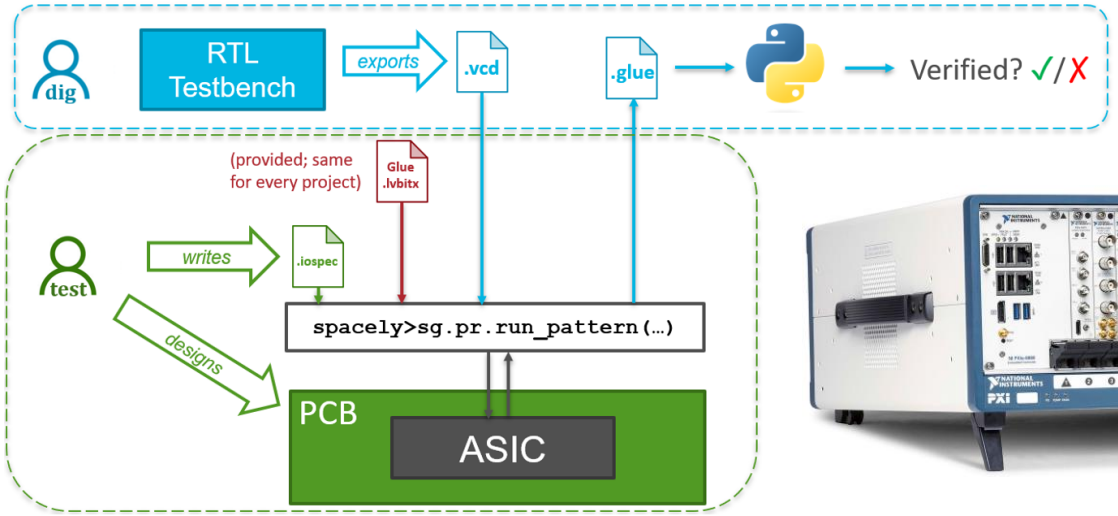
Spacely + Caribou: Firmware Integration

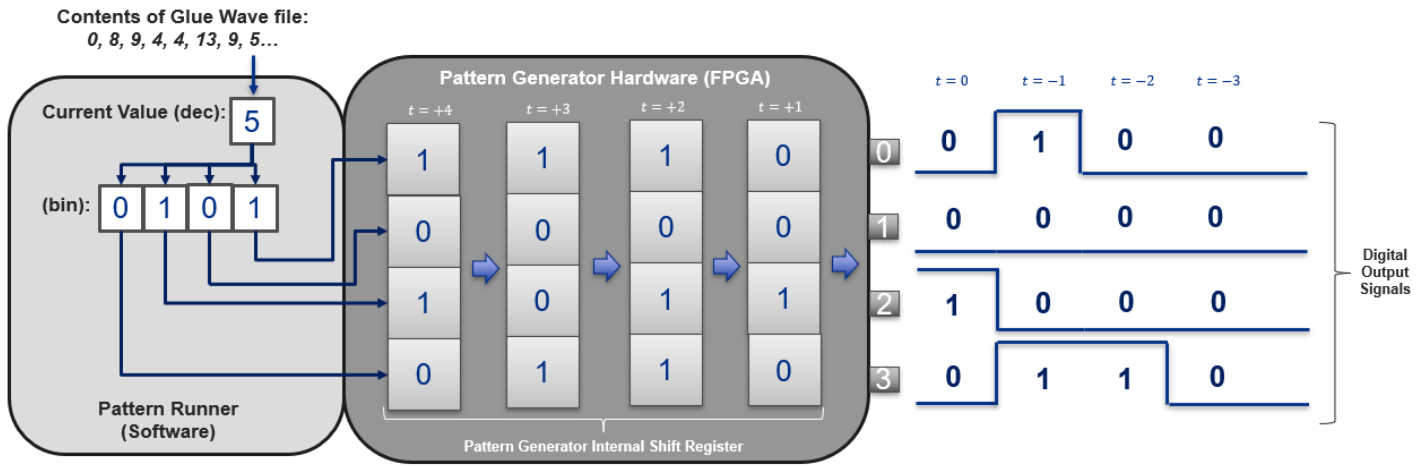
- Spacely integrates w/ Caribou software (Peary) to access full suite of mixed-signal test capabilities: ADCs, DACs, current/voltage sources…
- Provide a repo of **reusable firmware blocks** to simplify firmware design.
  - https://github.com/SpacelyProject/spacely-caribou-common-blocks
- Common AXI interface → Plug & play w/ Vivado Block Designs
- Use **dr_gth** block to access 10.24 Gb/s **SFP+ Optical Tx/Rx** via



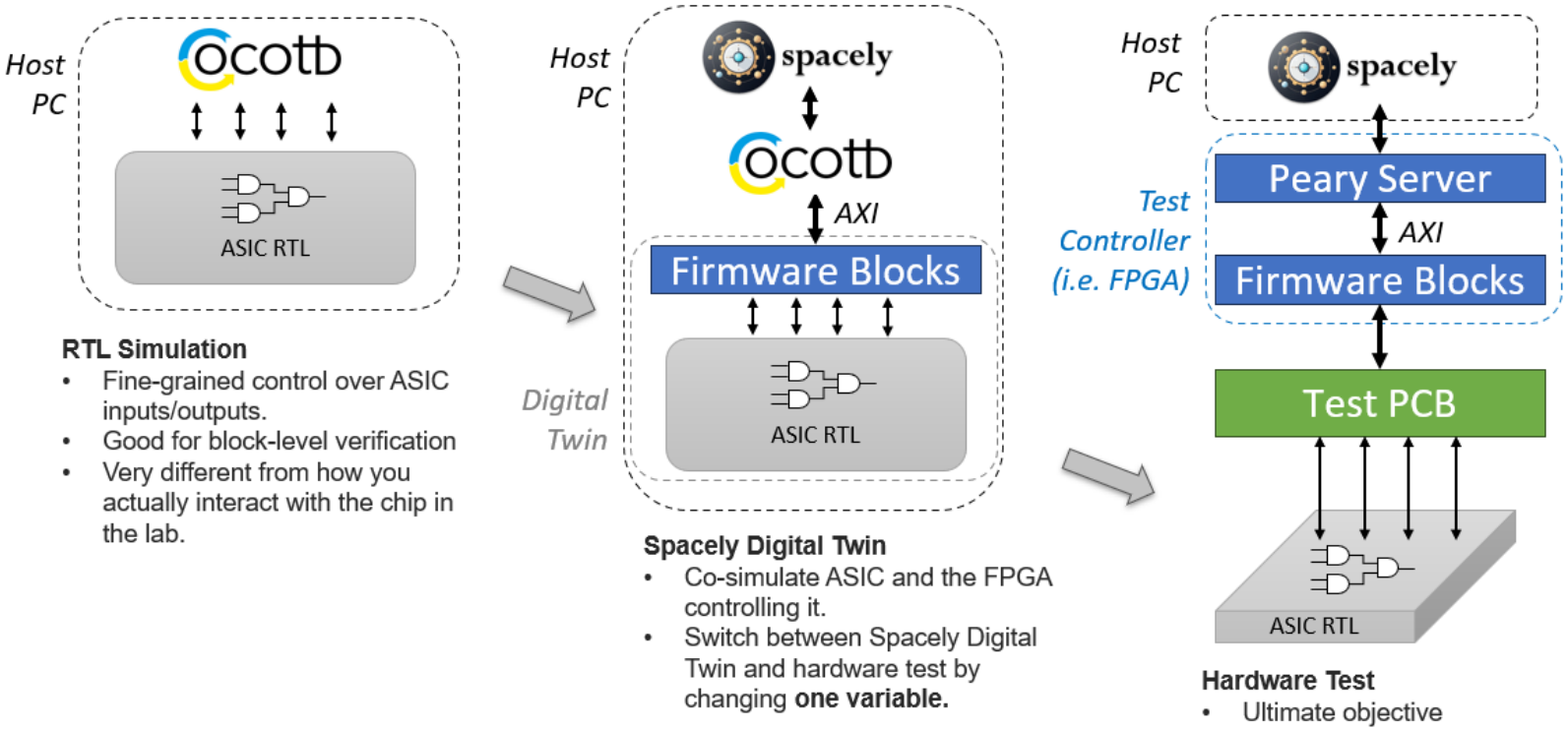
Spacely + National Instruments

- NI-PXI: Industry-standard extensible slots for test equipment.
  - High-bandwidth FPGA cards… but need LabView to program ☹
- Spacely Pattern Generator:
  - Single generic IP, use NI FPGA as a high-bandwidth Python-programmable pattern generator.
- **Export VCD from digital testbench** or generate natively in Python.





Spacely + Digital Twin



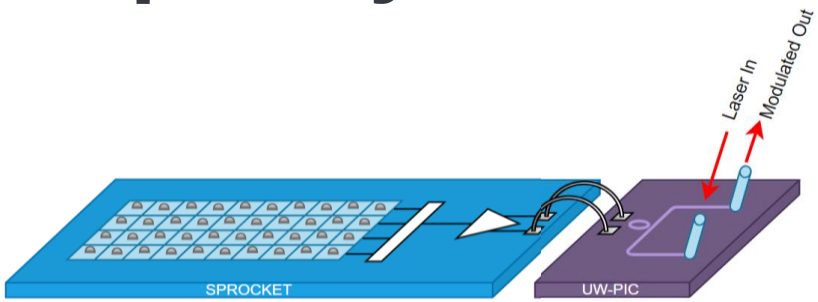
Problem:

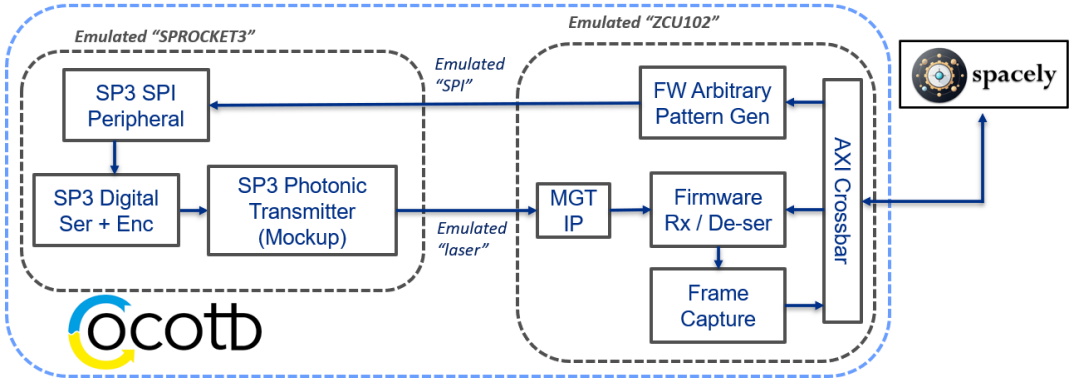
- **FPGA Test Vectors ≠ RTL simulation**
- Issues may crop up due to the FPGA infrastructure, causing delays.

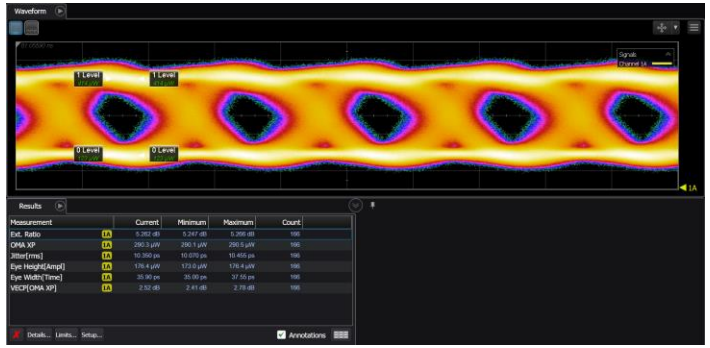
Solution:

- **Full digital twin –** simulate FPGA firmware w/ Cocotb + AXI
- **One-click switch** between running on hardware vs digital twin.

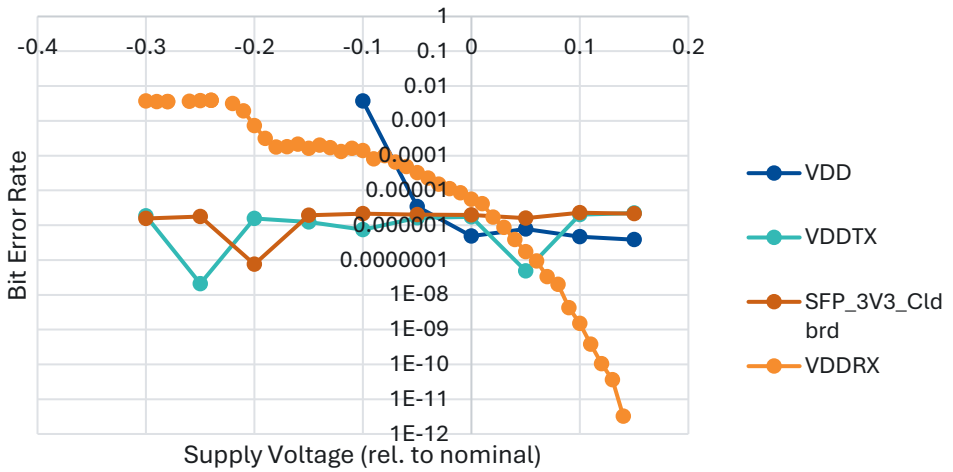
Spacely + SParkDream





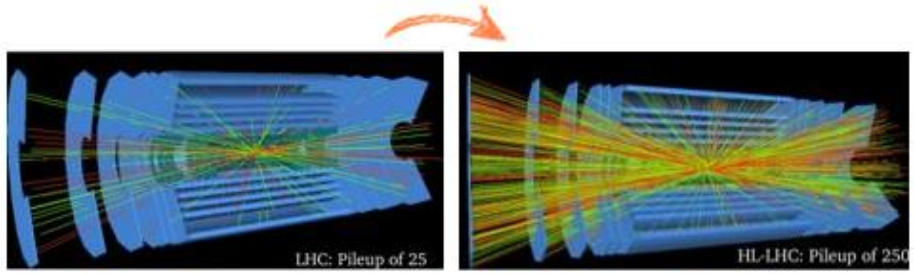


BER vs Supply Voltage Deviation (from Nom. 1.2V)



- **SParkDream** is Fermilab's internal program to develop silicon photonic capabilities.
- Complex demonstrator involving CMOS ASICs, photonic integrated circuits, lasers, and FPGA readout.
- Spacely integrates all parts of the system, allowing us to easily sweep supply voltages and measure bit error rates (left).
- When chips were delayed, Spacely's **Digital Twin Flow** was used to simulate the entire readout path (top).

Spacely + CMSPIX



	Currently being Installed	Beyond 2030:What we are investigating
Technology	**65nm CMOS**	**28nm CMOS**
Pixel ROIC size	**50x50 µm²**	**25x25 µm²**
Pixel Sensor size	**100x25**	**50x12.5 µm²**
Pixels	394x400 = 157.6k	788x800 = 0.63M
Detection threshold	**~1000e-**	**~500e-**
Hit rate	**< 3GHz/cm²**	**< 3GHz/cm²**
Trigger rate	**1MHz**	**40MHz (?)**
Digital buffer	**12.5 µs**	**(?)**
Readout data rate	1-4 links @ 1.28Gbps	Photonic link @ 30-100 Gbps
Radiation tolerance	**500Mrad at -15°C**	**1Grad at -15°C**
Power	**1 W /cm²**	**1 W /cm²**

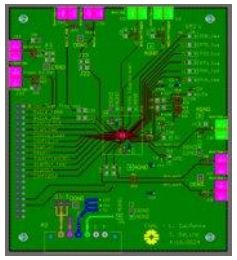
- CMSPIX ASIC currently under design for the Phase II Upgrade of the Large Hadron Collider
- Smaller pixels / greater trigger rate needed to accommodate increased luminosity + pileup.
- Developed by the **Smartpixels Collaboration** (multiple universities + Fermilab):
  - How to enable university partners to participate in testing?
  - How to ensure consistent results across test setups?
- Spacely-based test stand:

python spacely

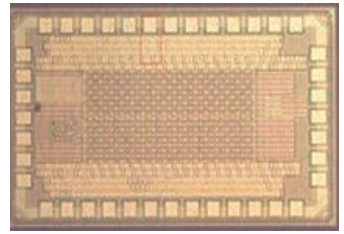
Python Inteface

Zynq UltraScale+ MPSoC ZCU102 Eval Board

Caribou Board (Open-source DAQ System)



Custom ASIC board



CMSPIX ASIC

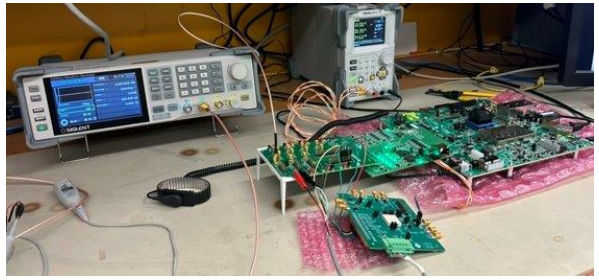
Spacely + CMSPIX: Current Status

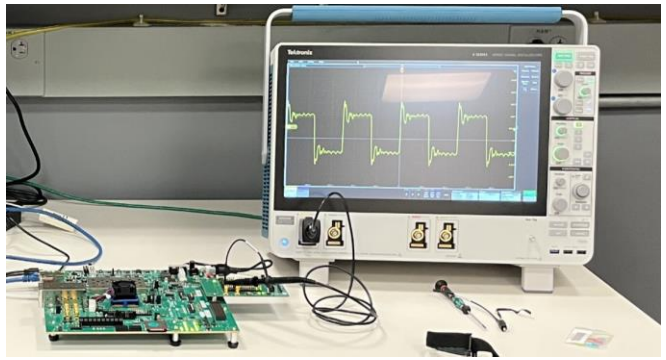
Current status: Four test stands running at three different institutions with eight people (students, researchers, engineers) involved in testing.

Fermilab

Cornell University

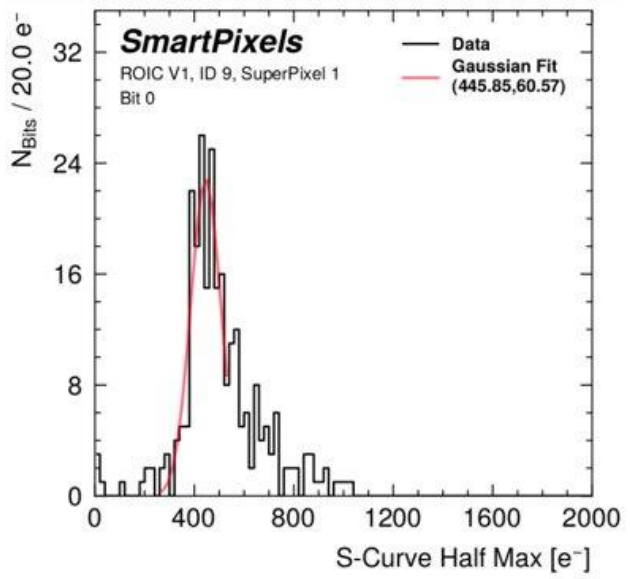
THE UNIVERSITY OF CHICAGO



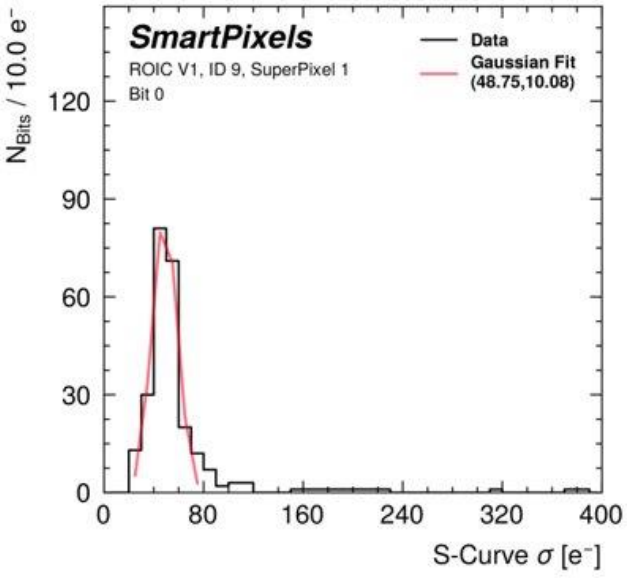


Measured Results

Q_{TH} (Extracted 60.6e-)



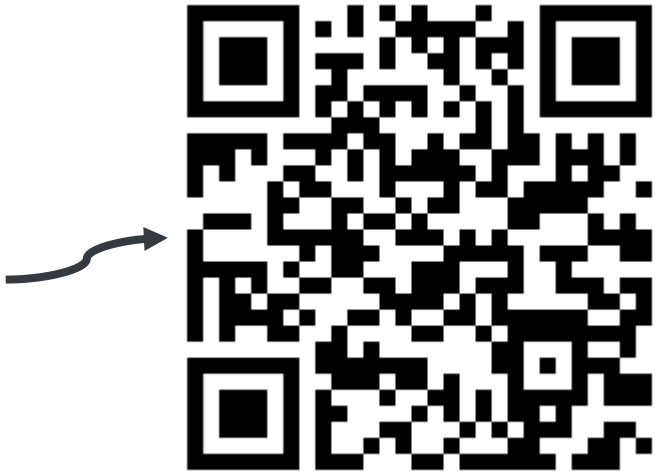
ENC (Extracted 48.9e-)



FERMI NATIONAL ACCELERATOR LABORATORY

Let's Collaborate!

Read the docs on Github:
https://github.com/SpacelyProject/spacely-docs

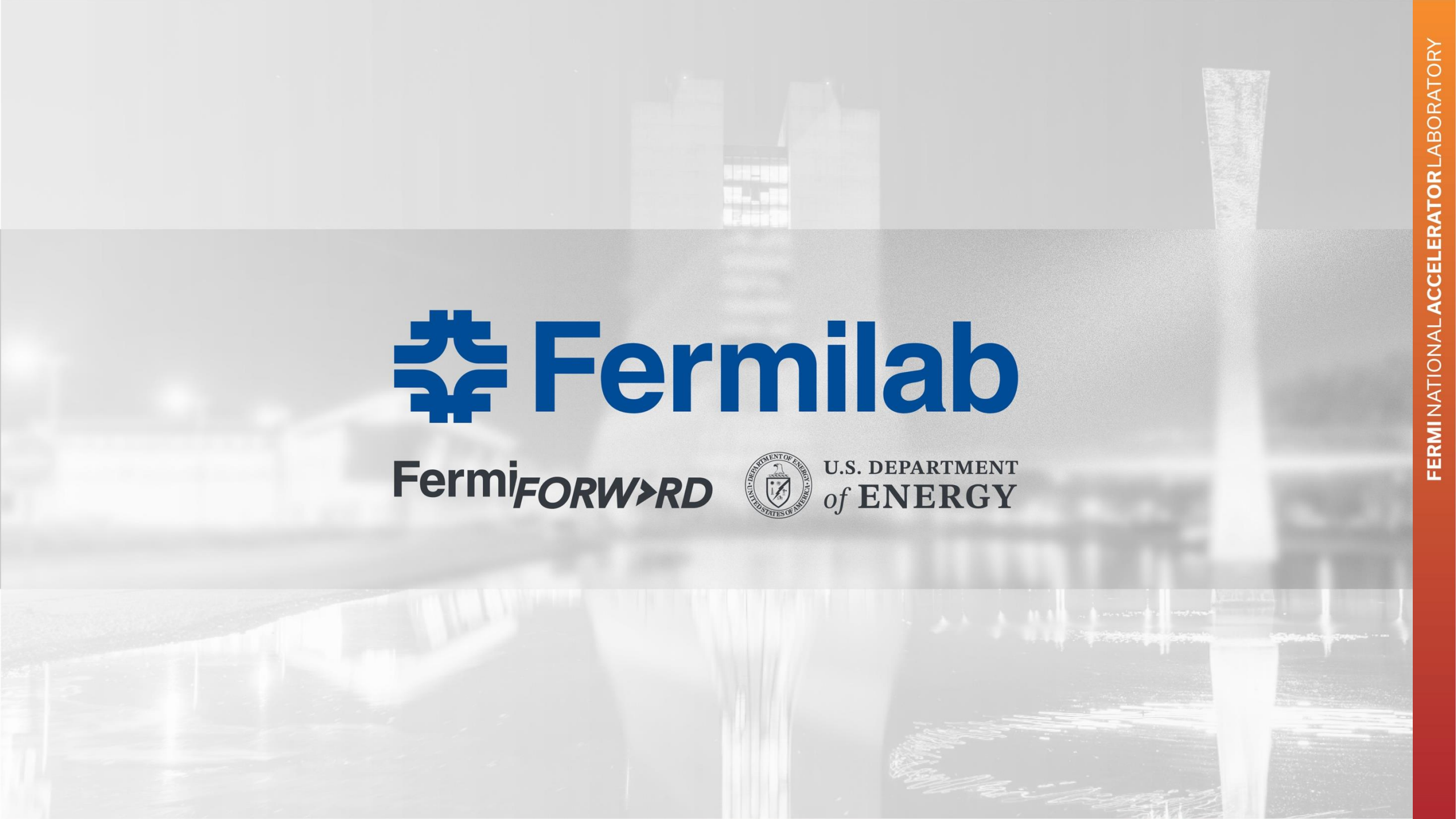


Chat with us at ISCA '25

Continuously looking to onboard new users / developers.

OR send us questions by email:

Spacelydevelopers [at] fnal [dot] gov


Fermilab
FermiFORWARD
U.S. DEPARTMENT of ENERGY
FERMI NATIONAL ACCELERATOR LABORATORY