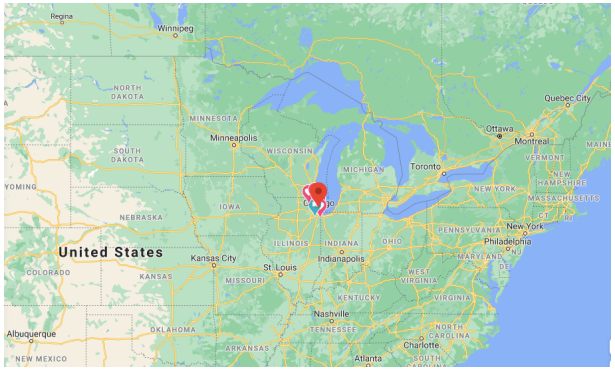
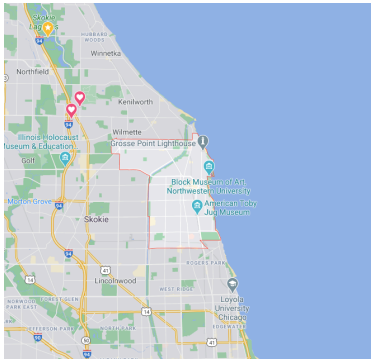
Northwestern





Computer Engineering Research @Northwestern

- Housed within the Electrical and Computer Engineering Department, co-owned by the Computer Science Department
  - Faculty have dual appointments
  - PhD programs in both departments have Computer Engineering tracks
- Design Automation and VLSI, Computer Architecture, Internet of Things, Embedded Systems, Big Data & AI

Computer Engineering Research @Northwestern

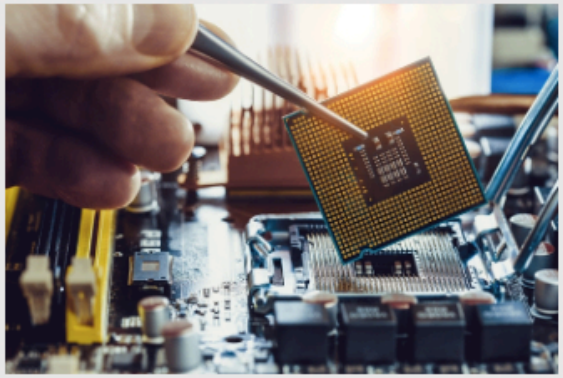
- Hardware assisted ML
- Neuromorphic computing
- Hardware security
- Photonic circuits in computer architectures
- Emphatic computing
- Batteryless computing
- Secure and verified software for automotive application
- Adaptive architecture – compiler in the loop

Research Interests

- Electronic Design Automation (EDA)
- Circuits and Systems
- High Performance Computing (HPC) Systems

Research Activities

- Design Automation
  - High-Level Synthesis
    - Converting applications described with programming languages to hardware
- Circuits and Systems
  - Thermal and Power Management of ICs
    - Temperature Sensors, 3D Integration, Power and Performance Modeling using Machine Learning



The end of an era

As improvements to the silicon chip come to an end, where will high-performance computing go next?

Research Activities

- High Performance Computing Systems
  - FPGA-based accelerators for HPC applications
  - Thermal and Power Management in large scale systems

Microsoft Azure: It's getting hot in here, so shut down all your cores

US customers wake up to sleepy cloud service

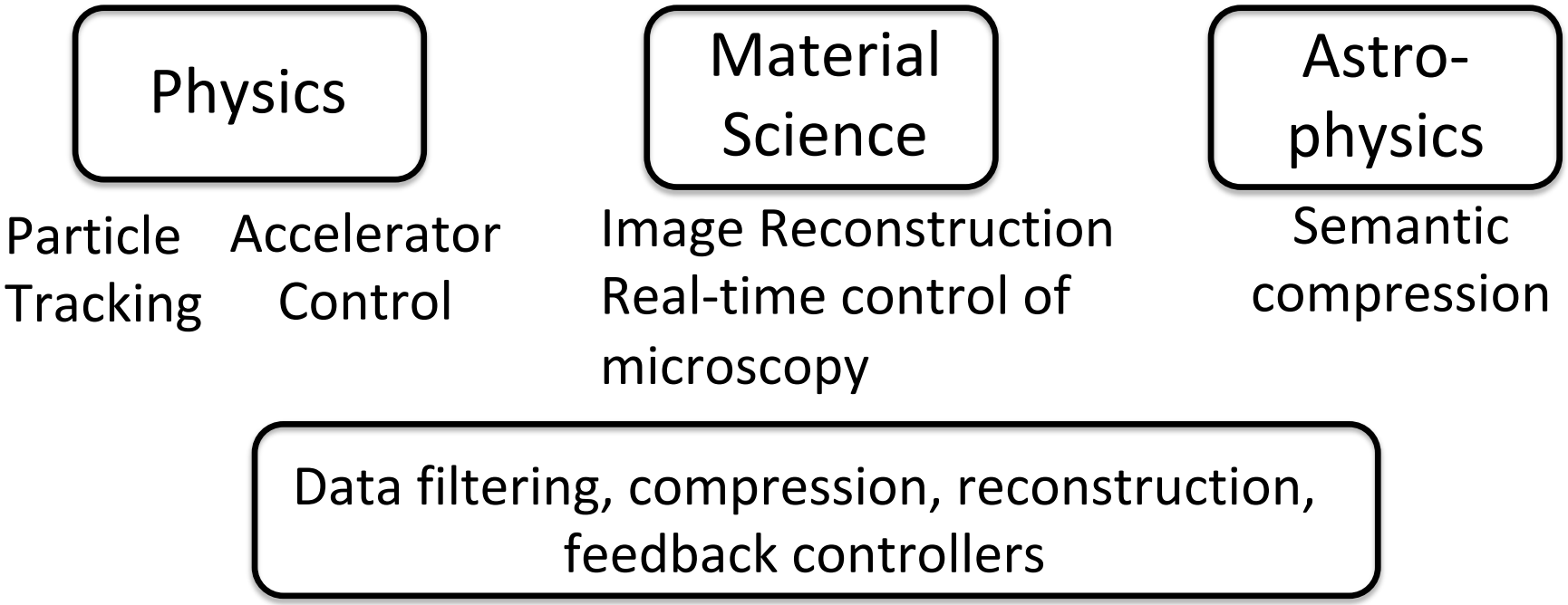
By Richard Speed 4 Sep 2018 at 13:19 53 SHARE

Updated Microsoft has warned that a "subset of customers in South Central US" may experience Azure problems today after cooling issues sent the servers scurrying for the shutdown button.

The warning was first raised by Microsoft at 09:29 UTC as pretty much everything in the South Central US region went offline thanks to a temperature spike that caused servers to automatically shut down to avoid damage.

HW Assisted ML

- Design Automation for real-time AI
- Cyberinfrastructure for scientists who deploy HW systems for ML



Handling Big Data

- Source of the computational challenge
  - Particle acceleration experiment in high-energy physics
    - Billions of “event” happen every ~25ns creating Pb/sec data rates
    - “Interesting” events (~3% of all) need to be recognized and filtered for further processing
  - Sky survey data collected from observatories stream in at rates ~Gb/s (or more)
    - Limited energy (generators in South Pole) to move data to datacenters

Handling Big Data

- Source of the computational challenge
  - CPU-GPU systems perform streaming inference on images captured by an electron microscope within ~300ms latency
  - Desired goal is to perform inference on images captured at 10s of millions of frames per second under 50ms latency
  - ~50ms latency could allow real-time control of the EM ***during*** material synthesis
    - Think of making defects for a quantum material, deposition of nano layers, etc.

Key areas of Overlap: System Constraints

Table 1. Domains and practical constraints.

Domain	Data	Event Rate	Latency	Systems	Energy Constrained
Detection and Event Reconstruction					**No**
HEP	Spatial Point Cloud	10s Mhz	us-ns	CPU-FPGA-ASIC	
Nuclear Physics	Spatial Point Cloud	10s kHz	ms	CPU	
Dark Matter - Neutrino	Temporal	10s MHz	us	CPU-FPGA	
Image Processing	**Spatial**				
Material Synthesis	Temporal	10s kHz	ms	CPU-FPGA	
Scanning Probe Microscopy	Temporal	kHz	ms	CPU-FPGA	
Electron Microscopy	Temporal	MHz	us	CPU-FPGA	
Biomedical Engineering	Temporal	kHz	ms	CPU-FPGA-ASIC	Yes (mobile settings)
Cosmology		Hz	s	CPU	
Astrophysics		kHz-MHz	ms-us	CPU-FPGA	Yes (remote locations)
Signal Processing	Spatial				
Gravitational Waves		kHz	ms	CPU	
Health Monitoring	Temporal	kHz	ms	FPGA-ASIC	Yes
Communications	Temporal	kHz	ms	CPU-FPGA	Yes (mobile settings)
Control Systems	**Temporal**				
Accelerator Controls		kHz	ms-us	CPU-FPGA	

Key areas of Overlap: System Constraints

Table 2. Examples of different data representations.

Domain	Data Type	Examples
Detection and Event Reconstruction	Spatial Point Cloud	Detector coordinates in space
	Energy signatures	Neutral and charged current measurements
Image Processing	**Spatial**	
Material Synthesis	Temporal	high-speed video of plasma plume
Scanning Probe Microscopy	Temporal	conductivity, piezoresponse
	Hyperspectral time series	resistivity, magnetic force
Electron Microscopy	Temporal, Hyperspectral	diffraction patterns
Biomedical Engineering	Temporal	cell, tissue images, videos, genomic sequence
Cosmology	Multispectral	images, temperature measurements
Astrophysics	Hyperspectral	images, electromagnetic spectrum
Signal Processing		
Gravitational Waves	Spatio-Temporal	laser interference patterns
Health Monitoring	Spatio-Temporal, Multispectral	physiological sensor data, audio
Communications	Temporal, Multispectral	electromagnetic spectrum
Control Systems	**Temporal**	
Accelerator Controls		sensor readings

Key areas of Overlap: System Constraints

Table 3. Classification of domains and their system requirements with respect to real-time needs.

Domain	Real-time Data Processing	Real-time analysis	Closed-loop Control
Detection and Event Reconstruction			
HEP	Yes	(only for anomaly detection)	No
Nuclear Physics	Yes	No	No
Dark Matter - Neutrino	Yes		No
Image Processing			
Material Synthesis	Yes	Yes	Yes
Scanning Probe Microscopy	Yes		
Electron Microscopy	Yes		
Biomedical Engineering	Yes		
Cosmology	Yes	No	No
Astrophysics	Yes	No	No
Signal Processing			
Gravitational Waves	Yes	No	No
Health Monitoring	Yes	Yes	Yes
Communications	Yes	Yes	Yes
Control Systems			
Accelerator Controls	Yes	Yes	Yes
Plasma Physics	Yes	Yes	Yes

Projects – Complete/Underway

- READS – Accelerator Real-time Edge AI for Distributed Systems
- Design of a reconfigurable autoencoder algorithm for detector front-end ASICs

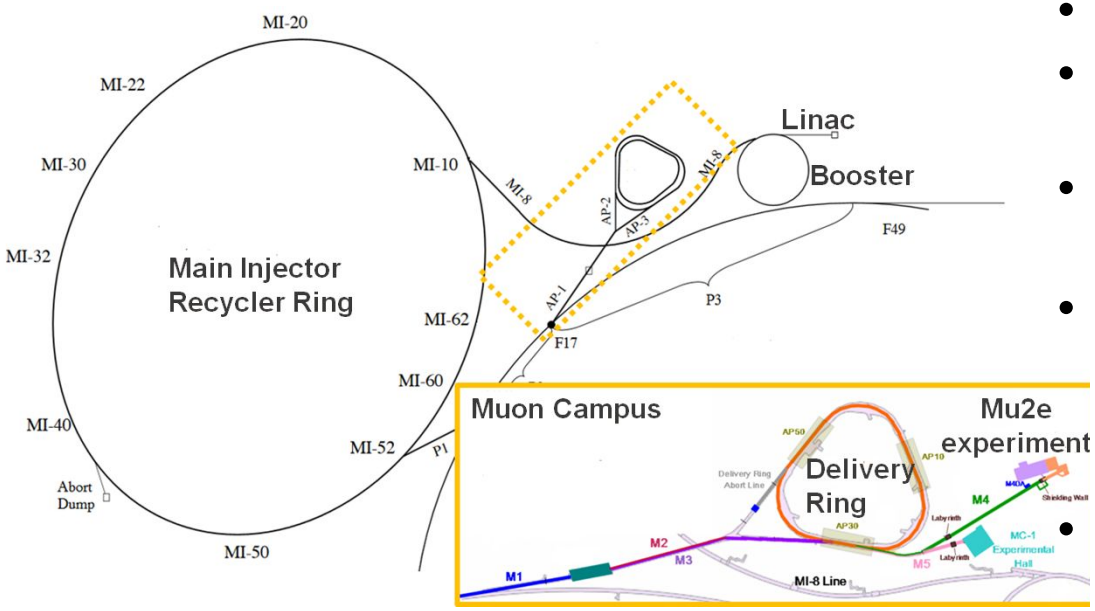
Projects – Preliminary

- CryoAI - 22nm testchip
- Adaptive ML accelerators

READS – Accelerator Real-time Edge AI for Distributed Systems

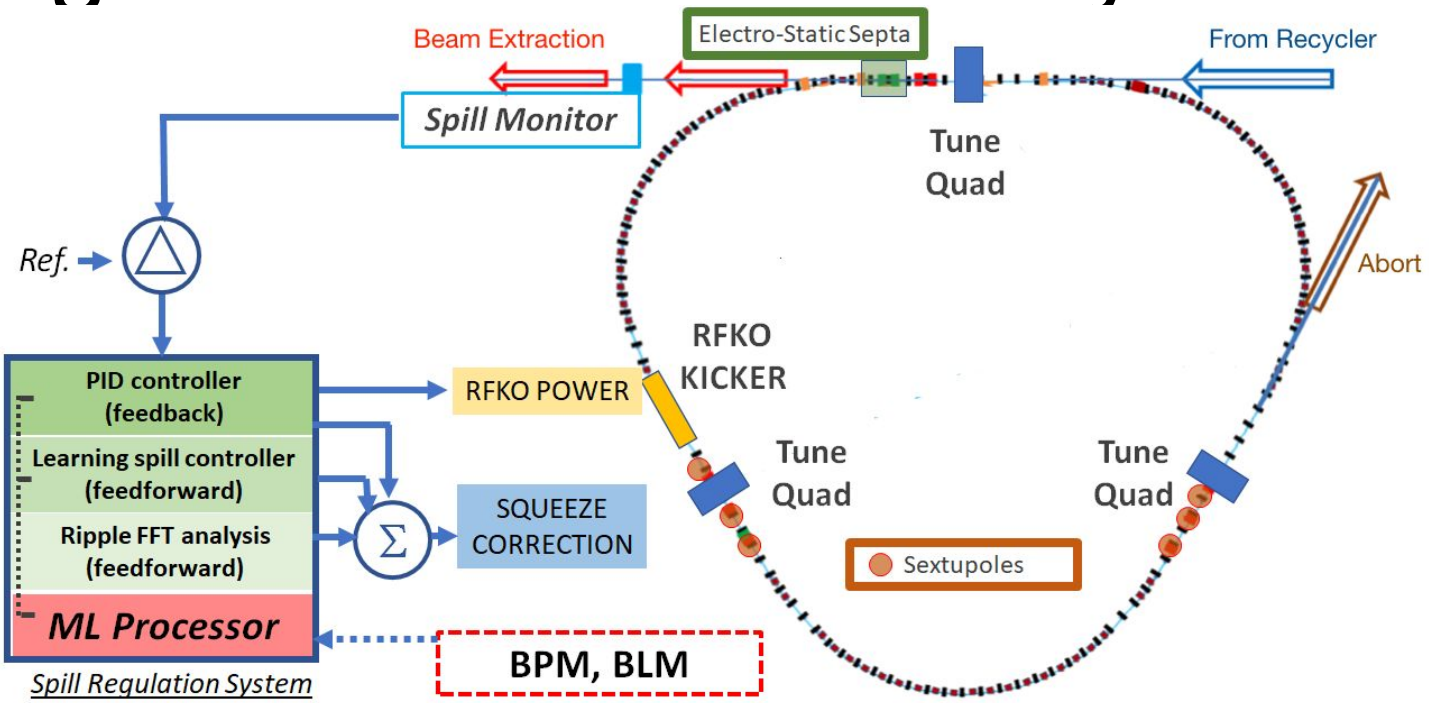
- Goal: integrate ML into accelerator operations
- Challenges: Beam Loss
  - High Energy Physics (HEP) experiment use proton beams
  - Particles get lost through interactions with the beam vacuum pipe
  - Intensity/pace of beam extraction affects radiation in the environment
  - Human operators tune parameters of hundreds to thousands of devices inside the accelerator complex
  - If beam loss (“leak”) is at extreme levels, system needs to shut-down – loss of access to facility

READS – Accelerator Real-time Edge AI for Distributed Systems



- Muon Complex
- Protons are accelerators within the Booster
- Injected into the Recycler Ring
  - The proton beam is “prepared”
- Proton beam is released into the Delivery Ring
  - Multiple experiments stationed around it
- Proton beam is extracted to match the exact intensity required for an experiments - spill

READS – Accelerator Real-time Edge AI for Distributed Systems



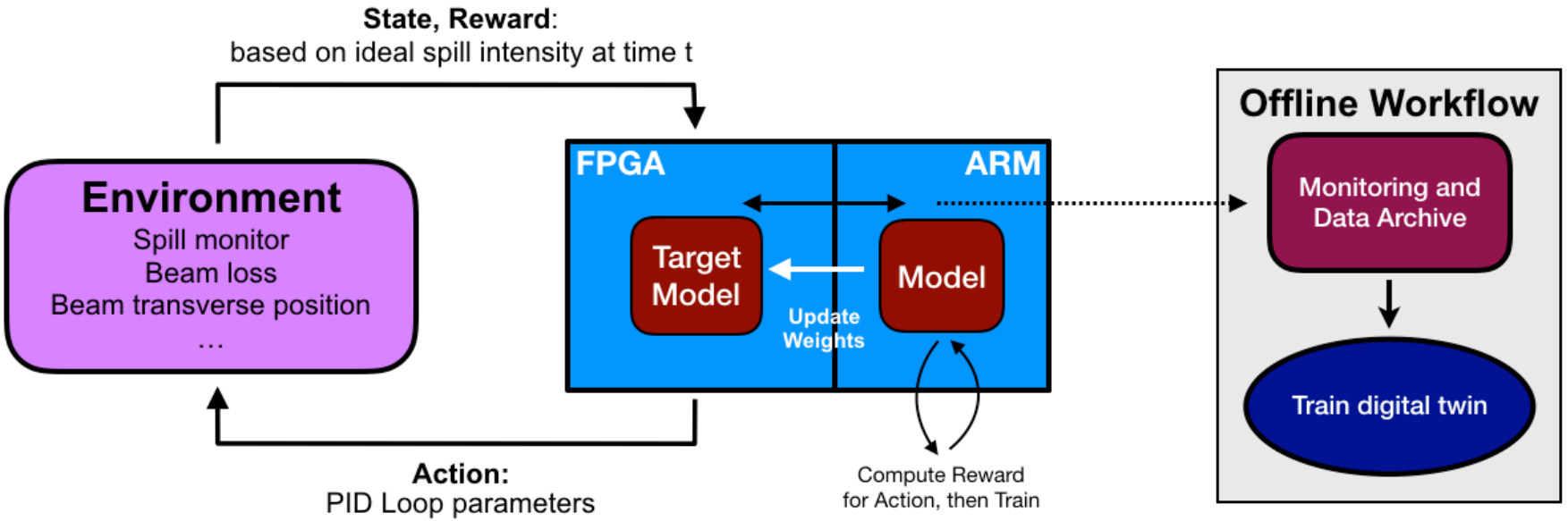
- Two main mechanisms for beam control/correction
  - Tune Quads – quadrupole magnets
  - RFKO Kickers - heaters

READS – Accelerator Real-time Edge AI for Distributed Systems

- System Design Goal: PID Controller gains need to be optimized in real-time ~ms
- The ML Processor receives inputs from sensors
  - beam position monitor (BPM)
  - beam loss monitor (BLM)

READS – Accelerator Real-time Edge AI for Distributed Systems

- System Architecture:

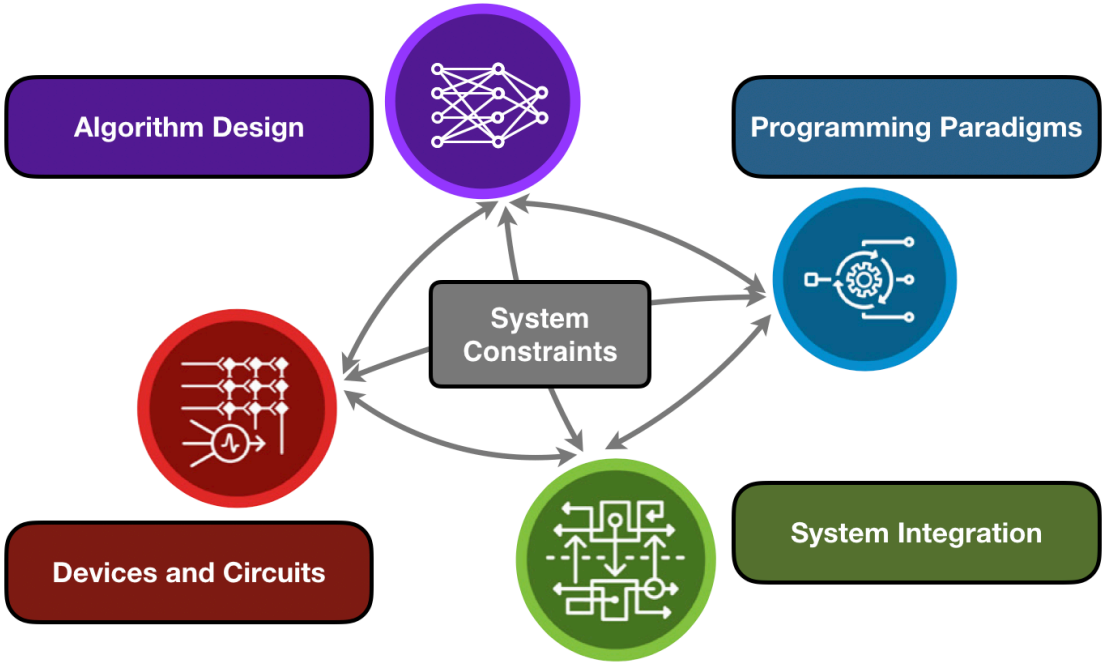


READS – Accelerator Real-time Edge AI for Distributed Systems

- System Architecture:
  - Edge device continuously optimizes the online control agent
  - Data streamed to a cloud system for large scale training

READS – Accelerator Real-time Edge AI for Distributed Systems

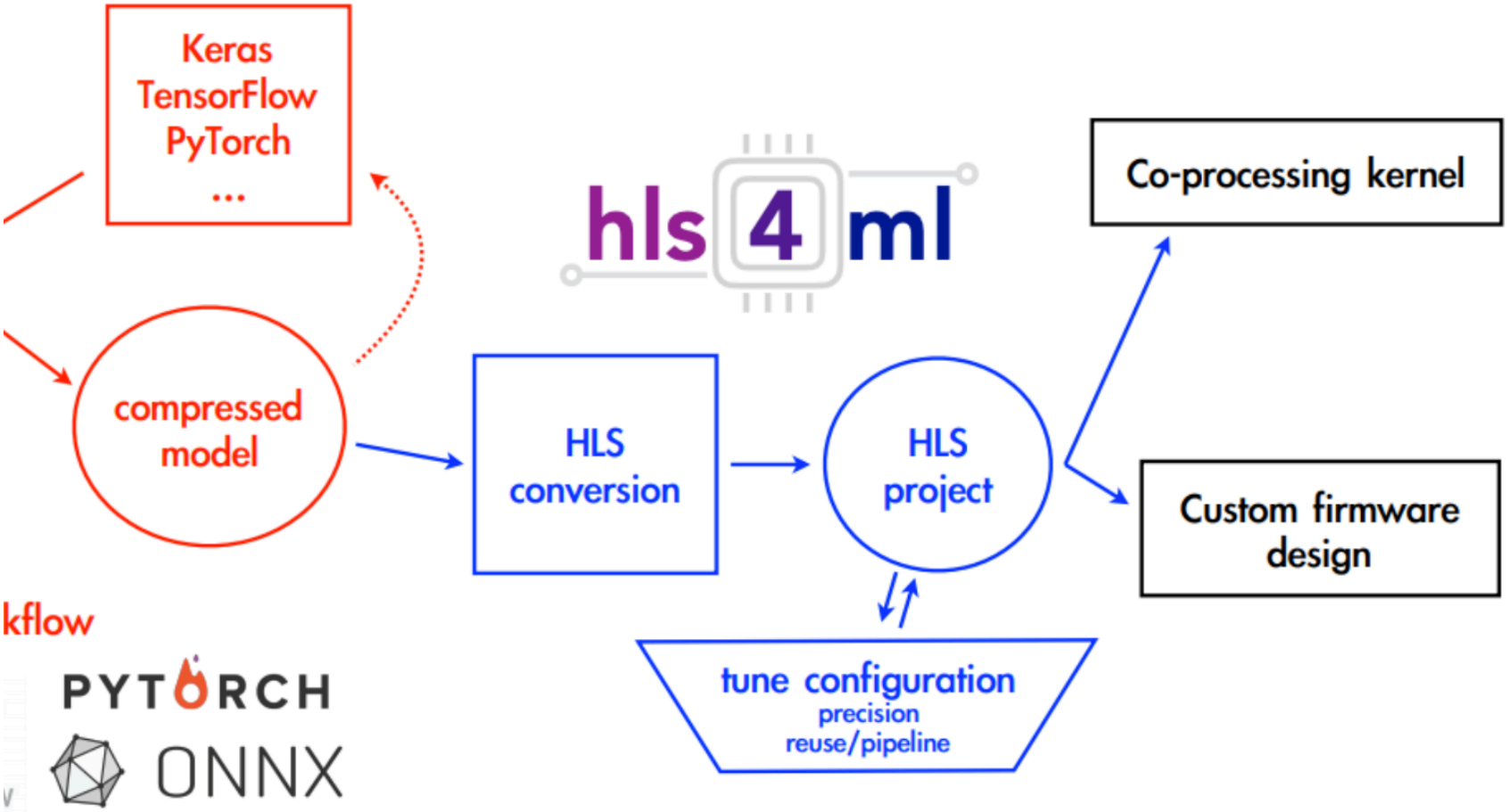
- Algorithm-Architecture Co-design
  - A common toolchain to program FPGA devices as well as create interfaces



READS – Accelerator Real-time Edge AI for Distributed Systems

- Algorithm-Architecture Co-design
  - Create modular neural network components
    - Regroup, recombine
  - Establish physics/science-aware methodology
    - Hardware-aware quantization and pruning techniques
    - Homogeneous versus heterogeneous quantization
    - Quantization-aware training
    - Control resource re-use trade-offs of high level synthesis tools
    - Differentiate between the relative hardware cost of storage, DSP, interconnect

READS – Accelerator Real-time Edge AI for Distributed Systems



hls4ml

- Aids Algorithm-Architecture Co-design
  - Translation to neural network building blocks
    - Basic layers (conv, fully connected)
    - Activation functions
  - Ongoing investigations to build a comprehensive library
    - Expansion towards graph neural networks, custom scientific kernels
    - Custom neurons (long-short-term memory, multi-head attention, etc.)

hls4ml

- Aids Algorithm-Architecture Co-design
  - Ongoing work to automate compatibility with several backend flows
    - Xilinx Vivado, Intel Quartus HLS, Mentor Graphics Catapult
    - Integration of model development (training, model compression) and hardware synthesis
    - Automation of insertion for optimization directives and pragmas

hls4ml

- Democratize design tools
- Allow domain scientists to build edge computing systems with minimal specialized hardware design training
- Open source
- HLS raises level of abstraction
  - Faster simulation, faster evaluation of design alternatives
  - Allow domain expert trade-off parallelism, resource re-use, latency constraint, power budget

Reconfigurable autoencoder for detector front-end ASICs

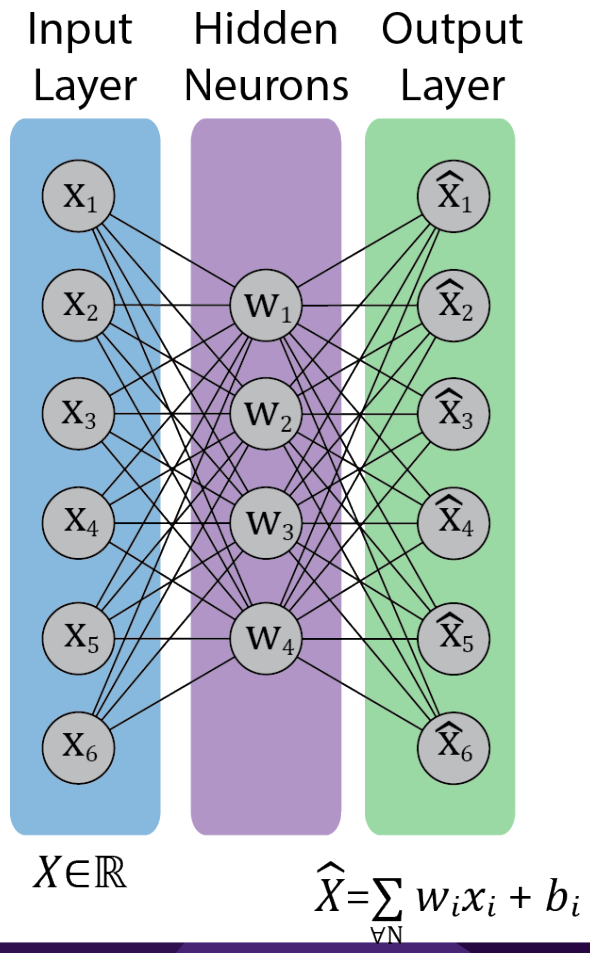
- Edge computing for particle collider experiments
- Data collected from a large number of photon detectors are compressed to representative information of the “shape”
  - Charge measurements from the detectors are compressed to a radiation pattern
  - 6 million detector channels sending data at 40MHz
  - The data from the original space is compressed to lower dimensionality by the edge ASIC, transmitted, then decoded on the receiving end

Reconfigurable autoencoder for detector front-end ASICs

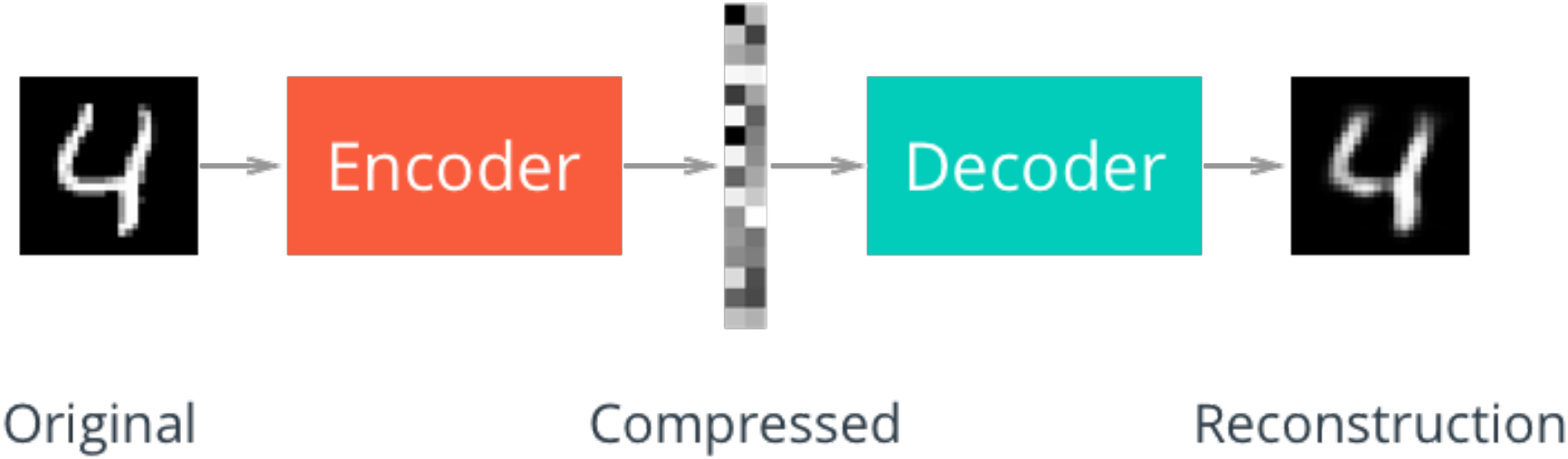
- System constraints
  - Low power
    - Will be part of a larger system with power budget
  - Radiation tolerant
  - Reprogrammable weights through accessible registers to enable updates and customization

Reconfigurable autoencoder for detector front-end ASICs

- Autoencoder: neural network with single convolutional layer followed by a dense layer



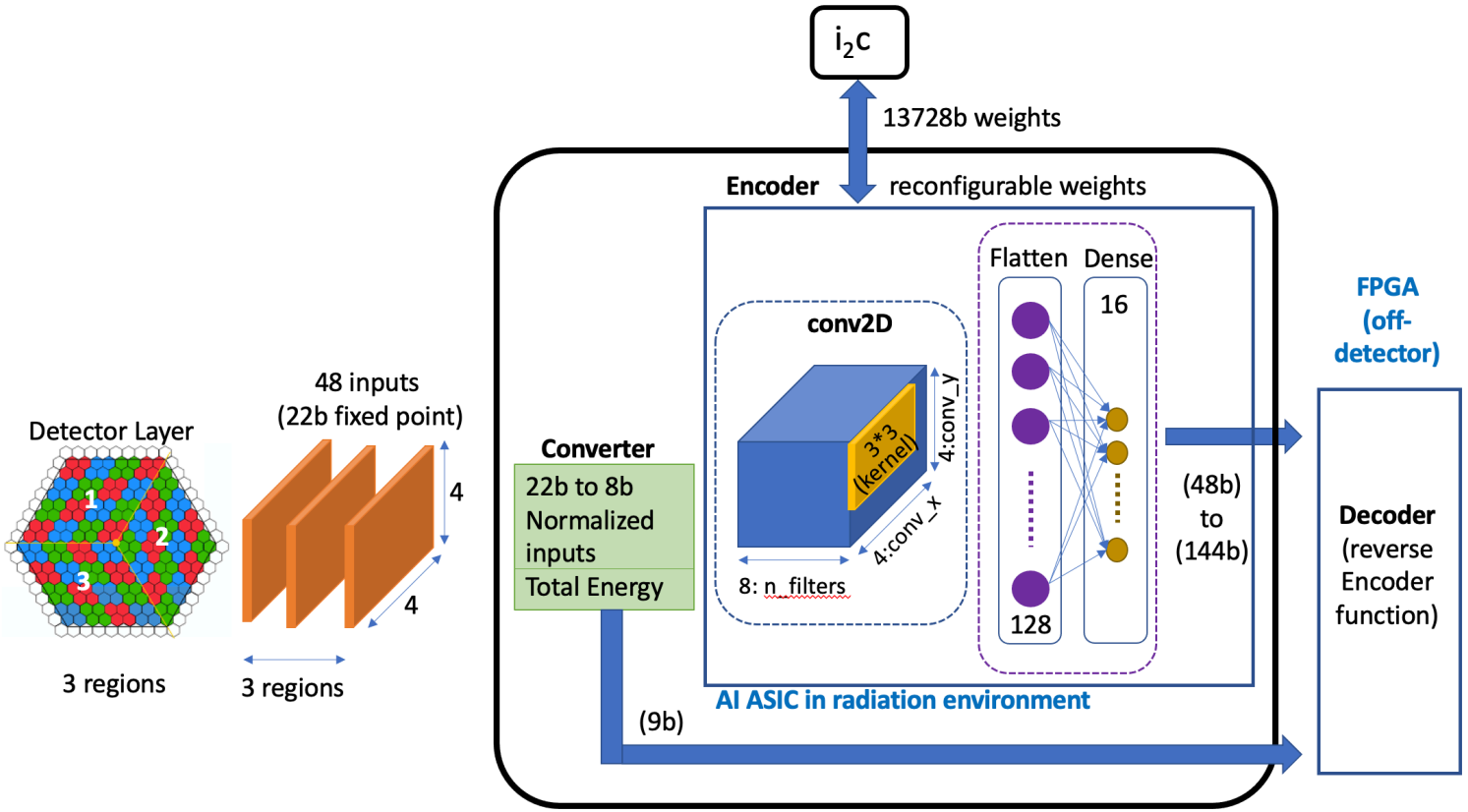
Reconfigurable autoencoder for detector front-end ASICs



Glossary of Deep Learning: Autoencoder, by Jaron Collins

Reconfigurable autoencoder for detector front-end ASICs

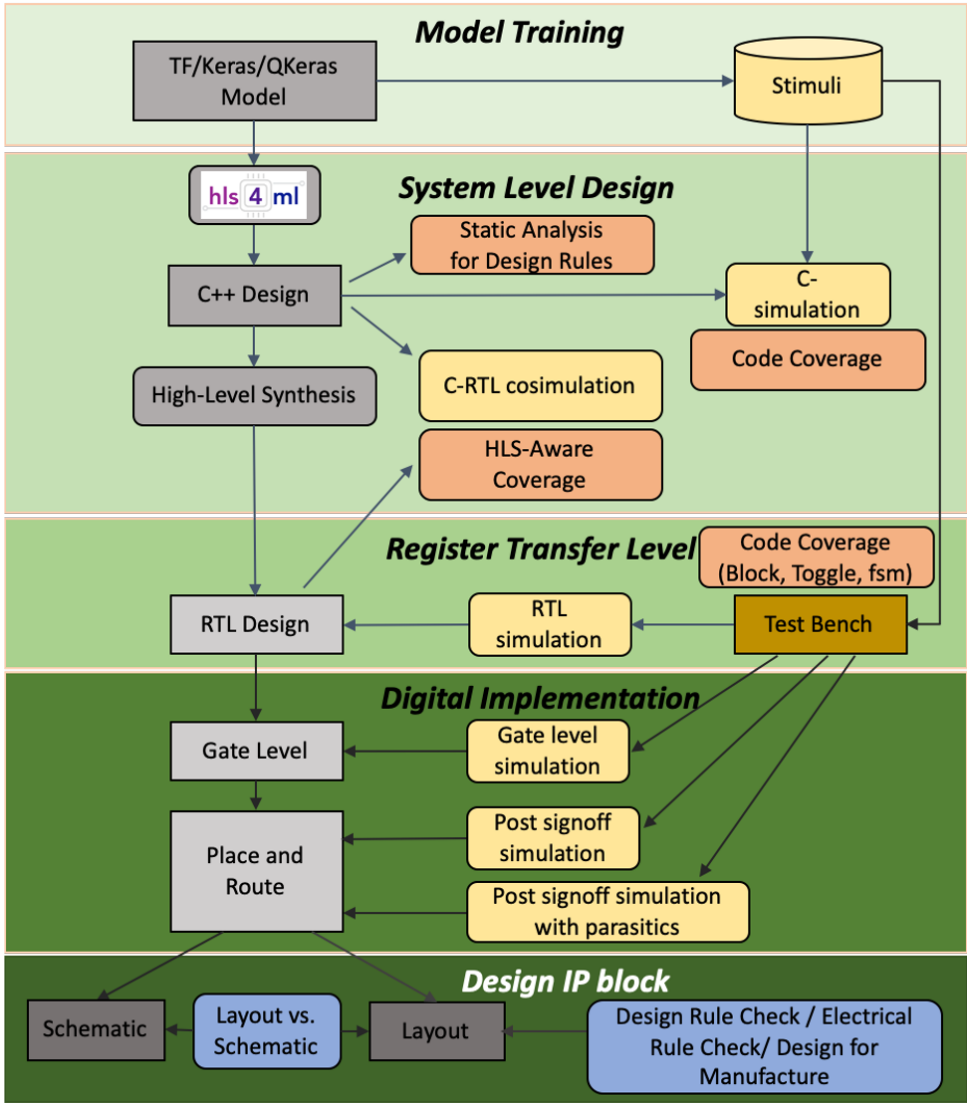
- Dataflow
  - 22-bit signals from 48 detector cells



Reconfigurable autoencoder for detector front-end ASICs

- Network properties
  - CNN layer: eight 3x3x3 kernel matrices resulting in 128 outputs
  - ReLu activation after CNN and after final dense layer
  - 6-bits weights
  - Dense layer produces 16 10-bit outputs
  - Chip can be reconfigured to produce as low as total of 64 bits in output

Reconfigurable autoencoder for detector front-end ASICs



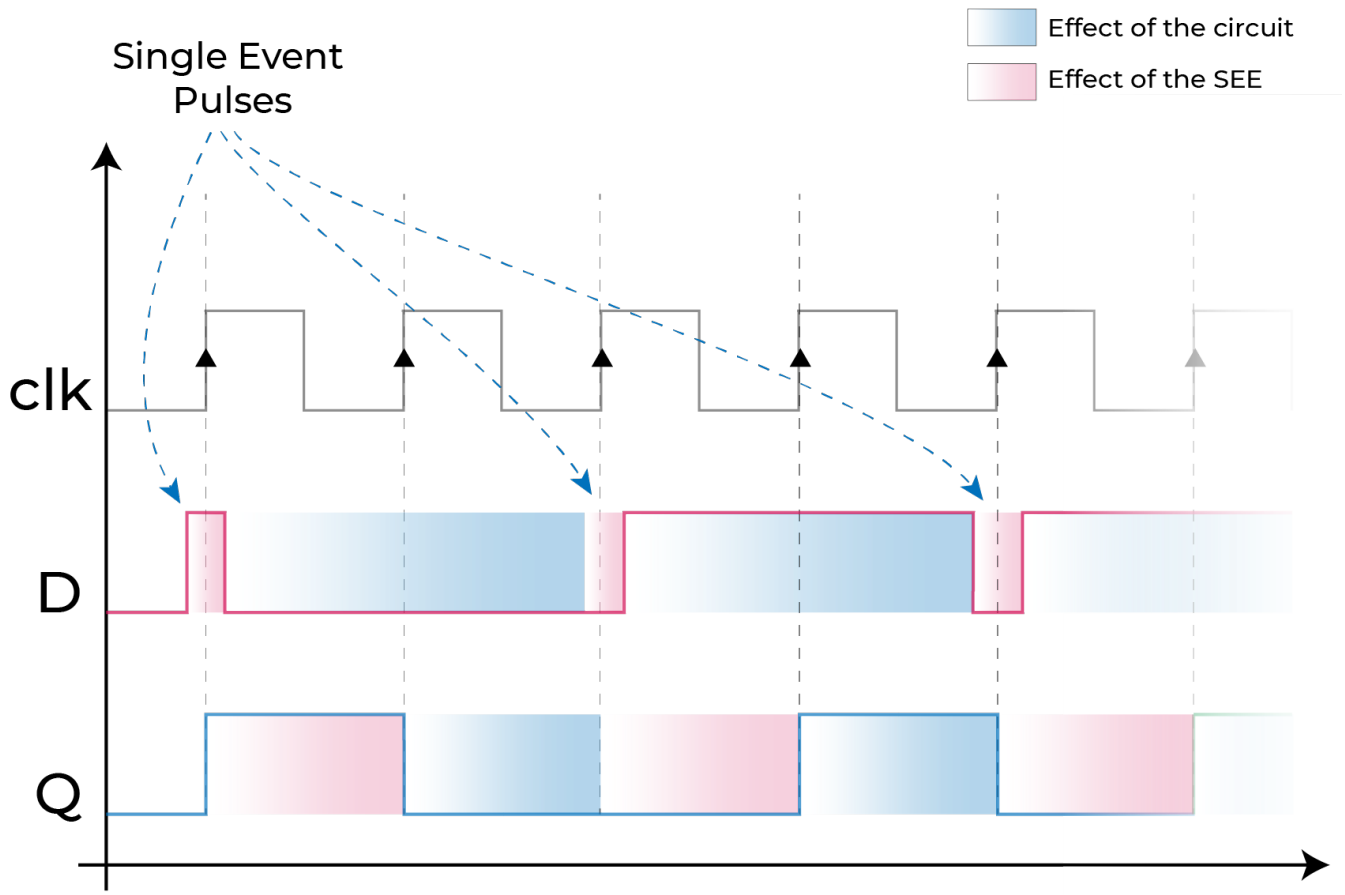
Reconfigurable autoencoder for detector

DESIGN (D) AND VERIFICATION (V) METRICS

STEP	TIME	ITERATIONS	SIZE
Model generation (D) C simulation (V)	0.98s 0.14s	50-100	1089 C++ LoC
High-level synthesis (D) RTL simulation (V)	00:30:17 00:00:46	2-3	39,716 Verilog LoC
Logic synthesis (D) Gate-level simulation (V)	06:04:19 00:25:19	6	900,810 Gates
Place and route (D) Post-layout simulation (V) Post-layout parasitic simulation (V)	71:03:53 00:51:41 01:51:30		1,026,387 Gates
Layout (D) LVS & DRC (V)	00:20:00 01:00:00	1	12,768,389 Transistors

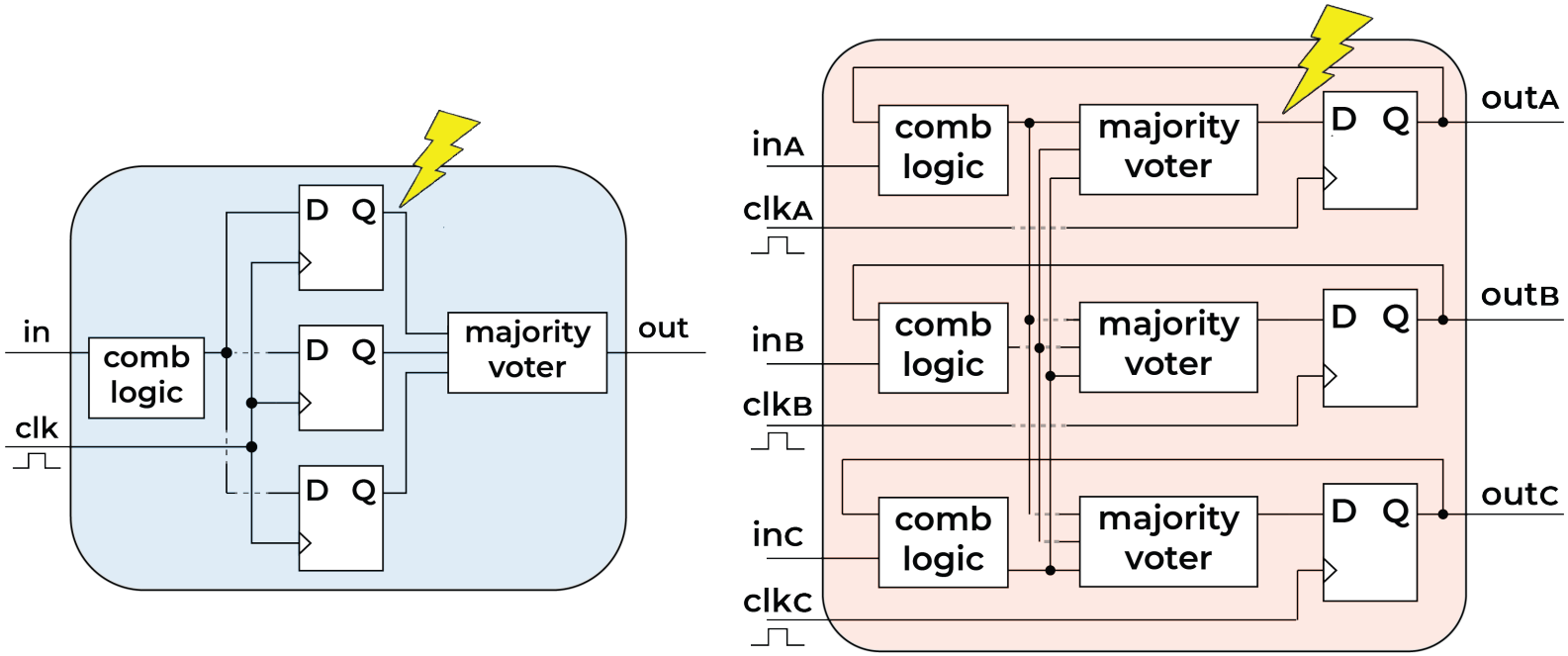
Reconfigurable autoencoder for detector front-end ASICs

- Protection against Single Event Upsets



Reconfigurable autoencoder for detector front-end ASICs

- Protection against Single Event Upsets



Reconfigurable autoencoder for detector front-end ASICs

- Chip specs
  - 6b weight and bias parameters
    - Total: 2,286 = 13,724 bits
  - Parameters loaded via I2C interface
  - Decoder component implemented off-detector on FPGA
  - Chip latency – 25ns
  - 7nJ per inference
  - 280mW
  - Can withstand 200MRad ionizing radiation
  - 2.5mm^2

Projects – Preliminary

- CryoAI - 22nm testchip
- Adaptive ML accelerators

CryoAI - 22nm testchip

- A system-on-chip (SoC)
- Goal: test performance and leakage power trends of the technology at cryogenic temperatures
- An opportunity to showcase the hls4ml flow
  - Contribute a neural network accelerator module as part of the SoC
  - Perform anomaly detection
- Design considerations
  - Programmability -- Cannot reload all of the parameters
  - Optimal sequencing of quantization & pruning
  - Assignment of sub-regions of the network to voltage domains

Adaptive ML accelerators

- Why adaptation?
  - Energy constraints
  - Input variability
  - Translation to new platform/device
  - Transfer Learning

Adaptive ML accelerators

- Some directions
  - Emulate loss through drop out/connect
    - Techniques in ML literature equate these phenomena to forcing weights to zero
    - Loss in (because of) hardware may look very different
  - Continuous diagnostics
    - Evaluation of network certainty
    - Concept of surprise
  - Detection of temporal dominance of classes
    - Reconfigure optimized version for dominant class

Adaptive ML accelerators

- Some directions
  - Critical path based hardware-aware resource management
    - **Class-based CP:** using contributions of a neuron to a specific class
      - Mean Absolute Activation, first order Taylor Approximations
    - **Generalized CP:** relative participation of output channels at the routing of the output from a layer
  - Distribute resources (e.g., total number of bits allocated for weights) according to a criticality metric

Adaptive ML accelerators

- Some directions
  - Characterize circuits (e.g., SRAM) to create models
    - Associate voltage drop/power outage with cell decay
    - Create characteristic bit masks for weights
  - Neural network architecture search
    - Lessons learned from design space exploration in high-level synthesis
    - Search for a new cell from basic building blocks
      - Input: a set of convolutions and pooling of varying size
      - Think of it as your module library

Future Directions

- Fluid implementations
  - Quickly re-targetable from software to hardware
- Scientific domains offer a vast space of computational challenges
  - They need interpretable systems
  - Laws of nature apparent in the system's output
- Multiple paths need to converge
  - Photonic circuits – not much automated
  - FPGA
  - Neuromorphic – not much automated
  - Quantum - ??

Collaborators

- Northwestern University
  - Han Liu (CS), Kristian Hahn (Physics)
  - Manuel Blanco Valentin, Rui Shi, Bincong Ye, Yingyi Luo (now at Google), Sid Joshi (now at Intel)
- Fermi National Laboratories
  - Nhan Tran, Farah Fahim, Christian Herwig, Kiyomi Seiya, et al.
- Columbia University
  - Giuseppe di Guglielmo
- Lehigh University
  - Josh Agar

THANK YOU!