Design of a reconfigurable autoencoder algorithm for detector front-end ASICs

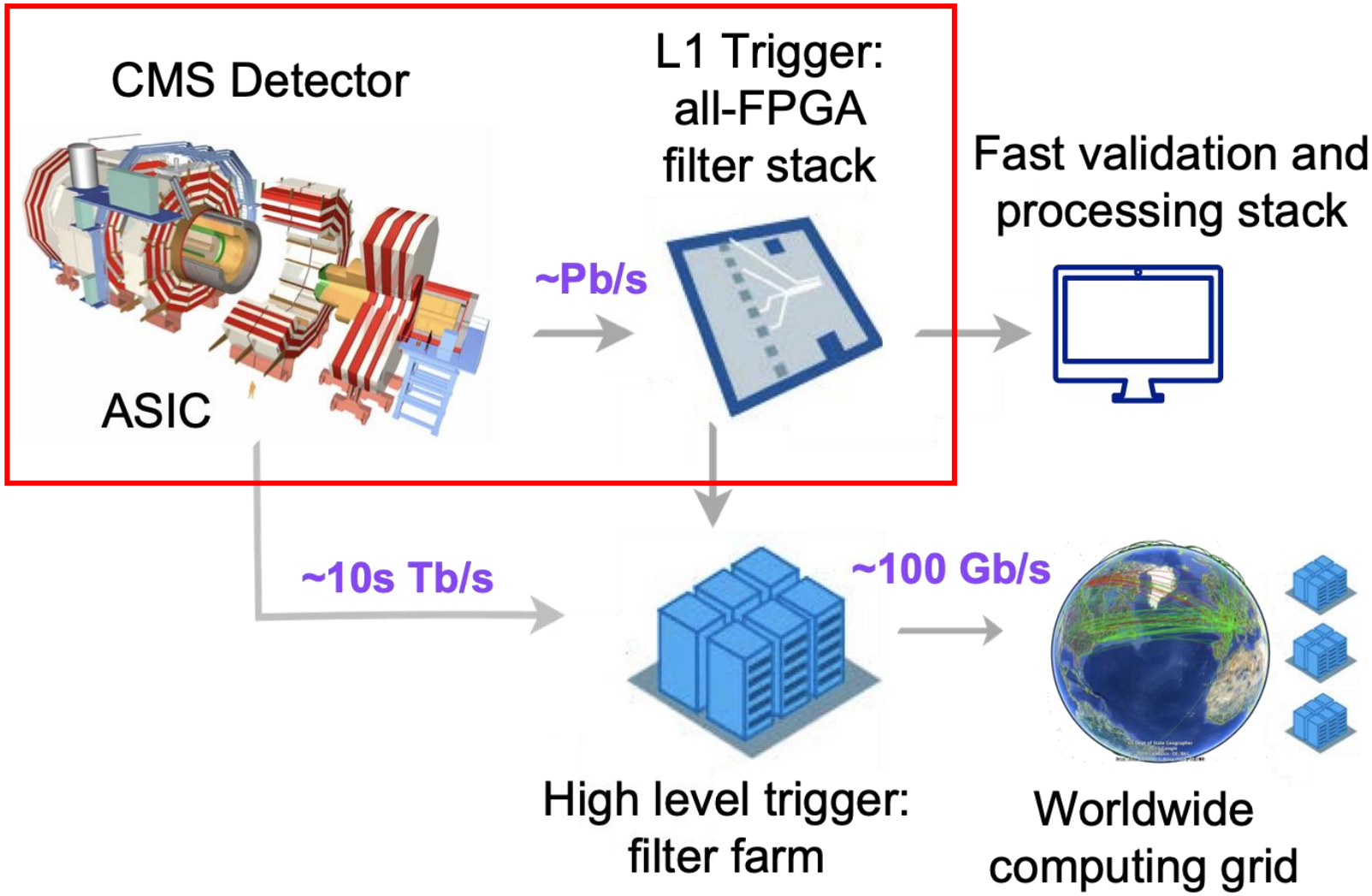
Fast Machine Learning for Science – November 30, 2020

Brown University: Ka Hei Martin Kwok
Columbia University: **Giuseppe Di Guglielmo**, Luca Carloni
Fermilab: Farah Fahim, Benjamin Hawks, Christian Herwig, Jim Hirschauer, Nhan Tran
Florida Institute of Technology: Daniel Noonan
Northwestern University: Manuel Blanco Valentin, Yingyi Luo, Seda Memik

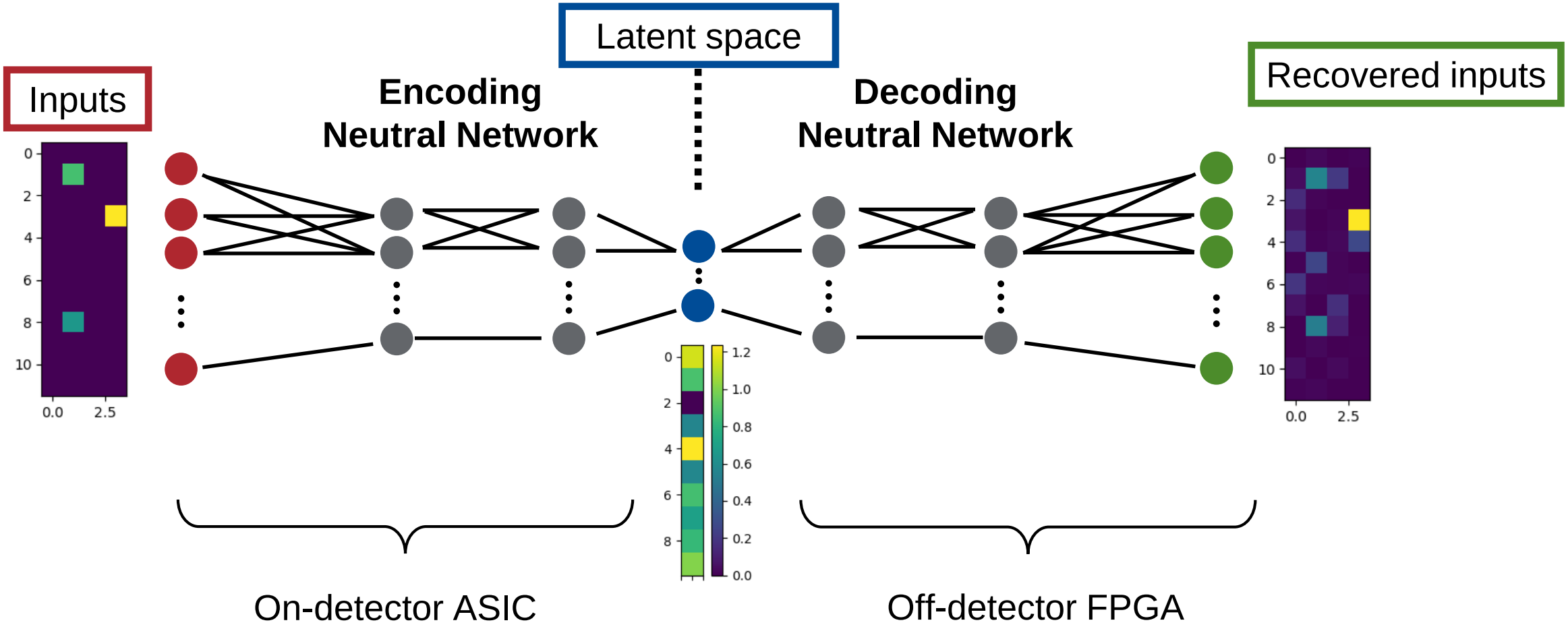


Major challenges

- Bandwidth
- Low latency
- Low power
- High-radiation

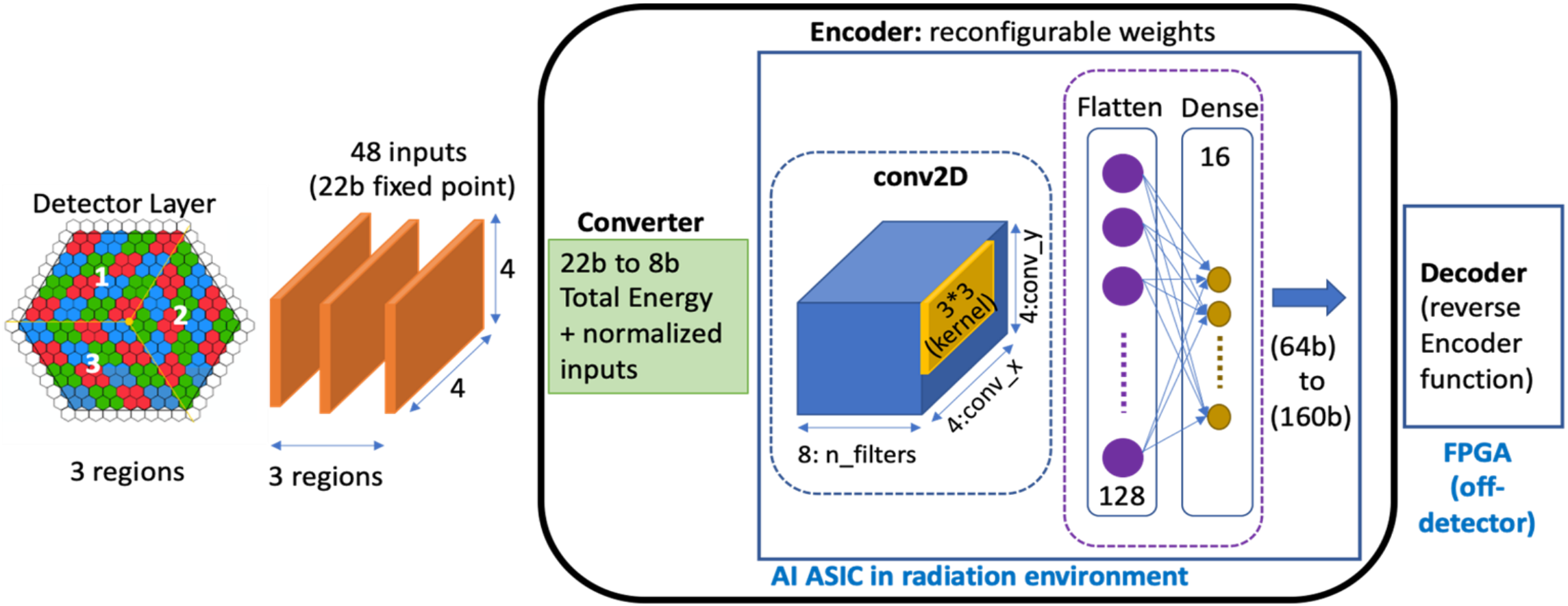


Autoencoder for data compression



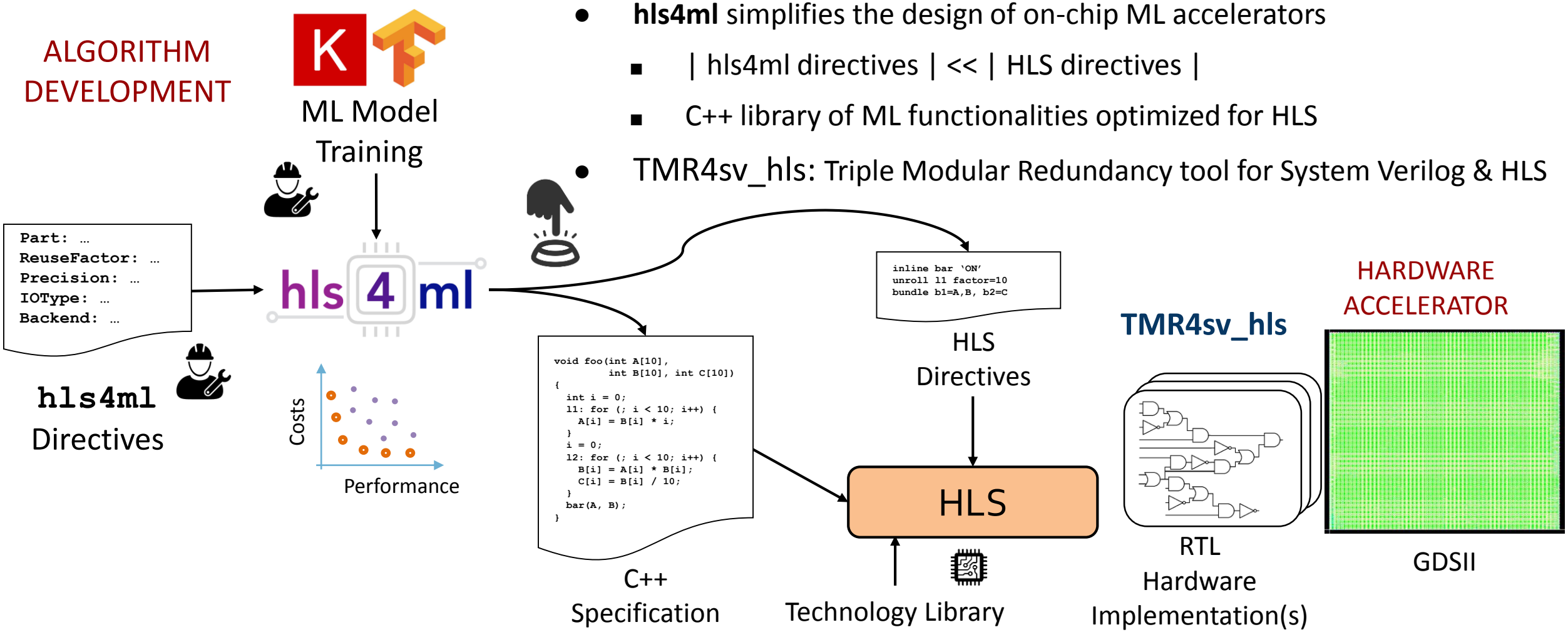
Network weights are fully reconfigurable!

Encoder architecture



Physics-driven hardware co-design

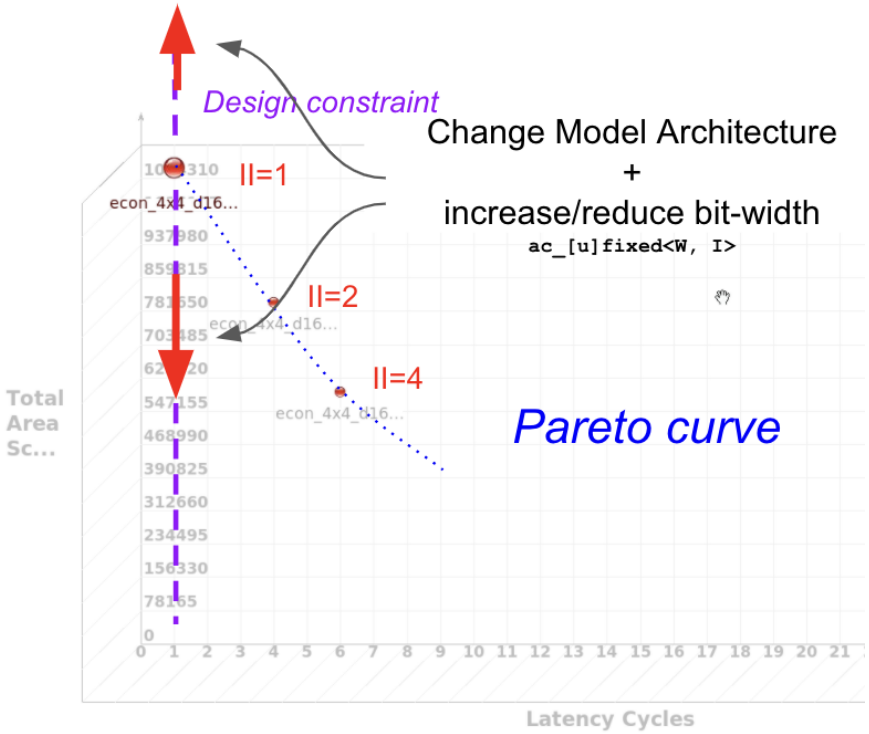
- Algorithm development based on Physics data
- **hls4ml** simplifies the design of on-chip ML accelerators
  - | hls4ml directives | << | HLS directives |
  - C++ library of ML functionalities optimized for HLS
- TMR4sv_hls: Triple Modular Redundancy tool for System Verilog & HLS

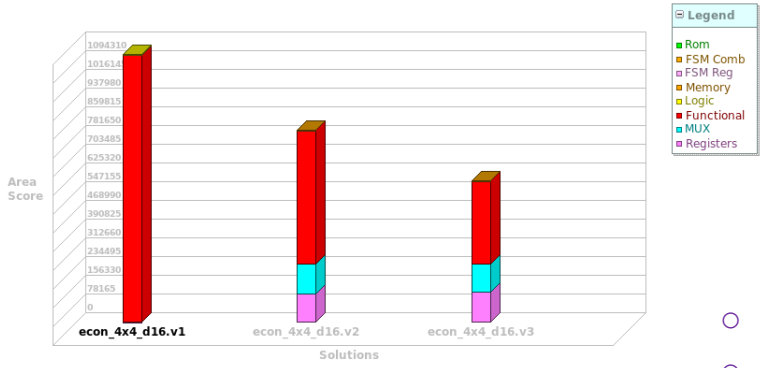


HLS: Design space exploration

Report: General

Solution	Latency Cycles	Latency Time	Throughput Cycles	Throughput Time	Slack	Total Area
solution.v1 (new)						
econ_4x4_d16.v1 (extract)	1	25.00	1	25.00	13.61	1116589.46
econ_4x4_d16.v2 (extract)	4	100.00	2	50.00	7.54	802319.52
econ_4x4_d16.v3 (extract)	6	150.00	4	100.00	0.47	591675.33

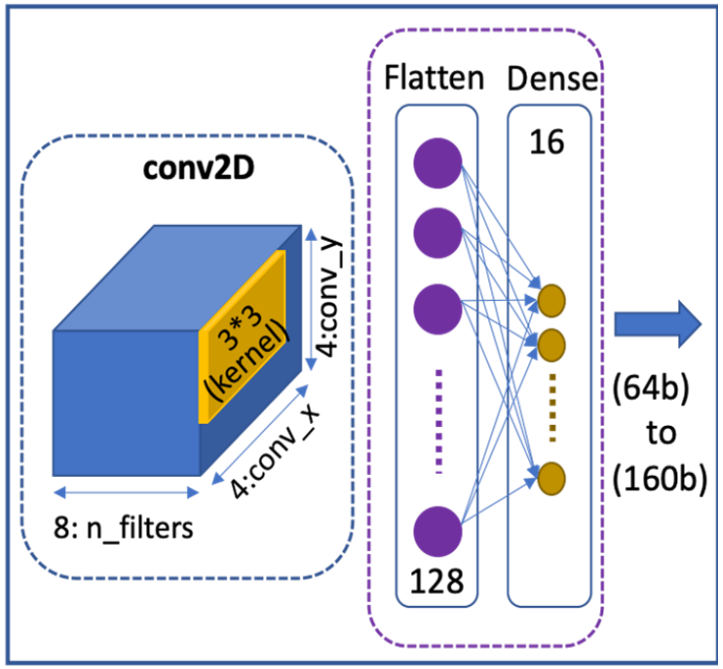


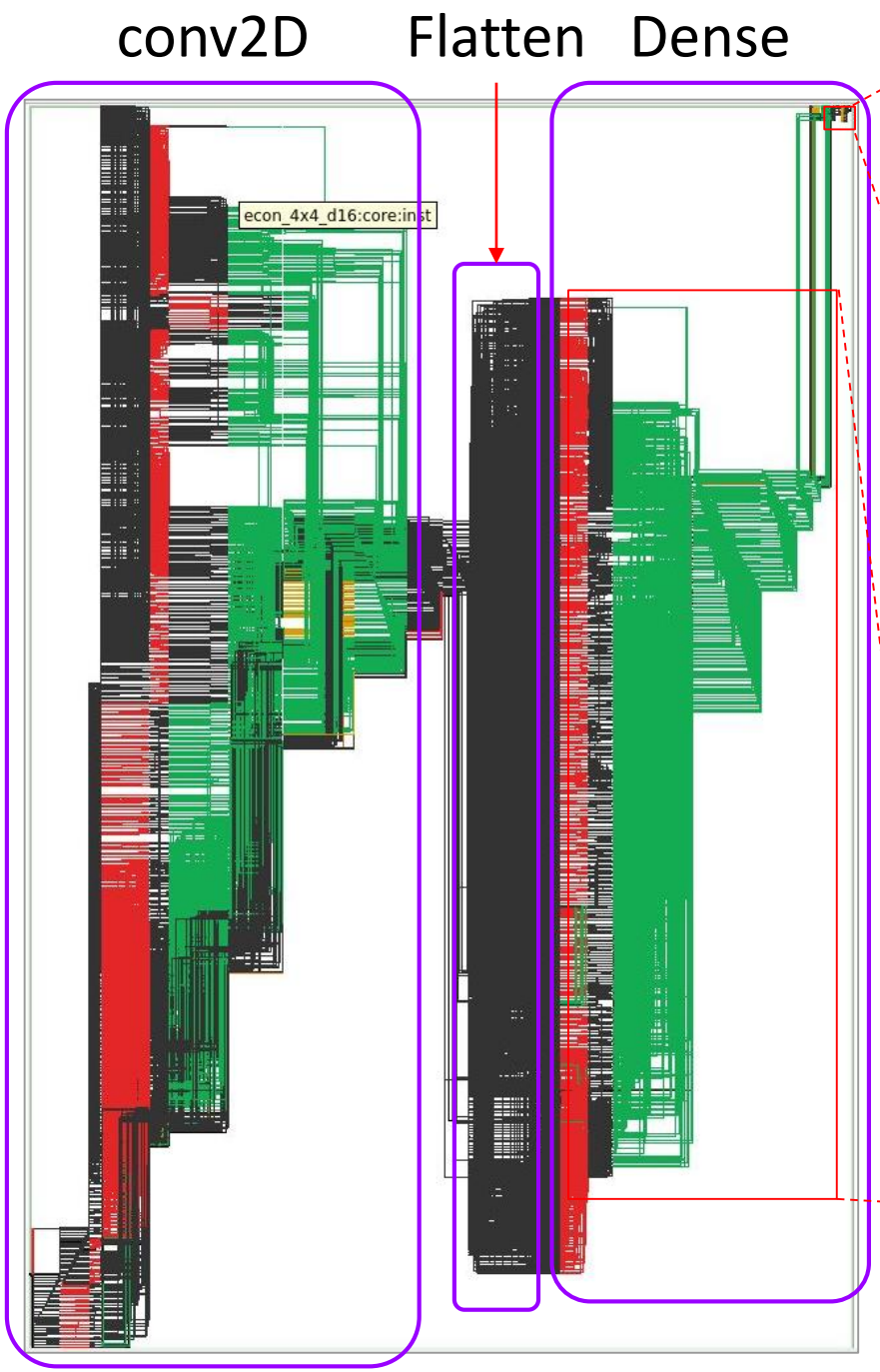


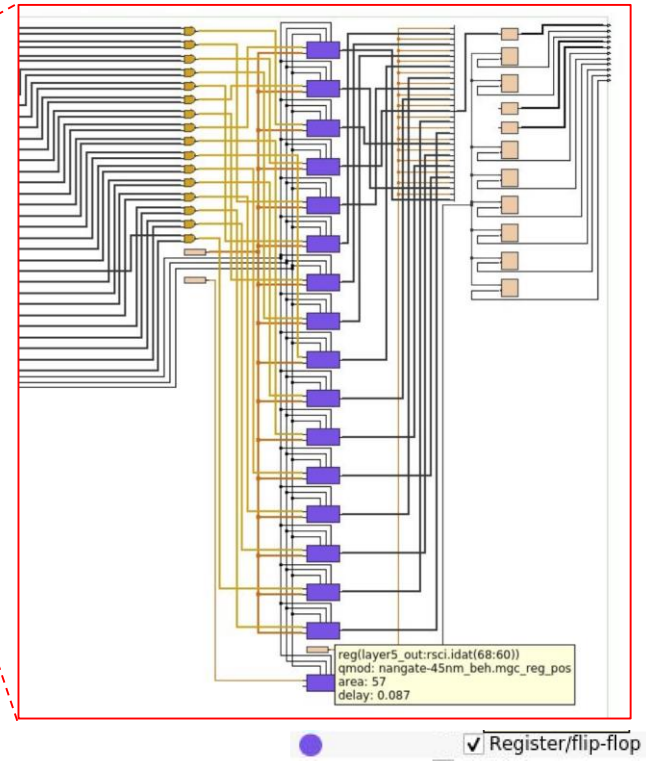
- Initiation interval = 1
- Clock period = 25 ns
- I/O fixed-point precision
  - Inputs : 8b
  - Weights : 6b
  - 16 Outputs : 9b
  - Programmable to 3b, 5b or 7b
- **No pipeline, unroll all loops**
- **No SRAMs, only registers**
- **Map all arrays to registers**
- **Inputs are wires, Outputs are registered**

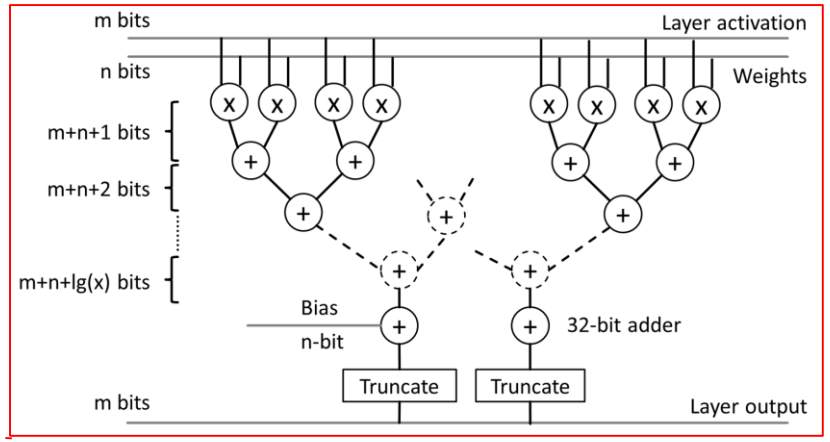
HLS: Encoder RTL schematic

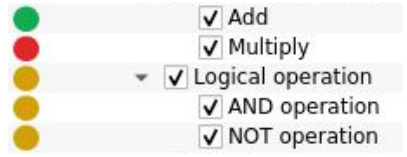
Solution: Conv + Flatten + Dense







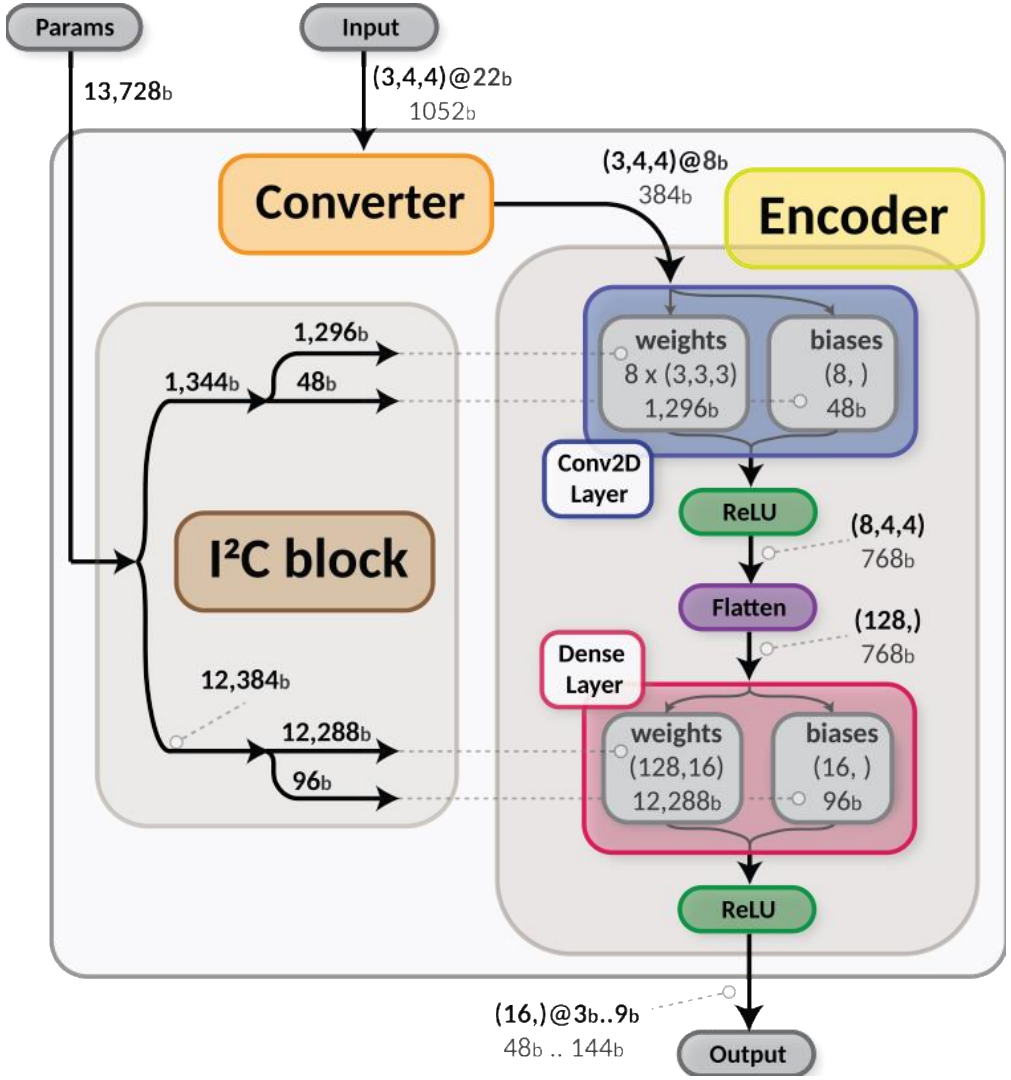




225,000 multiply and accumulate every 25 ns

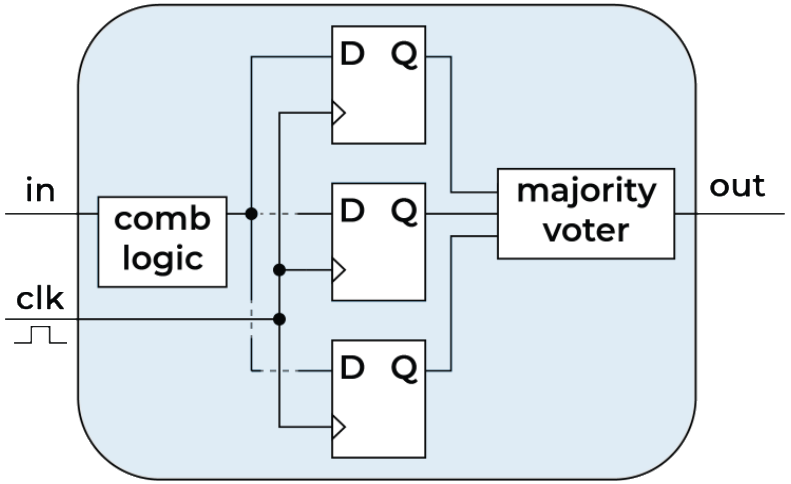
Combining RTL from various sources

- Encoder
  - ML model converted with hls4ml
  - HLS-generated Verilog RTL
- Converter
  - C++, manually written
  - HLS-generated Verilog RTL
- I2C Peripheral
  - System Verilog RTL, manually written



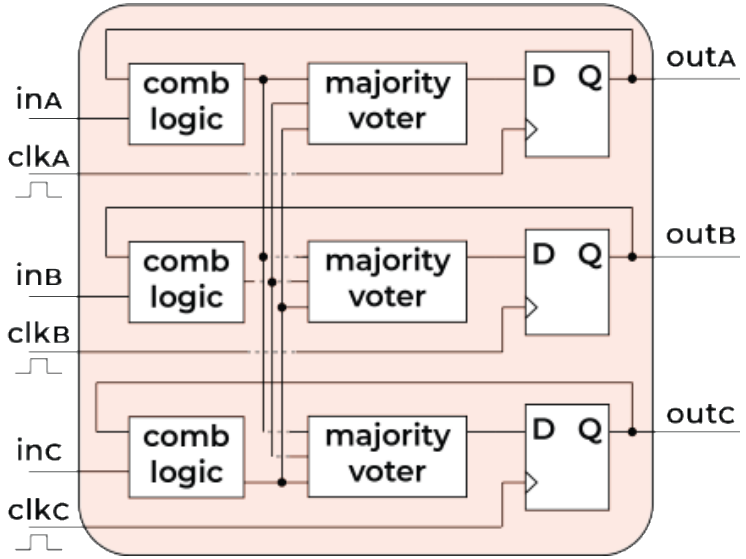
Single-Event Effect Mitigation: Triple modular redundancy strategy

Encoder & Converter



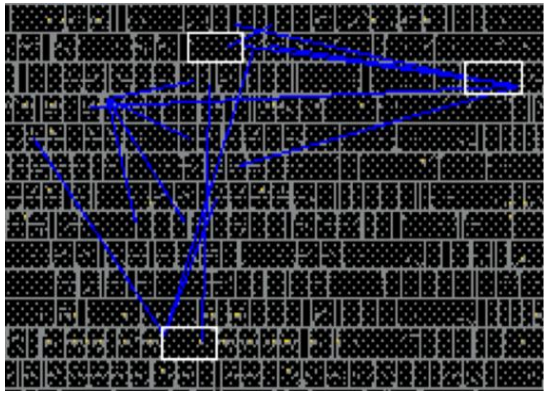
- Data path - new data every 25 ns
- **Triplicated registers only**
- No auto-correction or feedback
- (0.2% of design = 546 registers for data storage)
- No state machines: parallel architecture

I2C Peripheral



- Weights storage: Auto-correction and feedback
- **Full module triplication**
- 75% design is registers: which need to be triplicated
- Doesn't require additional Error Correction code
- I^2C **-** RW: Bidirectional - can be readout to check weights

Spacing: at least 15 µm apart



Conclusions

- We proposed a design methodology that spans from the ML model generation to the ASIC IP block creation
- We implemented ML compressions for detectors in low power, low latency, high radiation environment

Rate	**II**	**Latency**	**Energy/inference**	**Power**
40 MHz	1	50 ns	2.38 nJ/inf.	95 mW

Area	**Gates**	**Tech. Node**
3.6 mm^2	800K	TSMC 65nm LP CMOS

Radiation tolerance
Up to 200 MRad

Acknowledgments

- Thanks to the Fermilab ASIC group, CMS HGCal and Fast Machine Learning communities
- Thanks for the CAD support
  - Sandeep Garg and Anoop Saha (Mentor/Siemens Catapult HLS)
  - Bruce Cauble and Brent Carlson (Cadence Innovus and Incisive)