AI on chip – Algorithm to Accelerator

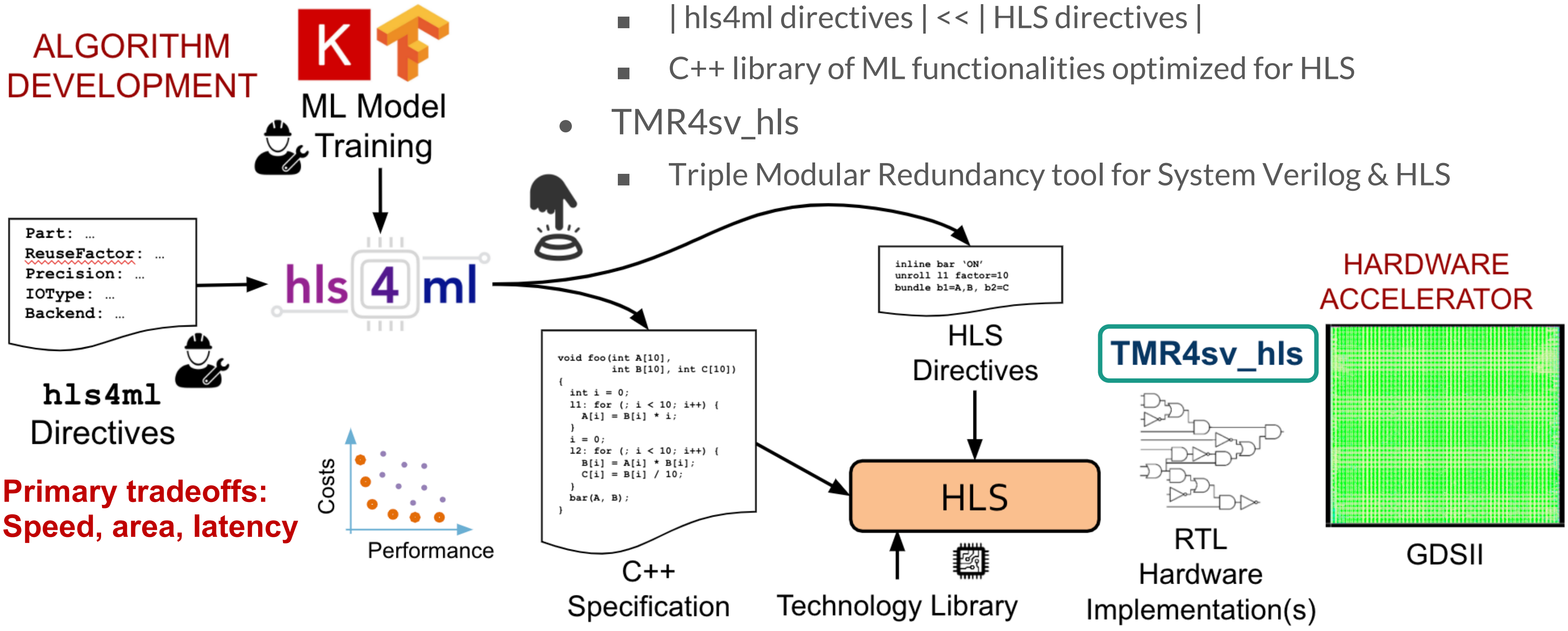
Farah Fahim - Fermilab

20200730

Contributors
Fermilab: Nhan Tran, Christian Herwig, Jim Hirschauer
Northwestern University: Seda Memik, Yingyi Luo, Manuel Valentin
Columbia University: Giuseppe DiGuglielmo

New Paradigm for ASIC Development: 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



```
Part: ...
ReuseFactor: ...
Precision: ...
IOType: ...
Backend: ...
```

hls4ml Directives

```
inline bar 'ON'
unroll l1 factor=10
bundle b1=A,B, b2=C
```

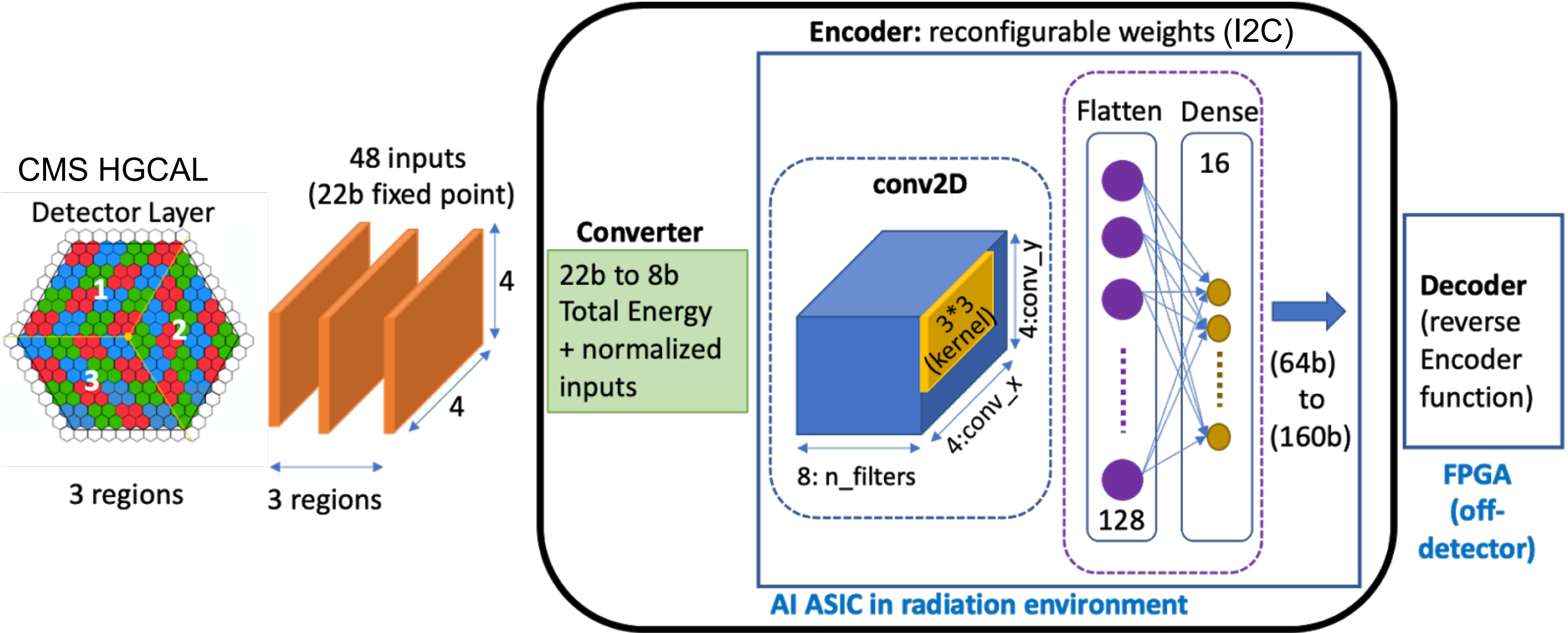
```
void foo(int A[10],
         int B[10], int C[10])
{
  int i = 0;
  l1: for (; i < 10; i++) {
    A[i] = B[i] * i;
  }
  i = 0;
  l2: for (; i < 10; i++) {
    B[i] = A[i] * B[i];
    C[i] = B[i] / 10;
  }
  bar(A, B);
}
```

Primary tradeoffs: Speed, area, latency

Autoencoder: Reconfigurable data compression

- Enable edge compute : Data compression of detector output using deep neural networks
- Programmable and Reconfigurable: ability to update weights based on **real-time feedback** (ms)
- Training **adaptable** to changing detector conditions (pileup), different geometries, lost channels, etc.

- Unsupervised learning (2.375nJ/inference, every 25ns, 15x compression, < 4mm^2)



CMS HGCAL

Converter

22b to 8b Total Energy + normalized inputs

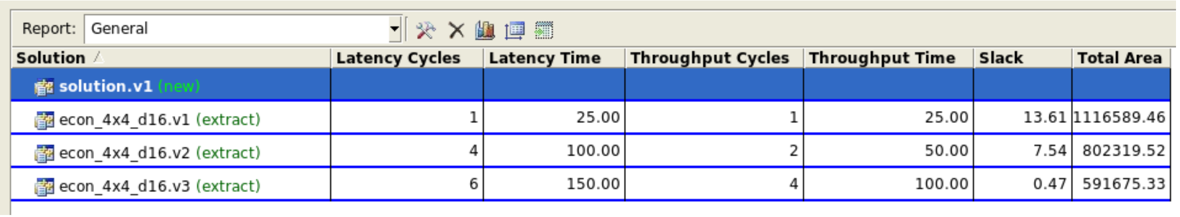
Encoder: reconfigurable weights (I2C)

Decoder (reverse Encoder function)

FPGA (off-detector)

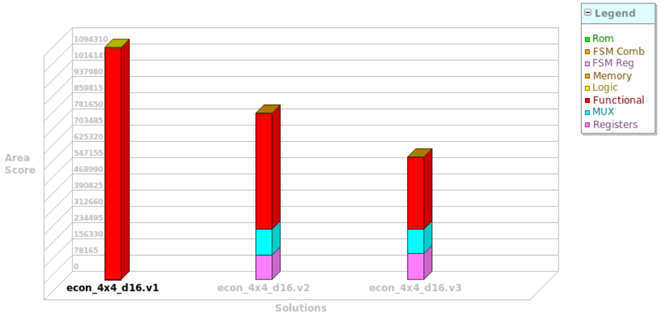
AI ASIC in radiation environment

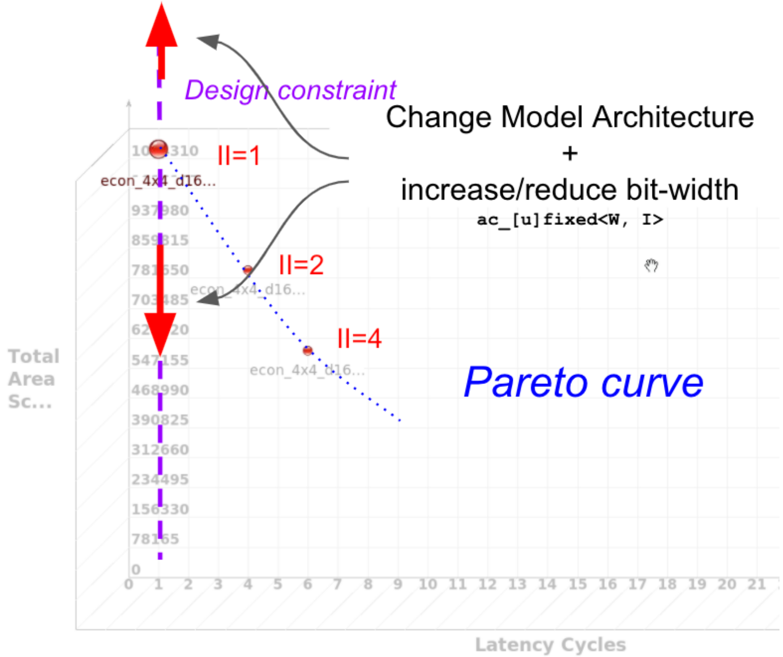
Optimization / Design Space exploration



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

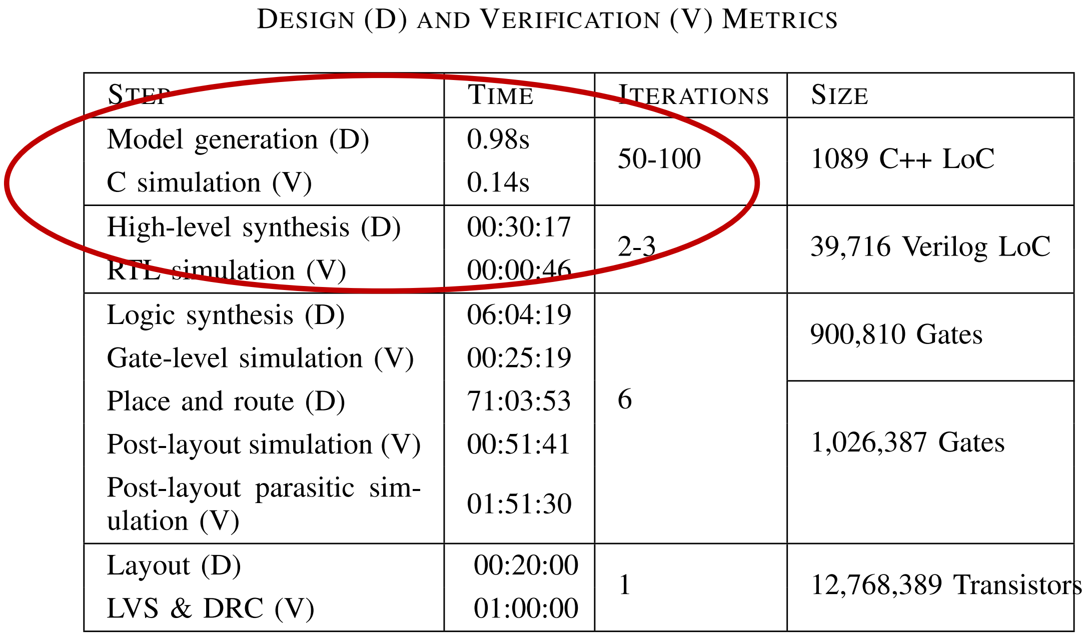




Design constraint

Change Model Architecture
+
increase/reduce bit-width
`ac_[u]fixed<W, I>`

Pareto curve



DESIGN (D) AND VERIFICATION (V) METRICS

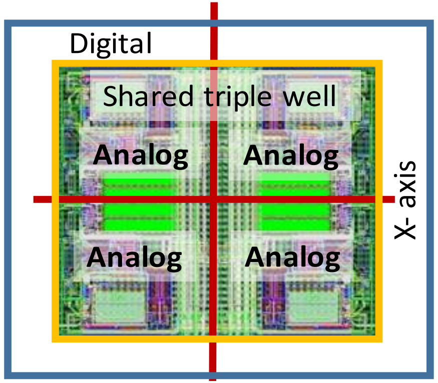
STEP	TIME	ITERATIONS	SIZE
Model generation (D) C simulation (V)	0.98s 0.14s	50-100	1089 C++ LoC
High-level synthesis (D) RTL simulation (V)	00:30:17 00:00:46	2-3	39,716 Verilog LoC
Logic synthesis (D) Gate-level simulation (V)	06:04:19 00:25:19	6	900,810 Gates
Place and route (D) Post-layout simulation (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

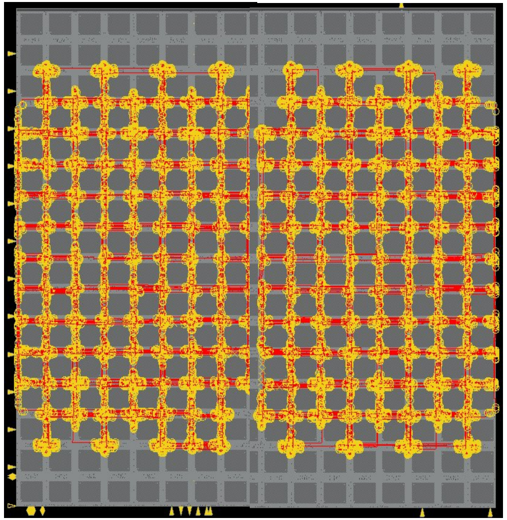
Synergistic Applications

AI on edge to AI in pixel

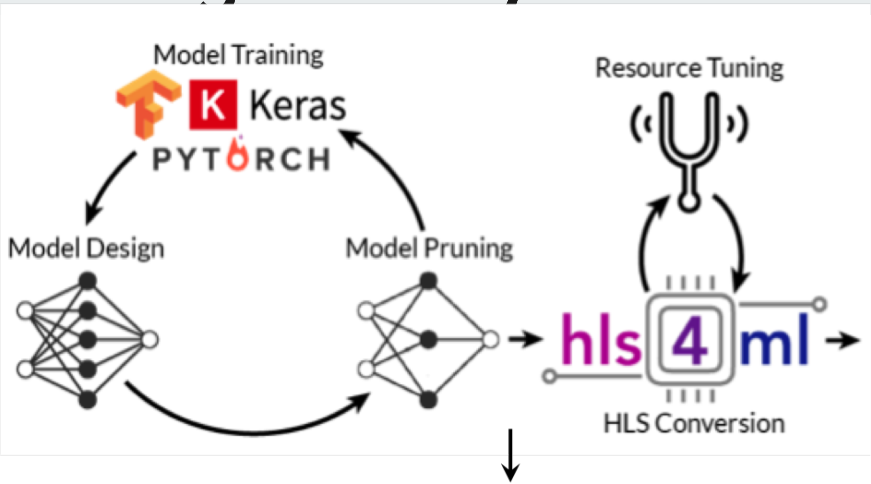
- Power consumption ~ 1pJ/bit - center to periphery (~ 5mm): routing capacitance
  - Why not do local calculations???

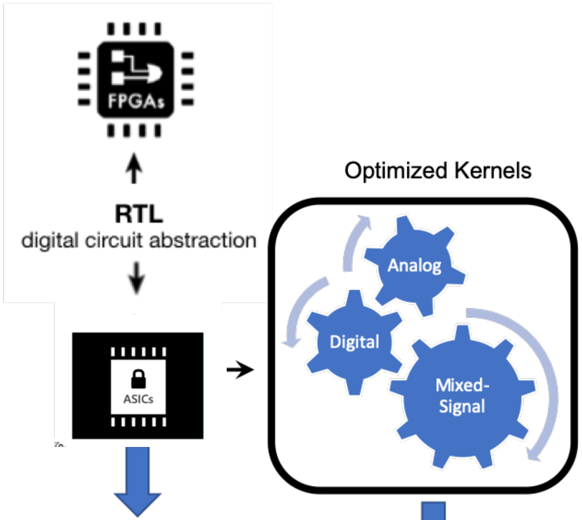
- **Feature extraction and data compression**
- **Hardware driven co-design of algorithm**





Towards heterogenous system on-chip





Photonic implementation

Electronic - Photonic hybrid

Integrated memory for in-memory compute

OPTIMIZATION FOR POWER

- Analog Mixed-Signal Kernels
- In-memory compute

Lowest Power Implementation of DNN

- Photonic based solution (NO Reconfigurability)

Electronic – Photonic Integrated Solution ?