AM-based Level-1 Track Trigger for CMS Phase II Upgrade

Zhen Hu

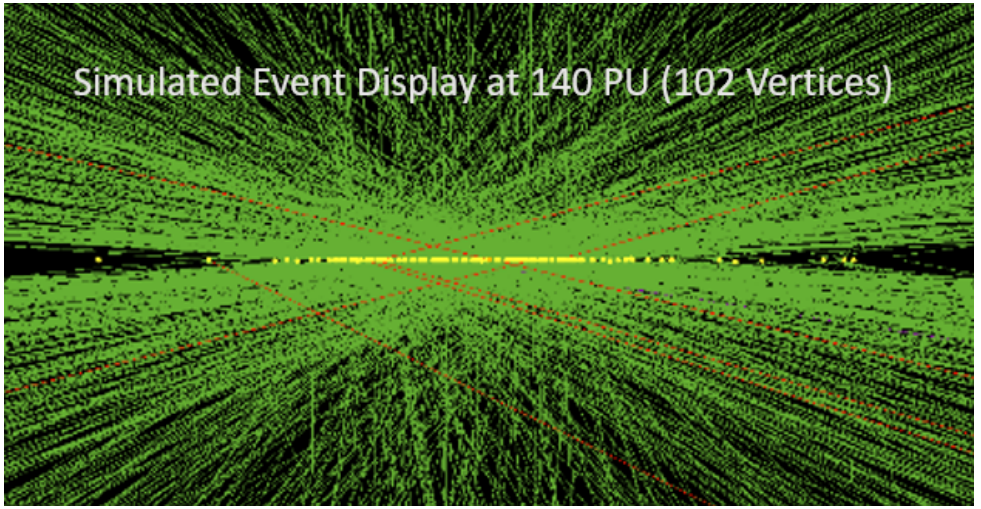
Fermilab


Contents

- Motivation
- System overview
  - Divide and conquer
  - AM introduction
  - PR core firmware introduction
- Hardware R&D
- Full system demonstration
  - Latency
  - Efficiency
- Summary and outlook

HL-LHC Trigger Upgrade

- High Luminosity LHC in 2026
  - 40 MHz bunch crossing
  - Up to 200 pileup (PU)
  - Tons of data
- Current Level-1 Trigger not working
  - Maximum bandwidth
    - only 100 kHz
  - For HL-LHC, current trigger system would give
    - EG rate @25 GeV → 100 kHz
    - Overall Trigger Rate → 1000 kHz (*unsustainable*) to reach physics goals
  - Increasing trigger threshold → lose the opportunity of new physics with low threshold



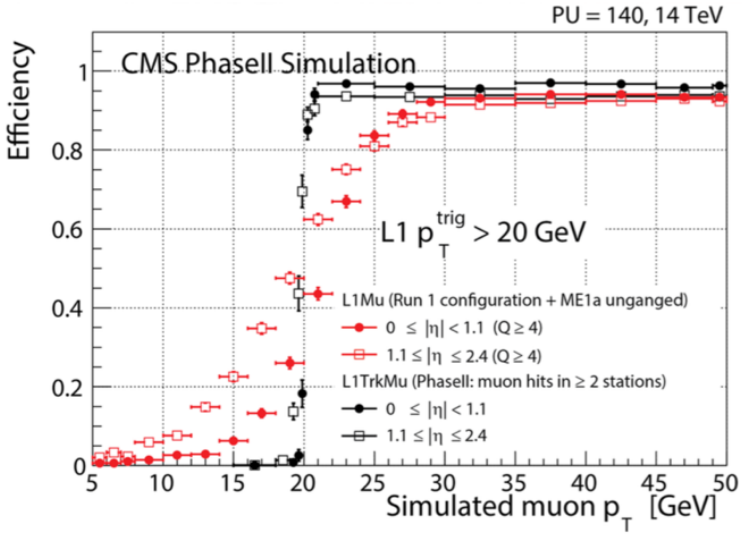
- Upgrade trigger system
  - Must increase total bandwidth
  - Must increase trigger capabilities
  - ❖ **Level-1 Tracking is a completely NEW handle**

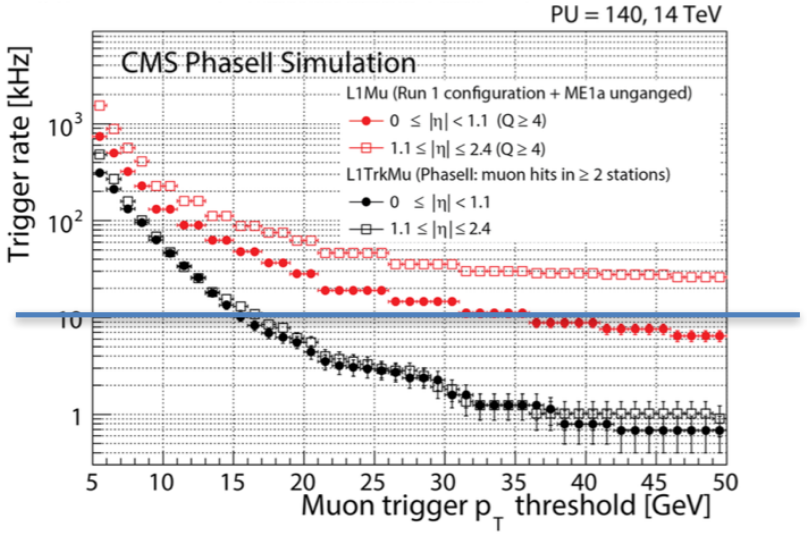
Track trigger advantage and challenge

- Silicon based tracking trigger is crucial for CMS Phase2 upgrade
  - Sharp turn-on efficiency curve
  - Background rate reduction → allows for low object threshold

- Huge challenges
  - How to handle readout of the entire tracker?
    - 260M channel, 40 MHz, 100 Tbps data (the total bandwidth of the world submarine cables a few years ago)
  - ~5 microseconds latency:
    - Data distribution, track reconstruction, track fitting …

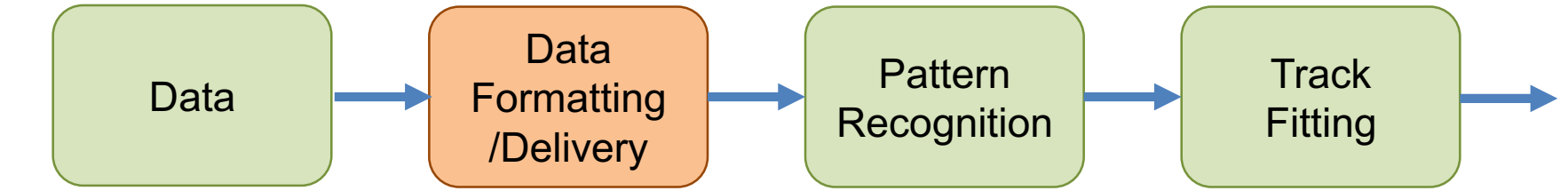
Red: without tracker
Black: with tracker



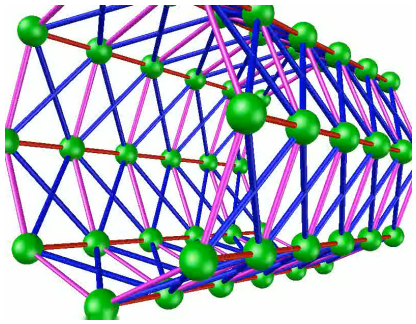


TP for Phase-II Upgrade

Solution : divide and conquer

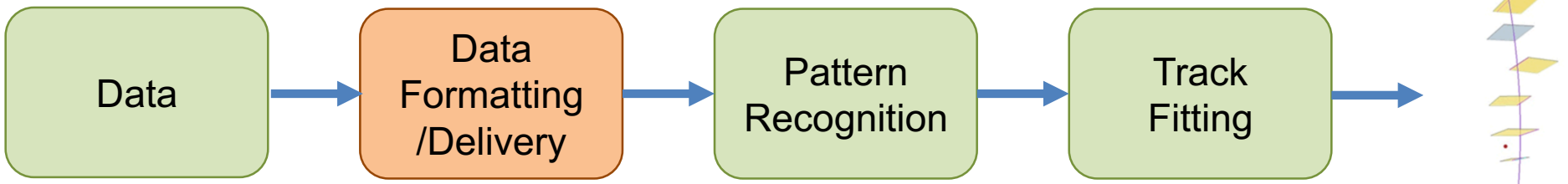


CMS Tracker

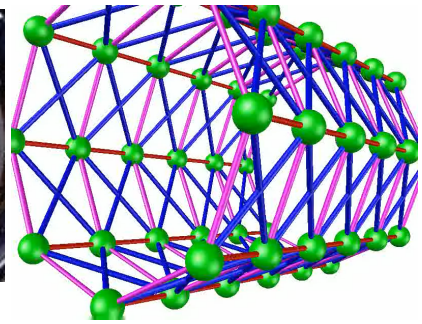


- Space parallel
  - 6*8 trigger tower
  - 100 Tbps → ~2Tbps

Solution : divide and conquer



CMS Tracker



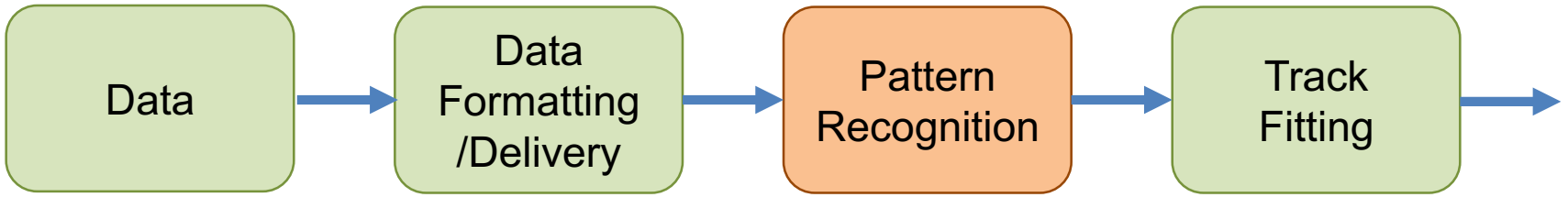
- Space parallel
  - 6*8 trigger tower
  - 100 Tbps → ~2Tbps
- Time parallel
  - 8x time multiplexing
  - 25ns → 200ns

Conventional

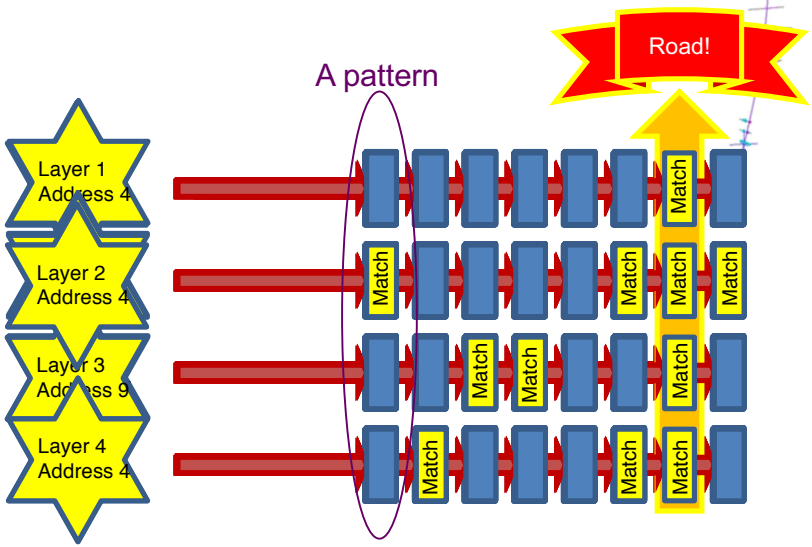
ATCA full mesh

Huge amount of cabling work without ATCA
We were the only CMS group developed ATCA in 2016

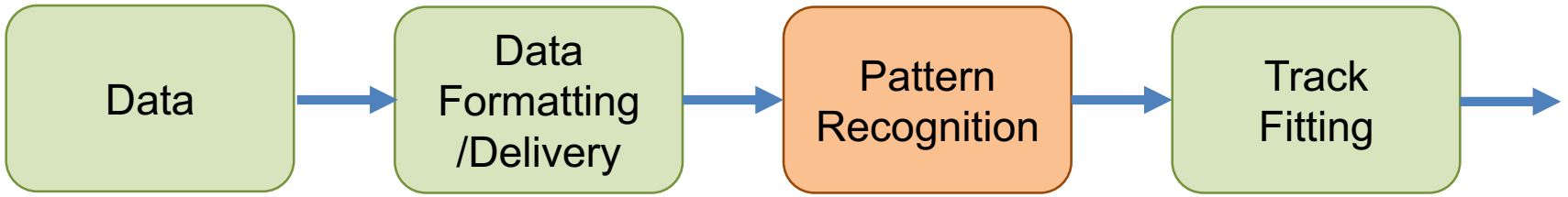
Solution : Associative Memory



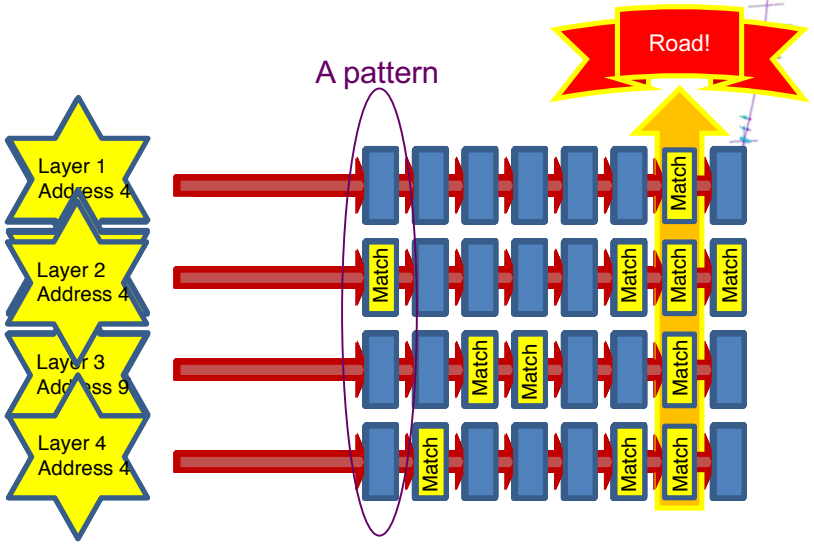
- Associative Memory
  - Based on *CAM cells to match* and *majority logic* to associate hits with a set of pre-determined hit patterns that represent allowed track trajectories (massively parallel, *finishes right after all hits arrive)*

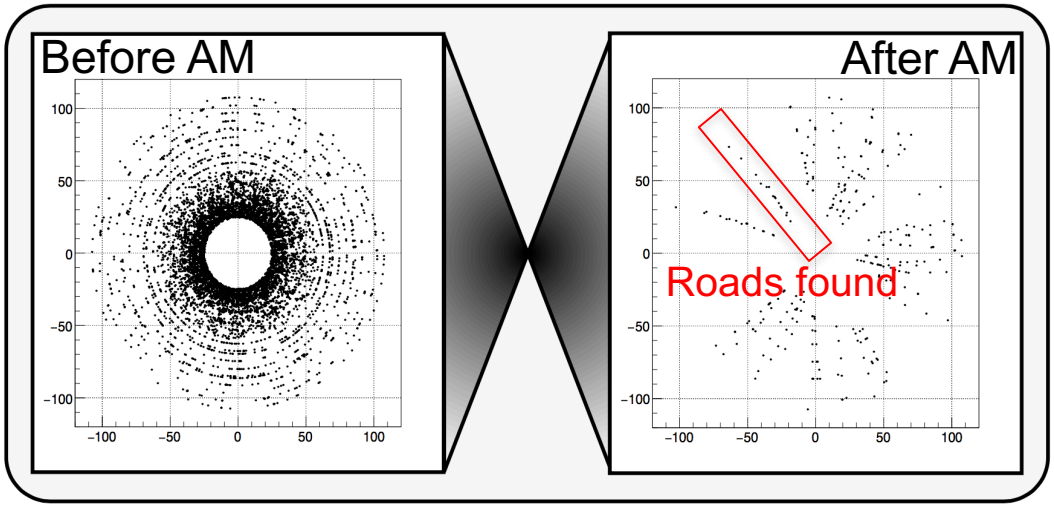


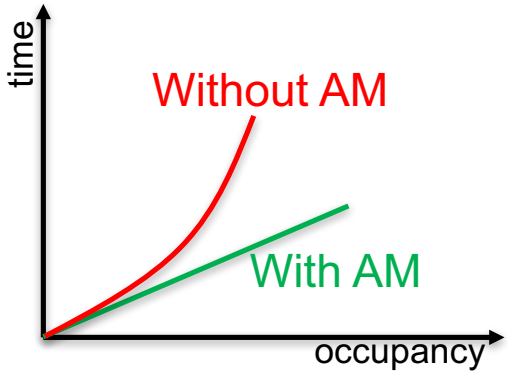
Solution : Associative Memory



- Associative Memory
  - Based on *CAM cells to match* and *majority logic* to associate hits with a set of pre-determined hit patterns that represent allowed track trajectories (massively parallel, *finishes right after all hits arrive)*
- Perfect tool to mediate the high pileup effect
  - Avoiding the typical power law dependence of execution time on occupancy

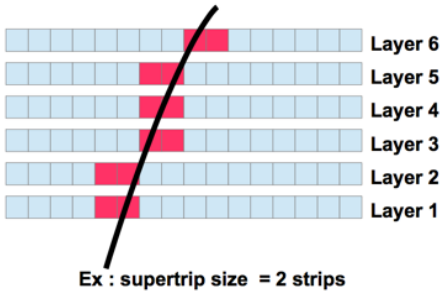


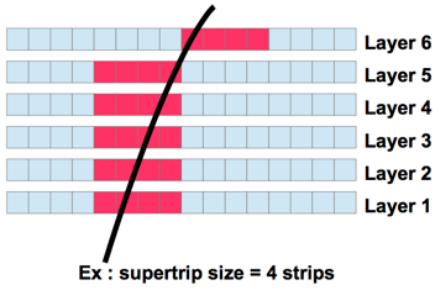


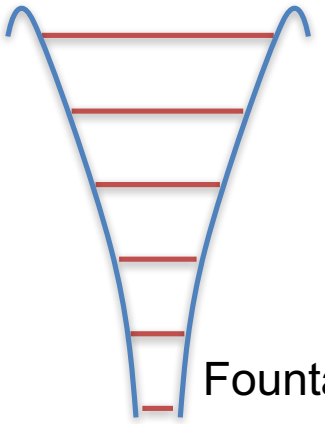


AM Pattern and Bank

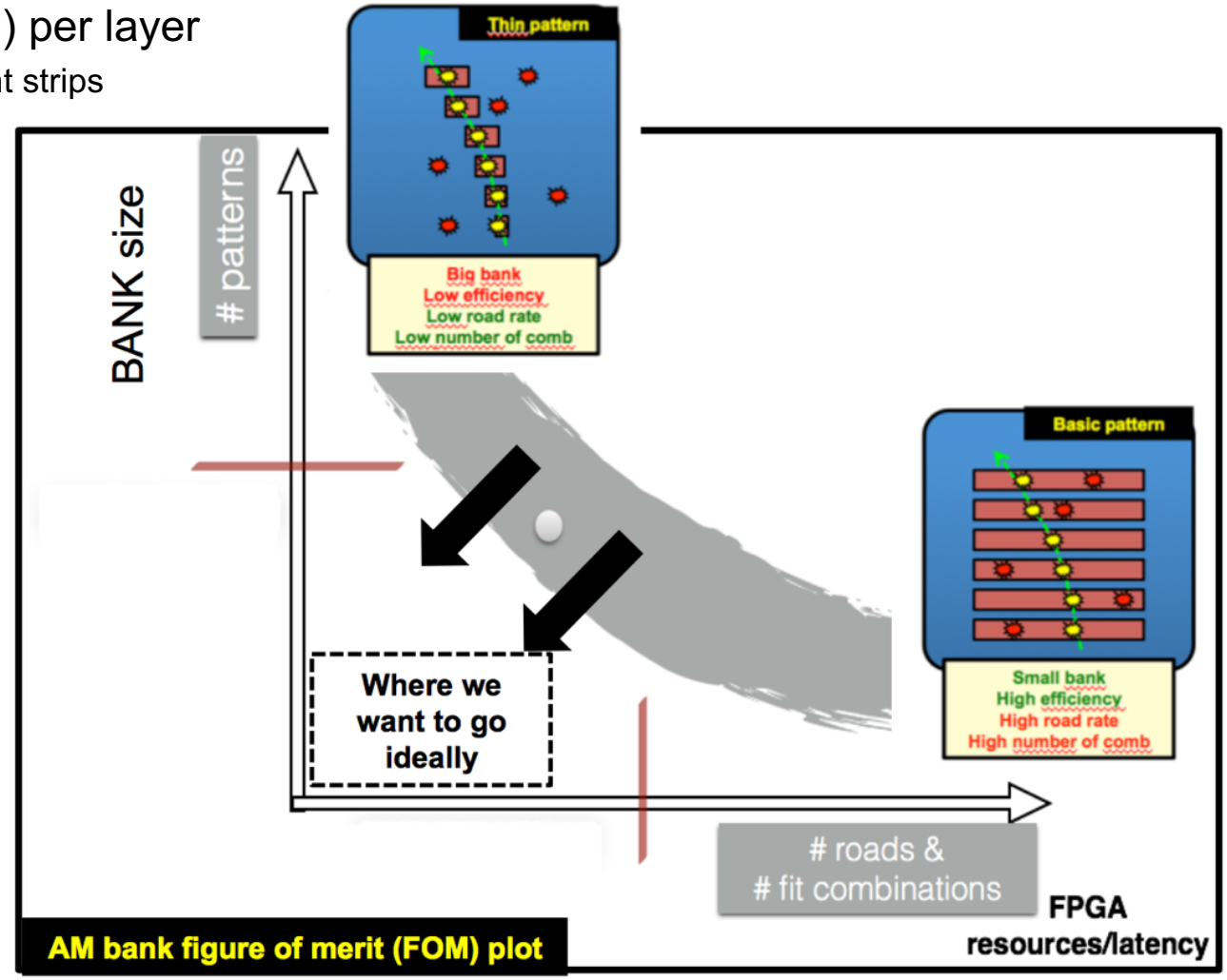
- A pattern is a low resolution track
  - Made of 1 superstrip (SS) per layer
    - A SS is a group of adjacent strips





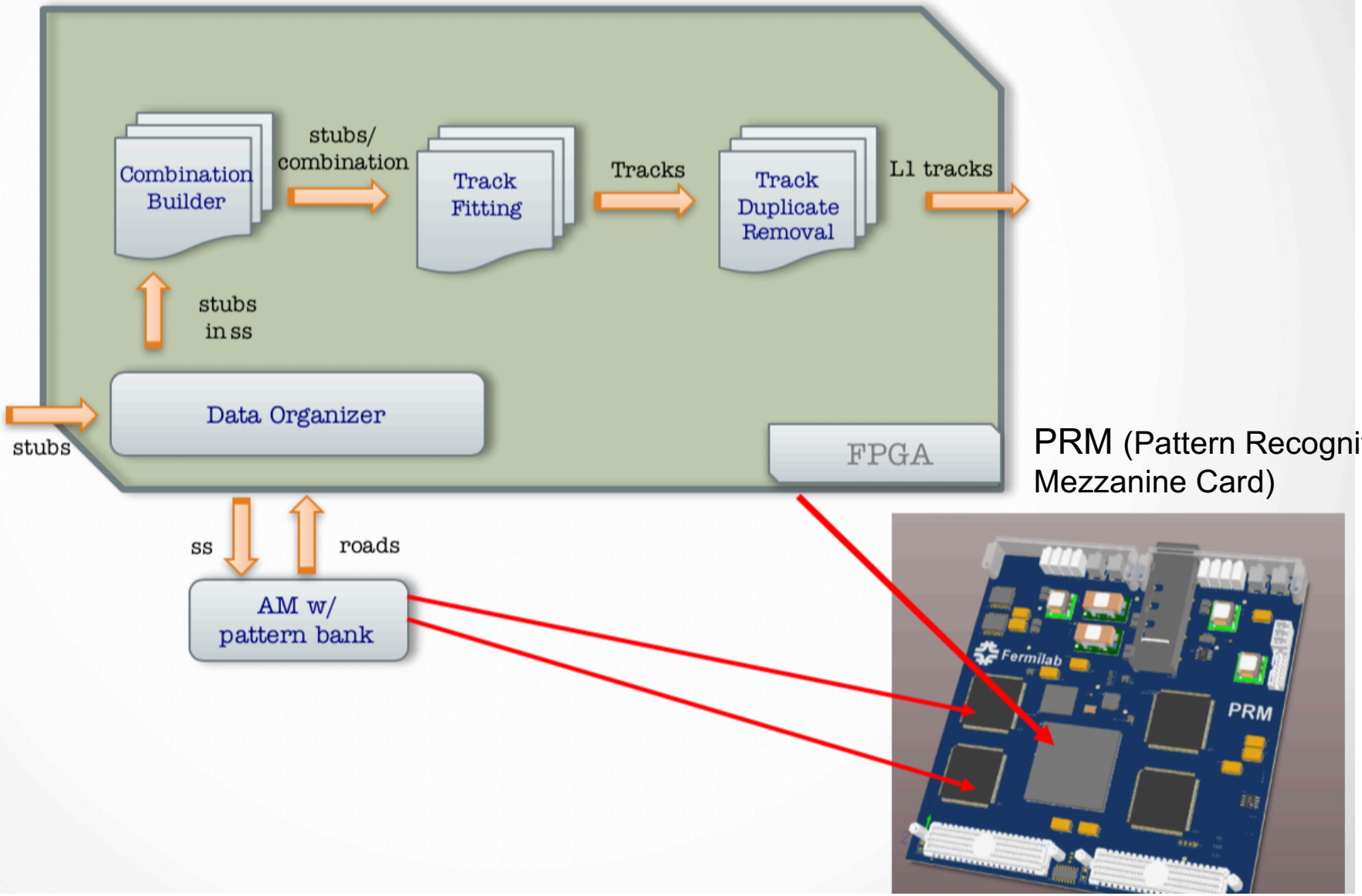


Fountain SS



More powerful AM => less demand on the FPGA
More powerful FPGA => less demand on the AM

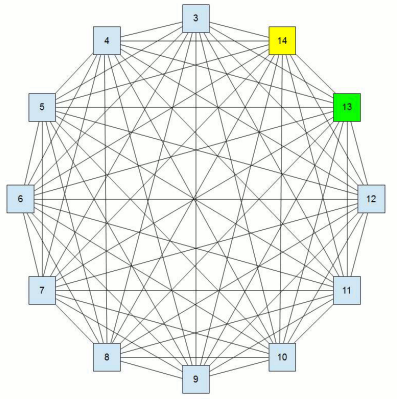
Core firmware design



PRM (Pattern Recognition Mezzanine Card)

FNAL based hardware R&D

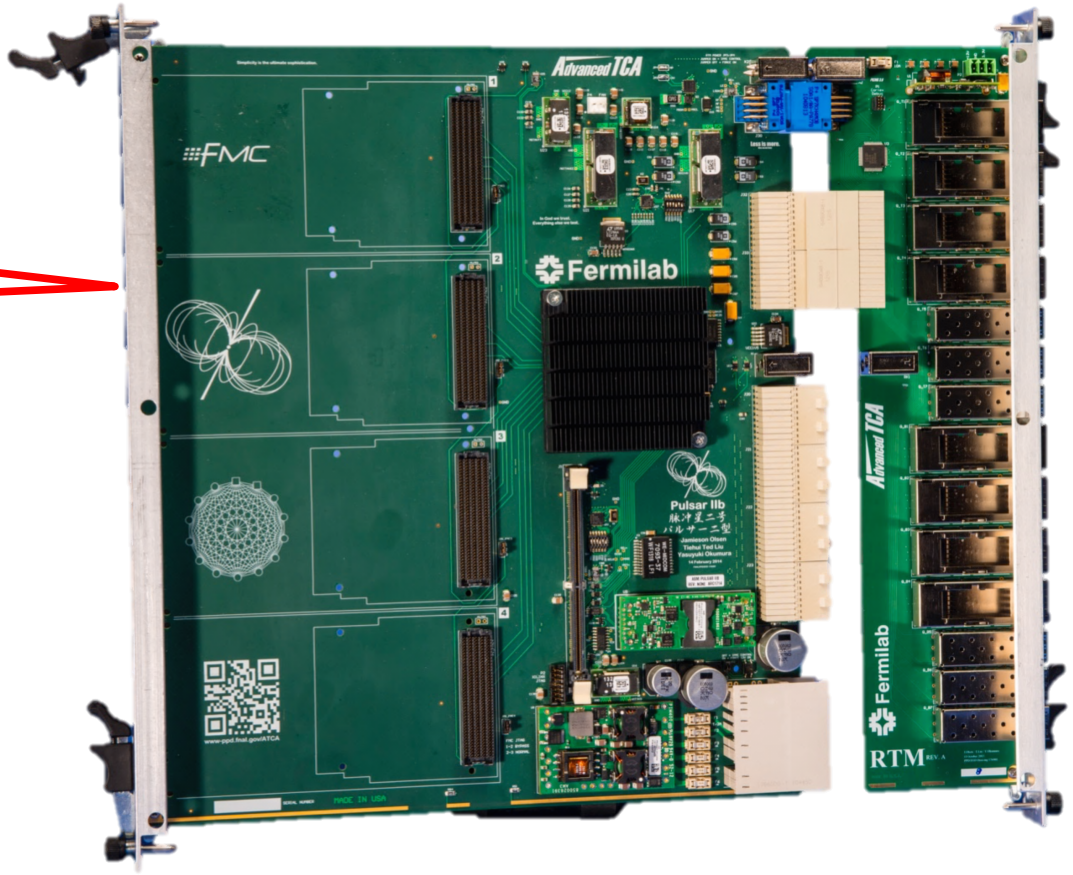
ATCA shelf



FNAL based hardware R&D

ATCA shelf

ATCA Processing Blade: Pulsar2b



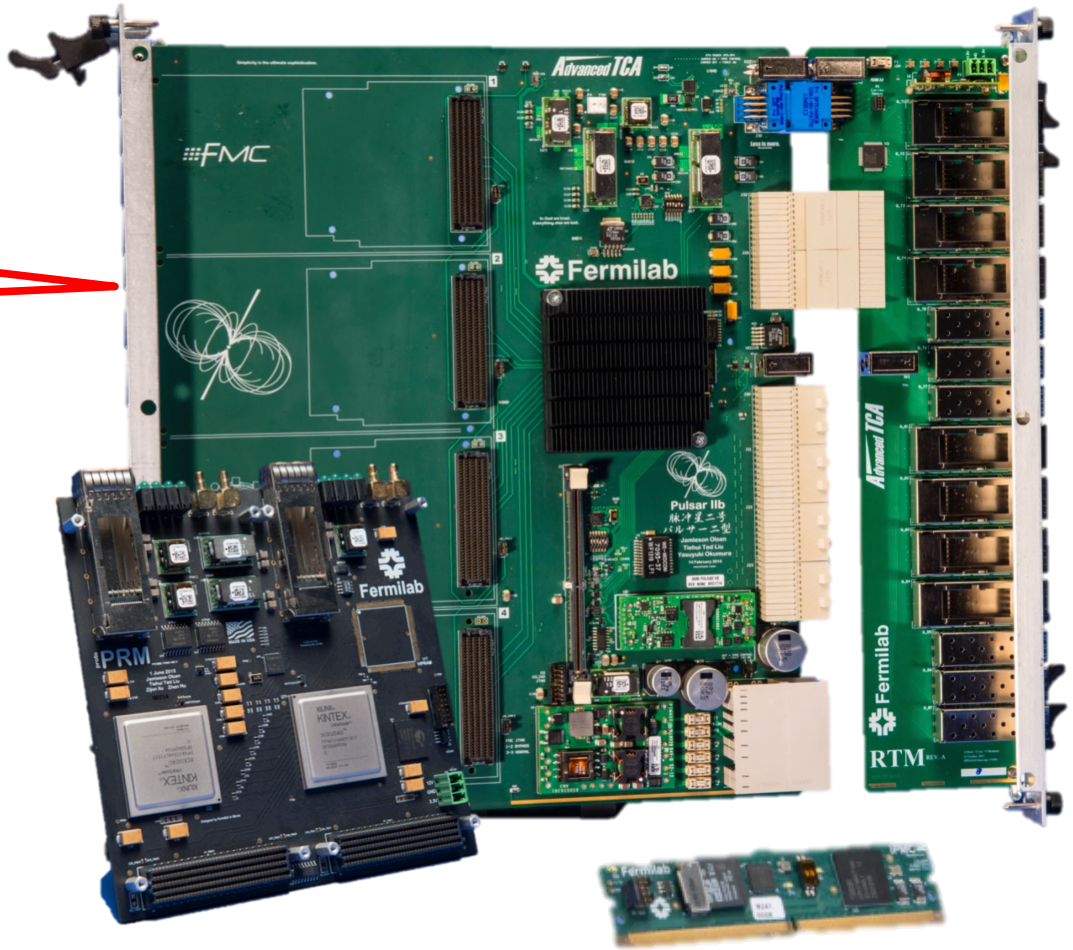
Pulsar2b RTM (Rear Transition Module)

We tested the data transfer performance for the full mesh back plan, Pulsar2b and RTM

FNAL based hardware R&D

ATCA shelf

ATCA Processing Blade: Pulsar2b



Pulsar2b RTM (Rear Transition Module)

We tested the data transfer performance for PRM

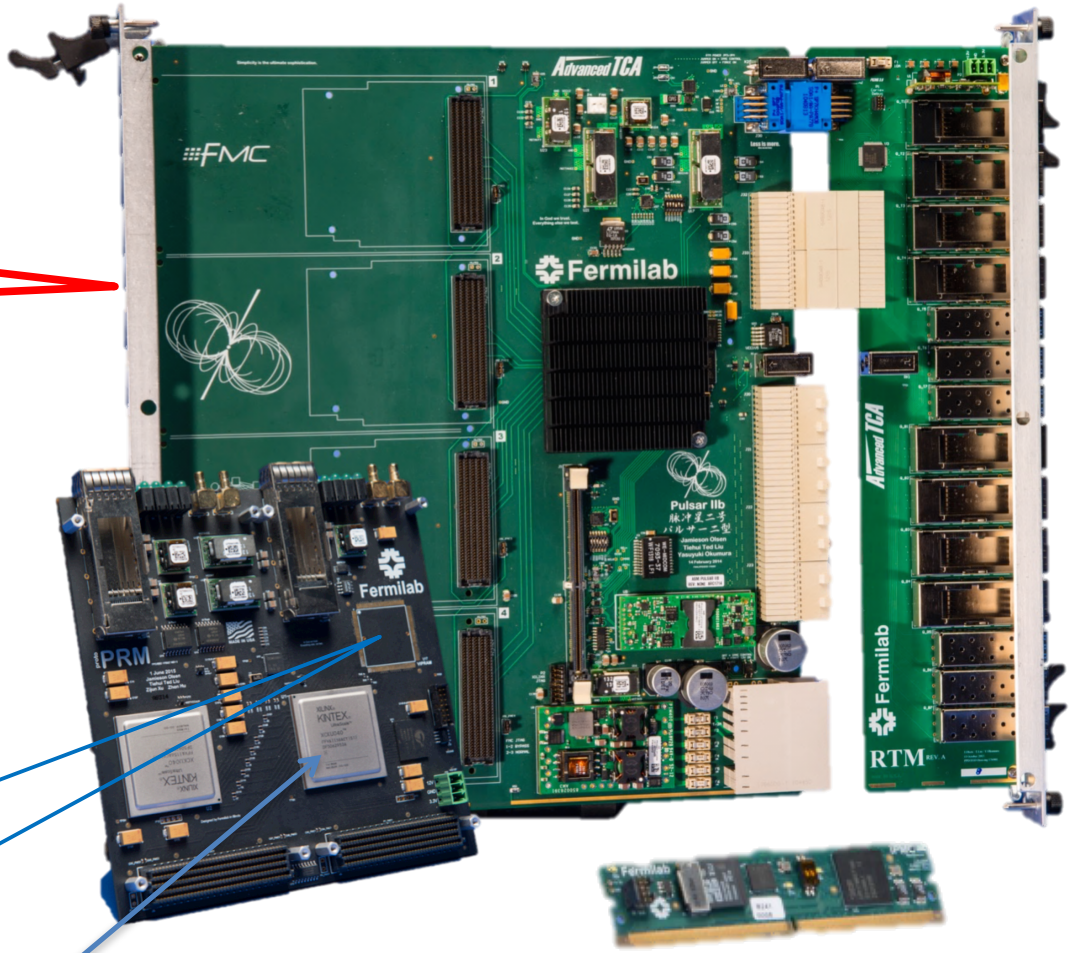
PRM (Pattern Recognition Mezzanine Card)

IPMC (Intelligent platform management controller)

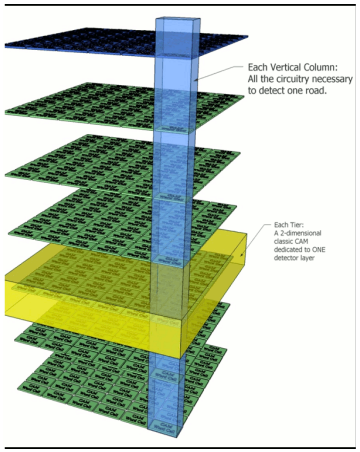
FNAL based hardware R&D

ATCA shelf

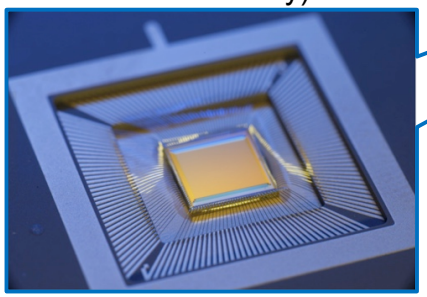
ATCA Processing Blade: Pulsar2b



Pulsar2b RTM (Rear Transition Module)



VIPRAM ASIC
(Vertically Integrated
Pattern Recognition
Associative Memory)



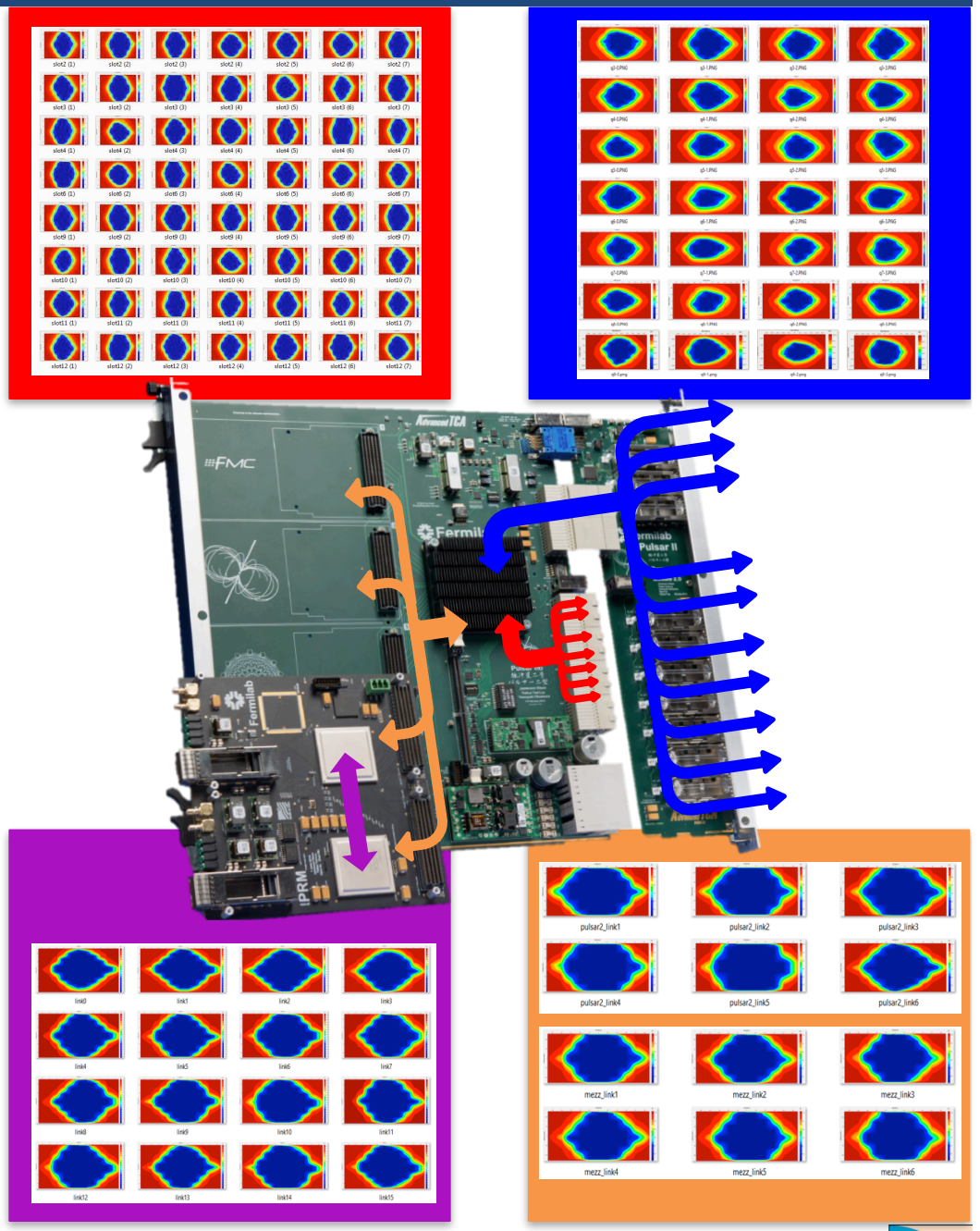
PRM (Pattern Recognition Mezzanine Card)

IPMC (Intelligent platform management controller)

Currently we emulate AM with FPGA

Excellent hardware performance

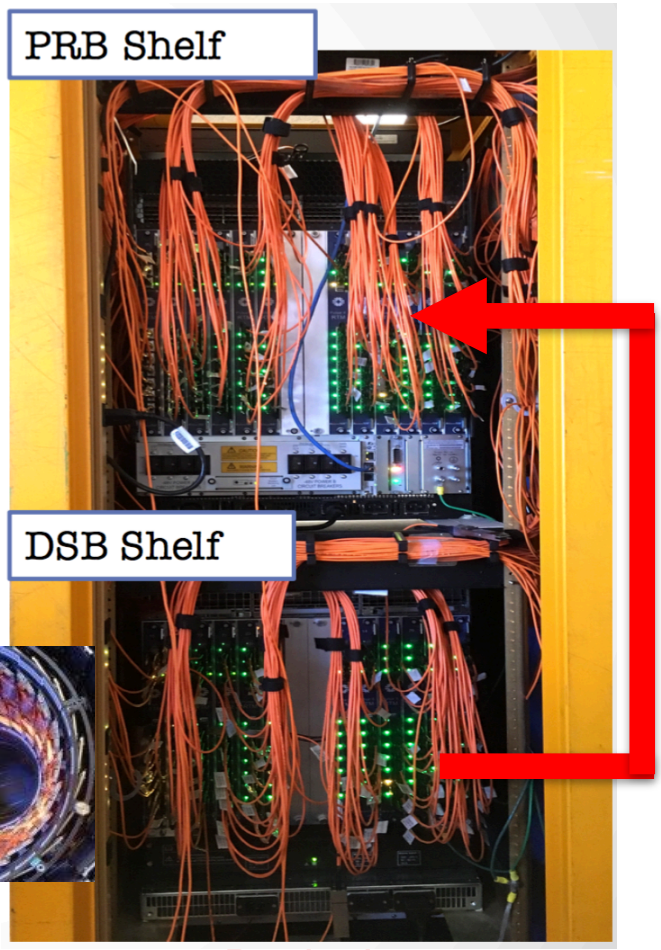
- ATCA shelf
  - 10 blades for parallel processing
  - Full mesh backplane is a natural solution for time multiplexing
    - All of the 56 bidirectional links among 8 Pulsar2b boards were tested at 10Gbps
- Rear Transition Module
  - 10 QSFP bidirectional links
    - 10 Gbps per link achieved
- PRM performance
  - Communication between Pulsar2b and PRM FPGAs
    - 10 Gbps achieved
  - Two latest generation of Xilinx FPGAs
    - Interconnection achieved 16.3 Gbps



Full system demonstration at Fermilab

- Using the technology today to demonstrate track trigger feasibility
- Two shelves fully loaded with Pulsar-2b boards for 1 Trigger Tower

Front view



Back view

Pattern Recognition Board (PRB) shelf

- 10 Pulsar IIb
- Some boards with PRM Mezzanines

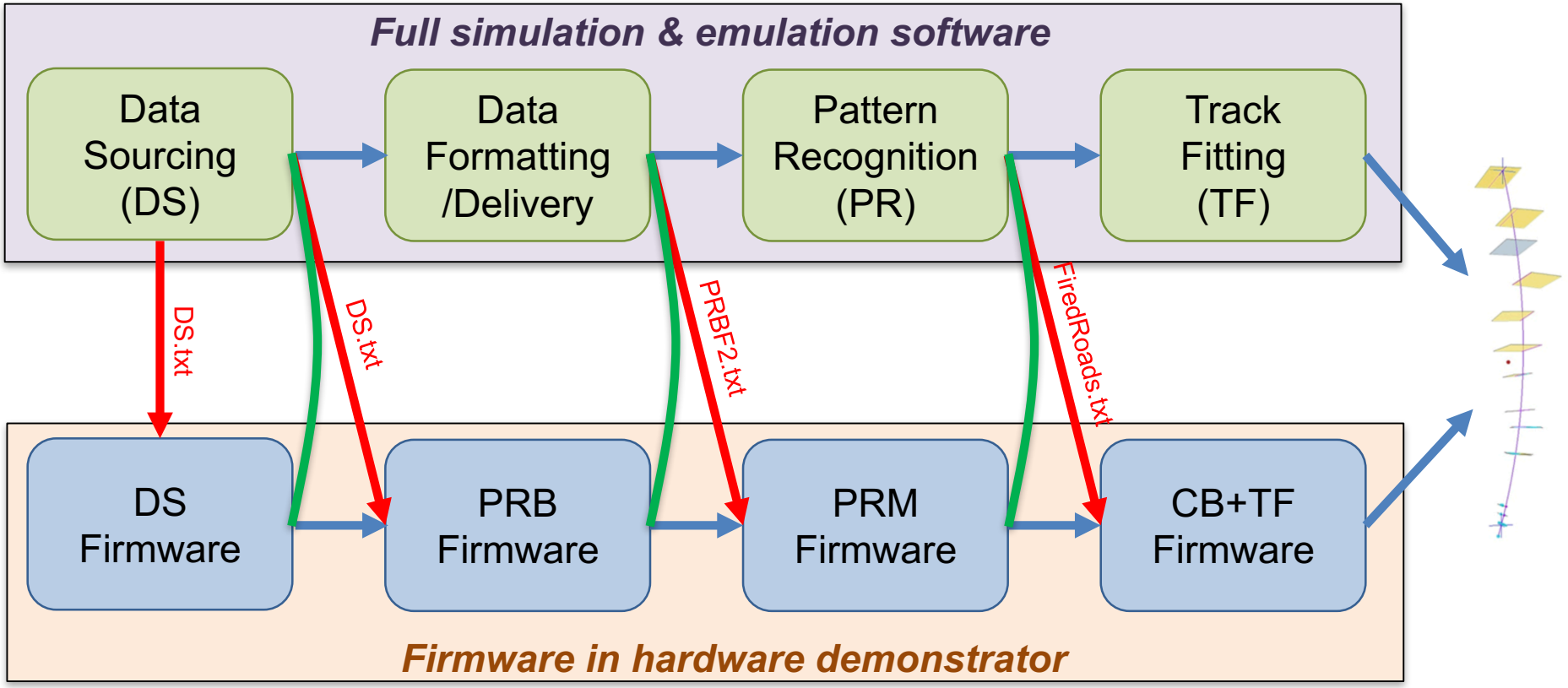
(Bandwidth between any pair of Pulsars is 20Gbps)

Data Source Board (DSB) shelf

- Emulates the output of ~400 modules
- 10 Pulsar IIb
- 100 QSFP+ fibers

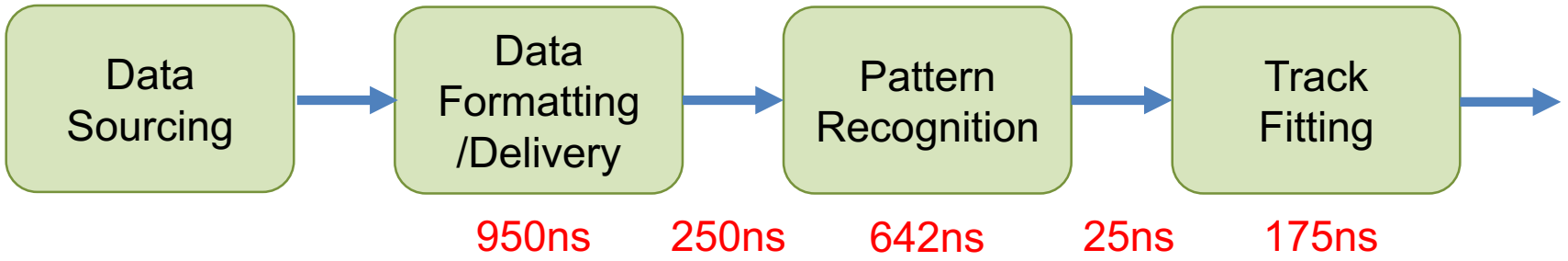
(Capable of sourcing up to 4.8 Tbps data with full shelf)

Demonstrator validation



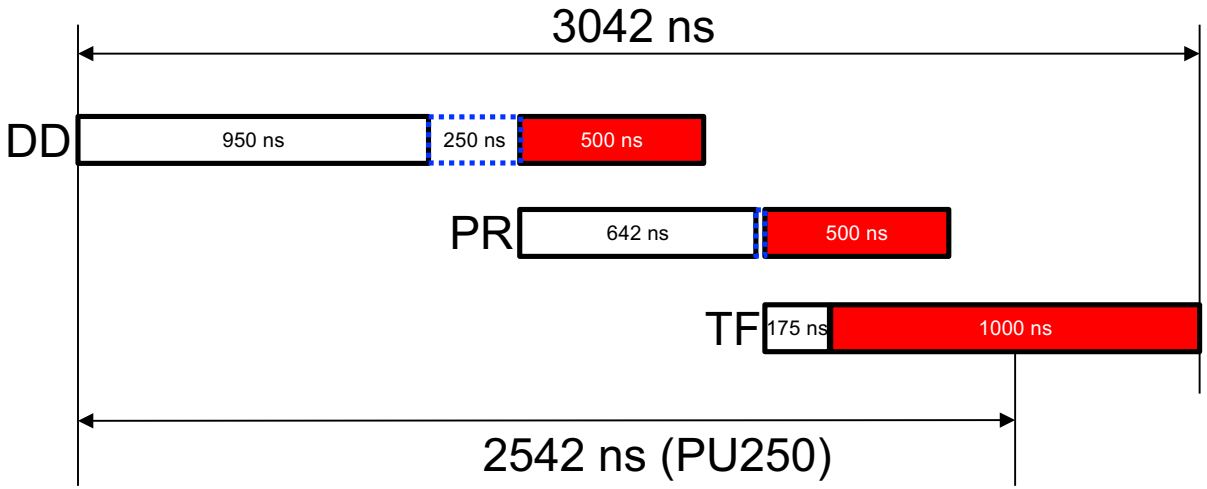
- Hardware and emulator perfectly matched
  - Output from each stage validated bit-by-bit
- With the full chain demonstrator, we have measured:
  - Latency, FGPA resource usage, efficiency, resolution

Excellent latency achieved

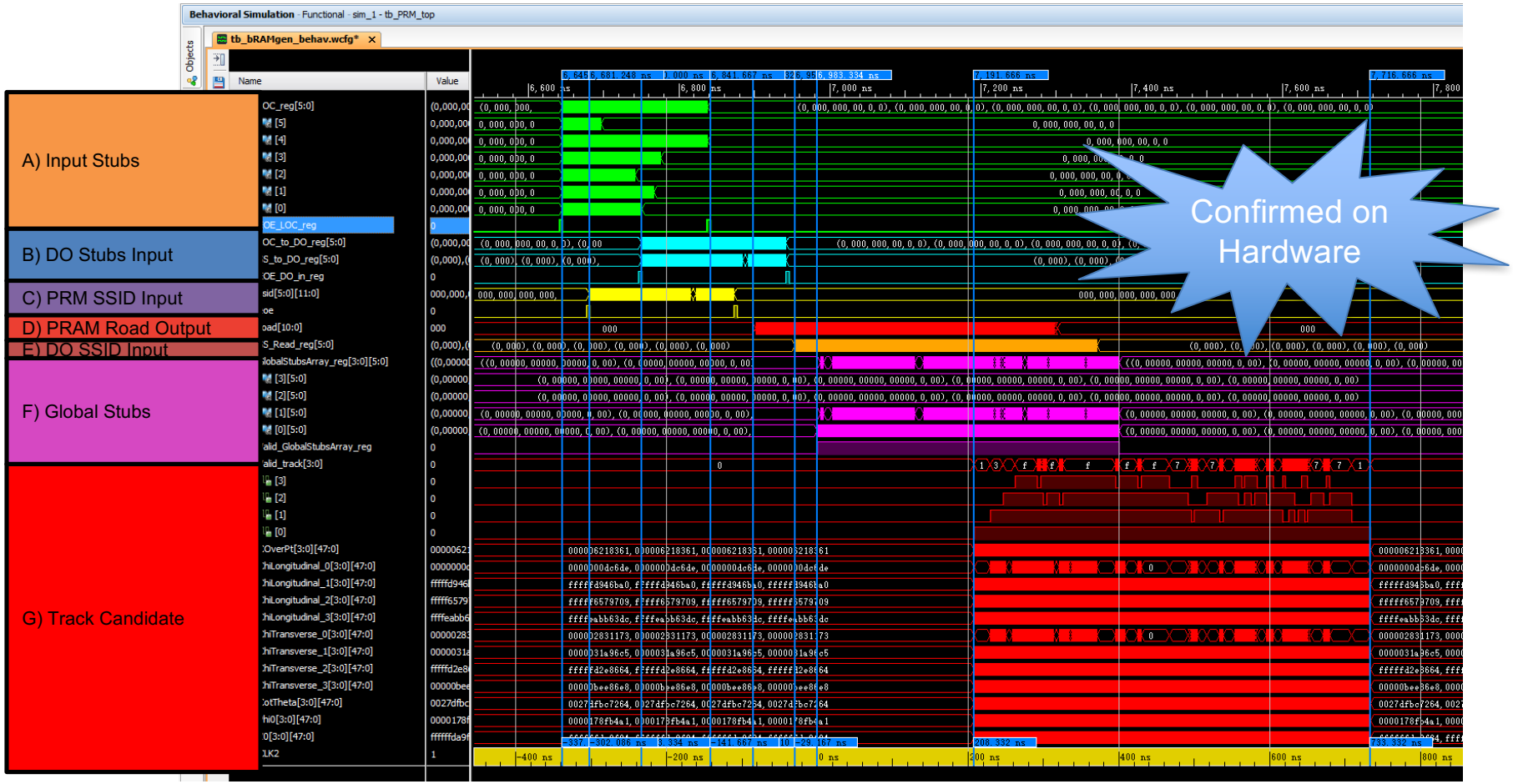


	Description	Latency	Start (ns)	End (ns)
1	Module data arrives at RTM		0	200
	MGT RX + Align	125ns		
	Input Flatten and Format	200ns		
2	Stubs after formatting		325	525
	PRB BX sort	200ns		
	MGT TX	100ns		
3	Stubs transfer on Full Mesh Backplane		**625**	**825**
	MGT RX + Align	125ns		
	PRB Layer sort	200ns		
4	PRB stubs after layer sort		950	1450
	PRB PRBF.2 Formatting, FIFO	6 clk		
	MGT TX	100ns		
5	Stubs transfer to PRM		**1075**	**1575**
	MGT RX + Align	125ns		
6	Stubs in PRM after alignment		1200	1700
	Local to SSID conversion	4 clk		
7	Output of Local to SSID block		1217	1717
	ODDR output DDR registers	4 clk		
8	SSIDs input to AM		1233	1733
	AM latency EOE to Road output	6 clk		
9	AM roads out		1758	2258
	Input DDR registers + FIFO	10 clk		
10	Road ID after FIFO		1800	2300
	Road to SSID conversion	3 clk		
11	SSIDs input to DO		1813	2313
	DO read latency	3 clk		
12	DO Local stubs output		1825	2325
	Local to Global conversion	4 clk		
13	Global Stubs write into FIFO		1842	2342
	Global Stub FIFOs	6 clk		
14	Global stubs available to CB+TF array		1867	2367
	CB+TF latency	42 clk		
15	Track Parameters out of TF array		**2042**	**3042**

Data delivered to PRM starts/ends: @1.20 – 1.70 μs
Pattern Recognition output starts/ends: @1.84 – 2.34 μs
Track Fitting output starts/ends: @2.04 – 3.04 μs



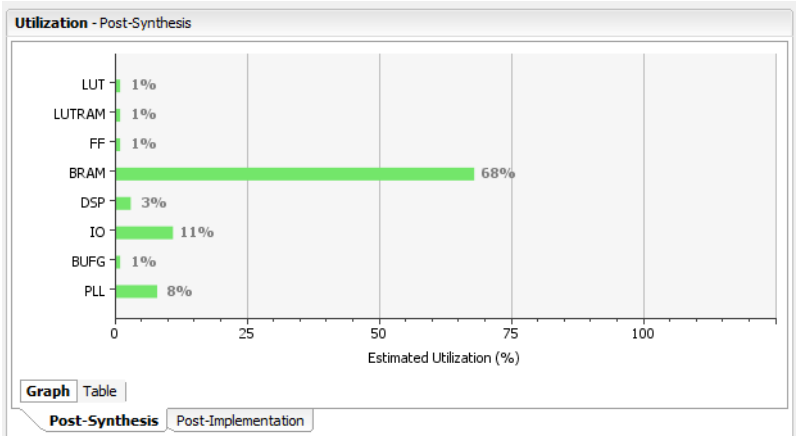
Example events in Vivado



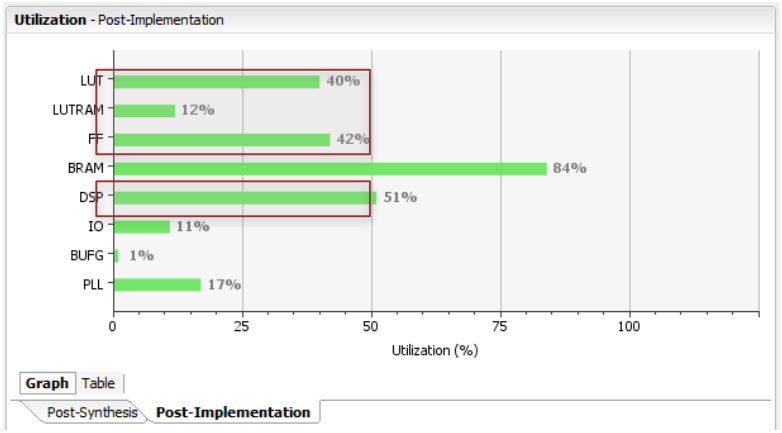
4x(ACB+LTF), 7191.666 ns to 7716.666ns = 525ns, 126 clk @240MHz

This is with a selected ttbar + PU200 event with high tail of combinations

FPGA Resource Utilization (KU060)



PR stage only

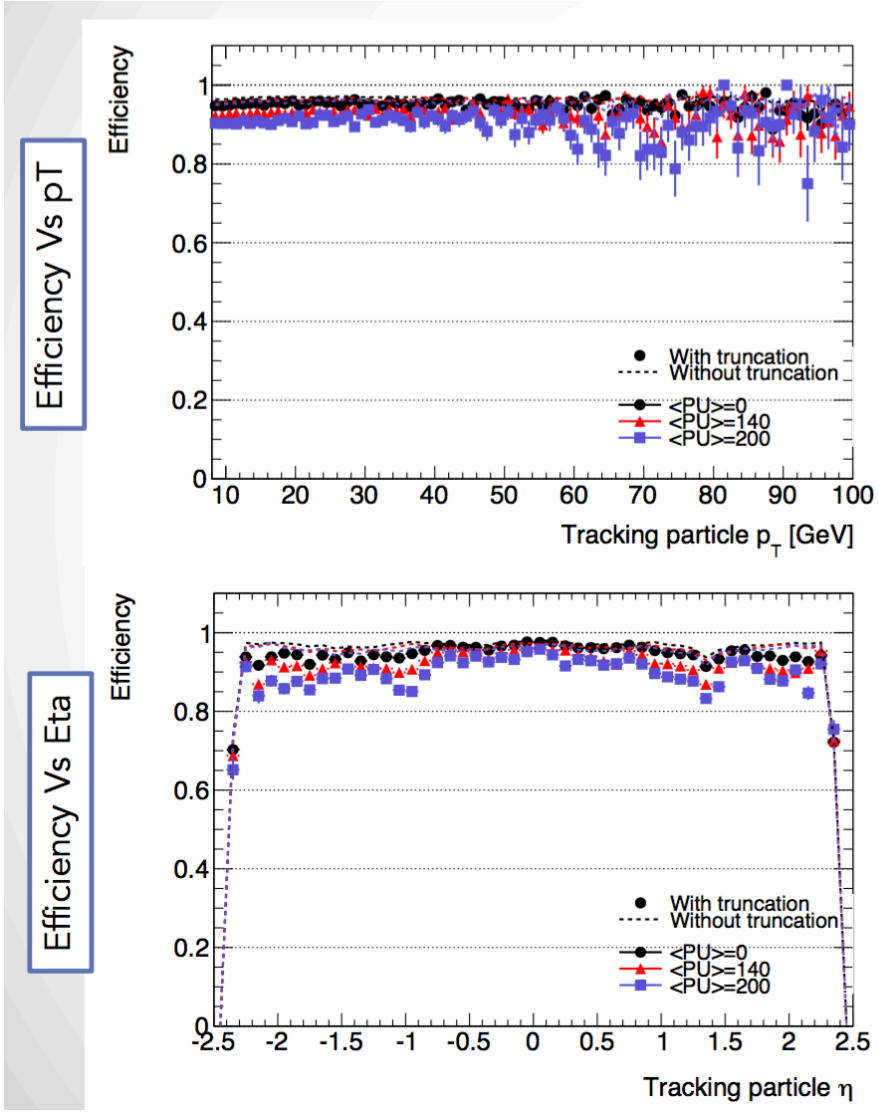


PR + 2[FIFO +4(CB+TF)]

- Very light weighted design
- BlockRAM mainly used for DO
  - TF does not increase BlockRAM usage, leaving enough room for TF
- Modest increase in registers and DSP blocks
  - Plenty of room for parallel copies of the fitter

High efficiency up to PU250

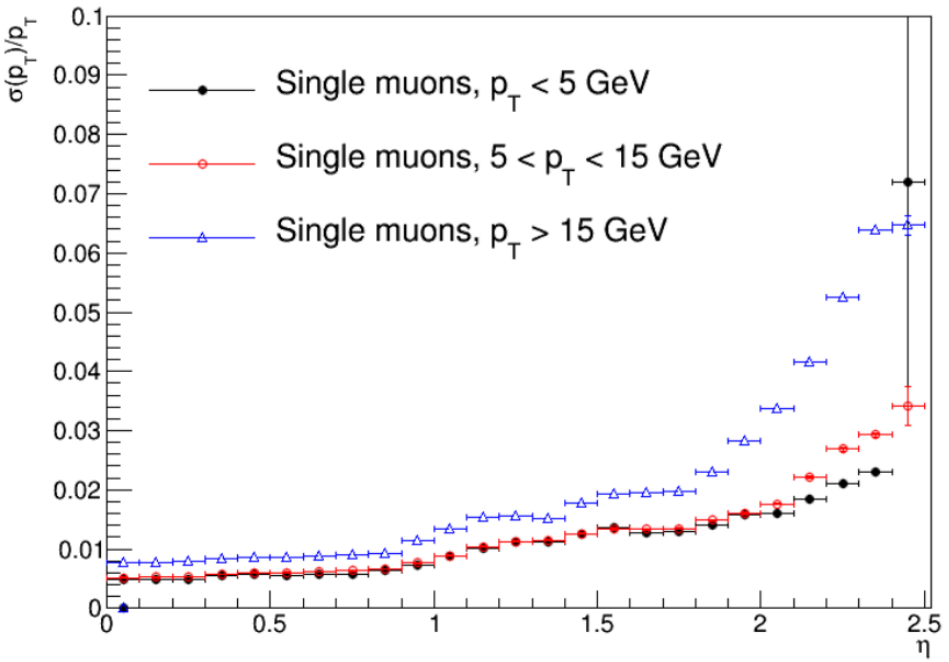
- System is robust against higher luminosity or increase in stub occupancy
  - We demonstrate that the system reconstructs all tracks for events with PU250 within 2.5 μs (no truncation needed)
  - Only for high pT jet, truncation needed to meet the pipeline window



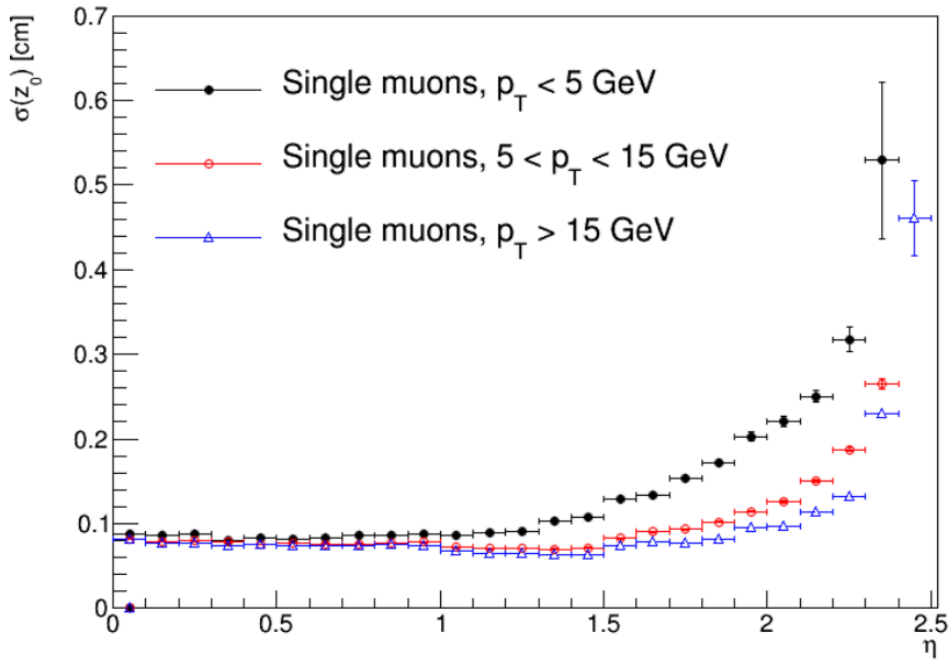
Resolution

- Excellent performance for L1 application

p_T resolution



z_0 resolution



Summary

- A open and flexible ATCA system architecture
  - Regardless of what type of tracking algorithm
- Demonstrated with a vertical slice
  - Excellent performance with today's technology for high luminosity up to PU 250
  - Very low latency: reconstructs all tracks within 2.5 μs
  - Safety margins: 1.5 μs left to do more processing
- Even with AM ASIC doing the most heavy lifting, we still have challenges for tracking within very high p_T jets. However, we have many ways to improve this.

AM in FPGA: Overview

- AM in FPGA: very closely follows the AM ASIC (chip) design
  - Match two silicon tiers in ASIC with two modules in FPGA firmware
    - CAM Tier -> a 2D array of Pattern Modules
    - I/O Tier -> fired roads serialization and output
  - Pipelined operation
    - CAM tier: processes pattern matching with stubs for current event N
    - I/O tier: outputs road addresses for event N-1 at the same time
- CAM tier logic is optimized for 7-Series/UltraScale FPGA architecture