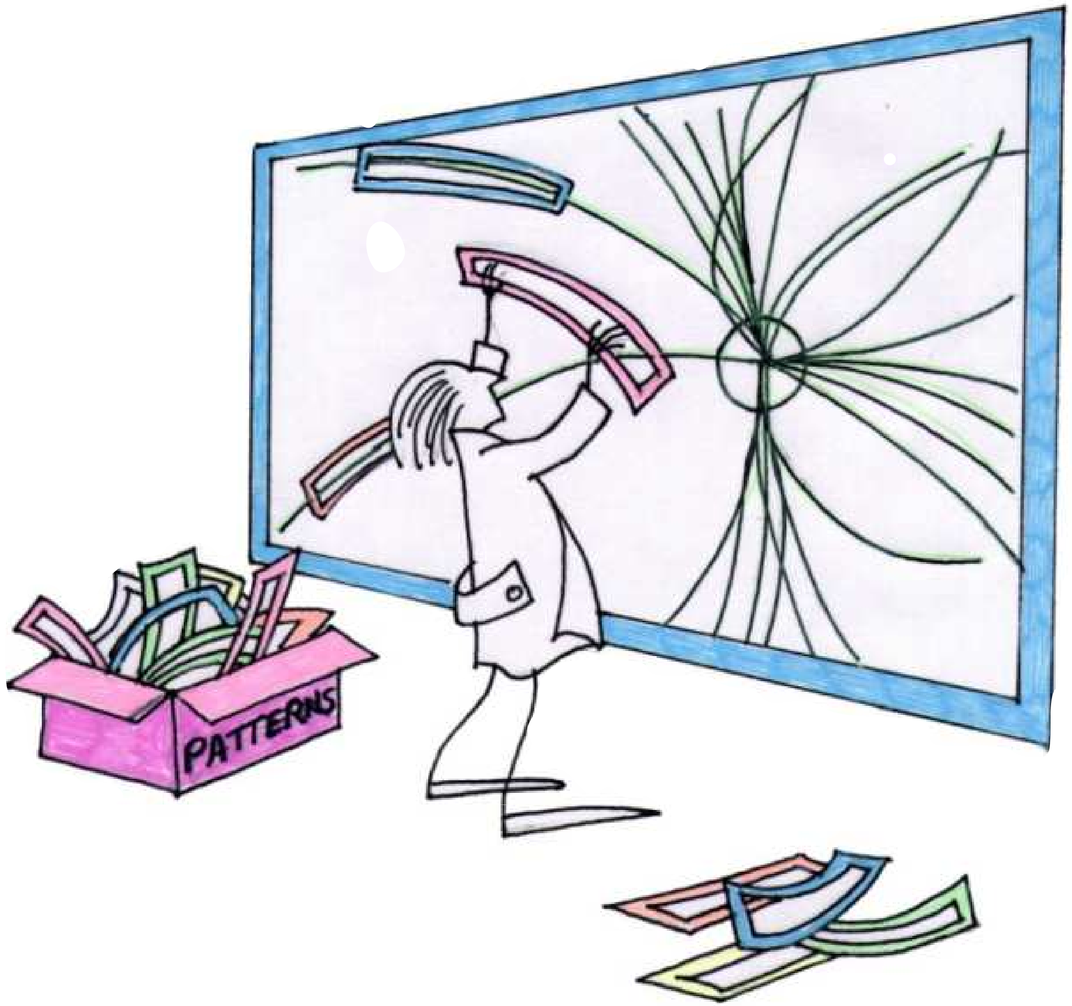
THE FTK EXPERIENCE

LAUREN TOMPKINS

MU2E-II TDAQ WORKSHOP

SEPT 14 2020



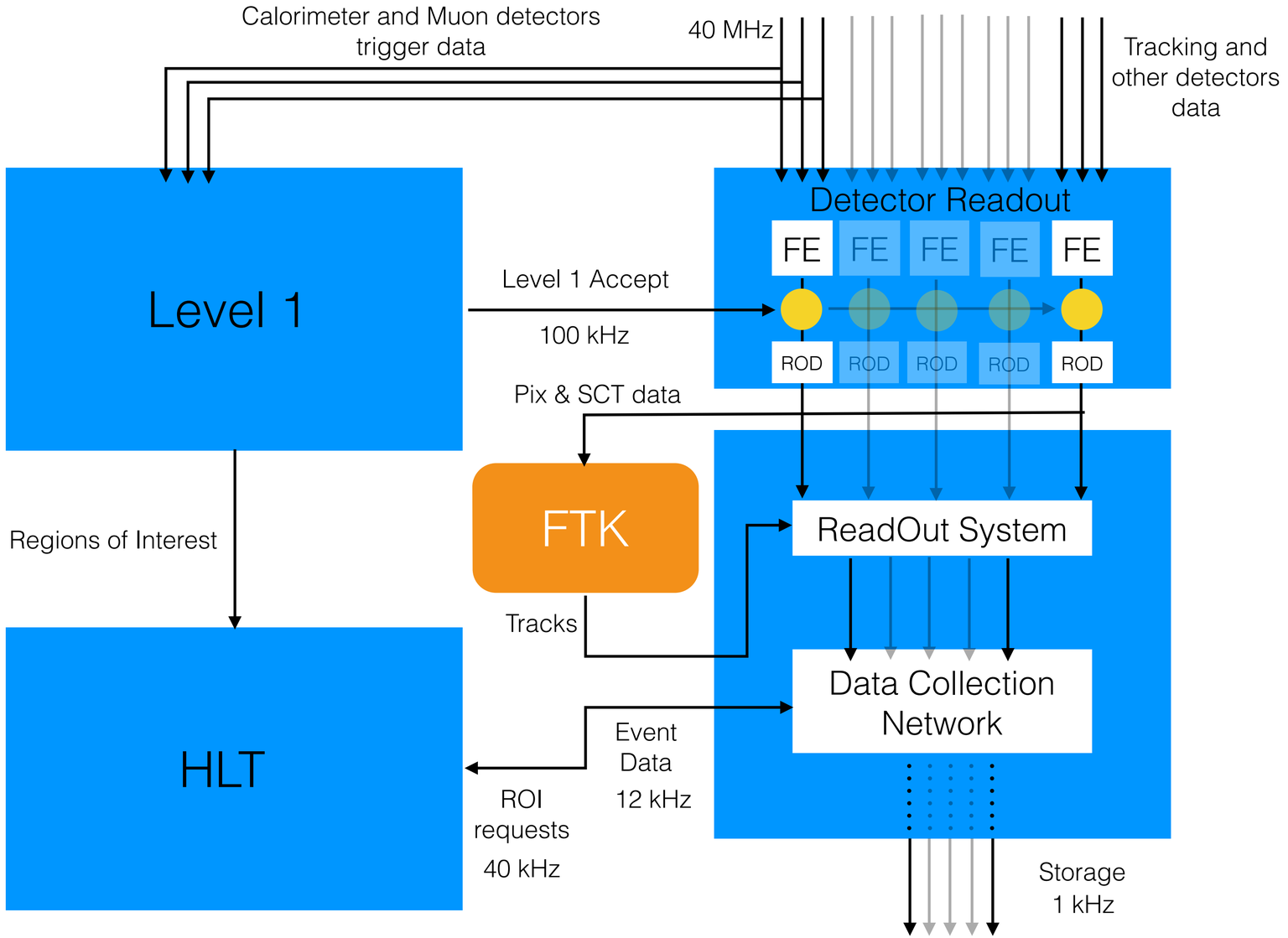
Stanford University

OUTLINE

- FTK in context
- How it worked: FTK Design & Performance
- Lessons Learned

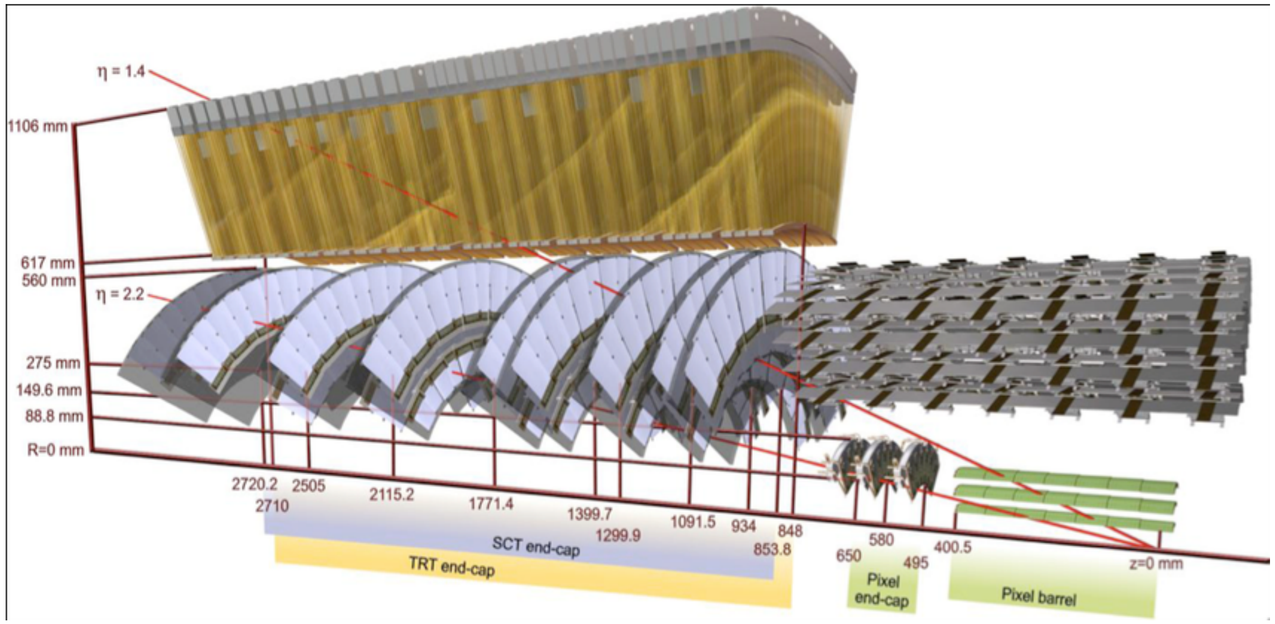
FTK IN THE ATLAS TRIGGER SYSTEM

- FTK design:
  - Co-processor for the ATLAS HLT
  - Provide full detector silicon tracking for every L1 accepted event (pT > 1 GeV, |eta| < 2.5 @ 100kHz)
    - Hadronic final states
    - Long-lived final states
    - B-physics triggers
    - Pile-up mitigation, beam spot
  - System design studies at pile-up of 70

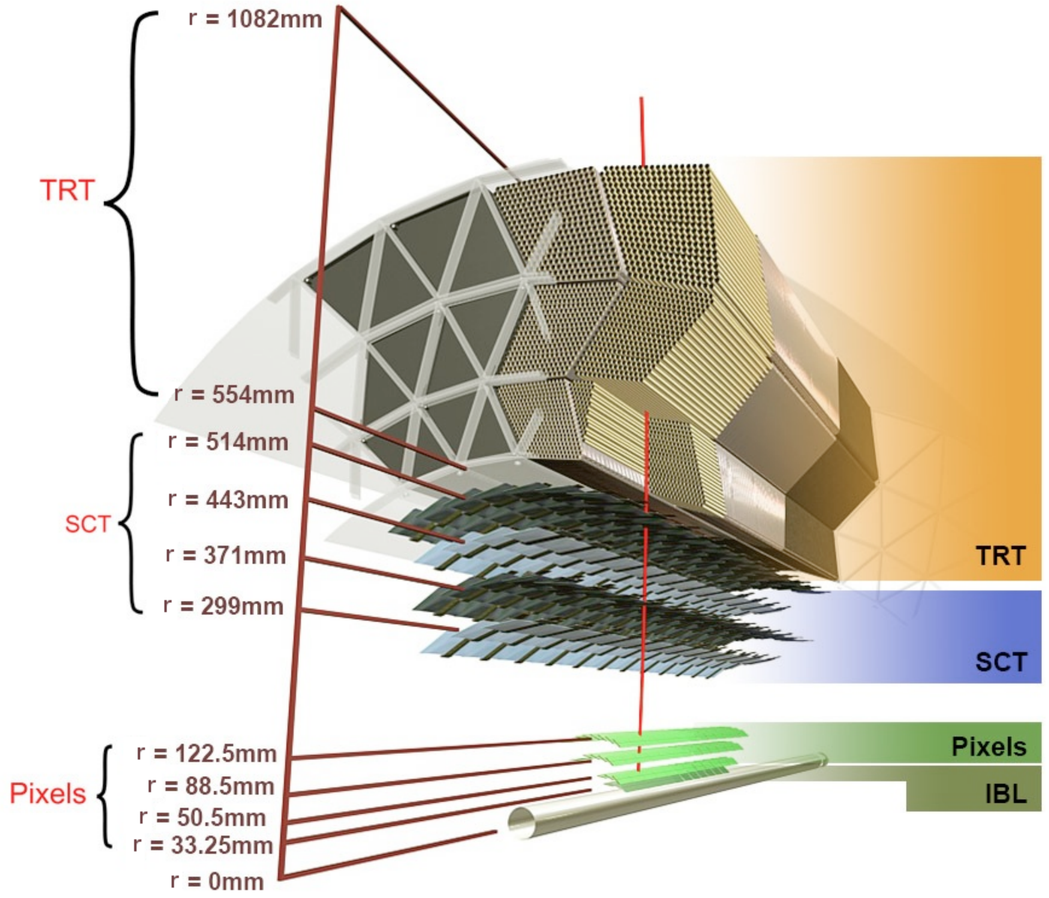


ATLAS TRACKERS

- Pixels: 50x200 microns, 4 layers, ~100 million channels
- Strips (SCT): 80micron strips, double sided axial/stereo, ~4 million channels
- 2T solonoidal magnetic field

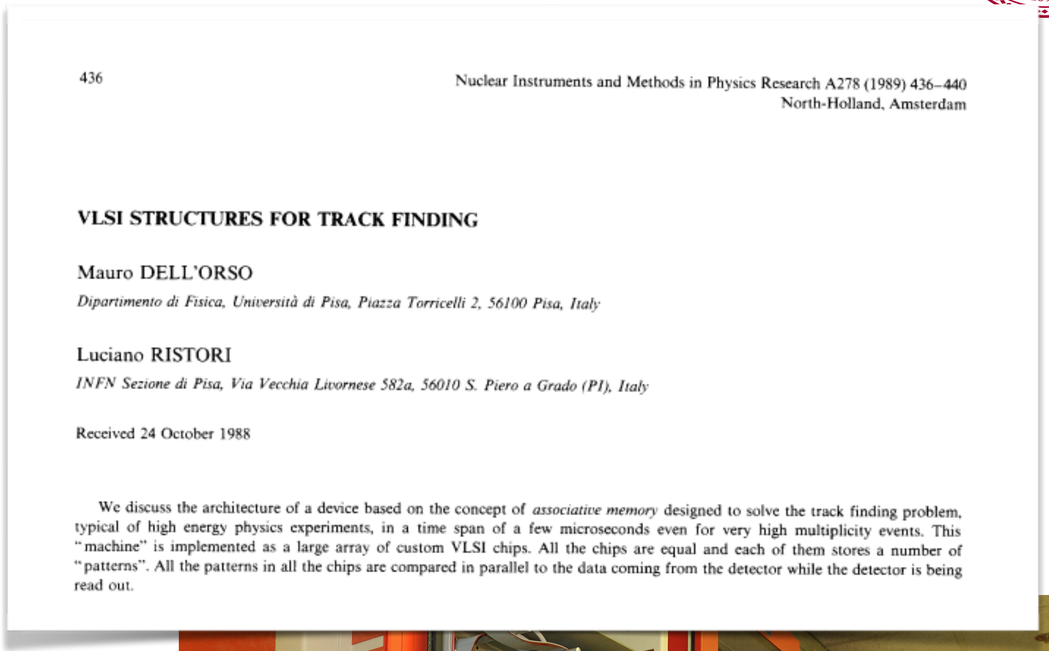


+ IBL!



CONCEPTUAL DESIGN

- **Parallelize** the problem: Divide the detector η-ϕ towers
- **Reduce** the data volume: Convert clusters into coarse resolution hits
- **Eliminate** costly loops: Compare hits to pre-stored patterns simultaneously
- **Simplify** algorithms: Use a linearized fit for track candidates
- **Hardware** solution: Implemented in FPGAs or custom ASICs



436 Nuclear Instruments and Methods in Physics Research A278 (1989) 436–440
North-Holland, Amsterdam

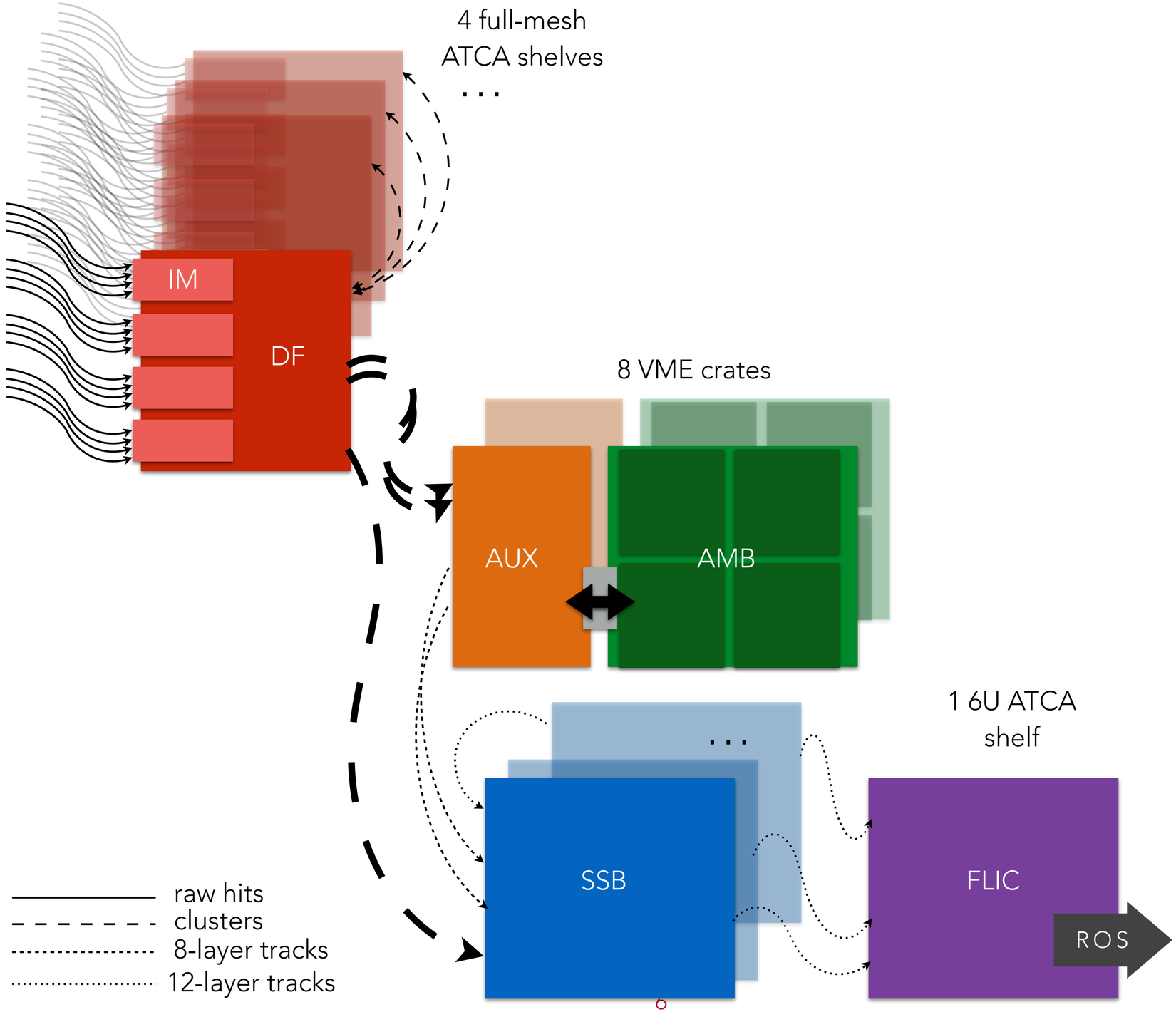
VLSI STRUCTURES FOR TRACK FINDING

Mauro DELL'ORSO
Dipartimento di Fisica, Università di Pisa, Piazza Torricelli 2, 56100 Pisa, Italy

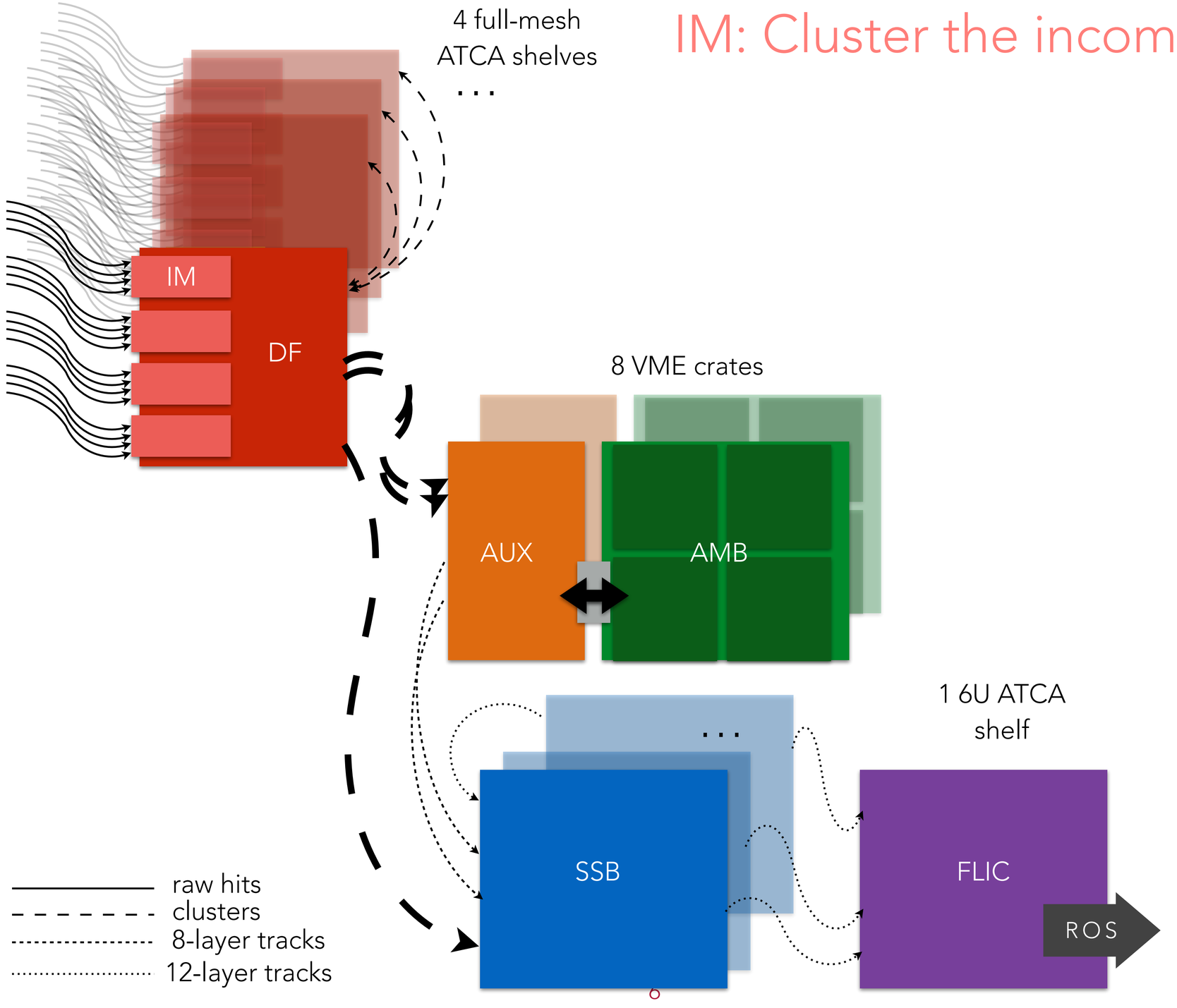
Luciano RISTORI
INFN Sezione di Pisa, Via Vecchia Livornese 582a, 56010 S. Piero a Grado (PI), Italy

Received 24 October 1988

We discuss the architecture of a device based on the concept of *associative memory* designed to solve the track finding problem, typical of high energy physics experiments, in a time span of a few microseconds even for very high multiplicity events. This "machine" is implemented as a large array of custom VLSI chips. All the chips are equal and each of them stores a number of "patterns". All the patterns in all the chips are compared in parallel to the data coming from the detector while the detector is being read out.

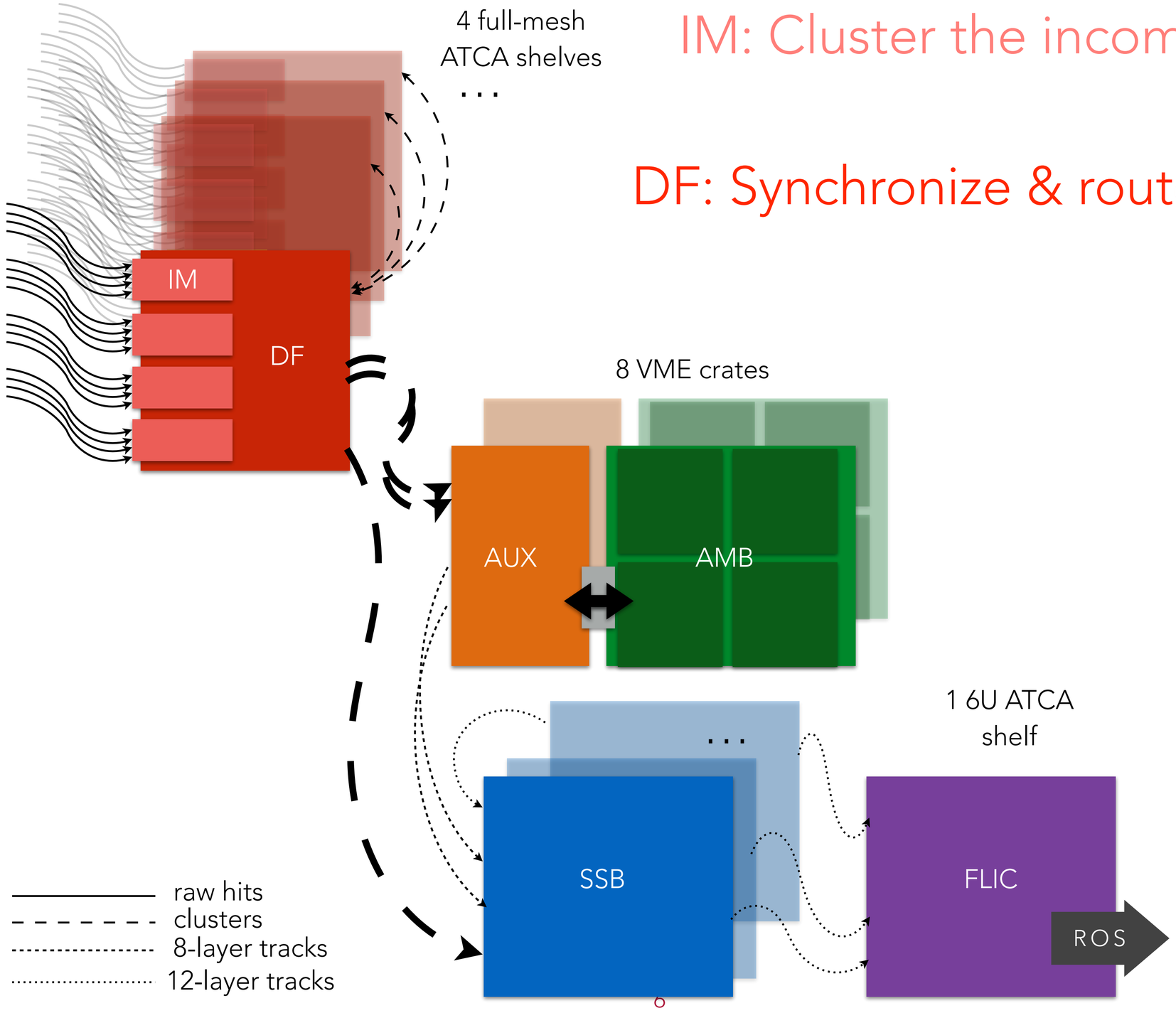

4 full-mesh
ATCA shelves
. . .
IM
DF
8 VME crates
AUX
AMB
1 6U ATCA
shelf
. . .
SSB
FLIC
ROS
raw hits
clusters
8-layer tracks
12-layer tracks

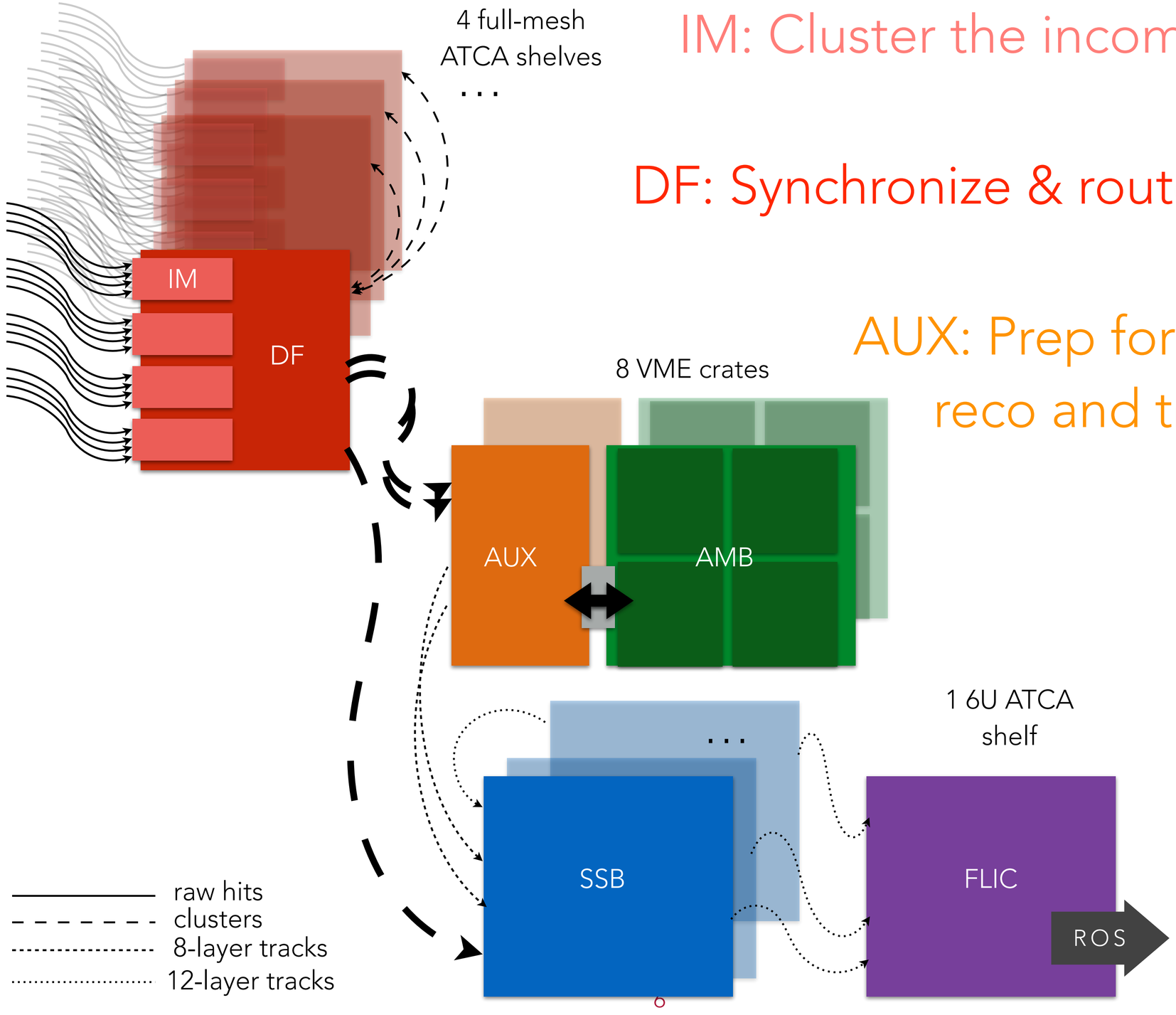
IM: Cluster the incoming data (128)



IM: Cluster the incoming data (128)

DF: Synchronize & route to towers (32)

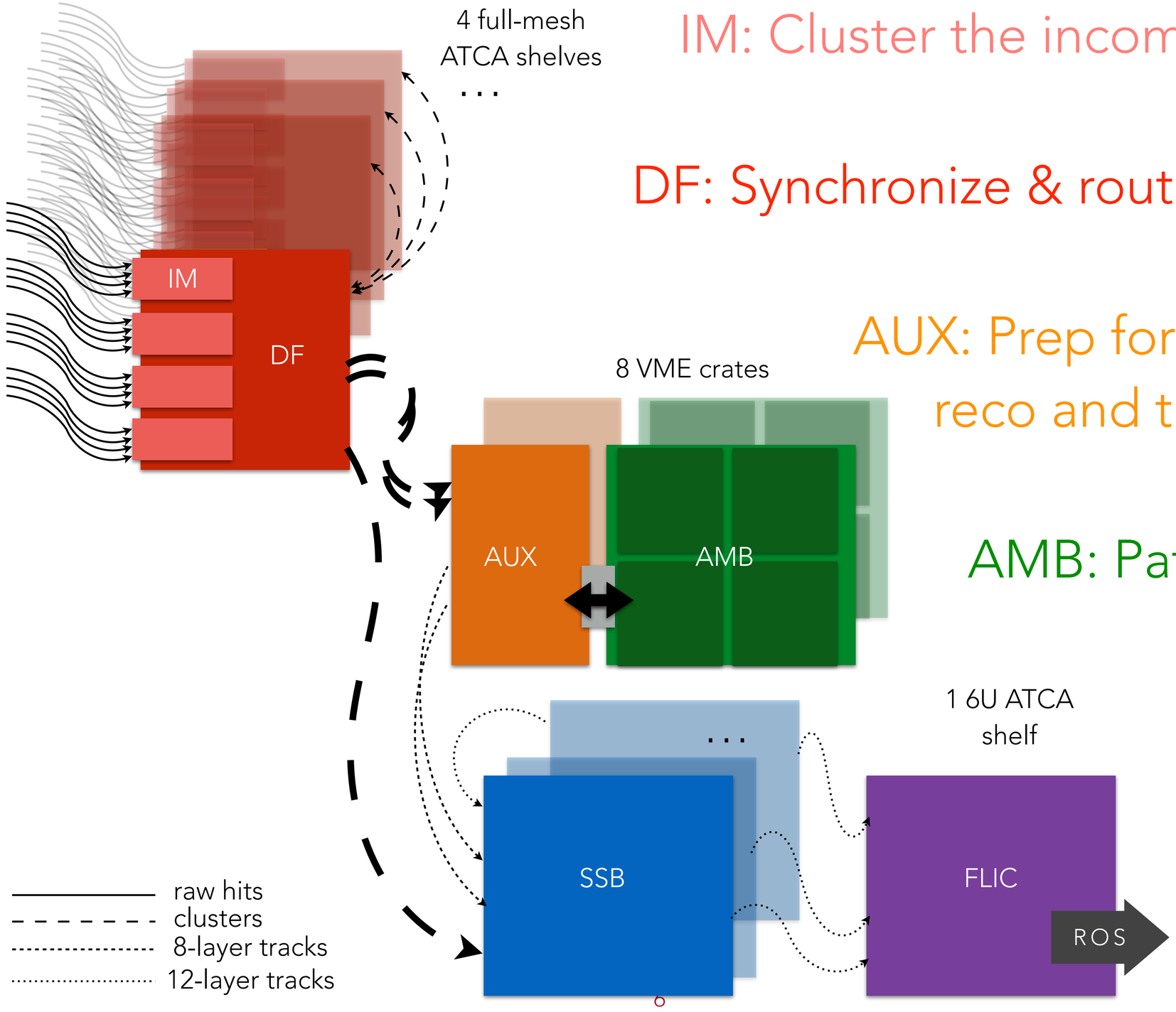




IM: Cluster the incoming data (128)

DF: Synchronize & route to towers (32)

AUX: Prep for 8-layer pattern reco and track fit (128)

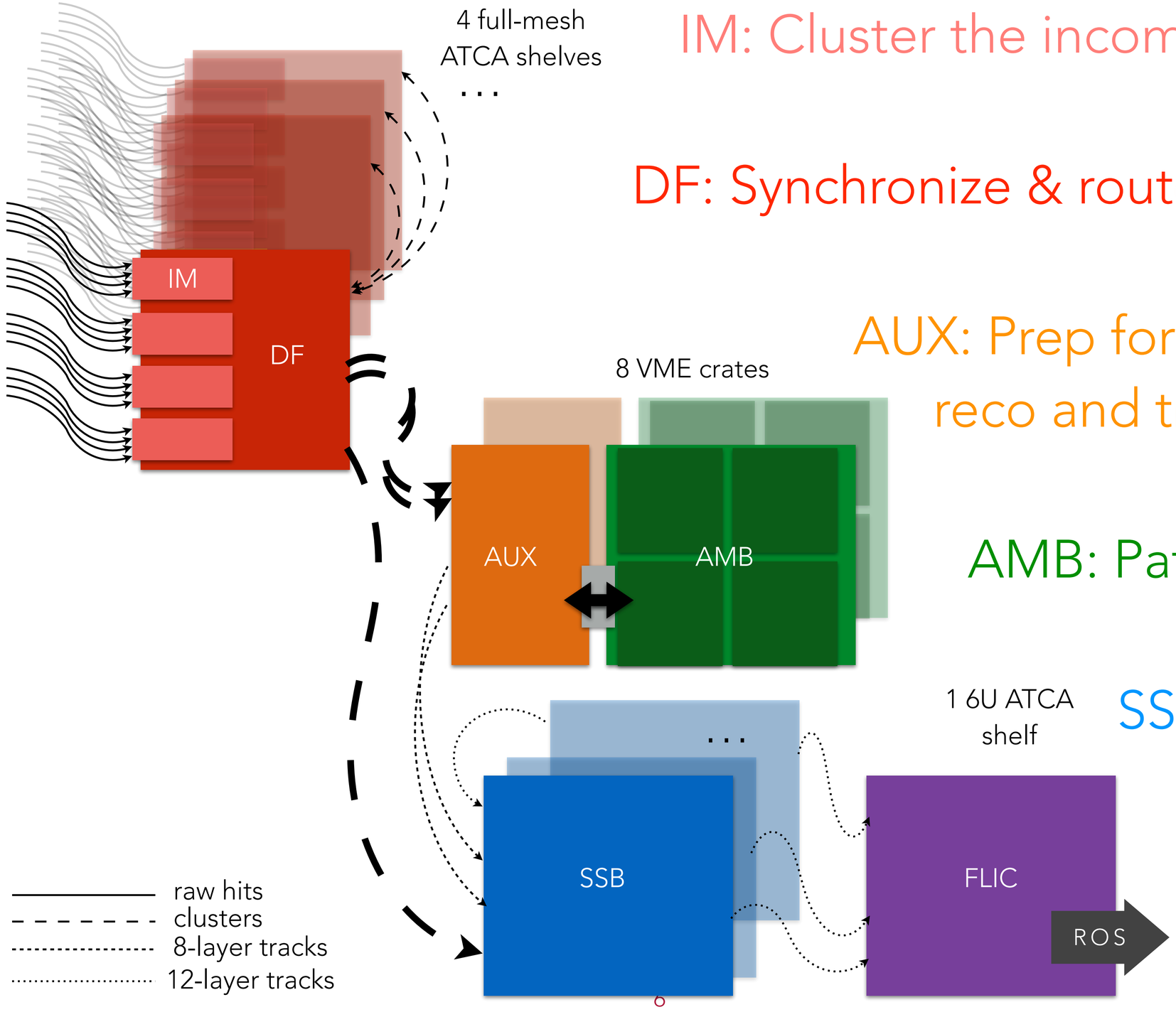


IM: Cluster the incoming data (128)

DF: Synchronize & route to towers (32)

AUX: Prep for 8-layer pattern reco and track fit (128)

AMB: Pattern reco (128)



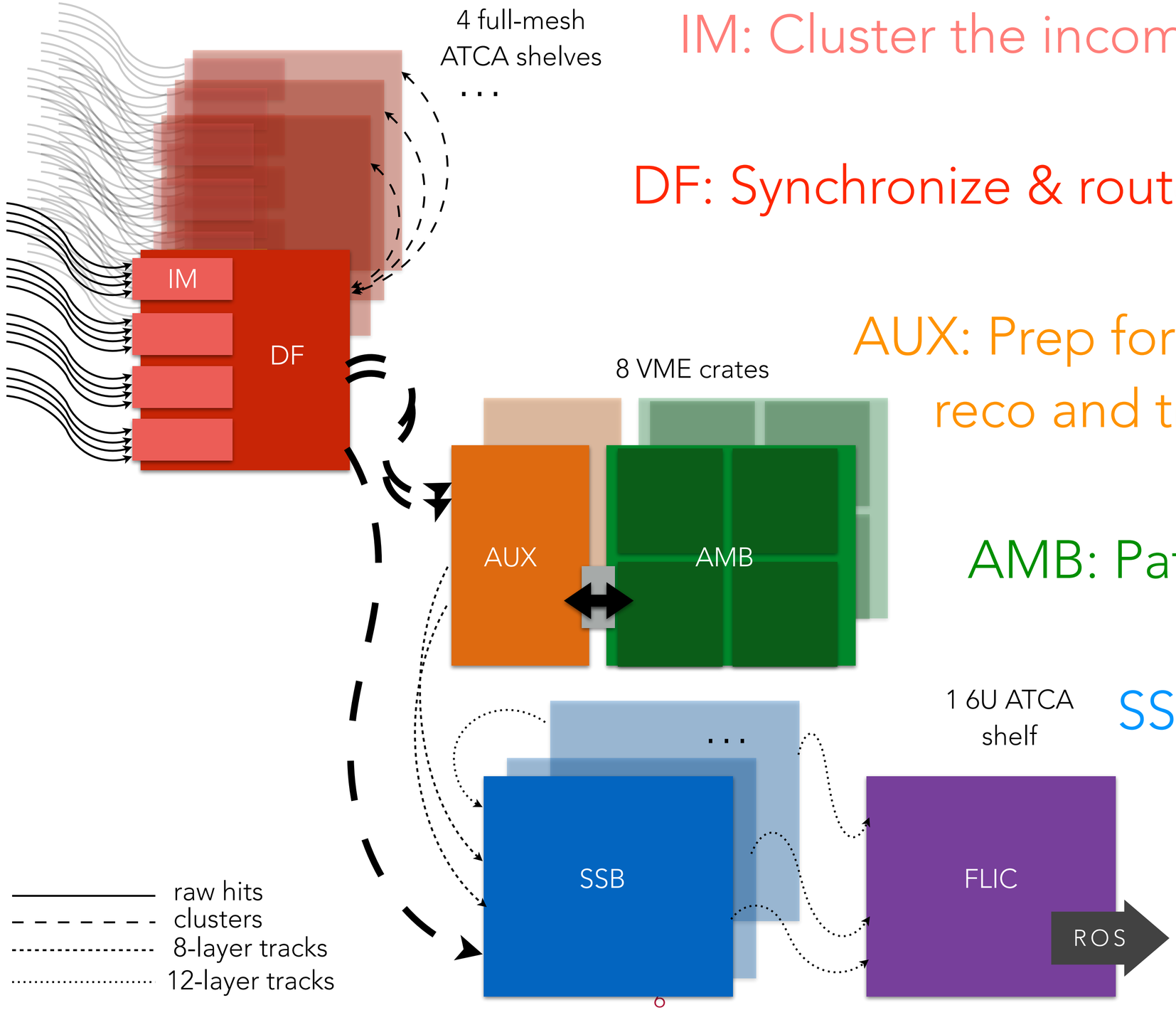
IM: Cluster the incoming data (128)

DF: Synchronize & route to towers (32)

AUX: Prep for 8-layer pattern reco and track fit (128)

AMB: Pattern reco (128)

SSB: 12-layer fit (32)



IM: Cluster the incoming data (128)

DF: Synchronize & route to towers (32)

AUX: Prep for 8-layer pattern reco and track fit (128)

AMB: Pattern reco (128)

SSB: 12-layer fit (32)

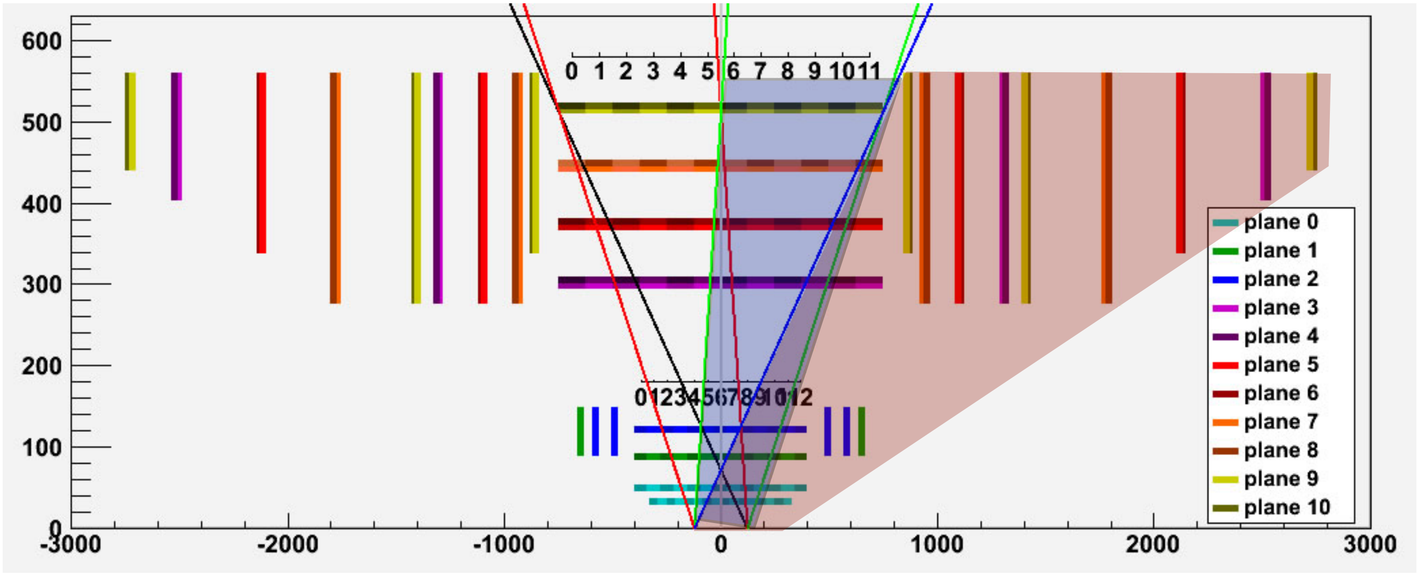
FLIC: Format for HLT (2)

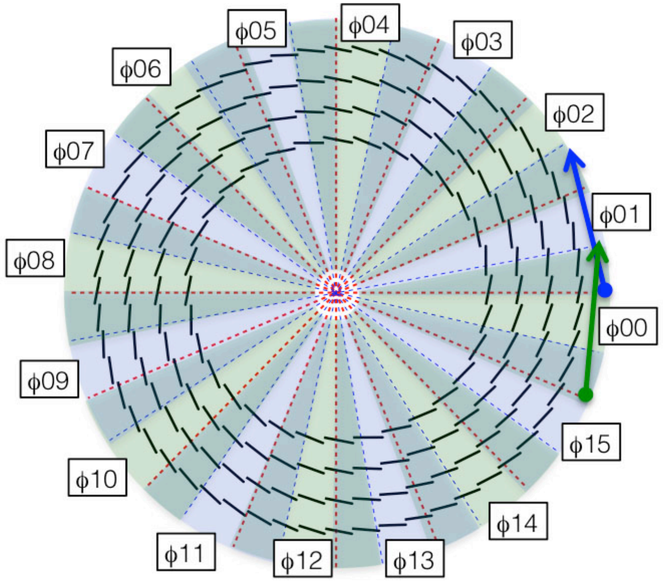
STAGE 1: DATA FORMATTING

- Cluster hits (sliding window algorithm on Spartan 6 and Artix 7 FPGAS)
- Synchronize events
  - Majority logic on input and output
- Route clusters to FTK eta-phi towers
  - High overlap fraction of towers
  - Implemented in Banyan switches

STAGE 1: DATA FORMATTING

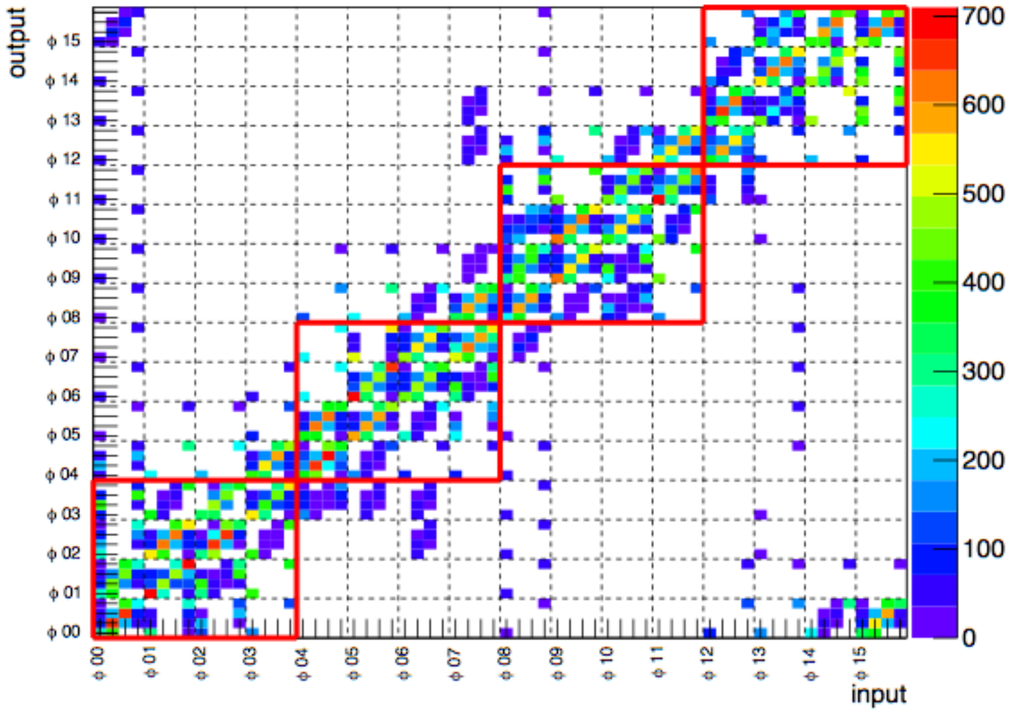
- Cluster hits (sliding window algorithm on Spartan 6 and Artix 7 FPGAS)
- Synchronize events
  - Majority logic on input and output
- Route clusters to FTK eta-phi towers
  - High overlap fraction of towers
  - Implemented in Banyan switches





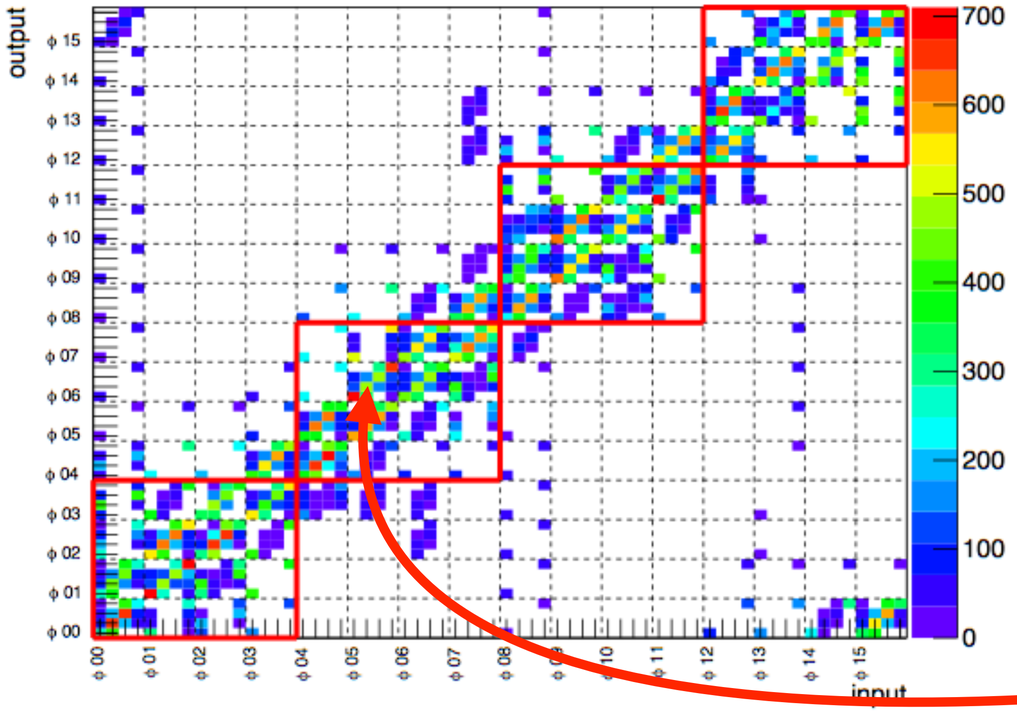
STAGE 1: DATA FORMATTING

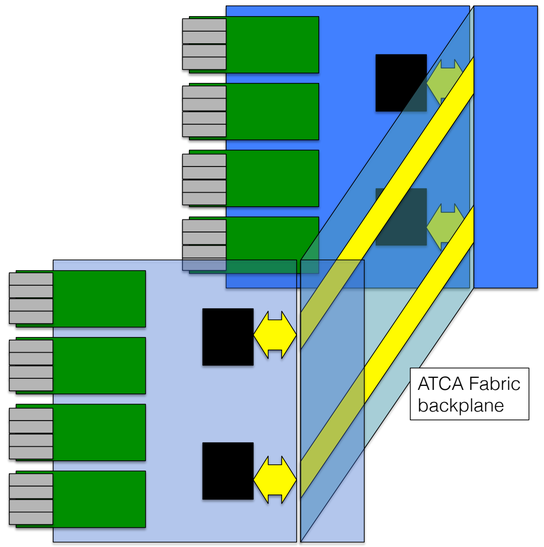
- Implemented in 32 boards in 4 ATCA crates with full mesh backplane (40Gbps), Virtex 7 FPGA on each board
- Data aggregation challenges:
  - Silicon detector readout not projective!
  - System had no L1 decision info as input — had to decode from input data



STAGE 1: DATA FORMATTING

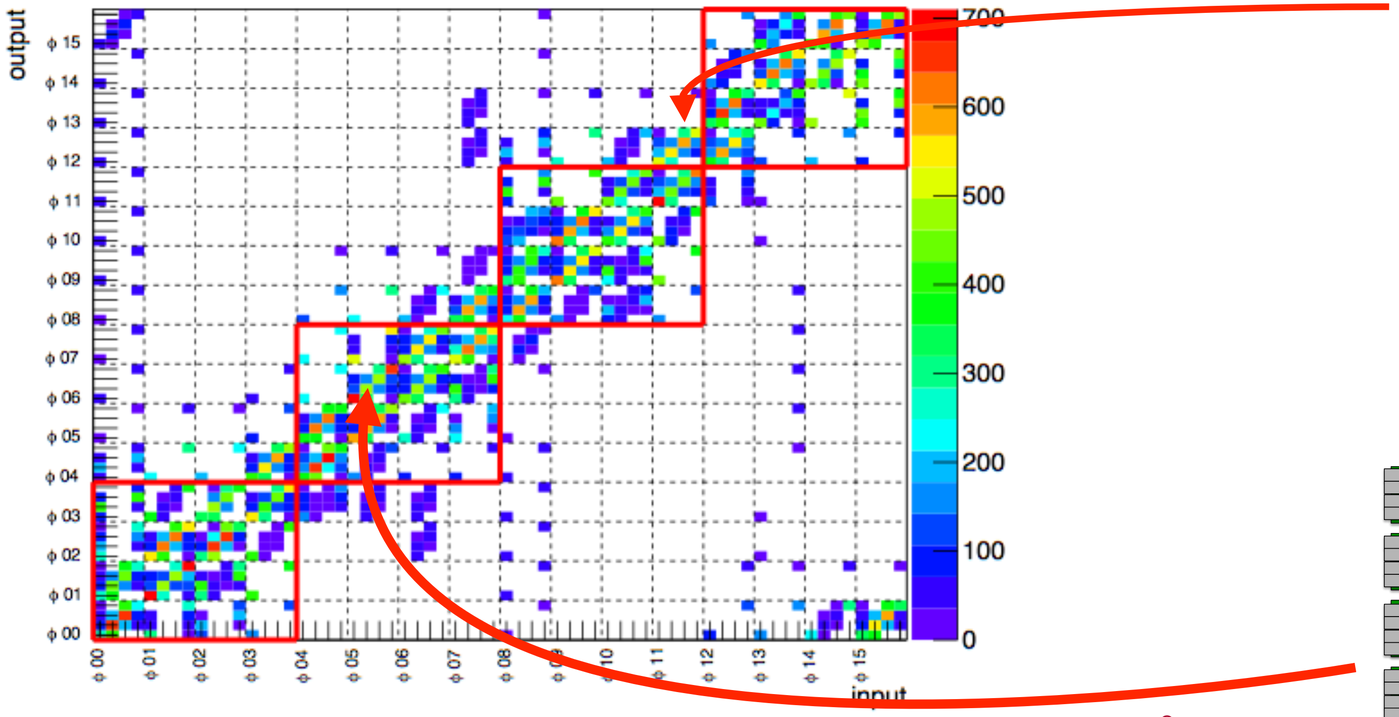
- Implemented in 32 boards in 4 ATCA crates with full mesh backplane (40Gbps), Virtex 7 FPGA on each board
- Data aggregation challenges:
  - Silicon detector readout not projective!
  - System had no L1 decision info as input — had to decode from input data

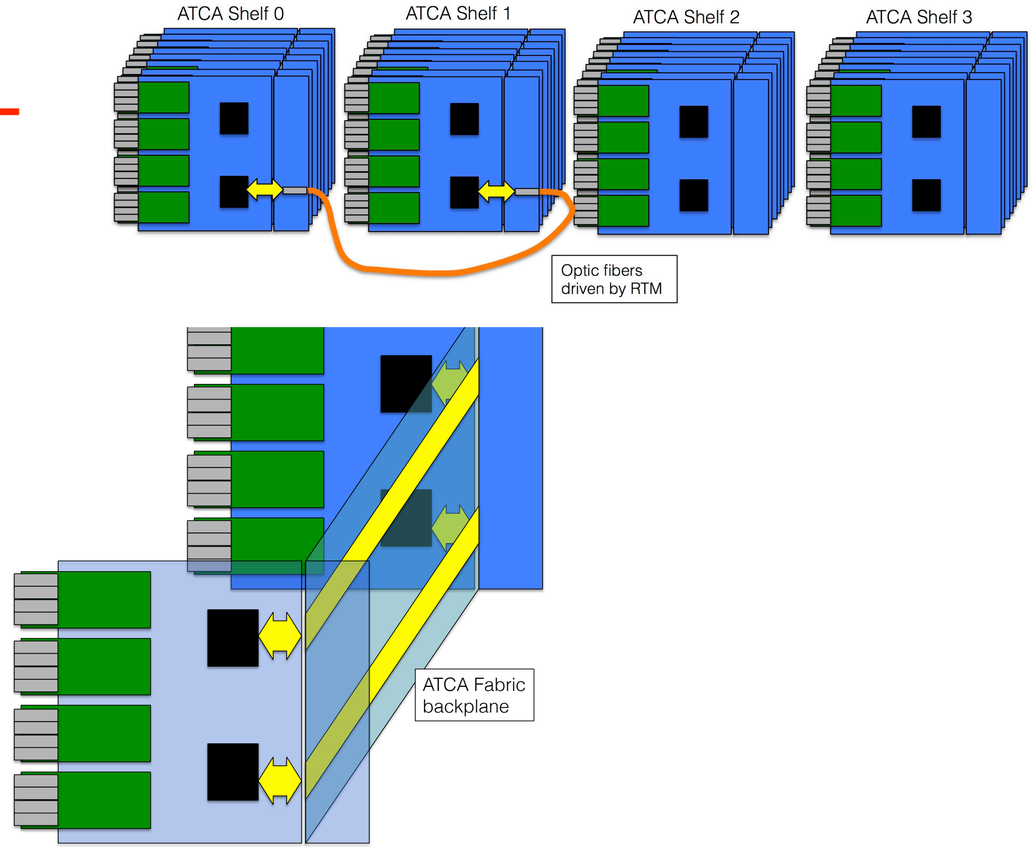




STAGE 1: DATA FORMATTING

- Implemented in 32 boards in 4 ATCA crates with full mesh backplane (40Gbps), Virtex 7 FPGA on each board
- Data aggregation challenges:
  - Silicon detector readout not projective!
  - System had no L1 decision info as input — had to decode from input data

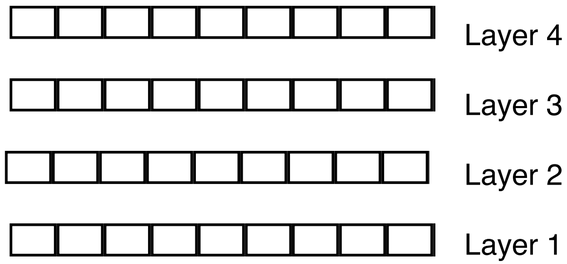




STAGE 2: BINGO
DATA REDUCTION AND PATTERN RECOGNITION

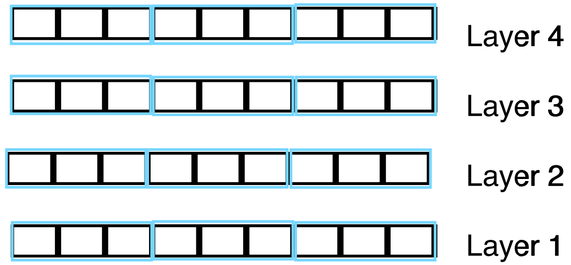
- Hits are ganged together into coarse resolution hits

STAGE 2: BINGO
DATA REDUCTION AND PATTERN RECOGNITION



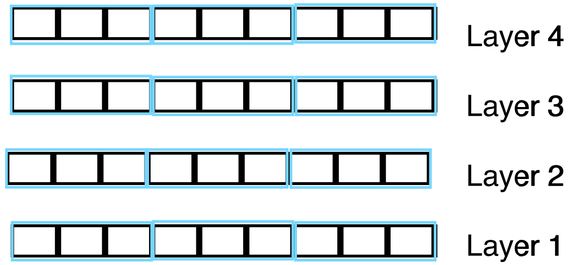
- Hits are ganged together into coarse resolution hits

STAGE 2: BINGO
DATA REDUCTION AND PATTERN RECOGNITION



- Hits are ganged together into coarse resolution hits

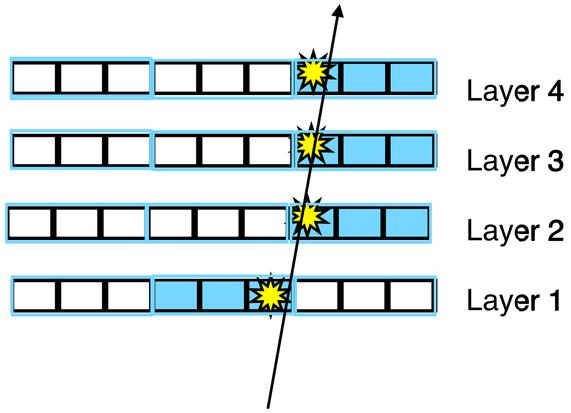
STAGE 2: BINGO
DATA REDUCTION AND PATTERN RECOGNITION



- Hits are ganged together into coarse resolution hits

- All possible patterns of coarse resolution hits determined from simulation

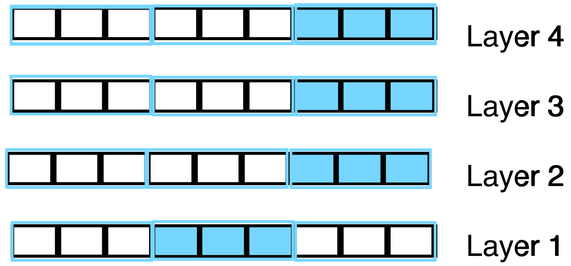
STAGE 2: BINGO
DATA REDUCTION AND PATTERN RECOGNITION



- Hits are ganged together into coarse resolution hits

- All possible patterns of coarse resolution hits determined from simulation

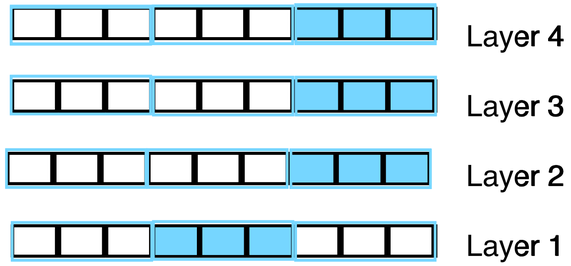
STAGE 2: BINGO
DATA REDUCTION AND PATTERN RECOGNITION



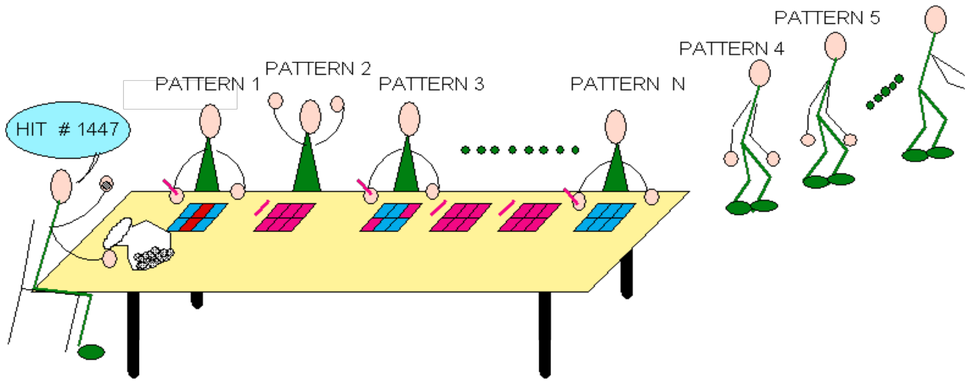
- Hits are ganged together into coarse resolution hits

- All possible patterns of coarse resolution hits determined from simulation

STAGE 2: BINGO
DATA REDUCTION AND PATTERN RECOGNITION



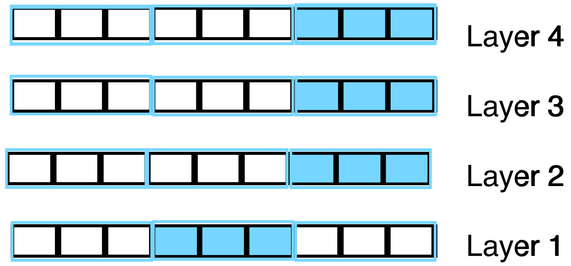
- Hits are ganged together into coarse resolution hits



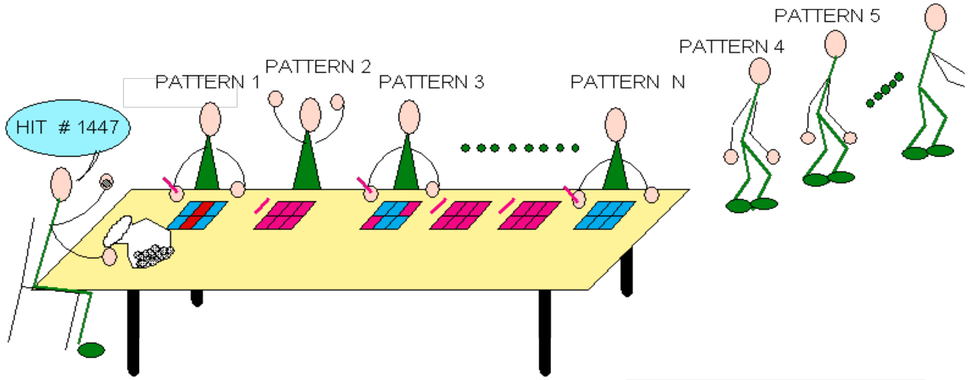
- All possible patterns of coarse resolution hits determined from simulation

- Custom associative memory chips are used to **compare hits** to O(10^9) patterns **simultaneously** (bingo cards)

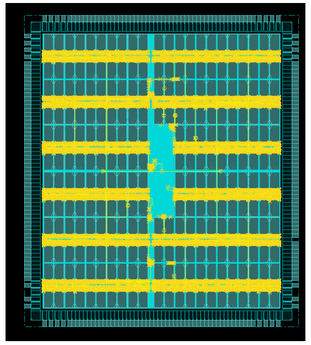
STAGE 2: BINGO
DATA REDUCTION AND PATTERN RECOGNITION



- Hits are ganged together into coarse resolution hits



- All possible patterns of coarse resolution hits determined from simulation

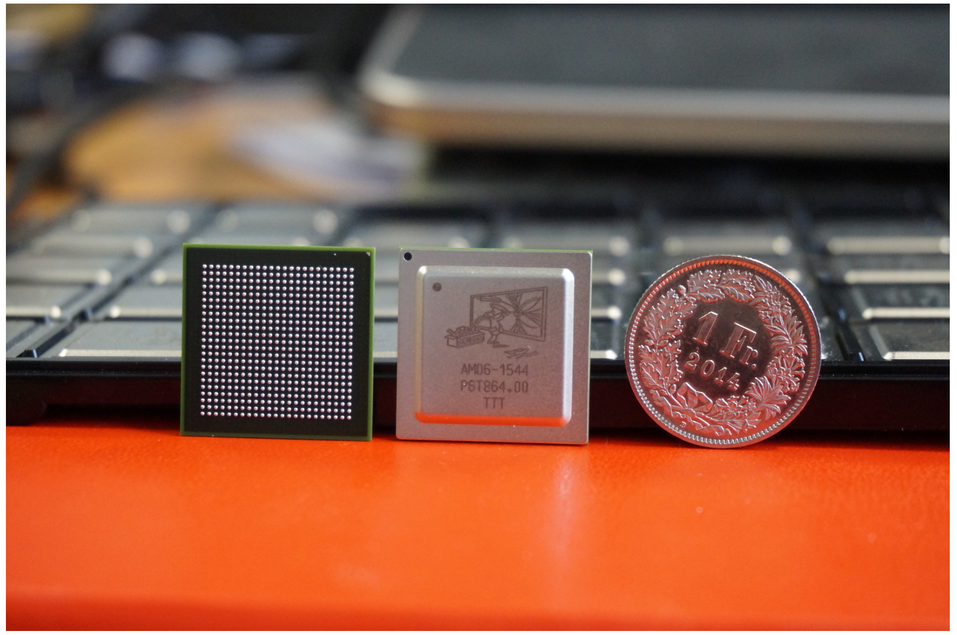


AMChip06

- Custom associative memory chips are used to **compare hits** to O(10^9) patterns **simultaneously** (bingo cards)

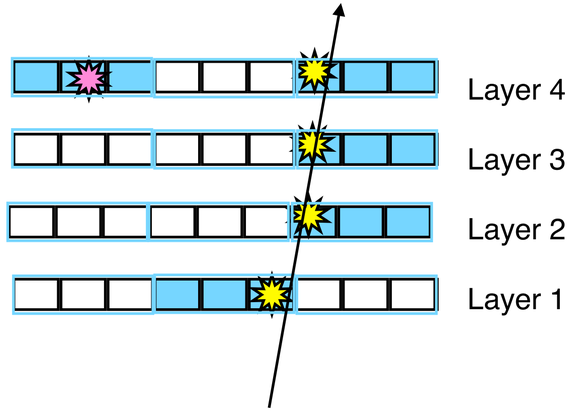
AMCHIP 06 VITAL STATS

- 65 nm fabrication
- 60mm^2 area
- 2 Gb/s I/O
- 23x23 BGA
- 128k patterns, 400M transistors
- 1.15V, 3.3 A, 3.8 W / chip
- ~ 85% production yield

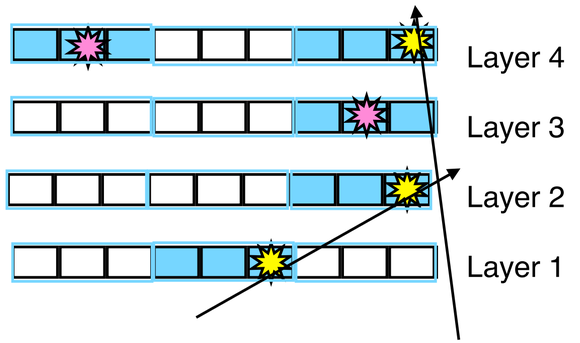


TRACK FITTING

- Problem: >90% of matched patterns (BINGOs) are from random association of hits

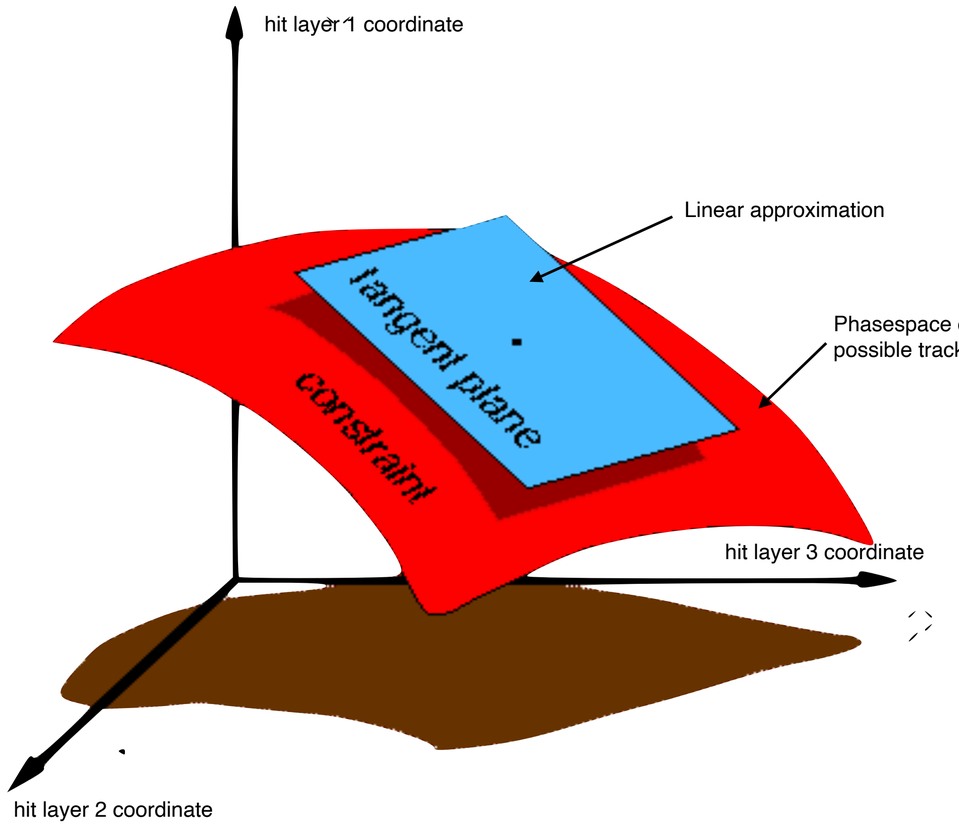


- Solution: check if **full resolution** hits in matched patterns are compatible with a single charged particle



5 PICOSECOND TRACK FITTING

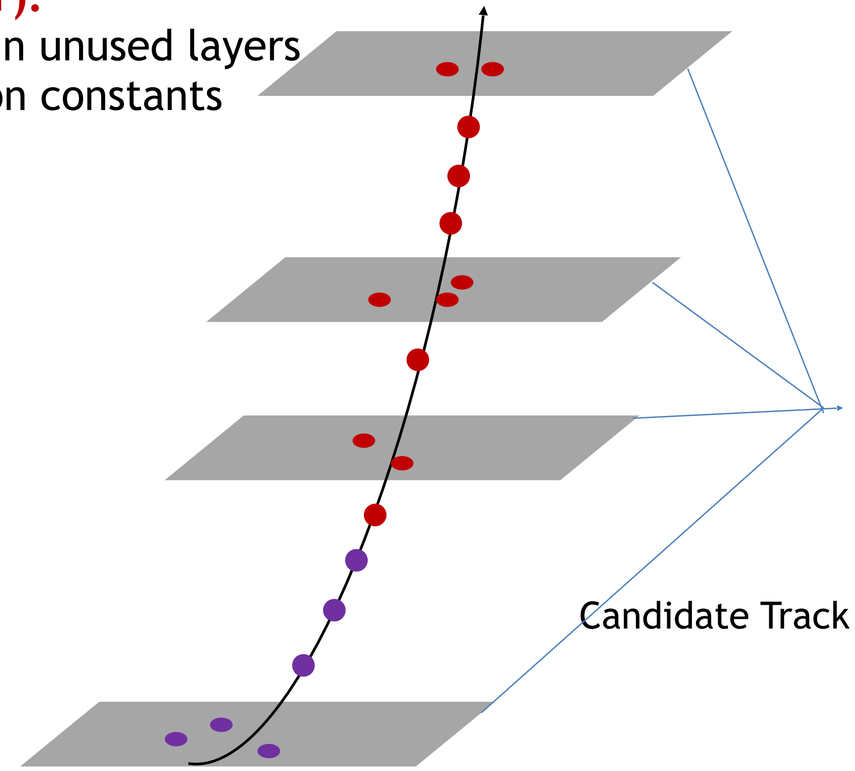
- Linearized fits on FPGAs:
  - Determine phasespace of possible tracks (χ^2)
  - Linear approximation calculated and defined by sector
  - FPGAs multiply and add coordinates by constants to get χ^2
- Keep roads with at least 1 good track
- Fit 1 track / ns (1 track every 5 ps for full system)!



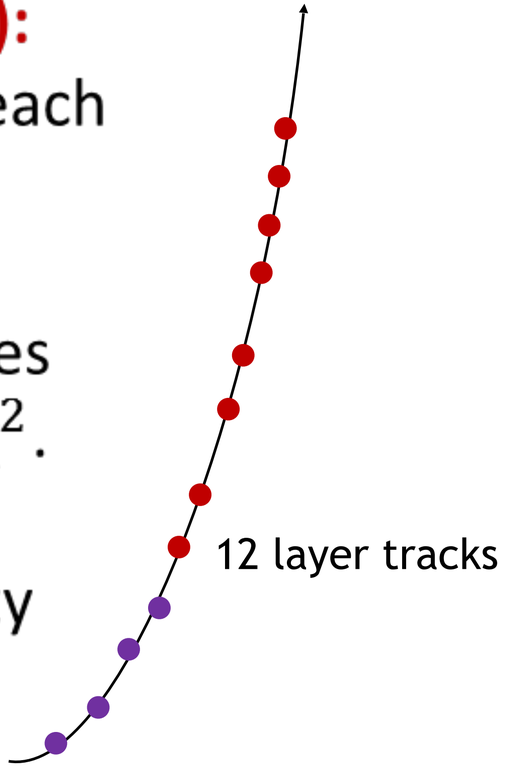
$$\chi_i = \sum_{j=1}^{N_c} S_{ij} x_j + h_i; i = 1, \ldots, N_\chi$$

FINAL TRACK PRODUCTION

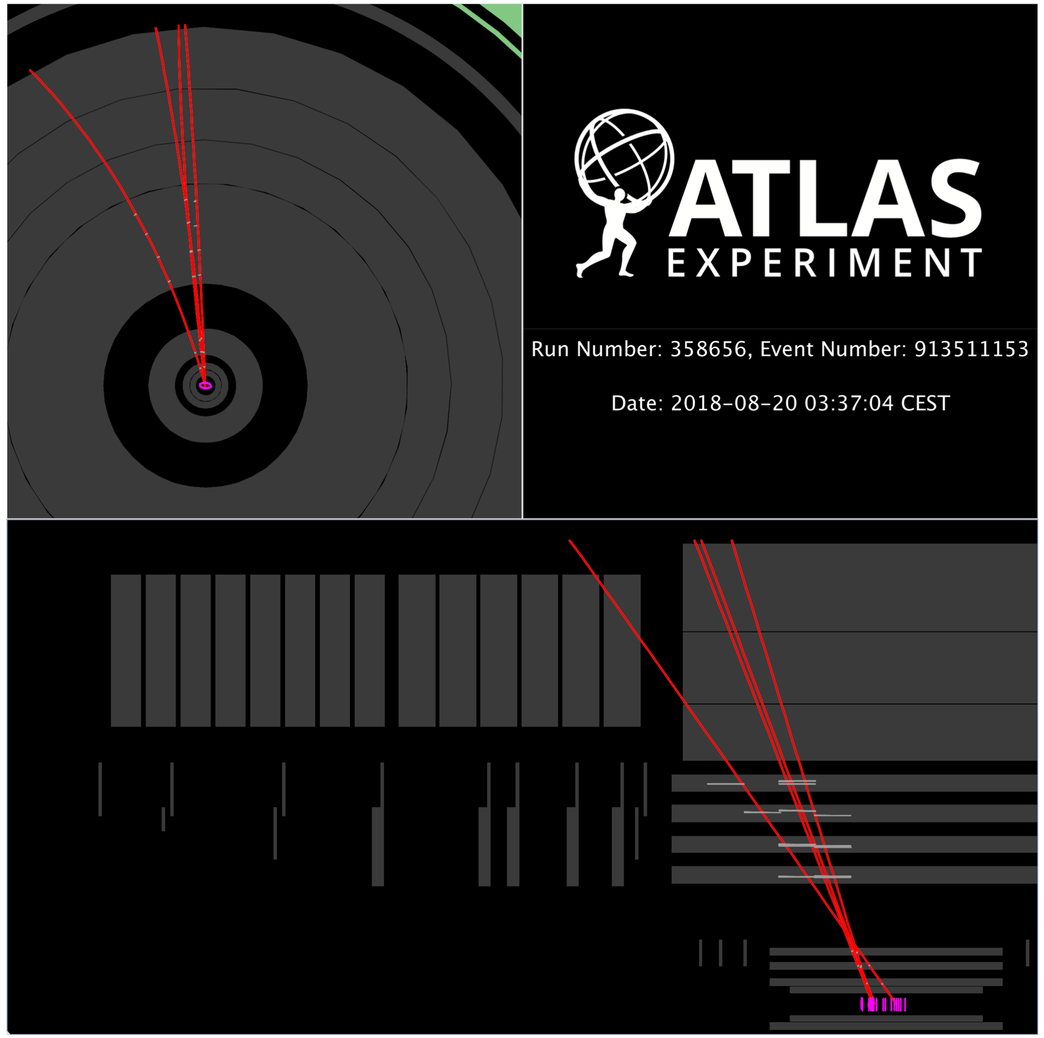
Extrapolator (EXT):
Finds candidates in unused layers using extrapolation constants

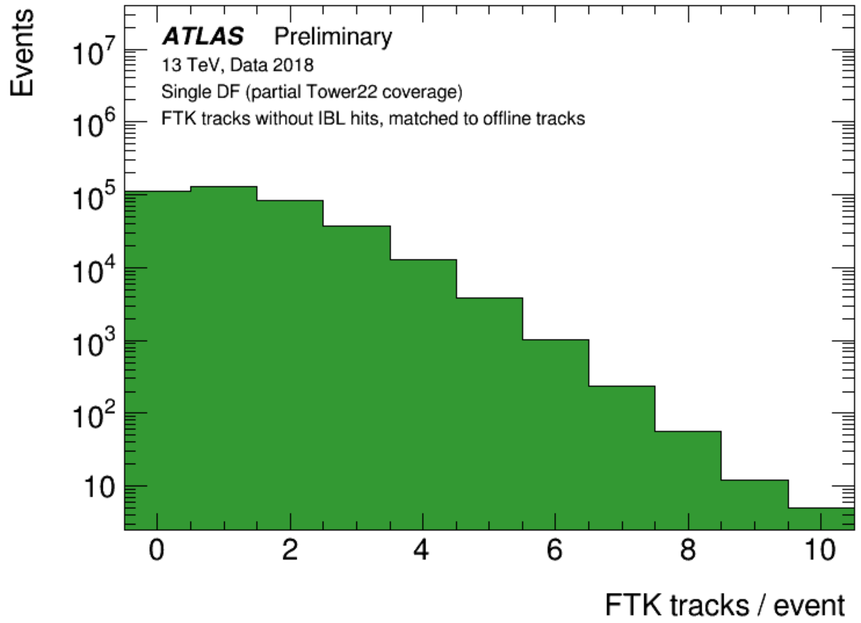


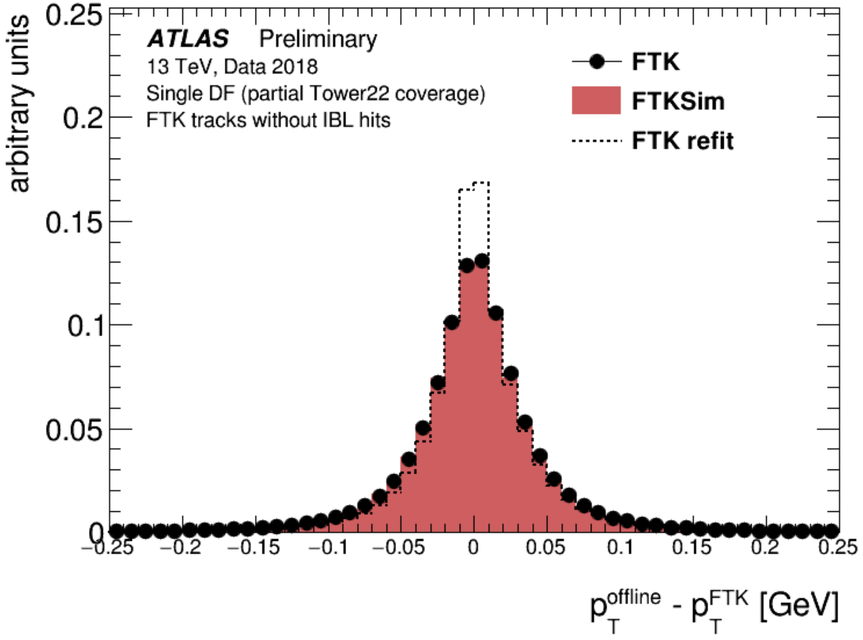
Track Fitter (TF):
Fits track with each combination of possible candidates, takes one with best χ^2. (Nominal and Pix/SCT Majority Fitters)

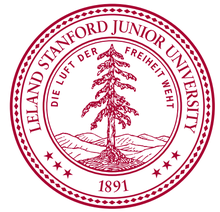


SLICE DATA IN 2018





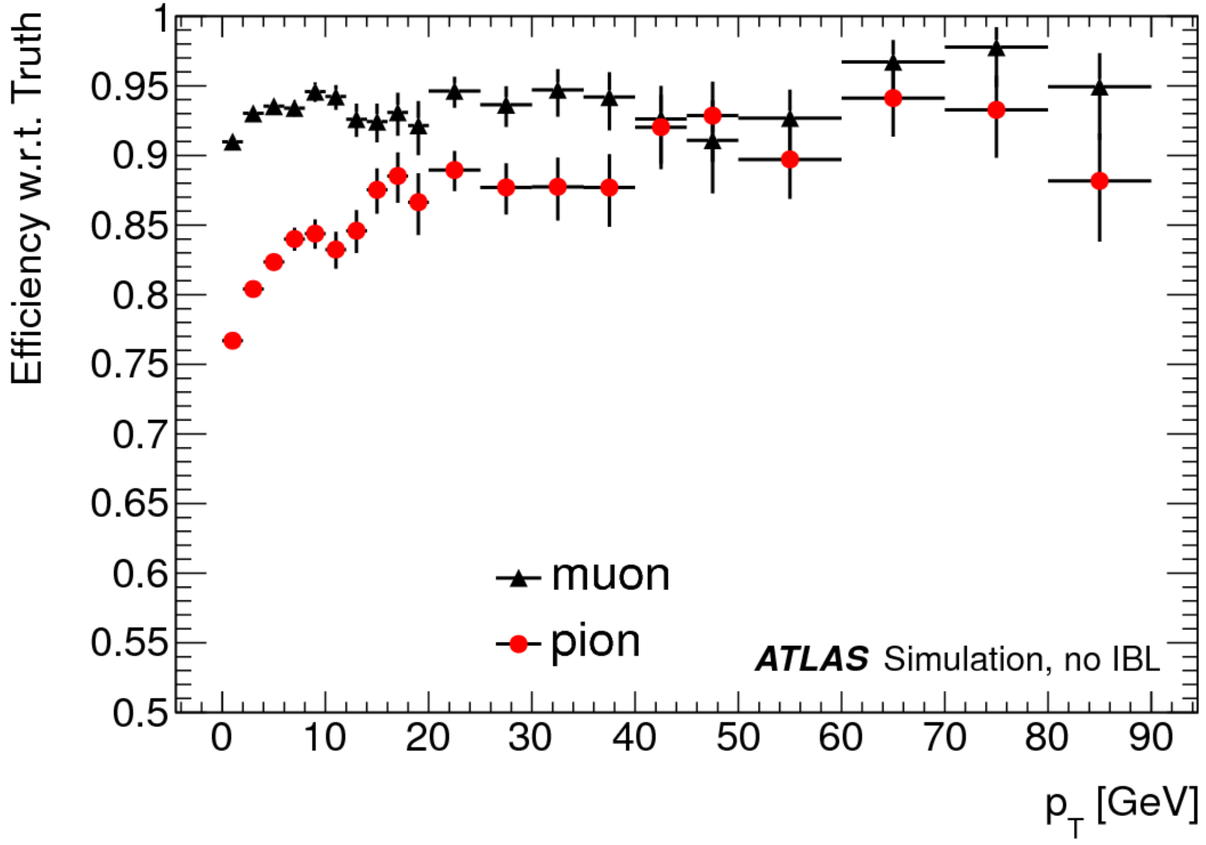




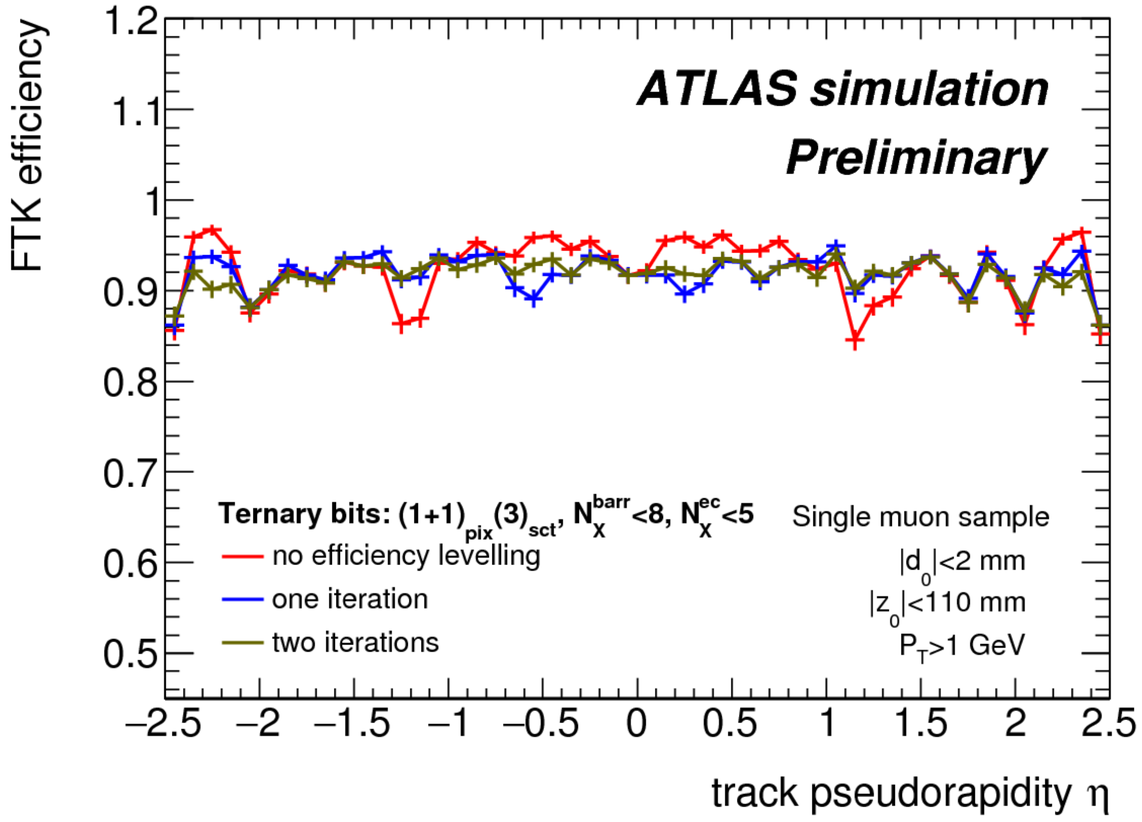
FTK AS A SYSTEM

PERFORMANCE

EXPECTED PERFORMANCE



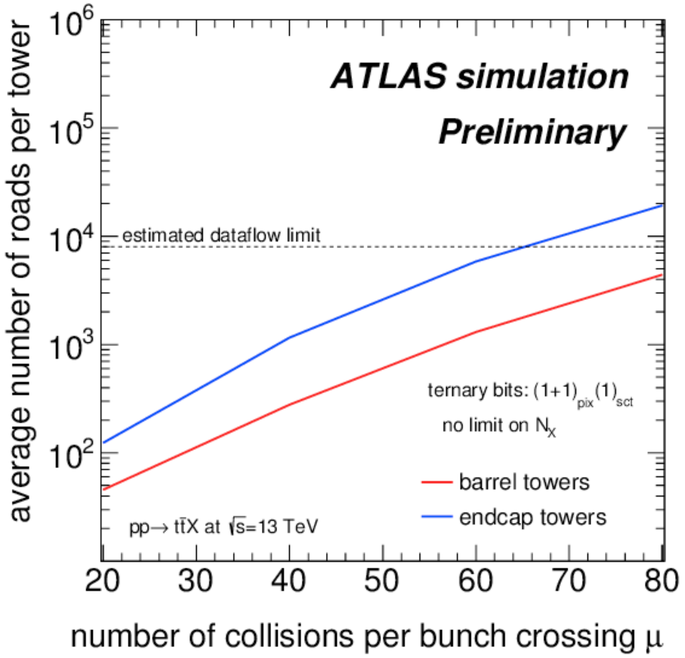
TDR, Optimistic

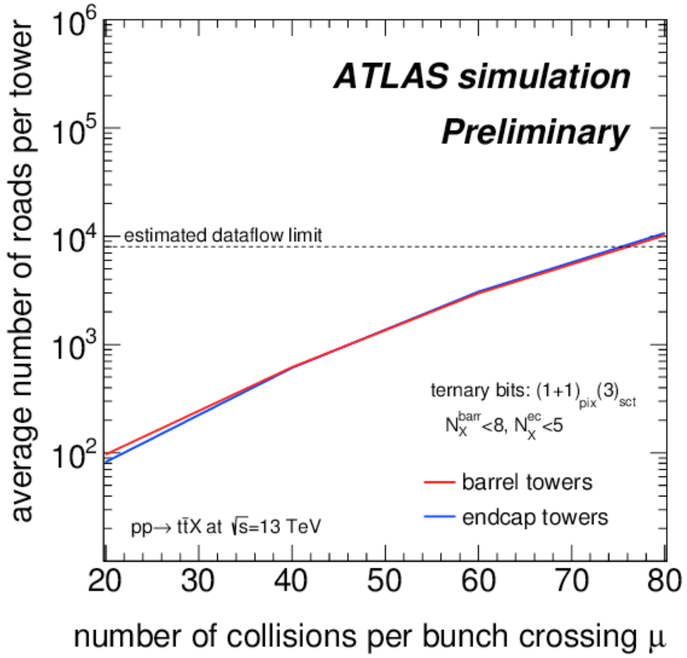


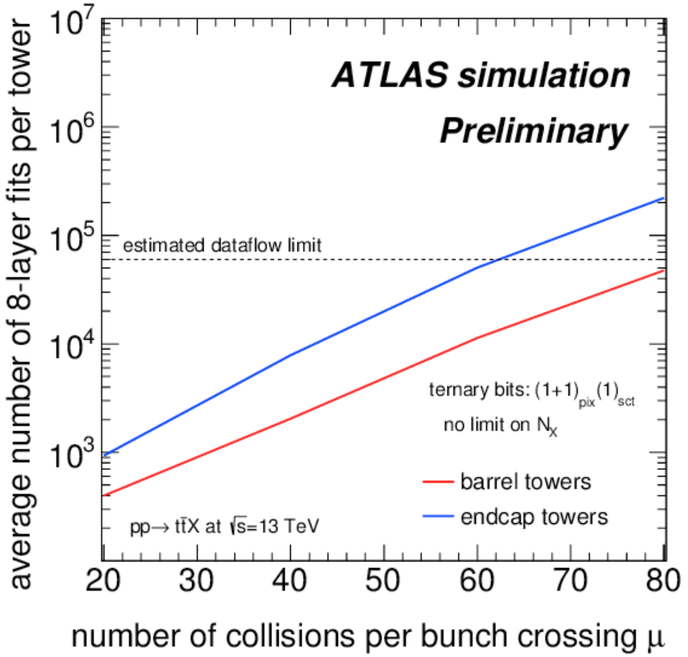
More realistic

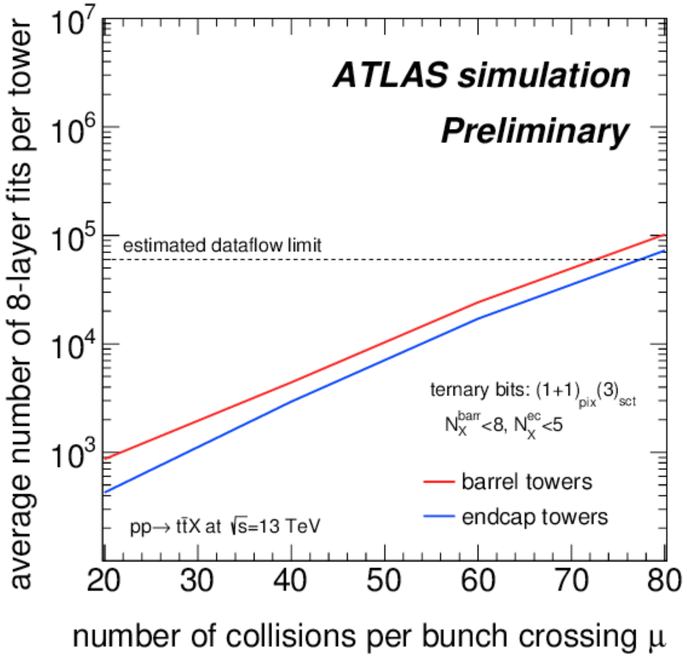
DATAFLOW

- Challenge was to have pattern banks efficient (~90%) but giving reasonable data flow
  - Many configuration handles:
    - Pattern size
    - Pattern variability



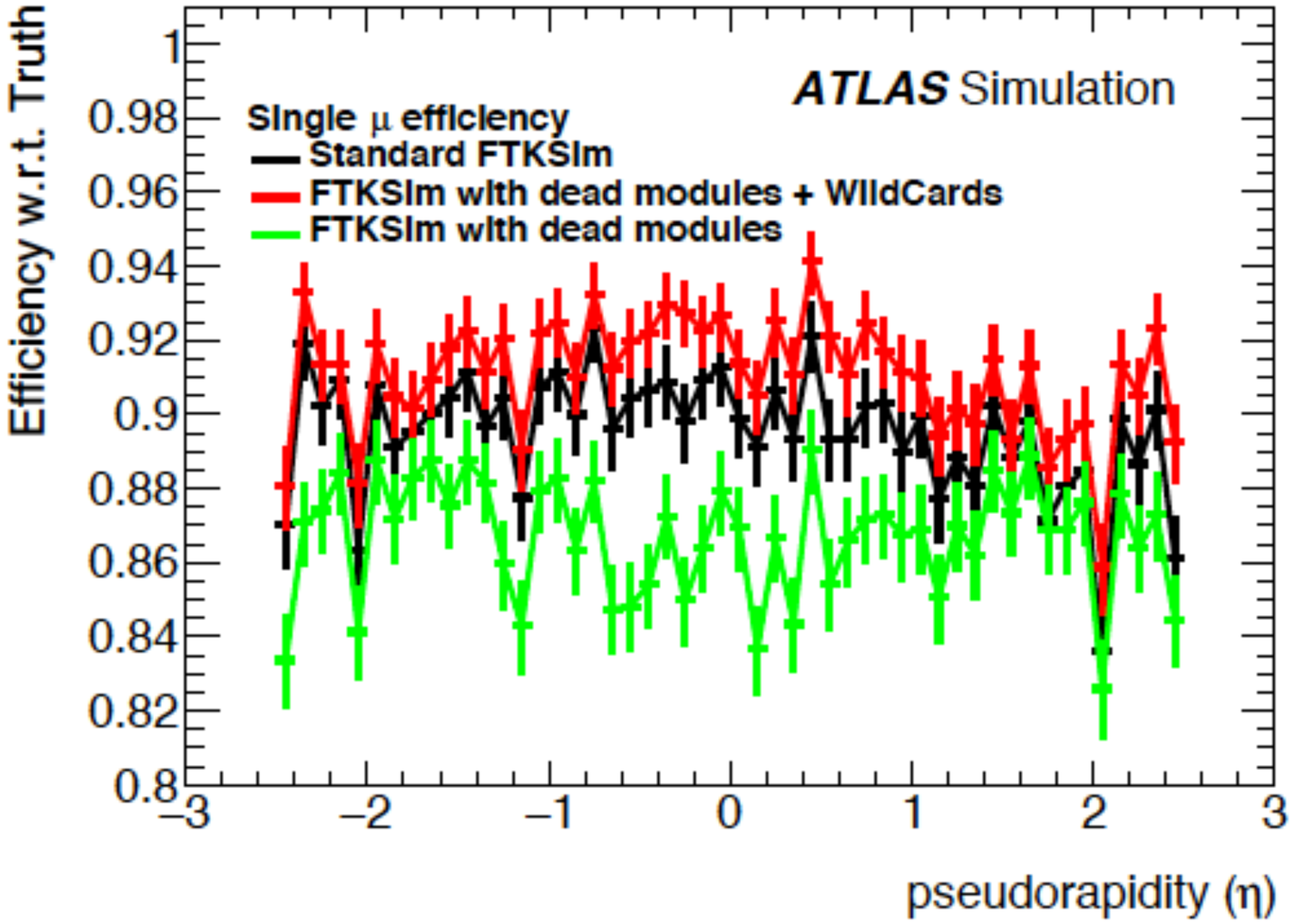






DETECTOR PERFORMANCE

- Detector performance strongly effects efficiency
  - Use wildcards to address this
    - Cost of higher dataflow



FTK AS A SYSTEM

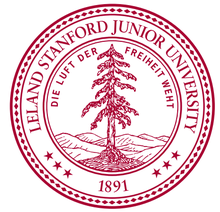
LESSONS LEARNED

THE HIGHLIGHTS

- Heterogeneous systems are difficult to integrate and debug:
  - Mixed ATCA/VME ; mixed Xilinx/Altera — good reasons at the time but made life more challenging
- Resource usage estimates are hard without firmware in hand — nearly every FPGA was full
- Data push architecture with no external synchronization source is difficult in real (buggy) conditions
- Need significant engineering involvement of firmware writing
  - HLS was not mature when much of FTK development was happening (FW development started in ~2012)
- Early emphasis on interfaces/unit tests/CI&CD/simulation test benches would've streamlined integration and development
- Monitoring FW and online software is critical for debugging in situ
- Dedicated person-power is critical!

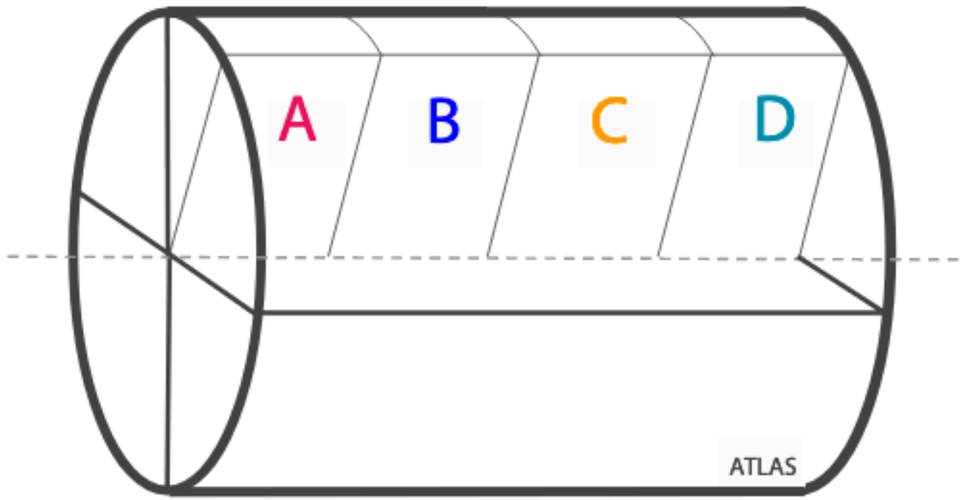
REFERENCES

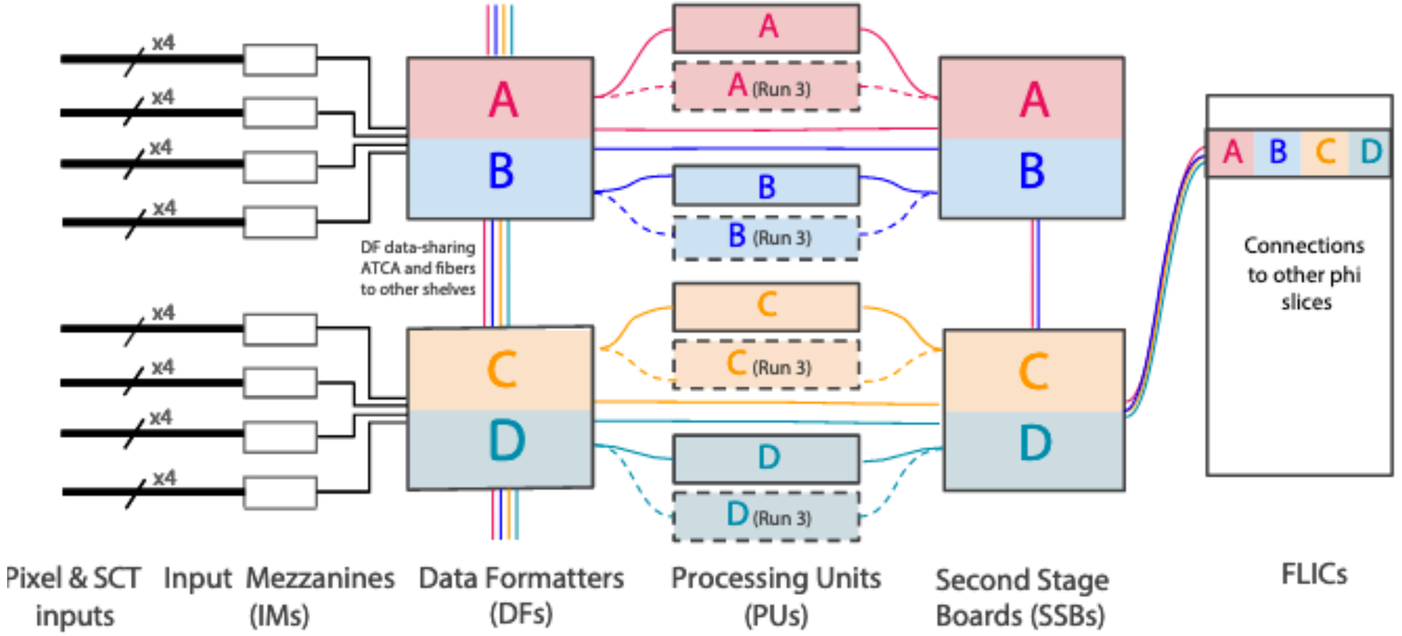
- FTK Technical Proposal and FTK Technical Design Report
- Original AMChip paper ; SVT Gigafitter reference
- FTK Sim proceedings
- FTK NIM paper on the arXiv soon! (next few weeks)


LELAND STANFORD JUNIOR UNIVERSITY
DIE LUFT DER FREIHEIT WEHT
1891

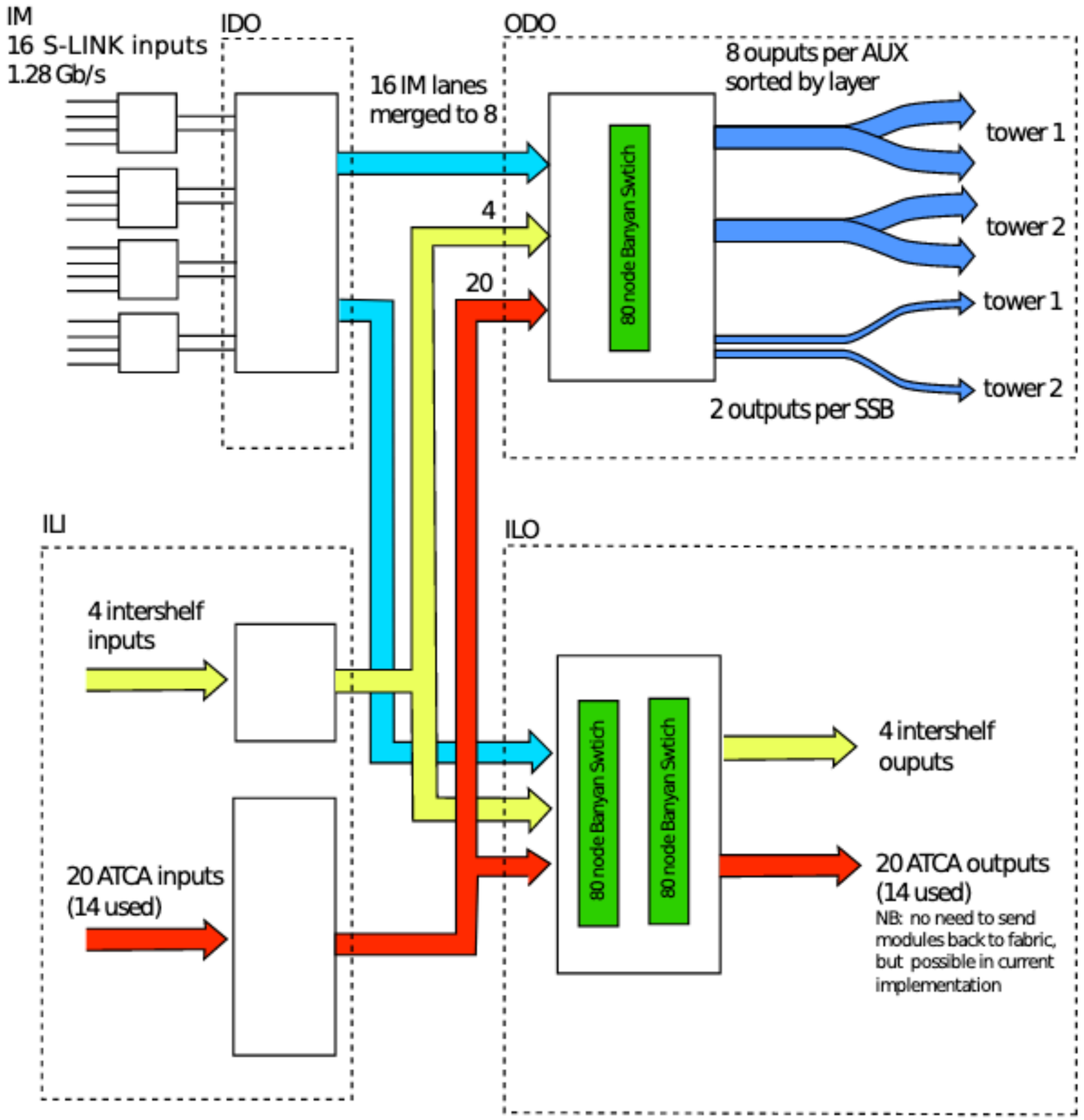
THE END

TOWER GEOMETRY

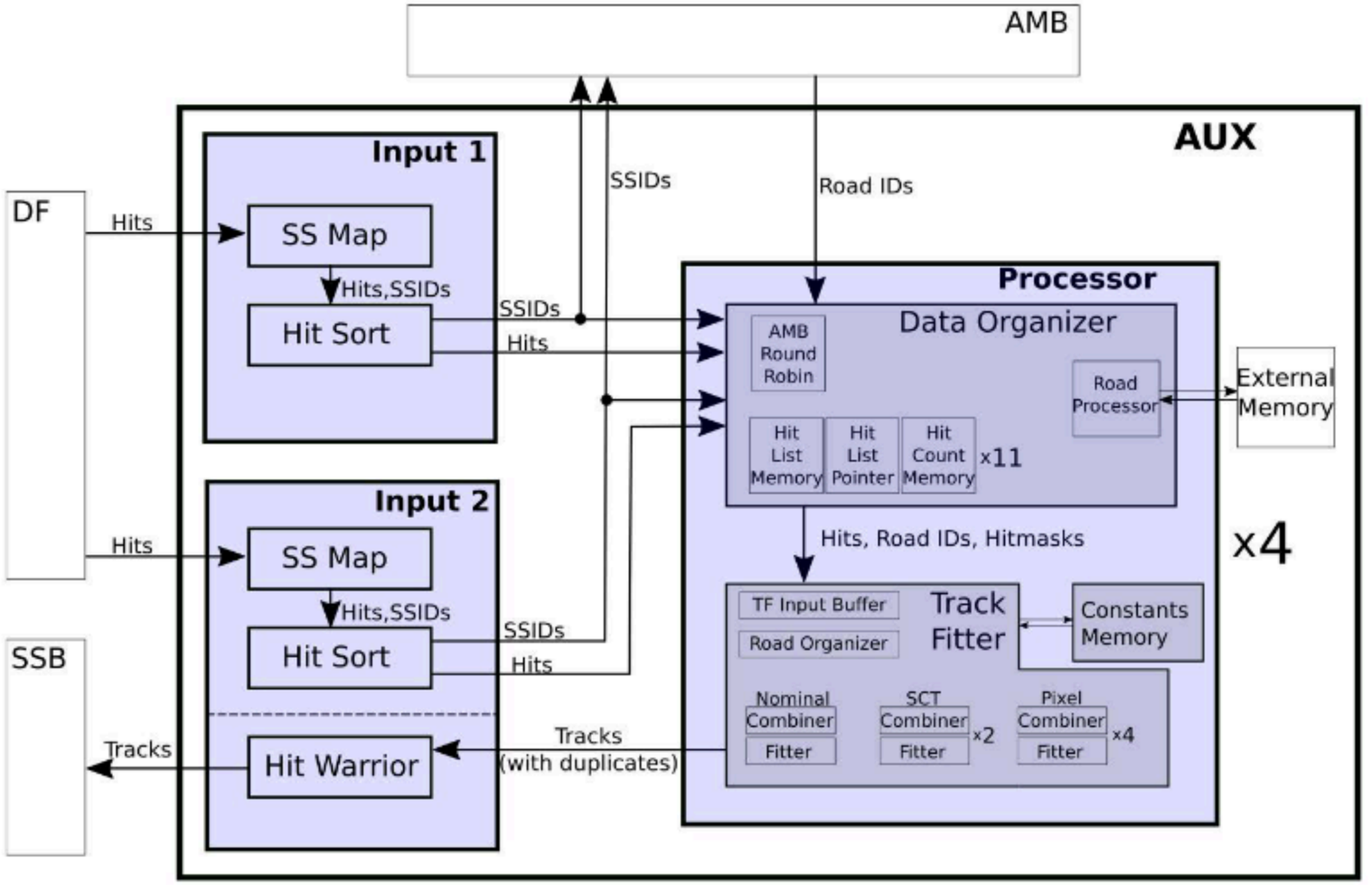




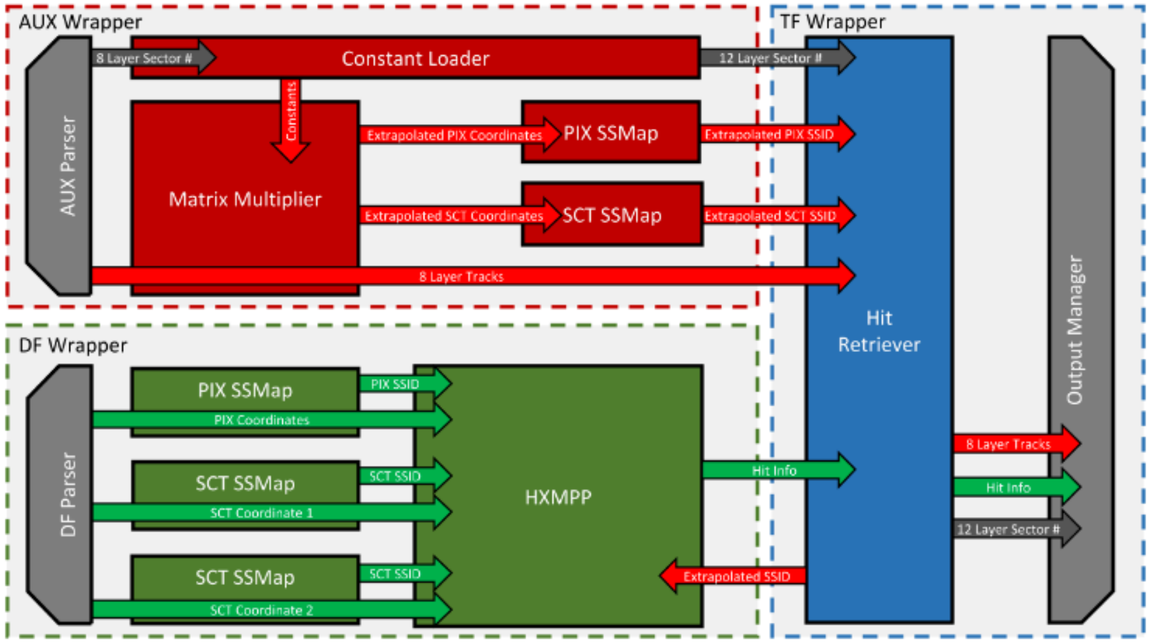
DATA FORMATTER LOGIC

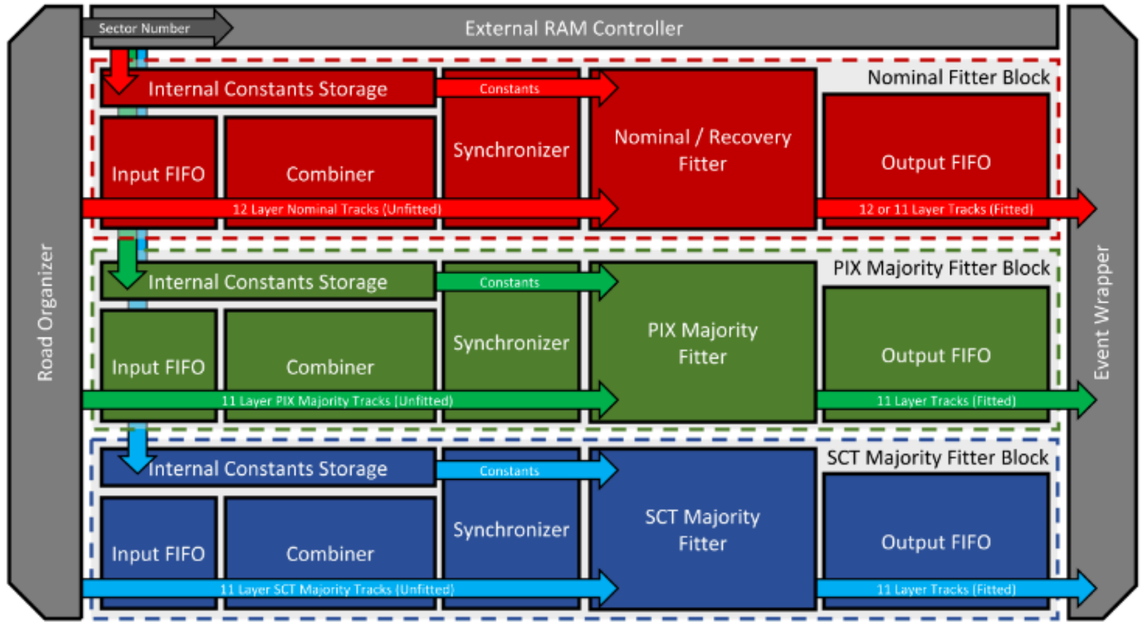


AUX LOGIC


AMB
AUX
Input 1
SS Map
Hits,SSIDs
Hit Sort
DF
Hits
SSIDs
Road IDs
Processor
Data Organizer
AMB Round Robin
Road Processor
External Memory
Hit List Memory
Hit List Pointer
Hit Count Memory
x11
Hits, Road IDs, Hitmasks
x4
Input 2
SS Map
Hits,SSIDs
Hit Sort
SSB
Hits
SSIDs
TF Input Buffer
Road Organizer
Track Fitter
Constants Memory
Nominal Combiner
Fitter
SCT Combiner
x2
Pixel Combiner
x4
Tracks
Hit Warrior
Tracks (with duplicates)

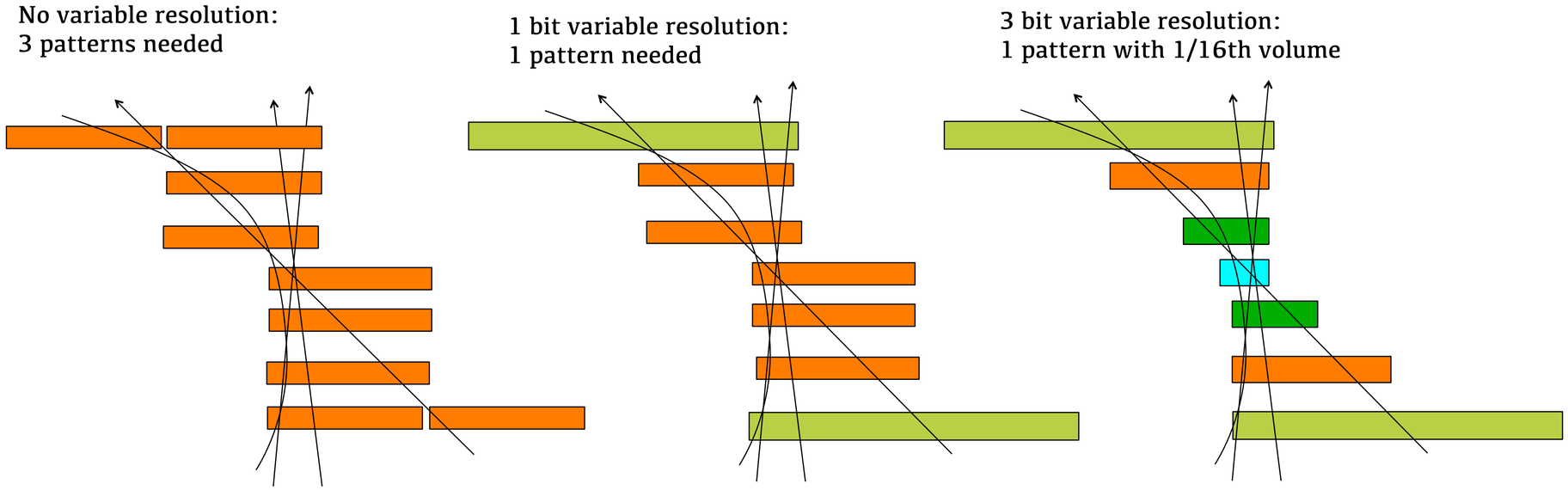
SSB LOGIC


AUX Wrapper
TF Wrapper
AUX Parser
8 Layer Sector #
Constant Loader
12 Layer Sector #
Constants
Matrix Multiplier
Extrapolated PIX Coordinates
PIX SSMap
Extrapolated PIX SSID
Extrapolated SCT Coordinates
SCT SSMap
Extrapolated SCT SSID
8 Layer Tracks
Hit Retriever
Output Manager
DF Wrapper
DF Parser
PIX SSMap
PIX SSID
PIX Coordinates
SCT SSMap
SCT SSID
SCT Coordinate 1
SCT SSMap
SCT SSID
SCT Coordinate 2
HXMPP
Hit Info
Extrapolated SSID
8 Layer Tracks
Hit Info
12 Layer Sector #


Road Organizer
Sector Number
External RAM Controller
Internal Constants Storage
Constants
Nominal Fitter Block
Input FIFO
Combiner
Synchronizer
Nominal / Recovery Fitter
Output FIFO
12 Layer Nominal Tracks (Unfitted)
12 or 11 Layer Tracks (Fitted)
Internal Constants Storage
Constants
PIX Majority Fitter Block
Input FIFO
Combiner
Synchronizer
PIX Majority Fitter
Output FIFO
11 Layer PIX Majority Tracks (Unfitted)
11 Layer Tracks (Fitted)
Internal Constants Storage
Constants
SCT Majority Fitter Block
Input FIFO
Combiner
Synchronizer
SCT Majority Fitter
Output FIFO
11 Layer SCT Majority Tracks (Unfitted)
11 Layer Tracks (Fitted)
Event Wrapper

REFINEMENTS

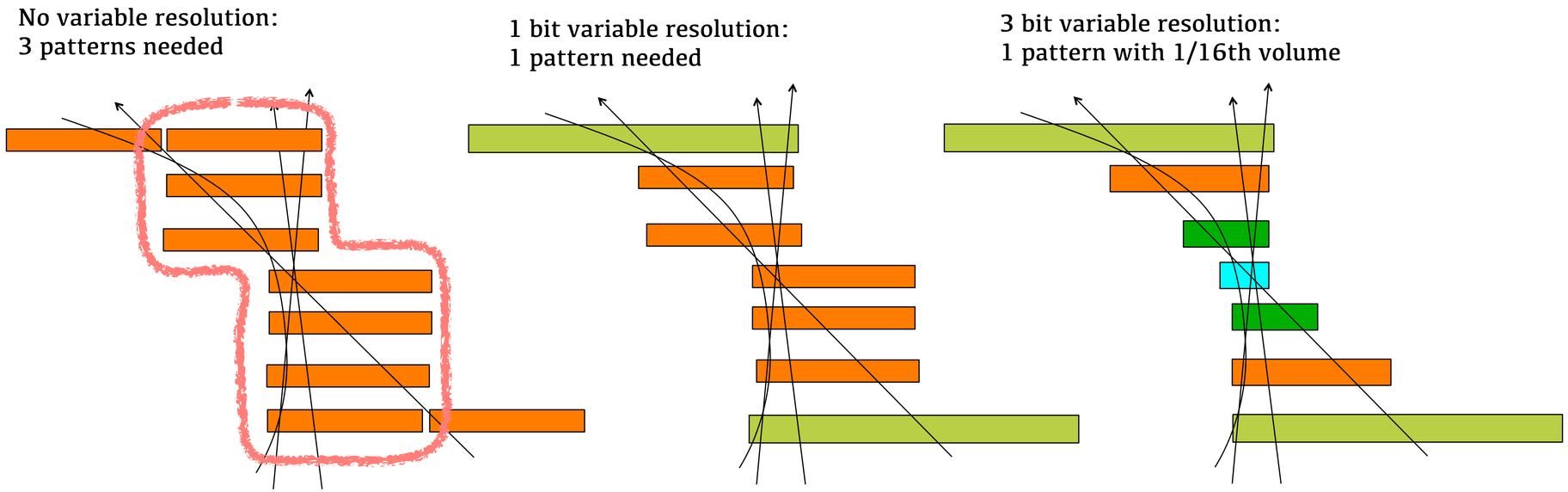
- Majority Logic: Only require N out of M layers have a match
  - Gains efficiency
- Variable Resolution Patterns (Don't Care Bits)
  - Reduces the number of patterns and fake matches



- Number of don't care bits set on a layer by layer, pattern by pattern basis

REFINEMENTS

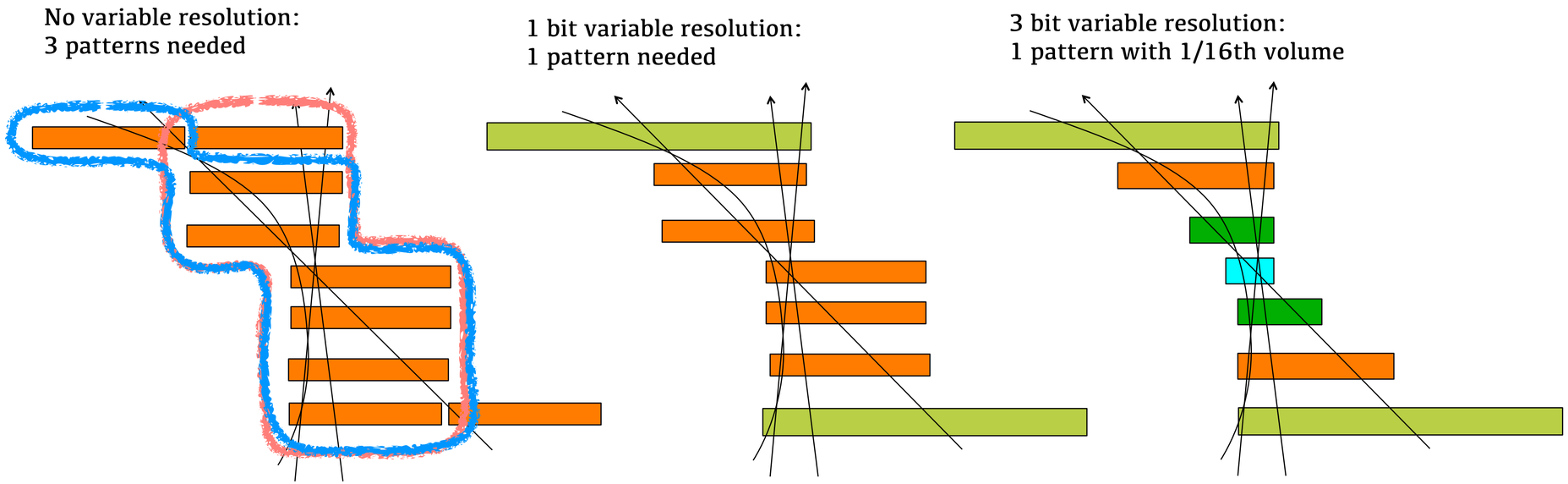
- Majority Logic: Only require N out of M layers have a match
  - Gains efficiency
- Variable Resolution Patterns (Don't Care Bits)
  - Reduces the number of patterns and fake matches



- Number of don't care bits set on a layer by layer, pattern by pattern basis

REFINEMENTS

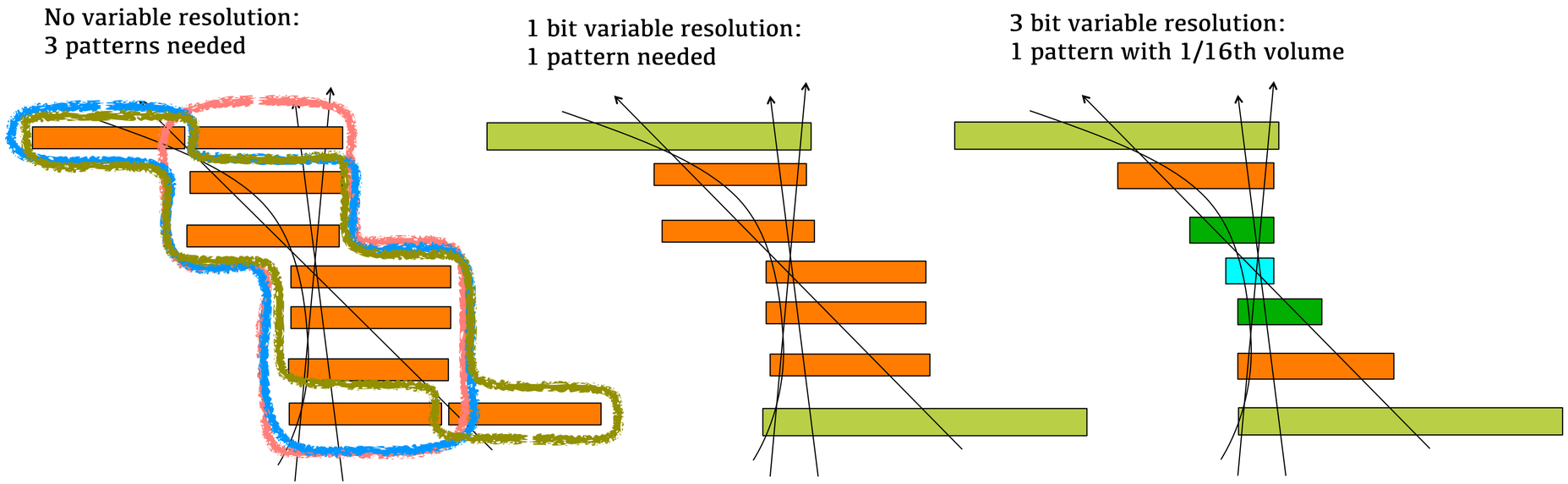
- Majority Logic: Only require N out of M layers have a match
  - Gains efficiency
- Variable Resolution Patterns (Don't Care Bits)
  - Reduces the number of patterns and fake matches



- Number of don't care bits set on a layer by layer, pattern by pattern basis

REFINEMENTS

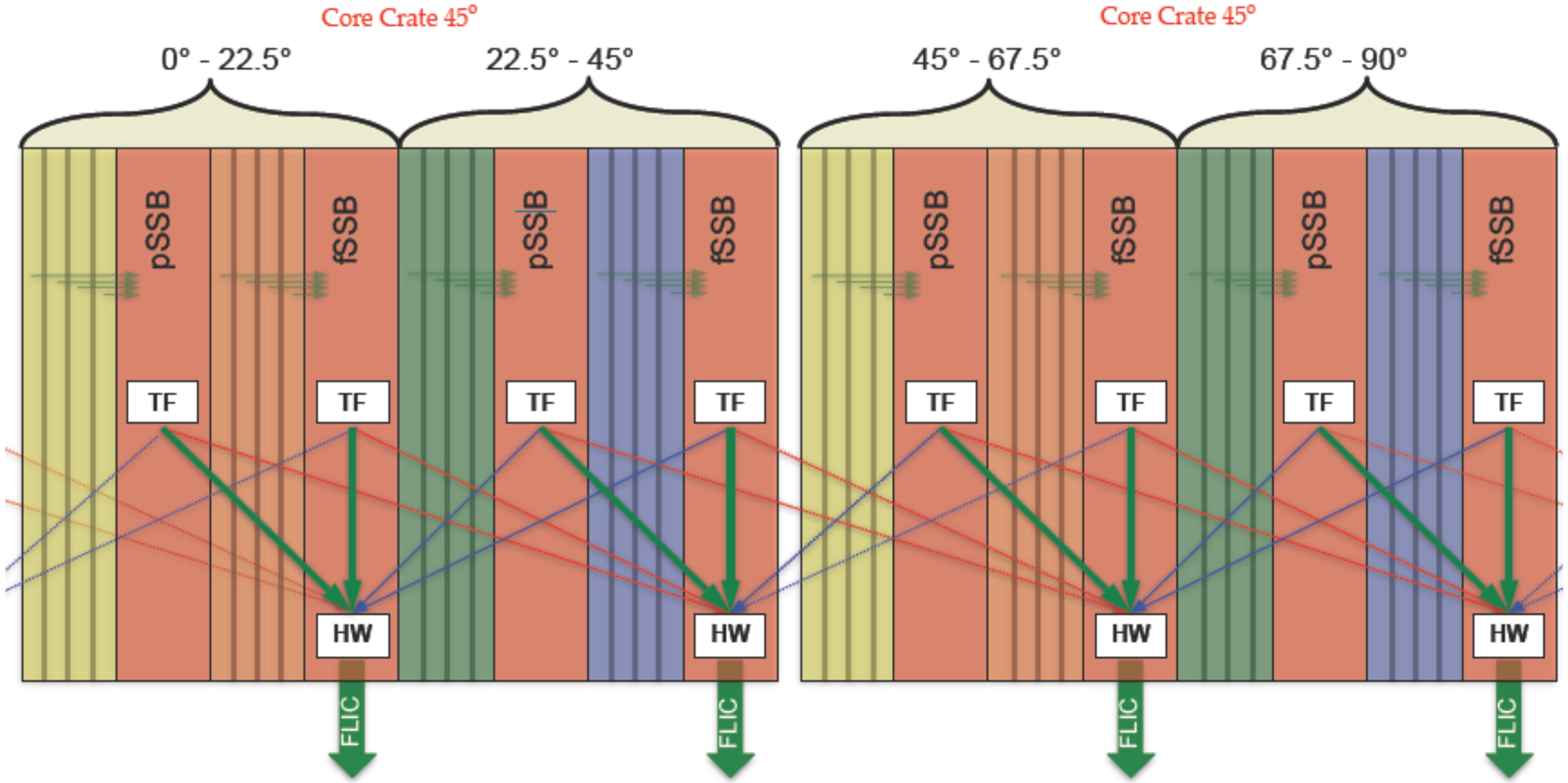
- Majority Logic: Only require N out of M layers have a match
  - Gains efficiency
- Variable Resolution Patterns (Don't Care Bits)
  - Reduces the number of patterns and fake matches



- Number of don't care bits set on a layer by layer, pattern by pattern basis

HW: SSB track flow

Illinois



Green arrows show the flow of tracks. Tracks flow out of the TF and are merged in the HW. **Red** and **Blue** arrows show tracks sent to neighboring φ regions for overlap removal

FLIC functionality

- Receive event records from upstream FTK system, 1/16th of the detector per channel
  - Full bandwidth output from the FLIC to HLT
    - Baseline: 300 tracks per event @ 100 kHz
- Convert FTK identifiers to ATLAS global identifiers using SRAM lookup
- Repackage event record into standard ATLAS format
- Communicate with HLT
  - Sends records
  - receives xoff signal and propagates it upstream to FTK
- Monitoring and Processing on ATCA Blades via backplane