Trigger and Data Acquisition: Hardware

Samuel Silverstein, Stockholm University

- Forward/organization
- Front-end electronics
- Trigger architectures
- Higher-level triggers

“Tools” used

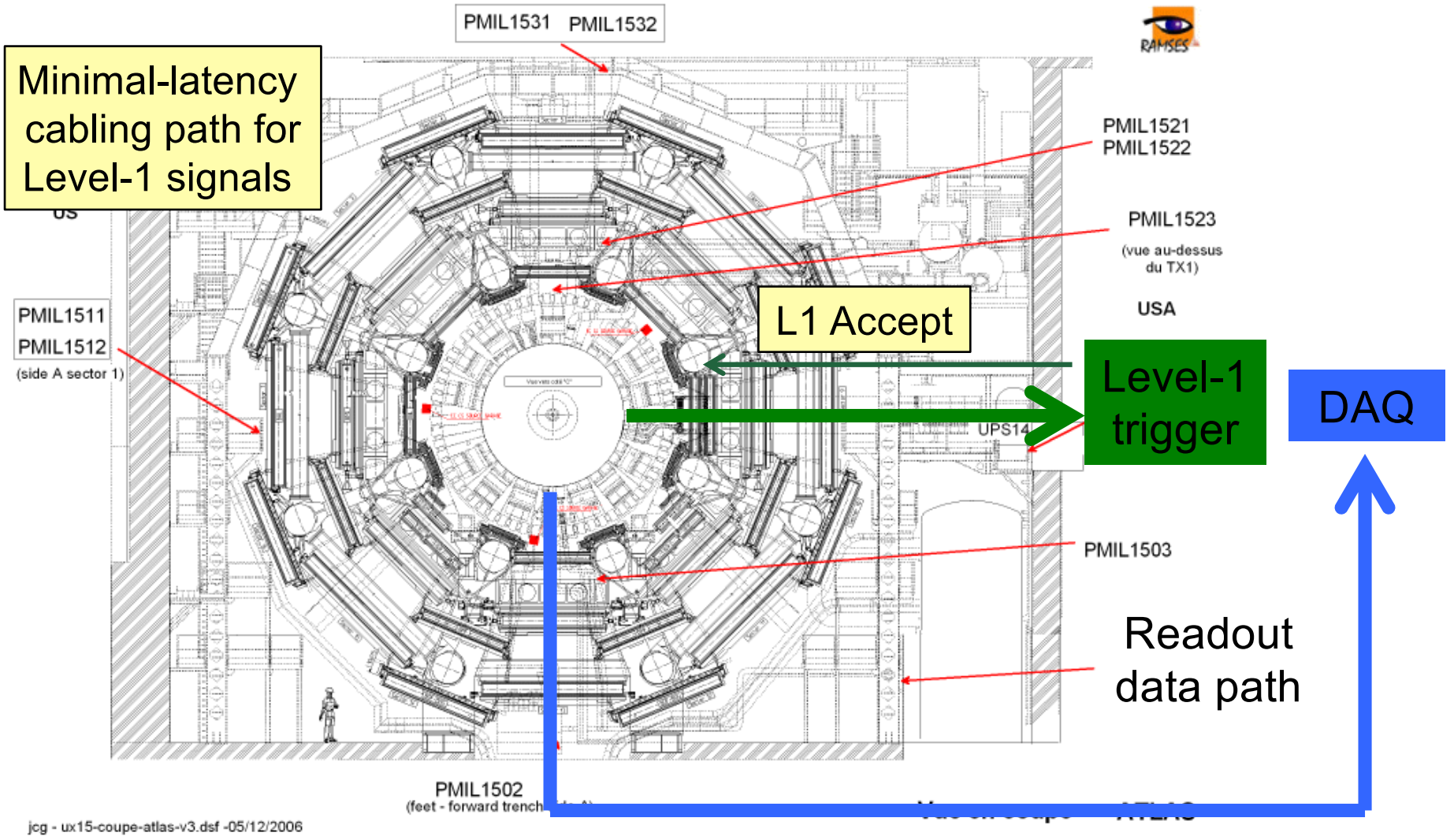
Foreword

- This presentation is weighted towards hardware-based Level-1 trigger systems
  - Greatest burden in terms of efficiency versus rate reduction (at least for hadron experiments)
  - Generally less-well understood than higher-level, processor-based triggers, which overlap with analysis software
- Trigger/DAQ is not an exact science!
  - Different detector systems have different requirements because of unique design choices
  - Even similar requirements are often solved differently, due to personal preferences and earlier experiences of the designers.

Trigger organization

- On detector (typically):
  - Pre-amplification, shaping
  - Data sampling, pipeline buffers, (ADC)
  - Trigger “tower” summation*
- Transmission to Level-1:
  - “Short” cables/fibers (50-100m)
- Off detector (typically):
  - patch-panels/receivers
  - A/D conversion of analog signals
  - Distribution to digital processors

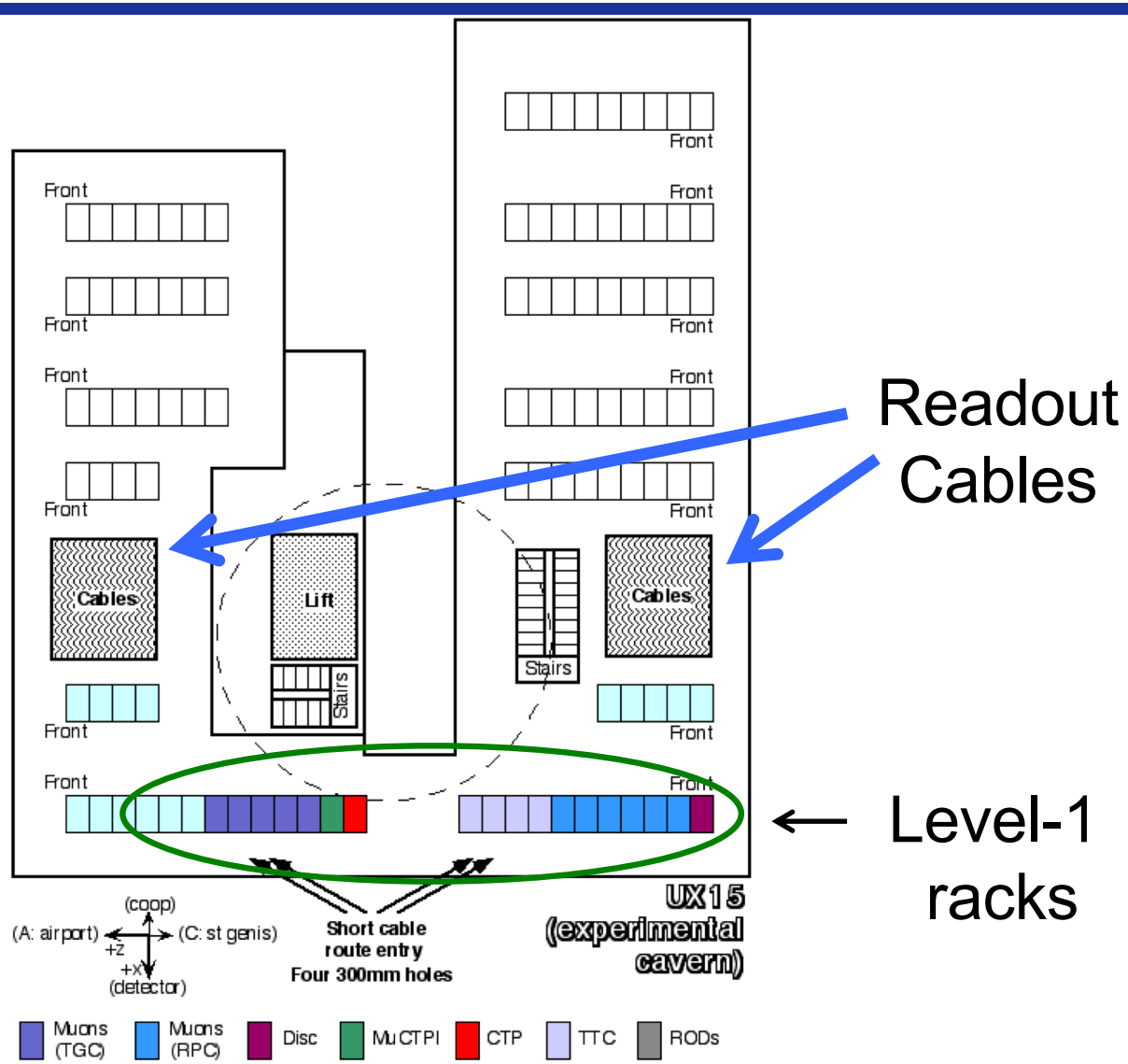
Level-1 data path



Level-1 positioning (ATLAS)

Placed to reduce cable and fiber delays

Experiment

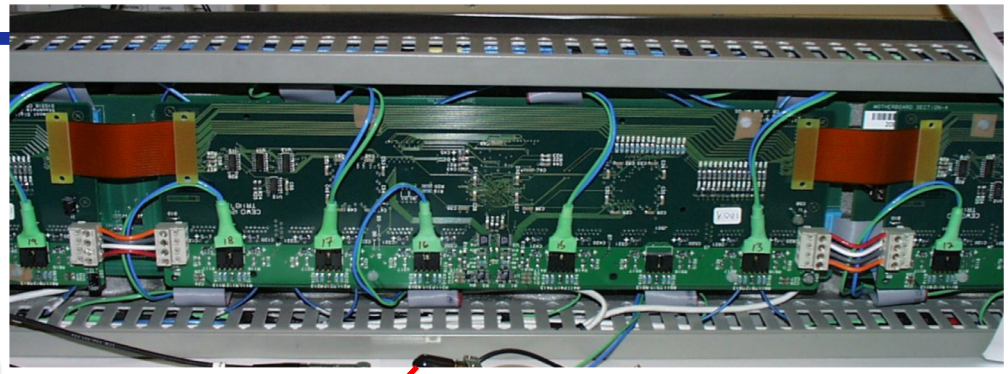


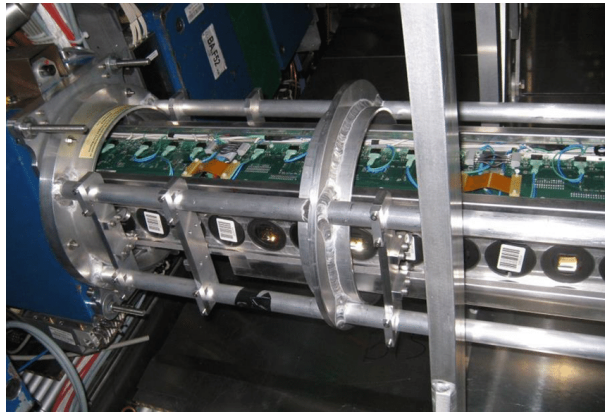
Electronics

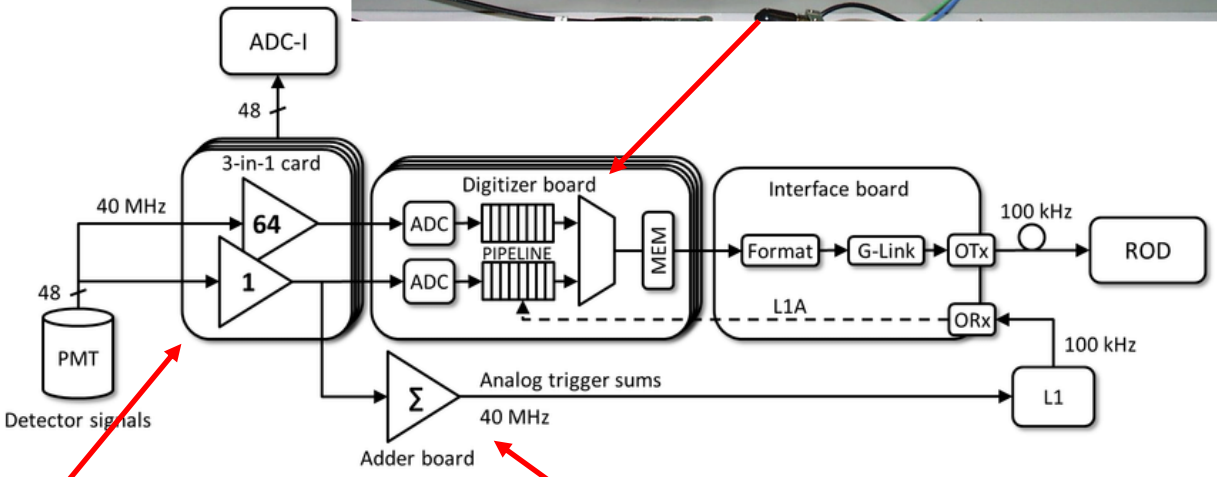
Analog inputs

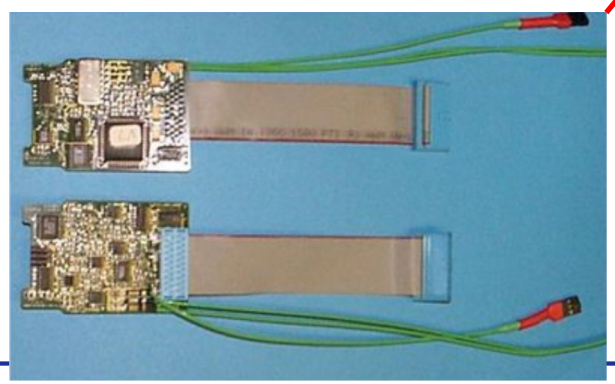
- Analog signal conditioning
  - Application specific, usually need to be optimized to the detector
  - Discrete components, custom circuit boards
- Analog-digital conversion
  - Mixed-signal integrated circuit – hard to get right
  - Often commercial components, with exceptions for on-detector applications with special demands:
    - Radiation tolerance
    - Low power consumption

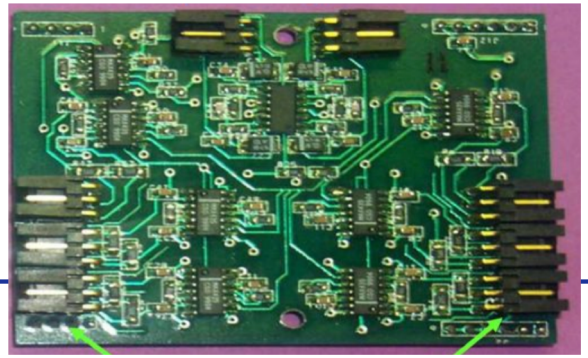
ATLAS TileCal front end







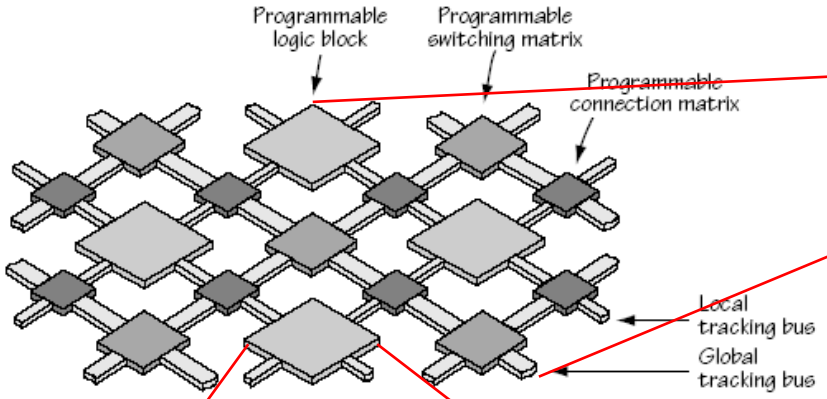


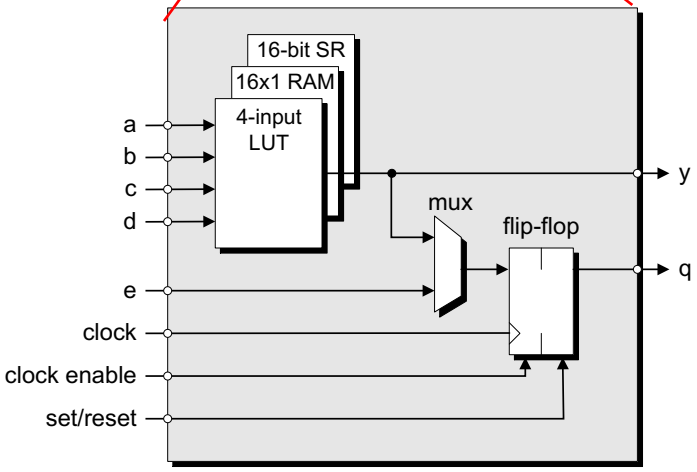


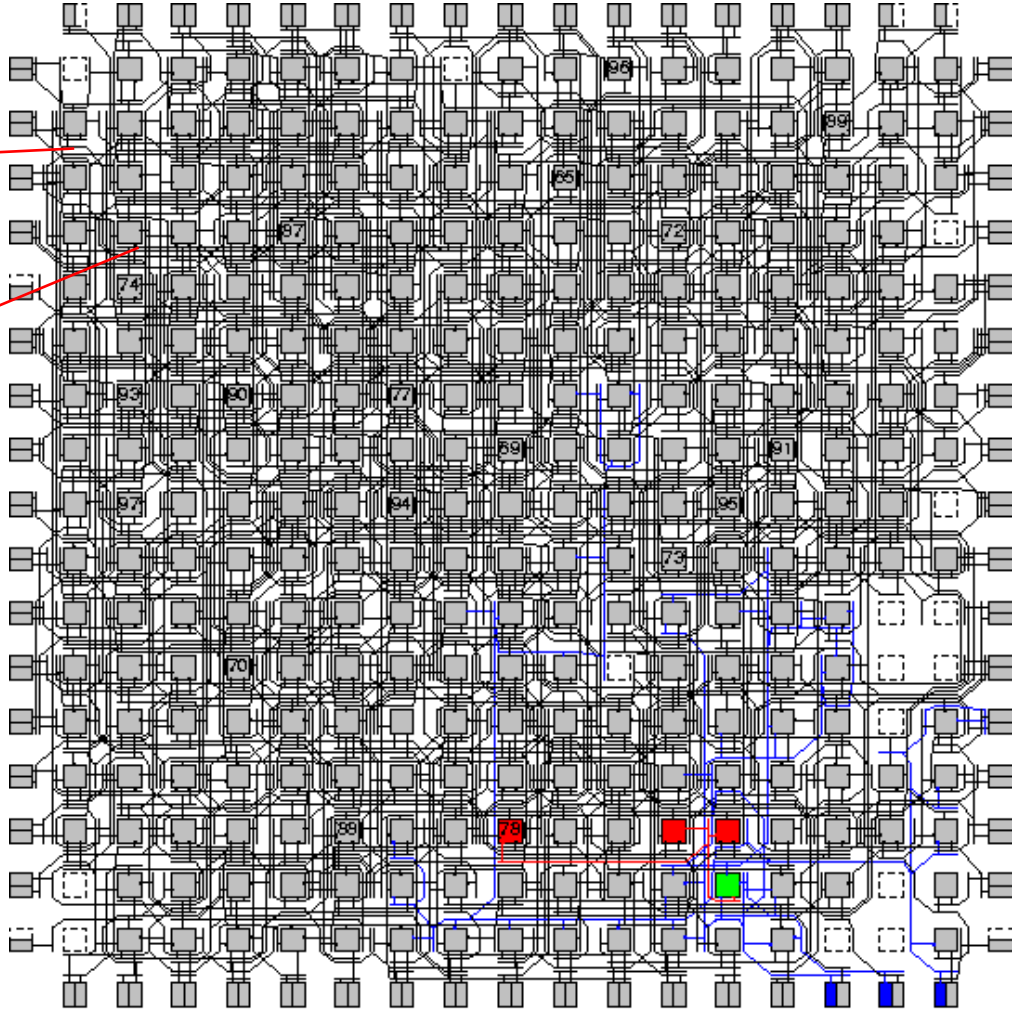
Integrated digital circuits

- A long time ago (pre-1990)
  - Large, complex circuit boards with many discrete, commercial components
    - Logic gates (and, or…)
    - Registers (flip-flops) for storing data between clock cycles
    - Memory (RAM)
    - Input/output signal drivers (buffers)
    - Microprocessors, etc.
- Today
  - Digital functionality condensed into a few highly integrated circuits (often FPGAs)

Digital logic in FPGAs

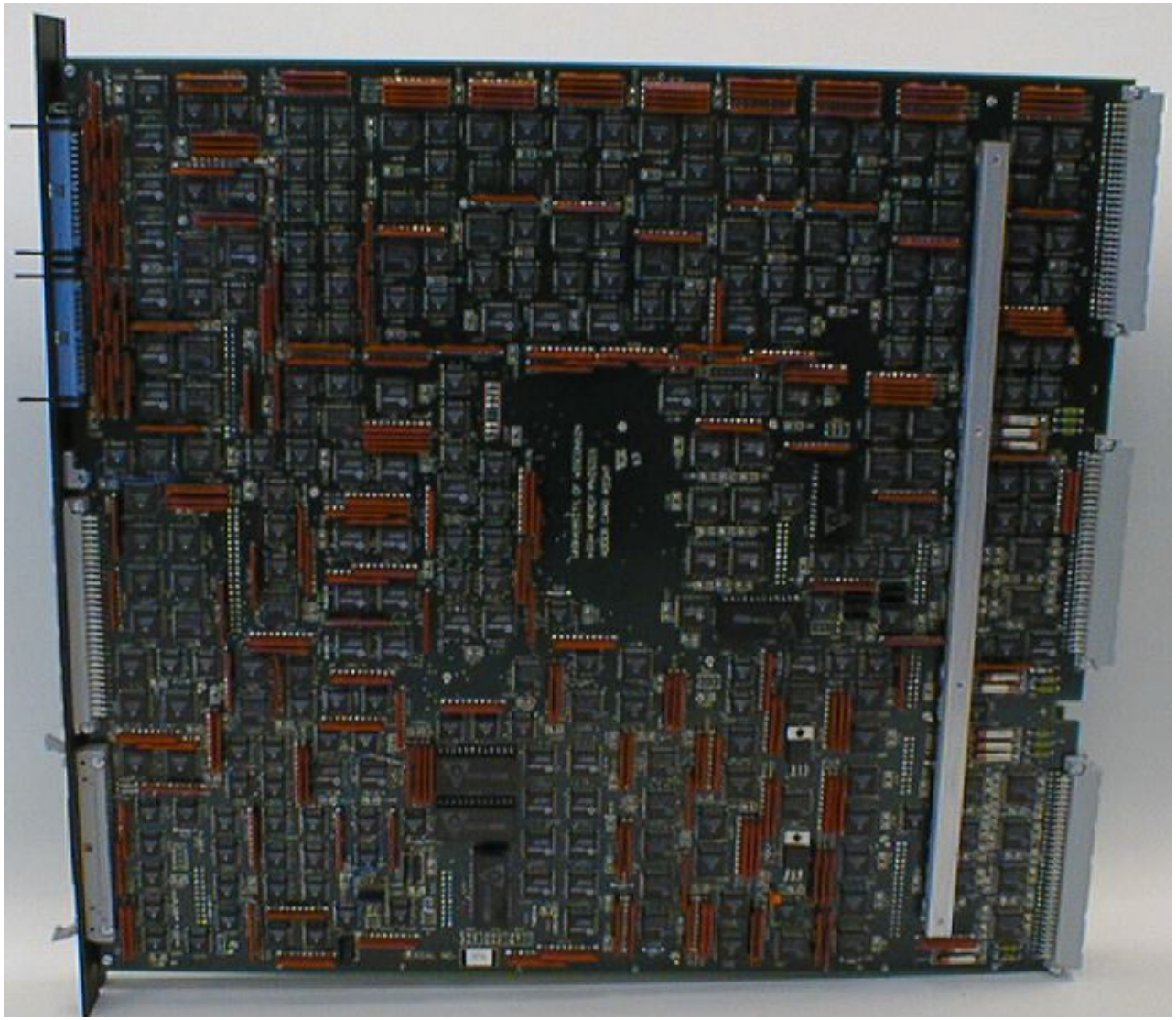






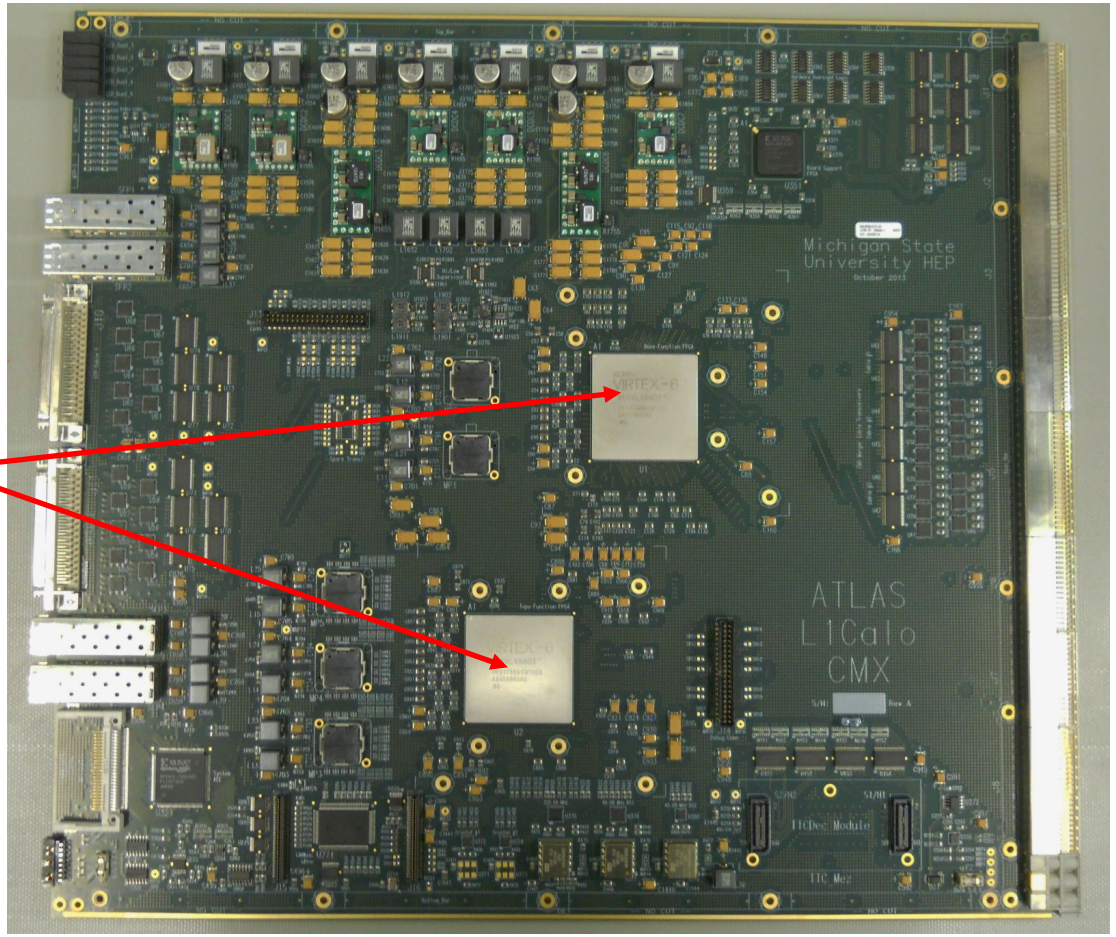
See FPGA lab exercise

Discrete logic (ZEUS)



FPGA-based (ATLAS CMX)

Functionality
implemented
in large FPGAs

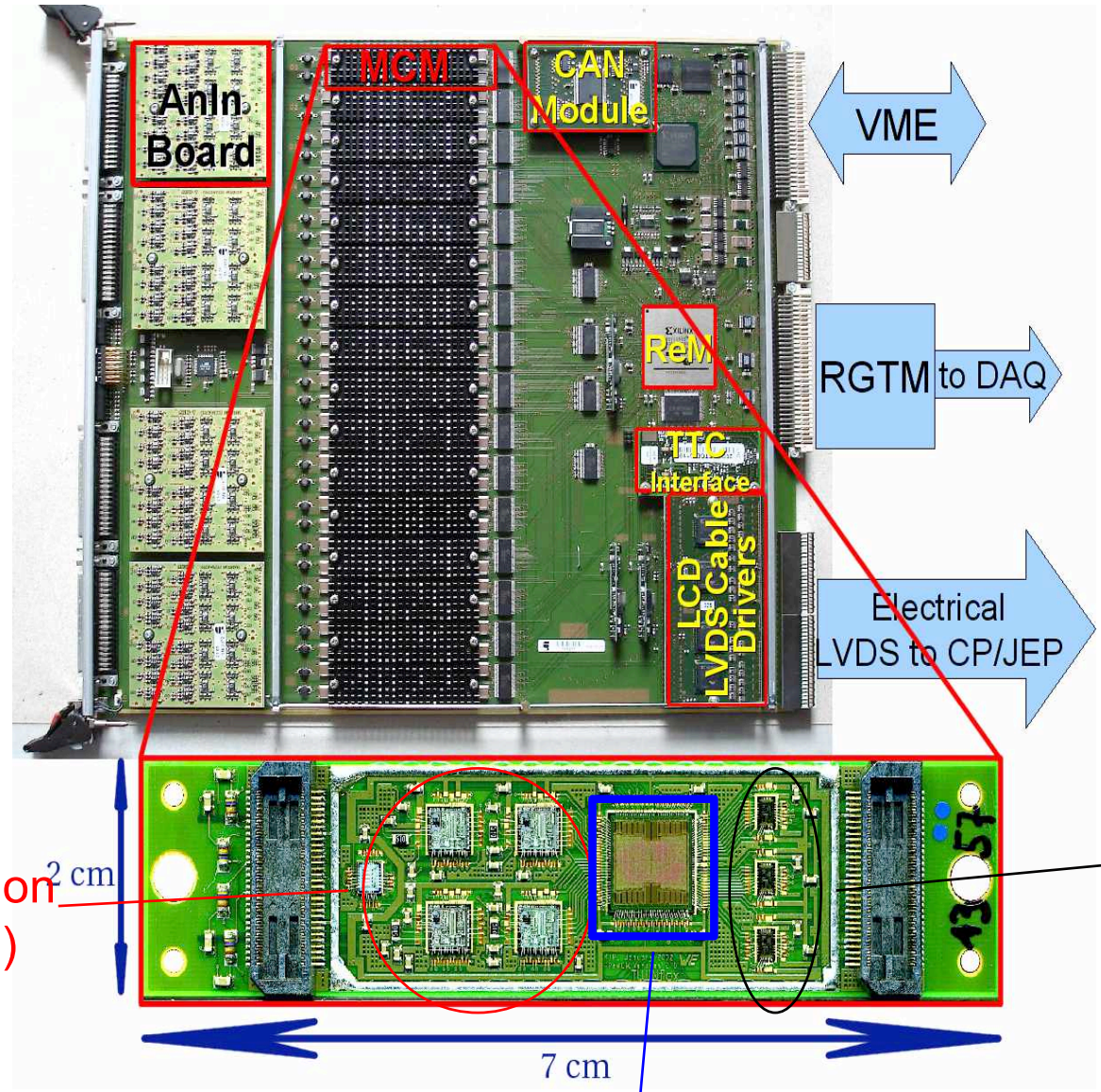


Integrated digital circuits

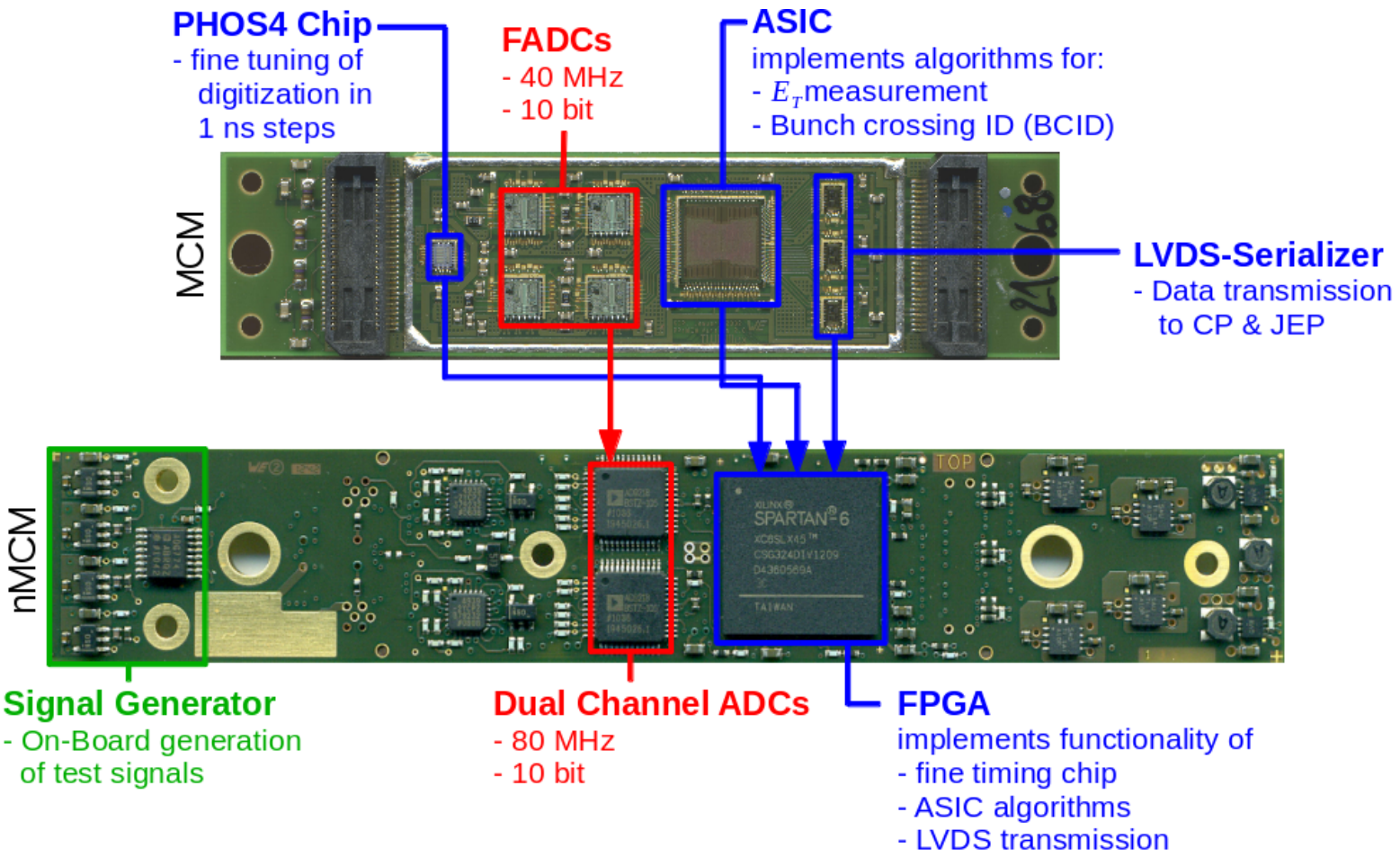
- Two approaches:
  - Application-specific integrated circuit (ASIC)
    - Custom-designed circuit with single, fixed function
    - Expensive and time-consuming to design and produce (economical if you produce many copies)
    - Advantages of ASICS
      - Low power, high-speed
      - Can choose e.g. radiation tolerant processes
      - Possible to make mixed digital/analog designs
  - Field-programmable Gate Array (FPGA)
    - Commercial, user-configurable digital circuit
    - Flexible, can be reconfigured

ATLAS L1Calo PreProcessor used ASICs in Run 1, switched to FPGAs for Run 2

ATLAS L1Calo PreProcessor



MCM upgraded for Run 2



FPGA allows digital logic to be upgraded!
So far: improved filter performace. dynamic baseline subtraction

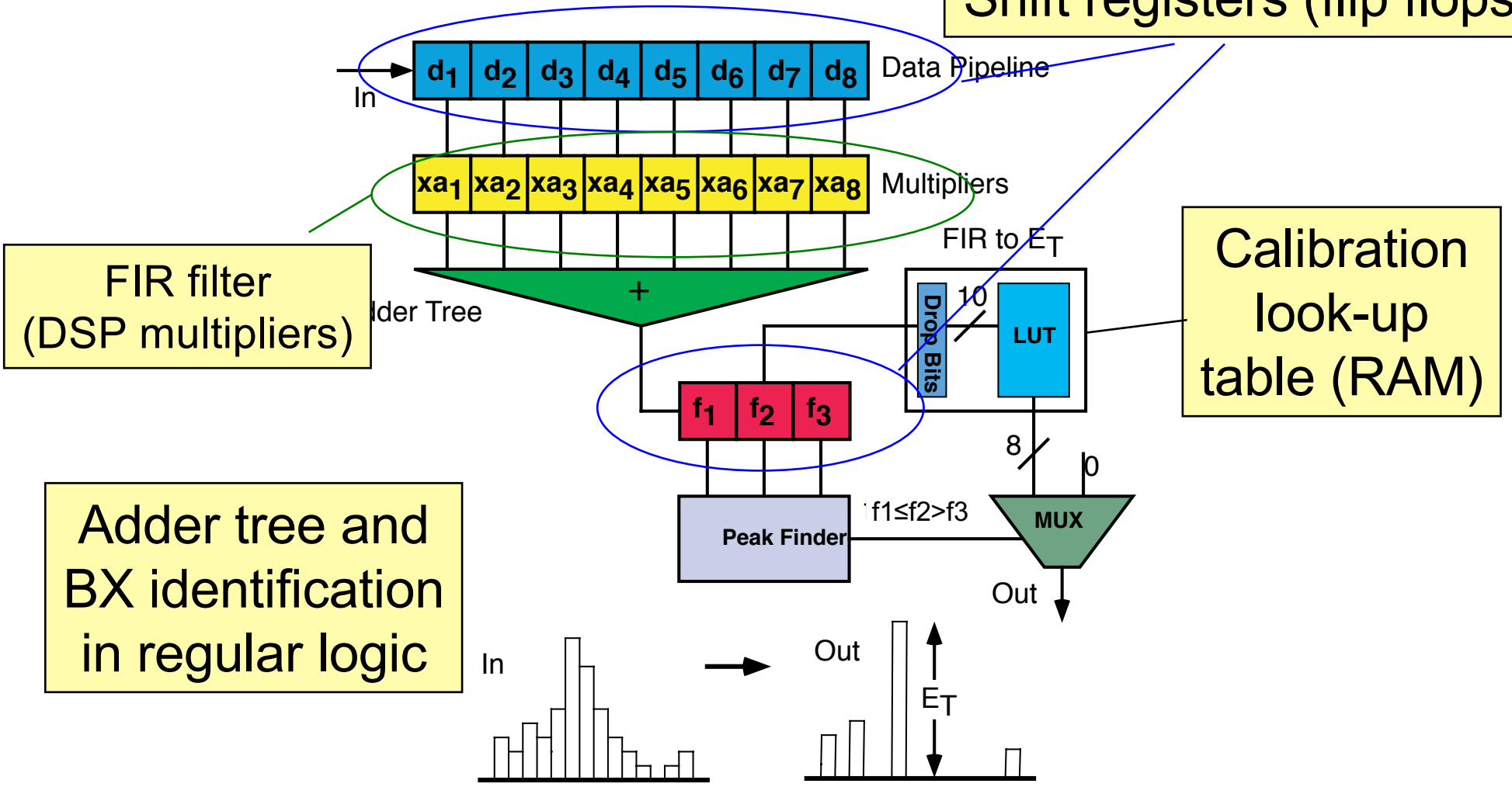
PreProcessor ASIC algorithm

Shift registers (flip flops)

FIR filter (DSP multipliers)

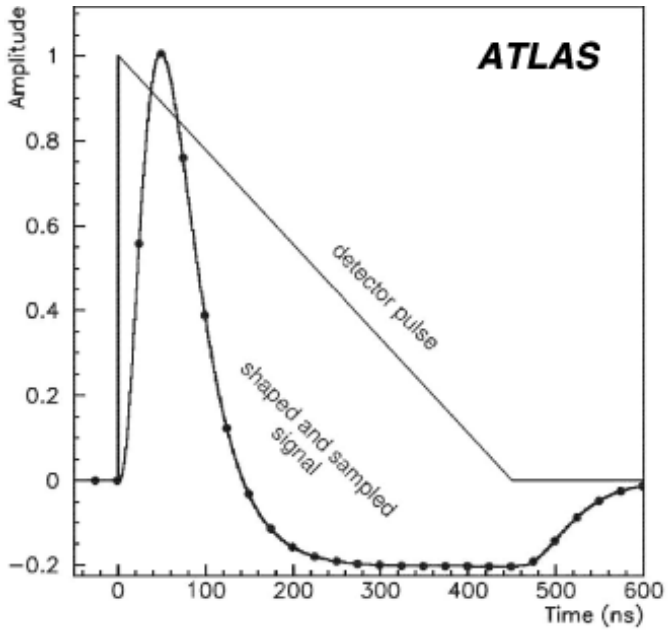
Calibration look-up table (RAM)

Adder tree and BX identification in regular logic

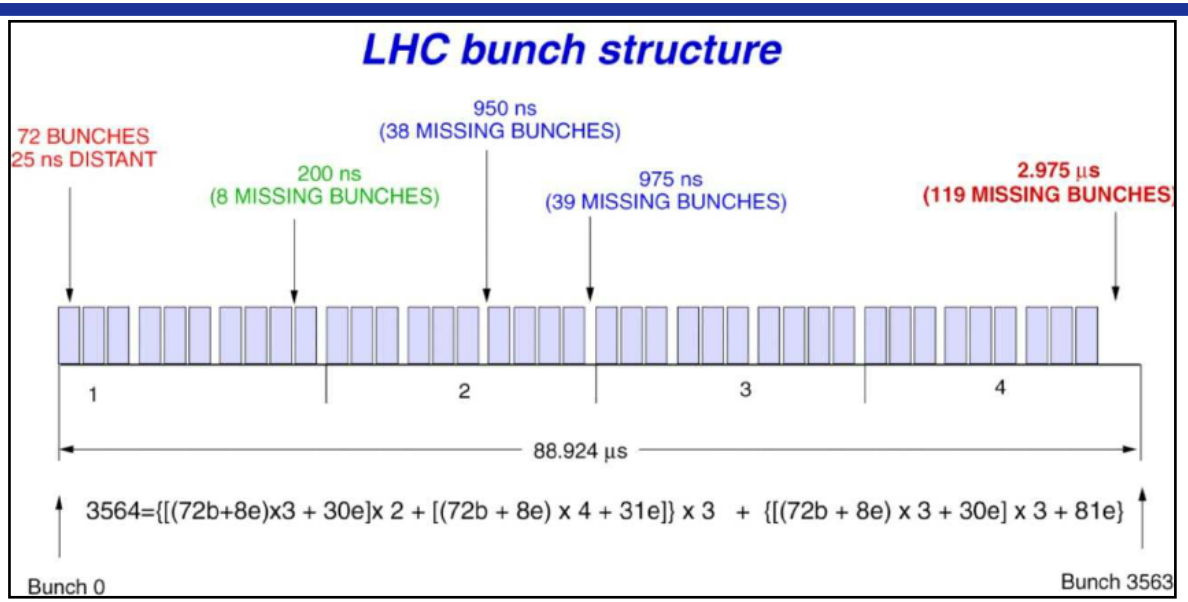


Bunch-dependent pedestal shift

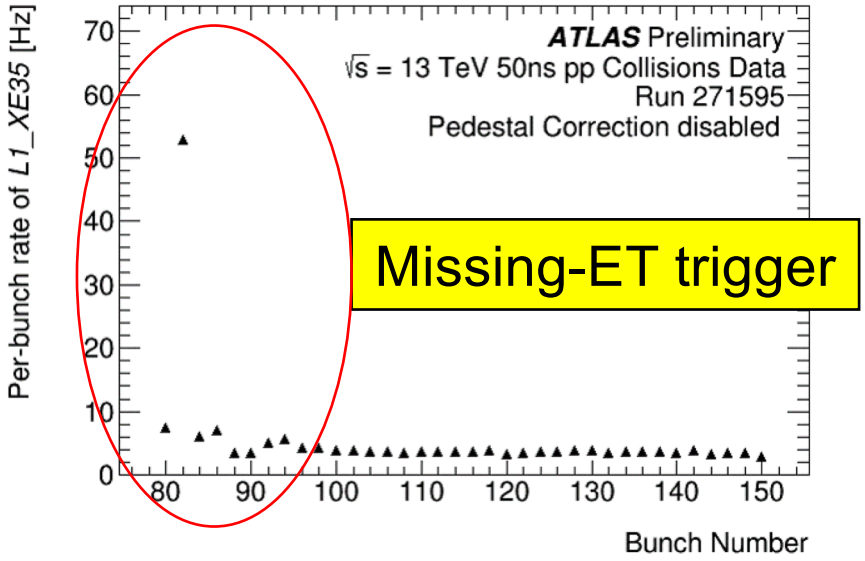
Bipolar LAr pulse shape



plus

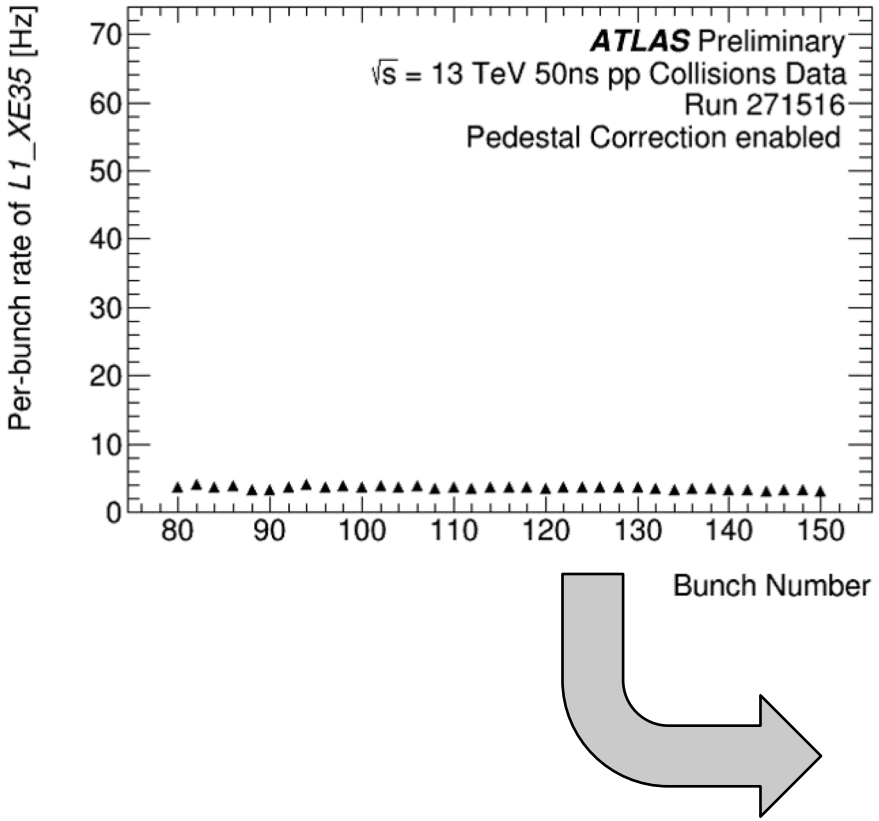


causes this!

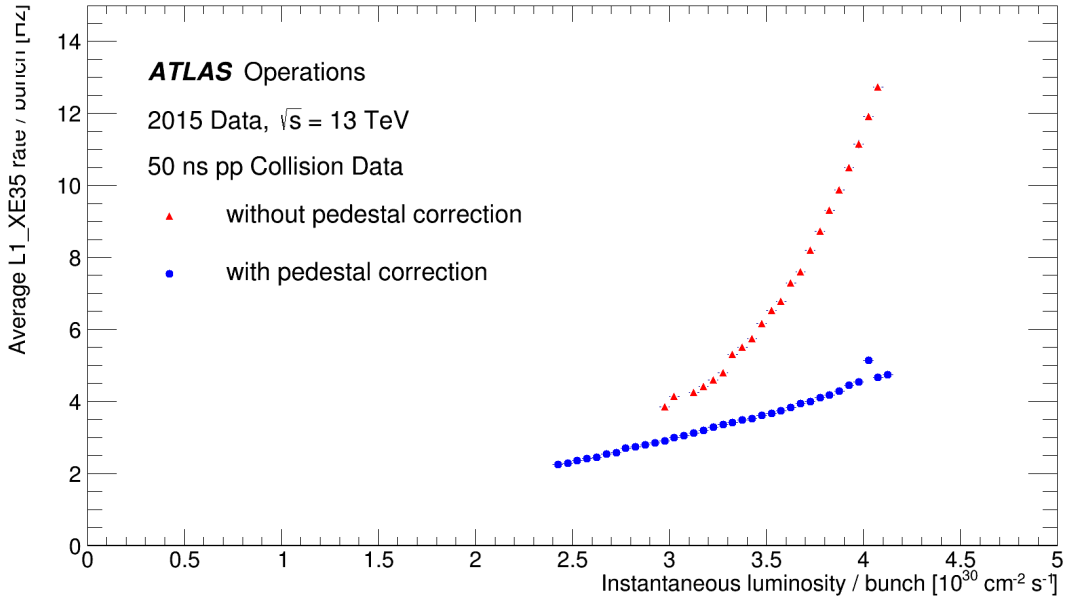


Missing-ET trigger

Pedestal correction w/ new MCM



Missing ET rates now scale with luminosity:



New FPGAs

- FPGAs in original trigger systems already >15-20 years old
- New families of FPGAs include:
  - Large amounts of resources
    - Millions of logic cells
    - RAM, DSP multipliers, embedded processors
    - System speeds 600 MHz (or more!)
  - Multi-Gbit transceivers
    - Speeds up to 13 Gbit/s (Ultrascale: 32.5)
    - Large FPGAs can have 80-96 (Ultrascale:128)
- System architectures limited by I/O, not logic!

Serial data links

- Preferred choice for data distribution
  - High bandwidth over a single optical fiber or electrical signal pair
  - High signal densities to single processing unit (more on that later)
- Physical implementation
  - Previously: discrete, commercial devices with speeds up to 1 Gbit/s
  - Today: high-speed transceivers built into modern FPGAs and large ICs
    - ATLAS, CMS upgrades use >10 Gb/s links
    - implemented with multi-Gb/s FPGA transceivers

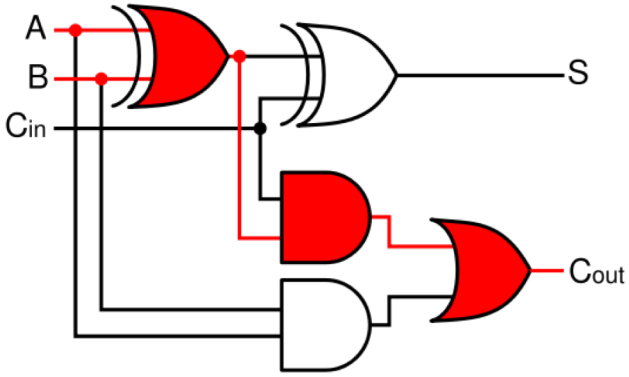
Serial links

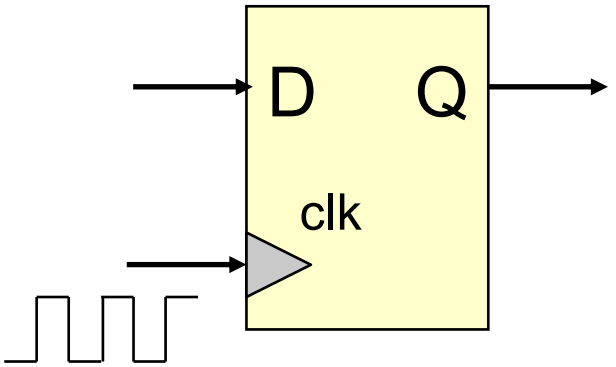
- Legacy LHC systems
  - ~1 Gbit/s electrical or optical
  - Single link per transmitter/receiver
- New: Parallel optic modules
  - Bundle of 12 fibers on optical ‘ribbon’
  - Link speeds: up to >10 Gbit/s per fiber
  - Bundles of 12-72 fibers for routing between systems
    - MTP/MPOConnector
      ~ same size as Ethernet

Algorithm processing
“Toolkit”

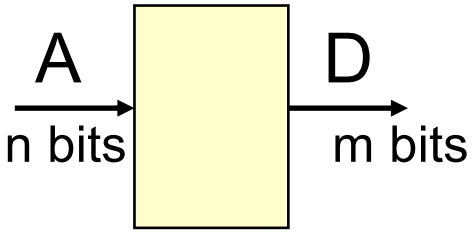
Digital logic

- Combinatorial logic
  - Boolean logic gates (AND/OR/XOR/NOT)
  - Configurable lookup tables (FPGA)
- Registers (flip-flops)
  - Hold intermediate results between clock cycles (sequential logic, pipelined algorithms)





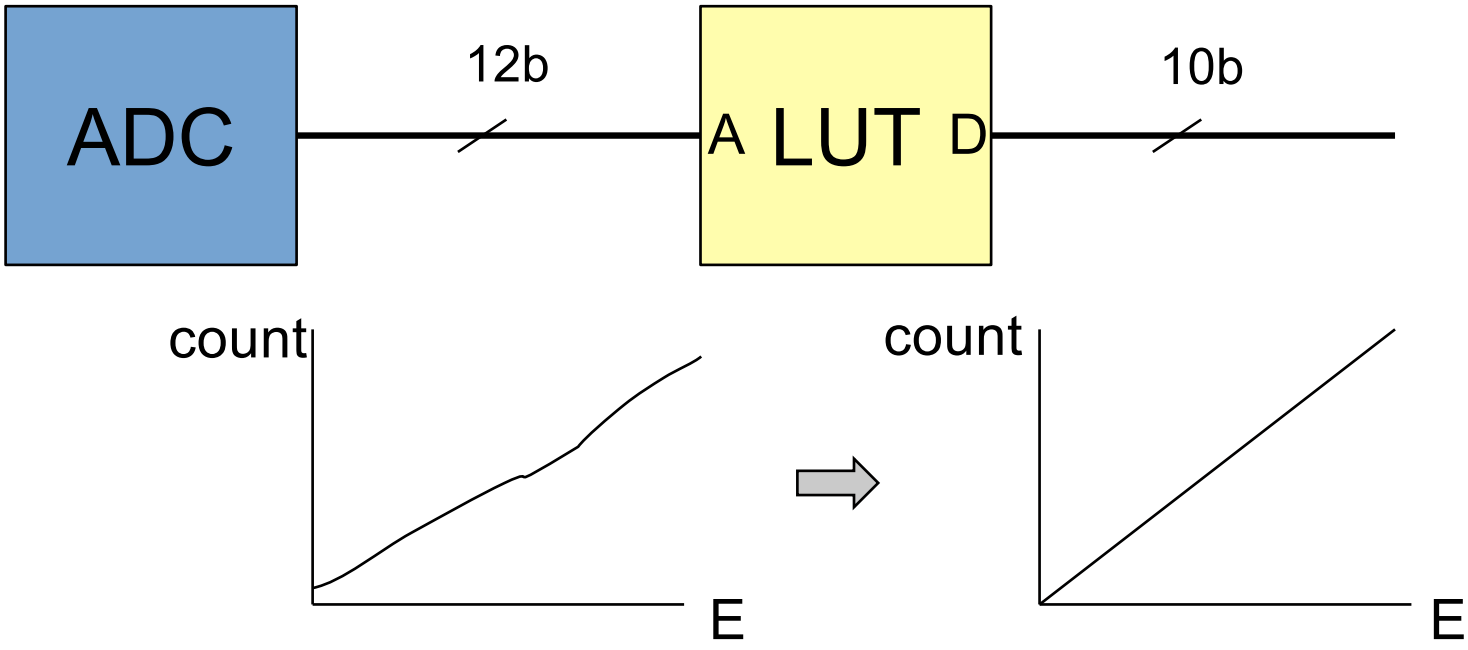
Memory lookup tables (LUT)



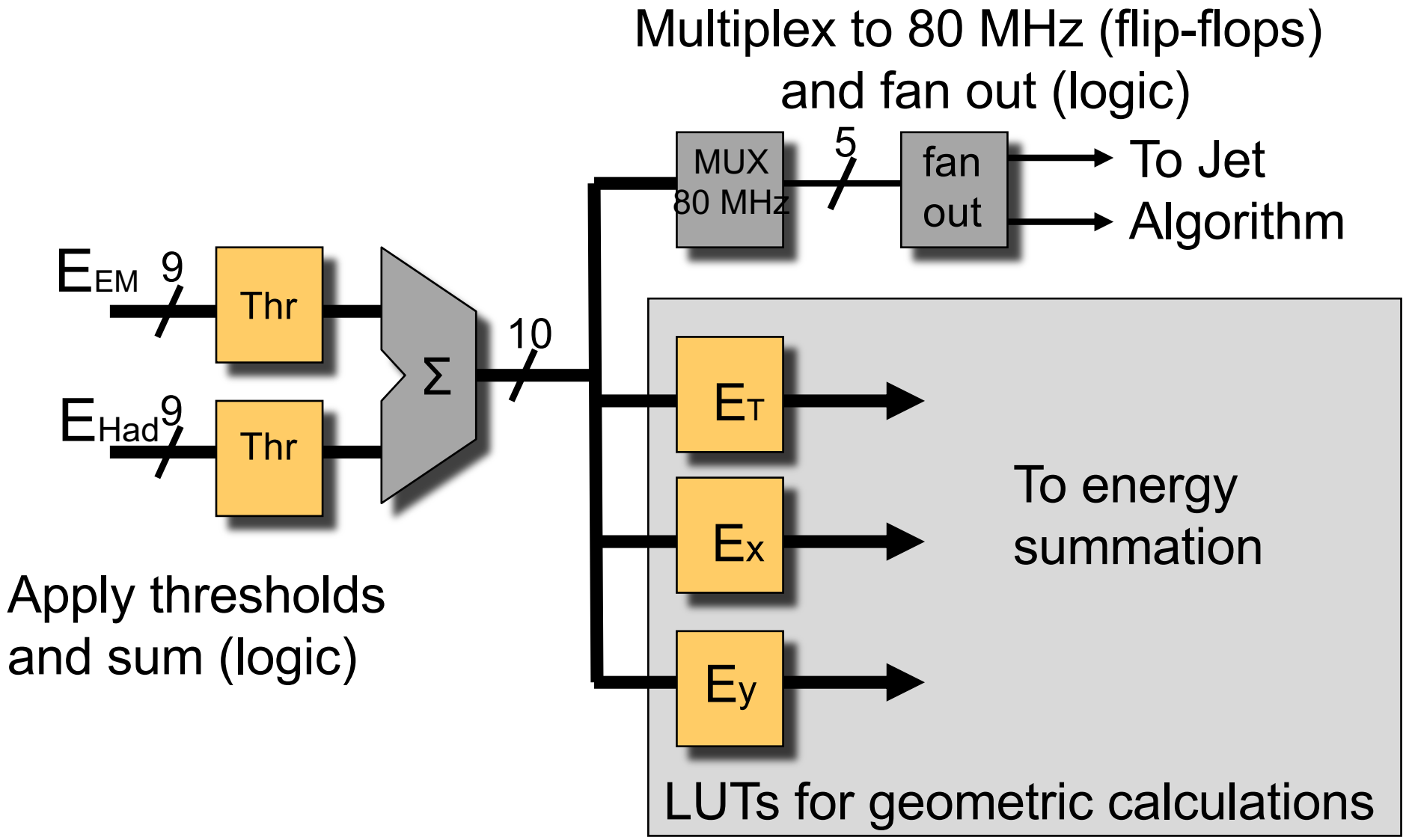
- Random access memory (RAM)
  - n address bits (input)
  - m data bits (output)
- Input points to the memory address containing the corresponding output
- Data bits drive output to the next stage
- Good for operations where n ≤ 14-16 bits, n ~ m
  - Larger blocks of static RAM blocks get expensive
- Fast, low latency
  - Any arbitrary function in a single clock cycle
- Flexible
  - Just change the memory contents

LUTs for calibration

Common application: apply LUT to raw data to correct for calibration, linearity, etc.



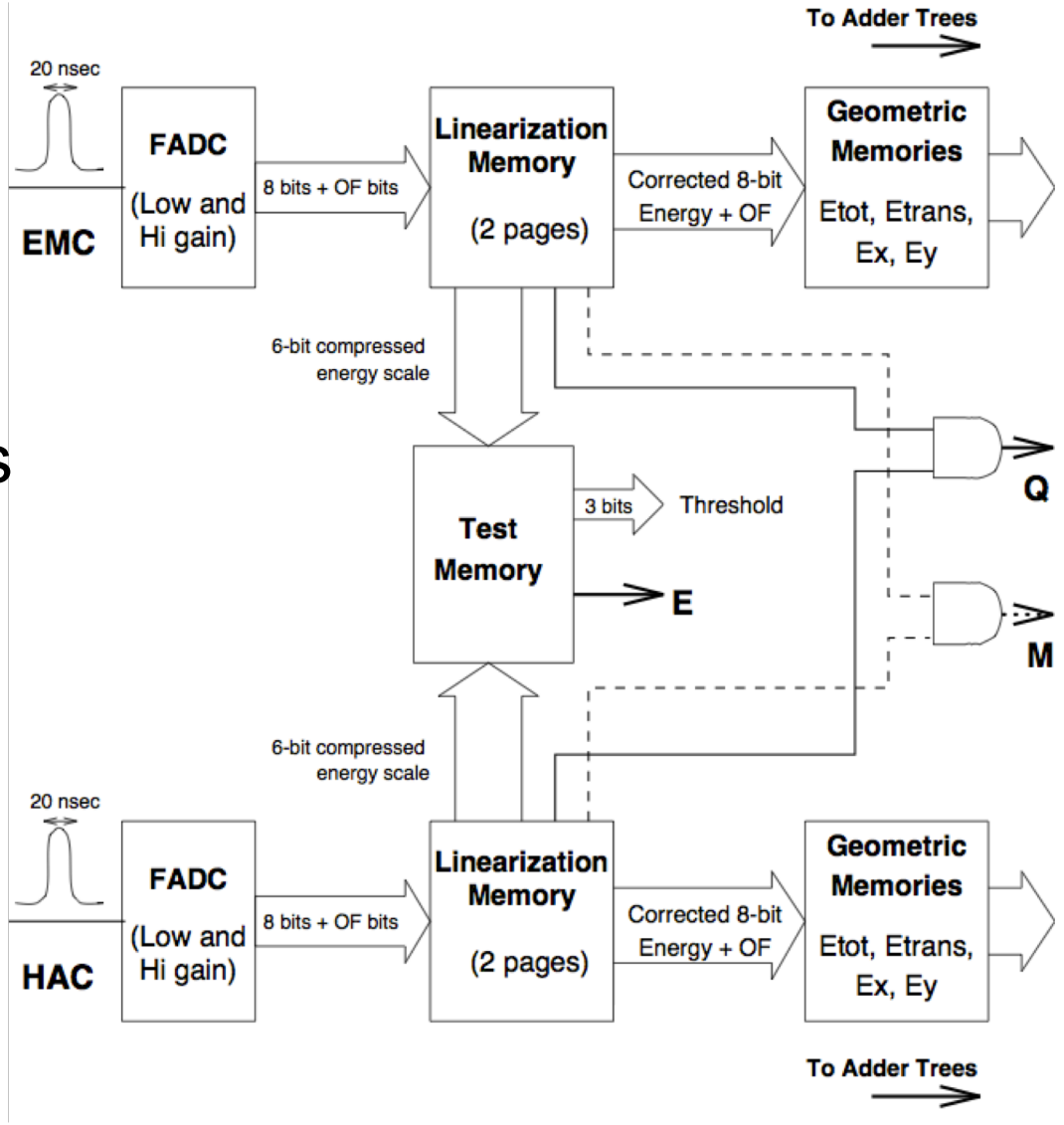
LUTs for thresholds/calculations



ZEUS trigger encoder

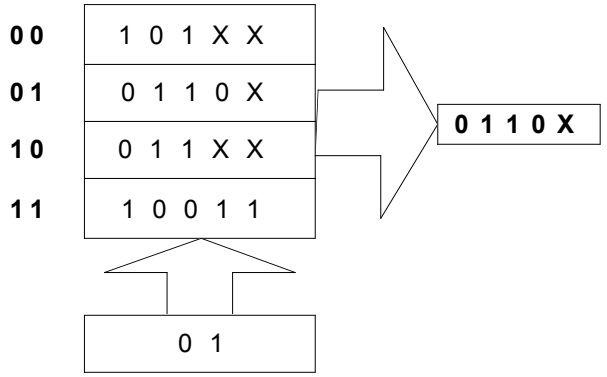
LUTs used for:
- Calibration/linearity
- Geometric calculations
- Tower 'thresholds'
- Pattern logic bits

Implemented in discrete static RAM

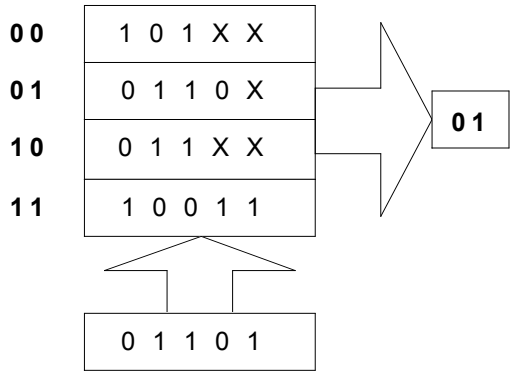


Content Addressable Memory (CAM)

- CAM (also Associative Memory) is a special kind of memory LUT
- In traditional LUT:
  - Input is address of the result that we are interested in.
  - Output is the data at that address.
- In CAM it is the reverse:
  - Input is the content we want to find in the memory.
  - Output is the address(es) where that content is stored.
- CAMs support Ternary data
  - 1, 0, X (don't care)
- Useful for parallel search through many possible patterns
  - For example, fast track finding



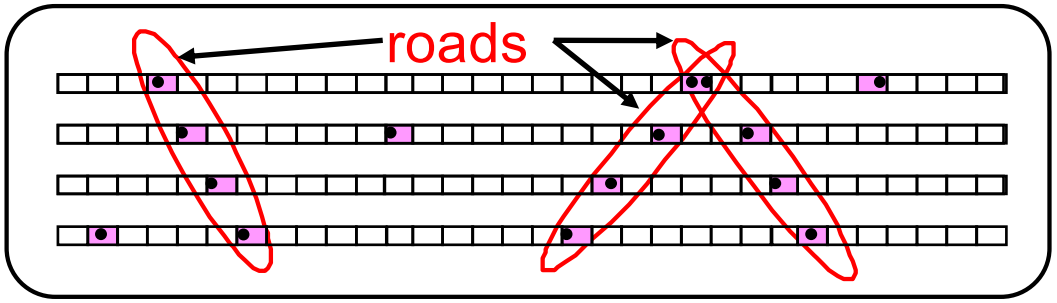
Traditional Memory



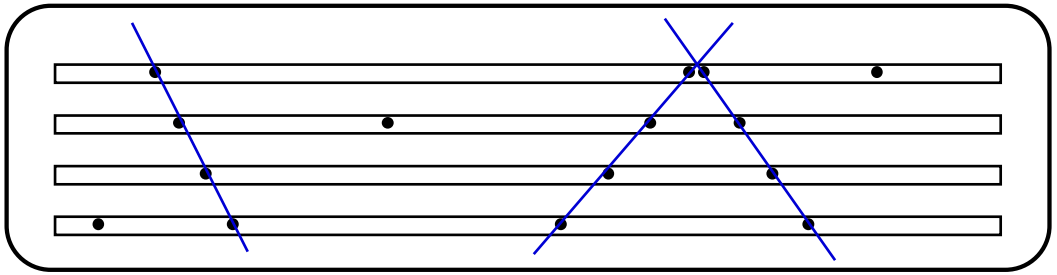
Content Addressable Memory

Example: track finding

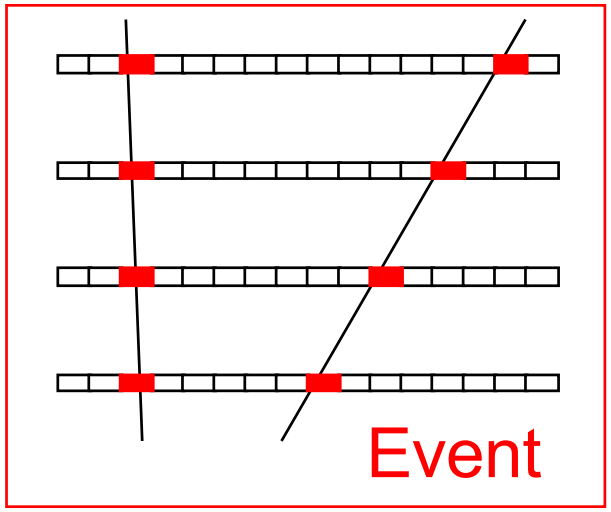
- Separate track-finding into two stages:
  1. Find low-resolution “roads”—track candidates.
     - Use brute force
     - Solves combinatorics
     - Parallel processing

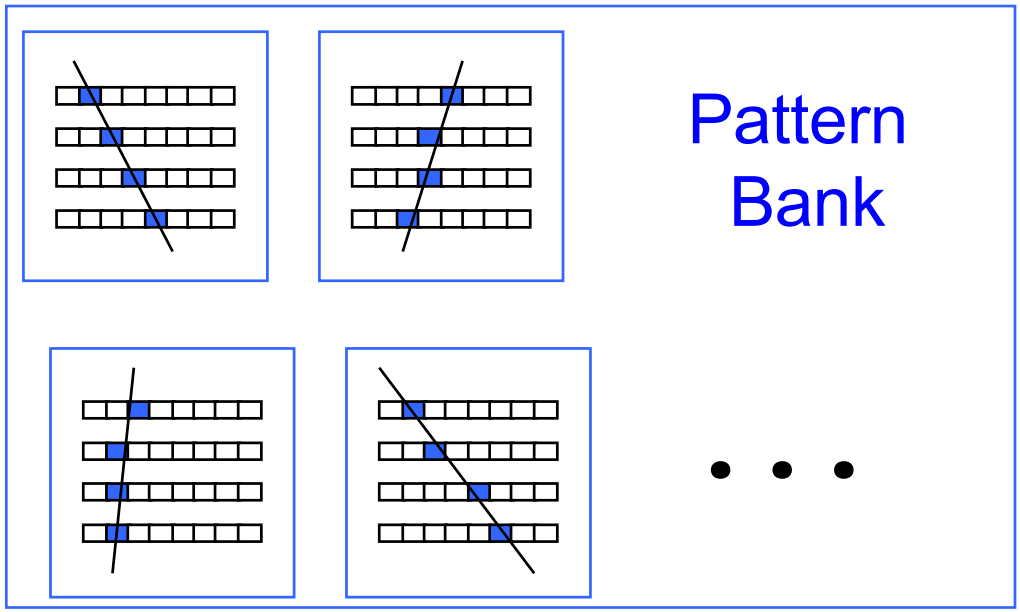


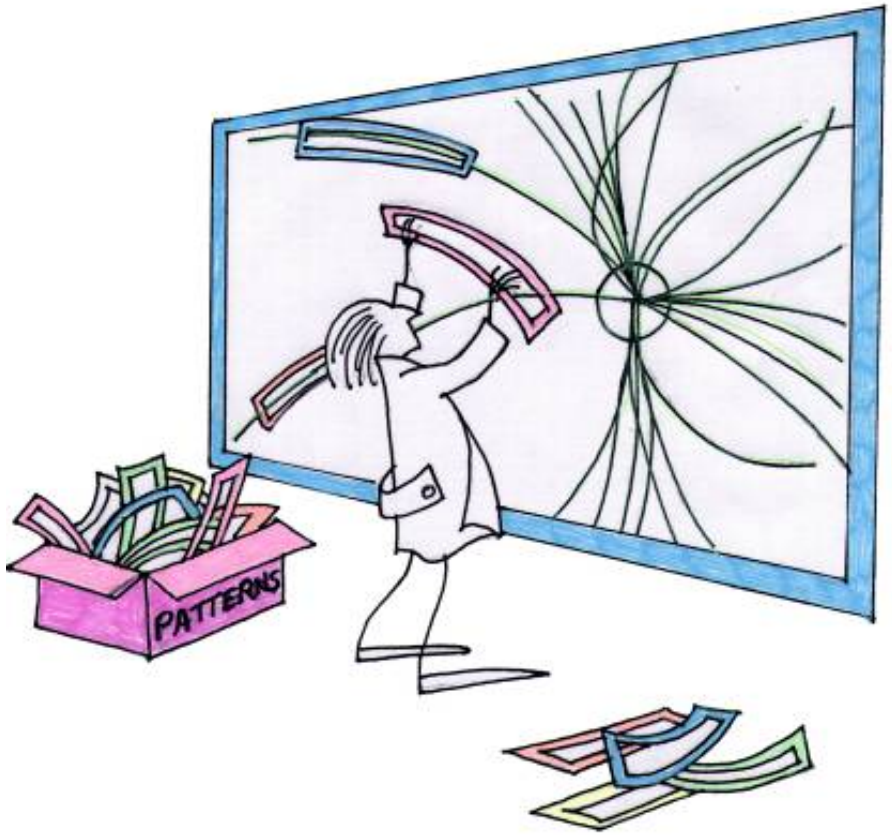
  2. Fit high-resolution tracks inside roads.
     - Easier after 1st step
     - Pin down track parameters
     - Cut on fit quality

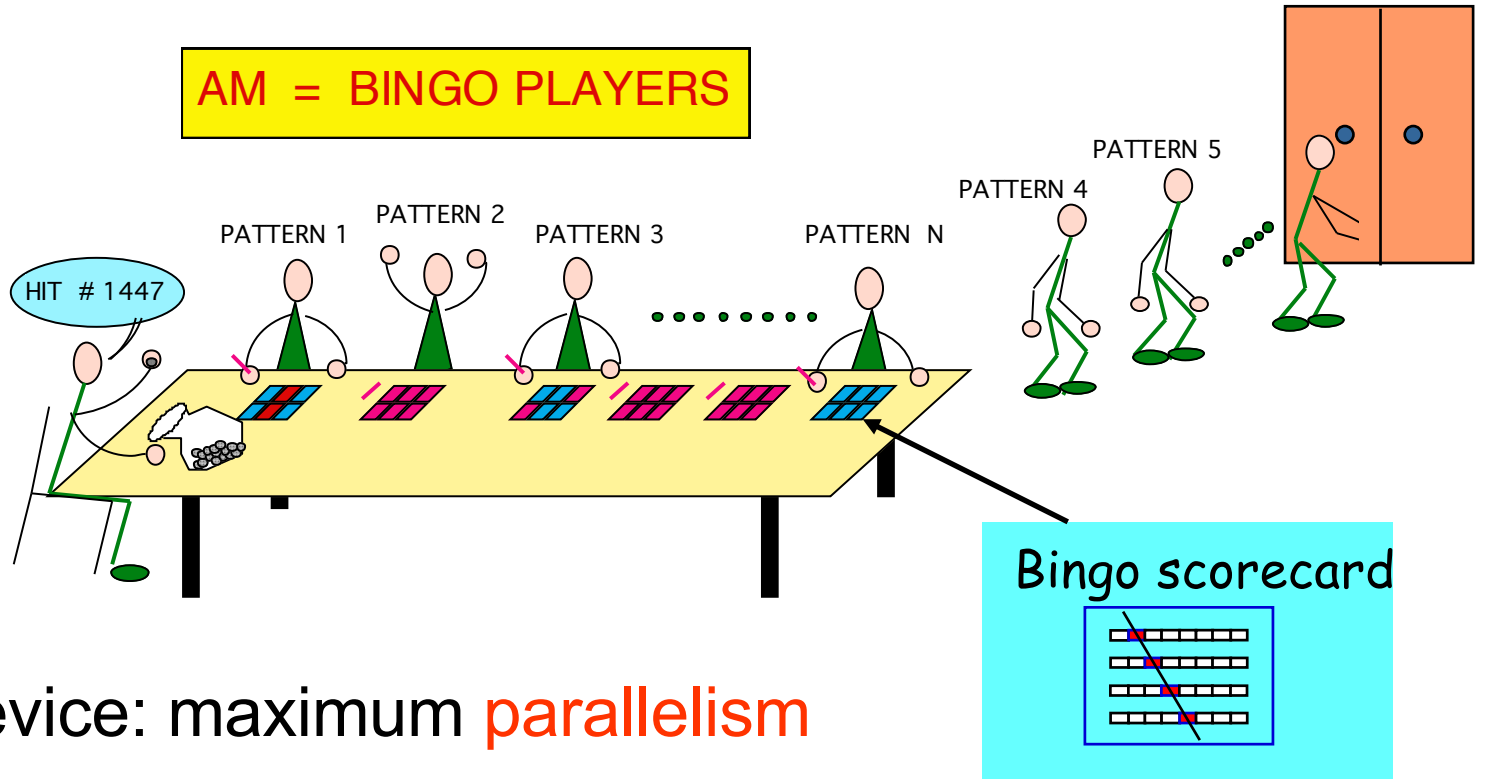


Pattern Matching







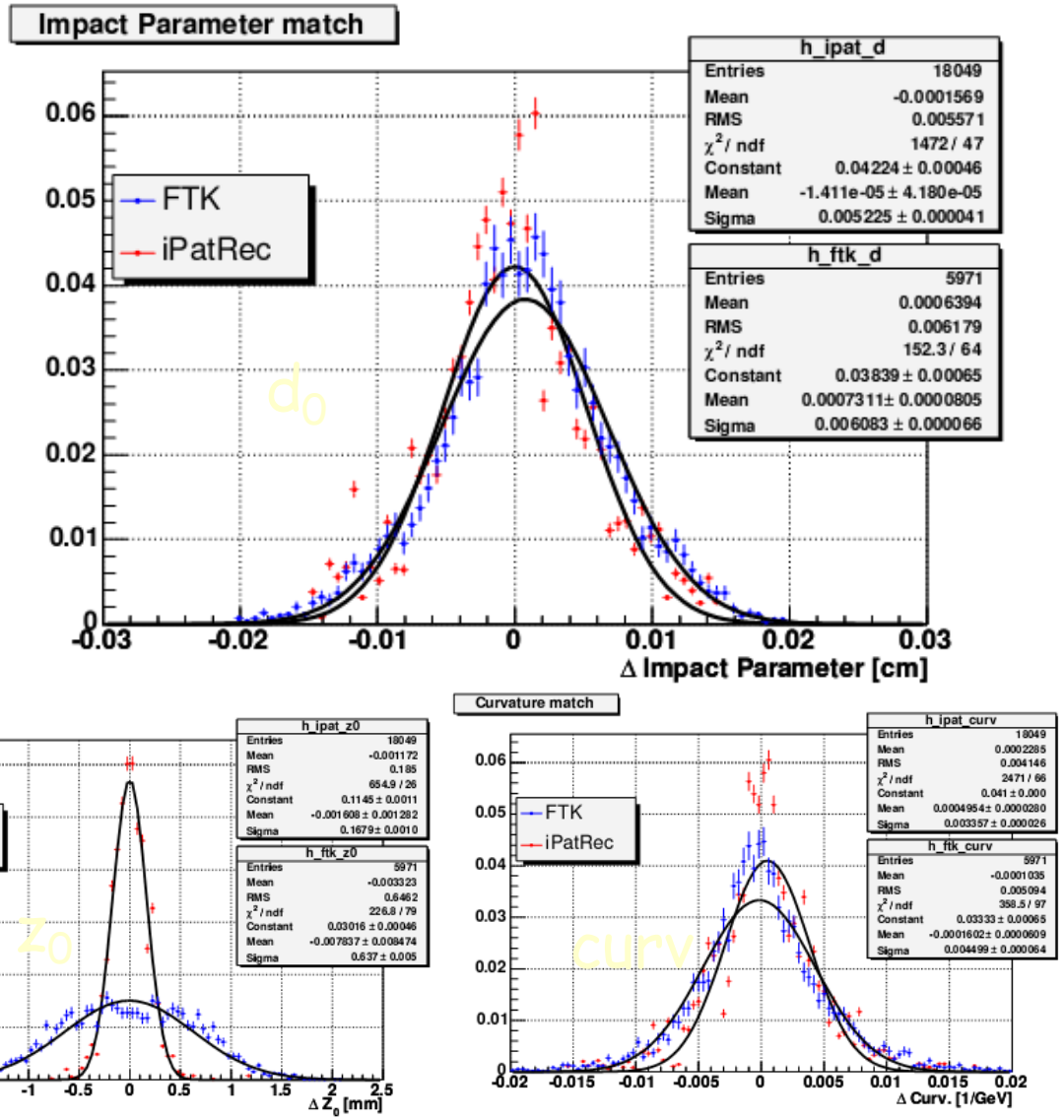


- Dedicated device: maximum parallelism
- Each pattern with private comparator
- Track search during detector readout

Final track fitting done by CPU, using CAM results

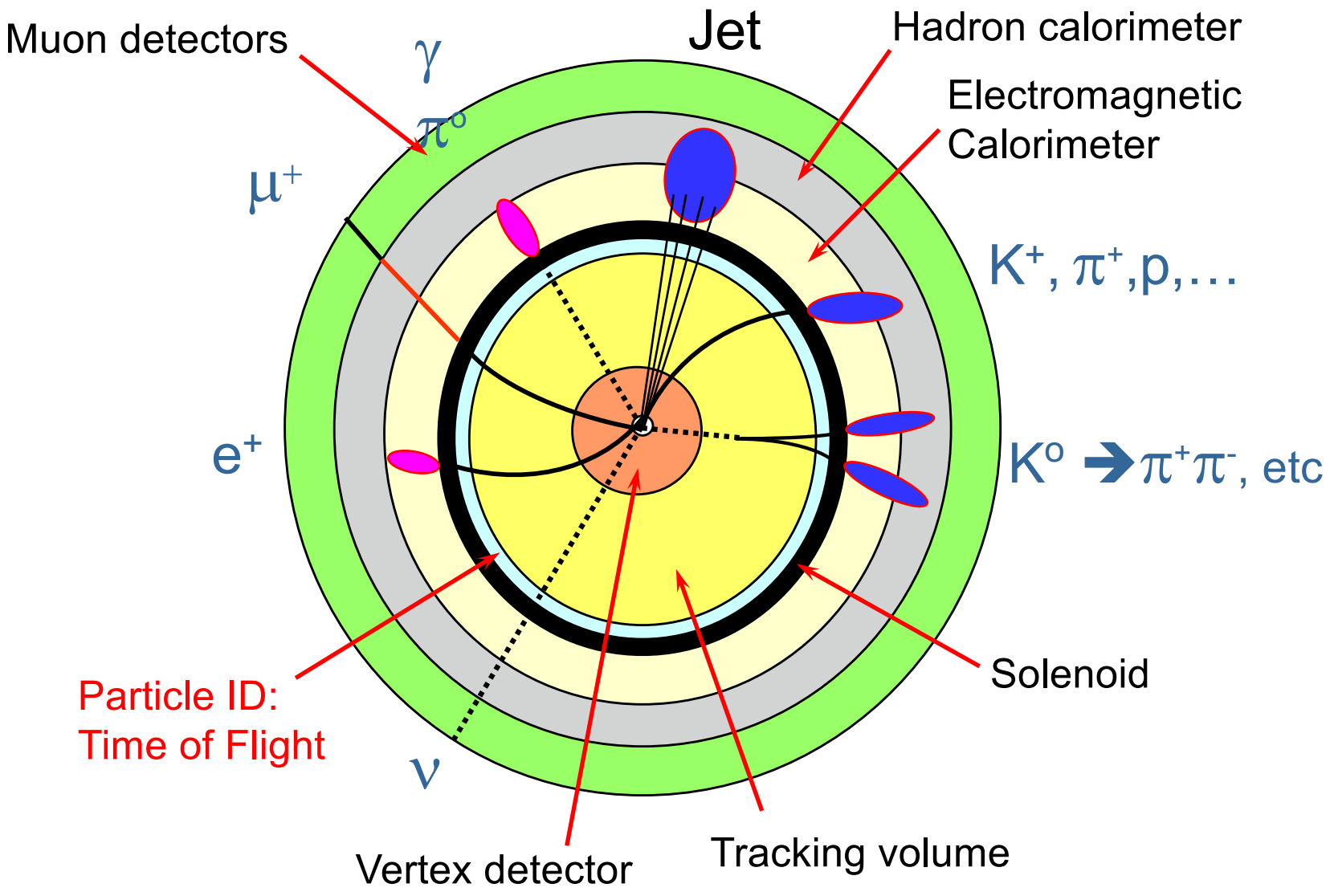
Track fitting (CPU)

- Final track fitting within narrow roads already found.
  - Linearized equations give good track parameters
  - All linear coefficients stored in hardware memory; fast!
- Tracks close to offline quality



Trigger architectures

Particle signatures



Triggering strategy

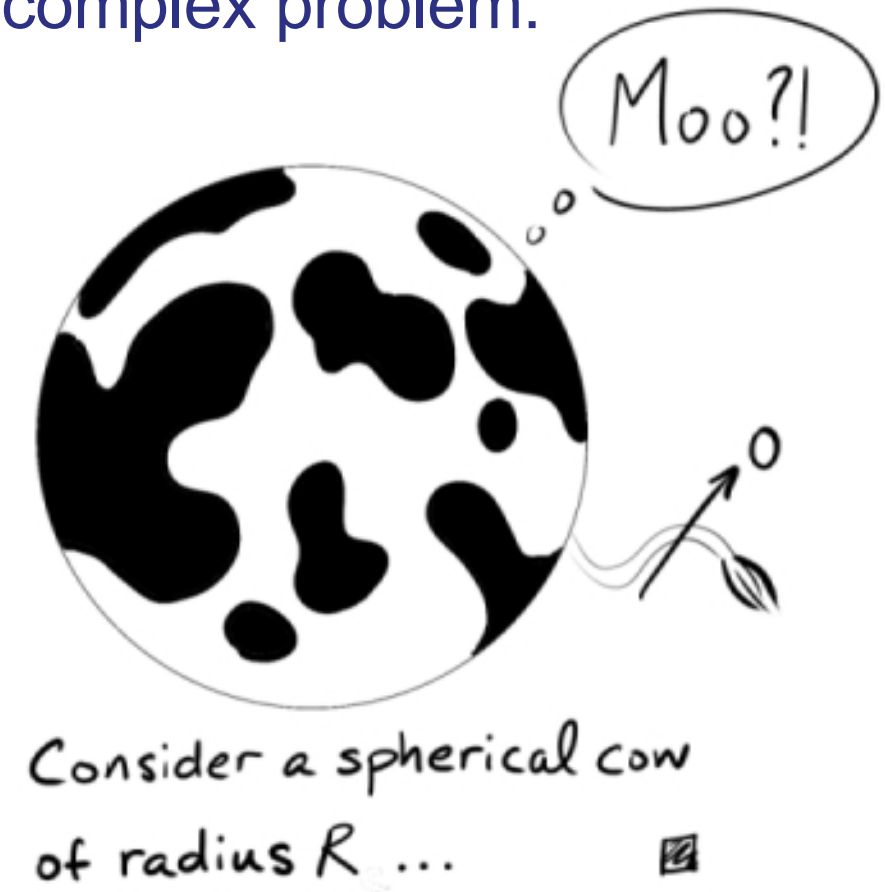
- Level-1
  - Fast, low-latency processing in hardware, based on simple algorithms:
    - Jets, isolated e, γ, τ, hadrons in calorimeter
    - Sum and missing ET
    - Muons
    - NEW: topology of Level 1 objects
- Level-2
  - More detailed processing in CPUs, with full detector readout available:
    - Refined analysis of Level-1 objects
    - Detector triggers not included in Level-1 (tracking, for example)
- Level-3 (event filter)
  - Physics-like analysis of full event

The challenge:

- Need to keep events with “interesting” features:
  - Isolated electrons and taus, jets, missing transverse energy, etc…
- HLT can perform detailed algorithms with full detector resolution and granularity
  - But Level-1 has limited resolution, and limited time to perform algorithms
- How do we look for physics-like signals in Level-1?

Use a simple approximation

Example of a simple approximation of a complex problem:



In ATLAS L1Calo:

- Rectangular electrons
- Square jets

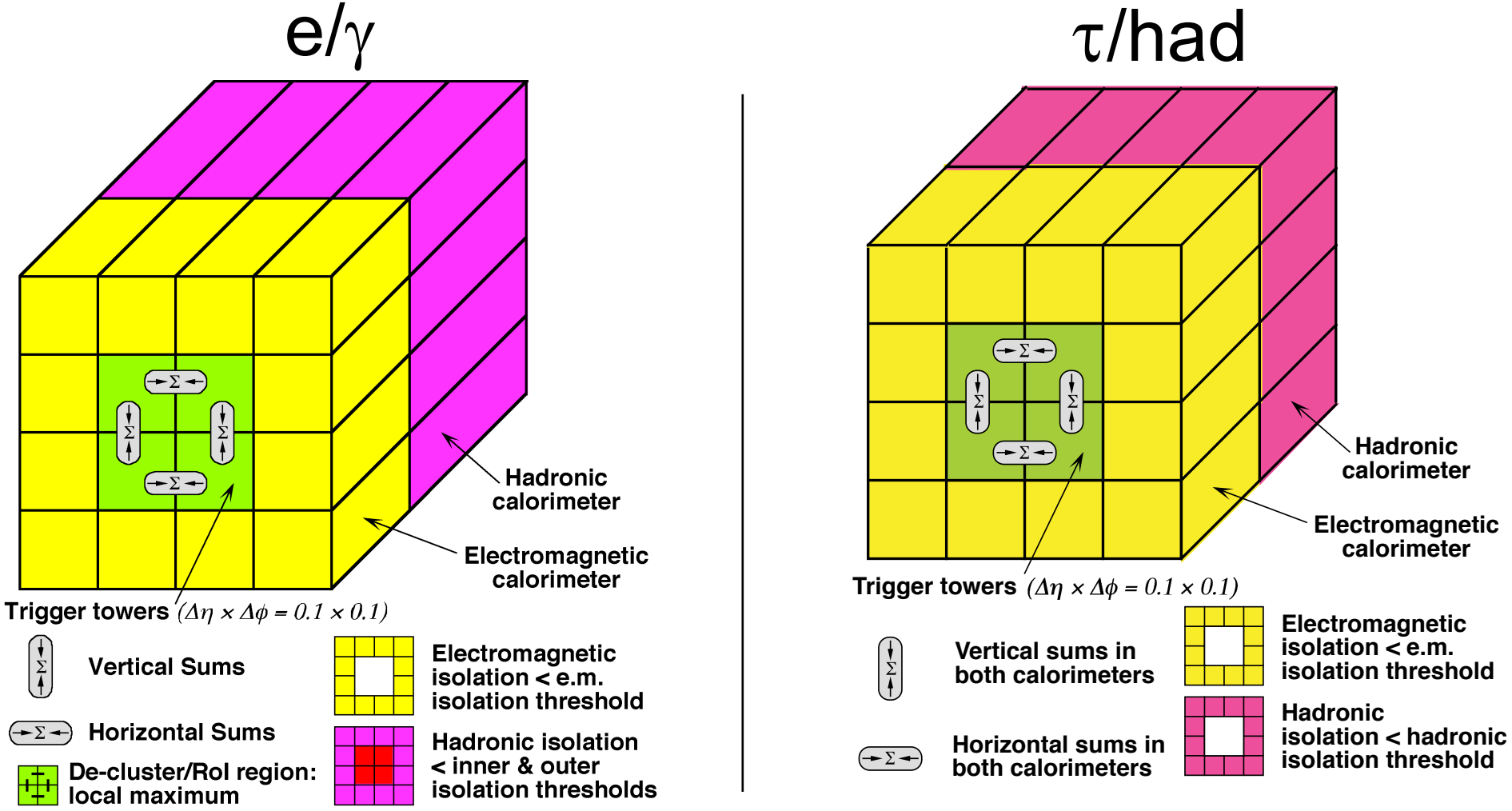
Simple trigger algorithms

- Electron/gamma
  - Local energy deposit in EM calorimeter
  - E/H: little/no energy in hadronic layer behind EM deposit
  - Isolation: little/no energy in surrounding cells
- Isolated tau/hadron
  - Similar to E/gamma, but with energy in both EM & hadronic layers
- Jets
  - Summed EM+hadronic energy in a defined area/radius, greater than a minimum threshold
- “Missing” energy
  - Vector sum of calorimeter energy deposits

Triggering involves trade-offs

- Real electrons and jets are not rectangular
  - But easier and faster in hardware
  - HLT can do a more realistic job later
    - If the event makes it that far
- Reality:
  - Over 99% of what Level-1 accepts is junk!
  - The job of Level-1 is to reduce rates
    - While not throwing away too much good physics
  - Latency/bandwidth/cost are major constraints
    - Do the “best we can” within these constraints

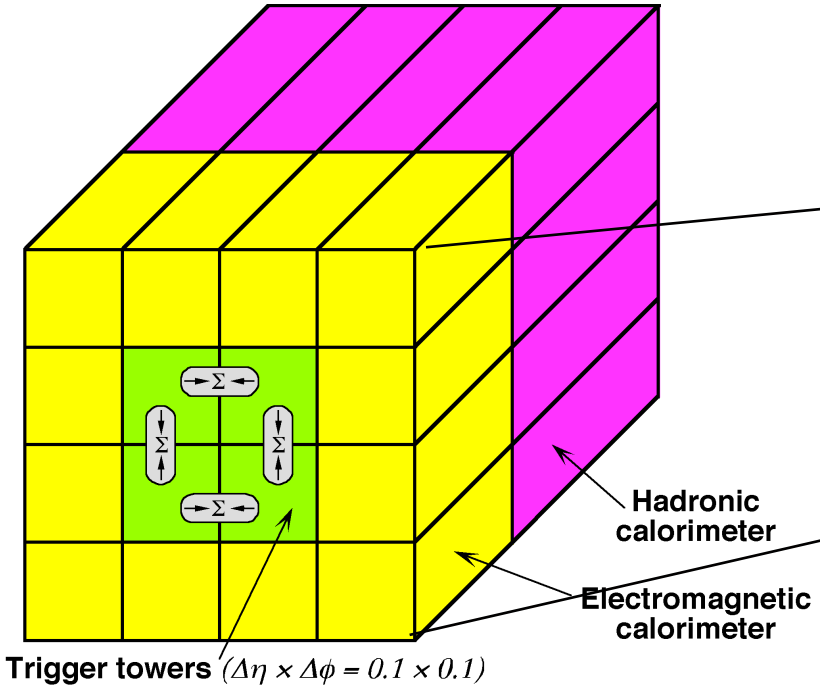
ATLAS e/γ and τ algorithms

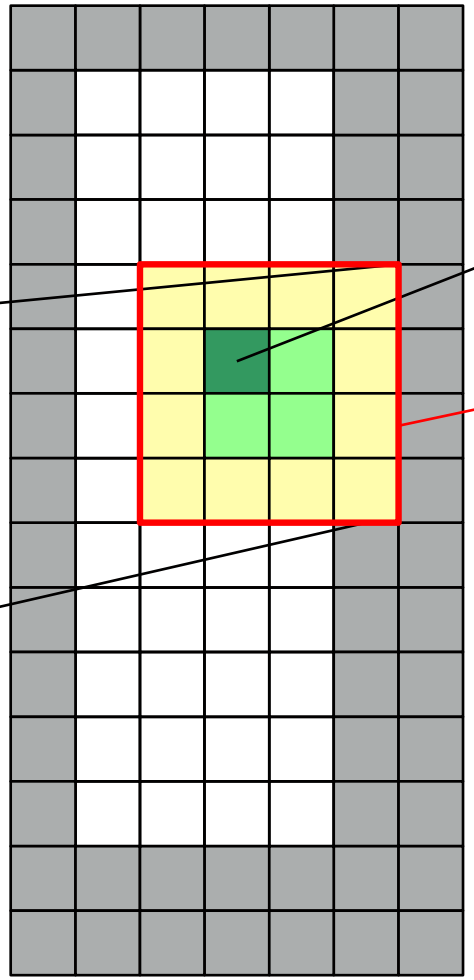


“Sliding window” EM algorithm

ATLAS CPM

EM cluster algorithm:





Cluster processor module

“Environment data sharing”

- Consider an algorithm with an n × n window
- If a single board processes j × k overlapping windows, then
  - Total towers needed: (j + n-1) × (k + n-1)
  - Example:
    - Atlas cluster algorithm has a 4 × 4 window
    - 64 windows per module: 16 × 4 “core” (64 towers)
    - Total towers needed: 19 × 7 “environment” (133 towers)
    - More than half of the towers must be duplicated from adjacent bords!

Optimizing an architecture

- A larger “core” to “environment” ratio makes more efficient use of each “processing unit”
  - Limited by the amount of data each unit (FPGA or board) can receive and process
- How to optimize an architecture
  - Maximize capacity of each processing unit:
    - Input bandwidth (link number and speed)
    - Dense algorithm processing logic
  - Basically this means using the largest affordable FPGAs, and high-speed serial data links
- Balanced strategy for system-wide data duplication and distribution

System partitioning

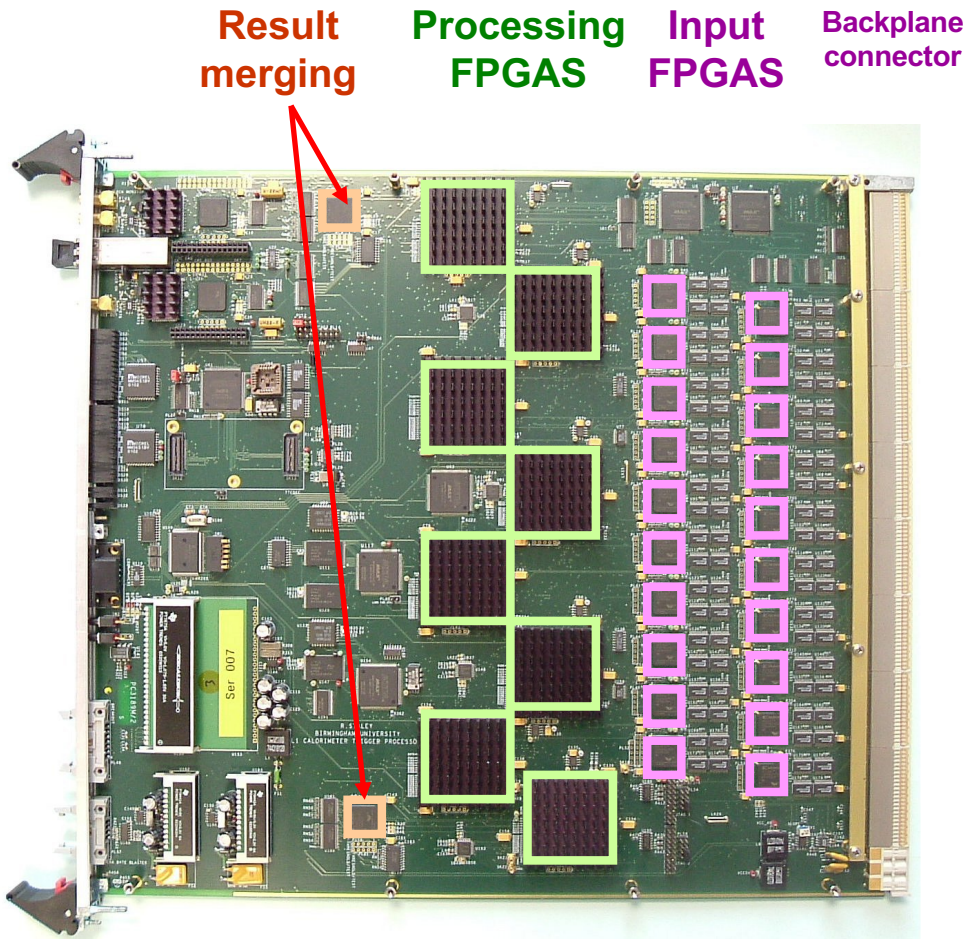
- Data sharing and distribution are the most important part of modern trigger designs
  - Modern FPGAs can do what you want with the data, as long as they can get it in!
- Modern system architectures based on:
  - Number of bits per trigger tower,
  - Input link speed (towers per link),
  - Number of links in largest affordable FPGA,
  - Number of links and FPGAs that can fit on a processor board, and
  - Number of boards that can fit in a crate

Data sharing and distribution

- In general:
  - Data fanout limited by available signal pins on boards and integrated circuits
  - Best to use high-speed serial data:
    - fewer pins/wires per data word
- Environment sharing on same board:
  - On-board duplication of input links, distribution to adjacent ICs (FPGAs)
  - Limited by board size/density/complexity
- Environment sharing between boards:
  - Backplane links, if in the same crate (maybe)
  - Duplicated signals at the source (high-speed links)

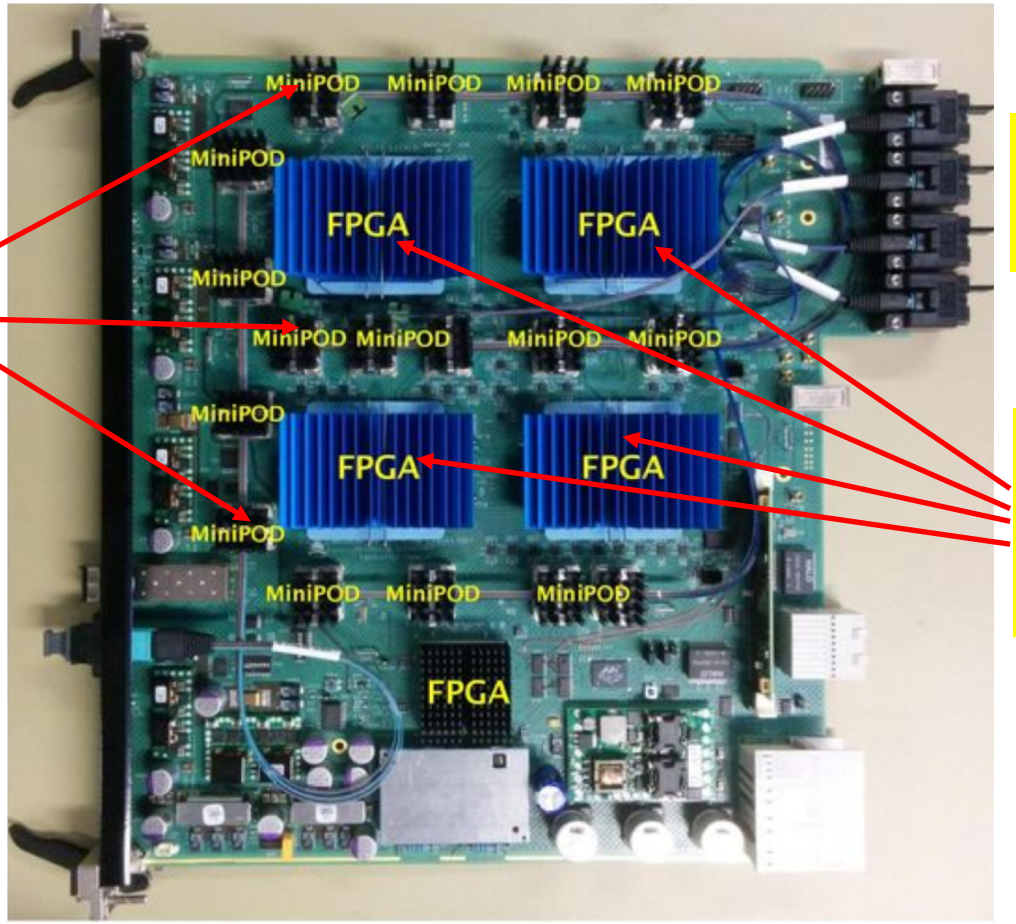
ATLAS L1Calo Cluster Processor Module (CPM) (currently running)

- Input Stage:
  - 20 FPGAs (XCV100E)
- Processing Stage:
  - 8 FPGAs (XCV1000E)
- Merging Stage:
  - 2 FPGAs (XCV100E)
- 18 layer PCB, minimum feature size 0.003"



ATLAS eFEX (Phase-I)

Similar angular coverage, but 7x EM granularity



Opto-electric Transceivers (MiniPOD)

Fiber optic connectors

Four large processor FPGAs

CMS e/γ algorithm

Trigger Primitive Generator

Fine grain — Flag Max of (, , ,) & Sum ET

Regional Calorimeter Trigger

E_T cut — + Max ( ) > Threshold

AND

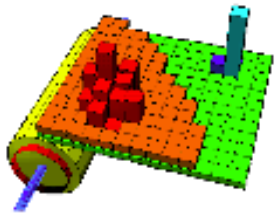
Longitudinal cut (H/E) — / < 0.05

AND

Isolation, Hadronic & EM — < 2 GeV

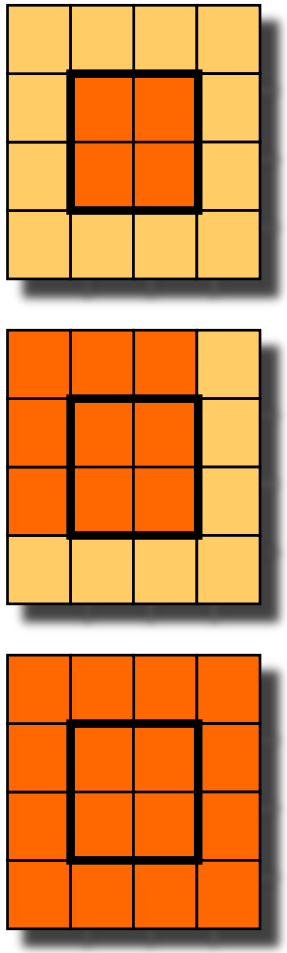
AND

One of (, , ,) < 1 GeV



ELECTRON or PHOTON

ATLAS Jet algorithm



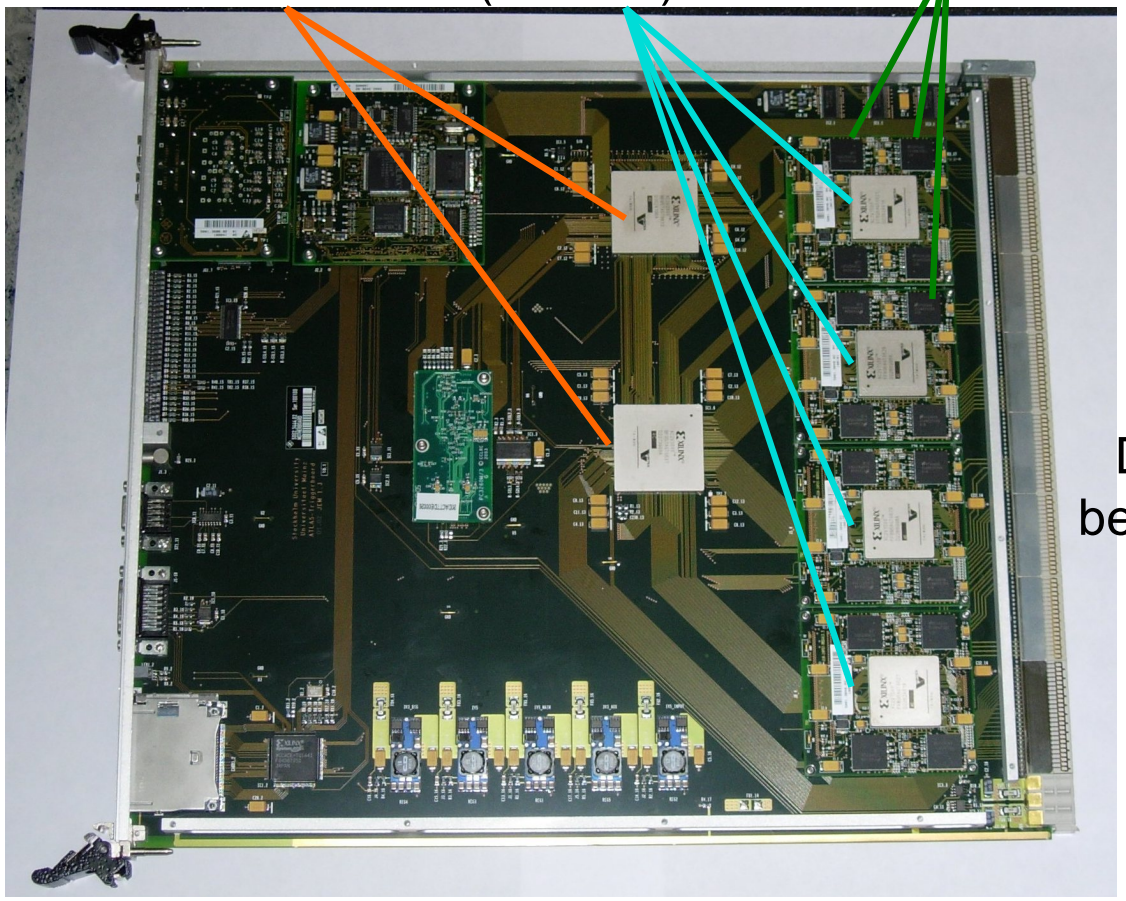
- Local maximum in a 0.4 x 0.4 window sliding by 0.2
- Compare transverse energy sum against thresholds for Jet window(s) surrounding the local maximum
  - Jet sizes: 0.4, 0.6, 0.8
  - 8 Jet definitions available:
    - Jet size
    - Et threshold

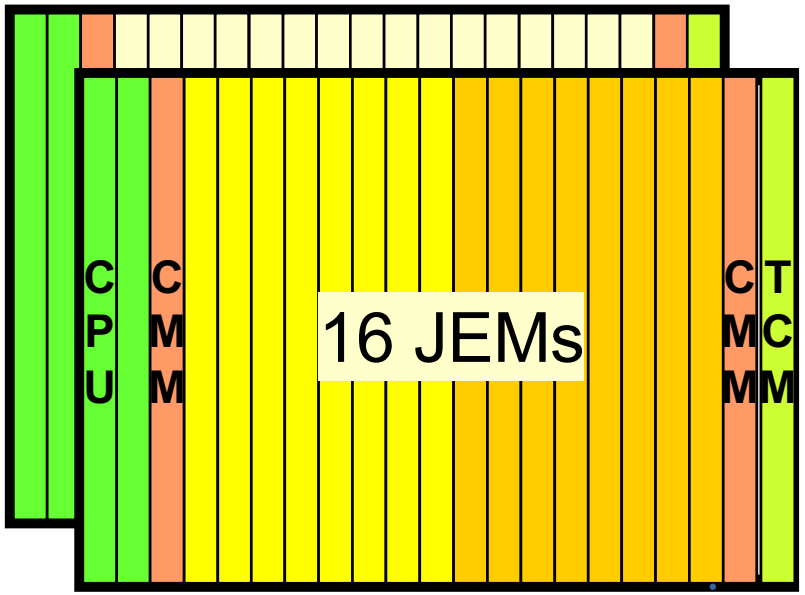
ATLAS Jet/Energy processor

Algorithm processing (FPGAs)

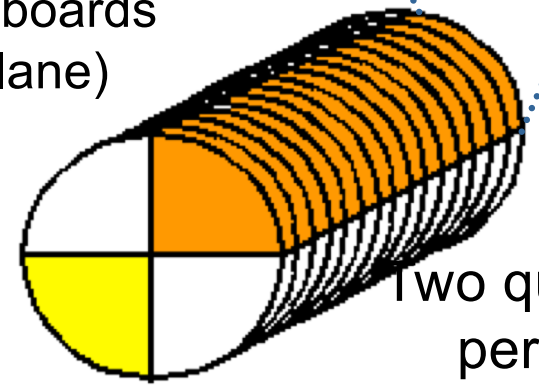
Input summing and distribution (FPGAs)

Digital input links



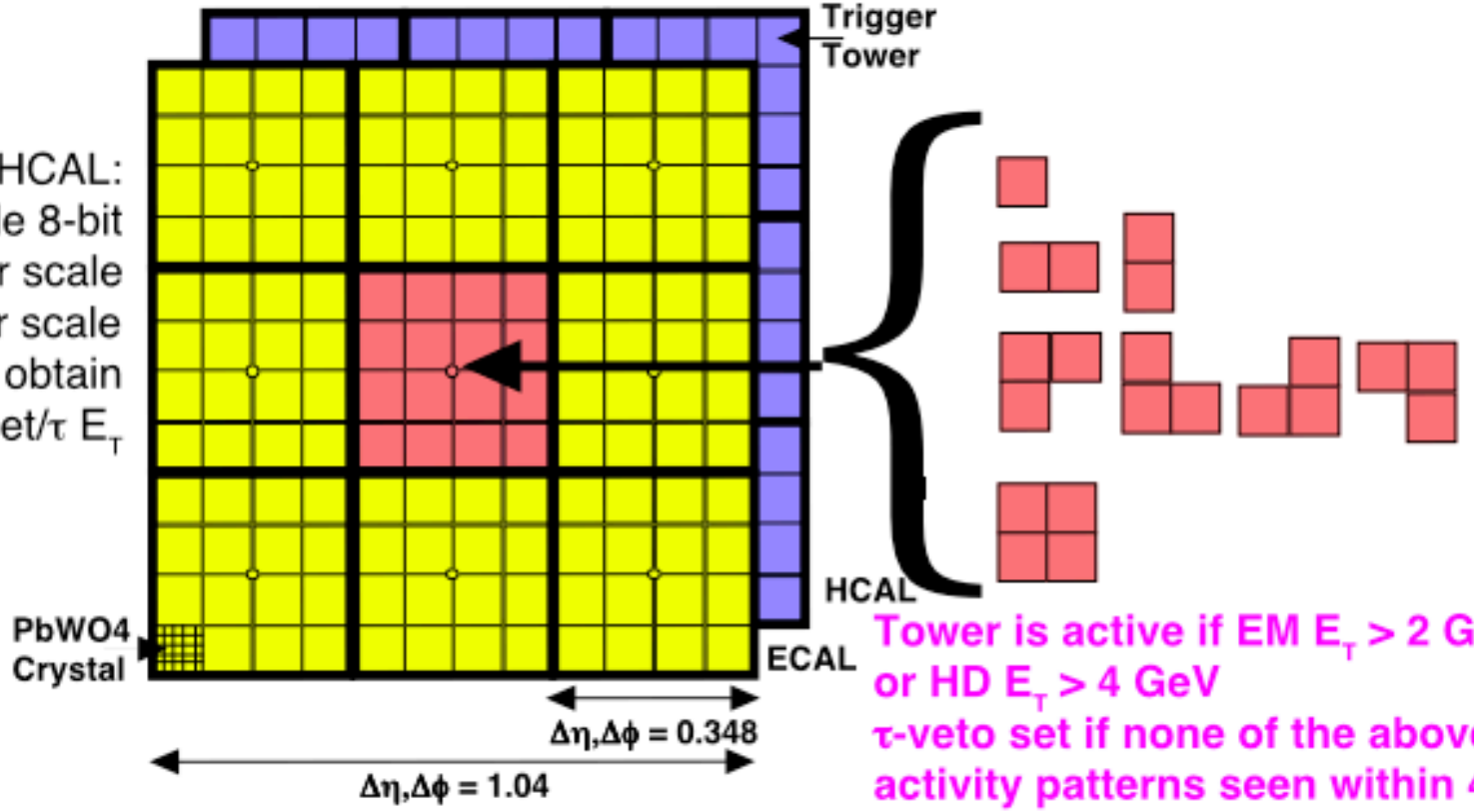


Data sharing between boards (backplane)



Two quadrants per crate

CMS τ / Jet algorithm



Tower is active if EM E_T > 2 GeV or HD E_T > 4 GeV
τ-veto set if none of the above activity patterns seen within 4x4

Jet or τ E_T

- 12x12 trigger tower E_T sums in 4x4 region steps with central region > others
- Larger trigger towers in HF but ~ same jet region size, 1.5 η x 1.0 φ

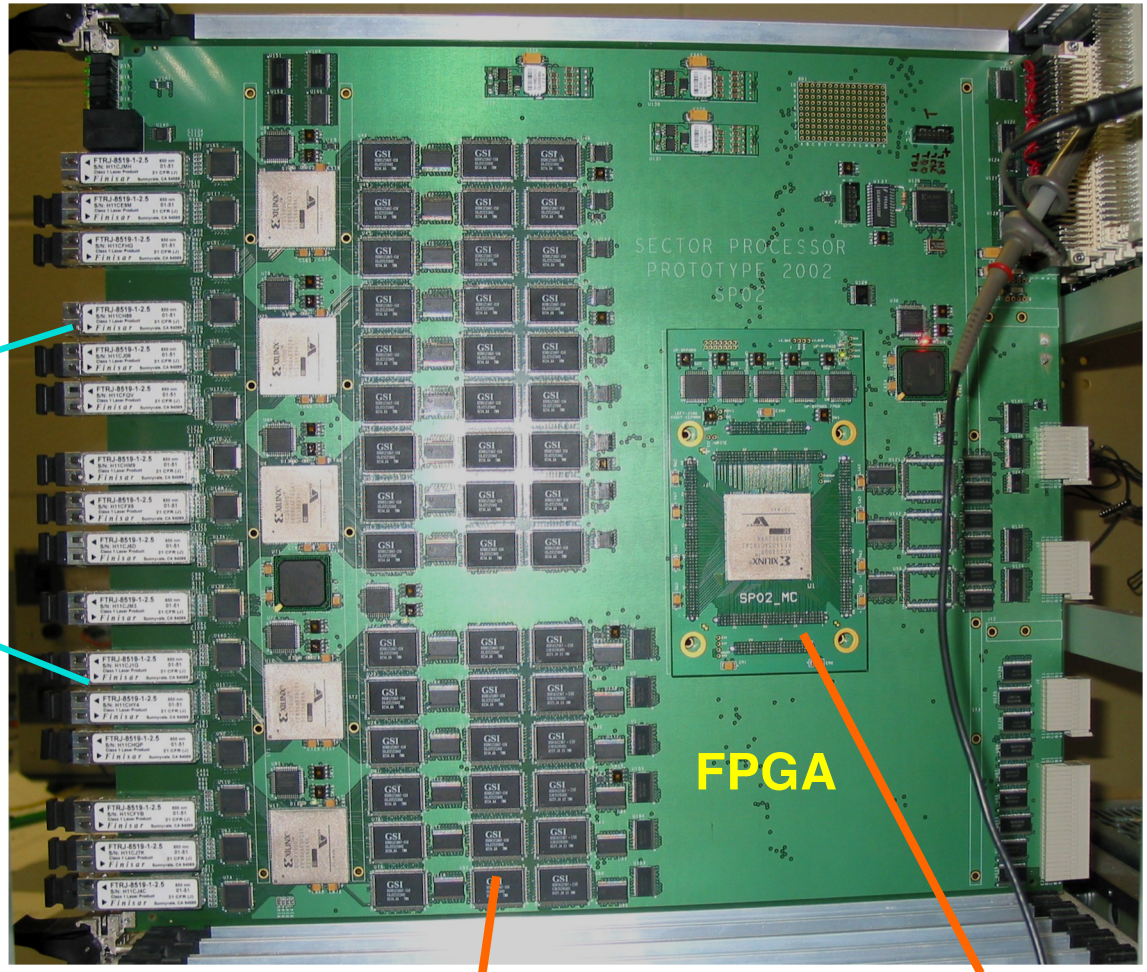
τ algorithm (isolated narrow energy deposits), within -2.5 < η < 2.5

- Redefine jet as τ jet if none of the nine 4x4 region τ-veto bits are on

Output

- Top 4 τ-jets and top 4 jets in central rapidity, and top 4 jets in forward rapidity

CMS L1 muon track finder



Input links
(Gbit optical)

Track assemblers
(memory LUTs)

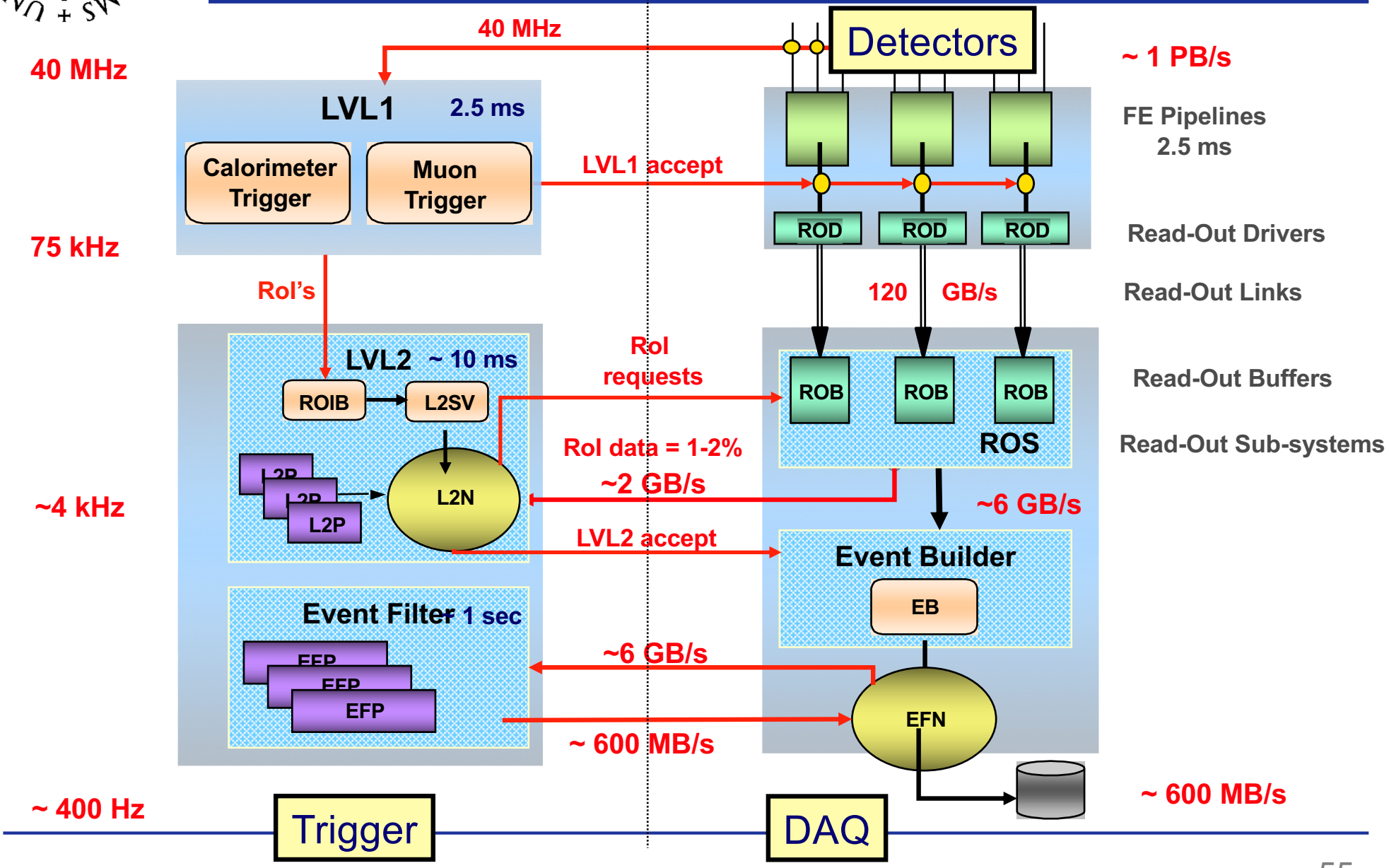
Final assembly
(FPGA)

CSC
Track-Finder

Higher level triggers
Event builder
Readout

(Some of) ATLAS HLT/DAQ

Trigger/DAQ (ATLAS)



Level-2 trigger

- Reduce rate from 75-100 kHz to few kHz
  - Basically by validating Level-1 triggers using full-resolution detector data (and track matching)
- Because the input rate is high, Level-2 needs to be resource-efficient
  - Limit analysis to objects identified in Level-1 ATLAS uses L1 Regions of Interest (RoI) to select “interesting” parts of the event data
  - Fast rejection of failed Level-2 validations to save CPU resources

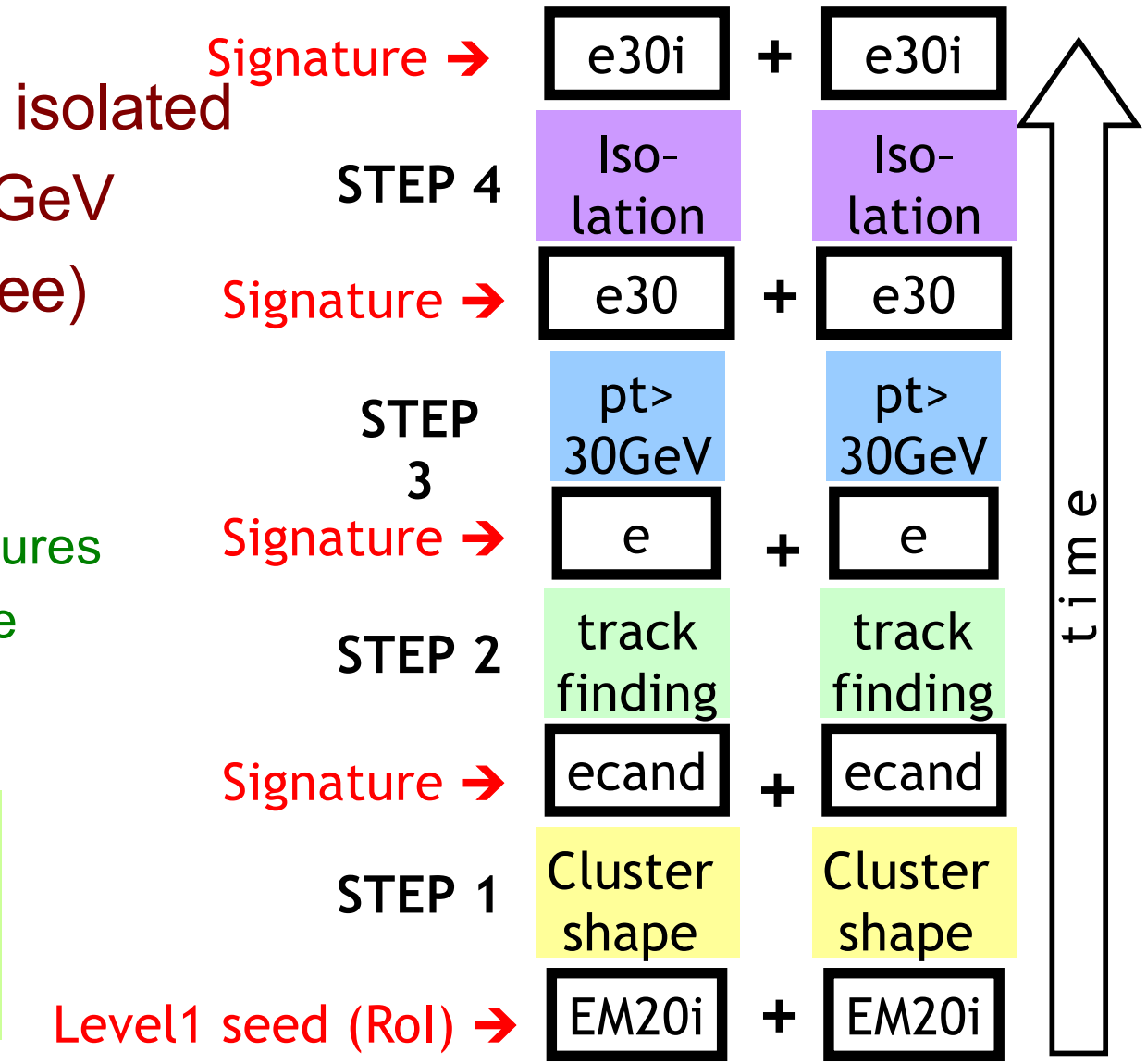
Level-2 example (2e)

Level-1 triggered on two isolated e/m clusters with pT>20GeV (possible signature: Z–>ee)

HLT Strategy:

- Validate step-by-step
- Check intermediate signatures
- Reject as early as possible

Sequential/modular approach allows early rejection



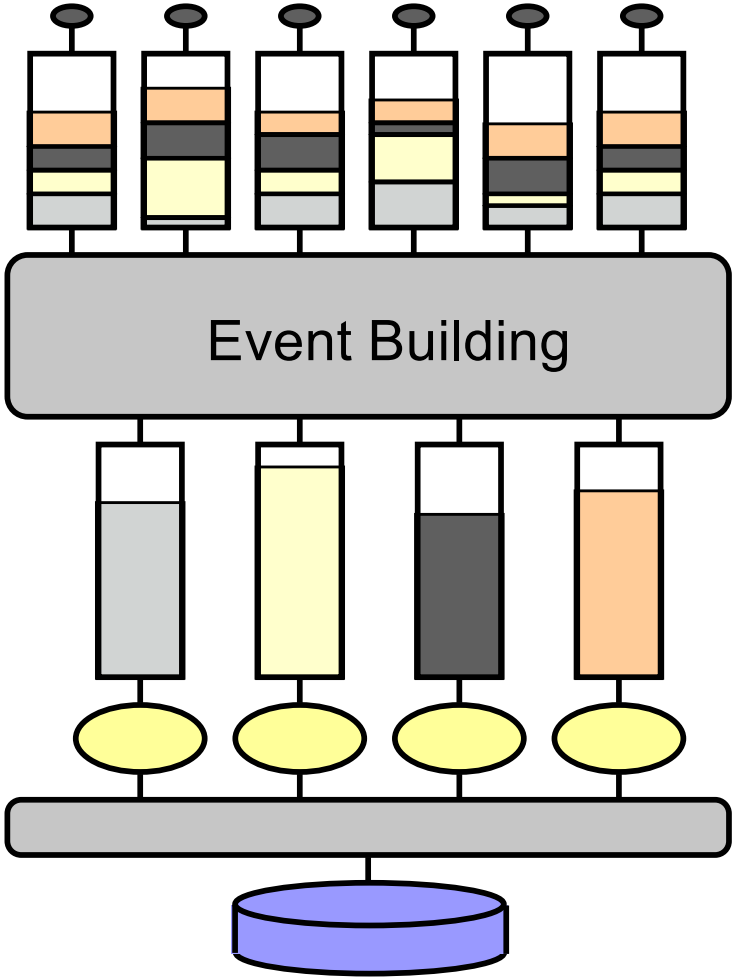
Event Builder (EB)

- After Level-2 validation, subdetector data from selected events are sent to the EB
- EB is essentially a large network that collects event fragments from different detectors and organizes them into a unified event structure
- Events are then passed to the Event Filter (Level-3) for “physics-like” analysis
- Event filter reduces rate to ~few hundred Hz, to be sent to storage.

Event Building to a CPU farm

Data sources

Event Fragments

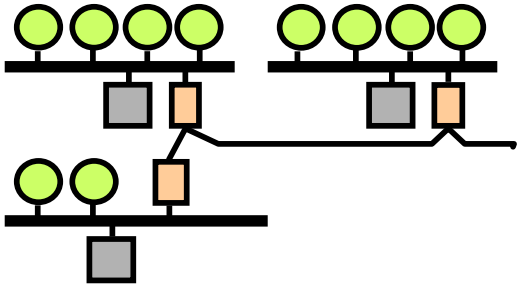


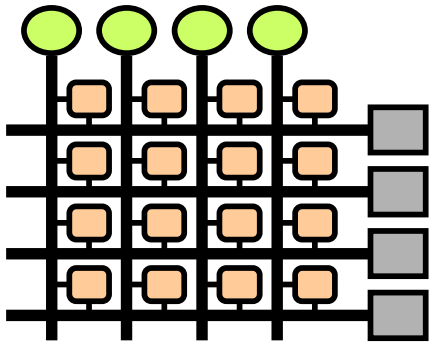
Full Events

Event filter CPUs

Data storage

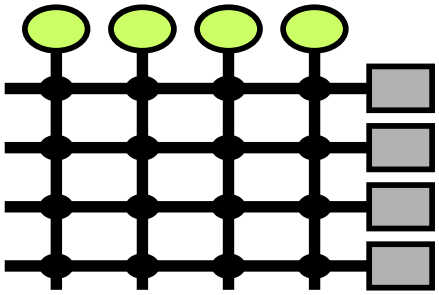
Event builder architectures





- **Time-shared Bus**
  - Most common at LEP (VME, Fastbus)
  - Bi-directional
  - Limited to maximum throughput of bus
  - Staged event building by independent buses in parallel (trees). No real gain, but reduces overhead.
- **Dual-port memory**
  - Event fragments are written in parallel and read sequentially by the destination processor.
  - Easy to implement. Commercially available. ex. D0, OPAL (VME/VSB)

Event builder architectures (2)



- **Cross bar switch**
  - Complete, non blocking interconnection all inputs/all outputs.
  - Ideal bandwidth efficiency.
  - N^2 crosspoints.
  - Control of the path routing:
    - External control (barrel shifter).
    - Auto-routing (by data). Data frame protocols.

- Switches vs. Buses
  - Total bandwidth of a Bus shared among all processors. Adding more processors degrades performance. Generally**, Buses do not scale** well.
  - With switches, N simultaneous transfers can co-exist. Adding more processors does not degrade performance (simply use a bigger switch). **Switches are scaleable**.

Event Building protocol

- Push protocol (LHCb)
  - Data are pushed to the destinations by the sources.
  - The source needs to know the destination address.
  - Assume sufficient data buffer at destination.
  - - No possibility to re-transmit an event fragment.
  - + Simple protocol.
- Pull protocol (ATLAS, CMS)
  - Data in the sources are pulled from the destinations.
  - Only buses can implement a pure pull protocol.
  - Sources indicate when data is ready. Destinations signal finished transfer (to free memory in source)
  - + Destinations can re-read the event fragments (correct errors)
  - - Heavier protocol.

Event Filter / Level-3

- ATLAS Event Filter reduce triggers to ~400 Hz
  - Event Filter latency budget ~ 4 sec average
- Full event detector data available, but to minimise resources needed:
  - Only unpack detector data when needed
  - Use Level-2 information to guide the process
  - Analysis proceeds in steps. Possibility to reject an event after each step
  - Use optimised offline algorithms

Bulk storage

Summary so far

- LHC trigger/DAQ systems are massive and complex!
- Different architecture approaches; depend both on individual needs of experiment and experience/preferences of teams that build them
- On the other hand....
  - Overall structures are similar
  - Mainly choose from the same basic building blocks for hardware algorithms, event building, etc.
- So...
  - Hopefully now easier to understand Trigger/DAQ systems
  - Experience from current systems important for future ones beginning with HL-LHC