ATLAS Fast TracKer

The first associative memory based hardware tracking system at LHC

11, November, 2014
PH-ESE Electronics Seminars, CERN

奥村恭幸
(OKUMURA, Yasuyuki)
University of Chicago

for ATLAS FTK team

Outline

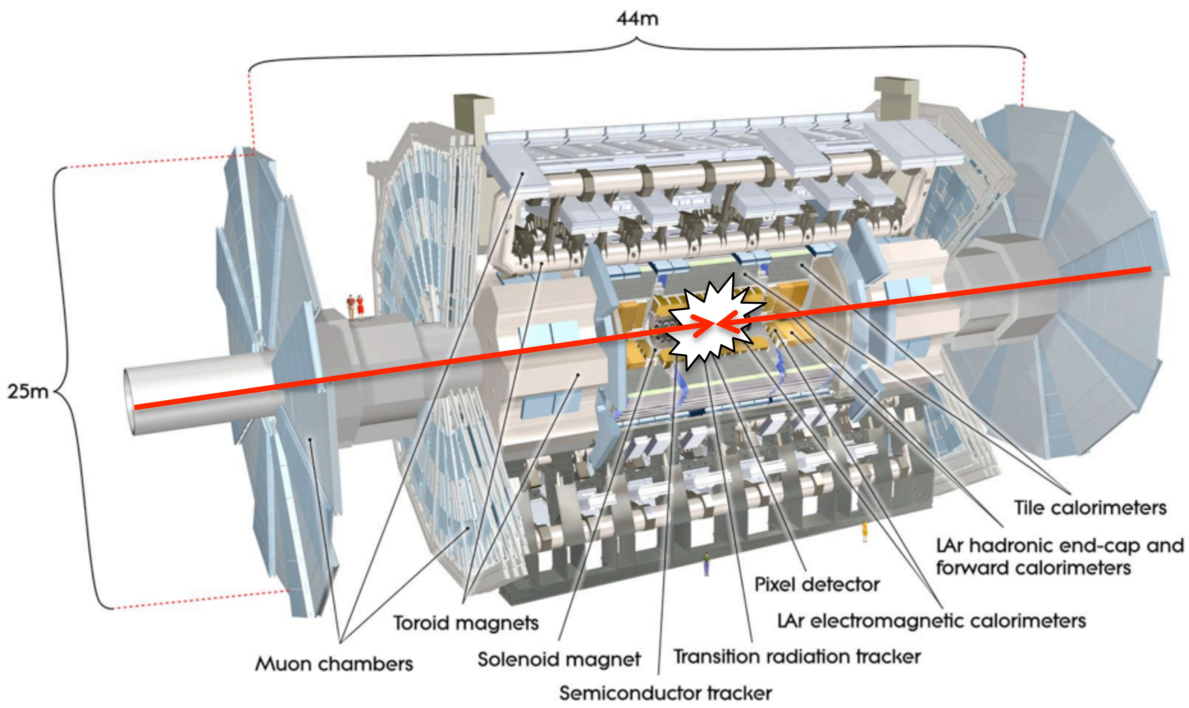
- Concept of associative memory approach for the hardware based tracking

- FTK system overview

- Details of technical challenges
  - ATCA based data formatter
  - Associative Memory implementation in FTK

LHC ATLAS experiment

Tracking :	Pixel, Silicon Strip, Transition Radiation Tracker
Calorimeter :	LAr & Scintillator
Muon :	Drift Tube, Resistive Plate Chamber, Thin Gap Chamber
(Magnets :	Solenoid (2T) & 3 Toroids (2Tm-8Tm))


44m
25m
Tile calorimeters
LAr hadronic end-cap and forward calorimeters
Pixel detector
LAr electromagnetic calorimeters
Toroid magnets
Muon chambers
Solenoid magnet
Transition radiation tracker
Semiconductor tracker

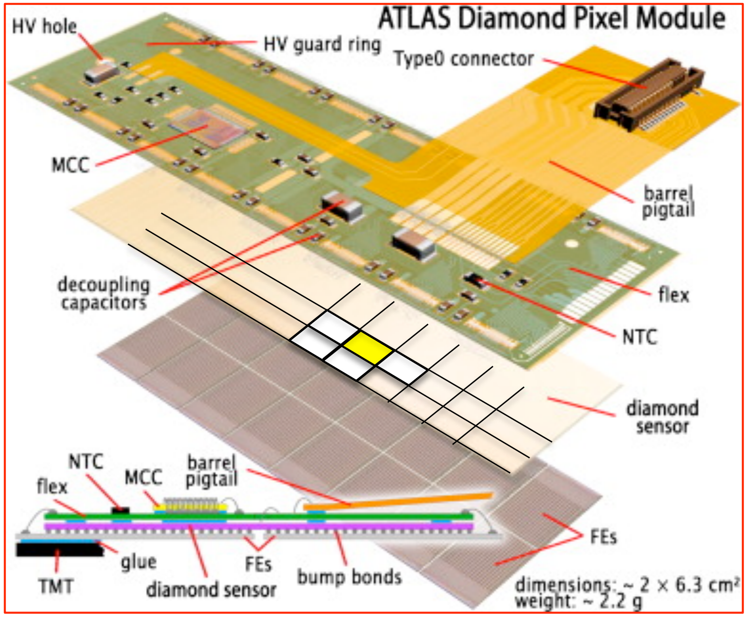
Objects Reconstruction

- ✓ electrons
- ✓ photons
- ✓ muons
- ✓ hadronic taus
- ✓ jets
- ✓ b-jets
- ✓ missing ET

Trigger

- ✓ Level 1 : 100kHz
- ✓ Higher Level Trigger (HLT) : 1kHz

Silicon detector at inner tracker

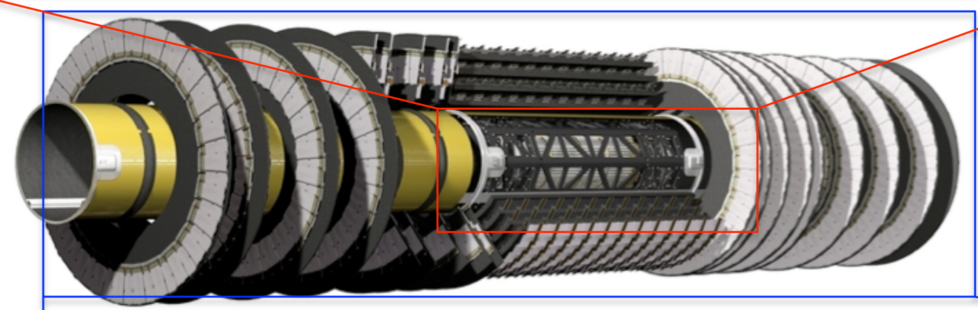


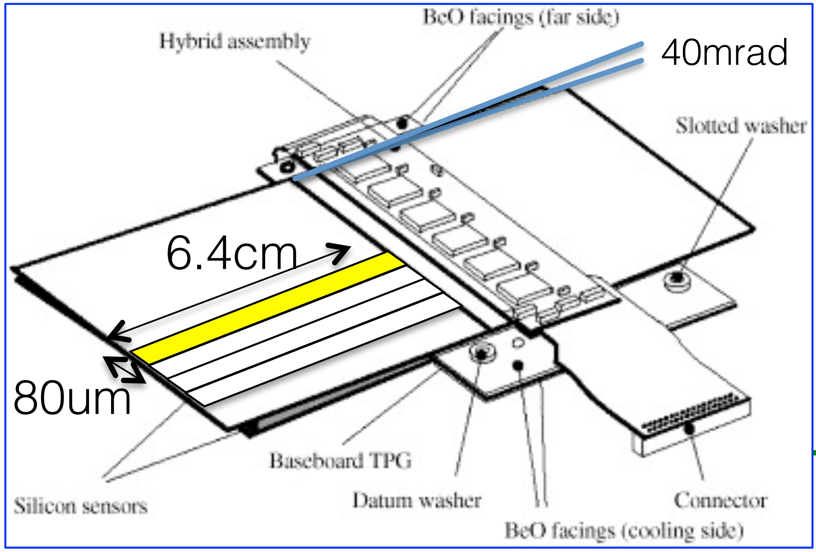
Silicon Pixel

✓ Pixel Size : 50um x 250um (400um, 600um)

✓ 4 Barrel Layers & 3 Endcap Disks

✓ ~90 Million Channels





Silicon Micro-strip

(Semi Conductor Tracker or SCT)

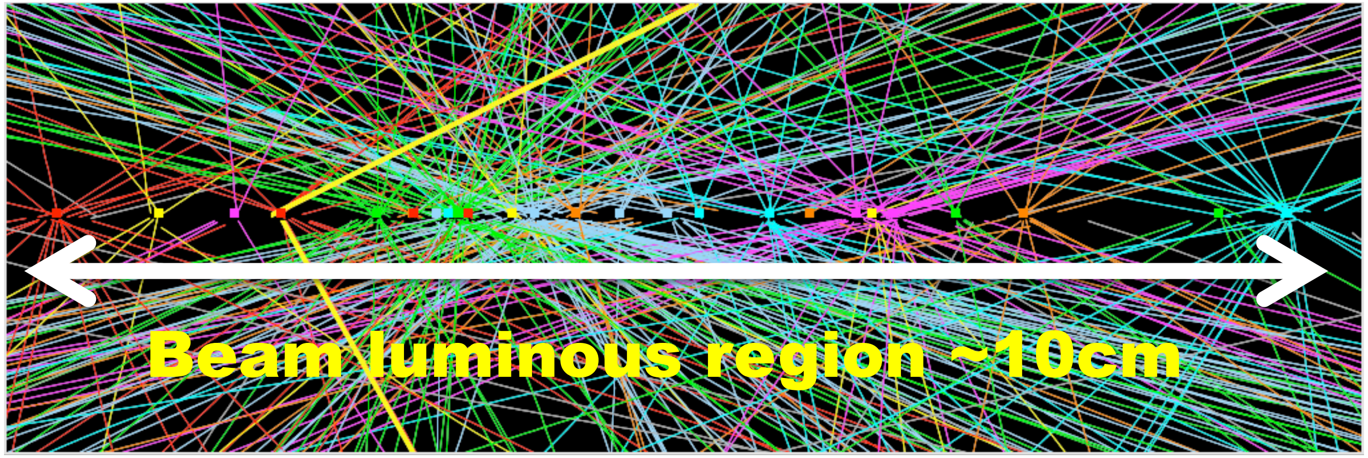
✓ Strip Width : 80um (Length 6.4cm)

✓ 4 Barrel Layers & 9 Endcap Disks

✓ Stereo angle : 40mrad

✓ 6.3 Million Channels

Physics application

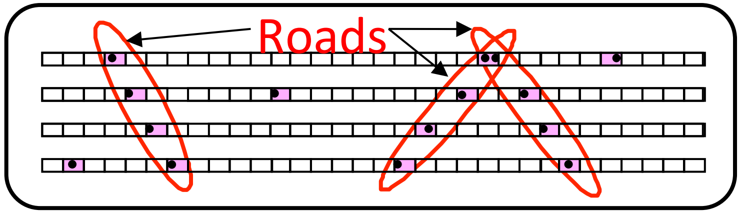


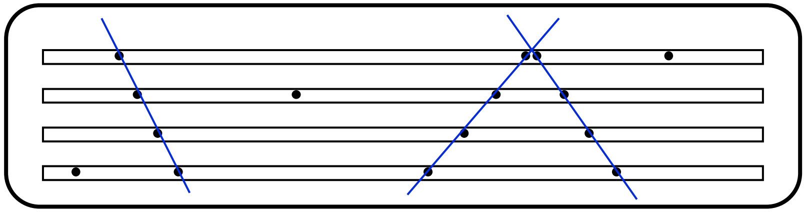
- Inner tracker is used for tracking & vertexing
  - Reconstruct and characterize charged particle trajectory
  - Vertex reconstruction, telling activities from interesting collisions
  - Input for particle identification (electron / muon / tau / b-jet)

Useful not only offline analysis but also triggering especially to maintain efficient physics analysis with higher luminosity in Run2 & 3

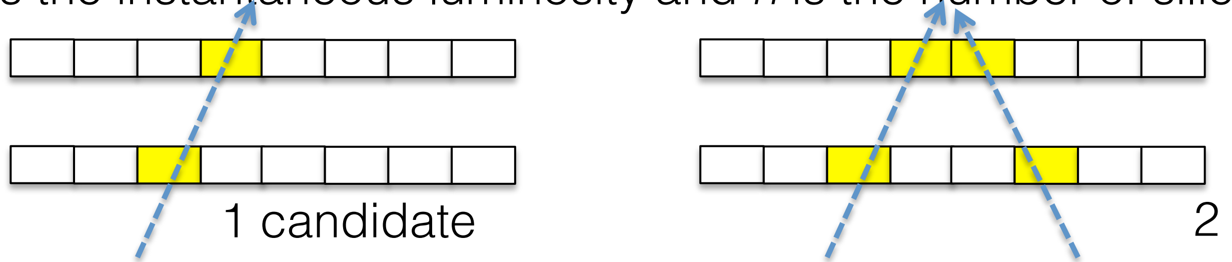
Fast track finding is challenging

- Tracking consists of two parts :
  - **Track finding** (or pattern recognition) with coarse resolution
  - **Track fitting for found patterns** with full hit resolution



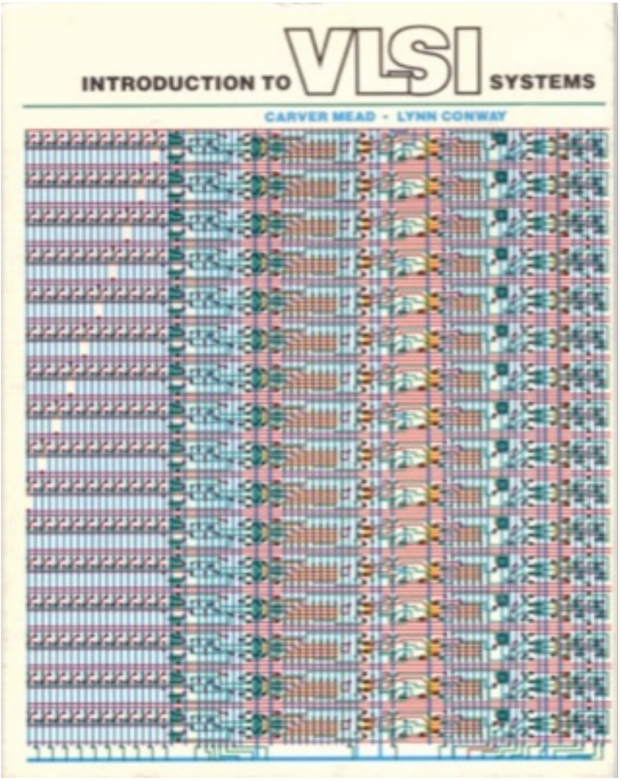


- Track finding in limited latency of the trigger is the major challenge
  - The number of hit combinations that have to be tested increases like L^n, where L is the instantaneous luminosity and n is the number of silicon layers

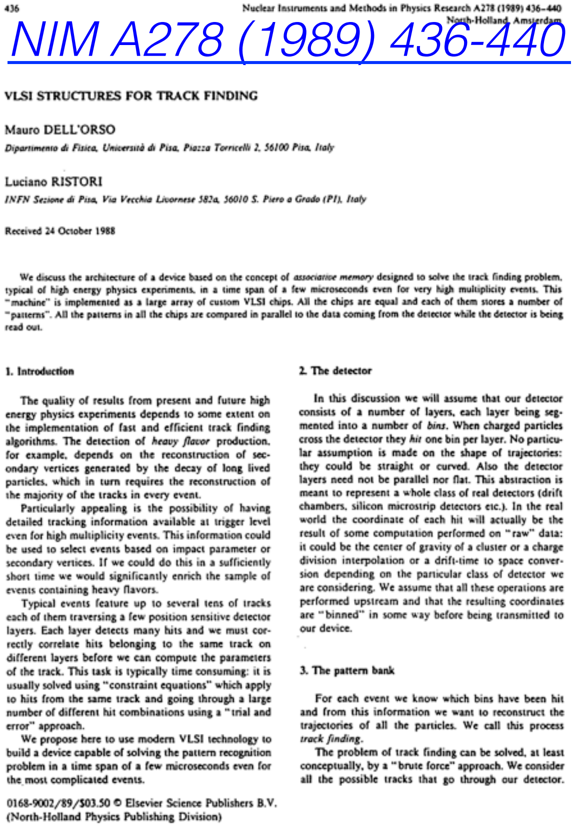


Associative memory approach allows fast track finding applicable to the triggers

Associative memory for track finding



NIM A278 (1989) 436-440



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.

1. Introduction

The quality of results from present and future high energy physics experiments depends to some extent on the implementation of fast and efficient track finding algorithms. The detection of *heavy flavor* production, for example, depends on the reconstruction of secondary vertices generated by the decay of long lived particles, which in turn requires the reconstruction of the majority of the tracks in every event.

Particularly appealing is the possibility of having detailed tracking information available at trigger level even for high multiplicity events. This information could be used to select events based on impact parameter or secondary vertices. If we could do this in a sufficiently short time we would significantly enrich the sample of events containing heavy flavors.

Typical events feature up to several tens of tracks each of them traversing a few position sensitive detector layers. Each layer detects many hits and we must correctly correlate hits belonging to the same track on different layers before we can compute the parameters of the track. This task is typically time consuming: it is usually solved using "constraint equations" which apply to hits from the same track and going through a large number of different hit combinations using a "trial and error" approach.

We propose here to use modern VLSI technology to build a device capable of solving the pattern recognition problem in a time span of a few microseconds even for the most complicated events.

0168-9002/89/$03.50 © Elsevier Science Publishers B.V.
(North-Holland Physics Publishing Division)

2. The detector

In this discussion we will assume that our detector consists of a number of layers, each layer being segmented into a number of *bins*. When charged particles cross the detector they *hit* one bin per layer. No particular assumption is made on the shape of trajectories: they could be straight or curved. Also the detector layers need not be parallel nor flat. This abstraction is meant to represent a whole class of real detectors (drift chambers, silicon microstrip detectors etc.). In the real world the coordinate of each hit will actually be the result of some computation performed on "raw" data: it could be the center of gravity of a cluster or a charge division interpolation or a drift-time to space conversion depending on the particular class of detector we are considering. We assume that all these operations are performed upstream and that the resulting coordinates are "binned" in some way before being transmitted to our device.

3. The pattern bank

For each event we know which bins have been hit and from this information we want to reconstruct the trajectories of all the particles. We call this process *track finding*.

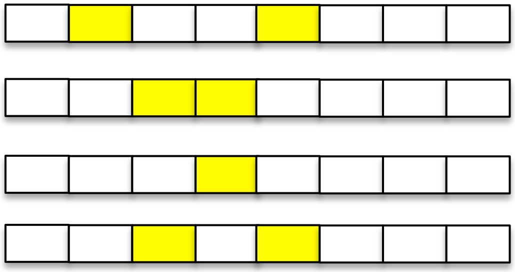
The problem of track finding can be solved, at least conceptually, by a "brute force" approach. We consider all the possible tracks that go through our detector.

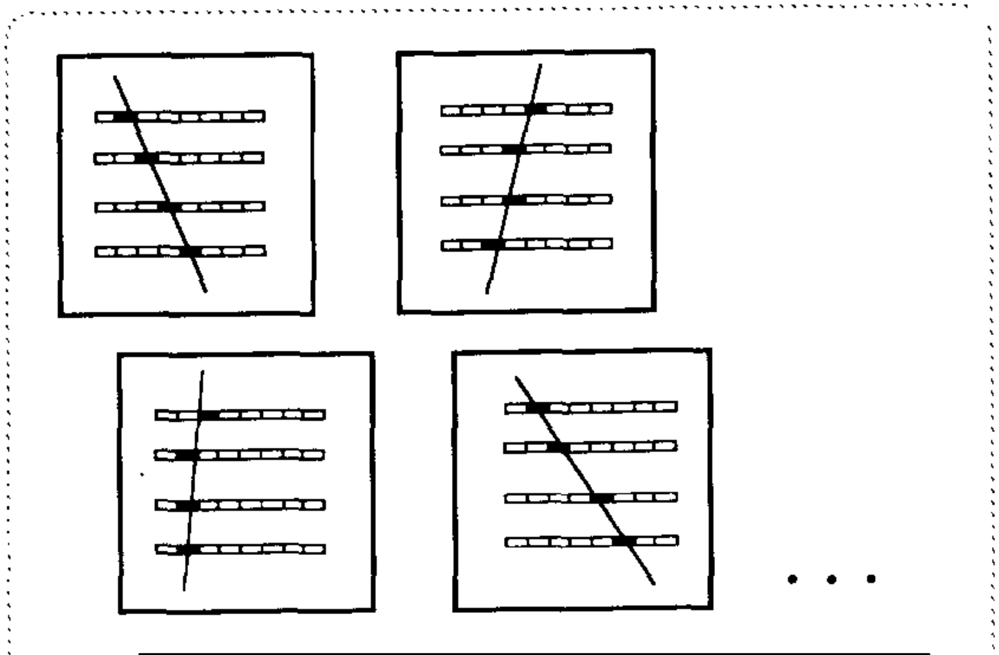
Original idea proposed in 1980's

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.

Concept of associative memory

- Comparison between predefined hit pattern for tracks and detected hit pattern

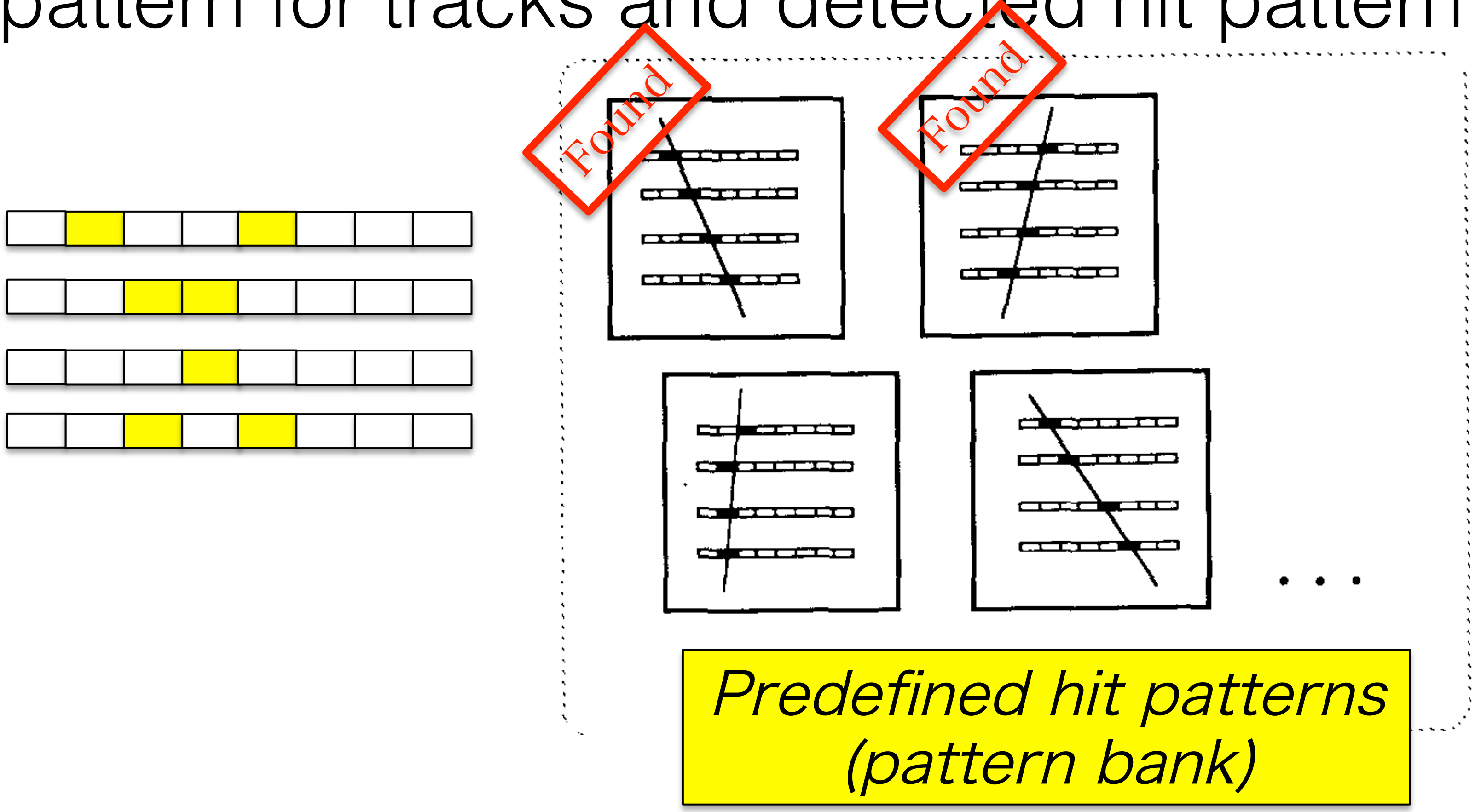




Predefined hit patterns (pattern bank)

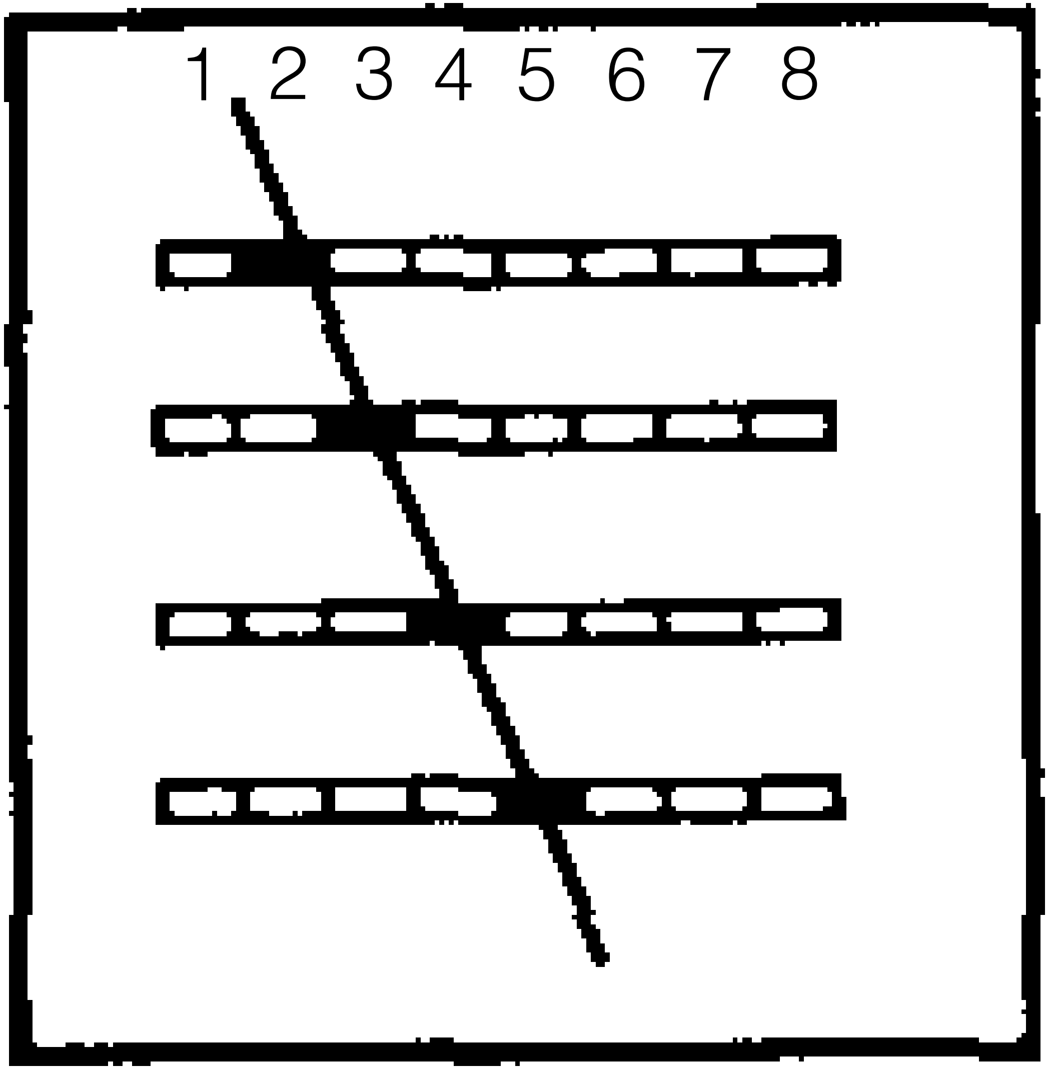
Concept of associative memory

- Comparison between predefined hit pattern for tracks and detected hit pattern



Predefined hit patterns (pattern bank)

Track Pattern "Bank"



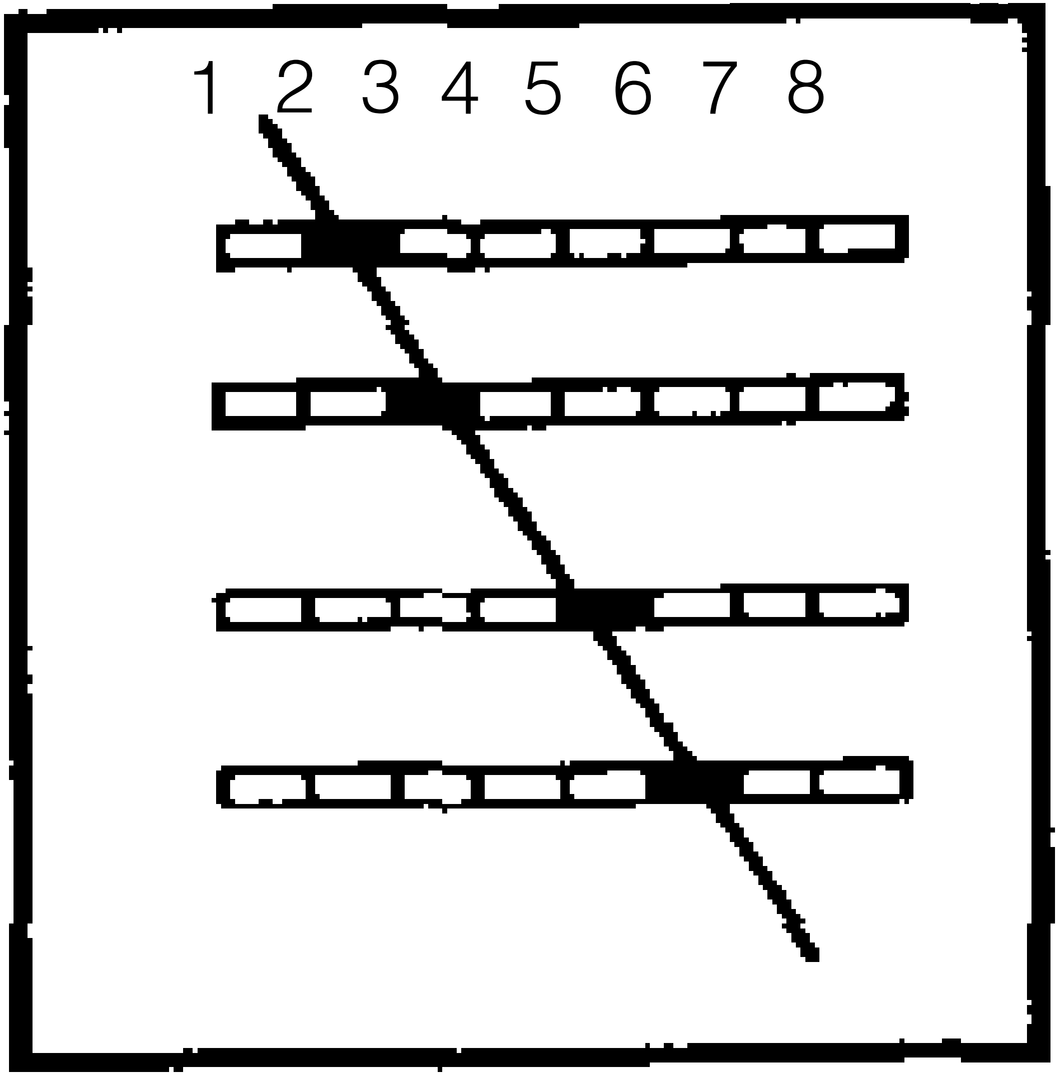
2 @ Layer4

3 @ Layer3

4 @ Layer2

5 @ Layer1

Track Pattern “Bank”



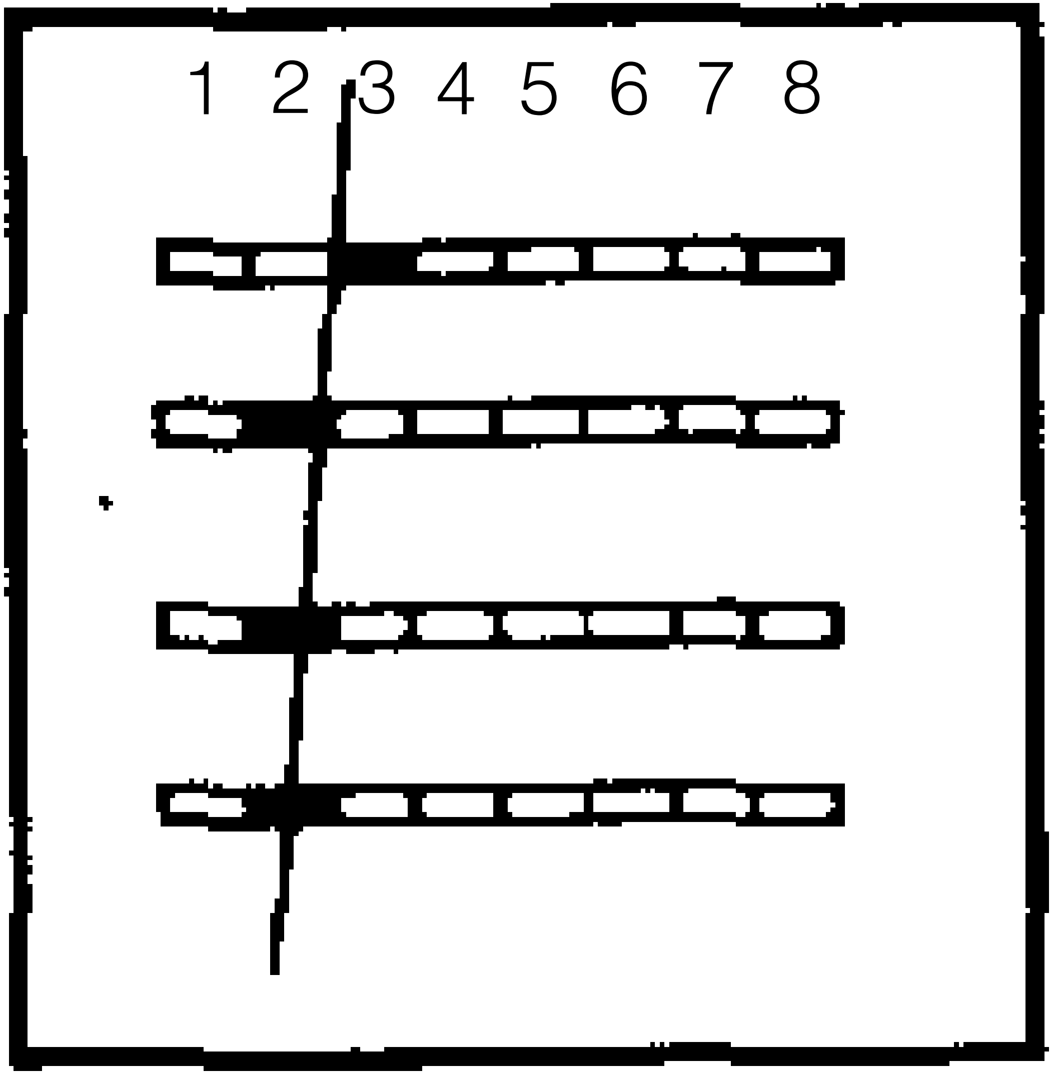
2 @ Layer4

3 @ Layer3

5 @ Layer2

6 @ Layer1

Track Pattern "Bank"



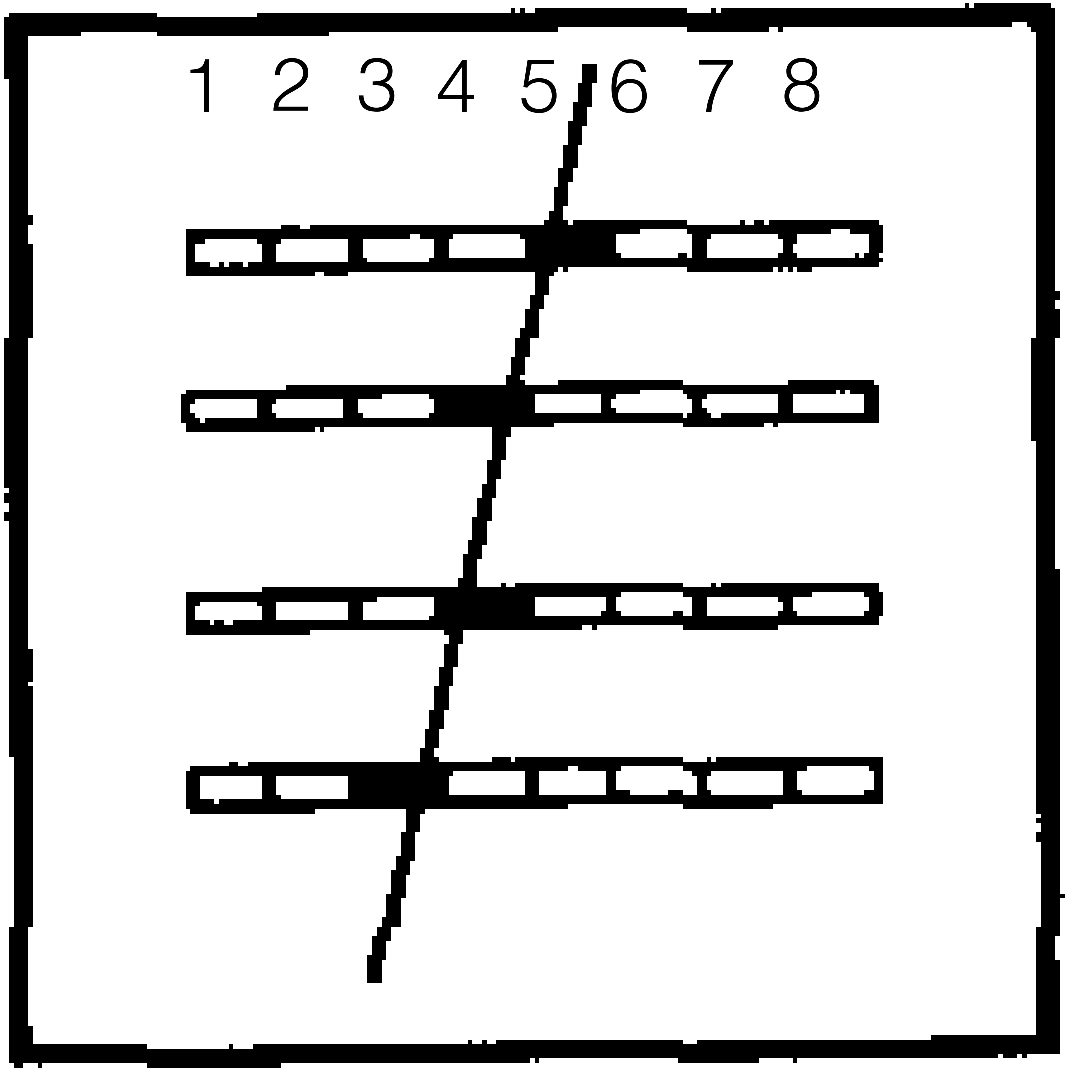
3 @ Layer4

2 @ Layer3

2 @ Layer2

2 @ Layer1

Track Pattern "Bank"



5 @ Layer4

4 @ Layer3

4 @ Layer2

3 @ Layer1

Associative Memory

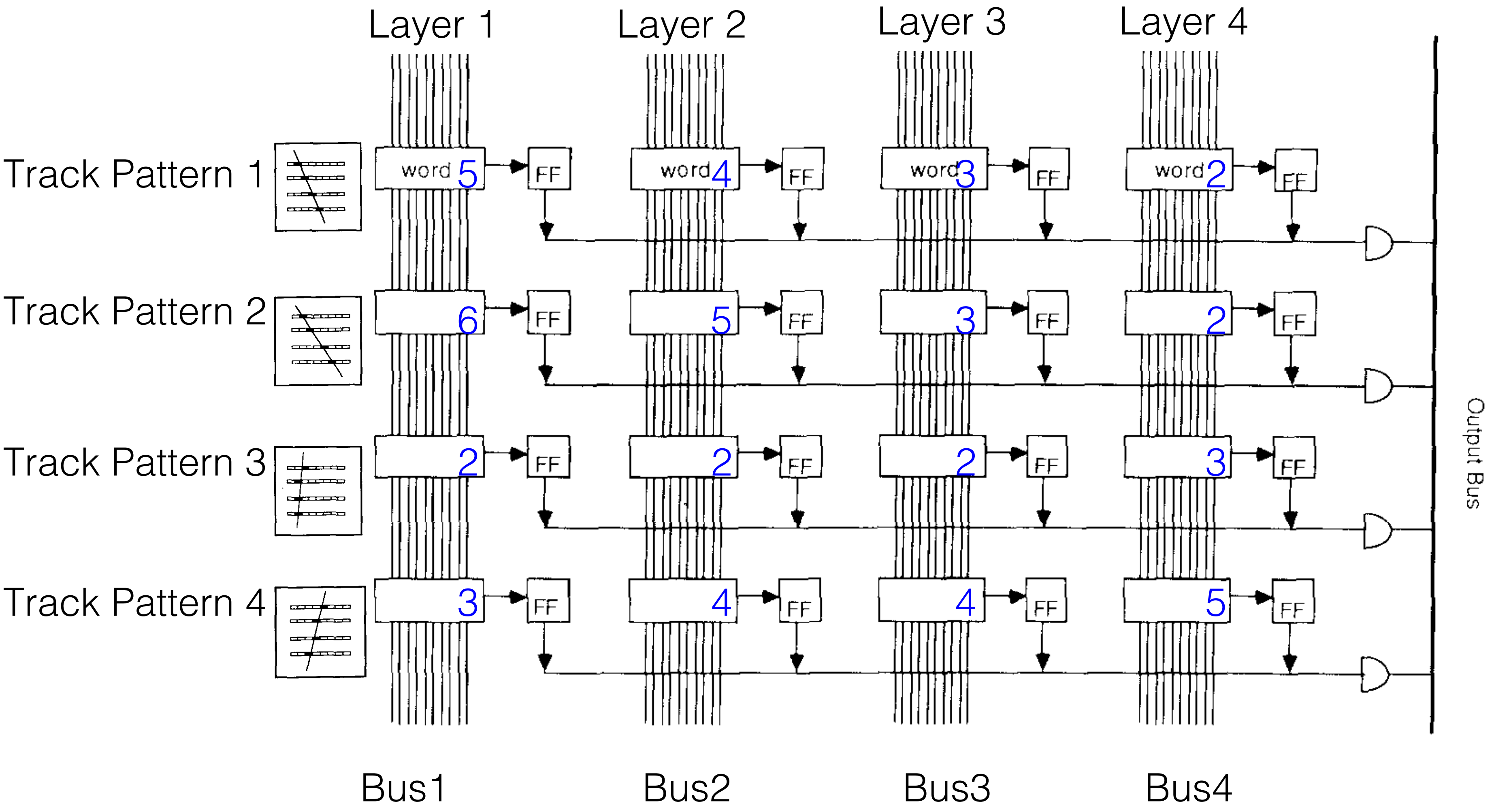
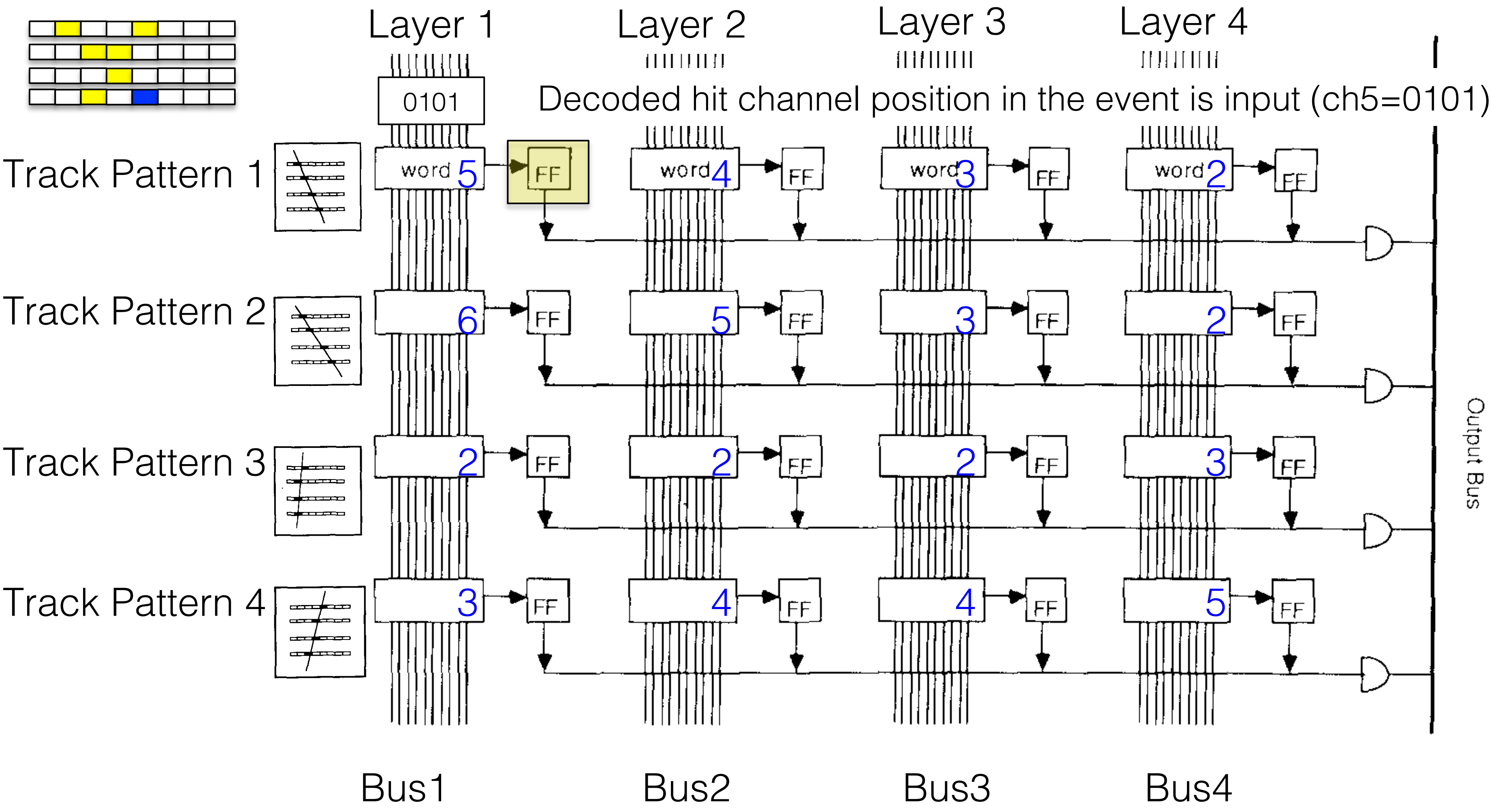


Fig. 3. Associative memory architecture.

Note: demonstration with four layers (in real FTK system, we have 8 layers)

Associative Memory



Layer 1 Layer 2 Layer 3 Layer 4

Decoded hit channel position in the event is input (ch5=0101)

Track Pattern 1

Track Pattern 2

Track Pattern 3

Track Pattern 4

Bus1 Bus2 Bus3 Bus4

Fig. 3. Associative memory architecture.

Associative Memory

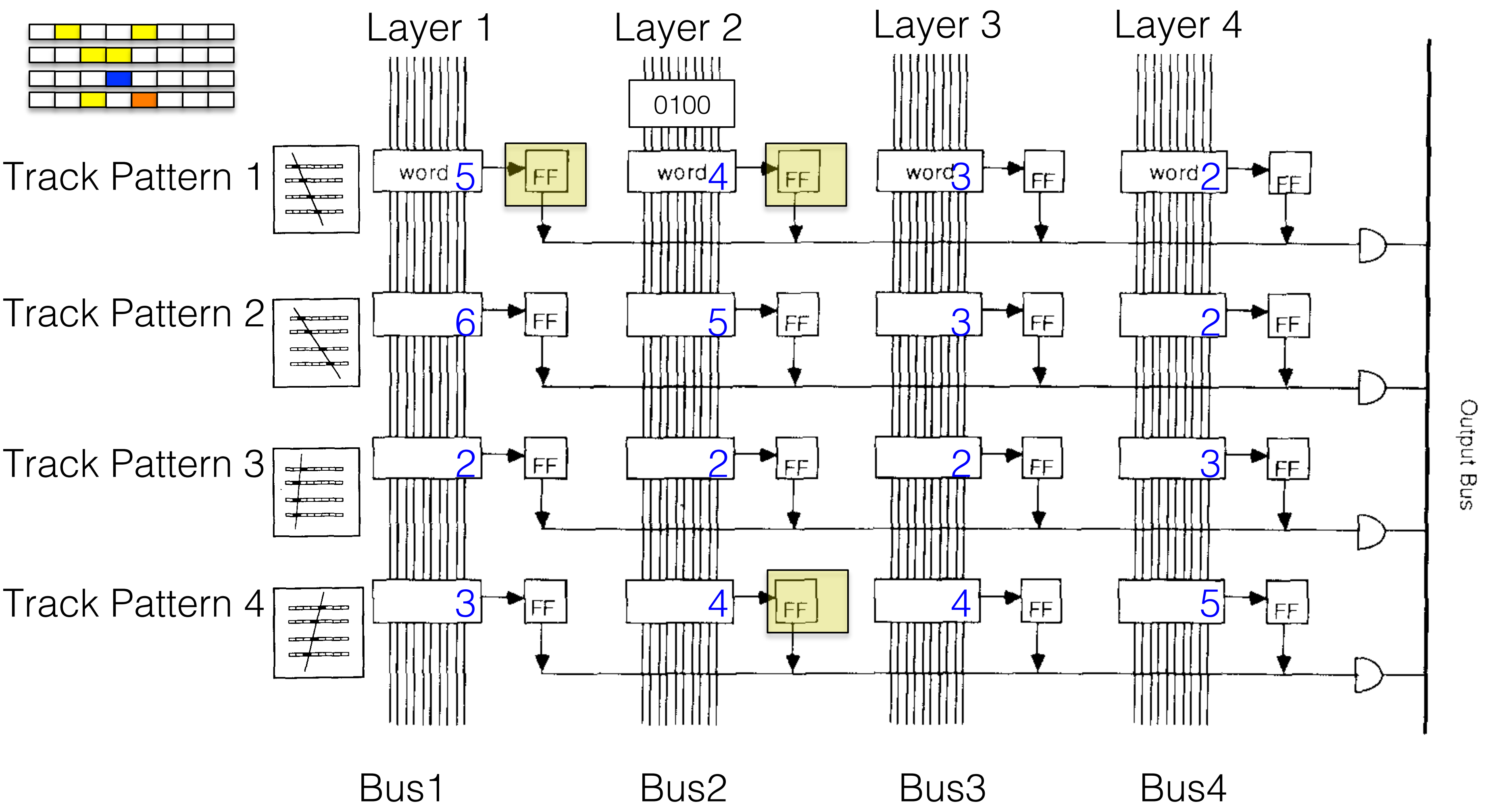


Fig. 3. Associative memory architecture.

Associative Memory

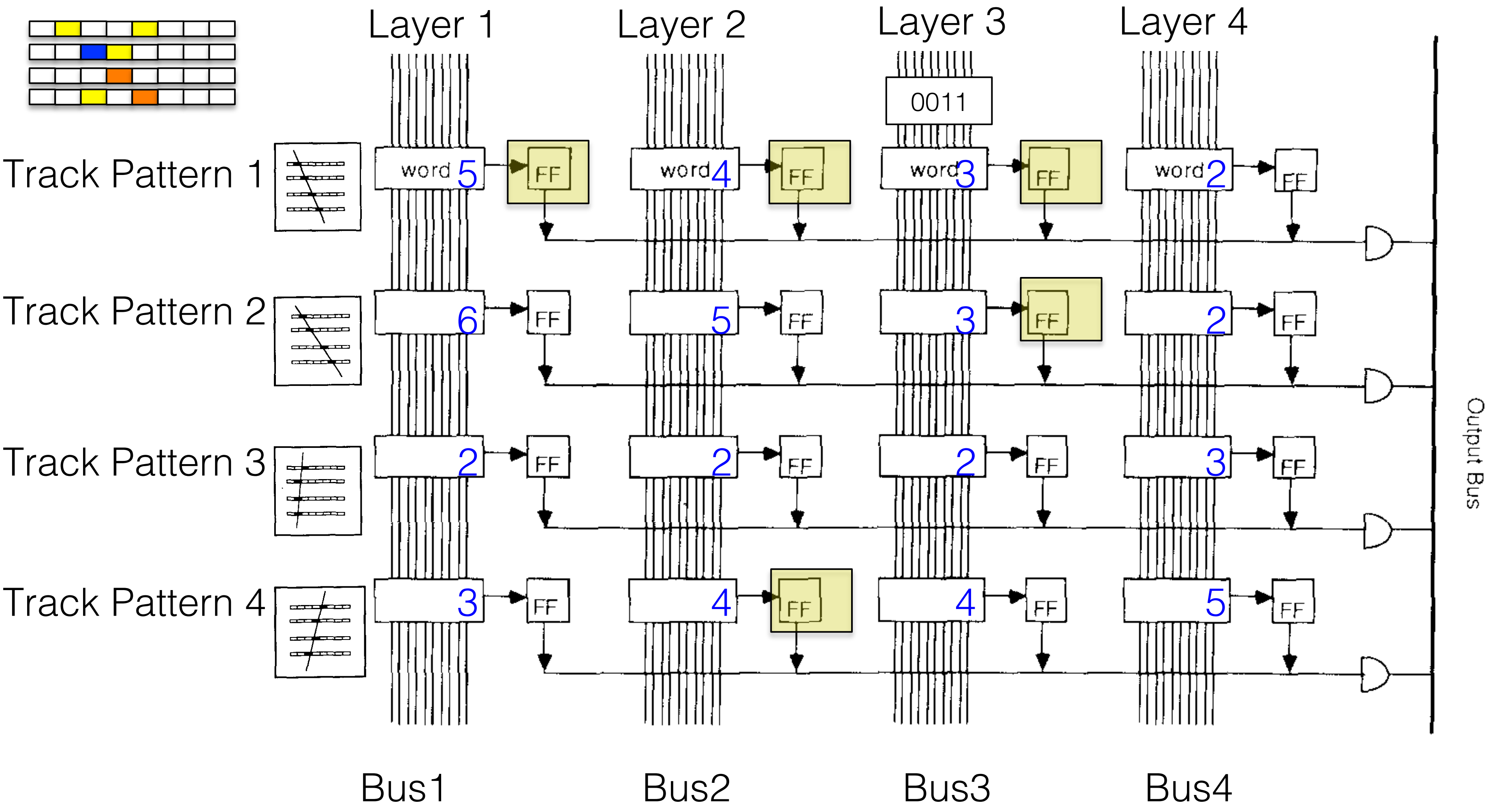


Fig. 3. Associative memory architecture.

Associative Memory

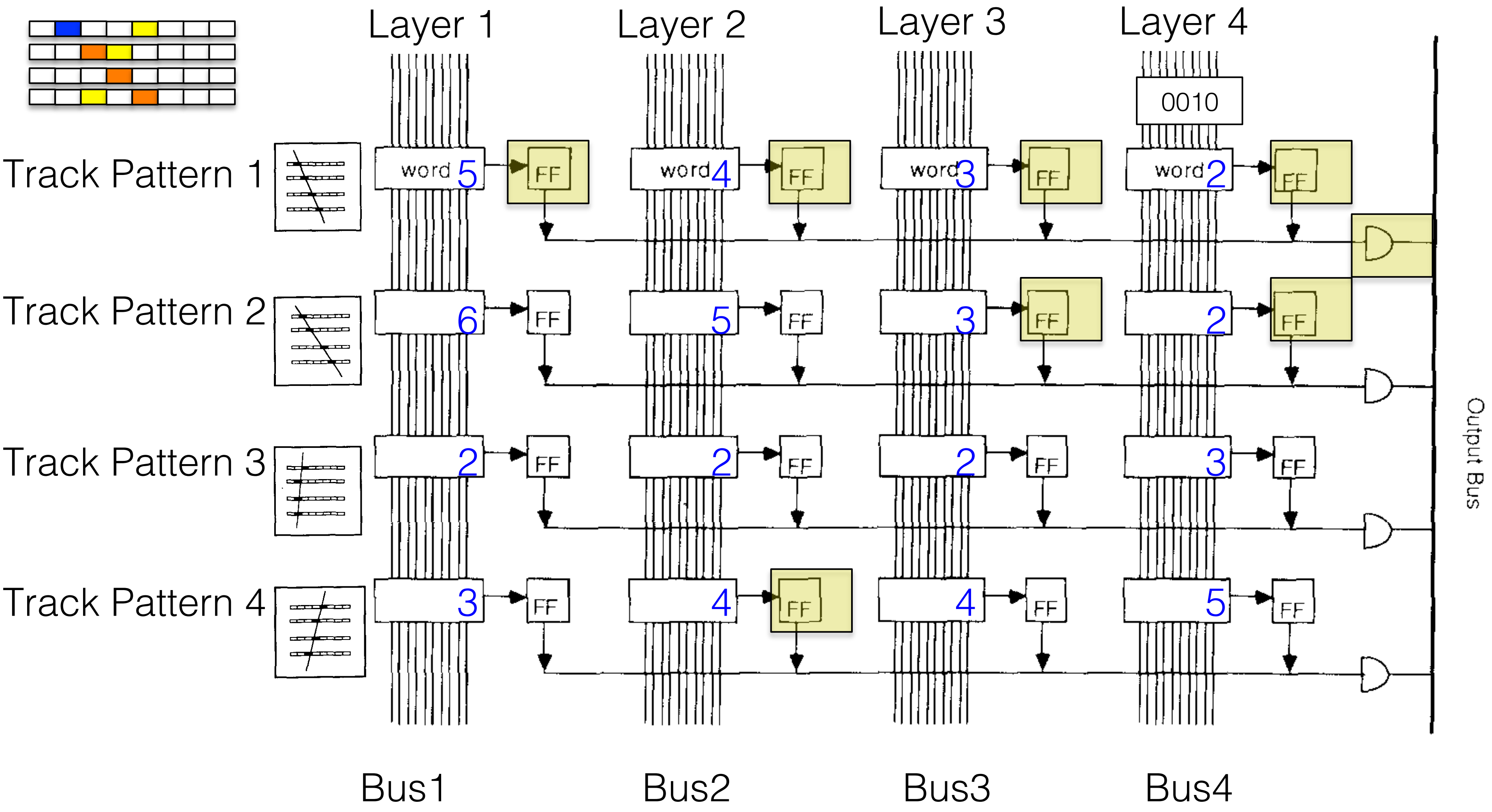


Fig. 3. Associative memory architecture.

Associative Memory

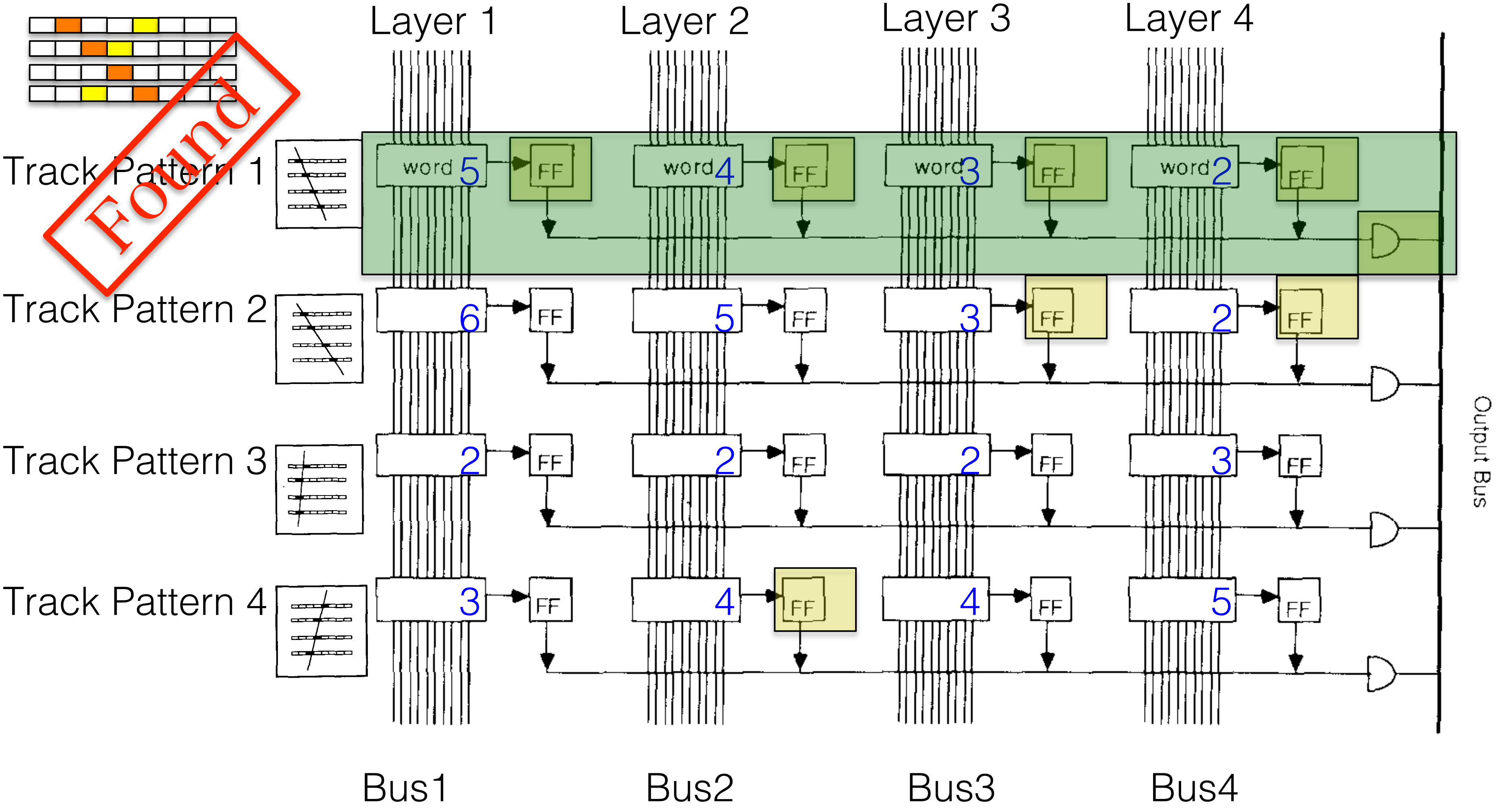


Fig. 3. Associative memory architecture.

Associative Memory

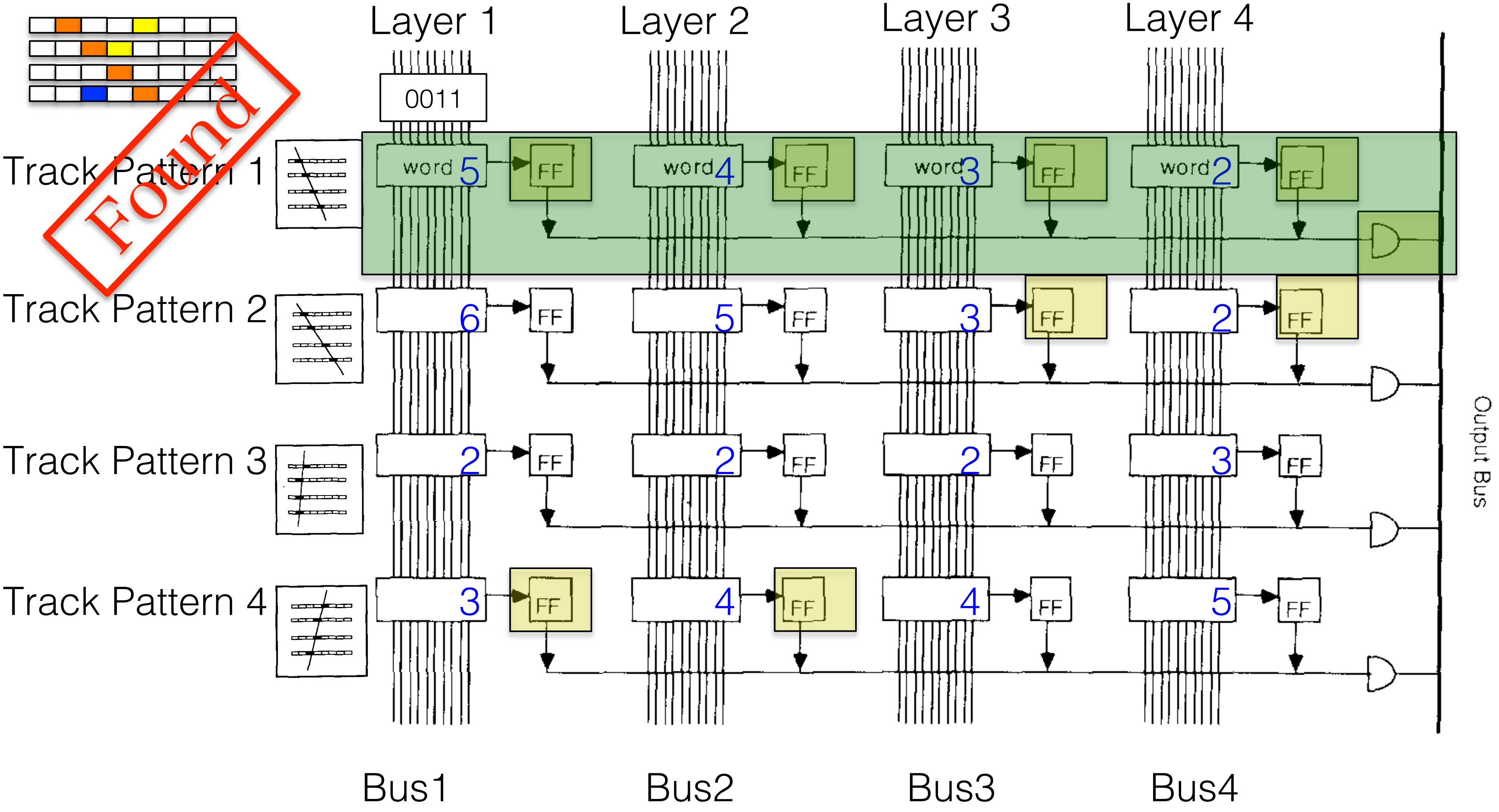


Fig. 3. Associative memory architecture.

Associative Memory

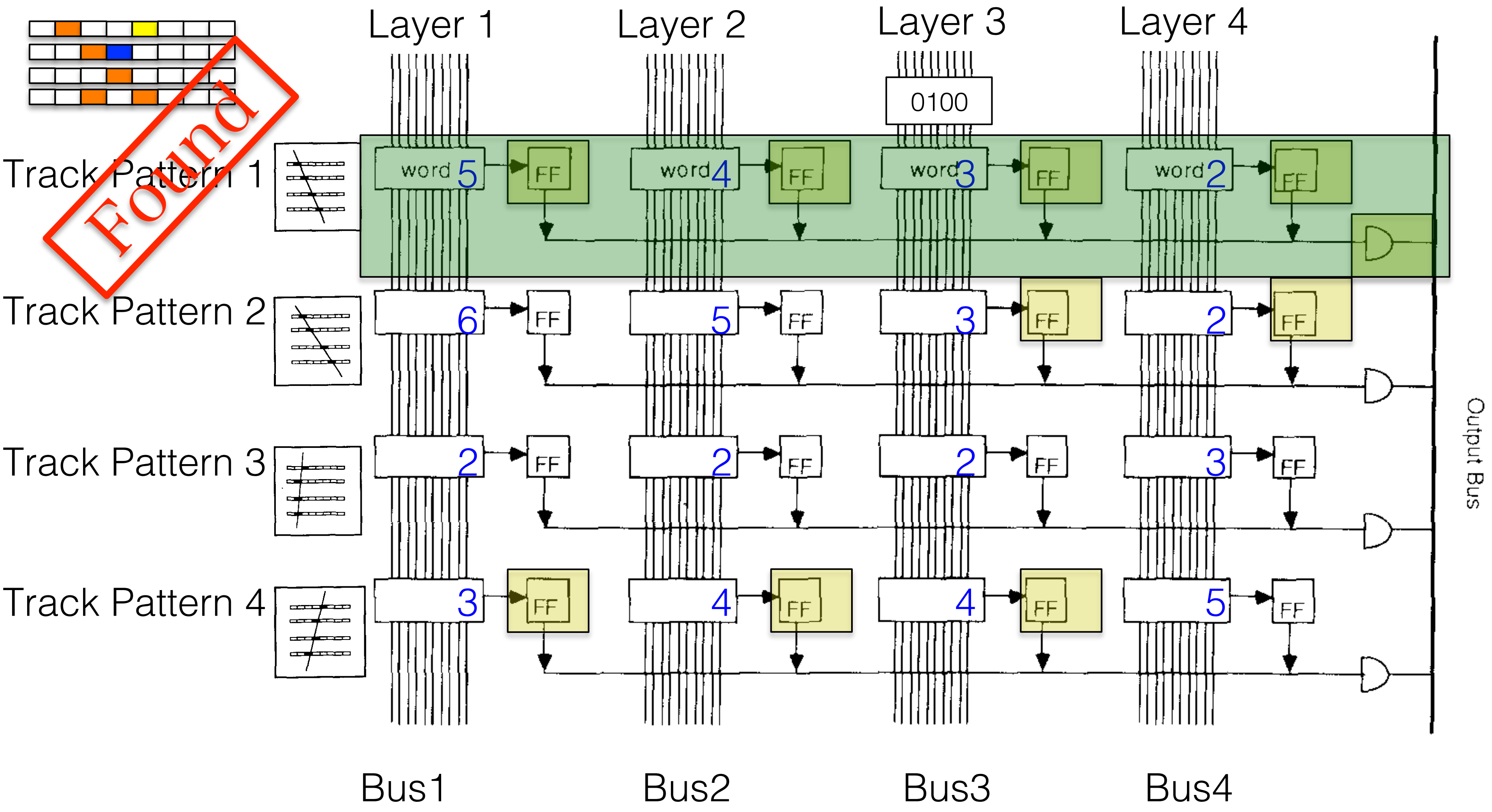


Fig. 3. Associative memory architecture.

Associative Memory

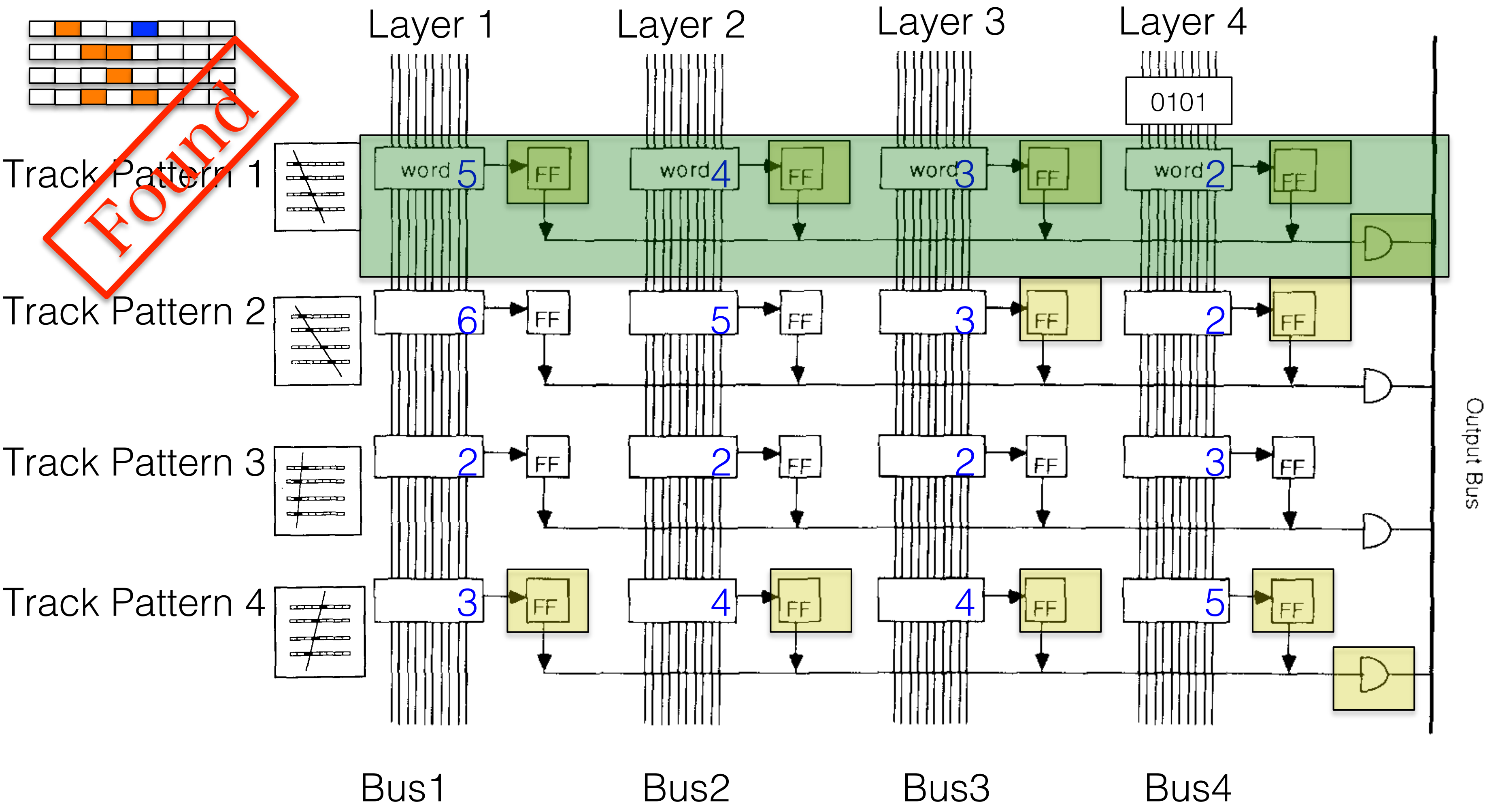
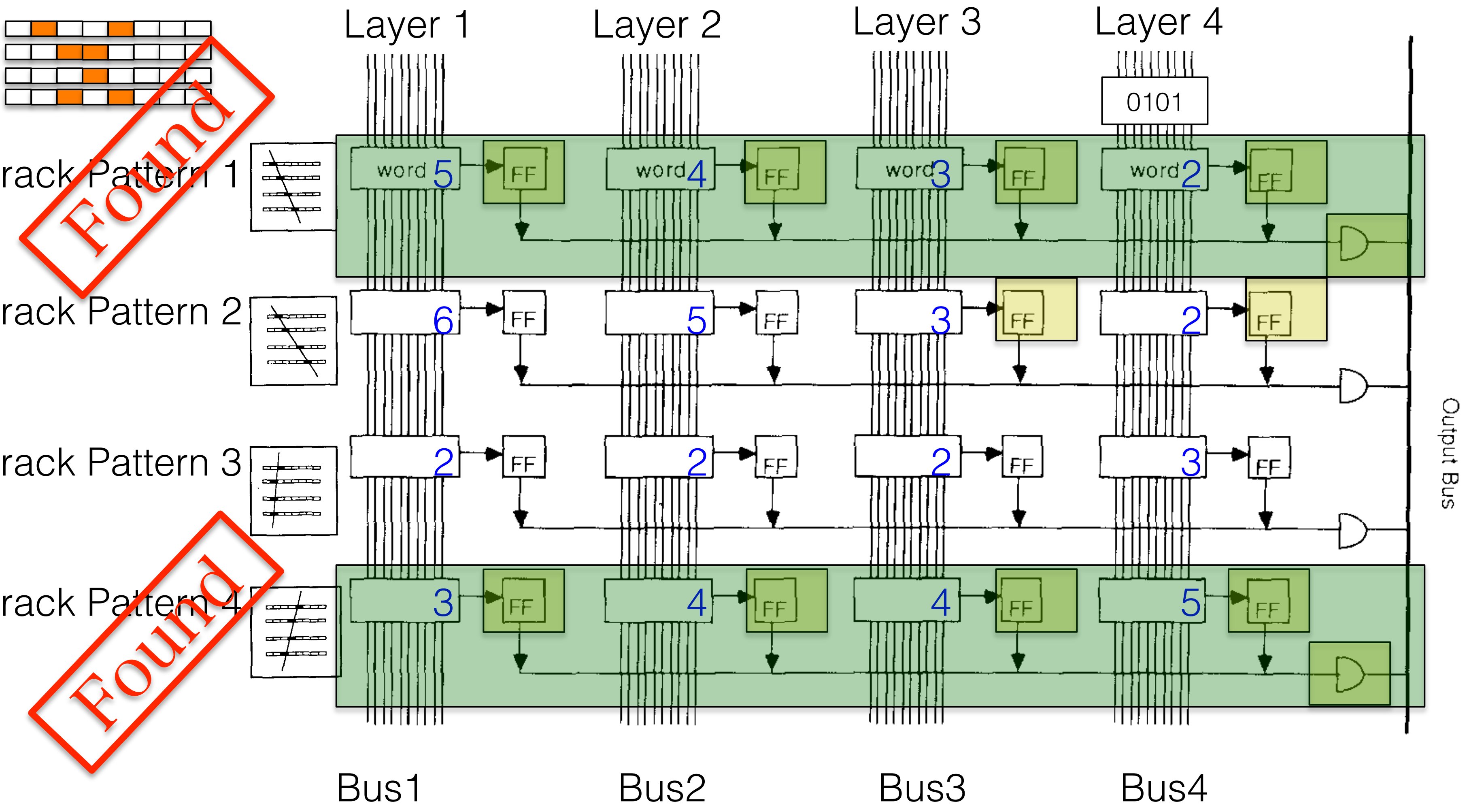


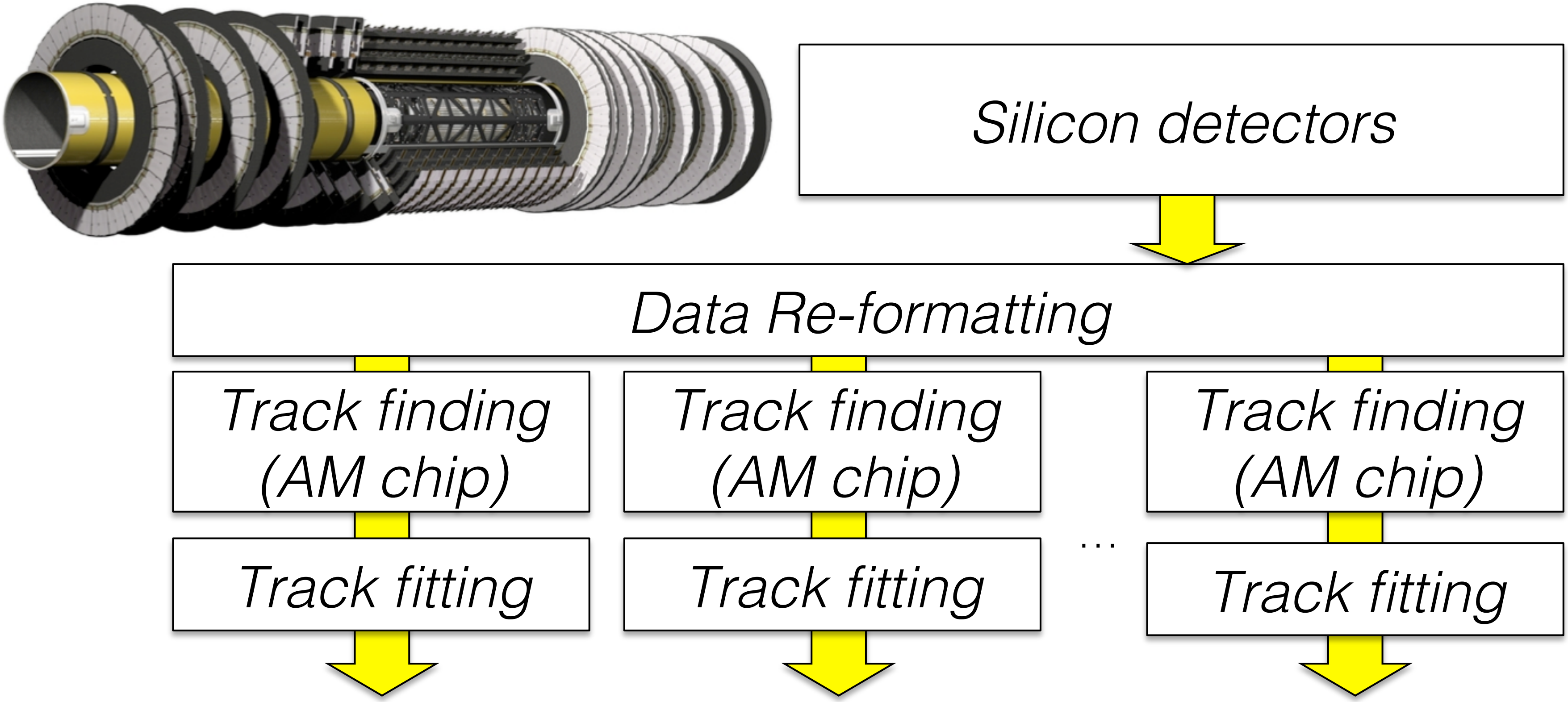
Fig. 3. Associative memory architecture.

Associative Memory



As soon as all the detected hits are loaded, the pattern recognition will be completed (i.e. processing time is linear in the number of hits)

segmentation for parallel processing (η-ϕ tower)



Silicon detectors

Data Re-formatting

Track finding (AM chip)

Track fitting

Track finding (AM chip)

Track fitting

…

Track finding (AM chip)

Track fitting

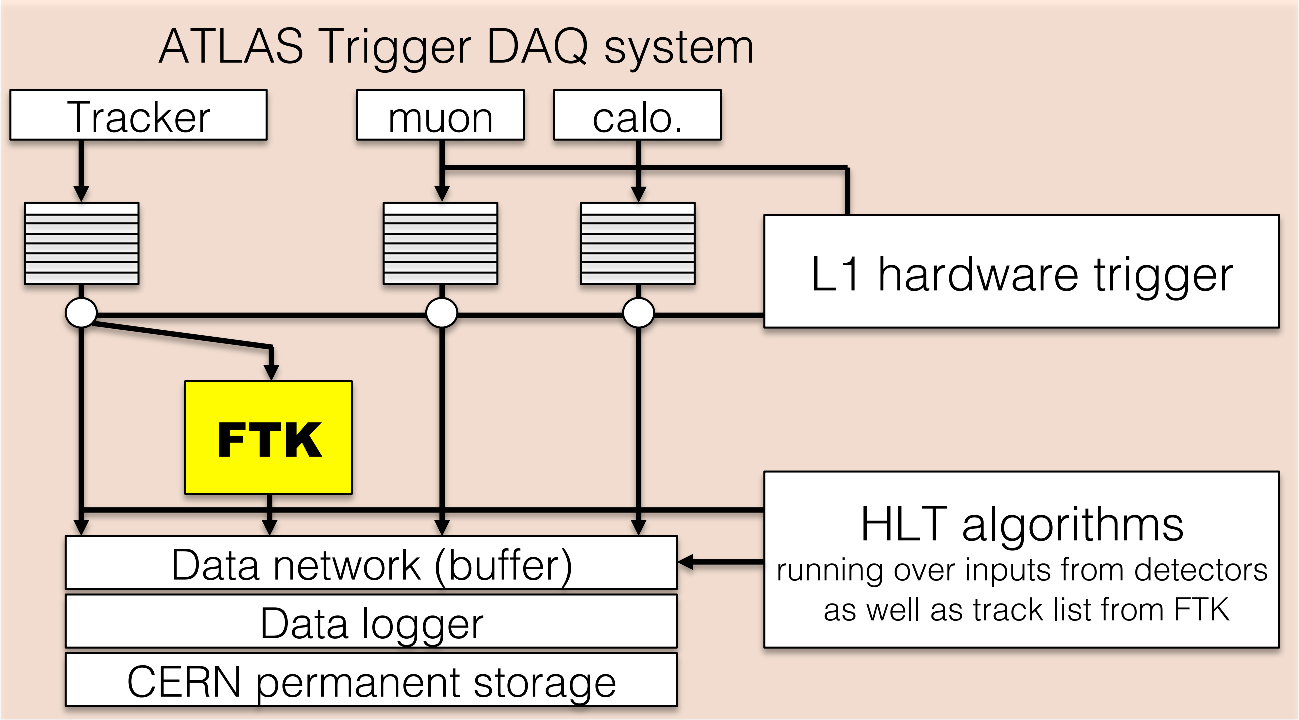
- To reduce number of hits, roads, tracks per board
- To reduce number of needed patterns per board

System overview of FTK

ATLAS FTK

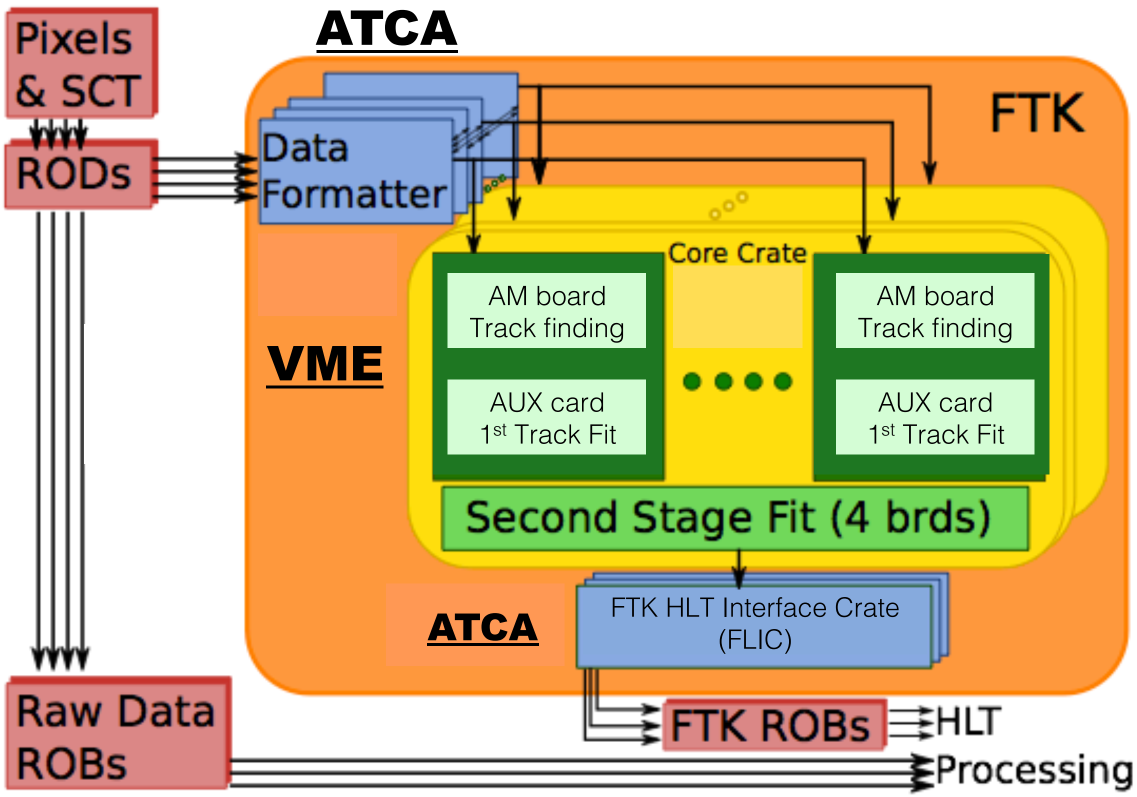
The first associative memory based hardware tracking system at LHC

- Provide full track list for HLT software trigger algorithms O(100us)
- All silicon pixel and micro-strip sensors involved
  - 100 million channels, 12 logical layers (4 Pixel + 4 x 2 Strip layers)
- Track finding and fitting will be done in 64 η-ϕ trigger towers



Fast TracKer system diagram

Copy hit data to FTK
(Dual S-link optical link interface)



Receive data and prepare for
For FTK Tracking in 4 ATCA

Parallel processing tracking Engines (64 trigger towers) in 8 VME crates

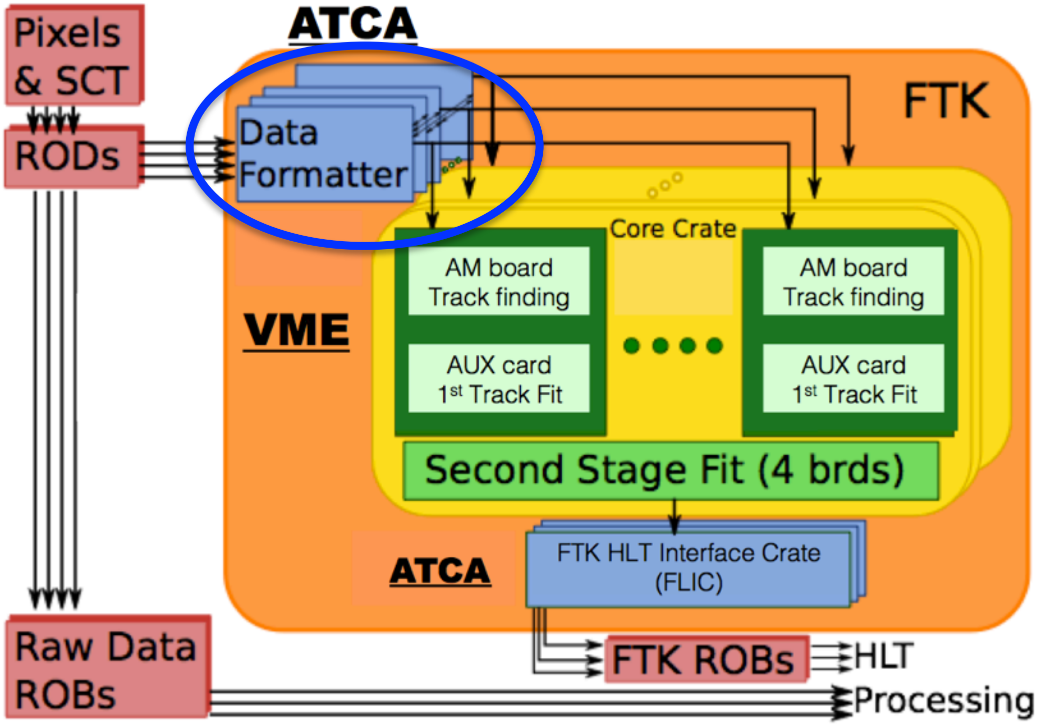
- Track finding with AM
- Track fitting

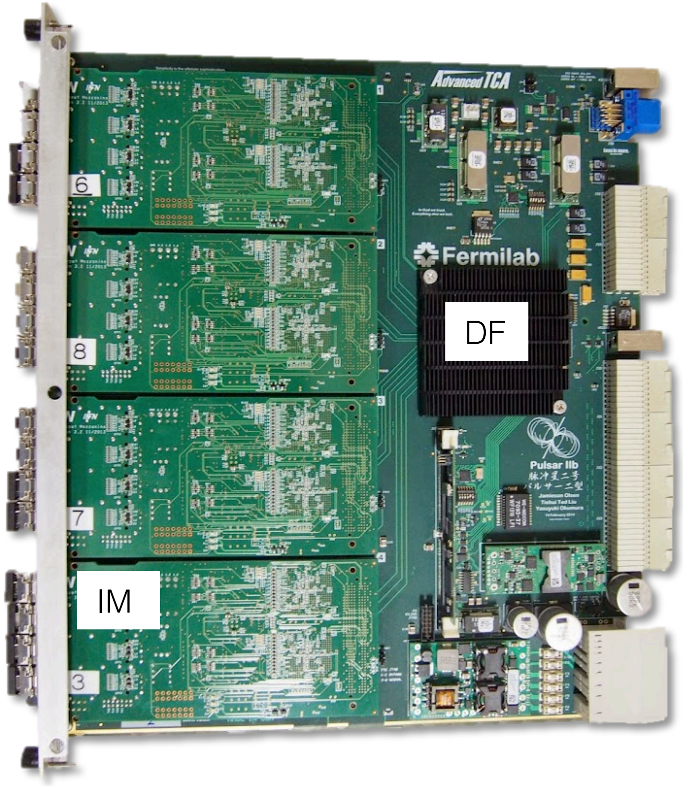
Send tracking results to HLT

Clustering + Data Formatting

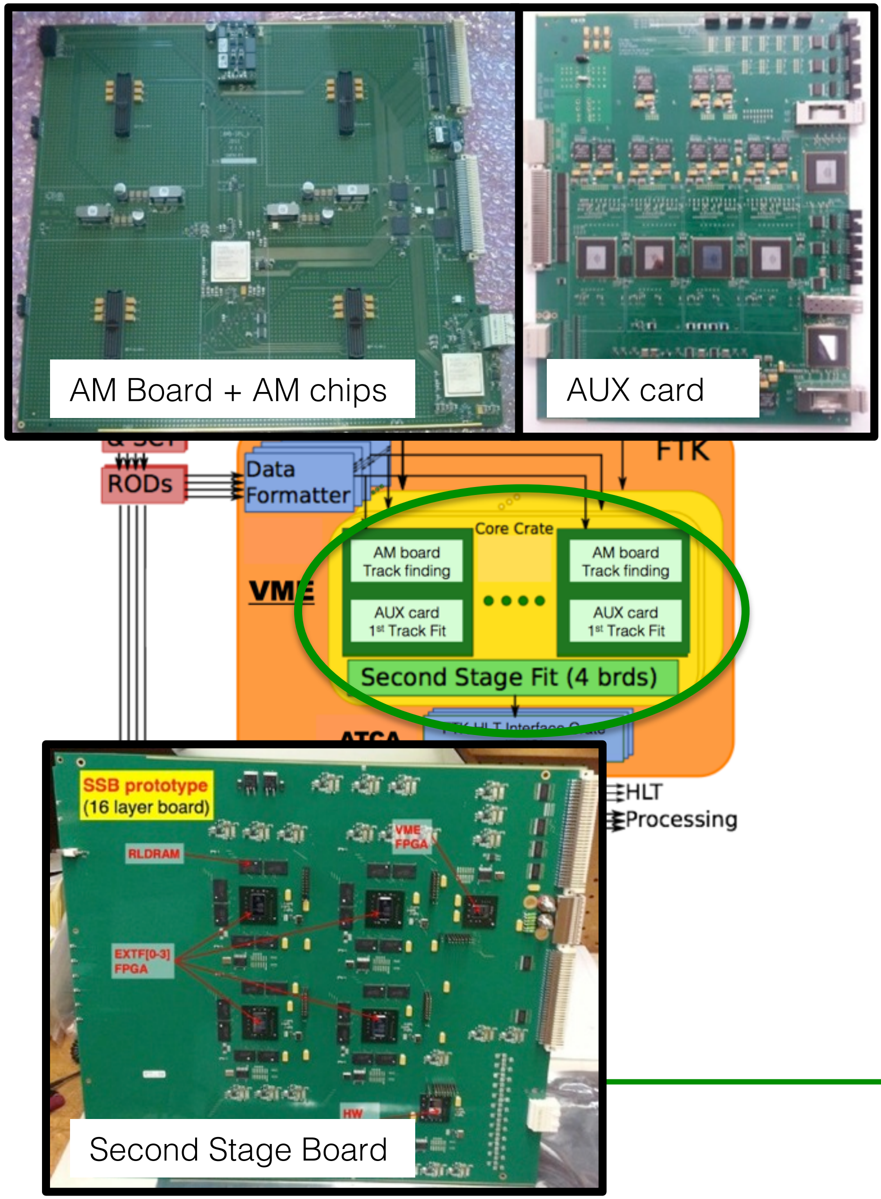
Format input data for all 64 trigger towers

1. Input clustering mezzanine (IM)
2. Data Formatter (DF)





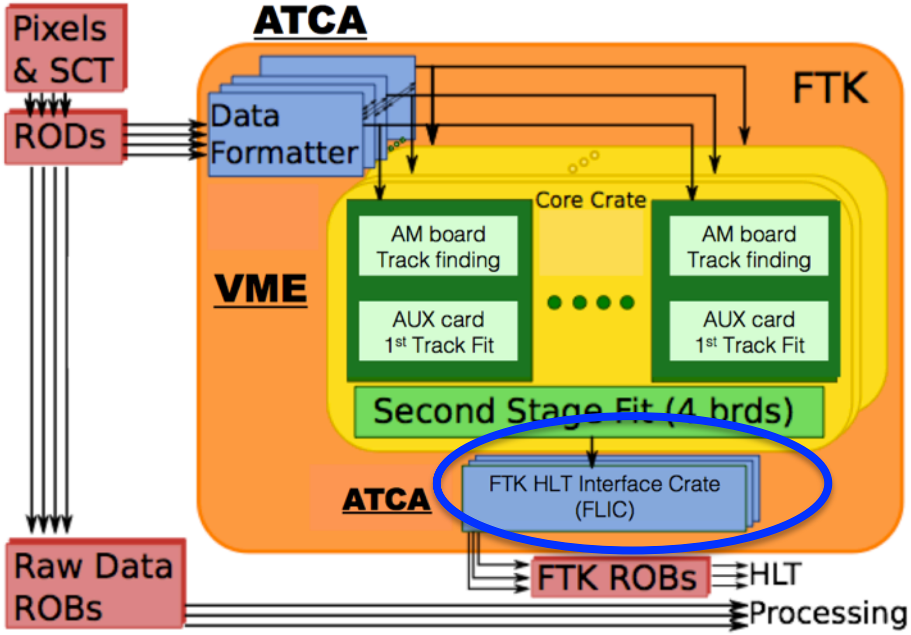
Track finding + Track fitting

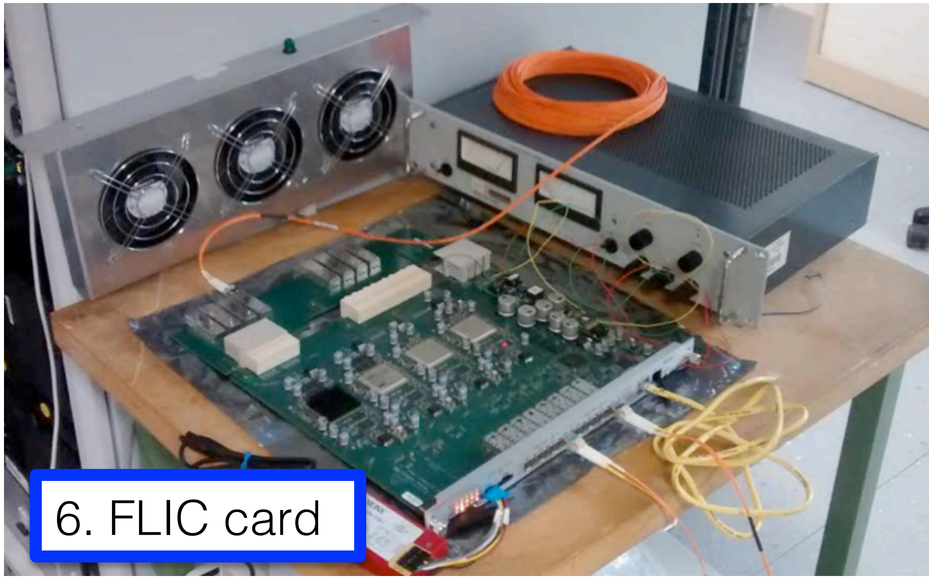


- Track finding in AM
- Fit in two stages
  - Optimal solution for FTK
  - 1st stage uses 8 layers:
    - 3 pixel+5 strip layers
  - 2nd stage uses entire 12 layers:
    - 4 pixel+8 strip layers
- Linear approximation
  - $p_i = \sum_j C_{ij} \cdot x_j + q_i$
    - p_i : ith parameter of interest (track parameters, fit goodness)
    - x_j : hit position in jth layer
    - C_{ij} qi : Pre-stored constants
  - 1fit / ns / FPGA (in 1st stage)
  - Used in track extrapolation in second stage fit

Interface to HLT

- Collect found tracks
- 2 boards cover all 64 eta – phi towers
- Global functions may be implemented
  - E.g. count tracks above threshold

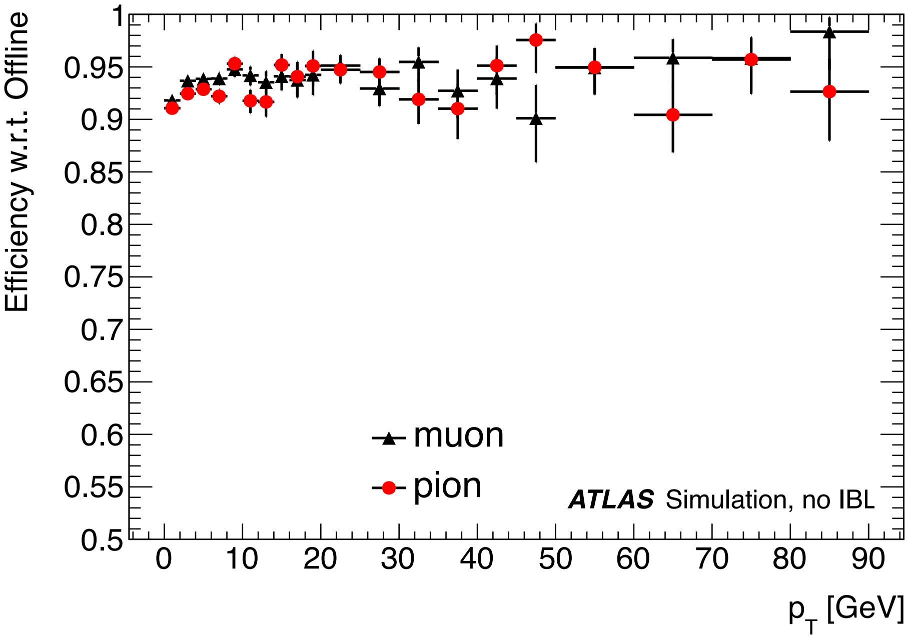



6. FLIC card

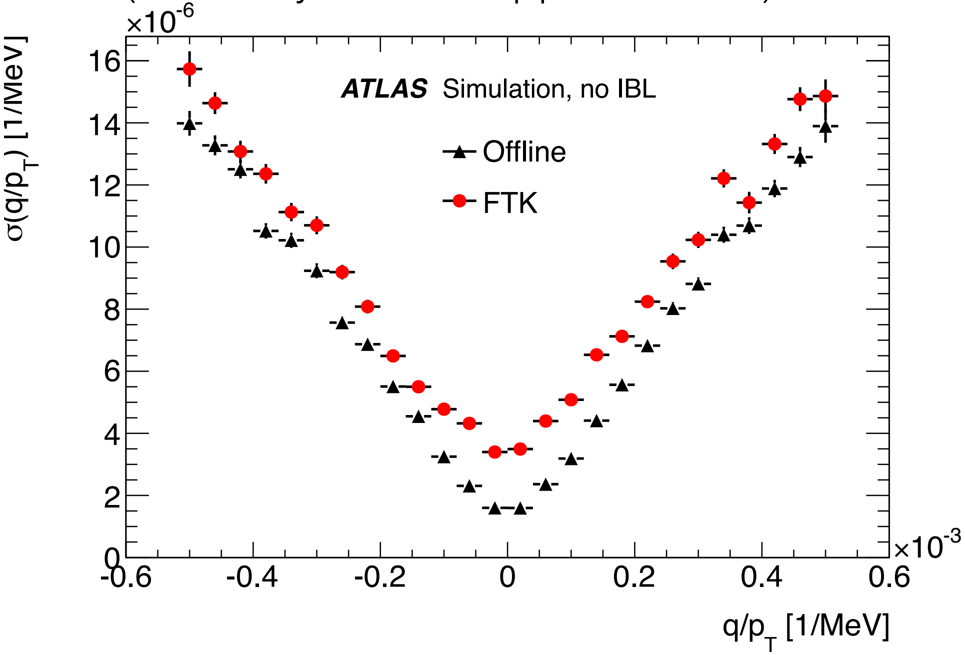
FTK performance

Provide tracks with “quasi-offline quality” in O(100u)

FTK efficiency wrt offline ~ 93%
(driven by the number of AM patterns)



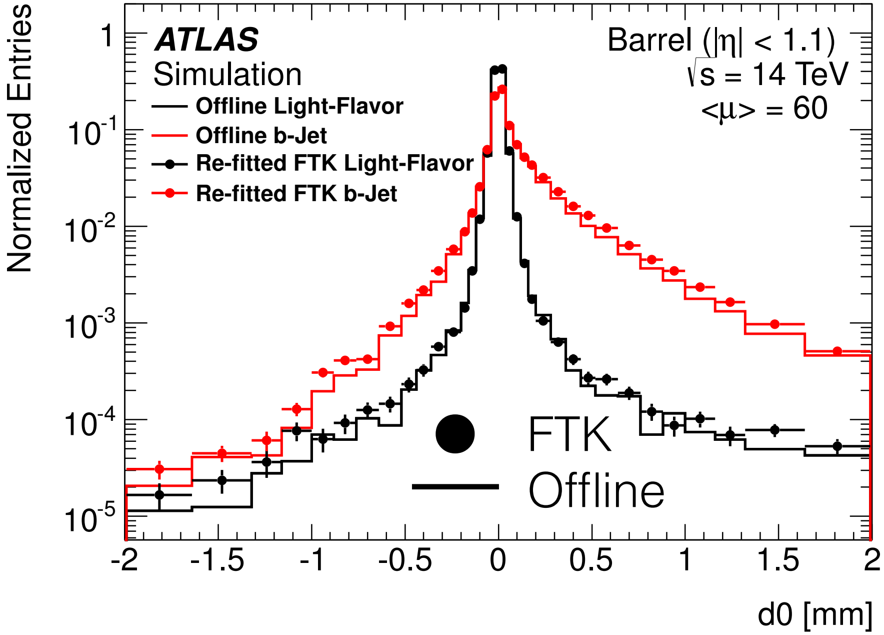
Curvature parameter
(driven by linear fit approximation)

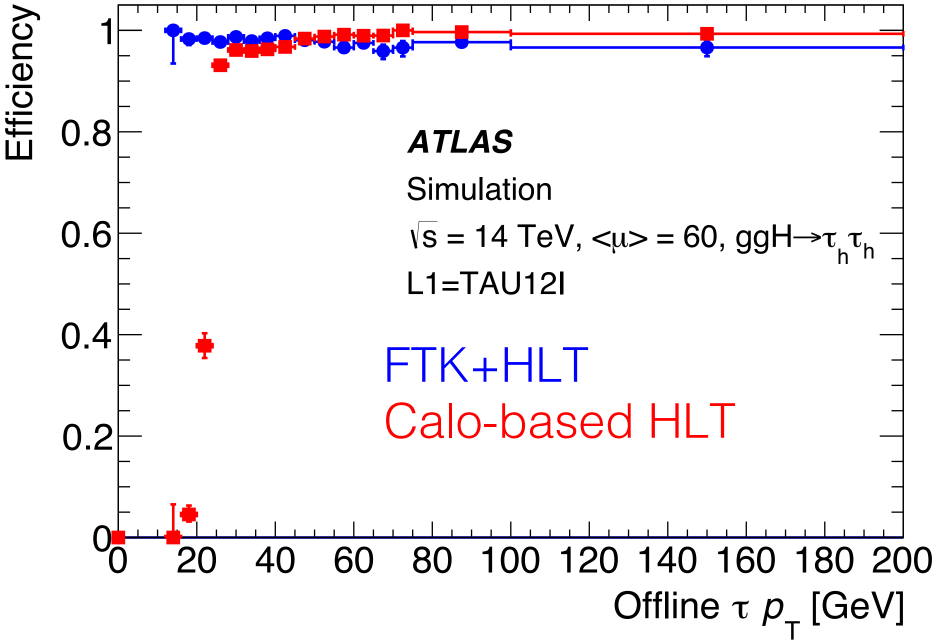


Note : Also there is room to improve the efficiency and resolution with optimization

Example of applications to HLT

- Typical application – b / tau identification
  - FTK Performance for displacement of tracks in b-jets is similar to offline
  - Efficiency will be recovered by early access to tracking information given by FTK



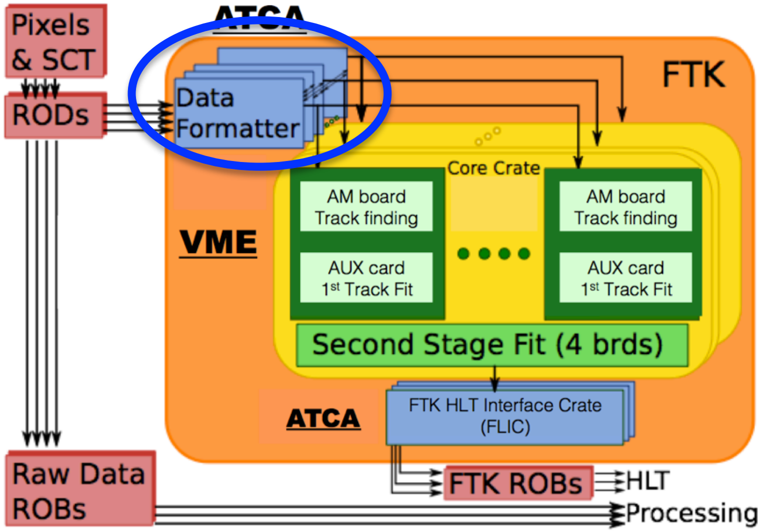


https://cds.cern.ch/record/1667495/
https://twiki.cern.ch/twiki/bin/view/AtlasPublic/FTKPublicResults

Details of technical challenges

- ATCA-based data formatting & clustering
- Associative Memory implementation in FTK

ATCA-based Data Formatting Clustering system

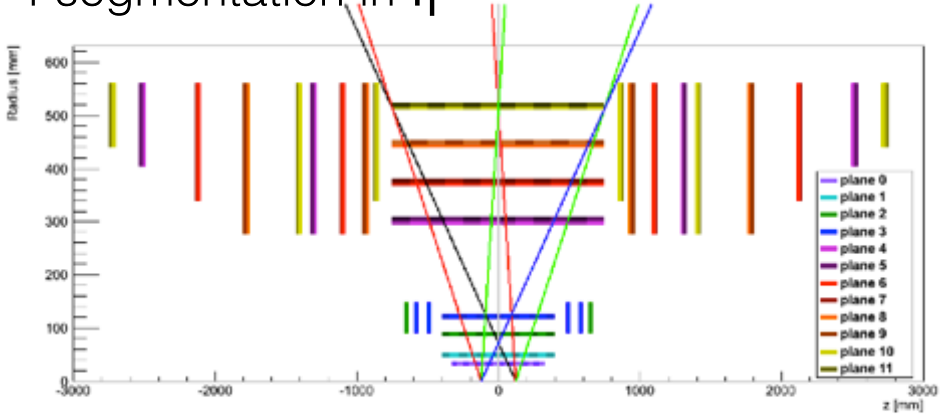


“ATCA-based ATLAS FTK input interface system” http://arxiv.org/abs/1411.0661

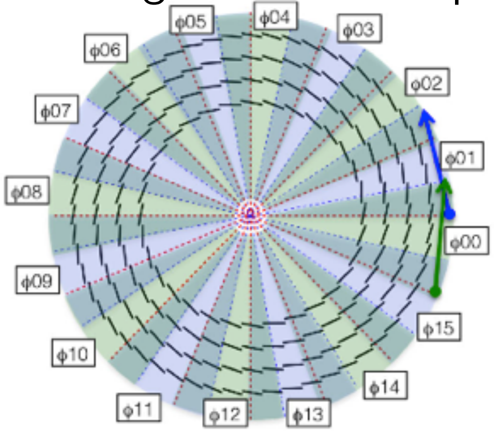
Needs for clustering and data formatting

- FTK is Implemented as parallel processing units
  - 64 eta-phi overlapping trigger towers
    - Finite size of beam luminous region / curvature of charged track
- Data volume is high from entire pixel and micro-strip sensors
  - More than 100k hits / event at highest target luminosity, running at L1 rate (100k)

4 segmentation in η



16 segmentation in ϕ

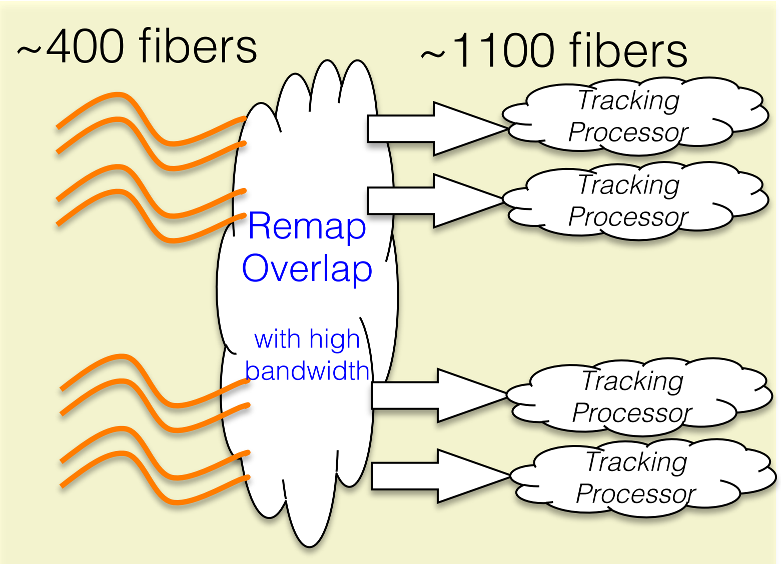


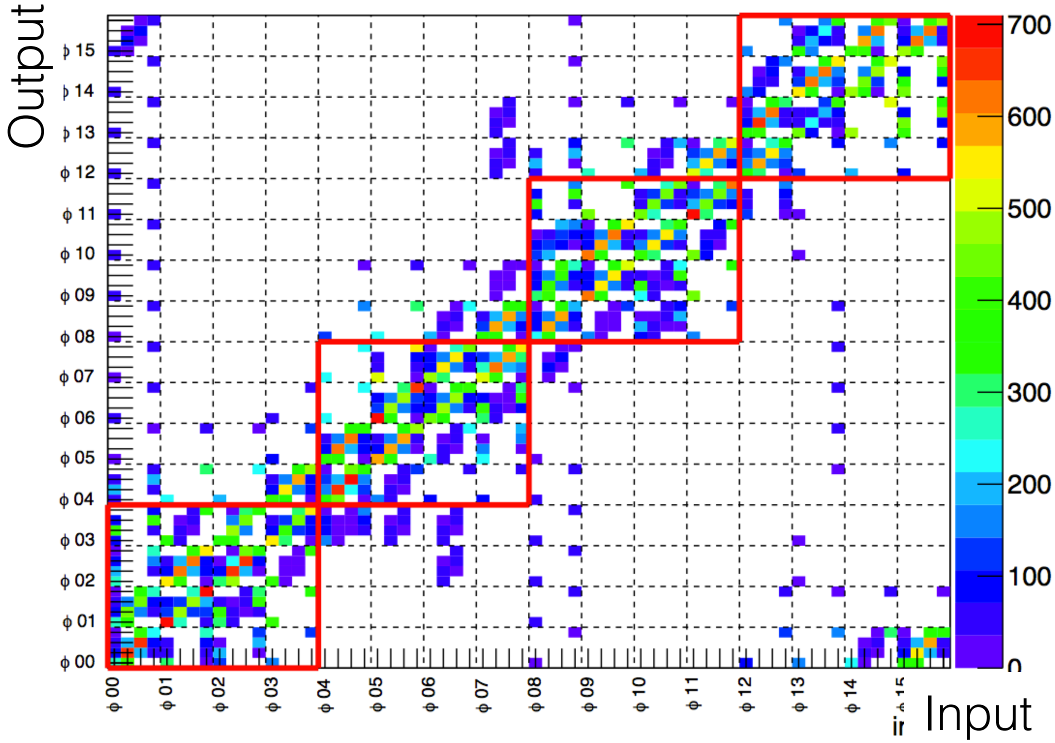
- FTK system requires an input interface system to manage :
  - "**Clustering**" to reduce data volume at the beginning, minimizing loss of efficiency and improving resolution
  - "**Data Formatting**" to distribute clusters to appropriate FTK η-ϕ towers

Data Formatting requirement drives system design concept

Requirement of Data Formatting prior to tracking

- Remapping input data into 64 FTK tower structure
    - ~400 fiber input from ROD~1,100 output fibers to downstream tracking boards
- Overlapping at boundary region of towers
- Handle large data volume

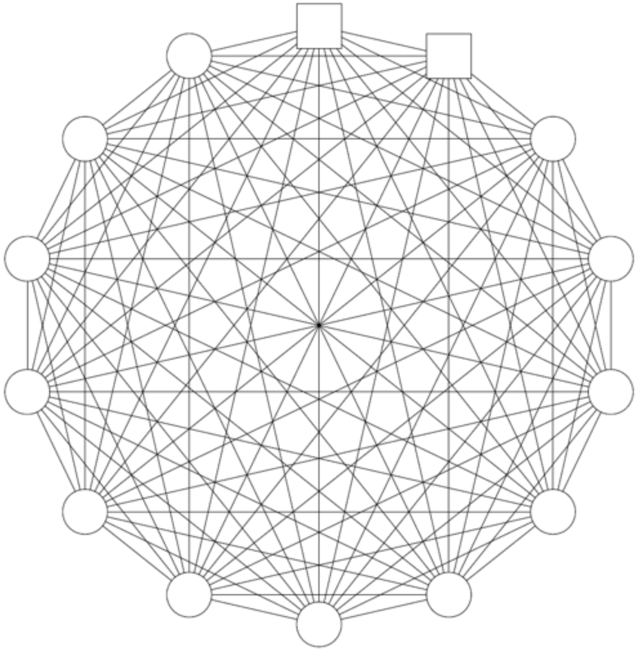




<u>Four-shelf system</u> with <u>flexible connectivity in each crate</u> and <u>inter-crate links</u>

Full mesh ATCA

Advanced Telecommunication Computing Architecture (AdvancedTCA / ATCA) backplane supports full mesh network topology with high speed

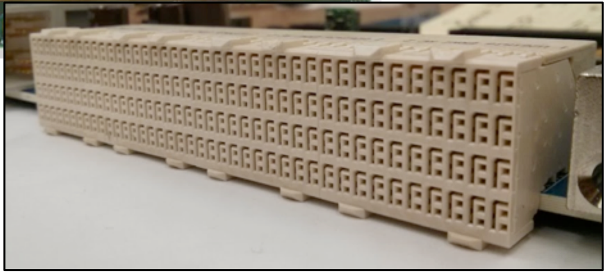


Network topology among the 14 boards in ATCA backplane

Up to 40 Gb/s point-to-point serial links

Zone2 Connector

for full mesh



System level design

- Four shelf-system with internal connections
  - Full-mesh communication with use of backplane
  - Inter-crate connections with optic fibers
- Eight boards per shelf, and 32 boards in total

From ROD

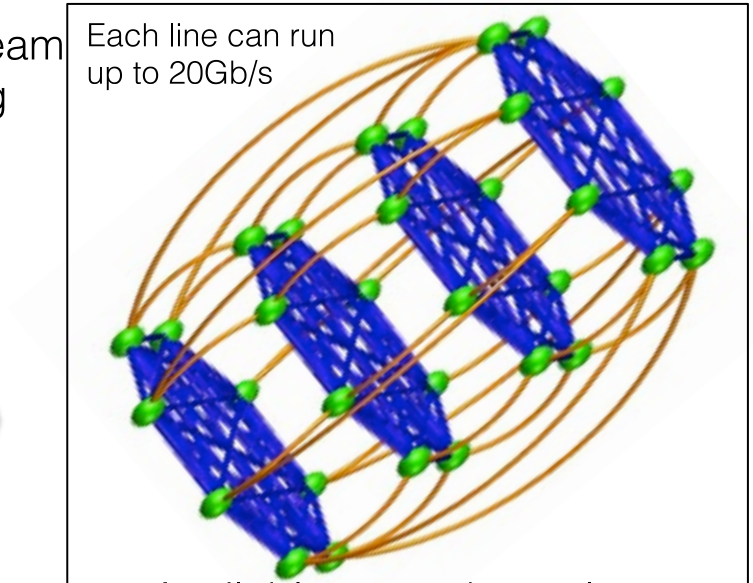
input fibers

ROD data (raw hits)

To Downstream FTK tracking boards

output fibers

FTK eta-phi Tower data (clusters)



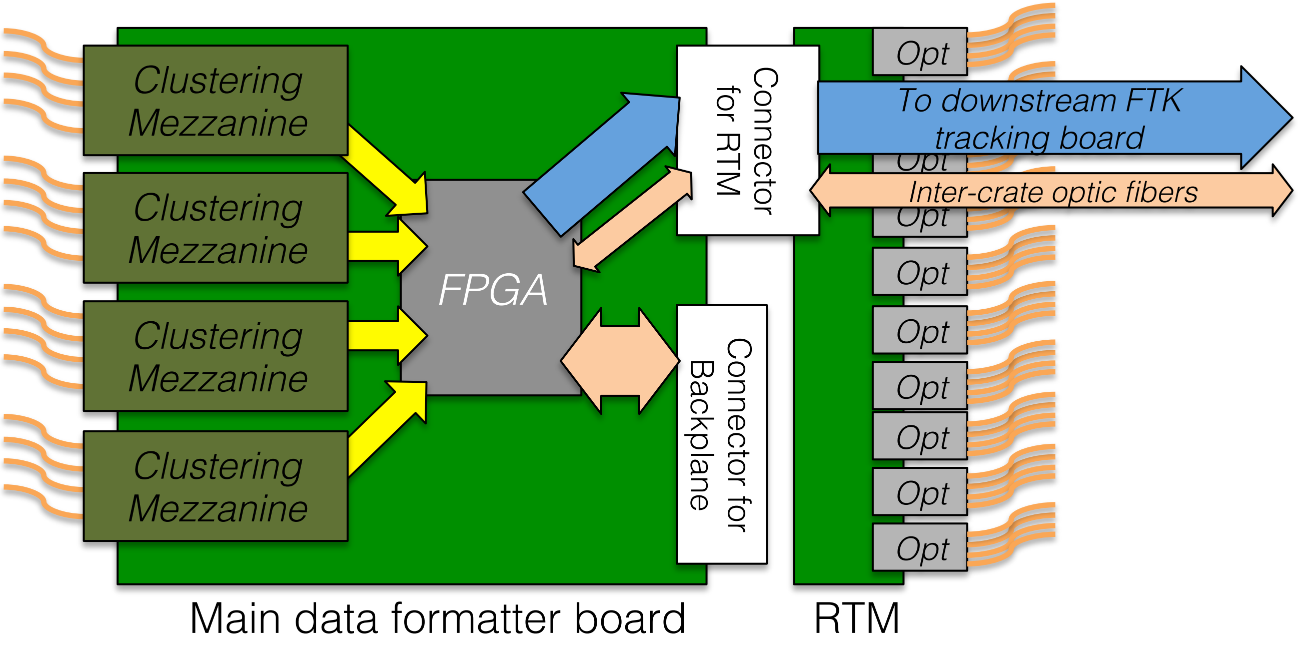
Each line can run up to 20Gb/s

Available network topology

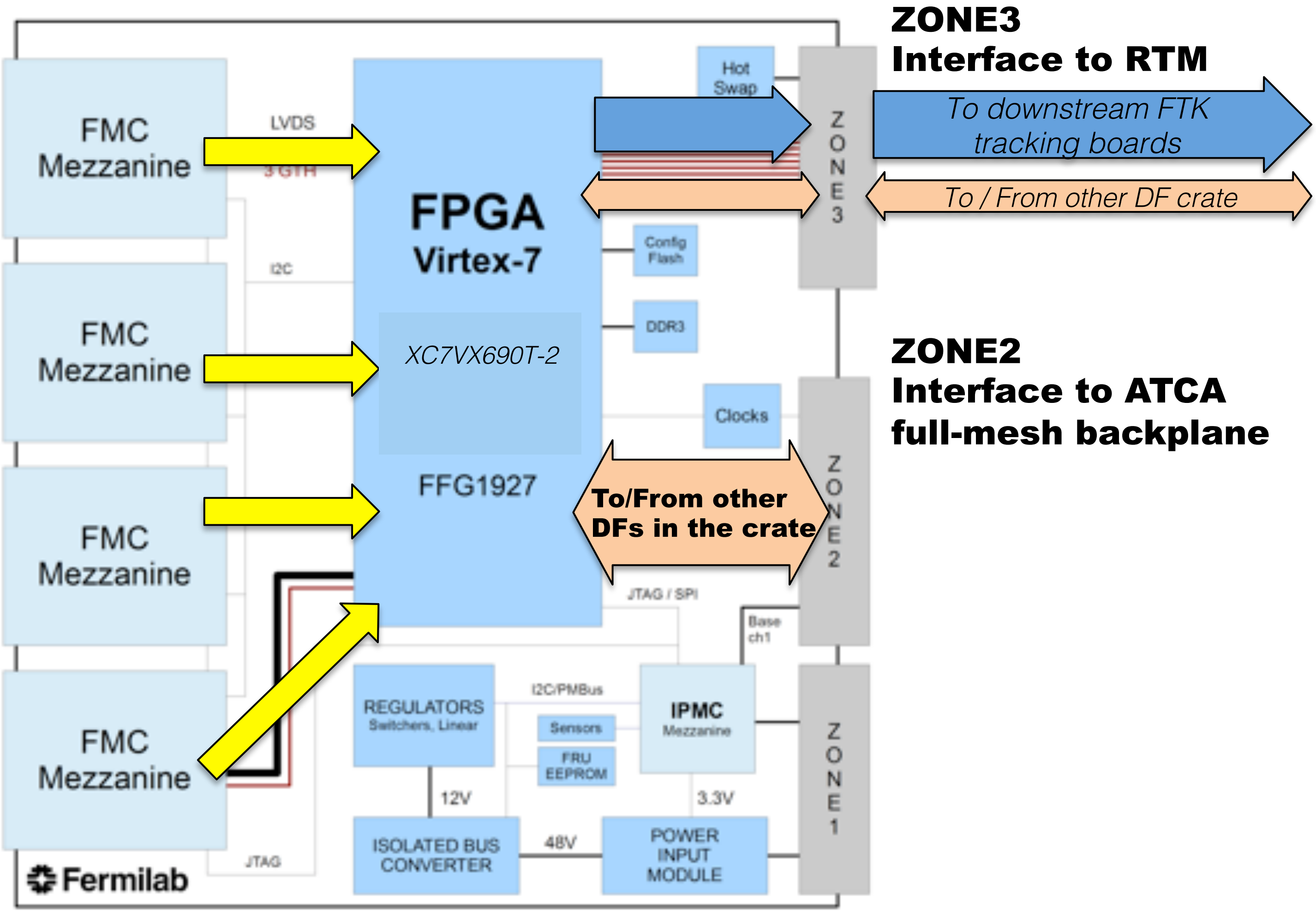
Green balls stand for 32 boards

Board level design concept

- One board consists of three main components:
    1. Input clustering mezzanine cards (FTK_IM)
    2. Main data formatter board (Main board)
    3. Rear Transition Module (RTM) for optical fiber I/Os



Design around Virtex7 FPGA



ZONE3
Interface to RTM

To downstream FTK tracking boards

To / From other DF crate

ZONE2
Interface to ATCA full-mesh backplane

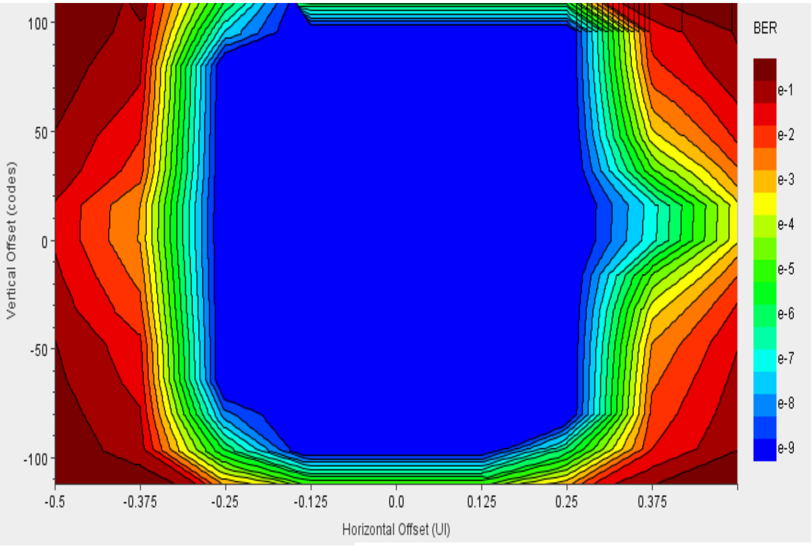
To/From other DFs in the crate

Data Formatter board measured speed performance

- Design compatible with the high speed data traffic
  - Material choice
  - Trace length matching

- Speed performance are verified in prototype board testing :
  - Bit error rate < 5×10^{-16}
  - RX margin analysis
  - With Xilinx Chipscope IBERT

Threshold @ RX



Sampling point @ RX

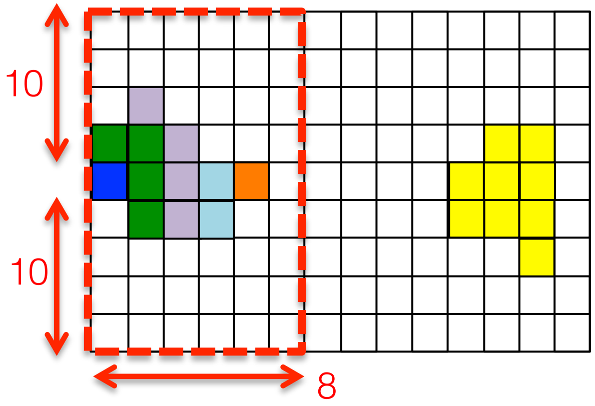
80 GTH at 10 Gb/s available (52 GTH at 6.4Gb/s required by FTK)

- 40 lanes for RTM (38 for FTK)
- 28 lanes for Fabric (14 for FTK)
- 12 lanes for Mezzanines (8 for FTK)

NOTE : Data Formatter board is a general purpose board design and its specifiaction well exceeds that of FTK DF requirement. (another application is for CMS L1 tracking trigger R&D : Pulsar-II project at Fermilab http://www-ppd.fnal.gov/ATCA/)

2D pixel clustering challenge

- Keep manageable processing time even with the highest target luminosity by avoiding too many hit loop
- Map 2D pixel hit information in 2D structure of cells in FPGAs so that hit selection can be done two-dimensionally
  1. Define clustering window (21x8) with respect to the reference hit
     - The first arriving or the leftmost not-clustered hit will be the reference hit
     - Entire pixel modules does not need to mapped (moving window technique)
  2. Load all hits in the window size into logic cells mapped in FPGA
  3. Select all hits neighboring to the reference hit (as seed)
  4. Repeat hit selection w.r.t. selected hits until there is no more neighboring hits
  5. Read selected hits as a cluster
  6. Start to build the next clusters with remaining hits
- Full implementation fits in available resources

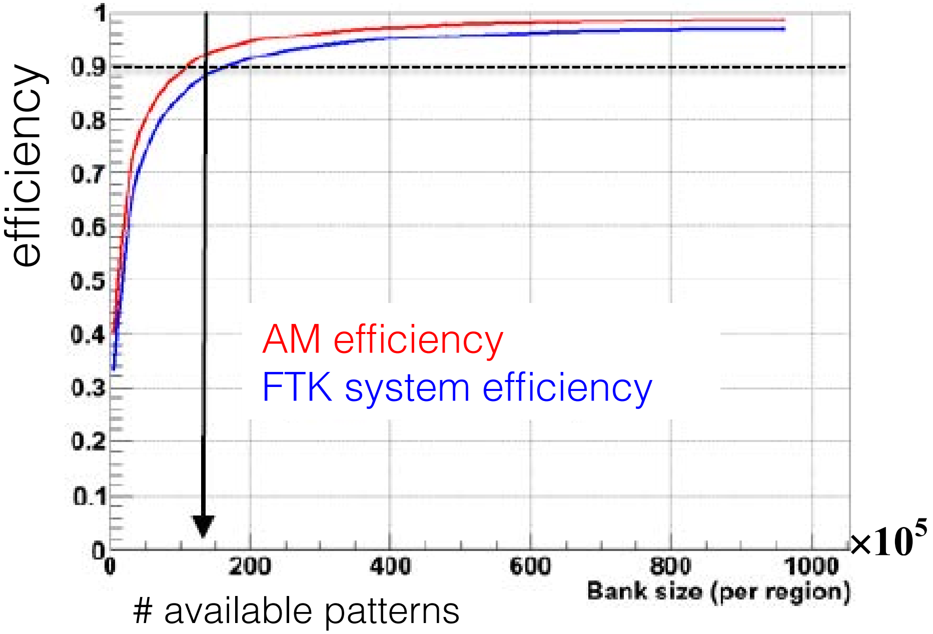


🟦 First hit as seed for the clustering
🟩 Hits selected in the second clock cycle
🟪 Hits selected in the third clock cycle
🟦 Hits selected in the forth clock cycle
🟧 Hits selected in the fifth clock cycle
🟨 Hits belonging to a different cluster

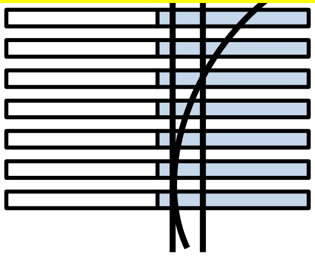
More details in IEEE TNS paper
http://ieeexplore.ieee.org/xpl/articleDetails.jsp?tp=&arnumber=6949160

AM implementation in FTK

Reminder : Challenges in AM approach at LHC

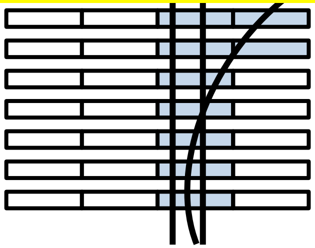


Low resolution in AM track finding



Smaller bank size
Lower S/N

High resolution in AM track finding

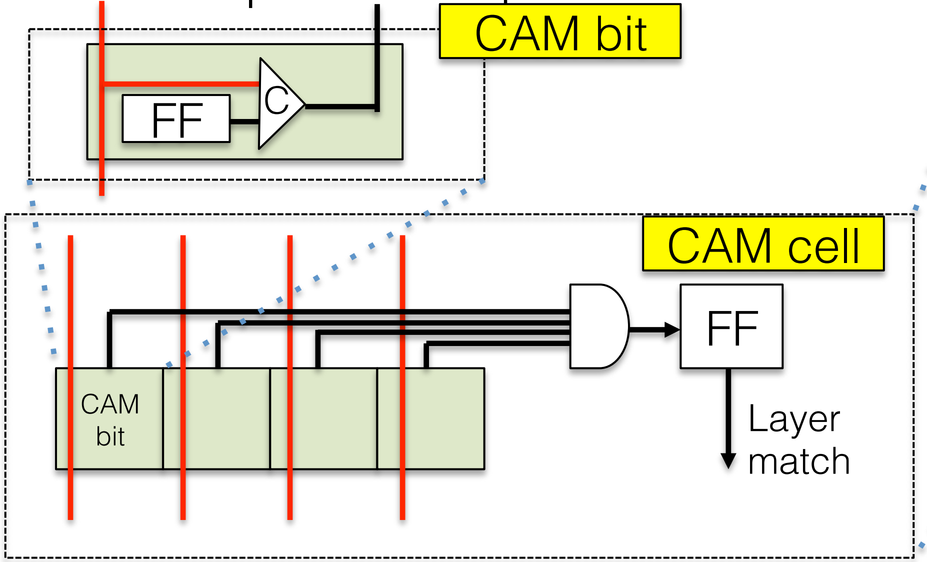


Larger bank size
Higher S/N

- High pattern density is needed for FTK
  - For high efficiency and higher resolution for better fake rejection power
- Solutions in FTK for good efficiency & fake rejection power :
  - Pursue higher pattern density, developing full custom chips (AMchip06)
  - Implement “Variable resolution AM” for optimal use of memory resource

Content Addressable Memory

- Functionality of the CAMs
  - Store the data
  - Compare the input data with stored data



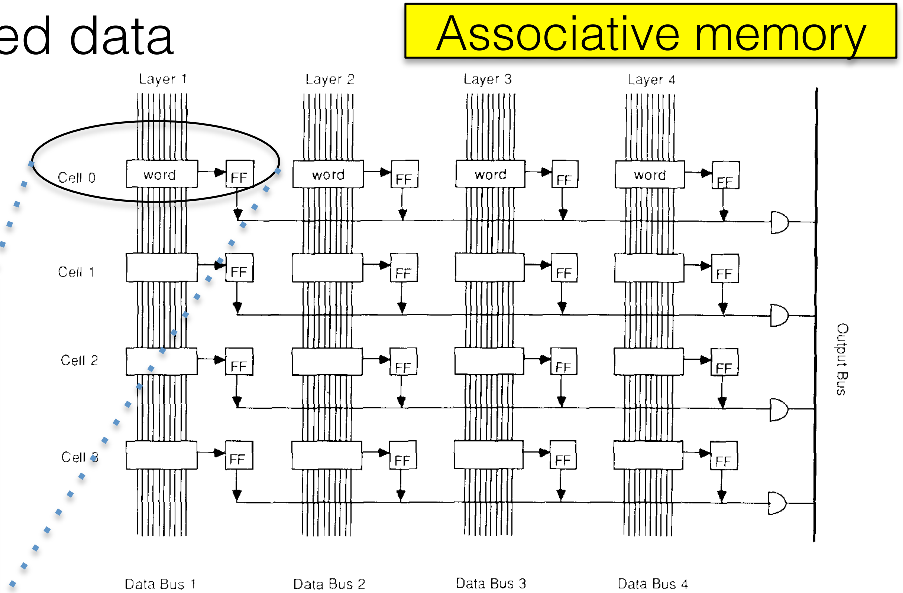
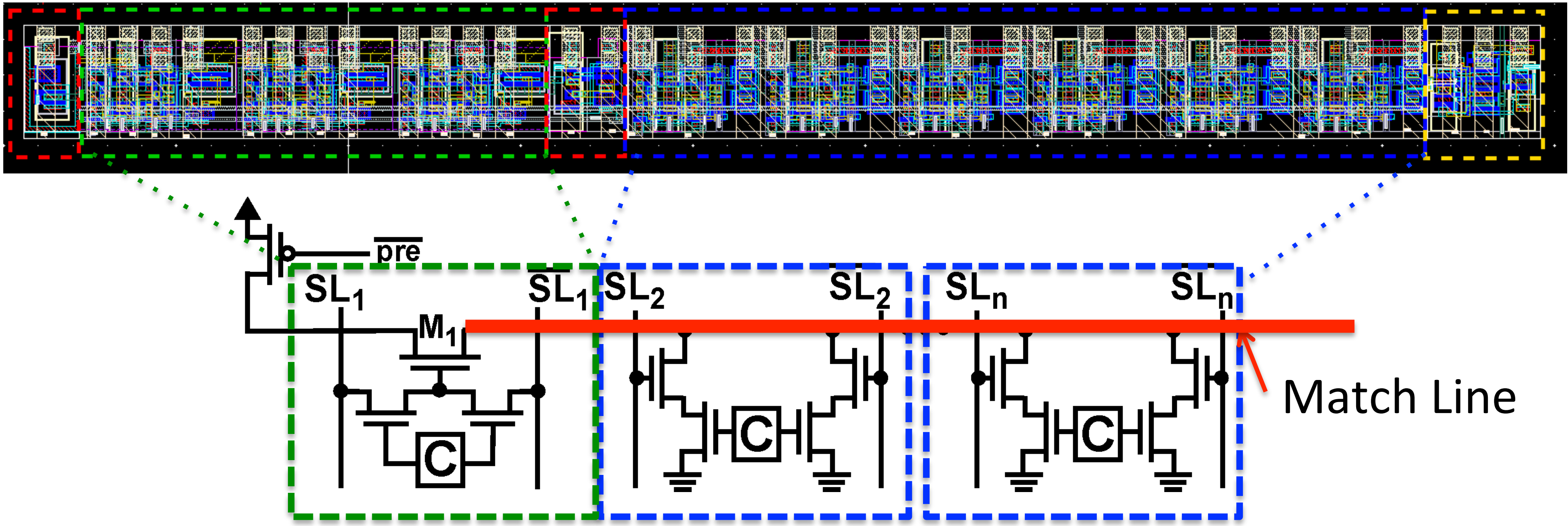


Fig. 3. Associative memory architecture.

- Associative memory is designed with CAM
  - Comparison layer level comparison is implemented with CAM
    - Bit level comparison (CAM bit)
    - Word level comparison (CAM cell) – 1 word consists of 18 bits
  - Track recognition is done by CAM cell x # layers (8) + majority logic

Full custom CAM cell for FTK



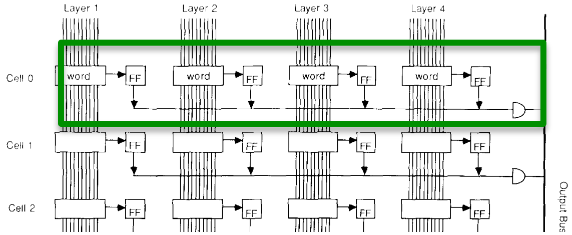
6 “NAND” CAM bits
(2.2um x 1.5um each)
9 transistors each

12 “NOR” CAM bits
(2.2um x 1.5um each)
9 transistors each

- For hit channel comparison in each layer
  - 18 CAM bits = 6 NAND CAM bits + 12 NOR CAM bits
  - The match line will be driven to high when all bits are matching
  - Size of the full custom CAM cell = 1.5 x 50 um^2

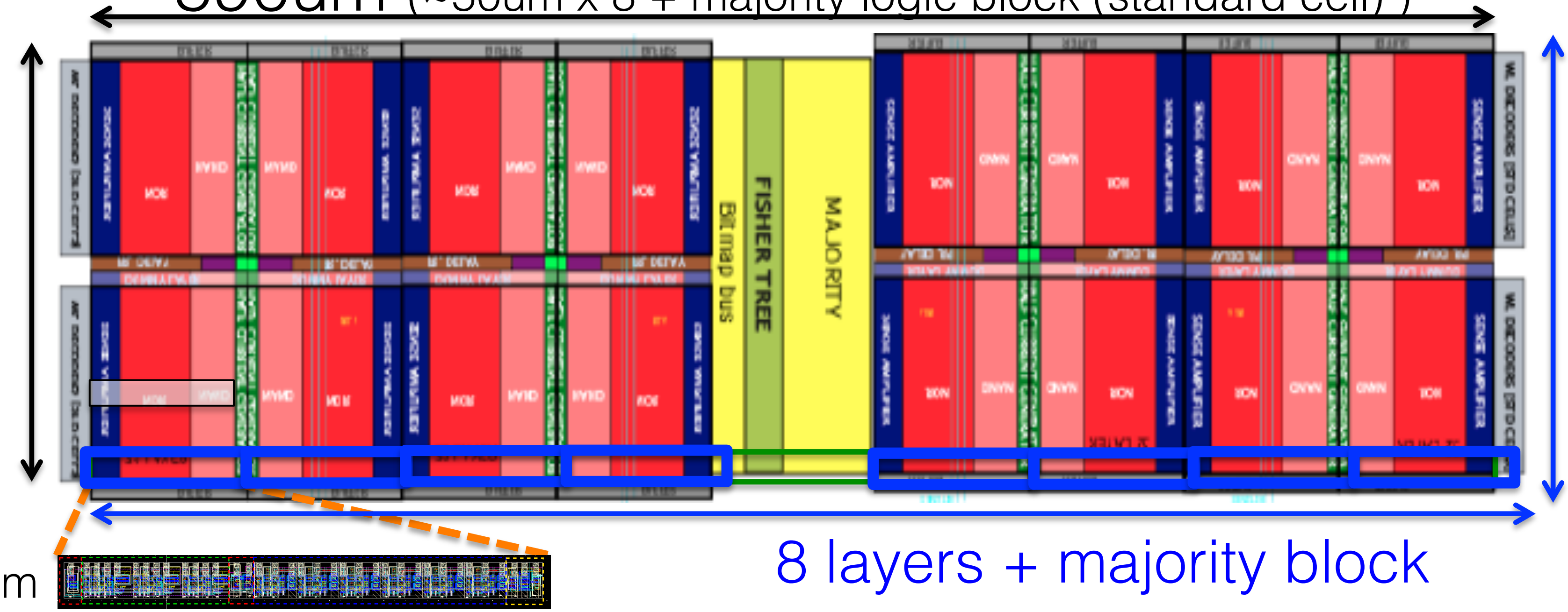
64-track-pattern block

- CAM Cell x 8 (for 8 layers)
  - 8 CAM Cell + Majority Logic



590um (~50um x 8 + majority logic block (standard cell))

109um (~1.5um x 64)



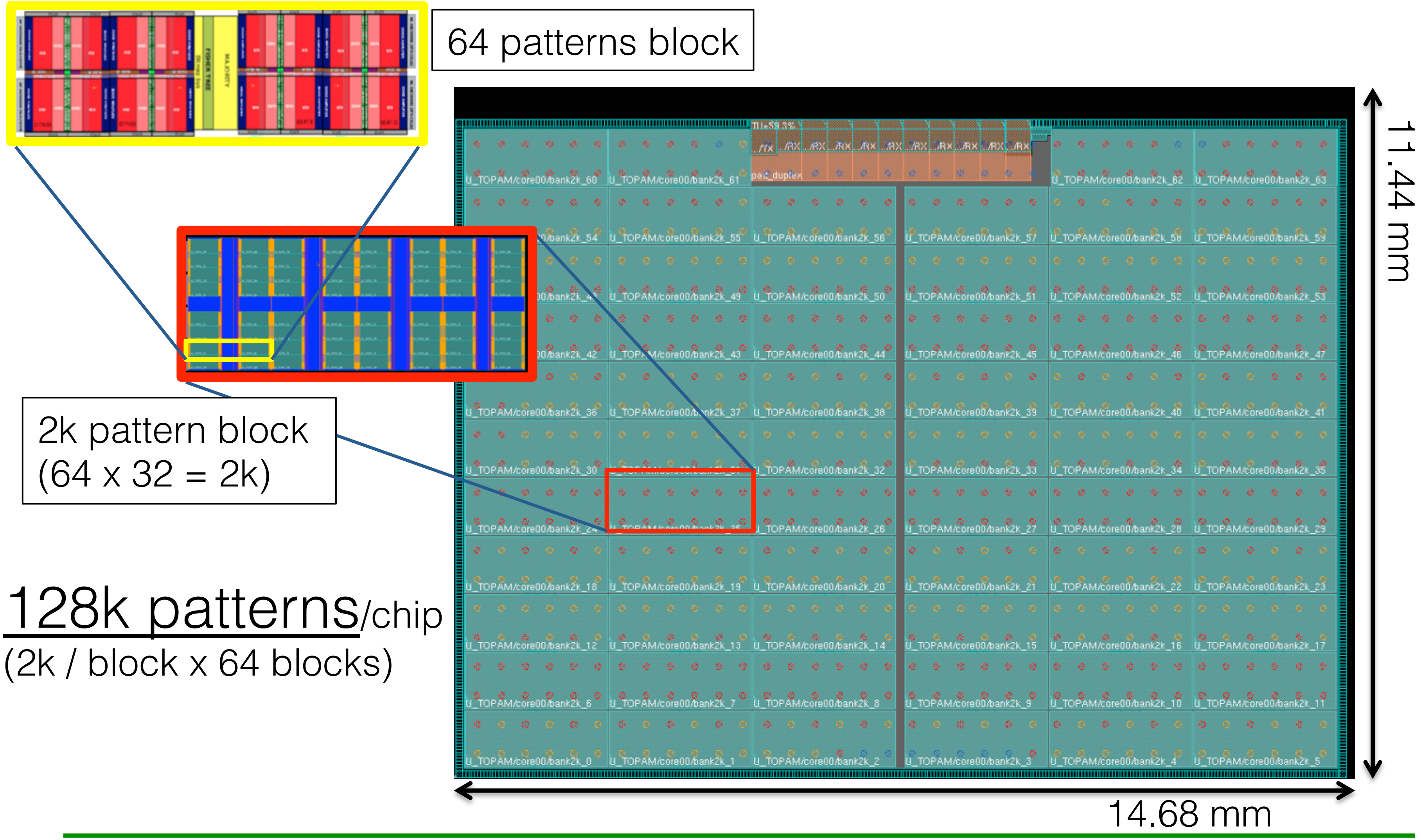
64 patterns

8 layers + majority block

1.5um

50um

Scale of chip design



64 patterns block

2k pattern block
(64 x 32 = 2k)

128k patterns/chip
(2k / block x 64 blocks)

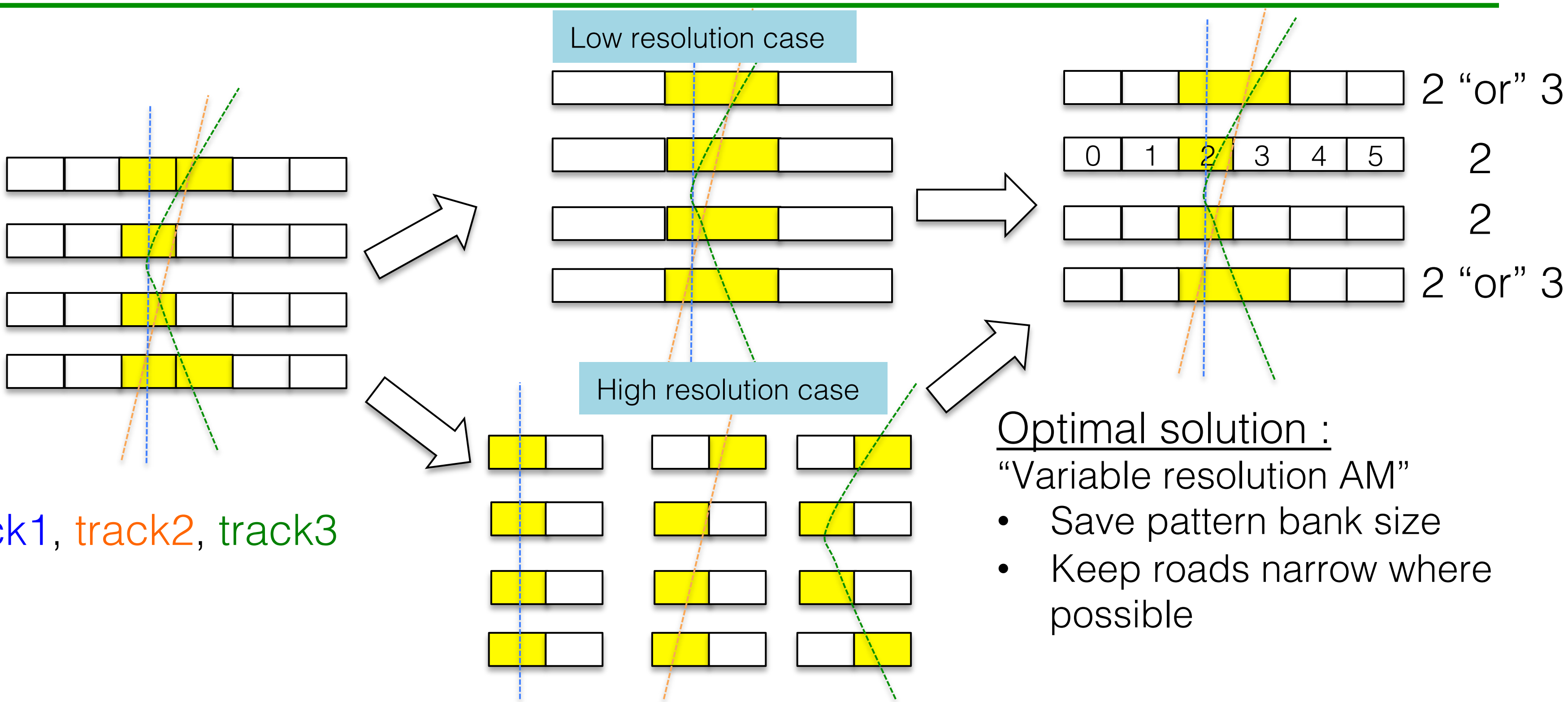
Associative Memory evolution

	Technology	Area	Patterns	Pattern Density (/cm^2)	Detector Layers	Speed (MHz)	I/O	For
SVT AM			128		6 (12bits/layer)	30	parallel bus	CDS SVT (1992)
AMchip03	180nm	100mm^2	5k	5k	6 (15.5bits/layer)	40	parallel bus	CDF SVT (2005)
AMchip04	**65nm**	14mm^2	8k	57k	**8** (18bits/layer)	**100**	parallel bus	FTK R&D
AMchip06	**65nm**	**160mm^2**	**128k**	**80k**	**8** (18bits/layer)	**100**	**SERDES**	**FTK**

- Many improvements between CDF SVT (2005) and FTK (2014)
  - Smaller CMOS technology, full custom cells for ATLAS FTK application
- Design improvement in past 10 years (compare with AMchip03)
  - Pattern density ~ 16 x AMchip03, CAM bit density ~ 25 x AMchip03
    - Factor 25 from smaller technology (~8) and design effort for full custom cell (~3)
  - Power consumption per bit comparison in CAM cells ~ 1/70 x AMchip03

AMchip06 allows 1 billion patterns with 8k chips in FTK system

Variable resolution AM



track1, track2, track3

Optimal solution :

“Variable resolution AM”

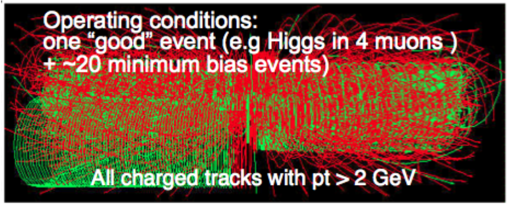
- Save pattern bank size
- Keep roads narrow where possible

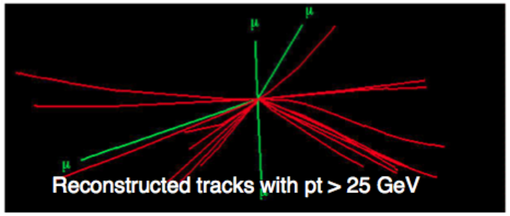
- CAM cell for FTK can use up to 6 ternary bits in word comparison
  - Ternary bit stores ‘0’, ‘1’, or ‘X (don’t care or wildcard)’, *IEEEE Journal of Solid State Cirtuits Vol. 41, NO. 3, March 2006*
- New feature of AM in FTK for most effective use of memory resource

Good efficiency with enough rejection for fakes with 1 billion patterns

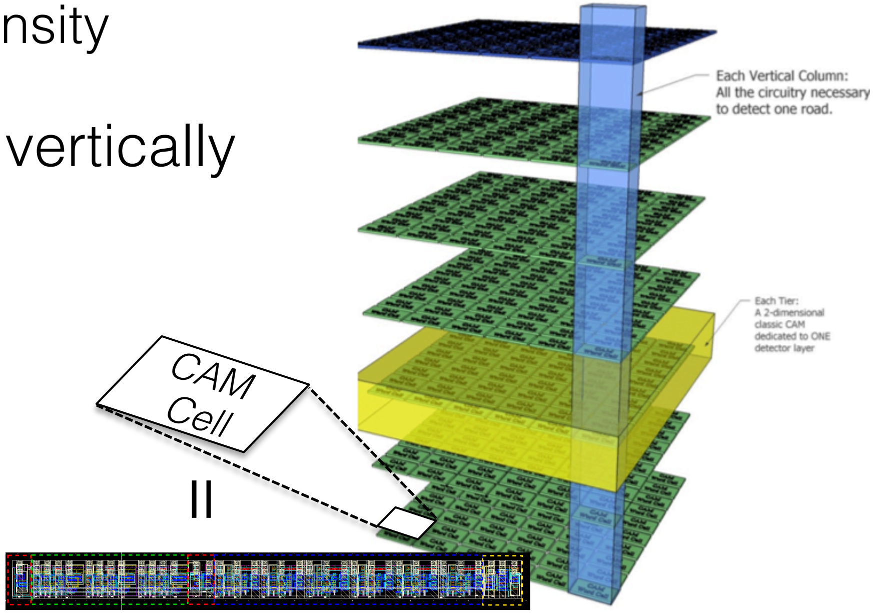
Beyond FTK : L1 tracking & possibility of vertical integrated AM

- Tracking information in early trigger stage is “must” in Run4 (HL-LHC)
  - Luminosity = 5 x 10^{34}
  - Pile-up multiplicity is up to 200 (peak)
- Challenges for Level 1 tracking
  - Shorter latency, higher event rate, and larger data volume than FTK
  - Needs for higher AM pattern density to achieve higher parallelism
- VIPRAM : CAM cells connected vertically
  - Higher pattern density
    - with same CMOS technology
  - Shorter layer match line in AM
    - Higher speed and/or low power


Operating conditions:
one “good” event (e.g Higgs in 4 muons)
+ ~20 minimum bias events)
All charged tracks with pt > 2 GeV



Reconstructed tracks with pt > 25 GeV



Each Vertical Column: All the circuitry necessary to detect one road.

Each Tier: A 2-dimensional classic CAM dedicated to ONE detector layer

Vertical Integrated Pattern Recognition AM (VIPRAM) projects @ Fermilab

http://inspirehep.net/record/1107828

Summary

- The associative memory approach allows us to do track reconstruction with short latency in the trigger
  - Concept of massive parallelism
  - The first application at LHC is ATLAS FTK for HLT
- Many cutting-edge techniques are applied to FTK implementation
  - ATCA based data formatting system
  - Higher AM chip memory density (1 billion patterns) and new variable resolution in associative memory
- FTK schedule as a phase I upgrade program
  - System level commissioning with prototype boards at CERN (currently on-going)
  - Board production & test (2015)
  - Commissioning with partial coverage with 1k AMchips (2015)
  - Operation with full coverage with 2k AMchips (2016)
  - Operation with full system with 8k AMchips (LHC Run3)

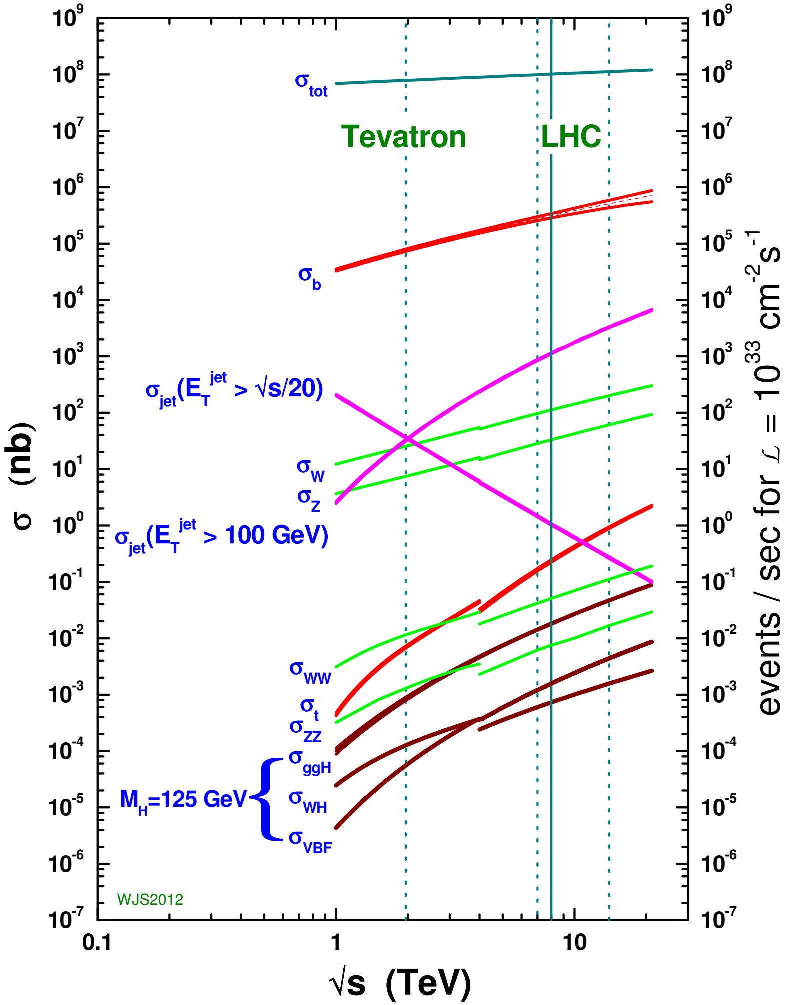
Also FTK experience provides foundation for further L1 tracking!

Stay tuned

and

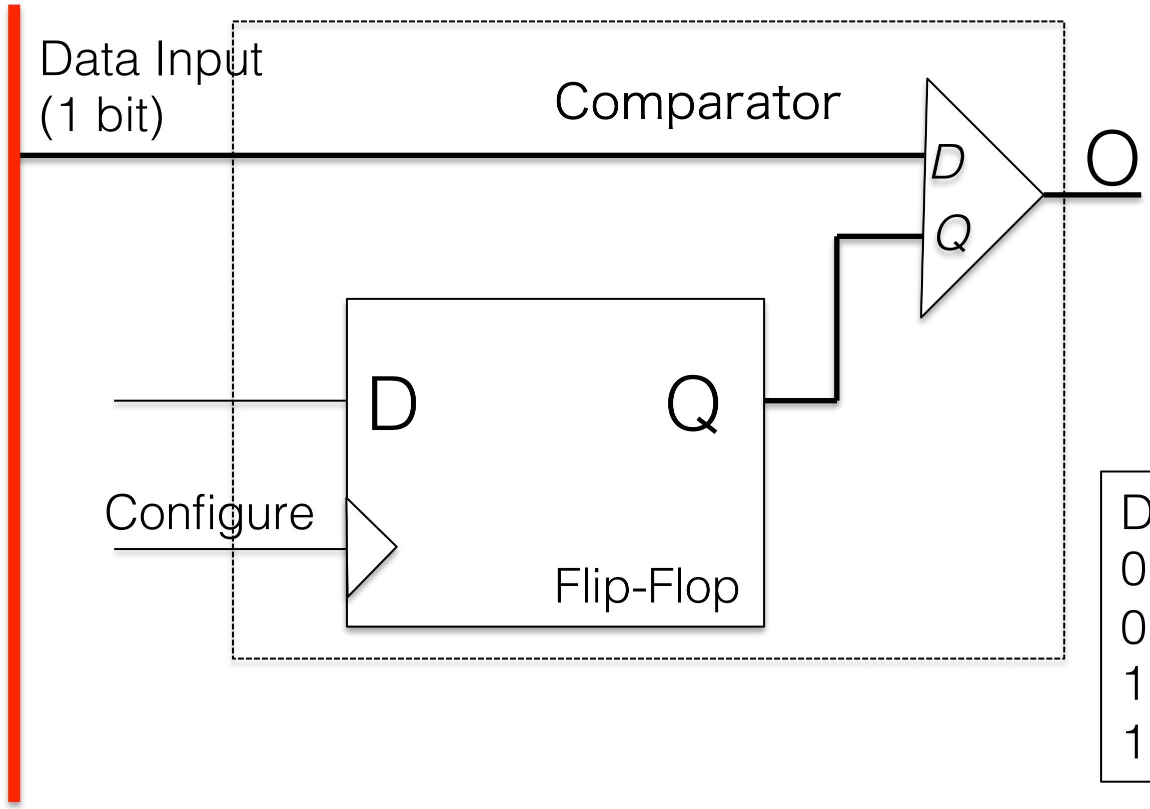
Thank you a lot for your attention

proton - (anti)proton cross sections



CAM bit

- For bit-level comparison consisting of Flip Flop (storage of data) and comparator

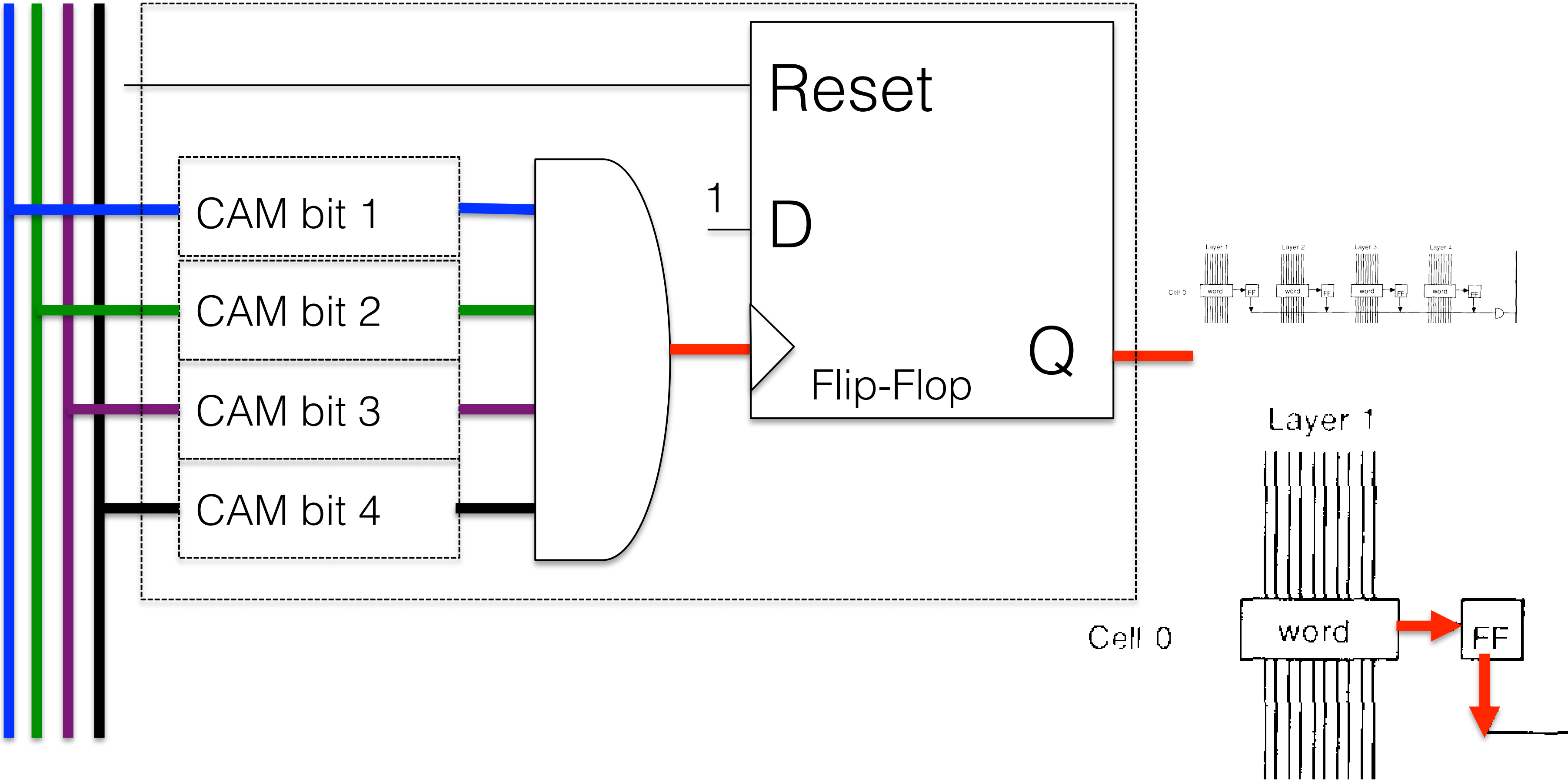


D	Q	O
0	1	0
0	0	1
1	1	1
1	0	0

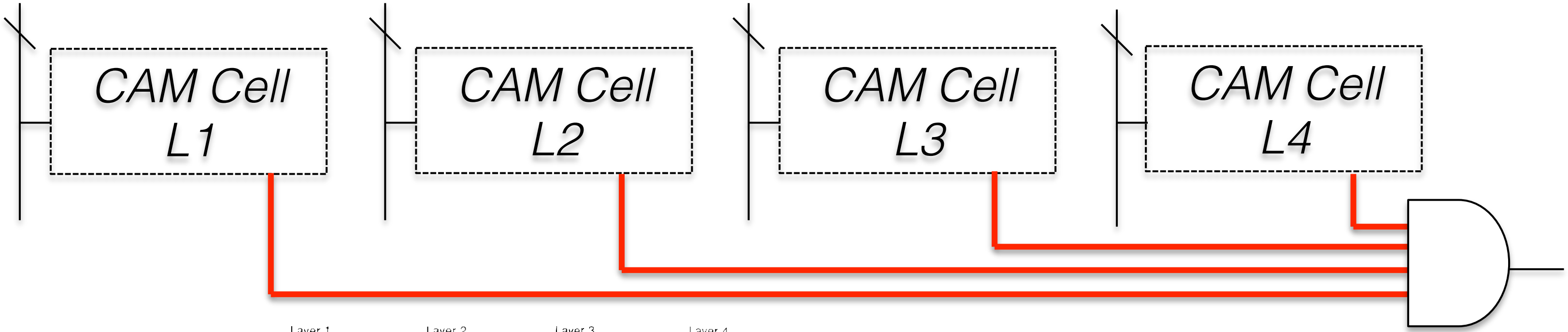
IEEEE Journal of Solid State Cirtuits Vol. 41, NO. 3, March 2006

CAM cell

- example for four-bit word case



Track pattern memory



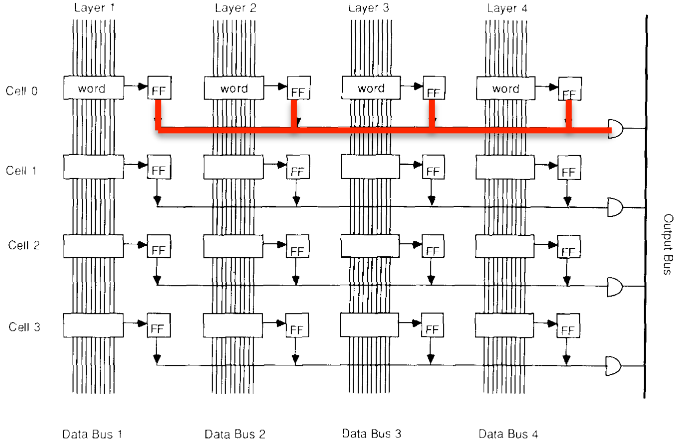
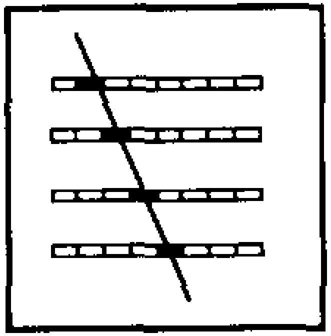


Fig. 3. Associative memory architecture.



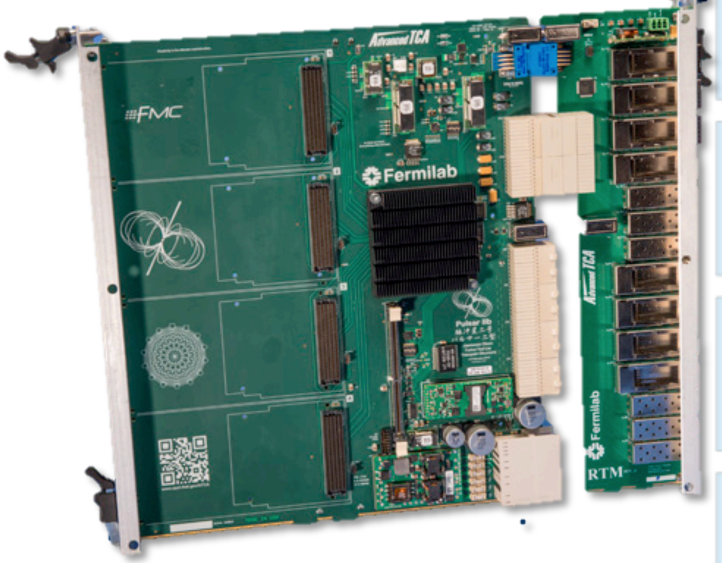
- Basic structure :
  - CAM Cell x # detector layers + Majority Logic = track pattern

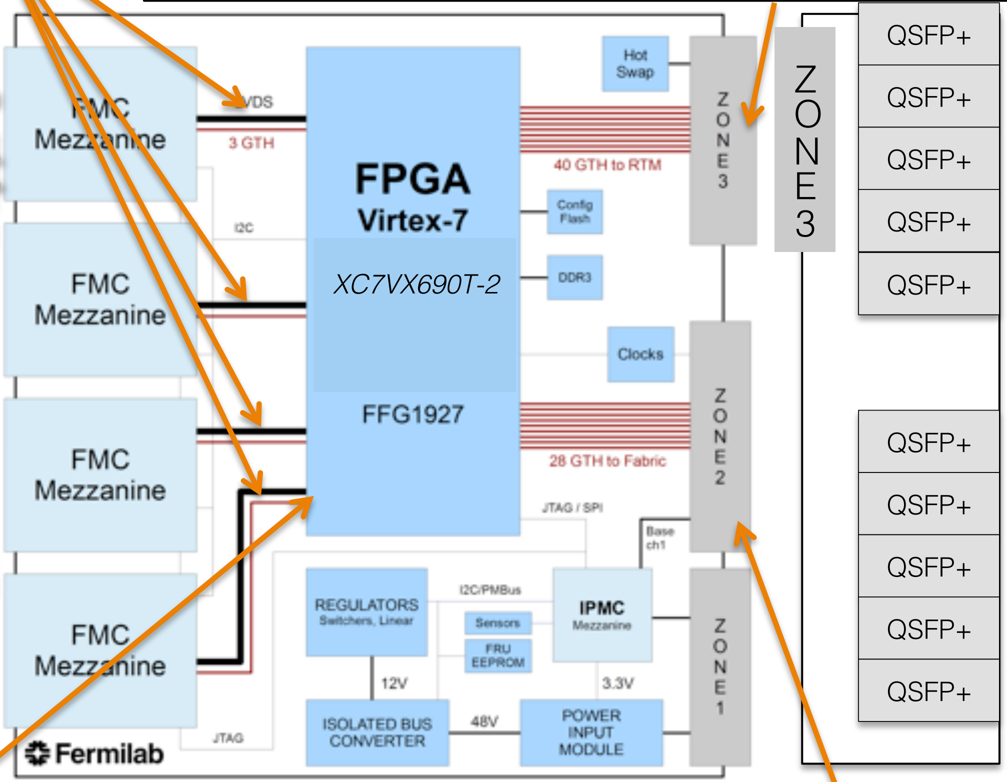
Required speed for FTK Data Formatter

Requirement is estimated for <μ> = 80 with enough margin

FMC connector
LVDS parallel bus
with 400Mb/s/ LVDS pair

ZONE3 for optical modules 6.4Gb/s x 38 fibers





V7 FPGA (VC7VX690T-2) :
main engine for data formatter

ZONE2 for ATCA backplane interconnect
6.25 Gb/s x 14 lanes

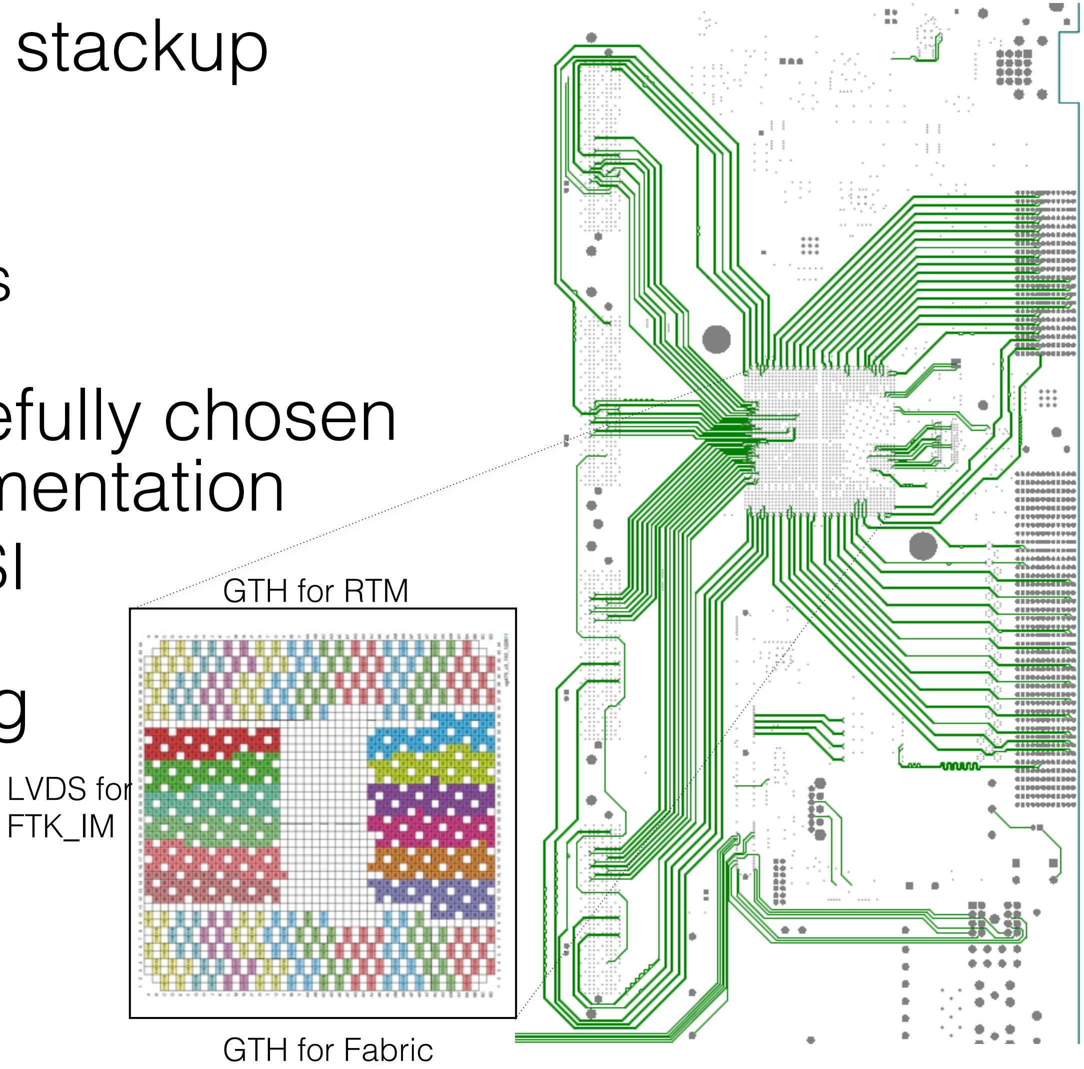
NOTE : This is requirement for FTK, and available speed on the board will be summarized in the later slides

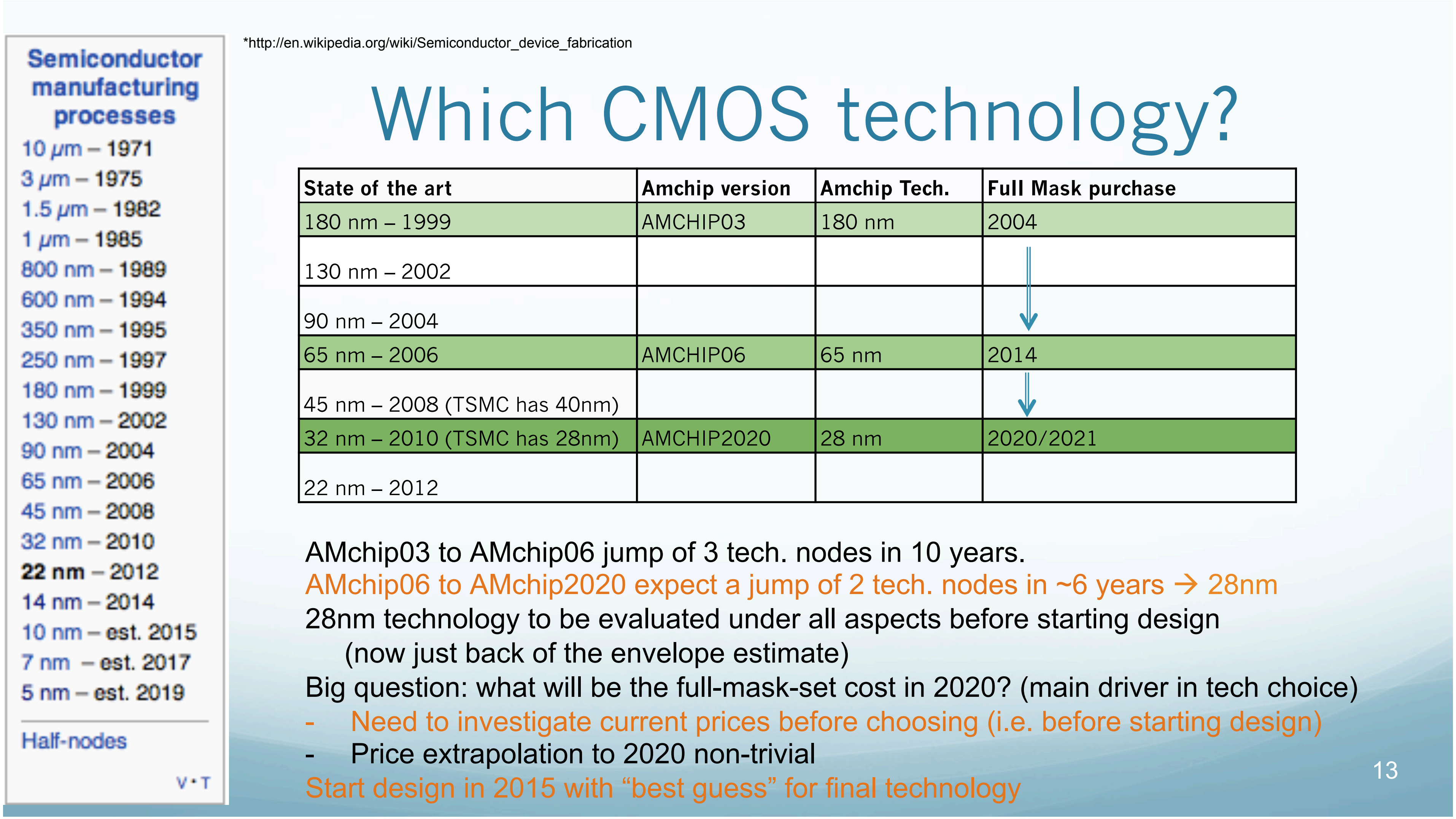
PCB design

- 14 Layers, symmetric stackup
  - 6 routing layers
  - 2 power planes
  - 6 solid ground planes

- Board material is carefully chosen for high speed implementation
  - Nelco N4000-13 EP SI

- Trace length matching in differential signals
  - GTH < 5 mil
  - LVDS < 50 mil
  - Clock < 5 mil





Semiconductor manufacturing processes

10 µm – 1971
3 µm – 1975
1.5 µm – 1982
1 µm – 1985
800 nm – 1989
600 nm – 1994
350 nm – 1995
250 nm – 1997
180 nm – 1999
130 nm – 2002
90 nm – 2004
65 nm – 2006
45 nm – 2008
32 nm – 2010
22 nm – 2012
14 nm – 2014
10 nm – est. 2015
7 nm – est. 2017
5 nm – est. 2019

Half-nodes

v · t

*http://en.wikipedia.org/wiki/Semiconductor_device_fabrication

Which CMOS technology?

State of the art	Amchip version	Amchip Tech.	Full Mask purchase
180 nm – 1999	AMCHIP03	180 nm	2004
130 nm – 2002			
90 nm – 2004			↓
65 nm – 2006	AMCHIP06	65 nm	2014
45 nm – 2008 (TSMC has 40nm)			↓
32 nm – 2010 (TSMC has 28nm)	AMCHIP2020	28 nm	2020/2021
22 nm – 2012			

AMchip03 to AMchip06 jump of 3 tech. nodes in 10 years.

AMchip06 to AMchip2020 expect a jump of 2 tech. nodes in ~6 years → 28nm

28nm technology to be evaluated under all aspects before starting design
(now just back of the envelope estimate)

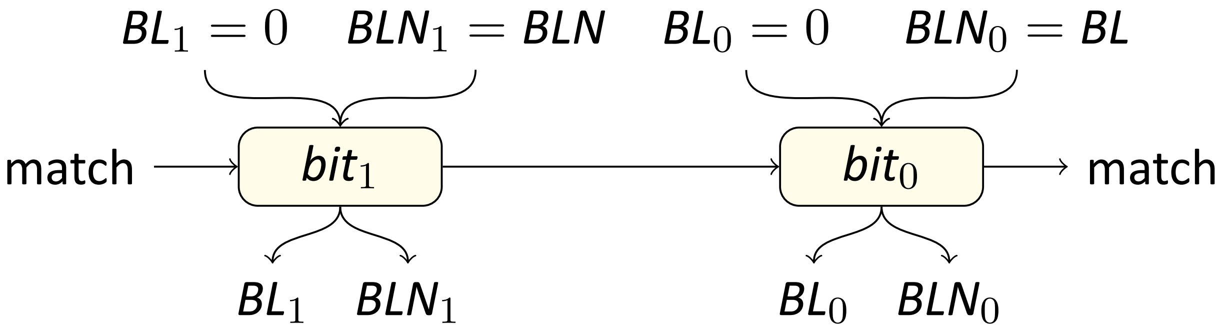
Big question: what will be the full-mask-set cost in 2020? (main driver in tech choice)

- Need to investigate current prices before choosing (i.e. before starting design)
- Price extrapolation to 2020 non-trivial

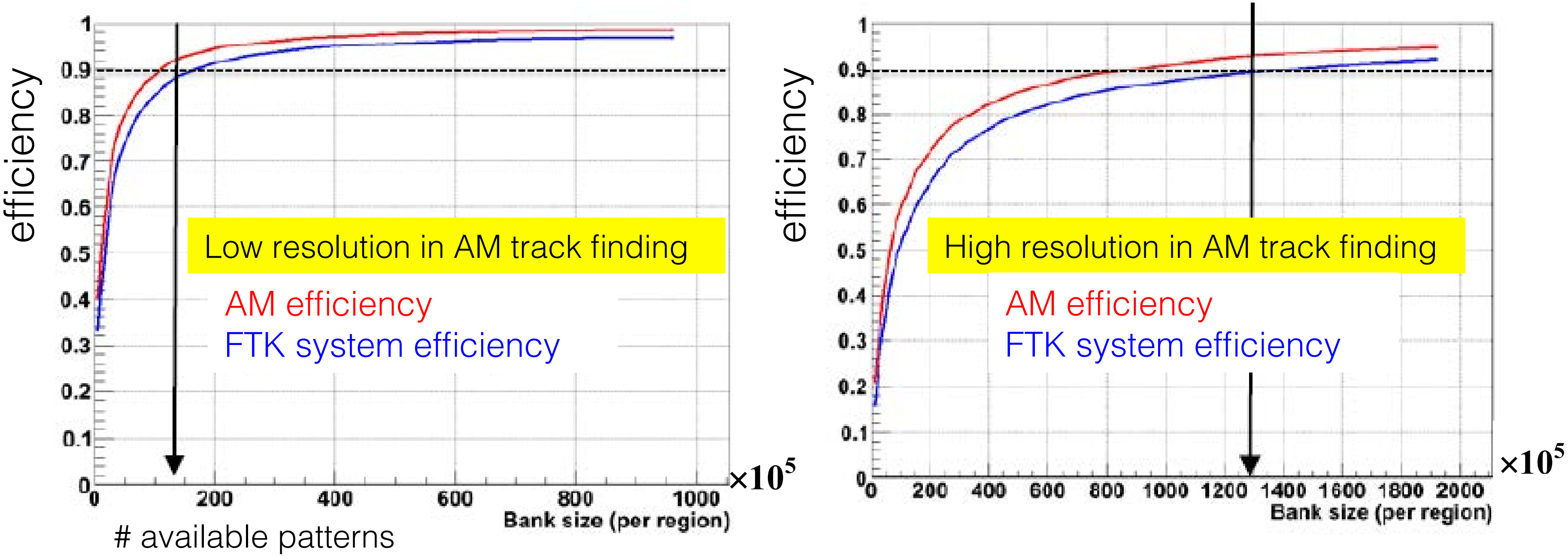
Start design in 2015 with “best guess” for final technology

With a non-complementary access to two bit cells we can encode don't care values. This is done by central logic and it's programmable: we can decide how many ternary bits have reconfiguring access to bit pairs

BL	BLN	2bit	match
0	1	01	1
0	1	10	0
1	0	10	1
1	0	01	0
0	1	00	1
1	0	00	1



AM chip patterns vs resolution



- Low resolution scenario
- High resolution scenario

Property of AM approach

- Input hits are compared with all stored patterns simultaneously
  - Massive “parallelism” of pattern recognition

- Processing time is linear in the number of hits
  - As soon as all the detected hits are loaded, the pattern recognition will be completed
  - Fast pattern recognition device

- Availability for optimization
  - Majority logic (such as 7 out of 8) for hit inefficiency

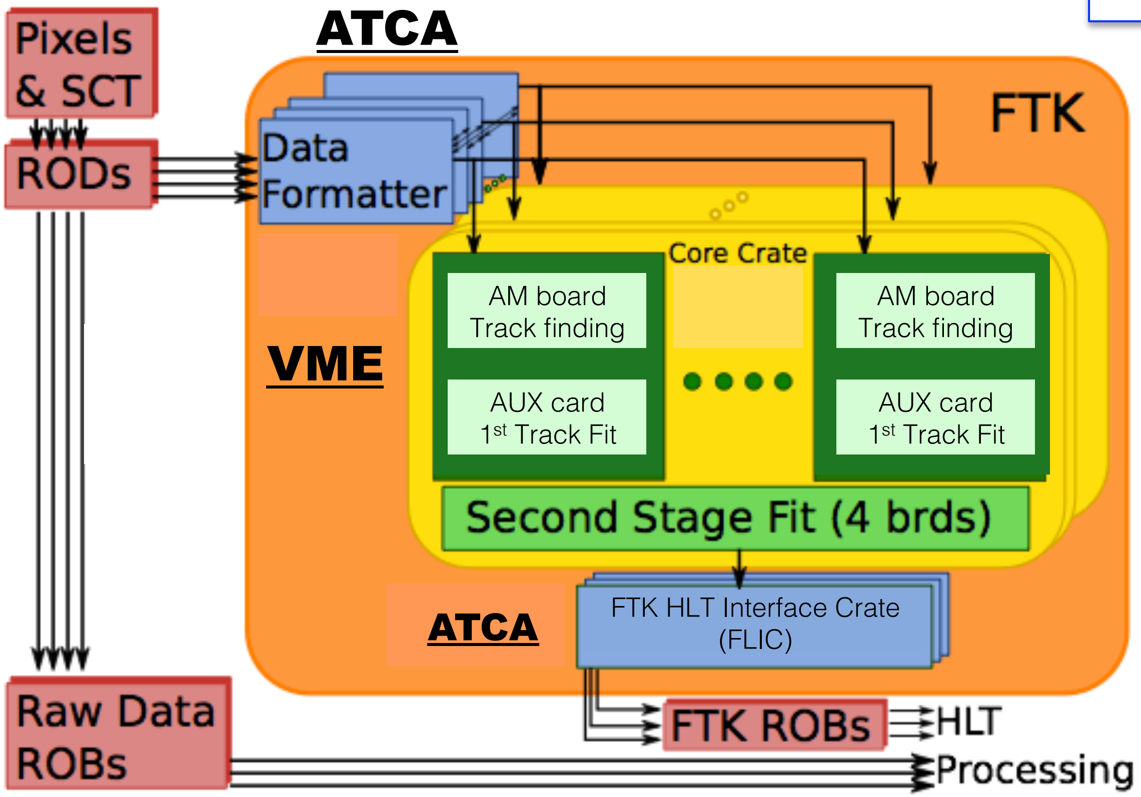
Fast TracKer system diagram

Transmitter to FTK

0. Copy hit data to FTK
(Dual S-link optical link interface)

Preparation for FTK tracking

1. Input clustering mezzanine (IM)
2. Data Formatter (DF)
- Organize SLINK input into 64 overlapping FTK eta-phi towers for parallel processing



Parallel processing

Tracking Engines (64 trigger towers)

3. AM Board + AM Chip
- Track finding with 8 layers on Associative Memory (AM) chips

4. AUX board
- 1st stage fitting with 8 layers
- Interface to the DF / AM board

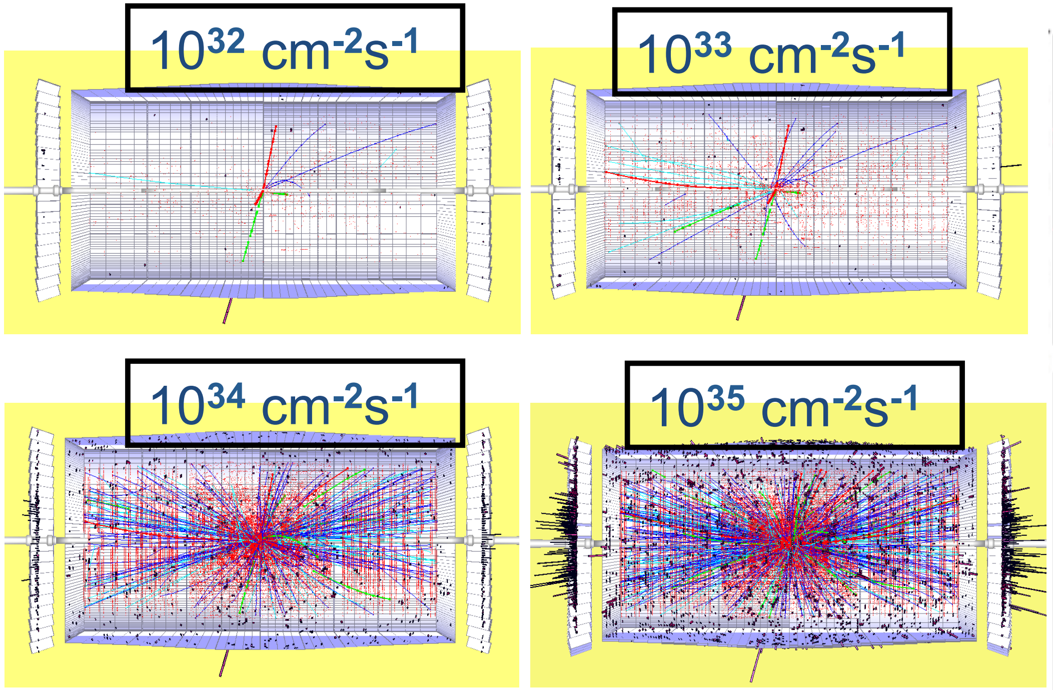
5. Second Stage Board (SSB)
- Track extrapolation to remaining layers
- 2nd stage fitting with 12 layers
- Global duplication removal

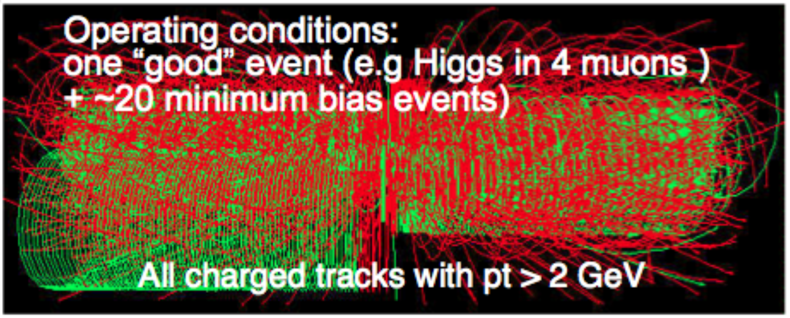
Interface to the HLT

6. FLIC board
transmission of tracks to the HLT ROSs using the standard ATLAS protocol

Beyond FTK : L1 tracking

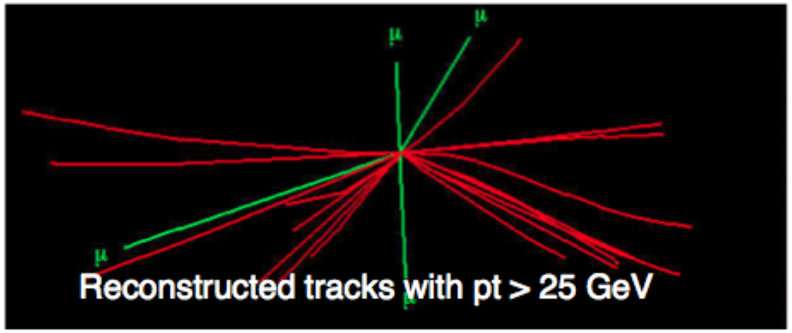
High Luminosity LHC (Run4) : Luminosity = 5 x 10^{34}
Expected pile-up event multiplicity average ~140 (peak ~ 200)
Tracking information in early trigger stage is "must"
in both ATLAS & CMS in order to maintain physics opportunity


10^{32} cm^{-2}s^{-1}
10^{33} cm^{-2}s^{-1}
10^{34} cm^{-2}s^{-1}
10^{35} cm^{-2}s^{-1}



Operating conditions:
one "good" event (e.g Higgs in 4 muons)
+ ~20 minimum bias events)

All charged tracks with pt > 2 GeV



Reconstructed tracks with pt > 25 GeV

Further challenges : shorter latency, higher event rate, larger data volume than FTK