

Development of FTK Architecture: A Fast Hardware Track Trigger for the ATLAS Detector

A. Annovi, M. Beretta, and P. Laurelli

INFN Frascati

E. Bossini, V. Cavasinni, F. Crescioli, M. Dell'Orso, P. Giannetti, M. Piendibene, G. Punzi, F. Sarri, I. Vivarelli, and G. Volpi

Univ. and INFN of Pisa

A. Boveia, E. Brubaker, F. Canelli, M. Dunford, A. Kapliy*, Y.K. Kim, C. Melachrinos, M. Shochet, and J. Tuggle

Univ. of Chicago

H. DeBerg, A. McCarn, and M. Neubauer

Univ. of Illinois at Urbana-Champaign

M. Franklin and C. Mills

Harvard Univ.

N. Kimura and K. Yorita

Waseda University

J. Proudfoot and J. Zhang

Argonne National Lab

L. Sartori

Univ. and INFN of Pisa and

Marie Curie Fellowship

L. Tripiccione

Univ. and INFN Ferrara

As the LHC luminosity is ramped up to the design level of $10^{34} \text{ cm}^{-2} \text{ s}^{-1}$ and beyond, the high rates, multiplicities, and energies of particles seen by the detectors will pose a unique challenge. Only a tiny fraction of the produced collisions can be stored on tape and immense real-time data reduction is needed. An effective trigger system must maintain high trigger efficiencies for the physics we are most interested in, and at the same time suppress the enormous QCD backgrounds. This requires massive computing power to minimize the online execution time of complex algorithms. A multi-level trigger is an effective solution for an otherwise impossible problem.

The Fast Tracker (FTK) is a proposed upgrade to the ATLAS trigger system that will operate at full Level-1 output rates and provide high quality tracks reconstructed over the entire detector by the start of processing in Level-2. FTK solves the combinatorial challenge inherent to tracking by exploiting the massive parallelism of Associative Memories (AM) that can compare inner detector hits to millions of pre-calculated patterns simultaneously. The tracking problem within matched patterns is further simplified by using pre-computed linearized fitting constants and leveraging fast DSP's in modern commercial FPGA's. Overall, FTK is able to compute the helix parameters for all tracks in an event and apply quality cuts in approximately one millisecond. By employing a pipelined architecture, FTK is able to continuously operate at Level-1 rates without deadtime.

The system design is defined and studied using ATLAS full simulation. Reconstruction quality is evaluated for single muon events with zero pileup, as well as WH events at the LHC design luminosity. FTK results are compared with the tracking capability of an offline algorithm.

1. Introduction

The Large Hadron Collider will collide proton bunches every 25 nanoseconds with a center-of-mass energy of 14 TeV. At the design luminosity, each collision on average produces 23 minimum-bias interactions that result in high detector occupancy and create a challenging environment for event readout and reconstruction. On one hand, limited data store bandwidth demands a significant online rate reduction of 5-6 orders of magnitude. On the other hand, events with interesting physics signatures must be selected very efficiently from the vast LHC background.

The ATLAS experiment employs a sophisticated three-level trigger system to achieve these goals [1, 2].

Level-1 selection is performed in dedicated hardware that uses coarse-granularity information from calorimeters and muon spectrometers to apply cuts to a variety of objects, such as jets, muons, electromagnetic clusters, and missing energy. Although adding tracking information at this stage would be extremely beneficial to discern certain physics objects, the small timing window available to Level-1 ($2.5 \mu\text{s}$) makes this currently impossible. Moreover, even projected CPU farms that constitute the Level-2 trigger cannot perform global track reconstruction within their time budget of about 10 ms. Instead, Level-2 does limited tracking inside Regions of Interest (ROI) identified by the Level-1 trigger.

FastTracker (FTK) [3, 4] is a proposed dedicated

hardware track processor inspired by the Silicon Vertex Trigger (SVT) [5] from the CDF detector. FTK operates in parallel with the normal silicon detector readout following each Level-1 trigger and reconstructs tracks over the entire detector volume (up to $|\eta|$ of 2.5) in under a millisecond. Having tracks available by the beginning of Level-2 processing allows reduced Level-1 p_T thresholds since the Level-2 trigger is able to reject non-interesting events more quickly. Furthermore, since Level-2 is freed up from tracking, the extra processing time becomes available for more advanced online algorithms.

2. Data Flow

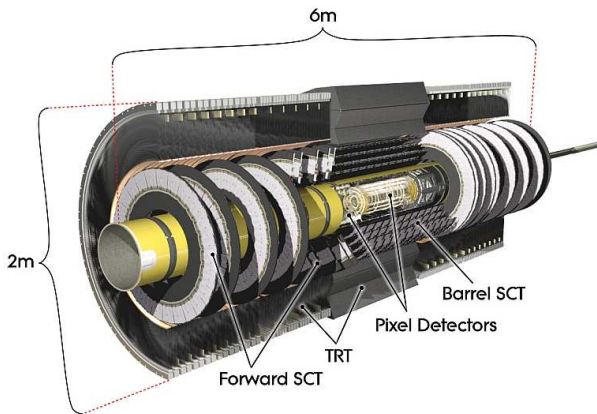


Figure 1: ATLAS Inner Detector. FTK uses data from the Pixel and SCT detectors.

In order to perform tracking with good efficiency and resolution, FTK uses data from the two innermost subsystems of the ATLAS Inner Detector [6] (Fig. 1). Pixel sensors provide a two-dimensional measurement of hit position and include over 80 million readout channels. SCT layers are arranged in pairs of axial and narrow-angle stereo strips and consist of 6 million channels. A typical track passes through 11 detector layers and can be reconstructed from the 14 coordinate measurements ($3 \cdot 2$ from two-dimensional pixels and 8 from SCT). Custom-designed optical splitters duplicate Pixel and SCT readout data and send it to FTK - at full Level-1 rate. Since FTK non-invasively eavesdrops on the Inner Detector data, it easily integrates with the current ATLAS trigger system (Fig. 2).

In order to increase the overall throughput of the system, FTK splits incoming data into 8 or more *regions* in ϕ with sufficient overlap to account for inefficiencies at the edges. Each region is served by a separate crate that consists of the subsystems shown in Fig. 3. Raw Pixel and SCT data are first received by the Data Formatter that uses fast FPGA's to perform clustering [7]. Clustered hits proceed into the Data

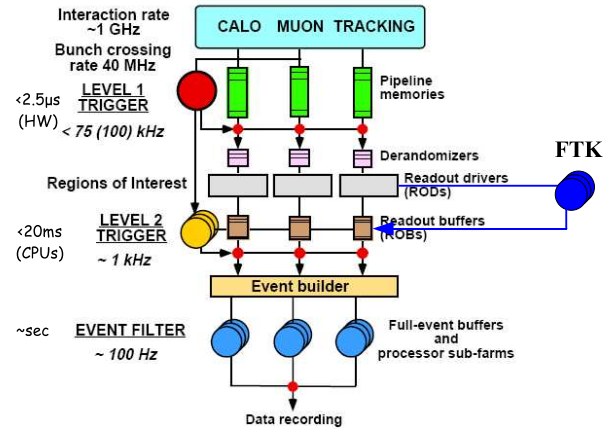


Figure 2: ATLAS trigger system and its integration with FTK.

Organizer, where they are buffered and merged into coarse superstrips used in pattern recognition. The superstrips are then sent into a pipelined array of Associative Memory boards that perform fast pattern finding using a pre-calculated table of particle trajectories. Matched patterns are reconnected with their corresponding full-resolution hits in the Data Organizer and sent to the Track Fitters. After removal of duplicate tracks, the resulting set of FTK tracks is saved in the track data Read-Out Buffer (ROB) and made available to Level-2 processors.

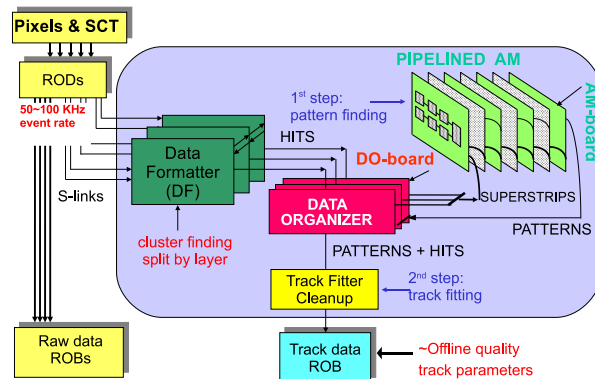


Figure 3: FTK Data Flow

3. Pattern Recognition in Associative Memories

Luminosities above $10^{34} \text{ cm}^{-2} \text{ s}^{-1}$ combined with 86 million readout channels create a unique combinatorial challenge for tracking. FTK overcomes this with the help of specialized hardware called Associative Memory (AM) - a massive, ultra-fast lookup table that enumerates all realistic particle trajectories (pat-

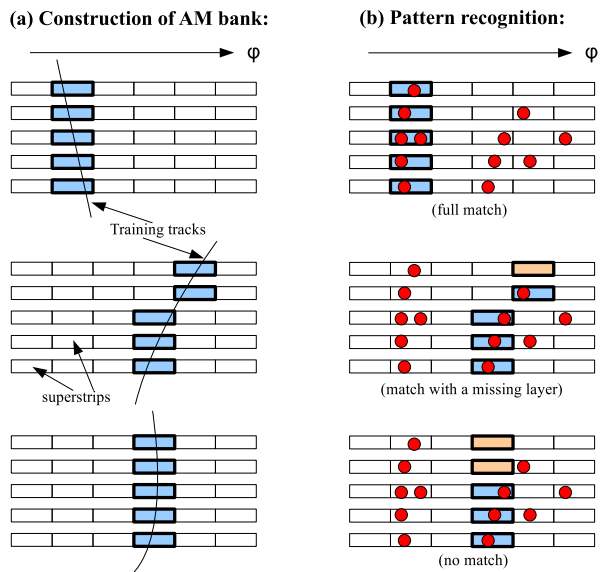


Figure 4: Associative Memory and pattern bank operation

terns) through the 11 detector layers [8]. In order to keep the size of the trajectory lookup under control, detector hits are merged into coarse-resolution *superstrips* having a width of a few millimeters¹. The pattern bank is precalculated either from single track Monte-Carlo or from real data events (Fig. 4(a)) and stored in the AM boards.

Each pattern in the AM includes its own comparison logic. When hits from a given event enter AM boards, they are simultaneously compared with millions of pre-stored patterns. In order to account for inefficiencies in individual detector layers, FTK also matches patterns with one missing layer (Fig. 4(b)).

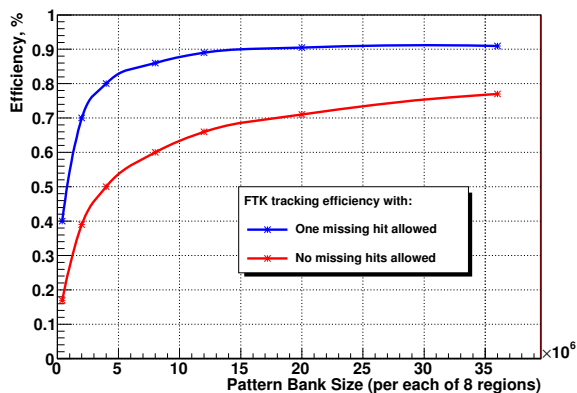


Figure 5: Efficiency as a function of pattern bank size for a particular choice of superstrip widths.

¹These coarse superstrips are only used in the pattern recognition stage; all final fits are performed with full resolution hits.

Fig. 5 shows the efficiency for muon track reconstruction as a function of pattern bank size for the pixel superstrip size of 3 mm and SCT superstrip size of 5 mm. With missing-layer pattern matching, efficiency quickly rises to the 90% level.

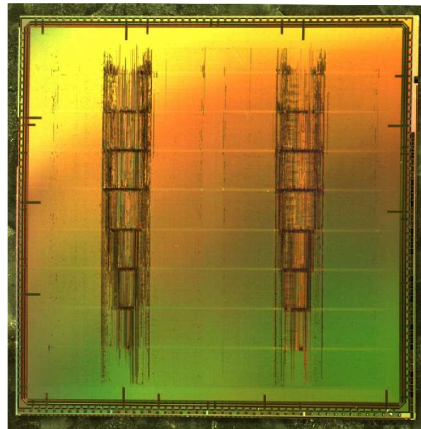


Figure 6: AM chip currently used in CDF detector at Fermilab

Fig. 6 shows the current AM chip used in the CDF detector at Fermilab. It uses 0.18 μm custom cells and contains up to 2,500 patterns in the 12-layer configuration. Using standard cells with 90 nm technology, the capacity can be increased to 10,000 patterns per chip, with another factor of two gain possible with a custom cell design.

4. Track Fitting

FTK computes five track helix parameters (curvature, d_0 etc) and a χ^2 quality of fit from the *full resolution* hits within each matched pattern. Since patterns are constructed from reduced-granularity superstrips, multiple full-resolution hits can belong to a given superstrip. This results in some ambiguity, which is resolved by fitting all combinations within the superstrips.

Performing full χ^2 minimization with respect to five parameters is an extremely slow procedure. Instead, FTK reduces the track fitting problem to a set of scalar products, which can be computed efficiently using DSP units in modern commercial FPGA's. This is done by arranging geometrically similar patterns into a number of groups (called *sectors*), so that within each sector the relationship between hit positions (x_j) and track parameters (p_i) is approximately linear:

$$p_i = \sum_{j=1}^{14} c_{ij} \cdot x_j + q_i \quad (1)$$

The fitting coefficients for each sector are precomputed from the same training data that was used in

pattern generation. An added advantage of this approach is that when real detector hits are used in training, misalignments and other detector effects are automatically taken into account.

Overall, the linearized approach allows FTK to achieve near-offline resolution with a fitting rate of about 1 fit per nanosecond.

5. Performance

Table I - FTK and offline resolutions for tracks with $p_T > 1 \text{ GeV}$ and $|\eta| < 1$

Track Parameter	$\sigma(\text{FTK})$	$\sigma(\text{offline})$
$1/(2p_T)[c/GeV]$	$7.4 \cdot 10^{-3}$	$6.6 \cdot 10^{-3}$
$\phi[\text{rad}]$	$9.5 \cdot 10^{-4}$	$6.3 \cdot 10^{-4}$
$d_0[\text{cm}]$	$5.3 \cdot 10^{-3}$	$3.3 \cdot 10^{-3}$
$\cot(\theta)$	$2.0 \cdot 10^{-3}$	$1.4 \cdot 10^{-3}$
$z_0[\text{mm}]$	$2.1 \cdot 10^{-2}$	$1.9 \cdot 10^{-2}$

Table I compares FTK track parameter resolutions for muons with an offline algorithm. Overall performances are comparable; in particular, the FTK impact parameter resolution is equal to that of offline with an additional 30 microns added in quadrature (Fig. 7). FTK reconstruction remains robust in higher pileup environments (Fig. 8).

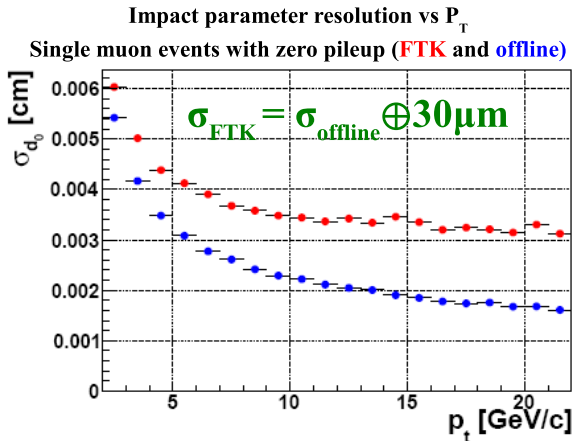


Figure 7: FTK reconstruction performance for zero-pileup muons

Preliminary timing estimates obtained from $10^{34} \text{ cm}^{-2} \text{ s}^{-1}$ simulation show that FTK is able to reconstruct complex events in about 1 ms. At higher luminosities, the number of fits performed in Track Fitters can become excessively large. This can be dramatically reduced by narrowing the superstrip width or modifying the pattern recognition and fit strategy. Several potential approaches have been identified and

simulated, promising to reduce FTK processing time by more than an order of magnitude.

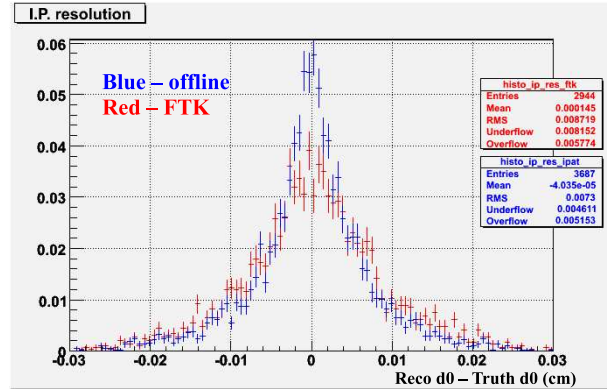


Figure 8: Impact Parameter resolution for all primary tracks with $p_T > 1 \text{ GeV}$ in WH ($H \rightarrow b\bar{b}$) events at the design luminosity - comparison between FTK and offline tracking

5.1. Physics Implications

Offline-quality b-tagging efficiency and light quark rejection can be achieved by using the savings in tracking time to apply more sophisticated b-tagging algorithms at Level-2. Fig. 9 compares b-tagging performance of FTK tracks with that of offline tracks using a simple transverse impact parameter likelihood algorithm. FTK tracks provide tagging performance competitive with offline tracks.

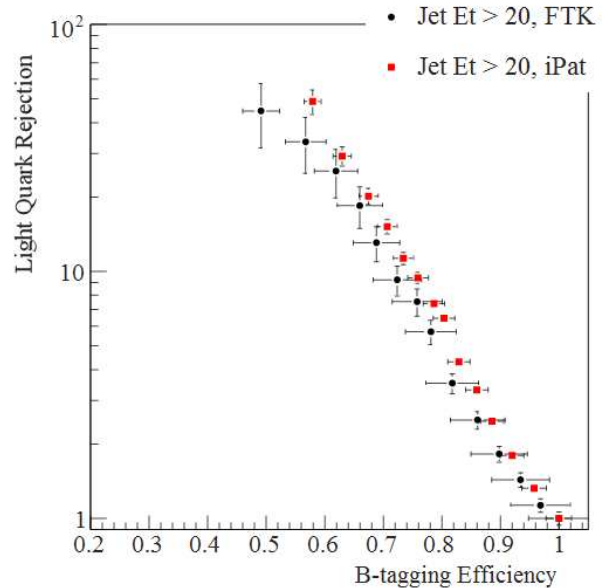


Figure 9: Likelihood ratio b-tagging performance with FTK and offline tracks using the same algorithm.

Studies are underway to quantify the efficiency and background rejection achieved with FTK tracks with respect to other high-Pt Level-2 objects, including tau-jets and isolated leptons.

6. Conclusions and Outlook

FTK performs global track reconstruction at the full Level-1 trigger rate and naturally integrates with the current ATLAS data acquisition system. Using massively parallel Associative Memories, it will provide a complete list of three-dimensional tracks at the beginning of Level-2 processing, including tracks outside of the Regions of Interest. The extra time saved by FTK can be used in Level-2 to apply more advanced algorithms and ultimately extend the physics reach of the detector.

FTK robustly and quickly reconstructs tracks at the LHC design luminosity and produces efficiencies and resolutions on par with offline tracking. Studies are undergoing to evaluate the performance of the system under higher pileup conditions ($3 \cdot 10^{34} \text{ cm}^{-2} \text{ s}^{-1}$).

We expect to produce a Technical Design Report in the fall of 2009 and the first board prototypes in 2010. The entire system will be ready in time for the LHC Phase I shutdown.

References

- [1] ATLAS Level-1 Trigger Group, *ATLAS trigger TDR Level-1 Trigger Technical Design Report, ATLAS TDR-12*, 1998.
- [2] ATLAS Collaboration, *ATLAS Trigger Technical Design Report, ATLAS TDR-12*, 2003.
- [3] A. Annovi et al. *The Fast Tracker Processor for Hadronic Collider Triggers. IEEE Trans. Nucl. Sci.*, **48**
- [4] A. Annovi et al. *Hadron Collider Triggers with High-Quality Tracking at Very High Event Rates. IEEE Trans. Nucl. Sci.*, **51**
- [5] A. Bardi et al. *SVT: an Online Silicon Vertex Tracker for the CDF Upgrade. NIM. A*, **409**(1-3):658-661, 1998
- [6] ATLAS Collaboration, *ATLAS Inner Detector TDR volume 1*, CERN/LHCC/97-16.
- [7] A. Annovi. *A Fast General-Purpose Clustering Algorithm Based on FPGAs for High-Throughput Data Processing*. [Proceedings from 11th Pisa Meeting on Advanced Detectors], 2009
- [8] *A VLSI Processor for Fast Track Finding Based on Content Addressable Memories. IEEE Trans. Nucl. Sci.*, **53**(4):2428-2433, 2006