PROCEEDINGS OF SCIENCE

Variable resolution Associative Memory optimization and simulation for the ATLAS FastTracker project

A. Annovi[a], A. Castegnaro[a], P. Giannetti[b], Z. Jiang[c], C. Luongo[*b], C. Pandini[d], C. Roda[eb], M. Shochet[c], L. Tompkins[c], G. Volpi[a]

[a] *INFN Frascati, Via E. Fermi 40, 00044 Frascati, Roma, Italy*
[b] *INFN Pisa, Largo Bruno Pontecorvo 3, 56127 Pisa, Italy*
[c] *University of Chicago, 5801 S Ellis Ave Chicago, IL 60637, United States*
[d] *Università di Milano, Via Conservatorio 7, 20122 Milano, Italy*
[e] *Università di Pisa, Largo B. Pontecorvo 3, 56127 Pisa, Italy*
E-mail: alberto.annovi@lnf.infn.it, andrea.castegnaro@lnf.infn.it, paola.giannetti@pi.infn.it, zihao.jiang@cern.ch, carmela.luongo@pi.infn.it, carloenrico.pandini@studenti.unimi.it, chiara.roda@cern.ch, shochet@hep.uchicago.edu, lauren.a.tompkins@gmail.com, guido.volpi@lnf.infn.it

ATLAS is planning to use a hardware processor, the Fast Tracker (FTK), to perform on-line track reconstruction at the level−1 trigger event output rate (100 kHz). The processor can perform this task using a very large number of precalculated track patterns stored in a dedicated bank and processing those with a sufficient number of hits in a given event.
In order to obtain a better trade-off between the number of patterns and the number of fits needed to perform track reconstruction, FTK exploits variable resolution pattern matching. This is carried out by a dedicated device, the Associative Memory (AM) chip, which is inspired from ternary commercial Content Addressable Memory (CAM) and includes ternary logic. This architecture is able to employ the variable resolution feature in which the width of each pattern can be varied layer by layer.
We have studied different methods of building the variable resolution pattern bank. We show how this new feature achieves the goal of having few enough patterns to fit in the hardware, while maintaining good efficiency and the required rejection against random combinations of hits for the luminosities and pileup conditions expected at the LHC after its Phase-I upgrades.

11th International Conference on Large Scale Applications and Radiation Hardness of Semiconductor Detectors
3-5 July 2013
Auditorium Cassa di Risparmio di Firenze, via Folco Portinari 5, Florence, Italy

[*] Speaker.

PoS(RD13)014

1. Introduction

ATLAS is one of the two general purpose detectors studying proton-proton collisions at the Large Hadron Collider (LHC) located at the CERN laboratory in Geneva, Switzerland [1]. One of the characteristic features of hadron colliders is that the most interesting processes are rare and hidden under an extremely large background. Only a very limited fraction of the produced data can be transferred to offline storage, so it is necessary for a multi-level trigger [2] to perform large real-time data reduction.

On-line track reconstruction can be a very important component in triggering at the LHC, especially during high luminosity running. The tracking allows particles produced in the hard proton-proton collision to be separated from those produced in the many overlapping collisions (pile-up). The Fast Tracker (FTK) system will perform this task for the ATLAS trigger [3].

The FTK hardware processor is an evolution of the CDF Silicon Vertex Trigger (SVT) [4, 5] which did track reconstruction in a few microseconds and proved to be an essential part of the CDF trigger.

The FTK system will operate at full level-1 trigger output event rate (100kHz) and provide high quality global track reconstruction at the beginning of level-2 trigger processing. The freed-up level-2 processor farm execution time will allow the downstream trigger algorithms to improve signal efficiency and background rejection.

The FTK algorithm consists of two sequential steps. In the first step pattern recognition is carried out by a dedicated device called the Associative Memory (AM) [6] which finds coarse-resolution track candidates named roads. In the second step, another processor (the Track Fitter) receives these matching patterns and fits the full-resolution hits inside the road to determine the track parameters. Only the tracks passing a χ^2 cut are kept.

The geometry of the ATLAS inner detector and the characteristics of the pattern bank used to perform the pattern recognition stage of the FTK algorithm are described in Section 2. The concept of variable resolution applied to pattern matching is treated in Section 3, while the results of simulations using this new pattern bank feature are discussed in Section 4.

2. The pattern bank for the ATLAS inner detector

The AM is a massively parallel system that simultaneously compares the incoming data (the event hits) with all the stored precalculated low resolution track patterns and returns the addresses of the matching patterns.

A critical figure of merit for the AM-based track reconstruction system is the number of patterns that can be stored in the pattern bank. The FTK processor proposed for the ATLAS experiment is going to operate at a much higher luminosity ($> 10^{34}$ cm^{-2}s^{-1}) than SVT, and this will increase the complexity of the events. A very large bank is necessary to obtain track efficiency greater than 95% over the entire 12-layer[1] inner detector with a small enough pattern width to provide the high rejection of fake tracks needed so the available track fitting processing time is not exceeded.

[1] 8 layers are used in the AM pattern matching. All 12 layers, 4 pixel layers and 8 silicon strip (SCT) layers, are used in track fitting.

The inner detector consists of layers at different radial distance from the beam axis. Each layer is in turn divided into bins of equal size, each one a rectangle in $\phi - z$ space for the central or barrel region and $\phi - r$ space for the endcaps.[2] A pattern is a sequence of Super Strips (SS), one per layer, where the Super Strip is the logical OR of several adjacent bins. Each real track generates a hit pattern. The collection of all these patterns defines both the space of the tracks we are looking for and how they appear in the detector; this collection is the pattern bank.

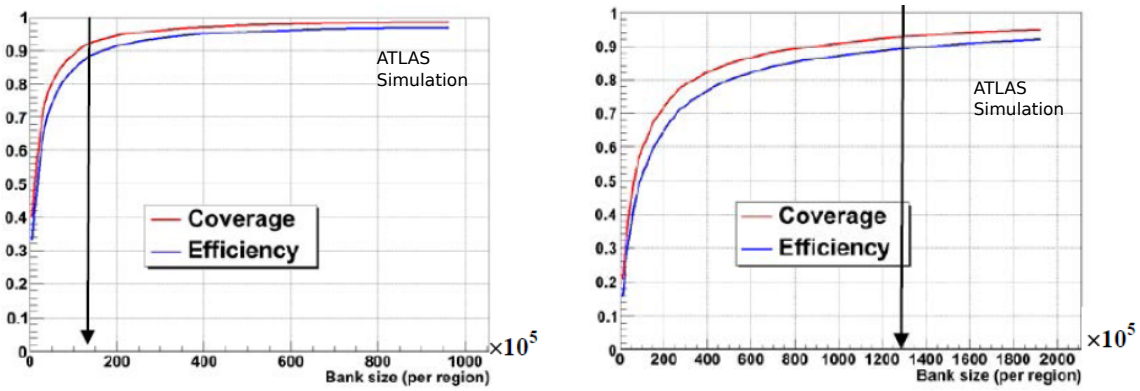
As already mentioned, the pattern recognition step finds the roads that will be sent to the track fitting stage. Here the critical parameter that must be optimized is the road width. If the roads are too wide, the load on the Track Fitter can become excessive due to the large number of uncorrelated hits within a road (fake roads). If the roads are too narrow, the needed size of the AM becomes too large and hence the cost becomes excessive. Therefore the trade off that we have to face is between the number of patterns that can be stored in the bank and the number of fits that the Track Fitter has to perform.

Therefore the quality of the AM bank can be characterized by two variables: the coverage and the fake road rate. The coverage is the fraction of tracks that are reconstructible based only on the detector geometry; it describes the geometric efficiency of the bank. This is in contrast to track efficiency, which includes algorithmic effects such as the χ^2 cut.

We generate tracks in the whole detector and store new patterns corresponding to the generated tracks until the bank reaches the desired coverage. In principle, the pattern bank may contain all possible tracks that go through the detector (a 100% efficient bank). Actually, when effects such as detector resolution smearing, multiple scattering, etc. are considered, a huge number of extremely improbable patterns arises that increase the pattern bank size significantly. For this reason, we use a slightly inefficient bank and we store new patterns until the bank reaches this coverage. To maximize the efficiency of the pattern bank for a given size, we order the pattern list by the number of training tracks that match a pattern. This number defines the coverage of the single pattern. This procedure automatically ensure that "high-coverage" patterns are stored while "low-coverage" patterns are left out.

Figure 1 shows typical curves of track coverage and track efficiency versus bank size with different road resolution. The coverage is determined by the bank only, while the track efficiency includes the contributions of all the algorithms used in FTK and the track fitting. In both cases, there is an initial rapid rise as the bank is filled with high-coverage patterns, patterns that match tracks with a high probability, and then the curves rise slowly as low-coverage patterns, less probable ones, are added to the bank. Although the efficiency for real tracks grows slowly in this region, the number of fake matched roads increases nearly linearly with the bank size. Thus the number of patterns stored in the AM bank has to satisfy both the need for a high efficiency and the need to limit the rate of fake roads. Figure 1a shows that in the case of low resolution roads, we need ~12 million patterns to reach coverage and efficiency of 90%. Therefore the pattern bank is about ten times smaller than the high resolution pattern bank where ~120 million patterns are needed to reach the same coverage and efficiency (Figure 1b). Thus reducing the pattern bank width by a

[2]The ATLAS reference system is a cartesian right-handed coordinate system, with the nominal collision point at the origin. The azimuthal angle is measured around the beam axis, and the polar angle θ is measured with respect to the beam line (z-axis). The pseudorapidity is defined as $|\eta| = -\ln(\tan(\theta/2))$ and p_T is the track momentum transverse to the beam direction.



(a) Low resolution

(b) High resolution

Figure 1: The coverage and the efficiency of the AM bank versus bank size per region. (a) The SS sizes are: 22 pixels in $r-\phi$, 36 pixels in z and 16 strips in SCT. (b) The pattern width is reduced by a factor of 2 along the ϕ direction, so the SS sizes are: 11 pixels in $r-\phi$, 36 pixels in z and 8 strips in SCT. The arrows show the chosen operating point. A region corresponds to 45 degrees in ϕ, one eighth of the detector [8].

factor of 2 in the ϕ direction requires a bank a factor of 10 larger to provide a similar efficiency. Unfortunately, the price of the first solution is that the number of fake matched patterns is about ten times bigger due to the large number of uncorrelated hits within a larger road. The challenge is therefore to obtain a solution that allows a small AM system and, at the same time, a low fake road rate to reduce the Track Fitter workload.

3. Variable Resolution Patterns

As discussed in the previous section it is important to find a compromise between the number of patterns stored in the AM bank and the number of fits to be executed by the Track Fitter. The solution that is proposed is based on the use of "variable resolution patterns". We include in the AM chip the variable resolution feature, that is the ability to employ fine pattern resolution only when necessary. In this way the shape of each pattern can be optimized to improve the acceptance for valid tracks with maximum fake rejection. Therefore this feature allows the use of a small AM bank, while profiting from the positive effects of high spatial resolution in pattern matching.

Figure 2 shows a diagram that illustrates how this solution works. The drawing on the left shows an example of three typical tracks crossing seven detector layers. The number of bins in each layer indicates the SS resolution (1 bin for coarser resolution and 2 bins for finer resolution). The tracks do not cross the white region inside the pattern, so this subregion of the pattern is not necessary, in fact it increases the probability of finding uncorrelated noise hits. The fixed resolution approach provides two ways to store the patterns in the AM bank (shown in the center of Figure 2). The first possibility (Figure 2, center top) is to use a single low resolution pattern stored in just one AM location. This fixed resolution pattern takes up little AM space, but its large Super Strips offer large acceptance for fakes (in the white bins not touched by any real track) The second

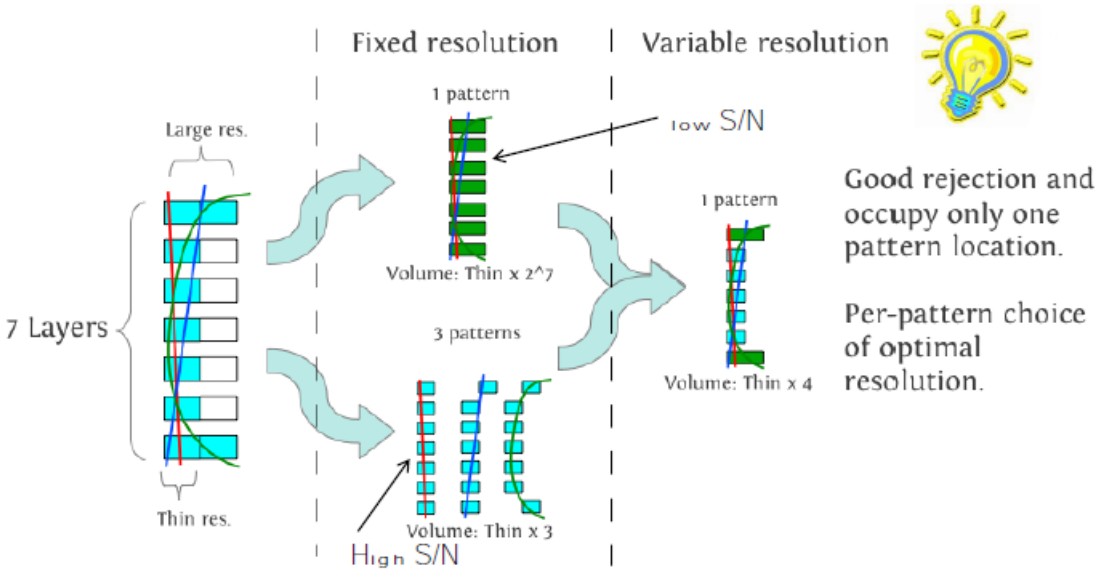


Figure 2: Diagram illustrating the variable resolution patterns.

possibility (Figure 2, center bottom) is to use three high resolution (thin) patterns. Of course the hit combinations in the fitting stage are significantly reduced; in fact the three patterns each contain one of the three tracks crossing the detector layers, but the price is the use of three AM locations. The best compromise is to use a single variable resolution pattern (Figure 2, right) minimizing both the use of AM locations and the fake rate. The size of the patterns changes layer by layer and pattern by pattern. We divide the Super Strips in two: if both halves are touched by real tracks (like the first and last layers in the example), the layer is used at low resolution applying the variable resolution feature; if, on the other hand, just one of the halves is compatible with a real track (the other layers in the figure), the layer is used at full resolution, reducing by a factor two the area that the SS offers to uncorrelated hits from other tracks. As a result we have, at the same time, low pattern bank size, low fake rate and high efficiency.

Figure 3 shows an advanced application of the variable resolution feature using more than one bit. On the left of Figure 3, the no-variable-resolution case, it is shown that three patterns are needed to accept the three tracks crossing the detector layers (the blue, the green and the pink patterns). The middle shows the 1-bit variable resolution case in which one pattern is enough to accept all the tracks. On the right there is an advanced application of this feature, 3-bit variable resolution. We need just one pattern like the previous configuration, but fewer random hits will be accepted because a SS can be even thinner thanks to the 3 bits. This solution increases the rejection of fake roads and reduces the fitting combinatorics produced by uncorrelated hits inside larger roads.

4. Simulation results

As discussed above, one of the main goals in the implementation of the AM pattern bank is

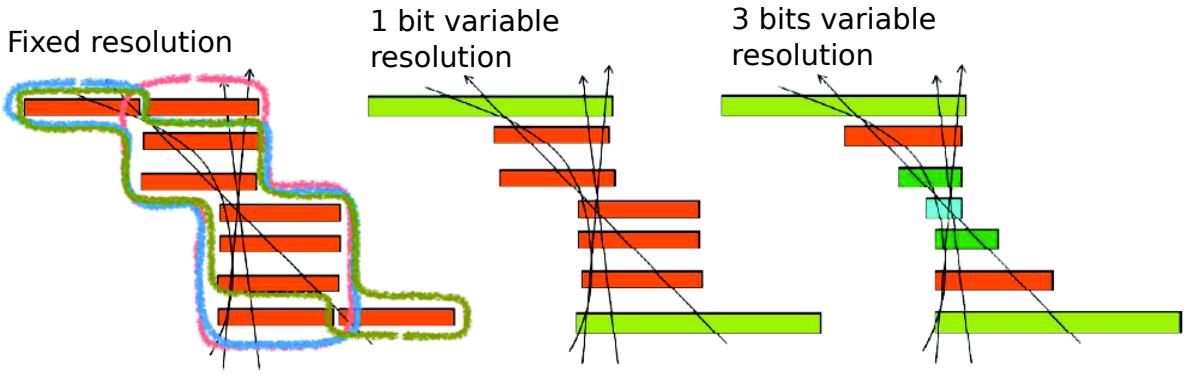


Figure 3: Diagram illustrating the use of many bits in the implementation of the variable resolution patterns: (left) fixed resolution patterns, (center) 1-bit variable resolution pattern, (right) 3-bit variable resolution pattern.

to maximize the bank efficiency while reducing the AM bank size and the Track Fitter workload. This goal can be reached by optimizing the use of the variable resolution patterns as discussed in this section.

The simulation program for FTK (FTKSim) processes complete ATLAS events and creates the same list of tracks that will be produced by the FTK hardware. The program allows us to evaluate the crucial parameters needed for the hardware design, such as pattern bank size, number of roads, roads size and number of fits.

In order to sustain the level-1 trigger rate, it is necessary to organize FTK as a set of independent engines, each working on a different region of the silicon tracker. Therefore the detector is divided into 64 towers corresponding to 16 intervals in ϕ and 4 intervals in η. Each tower uses two AM boards. Each AM board has the following hardware limits [7]:

- Number of AM patterns < 8M[3]
- Number of roads per event $< 8 \times 10^3$
- Number of fits per event $< 40 \times 10^3$

We have concentrated the simulation studies on two scenarios that correspond to the expected LHC conditions in 2015 and 2019 respectively. In the 2015, assuming 46 pile-up events per trigger, FTK will be implemented using 16 boards handling 32 detector towers. In the 2019, assuming 69 pile-up events per trigger, FTK will be implemented using 128 boards managing 64 detector towers. The constraints that each tower must fulfill in the two scenarios are listed in Table 1.

In the simulation we have studied different configurations of the variable resolution patterns. We start from one high resolution set of patterns, with a fixed SS size, and we try several different AM bank configurations. The starting point (the narrowest road width) is a SS of size 15x36x16, where 15x36 is the number of pixels clustered in an SS ($\phi \times z$) and 16 is the number of strips

[3] $M = 1024 \times 1024$

	2015	2019
<#AM patterns/tower>	$4M$	$16M$
<#Roads/event>	4×10^3	16×10^3
<#Fits/event>	20×10^3	80×10^3

Table 1: Constraints for each tower in the two scenarios.

clustered in an SS (ϕ). Various configurations having different resolution have been studied and classified according to the track efficiency, number of matched roads and number of candidate tracks at the input of the Track Fitter. The track efficiency has been evaluated in a single muon dataset while the number of matched roads and the number of candidate tracks have been obtained using simulated events with 69 pile-up interactions. For example, if we apply the 1-bit variable resolution feature on both pixel and SCT layers, the maximum SS size is increased by a factor of two, from 15x36x16 to 30x72x32.

Option	Maximum Road Width	#AM $\times 10^6$	Efficiency(%) R=64	Roads/evt $\times 10^3$	Fits/evt $\times 10^3$
1.0	$(30 \times 72)_{pix} \times 32_{sct}$	18	91.2	7.1	56
1.1	$(30 \times 72)_{pix} \times 32_{sct}$	16.8	91.2	6.9	55
1.2	$(30 \times 72)_{pix} \times 32_{sct}$	15	91.0	6.2	50
2.0	$(30 \times 144)_{pix} \times 32_{sct}$	8	92.0	5	90
3.0	$(30 \times 72)_{pix} \times 64_{sct}$	8	93.0	9	154

Table 2: Results in endcap towers for different road widths. The maximum road width is obtained applying the variable resolution feature to the 15x36x16 base AM bank configuration. #AM patterns, #Roads and #Fits are reported for one tower. #Roads and #Fits are evaluated with 2019 conditions (see text).

Option	Maximum Road Width	#AM $\times 10^6$	Efficiency(%) R=64	Roads/evt $\times 10^3$	Fits/evt $\times 10^3$
1.0	$(30 \times 72)_{pix} \times 32_{sct}$	21	94.8	3.9	33
1.1	$(30 \times 72)_{pix} \times 32_{sct}$	18	94.1	3.4	28
1.2	$(30 \times 72)_{pix} \times 32_{sct}$	16.8	93.3	3.2	26
2.0	$(30 \times 144)_{pix} \times 32_{sct}$	8	95.0	4	60
3.0	$(30 \times 72)_{pix} \times 64_{sct}$	8	96.0	6	98

Table 3: Results in barrel towers for different road widths. The maximum road width is obtained applying the variable resolution feature to the 15x36x16 base AM bank configuration. #AM patterns, #Roads and #Fits are reported for one tower. #Roads and #Fits are evaluated with 2019 conditions (see text).

Table 2 and 3 summarize the most interesting results obtained from the FTK simulation for the endcap and the barrel regions, starting from the 15x36x16 AM bank configuration and applying

the variable resolution feature with different configurations. The options are ordered by decreasing number of patterns. The maximum allowed is 16.8M, the number of memory locations on the two Associative Memory boards in a tower. Thus options 1.0 in the endcap and 1.0 and 1.1 in the barrel cannot be used. Option 1 uses 1-bit variable resolution in both pixel and SCT layers. Option 2 applies 1-bit variable resolution in the pixels in the ϕ direction and in the strips in SCT layers and 2-bit variable resolution in the pixels in the z direction. In the latter case the SS size can be as large as 30 pixels in the ϕ direction, 144 pixels in the z direction and 32 strips in SCT layers. Option 3 uses 1-bit variable resolution in pixel layers and 2-bit variable resolution in SCT layers. The first three lines in Table 2 show that for a given variable resolution configuration, reducing the number of patterns reduces the number of roads and fits, while the efficiency is just minimally reduced. We can see in the fourth line of the table the power of variable resolution. By applying just one more variable resolution bit in the pixels, we get increased efficiency and reduced number of roads with half the bank size.

5. Conclusions

Variable resolution pattern matching allows the patterns to change in shape and matching volume, reducing the pattern width only where needed. It increases the rejection of fake roads, reduces the number of roads out of the AM chip, and significantly reduces the number of patterns in the AM chip. Therefore variable resolution pattern matching is an innovative idea that makes it possible to reduce the cost, size and complexity of FTK. In fact this feature allows setting the architecture parameters so that all hardware constraints are satisfied.

References

[1] ATLAS Collaboration, *The ATLAS Experiment at the CERN Large Hadron Collider*, *JINST*, **vol.3**, pp.437, 2008

[2] W. Smith, *Triggering at LHC Experiments*, *Nucl. Instr. and Meth. A*, **vol.478**, pp.62-67, 2002

[3] A. Annovi et al., *Hadron Collider Triggers with High-Quality Tracking at Very High Event Rates*, *IEEE Trans. Nucl. Sci.*, **vol.51**, pp.391, 2004

[4] J. Adelman, *The Silicon Vertex Trigger upgrade at CDF*, *Nucl. Instr. and Meth. in Physics Research A*, **vol.572**, Issue 1, pp.361-364, 2007

[5] J. Adelman, *Real time secondary vertexing at CDF*, *Nucl. Instr. and Meth. in Physics Research A*, **vol.569**, pp.111-114, 2006

[6] M. Dell'Orso, L. Ristori, *VLSI Structures for Track Finding*, *Nucl. Instr. and Meth. A*, **vol.278**, pp.436, 1989

[7] ATLAS Collaboration, *ATLAS Fast Tracker Technical Design Report*, URL: `https://cds.cern.ch/record/1552352/files/ATL-COM-DAQ-2013-041.pdf`

[8] A. Annovi et al., *A New Variable-Resolution Associative Memory for High Energy Physics*, *Advancements in Nuclear Instrumentation Measurement Methods and their Applications (ANIMMA), 2011 2nd International Conference on*, pp.1-6, 2011