FTK AMchip05: an Associative Memory Chip Prototype for Track Reconstruction at Hadron Collider Experiments

A. Annovi  M. Beretta  R. Beccherle  N. Biezus  S.Citraro
F. Crescioli ¹  P. Giannetti  M. A. Mirzaei  V. Liberali  Y. Piadyk  E. Rossi  A. Stabile

¹(LPNHE - IN2P3 - CNRS)
seconded to CAEN SpA

EPS-HEP 2015 - 22-29/07/2015 - Vienna
Outline

▶ Introduction to the FTK tracking algorithm
▶ Associative Memory function and architecture
▶ AMchip05
   ▶ Features
   ▶ Performances
   ▶ Role in FTK
▶ Pattern bank optimization and variable resolution
▶ Toward HL-LHC
▶ Conclusions
FTK AMchip05: an Associative Memory Chip Prototype for Track Reconstruction at Hadron Collider Experiments

The ATLAS experiment and its TDAQ

**Tracking**

- **Level-1**
  - $< 2.5 \mu s$

- **FTK**
  - $\sim 100 \mu s$

- **High Level Trigger**
  - $\sim 200 \text{ ms}$

- **avg 1 kHz**

- **40 MHz**

See Patrick Czodrowski’s talk: *The ATLAS Trigger System: Ready for Run 2*
FTK AMchip05: an Associative Memory Chip Prototype for Track Reconstruction at Hadron Collider Experiments

The ATLAS experiment and its TDAQ

FTK is a big system:
- 8 full 9U VME crates
- 5 ATCA crates

Many different boards:
- 2 types 9U VME boards: Associative Memory Board (AMB), Second Stage Board (SSB)
- 9U/4 mezzanine for Associative Memory chip (AMchip06)
- 9U Auxiliary - AUX board (data organizer, track fitting and fake reduction functions)
- 2 types ATCA boards: Data Formatter (DF), FTK-to-Level-2 Interface Crate board (FLIC)
- FMC mezzanine for clustering

Complex custom parallel supercomputer

FTK is a big system:
- 8 full 9U VME crates
- 5 ATCA crates

Many different boards:
- 2 types 9U VME boards: Associative Memory Board (AMB), Second Stage Board (SSB)
- 9U/4 mezzanine for Associative Memory chip (AMchip06)
- 9U Auxiliary - AUX board (data organizer, track fitting and fake reduction functions)
- 2 types ATCA boards: Data Formatter (DF), FTK-to-Level-2 Interface Crate board (FLIC)
- FMC mezzanine for clustering

Complex custom parallel supercomputer
FTK is based on a two step algorithm

**Associative memory**

- Preloaded with the pattern bank
- Find patterns in a complex event in real time
- Act as a filter to reduce the complexity of fitting stage

Modular algorithm: scalable and parallelizable. Extremely fast associative memory hardware.

\[
\chi^2 = \sum_{ij} f_i \cdot F_{ij}^{-1} \cdot f_j = \sum_i \tilde{f}_i^2
\]

\[
F_{ij} \approx \sum_{kl} \frac{\partial f_i}{\partial x_k} \frac{\partial f_j}{\partial x_l} M_{kl}
\]

\[
\tilde{f}_i(x) \approx \vec{v}_i \cdot \vec{x} + c_i
\]

\[
p_j(x) \approx \vec{w}_j \cdot \vec{x} + q_j
\]

**Linearized track fit**

- Performed only on patterns found by the AM
- Combinatorial problem reduced: essential in crowded events with high pile-up
- Fast in FPGA
Associative Memory internal structure
FTK AMchip05: an Associative Memory Chip Prototype for Track Reconstruction at Hadron Collider Experiments

Associative Memory working principle

Content Addressable Memory (CAM)
memory + comparator

Layer Match Map

Match Logic counter + pattern match memory

Memory element to record partial match

bus0  bus1  bus2  bus3  bus4  bus5  bus6  bus7

Matched patterns are ready for readout as soon as the event data readout in the FTK AM system is completed.
FTK AMchip05: an Associative Memory Chip Prototype for Track Reconstruction at Hadron Collider Experiments

---

**Associative Memory working principle**

**Horizontal Line:** Each trajectory is stored as a pattern made by 8 CAM (memory+comparator) segments and a majority unit. Each segment is associated to a detector layer.

Matched patterns are ready for readout as soon as the event data readout in the FTK AM system is completed.
FTK AMchip05: an Associative Memory Chip Prototype for Track Reconstruction at Hadron Collider Experiments

---

Associative Memory working principle

**Vertical Line:** Each vertical bus receives data from a detector layer.
The detector hits are distributed to the comparators at the same clock cycle.
Matched patterns are ready for readout as soon as the event data readout in the FTK AM system is completed.
FTK AMchip05: an Associative Memory Chip Prototype for Track Reconstruction at Hadron Collider Experiments

Associative Memory working principle

Only patterns with layer matches above a certain threshold are globally matched: i.e. 0xDF (1101 1111) layer match map → 7 matched layers → pattern match.

The readout of matched pattern is done serially one pattern at each clock cycle with the Fischer Tree (FT).

It is a tree of small logic elements that creates a flexible priority encoder.
AMchip05

Main features:

- Two Associative Memory architecture designs:
  - 2k patterns XORAM arch
  - 1k patterns TOP2_LV arch
- 65 nm TSMC technology
- Full custom CAM-cell
- Standard Cells control logic
- 100 MHz target operating frequency
- 2 Gbps inputs for hits (x8 bus)
- 2.4 Gbps inputs/output for patterns
- Variable resolution
  - 2 to 9 bits configurable as ternary

AMchip05 is the last prototype. AMchip06 - the chip that will be installed in FTK - is functionally identical, but with 128k patterns.
FTK AMchip05: an Associative Memory Chip Prototype for Track Reconstruction at Hadron Collider Experiments

AMchip05 Tests

- High-speed serial data transfer verified on testbench
  - Direct testing: BER < $10^{-12}$ errors/s @ 2 Gbps
  - Stable link up to 3.2 Gbps
- XORAM Pattern bank tested
  - $O(10^{10})$ randomized pattern bank data/hits tests
  - $\sim 98\%$ of chips with zero errors out of $\sim 180$ tested
AMchip05 Integrated on the LAMB

- 16 AMchip05 per LAMB
- Hit data is distributed in parallel to all chips
- Pattern readout is arranged in 4 trees
  - Each AMchip05 can receive patterns from 2 neighbouring chips
  - It merges its own internal patterns with the received patterns to the output
  - Chips can be arranged in a binary-tree structure to increase pattern size without adding pattern inputs in the FPGA

Each FTK processor unit covering $\frac{1}{128}$ of the detector has one Associative Memory Board (AMB), each AMB has 4 LAMB mezzanines → 8 Mpatt with AMchip06
Variable Resolution feature
Regions of track parameter space might have different variance on different layers.

A "wide" family of tracks.

A "narrow" family of tracks.

Variable resolution is possible by encoding don't care bits in the pattern.
Variable resolution regions of track parameter space might have different variance on different layers. A "wide" family of tracks. A "narrow" family of tracks. Not good for these tracks on this layer: many patterns produced.

Good for these tracks on this layer: one pattern produced. Not good for combinatorial background: lots of space for noise/extra hits. Efficient for bank size: one pattern produced. Good for these tracks on these layers.

We need variable resolution for each layer of each pattern. Variable resolution is possible by encoding don't care bits in the pattern.
Variable resolution

Regions of track parameter space might have different variance on different layers.

A “wide” family of tracks.

A “narrow” family of tracks.

Not good for these tracks on this layer: many patterns produced

Good for these tracks on this layer: one pattern produced

Not good for combinatorial background: lots of space for noise/extra hits

Efficient for bank size: one pattern produced

Variable resolution is possible by encoding don’t care bits in the pattern.

We need variable resolution for each layer of each pattern.
We need **variable resolution** for each layer of each pattern.

Good for these tracks on these layers

Variable resolution is possible by encoding *don’t care* bits in the pattern.
A simple 2D layer with 32 pixels $4 \times 8$. Each pixel is identified by a number.
A 5-bit encoded value will match a certain pixel.

<table>
<thead>
<tr>
<th>00000</th>
<th>00001</th>
<th>00010</th>
<th>00011</th>
<th>00100</th>
<th>00101</th>
<th>00110</th>
<th>00111</th>
</tr>
</thead>
<tbody>
<tr>
<td>01000</td>
<td>01001</td>
<td>01010</td>
<td>01011</td>
<td>01100</td>
<td>01101</td>
<td>01110</td>
<td>01111</td>
</tr>
<tr>
<td>10000</td>
<td>10001</td>
<td>10010</td>
<td>10011</td>
<td>10100</td>
<td>10101</td>
<td>10110</td>
<td>10111</td>
</tr>
<tr>
<td>11000</td>
<td>11001</td>
<td>11010</td>
<td>11011</td>
<td>11100</td>
<td>11101</td>
<td>11110</td>
<td>11111</td>
</tr>
</tbody>
</table>
Using *don’t care* on a bit will match two pixels → a **lower resolution** pixel

<p>| | | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>00000</td>
<td>00001</td>
<td>00010</td>
<td>00011</td>
<td>00100</td>
<td>00101</td>
<td>00110</td>
<td>00111</td>
</tr>
<tr>
<td>01000</td>
<td>01001</td>
<td>01010</td>
<td>01011</td>
<td>01100</td>
<td>01101</td>
<td>01110</td>
<td>01111</td>
</tr>
<tr>
<td>10000</td>
<td>10001</td>
<td>10010</td>
<td>10011</td>
<td>10100</td>
<td>10101</td>
<td>10110</td>
<td>10111</td>
</tr>
<tr>
<td>11000</td>
<td>11001</td>
<td>11010</td>
<td>11011</td>
<td>11100</td>
<td>11101</td>
<td>11110</td>
<td>11111</td>
</tr>
</tbody>
</table>
A simple 2D layer with 32 pixels $4 \times 8$. Each pixel is identified by a number. A 5-bit encoded value will match a certain pixel. Using *don’t care* on a bit will match two pixels → a lower resolution pixel. *don’t care* can be placed to select different shapes. More than one *don’t care* bit can be used. Various shapes and resolutions are possible.

<table>
<thead>
<tr>
<th>00000</th>
<th>00001</th>
<th>00010</th>
<th>00011</th>
<th>00100</th>
<th>00101</th>
<th>00110</th>
<th>00111</th>
</tr>
</thead>
<tbody>
<tr>
<td>01000</td>
<td>01001</td>
<td>01010</td>
<td>01011</td>
<td>01100</td>
<td>01101</td>
<td>01110</td>
<td>01111</td>
</tr>
<tr>
<td>10000</td>
<td>10001</td>
<td>10010</td>
<td>10011</td>
<td>10100</td>
<td>10101</td>
<td>10110</td>
<td>10111</td>
</tr>
<tr>
<td>11000</td>
<td>11001</td>
<td>11010</td>
<td>11011</td>
<td>11100</td>
<td>11101</td>
<td>11110</td>
<td>11111</td>
</tr>
</tbody>
</table>

More than one *don’t care bit* can be used.

- **Variable resolution**

<p>| | | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>0000</td>
<td>00001</td>
<td>00010</td>
<td>00011</td>
<td>00100</td>
<td>00101</td>
<td>00110</td>
<td>00111</td>
</tr>
<tr>
<td>01000</td>
<td>01001</td>
<td>01010</td>
<td>01011</td>
<td>01100</td>
<td>01101</td>
<td>01110</td>
<td>01111</td>
</tr>
<tr>
<td>10000</td>
<td>10001</td>
<td>10010</td>
<td>10011</td>
<td>10100</td>
<td>10101</td>
<td>10110</td>
<td>10111</td>
</tr>
<tr>
<td>11000</td>
<td>11001</td>
<td>11010</td>
<td>11011</td>
<td>11100</td>
<td>11101</td>
<td>11110</td>
<td>11111</td>
</tr>
</tbody>
</table>

A simple 2D layer with 32 pixels $4 \times 8$. Each pixel is identified by a number.

A 5-bit encoded value will match a certain pixel. Using *don't care* on a bit will match two pixels → a lower resolution pixel. "don't care" can be placed to select different shapes. More than one *don't care* bit can be used. Various shapes and resolutions are possible.

<table>
<thead>
<tr>
<th>00000</th>
<th>00001</th>
<th>00010</th>
<th>00011</th>
<th>00100</th>
<th>00101</th>
<th>00110</th>
<th>00111</th>
</tr>
</thead>
<tbody>
<tr>
<td>01000</td>
<td>01001</td>
<td>01010</td>
<td>01011</td>
<td>01100</td>
<td>01101</td>
<td>01110</td>
<td>01111</td>
</tr>
<tr>
<td>10000</td>
<td>10001</td>
<td>10010</td>
<td>10011</td>
<td>10100</td>
<td>10101</td>
<td>10110</td>
<td>10111</td>
</tr>
<tr>
<td>11000</td>
<td>11001</td>
<td>11010</td>
<td>11011</td>
<td>11100</td>
<td>11101</td>
<td>11110</td>
<td>11111</td>
</tr>
</tbody>
</table>

FTK AMchip05: an Associative Memory Chip Prototype for Track Reconstruction at Hadron Collider Experiments

Variable resolution

With the **variable resolution** is it possible to have reasonable bank size at target efficiency with optimal background (fake tracks) reduction.
Experiments are studying AM-based solutions for HL-LHC trigger tracking. Prototypes using AMchip05 or AMchip06 are under study.

Develop a common AMchip

- 28 nm
- 2x input bandwidth (200 MHz)
- 4x patterns (512k patt)

The first prototype, AMchip07, is under development.

Level-2 FTK++ for HL-LHC

- Many patterns
- Low input bandwidth

L1 Tracking @ ATLAS for HL-LHC

- Many patterns
- Low input bandwidth

L1 Tracking @ CMS for HL-LHC

- Fewer patterns
- High input bandwidth
- Time multiplexing
Conclusions

- AMchip05 is the latest prototype for FTK before AMchip06 production
- It is identical to the production version except the pattern bank size (3k vs 128k)
- It has been extensively tested in laboratory and on the FTK LAMB prototypes
  - The chip works as expected and within specifications
- The AMchip05/06 architecture has configurable ternary bits
  - It enables the variable resolution of patterns
  - It is possible an optimization of the bank size with a significant gain in FTK efficiency
  - This gain has a direct impact in FTK trigger application to select physics objects
- AMchip05/06 is the last chip for FTK @ LHC
  - The development of the next generation for HL-LHC and other applications has started
  - It will benefit from the experience of AMchip05
- We will move to 28 nm technology
FTK AMchip05: an Associative Memory Chip Prototype for Track Reconstruction at Hadron Collider Experiments

AMchip development has been supported by

ANR-13-BS05-0011-01
ANR-FASTTRACK