Hardware Tracking for the HL-LHC era ATLAS Trigger and the Associative Memory ASIC

HEP2021 - 38th Conference on Recent Developments in High Energy Physics and Cosmology, Thessaloniki, Greece

A. Vgenopoulos¹, K. Kordas¹, D. Sampsonidis¹
(& the rest of the AUTH HTT team)
with extended input from the AM ASIC Team
¹Aristotle University of Thessaloniki
ATLAS TDAQ Phase-II TDR

**Baseline architecture**, based on a single-level hardware trigger with a maximum rate of 1 MHz and 10 μs latency

Divided into 3 main systems
- L0 Trigger (=L1 till Run3)
- Data AcQuistion (DAQ)
- Event Filter (EF) [with HTT]
The Hardware Tracker for the Trigger

- In this scenario, the Event Filter (EF) (known as High-Level Trigger – HLT) consists of: the the Processor Farm and the Hardware-based Tracking for the Trigger (HTT). Takes as input the detector data from events accepted by the preceding hardware trigger at 1 MHz.
- The baseline option, in which the HTT is used as a tracking co-processor in the EF, meets the high trigger rate and throughput requirements at the HL-LHC.
- The HTT Processes events (hits) from the 13 layers the ATLAS Inner Tracker (ITk), with around 80 million channels.
- Pattern Recognition Mezzanine (PRM) will host the Associative Memory (AM) ASICS [for pattern recognition] + FPGA [for track fitting] [scenario a].
- Hits from the rest of the 13 ITk layers will be used but the Track Fitting Mezzanine (TFM) which will perform the 2nd-stage fitting; tracks coming out of the HTT will have almost-offline quality.
- In re-optimization exercises, another option for the pattern matching is considered as well: using FPGAs running the Hough Transform (not covered in this talk) [scenario b].
- The EF selected events output has a rate of 10kHz and is transferred to permanent storage for offline analysis.
- Small overall power usage of all the blades (including required CPUs): <400 kW – can be optimized to <200kW.
- Performance HEP-SPEC0610 (HS06) – with the baseline EF Processing Units ~ 0.37MHS06 | see Hepix benchmarking.
**Baseline HTT diagram**

- An HTT unit is an ATCA self, hosting ATCA "Tracking Processor" (TP) cards. The AM09 ASICs will be mounted on the PRMs [1st stage tracking] which will be mounted on the TPs.

- Each HTT unit will get the data and the EF_requests from the HTTIF FELIX PC; the HTT tracks will be returned to the EF via the HTTIFs.
HTT baseline – the boards

- **PRM** will mount two FPGAs: Stratix 10 MX, MAX 10. The PRM demonstrator mounts the Intel Stratix 10 MX FPGA
- Hosts up to 20 AM chips that provide Track Finding via pattern matching
- 1st stage track fitting from the 8 Itk layers with custom FW on the FPGA

HTTIF: FELIX PCs (2 per HTT Unit) with dedicated HTTIF firmware – *not shown here*

Tracking Processor (TP): an ATCA board which carries two FMCs, a PRM and/or a TFM:
- on/off board I/O
- clustering pixel data
- time alignment of input data
- interface to crate
- monitoring internal FW dataflow

TFM (SSTP): provides 2nd stage track fitting - Extrapolates from ITk remaining layers

RTM: Optical interface (4 * 100Gbps) to HTTIF
The Algorithm steps performed

- ITk decoding and clustering
- Data distribution (within ATCA shelf)
- Pattern recognition
  - Associative Memory (later in this talk)
  - Or Hough Transform
- First stage fit (8 layers)
- Duplicate removal
- Extrapolation to inner layers and second stage fit (13 layers)
- Final duplicate removal

Alberto Annovi slides
The detector's finite resolution makes it “binned” → finite number of “hit patterns”

Because the detector has a finite resolution ("bin size"), many different tracks generate the same hit pattern. So we have a finite number of patterns and a finite-size pattern-bank.
Training: simulated tracks to find possible patterns

Pattern Bank:
- **Pattern #0:**
  - Patt0: 11 12 14 15
  - Patt1: 06 06 07 07
  - Patt2: 15 17 18 20

Pattern #1:

Pattern #2:

1. Each possible track becomes a “pattern” a series of numbers: one coordinate for each detector layer

2. All patterns are stored in a “pattern bank”
Coarse track finding = pattern matching: does your event contain any of these patterns?

The "event" is a list of hits in each detector layer.

3. Compare the hits in your event with the stored patterns.

Pattern Bank:

Pattern Patt0 11 12 14 15
Pattern Patt1 06 06 07 07
Pattern Patt2 15 17 18 20

The "event"
Compare ALL the hits in each event with ALL the stored patterns.

In this example:

**Pattern Bank:**

<table>
<thead>
<tr>
<th>Patt0</th>
<th>11 12 14 15</th>
</tr>
</thead>
<tbody>
<tr>
<td>Patt1</td>
<td>06 06 07 07</td>
</tr>
<tr>
<td>Patt2</td>
<td>15 17 18 20</td>
</tr>
</tbody>
</table>

4. After all comparisons are done, we have the list of matched patterns in the event = list of “roads” to perform refined tracking after
How to match data to patterns?

How to do the Comparison?
Check each of the 5x3x6x6 = 540 hit combinations to each pattern?
AM chip v.1, 700nm, 128 patterns – 1990

AM v.2 (v.1 on FPGA) - 1998

AM v.4, 65nm, 8000 patterns - 2010-12

AM v.5 (mini-mpw), 65nm, 128 -5k patterns - 2012-13

AM v.3, 180nm, 5000 patterns - 2004

AM v.7, 28nm, 8000 patterns - 2015-17

AM v.6, 65nm, 128k patterns - 2014

AM08, 28nm, 12k paterns – 2021 (NOW)

AM09, 28nm, 384k pat – 2022 (?)
Modes of Operation

- SRAM-Read mode
- BIST mode
- Stand-by/Idle mode
- Write CAM mode
- Compare CAM (the wished pattern matching mode)

AM08/09 components

- RX Decoder
- Cascade TX
- CORES:Enable-core, Design-core
- BIST
- ROAD Decoder
- TX Arbiter
- TX Encoder
- Controller
- CLKMAN
- BG Test
Associative Memory: CAM cells check the matching of each hit independently.

As soon as data are present from each Layer, they are put on the bus, to be seen by all stored words along this bus.
As soon as data are present from each layer, they are put on the bus, to be seen by all stored words along this bus.
Associative Memory: CAM cells check the matching of each hit independently

As soon as data are present from each Layer, they are put on the bus, to be seen by all stored words along this bus.

Flags raised if matching in each hit independently
Associative Memory: CAM cells check the matching of each hit independently

As soon as data are present from each Layer, they are put on the bus, to be seen by all stored words along this bus.

AND all flags to get a complete pattern matching.
# AM08/09 Specs

<table>
<thead>
<tr>
<th></th>
<th>AM08</th>
<th>AM09pre (and AM09)</th>
</tr>
</thead>
<tbody>
<tr>
<td>250MHz Internal Clock</td>
<td></td>
<td></td>
</tr>
<tr>
<td>12k patterns (3 x 4k cores)</td>
<td></td>
<td>384k patterns (96 x 4k cores)</td>
</tr>
<tr>
<td></td>
<td><strong>1,68 billion</strong> transistors</td>
<td></td>
</tr>
<tr>
<td>Area</td>
<td>150 mm²</td>
<td></td>
</tr>
<tr>
<td>AM09 about 0.2 zetta (10^{15}) or 200 trillion comparisons per second per chip</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Low-power chip - 1 fJ/comparison/bit with 50% bit change → 2.5W per chip estimate</td>
<td></td>
<td></td>
</tr>
<tr>
<td>KOXORAM+ cells, Quorum Logic in the output</td>
<td></td>
<td></td>
</tr>
<tr>
<td>High performance, high reliability <strong>CMOS</strong> technology at 28 nm (HPC – 10 metal layers + RDL) - TSMC process</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Inputs/Outputs</td>
<td>LVDS18 @ 1Gbps</td>
<td></td>
</tr>
<tr>
<td>Nominal Voltage</td>
<td>1.0V</td>
<td></td>
</tr>
</tbody>
</table>
AM08 Production and Testing

- Production of 100 dies by TSMC, handled by IMEC, received in May: 85 for packaging, 15 for bare die tests
- Ball Grid Array Substrate design signed off
- The Packaging will be completed within summer
- Tests in-house will start soon (next slides) and Industrial tests within the Fall 2021
- Next generation for use in ATLAS, AM09pre – design undergoing – will be submitted in early 2022
  - The ATLAS TDAQ decision will define the next steps for the project
AM08 Test Setup

- Based on an Enclustra AA1 Intel SoC module, which will run the custom Firmware+Linux
- V1 “motherboard” in production very soon
- Two types of Mezzanine cards:
  - Socket Mezzanine – for packaged chips
  - A second Mezz for the bare dies
- *For the bare Die*, the test setup, will be implemented in Heidelberg University
- 15 chips (not packaged), just received at CERN, will be shipped and tested in Heidelberg, within the summer
- Test Patterns(Vectors) produced from behavioral simulation will be used as stimuli (next slides)
Industrial testing is done with ATE machine (possibly an Advantest V930000)

- Behavioral simulations set (we have 36 testbenches) → output directly from Cadence Simvision: CSV or VCD (Value Charge Dump) text files → WGL/STIL → ATE specific format
Validation for AM08/09 - II

- VCD (and EVCD) are a standard (defined with Verilog in IEEE 1364-2001), offer interoperability between EDA software and other tools like TV writer/producer. We export these from the simulations.


- Used by the specialized company which offers the testing services - both standard tests and characterization.

- Extended evaluation of pattern production software, from two US companies:
  - ✔ TESSI Solstice-TDS
  - ✔ Test Spectrum VectorPro

- Validation for AM08
  - ➔ Based on SVA (Assertions)
  - ➔ Mostly targeted in error prone processes like the FSMs of the chip
  - ➔ Written in a design-independent way, linked to the design files
  - ➔ Relatively new task – steep effort and time resources required
Backup slides
AM08/09 components detailed description

- **RX Decoder**: is the first stage to decode the serialized data in input to the AM08. It receives a 33-bit signal (control: BUS_IN[33], data: BUS_IN[32:0]). Data is registered in DDR mode with CLKF.

- **Cascade TX**: This block transmits the incoming bus data (BUS_IN) as-is to a neighboring AM08 via BUS_OUT.

- **CORES**: Enable-core (32 x 12k patterns), Design-core (4k patterns)

- **BIST**: can check if there are stuck-one or zero CAM cells.

- **ROAD Decoder**: propagates the ROADs from the ROAD_In towards the TX Arbiter

- **TX Arbiter**: merges multiple data sources into a single stream for passing to the TX Encoder

- **TX Encoder**: transfers data reliably to the FPGA in 10 LVDS pairs

- **Controller**: Includes the status and the control registers that would be accessed by either the input interface (output of RX Decoder or the SPI), selected by a MUX

- **CLKMAN**: Clock Manager – the CLKF is the main clock

- **BG Test**: Two loopbacks (RX and TX) have been designed. Conventional. With Differential IN/OUT
AM08 Test Setup II

Bare Die Sockets – 3D rendering, shown below