Highly Parallelized Pattern Matching Execution for Event Real-Time Reconstruction

S. Citraro, N. Biesuz, P. Giannetti, P. Luciano, H. Nasimi, M. Piendibene, C.-L. Sotiropoulou, *Member, IEEE* and G. Volpi

***Abstract* — A high performance "pattern matching" implementation based on the Associative Memory (AM) system is presented. It is designed to solve the real-time track finding problem for particles produced in proton-proton collisions, in high energy physics experiments at hadron colliders. The processing time of the track finding problem increases rapidly with the detector occupancy when CPU-based algorithms are used, due to the limited computing and input-output power of hardware available on the market. The presented AM system is scalable. The elementary units are boards, which can be packed into crates to increase the computing power as needed. The board is implemented as an array of 64 AM chips. Each AM chip is a custom VLSI device, based on Content Address Memory (CAM) technology. All the chips are identical and each one of them stores a preset number of "patterns", compared in parallel to the incoming data while the detector is being read out. The system has a very powerful input-output capability: communication between chips is provided by a powerful network of more than 750 2 Gbit/s serial links per board. These features allow the AM system to perform the pattern matching while data are loaded. A complete AM-based processor is much smaller and less power consuming than its CPU equivalent: 8 crates of electronics are able to perform a task that would need a farm of thousands of commercial CPUs. A single board is in fact able to execute ~ 6 peta comparisons/s, with a peak power consumption below 250 W, uniformly distributed on the large area of the board.**

***Index Terms* — Pattern matching, Image processing, Parallel processing, Trigger circuits, Field programmable gate arrays, Application specific integrated circuits**

I. Introduction

IN recent years there has been substantial development in image detector technology that has led to a great increase in both resolution and produced data. These detectors target several different application fields from everyday applications (such as smart phone cameras) to complex and demanding applications (high energy physics, medical imaging, security service and others). Such applications demand an effective method for data reduction with minimal loss of information. Pattern matching is a common algorithm used for such processes.

Pattern matching algorithms look for a given sequence of tokens (data) that constitutes a predefined pattern. Pattern matching is not limited to image processing, but it is extended to other fields, such as data servers (e.g. search engines) and all types of data processing that require identification of specific data sequences (e.g. DNA identification, etc.). The complete system we propose is very flexible since it can integrate a variable number of processing boards as required by a specific application. Each board of the presented system can execute more than 67 million 16-bit word comparisons every 10 ns. All clock cycle data are distributed in parallel to the large number of patterns with fan-outs of 1:8 million. These processing boards work in parallel to achieve higher performances. Such high performance requirements can be found in high energy physics experiments at hadron colliders. These experiments search for extremely rare processes hidden in a much larger data sample.

Our system was developed for the general purpose experiments, ATLAS [1] and CMS [2] at the Large Hadron Collider (LHC) at CERN [3], but it is flexible enough to be adapted to the aforementioned generic image processing applications.

II. Pattern Matching at Hadron Colliders

A. Introduction to the tracking problem at LHC

The LHC is the most powerful particle accelerator that exists today. It produces rare particles accelerating high density bunches of protons. Bunches running in opposite directions in the accelerator cross each other at the center of each experiment built to record the proton-proton (p-p) collisions, with a frequency of 40MHz. At each bunch crossing, many p-p collisions are produced. This is also referred to as an "event". Most of the events contain only not interesting collisions, mainly soft collisions called pile-up. The amount of pile-up depends on luminosity, an accelerator

The AMBSLP prototype boards received support from Istituto Nazionale di Fisica Nucleare and the European Commission FP7 People grant FTK 324318 FP7-PEOPLE-2012-IAPP.

S. Citraro, N. Biesuz, H. Nasimi, M. Piendibene, C.-L. Sotiropoulou and G. Volpi are with the Department of Physics "Enrico Fermi" of the University of Pisa and INFN Pisa Section, Polo Fibonacci Largo B. Pontecorvo, 3, 56127, Pisa, Italy (email: saverio.citraro@for.unipi.it, nicolo.vladi.biesuz@cern.ch, hknasimi@gmail.com, marco.piendibene @pi.infn.it, c.sotiropoulou@cern.ch, guido.volpi@cern.ch).

P. Giannetti is with INFN Pisa Section, Polo Fibonacci Largo B. Pontecorvo, 3, 56127, Pisa, Italy (email: paola.giannetti@pi.infn.it).

P. Luciano is with the University of Cassino and Lazio Meridionale, Gaetano di Biasio, 43, Cassino, 03043, Italy and INFN Pisa Section (email: pierluigiluciano@pi.infn.it).

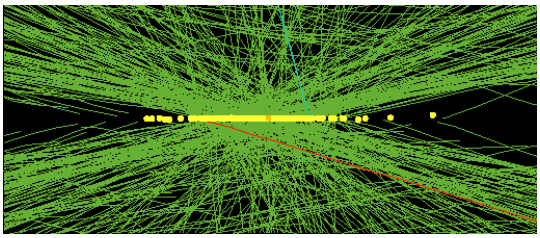

Figure 1. An event produced at LHC. Image credit: Andre Holzner (http://cms.web.cern.ch/news/reconstructing-multitude-particle-tracks-within-cms)

parameter that determines the probability to produce events: in order to discover rarer particles, more luminosity is required; As a consequence, the larger is the luminosity, the larger is the pile-up. The pile-up will reach average values of 80 contemporary collisions when the LHC upgrade will be completed. Each collision is characterized by its own set of particles originating from the point where the collision happened, the vertex. The particles leave traces in the detectors. The experiment is made of millions of detecting elements to record the collisions with high precision. An example of event recorded by the CMS [2] experiment in the past runs shows many p-p collisions in Figure 1. The figure shows how different vertices (yellow points) are clearly detected by the experiment. Green lines are the tracks of particles that we aim to reconstruct.

The data flow is so massive (1.7MB/25ns ~ 80TB/s [4]) that only a very small fraction of the events can be permanently stored for analysis. A drastic real-time data reduction must be obtained, with minimal loss of useful information. The trigger performs this task using fast logic that selects the events with the greatest scientific potential.

A multi-level trigger [5] is an effective solution for an otherwise impossible problem. Triggers for experiments built for hadron colliders like LHC had 3 trigger levels in the past. The level-1 (L1) has strong latency constraint (few microseconds) and is usually based on custom processors. It reduces the rate of events from the machine event production (40 MHz at the LHC) down to tens of kHz. For the upgraded LHC, the level-1 trigger will reduce the event rate to 100 kHz. L2 and L3 have been unified in a single high-level trigger (HLT) at CMS and ATLAS. They perform the event reconstruction using a single large dedicated CPU farm.

Tracking devices, and in particular silicon detectors that are becoming the predominant tracking technology, play an essential role in the identification of interesting events. In fact, they provide very detailed information for charged particles and they can separate most of the different particle trajectories in the overlapping collisions recorded in the same image (see Figure 1). However, these detectors contain hundreds of millions of channels, so they require huge computing power for full track reconstruction. They make the problem of complete tracking a formidable challenge even for large computing farms [6]. Therefore, complete high-quality tracking for real-time event selection at very high rates has been considered impossible in LHC experiments at the time they were built. Real-time tracking was planned for limited detector regions or on a small subset of events, previously selected using other detectors [5].

B. Our solution for the real-time tracking problem compared to other approaches

We overcome the problem by providing real-time tracking using a massively parallel high performance system. The ATLAS experiment has recently approved this technology for a HLT trigger upgrade [7].

Our goal today is the real-time reconstruction of all the tracks above a minimum energy for all the events selected by the L1 trigger, which correspond to an event rate of 100 kHz. Given the complexity of the events, the input data rate to the system is expected to be ~200 GB/s at the maximum LHC luminosity. The maximum supported data rate is 400 GB/s

Our strategy is based on the optimal mapping of a complex algorithm onto different technologies. The target is to get the best results by combining the high performance of VLSI dedicated hardware with the distinctive flexibility of modern programmable logic

A key role in the architecture is played by high-performance field programmable gate arrays (FPGAs), while most of the computing power is provided by full-custom ASICs, the Associative Memory (AM) chips [8].

The AM chip is a pattern matching device with characteristics similar to a Content-Addressable Memory (CAM) [9]. However, the design of the AM is conceptually different to that of a CAM. In the AM each pattern is not stored in a single memory location, like in the commercial CAM, but it consists of 8 independent 16-bit memory locations, in which the coordinate locations of the position where the particle hits the silicon detector (hit) [10] can be stored. The innovative characteristic of the AM is that each one of these 8 words has a comparator and a match flip-flop to compare continuously the stored data with its own input data stream. Data are sent on 8 parallel buses, one for each word of the pattern. All words in the AM make independent and simultaneous comparisons with the data serially presented on its own bus. Every time a match is found, the match flip-flop is set and remains set until the end of the event processing, when a reset signal is propagated. A pattern matches when a predefined number of the flip-flops is set (user defined threshold). All the matched patterns are read out. An extensive description of the AM and its operation can be found in [8].

FPGAs configure and control the AMs and their I/O, providing the flexible computing power to process the selected patterns. Distributed debugging and monitoring tools suited for a pipelined, highly parallelized structure and a high degree of configurability can cope with a variety of applications. Both AMs and CAMs have been used for real-time tracking in high energy physics. Commercial CAMs were used in the H1 experiment [11], where each bit of a CAM word corresponded to a detector channel. Due to the growth of the number of detector channels (~100 million), this approach is no longer

feasible. Furthermore, additional hardware was needed to reformat the incoming data before sending them to the CAM. This problem is solved in the AM. The first AM device, produced for the CDF [12] experiment at the Tevatron collider of Fermilab, was applied without problems to the high density detector case described above. In the AM, each word of the pattern refers to a different detector layer and represents the address of a possible hit channel on that layer, as received from the detector front end. Layer matches could happen at different times during the processing of an event, since they are stored in flip-flops and continuously checked for coincidence with the other layers to produce a track match.

The presented AM system is an evolution of the CDF design [13]. The requirements for the LHC application are more demanding than those for CDF: a bigger silicon detector with even more channels requires more patterns and higher trigger rate requires higher operating frequency, while the total power consumption must be contained.

The new system increases the data flow rate with respect to the system used at CDF by exploiting the parallel readout of the detector layers on 8 buses. The new system clock is 4 times faster and the design implements specific techniques to reduce the power consumption ([14], [15]).

Different technologies have been compared in the past looking for the best pattern matching implementation. The need to maximize available pattern density on a chip initially forced to adopt a full-custom VLSI approach, which implied a big development effort and a difficult upgrade path. In order to consider the best trade-off between pattern density and ease of design (and eventually re-design), an FPGA-based and a standard-cell-based design were performed. Despite recent impressive FPGA progress, these devices were and are still not adequate for our application, while a standard-cell based design was found to bring substantial advantages, as discussed in details in the paper [13].

The current generation of AM devices introduces a mixed architecture: full custom blocks for the CAM cells, standard cell logic for everything else, in particular the control logic. This approach has two advantages: (a) the full custom design effort was limited to a small piece of the large memory, a cell that could be replicated many times in a very structured area of the chip, occupying the largest fraction of the die; (b) the control logic implemented with standard cells was easily handled and simulated by the development software. By using this method, the design effort, the degree of reliability and the power consumption could be maintained inside the desired limits. The chosen technology is TSMC 65 nm. The use of full custom CAM cells enabled a higher pattern density, more than what was expected from simple node scaling from 180 nm to 65 nm: 128 k-patterns are placed today in 165 mm^2, while AMchip03 had only 5000 patterns in 100 mm^2. The full custom designed CAM cell of the best AM chip available today has been described in [15].

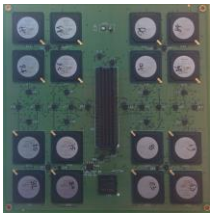

Figure 2. The LAMB assembled with 16 AM chips.

III. IMPLEMENTATION

A. System Segmentation and Scalability

The input bandwidth sets an upper limit either on the event rate or on the size of the detector connected to the processor. In order to sustain very high event rates, it is necessary to organize the system as a set of independent engines (typical input bandwidth of each one 1.6 GB/s), each one working on a different sector of the silicon tracker. Let us imagine dividing the detector into azimuthal sectors. This segmentation generates some inefficiency at sector boundaries that can be removed by allowing a small overlap region between adjacent sectors.

The system setup for the LHC experiments is organized into 9U VME boards (AMBSLPs), which process the tracker data in parallel, working on different sectors of the detector.

Thus, the system is scalable and can grow to provide higher computing power to cope with the detector occupancy increases due to the increment of the LHC luminosity. The AM system that is approved for the current ATLAS detector upgrade stores 1 billion (10^9) AM patterns [7] into 128 AMBSLP boards. We foresee an enlargement of more than a factor 10 for the LHC future upgrade without a significant increase of boards thanks to the miniaturization process due to the technology advancement.

The design of the AMBSLP is a challenging task, due to the following factors: (1) the high pattern density (8 million patterns per board today), which requires a large fraction of the board to be filled by AM chips; (2) the I/O signal congestion at the board level, which requires the use of a huge network of serial links; (3) the power limitation due to the available cooling system: as we can fit up to 20 AMBSLPs in a VME crate, the power should not exceed 250 W per AMBSLP.

B. AM Integration on a mezzanine: power dissipation and serial link communication optimization

The current LHC event complexity requires each 9U-VME board to hold 8 million patterns, equivalent to 64 AM chips. In order to simplify the I/O operations and increase the system modularity, the AM chips are grouped into daughter cards, each one with 16 chips, called Little Associative Memory Boards (LAMB, Figure 2).

Previous versions of the AM chip used parallel buses for I/O ([13]). This design, however, had some serious drawbacks: distributing 16×8 lines to 16 chips led to extreme complexity in the design of the mezzanine. The parallel data distribution

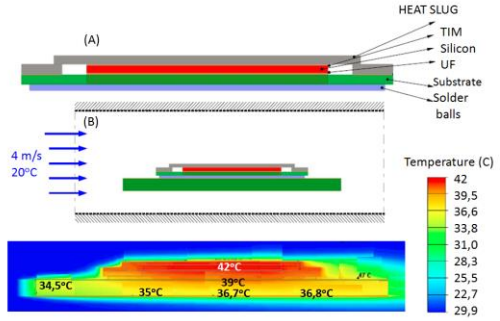

Figure 3. (A) the AM chip package model; (B) the simulation conditions; (C) the simulation results.

gave upper limits on data speed, not compatible with further system upgrades, and no space was available on the board to optimize the routing and decrease cross-talk as well as noise. In addition, several cooling issues appeared, due to the insufficient dissipation capabilities of the used package (TQFP208 [16]).

In order to solve these problems, a switch from parallel buses to high speed serial buses was decided. By using this strategy, we produced free space on the PCB and new opportunities for the layout, routing and power dissipation.

The new package of the AM chip was optimized for high dissipation capability. A Ball Grid Array package with Flip Chip interconnection and a high number of balls (FCBGA 23x23, 529 pins) was chosen. We included a heat slug for high dissipation capability. The very high number of pins and balls included between the die and the substrate, overesized with respect to the electrical need, have been activated in order to have a good thermal connection between the PCB, the substrate and the die. We also optimized the PCB for maximum dissipation, increasing the thickness of the metal layers and covering with a metal surface connected to the GND the areas where the 16 AM chips' GND pins have to be soldered. The board design was exported from the Cadence Allegro PCB Designer to ANSYS computational fluid dynamics (CFD) simulation software that allowed us to predict, with reliability, the impact of fluid flows on our product. We used the ODB++ Intelligent Data Format to export the project from Cadence to ANSYS, providing all the necessary inputs for an accurate temperature study: PCB layers, stack-up and component's information. Together with the use of the bill of material, we could generate the physical model of the boards. Particular attention was dedicated to the AM chip model. In Figure 3 the AM chip model (A), the simulation conditions (B), and the best simulation results (C) are shown. We performed the analysis of the influence of the copper distribution on the PCB. The AM chip detailed component level thermal simulation shows that the temperature of the die is almost the same of the temperature of the top of the package (42 ^{o}C in a uniform air flow of 4 m/s) and just few degrees higher than the temperature on the PCB (39 ^{o}C on the top of the PCB and 36,7 ^{o}C on the bottom side). The power dissipated on each LAMB is really high: 16 AM chips, each one dissipating roughly 2.4 W at 1 V (core) and 0.5 W at 2.5 W (I/O), and 44 fan-out chips for additional 7 W on 2.5 V. However, the power has been on purpose well distributed on the whole surface of the mezzanine, as shown in Figure 2, and the optimization of the AM package and the PCB metal thickness and shape allows to keep under control the temperature efficiently, even in absence of a heatsink.

Most of the 529 pins have been assigned to the 3 power domains and ground. A small number of pins, optimally routed, are used for the serial I/O: 8 input links to receive input data from the detector, one per layer used in the pattern matching, 2 links to receive pattern addresses from other AM chips, and an output to send out the addresses of patterns fired in the chip itself. In total, the AM chip needs 11 serial links. The AM chip serial communication interface is a Silicon Creations' IP. Its use and test is described in reference [15]. The main features of the Silicon Creation serializers/deserializes (SERDES) are: (a) data rate at least 2 Gbit/s to match 16 bit at 100 MHz; (b) 8b/10b encode/decode capabilities; (c) separate serializer and deserializer macro (the AM chip has 10 input buses,, but one output bus for patterns); (d) 32bit I/O buses; (e) driver and receiver circuits compatible with LVDS standard; (f) comma detection and word alignment; (g) BIST capabilities for fast debugging; (h) Low power [15].

We have produced 200 AM chips in a Multi-Project Wafer run, with the final functionality and package but a much smaller die, where only 2048 patterns/chip could fit. We have used those chips to build the first AM system. Sixteen AM chip locations are arranged into 4 quartets. Each quartet has a low jitter oscillator in the center, necessary for the 11 serial links handled by each AM chip. The 100 MHz LVDS clock is distributed to the 4 AM chips by a small fan-out chip, the black square in the middle of the quartet. Below the connector there is a small FPGA and its FLASH Memory, used for AM configuration. A closer view of the top-left corner of the LAMB PCB is in Figure 4.

The few green pads in the BGA package are the signal pads: the serial link and clock LVDS inputs are visible in the bottom side, while the single ended signals are on top of the BGA. All the other pads are used for GND (blue), 1 V (yellow), 2.5 V (pink) and 1.2 V (violet).

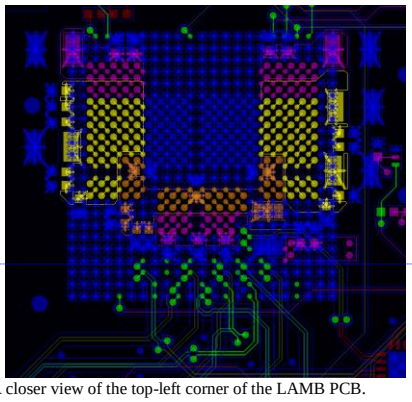

Figure 4. A closer view of the top-left corner of the LAMB PCB.

Comment [D1]: Cu? Non ci sono altre citazioni nel testo: sigla oppure refuso.

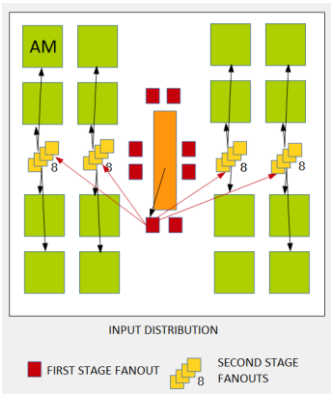

Figure 5. Input data distribution to AM chips.

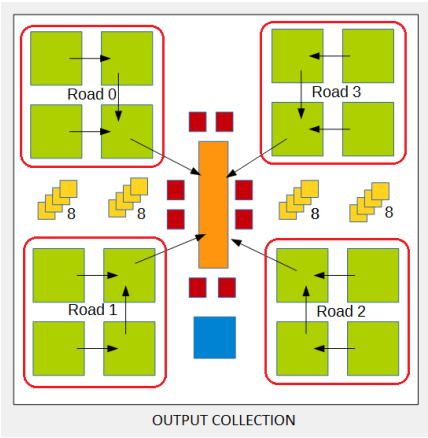

Figure 6. Output data collection from AM chips.

Particular care has been devoted to the PCB routing, especially for the many serial links (~ 200 links), in order to keep the differential impedance fixed at 100 Ω, minimizing the cross-talk. This is a 12 layer PCB where signal and power-GND planes are alternated. The serial links are all routed into internal layers, so that they are isolated between two metal planes. In addition, we studied the best solution to reduce noise and cross-talk between two adjacent lines in such high density PCB layout. Starting from the simulation reported in reference [17], we implemented three possible solutions: (1) Float guard lines; (2) Guard lines closed to GND through 50 ohm resistors; (3) Grounded guard line by vias. The effect of those configurations on the BER demonstrates that the grounded shielding is the best option (see section IV B).

Data must be distributed at 2 Gbit/s on each serial link with a very large fan-out: 8 words from 8 detector layers have to reach in parallel the 8 million of patterns at each clock cycle. The large input fan-out on the LAMB is obtained through 2 levels of serial fan-out chips and a very powerful data distribution tree inside each AM chip. Figure 5 shows the distribution of the input data through 40 1:4 fan-outs. The 8 red ones around the central connector (orange box) replicate each of the 8 incoming buses 4 times to make them available to each subgroup of 4 AM chips organized into vertical sections, as shown by the blue dotted lines.

The second level of fan-outs (16 yellow little squares on the top side and 16 on the bottom side of the PCB, not visible in the figure) replicates again each bus 4 times, one for each single AM device in the subgroup. The placement of chips on the LAMB has been studied and optimized to minimize the crossing of the serial links.

Figure 6 shows how the output words are collected from the 16 AM chips organized in 4 independent quartets. Each AM device has the capability to receive outputs from other two AM chips and merge them internally with patterns that fired in the chip itself. Each quartet has a single output that goes directly to the connector.

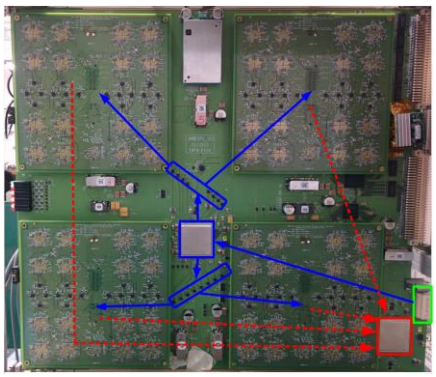

Figure 7. The data traffic in the motherboard.

C. Assembling four LAMBs on the motherboard

A 9U-VME board has been designed to hold 4 LAMBs. Figure 7 shows the motherboard. The LAMB and the motherboard communicate through a high frequency and high pin-count connector placed in the center of the LAMB. A network of high-speed serial links handles the data distribution from the input (the high-density connector in the green box on the bottom-right side of Figure 7, called P3) to the 4 LAMB connectors and back to the P3 connector. Twelve input serial links (in blue) carry the silicon data from the P3 to the LAMBs, and 16 output serial links (4 links from each LAMB represented by a red arrow in the figure) carry the fired patterns from the LAMBs back to the P3. Events are loaded to the board at a maximum rate of 100 kHz corresponding to a maximum input bandwidth of 1.6 GB/s. An even larger number of output words per link can be collected and sent back to the P3. Each board can read out up to 8000 matched patterns per average event, for a maximum output bandwidth of ~ 3.2 GB/s, thanks to 16 parallel output links (4 links per LAMB).

The data traffic is handled by 2 Xilinx Artix-7 XC7A200T

FPGAs, which have 16 Gigabit Transceivers (GTP) [18], each one providing ultra-fast data transmission. The FPGA in the blue box in Figure 7 handles the input data, while the FPGA in the red box near the P3 handles the output data. Two separate Xilinx Spartan-6 FPGAs (XC6SLX16 and XC6SLX45T) implement the data control logic and configuration. The 12 input serial links are merged into the 8 buses received by each AM chip, one bus for each detector layer used for pattern matching.

The AMBSLP motherboard is a PCB VME 9U (366mm x 400mm) and is made of 12 layers. Half of them are used to distribute power mainly to the LAMBs, 50 Watt each layer. The needed power is generated from 48 V using a large number of DC-DC converters from GE Critical Power: a device visible at the AMBSLP top in Figure 7, which generates 33 A at 12 V in order to power a set of smaller devices that generate the needed currents at 1 V, 2.5 V, 1.2 V and 1.8 V. On the other layers, we routed the differential lines following the same design rules adopted for the LAMB, and all the buses used for control and configuration.

D. Data Flow, Event Synchronization and Processing

The AM system is designed to be part of a data driven pipeline where a large number of devices are connected by thousands of links: thousands of dedicated custom chips (AM chips) that perform pattern matching and thousands of FPGAs for all the other functions.

A simple communication protocol is used for data transfers inside the AM system (between FPGAs and AM chips) and with neighboring boards. The data flow through serial links connecting one source to one destination. The protocol is a simple pipeline transfer driven by control words. Control words can be idle words and alignment words. An 8b/10b encoding [15] is used in the serial data stream in order to provide clock recovery, i.e. a 32-bit word is transmitted as 40 bits. The idle word is transmitted when no valid data is available. Alignment words are periodically transmitted between data words. Input words in each processing step of the pipeline are pushed into a de-randomizing FIFO buffer. All the words not identified as control words are pushed into the FIFO (write-enable signal asserted to the FIFO). The FIFO is popped by whatever processor sits in the destination device. To maximize speed, no handshake is implemented on a word-by-word basis. Data can flow at the maximum rate compatible with the link bandwidth, even when transit times are long. A hold signal (HOLD) is used instead as a loose handshake to prevent loss of data when the destination is busy. If the destination processor does not keep up with the incoming data, the FIFO produces an "almost full" signal that is sent back to the source as HOLD signal. The source responds to the HOLD signal by suspending data flow. Using "almost full" instead of Full gives the source sufficient time to stop. The standard clock frequency is 100 MHz for 16-bit words or 50 MHz for 32-bit words, which corresponds to 2 Gbit/s for serial transmission.

The End Event (EE) word, marked by a specific control word, separates data belonging to different events on each transmission link. Each device will assert an EE word in its output stream after it has received an EE word in each input stream and it has no more data to output. The EE word has a special format used to tag the event and to report the parity and any error flags.

The AM system has many independent input streams, and events are subdivided into these streams. Data arriving from different layers of the detector have to be synchronized, since the same event can arrive on different links at different times. The input FIFOs perform this task. Their depth covers fluctuations in the device processing time and arrival time of input data. When the device starts to process an event, words are popped from the input FIFOs for the various input streams. The data is processed and results are sent to the output stream. When the EE word is received on an input stream, no additional data is read from that FIFO until the EE word is received on all the other input streams. The device can issue a HOLD signal if a FIFO becomes "almost full", causing back pressure, but the goal is to have the FIFO deep enough to limit back pressure as much as possible. The EE words from the input streams are checked to make sure they contain the same event tag. Upon detection of different event sequences, a severe error is issued and the system must be resynchronized. Once the event is completely read out from the input FIFOs and the device finishes its processing, the event is closed by sending an EE word to the output with the same event tag as in the input streams.

Each event is processed in two phases: (1) first of all the detector data (hits) have to be loaded inside the AM to be compared with the whole pattern bank and (2), when the event is completely loaded, the matched patterns (roads) have to be read out of the system. The AM architecture is organized in such a way that the two phases can be executed in parallel on contiguous events, since the results of the comparison of one event can be kept inside the chip when a new event is downloaded. The event processing is controlled by two Finite State Machines, one for the hit and the other for road transit: when hits of the N+1 event are sent to the AM chips, the roads of the event N are read out.

E. System Control and configuration

The AM system is fully controlled and monitored by a CPU using the VME standard. The VME slave interface, implemented in a Spartan-6 FPGA, allows writing/reading functions to/from registers, memories and FIFOs, using random access and block transfer modes.

The most important implemented function is the configuration of the AM chips, in particular uploading the patterns that have to be stored in their memory. The AM chips are configured through a JTAG port. The 64 chips are organized into 32 chains of 2 AM chips each. The chains are handled in parallel to minimize configuration time. The VME 32-bit wide data transfer is segmented into 4 bytes, one byte assigned to each LAMB. On each LAMB, 8 JTAG chains are handled by a small Spartan 6 FPGA, an interface between the VME and the AM chips. Pattern uploading time for a single board is ~30 seconds using block transfer.

Another important part of the configuration is the very large number of serial I/O interfaces. The AM chips alone use 640 receivers and 64 transmitters that require proper initialization. These links are also used to download patterns, so they are the first step of the configuration procedure.

TABLE I
MEASUREMENT RESULTS FOR LONG SERIAL LINES WITH/WITHOUT REPEATER

	Eye Height (mV)	Eye Width (ps)	Eye BER
Line w/o Repeater	147	394.3	$8.15 \cdot 10^{-6}$
Line with Repeater	408.2	433.5	$807.5 \cdot 10^{-42}$

TABLE II
MEASUREMENT RESULTS FOR SERIAL WITH DIFFERENT GUARD LINES CONFIGURATIONS

	Eye Height (mV)	Eye Width (ps)	Eye BER
GND Guard	433.3	442.8	$2.95 \cdot 10^{-30}$
w/o Guard	378	442.6	$187.4 \cdot 10^{-18}$
Float Guard	271	423	$2.08 \cdot 10^{-9}$

F. System Monitoring

The AM processes a large quantity of data, a small part of which ends up in the event record. If an error occurs, the proper diagnosis of its source requires access to data at every step in the pipeline. To accomplish this, we have implemented the Spy Buffer system, a circular memory used as a logic state analyzer for input and output of the system. This memory is continuously written with the data processed by the board. The write operation is stopped when a Freeze signal is asserted, in order to preserve the data already written. After Freeze is set, no data can be written into the memory and the content of the memory can be read through VME access for diagnosis of error conditions or for standard monitoring functions. For each Spy Buffer there is a status register that contains a pointer to the first free memory location, an overflow bit that indicates if the memory has been written more than once, and the Freeze bit. Spy Buffers are 4-8k locations deep, allowing the storage of 4-8 average events.

The comparison of the content between a sender's output buffer and a receiver's input buffer makes it possible to check the quality of the data transmission. We check data processing comparing the board's input and output with emulation software. The memories also serve as sources and sinks of test patterns for testing single boards or a small chain of boards, as a standalone system.

IV. RESULTS

A. Event Processing Validation

To test the global functionality of the system, we use a comprehensive test called "Random Test". It generates a random bank and a bunch of random events (random inputs) in order to run it for a long time and test rare conditions that could escape standard specific systematic tests. For each bunch of events, it simulates the AM system to predict which patterns will fire. We will also use it for diagnostic purposes during real data taking to debug errors on the boards in the shortest time as possible, so that a minimum number of events from the detector are lost. For the "Random Test" we use a test board able to provide the input data and receive the output matched patterns at the real experimental rate. We perform these steps:

- We generate random patterns and we download the bank in the chips.
- We generate random data, enriched with words that fire patterns.
- We simulate the data flow of the AM system and calculate the patterns that are expected to fire, taking into account the contents of the bank and the data sent to the input.
- We download the input words to the test board through VME and we let them flow to the AM system at full speed.
- Fired patterns are sent back to the test board and saved into its spy buffers. We read them by VME.
- Finally, we compare these patterns with the expected ones.

The Board has been successfully tested using these events in a long test of 3 days without any error. It will be installed in the ATLAS experiment to take data for the first time at the beginning of 2016.

B. Quality of the Serial Links and clocks

We tested systematically all the serial links internal to the AM board and also those connecting the AM with the test board before producing the final prototype. We observed quality dependence on (a) the length of the link and (b) also on the design method. As a consequence, we could optimize the final PCB.

To improve the quality of the links that connect the farthest LAMB to the ROAD chip (~ 50 cm long), we placed a slink repeater in the middle of the path. In Table I are presented the measured parameters for the eye as a function of the length of the lines.

The parameters shown in Table I are defined into reference [19], [20]: (a) Eye Height: it measures the vertical opening of the eye diagram; (b) Eye Width: Time between the two points crossing the zero level; (c) Eye BER: is an extrapolation of the BER from the Eye Diagram. We also tested the three different configurations of serial lines shielding discussed in the reference [17]: (1) Float guard lines; (2) Guard lines closed to GND through 50 ohm resistors; (3) Grounded guard line. Table II shows the results for the different configurations.

The BER estimation extrapolated from the Eye diagram is a good quality parameter to understand the effect of the line configuration on the signal quality. It is clear from the measurements that in the grounded shielding case the noise and cross-talk effect is reduced.

Combining the solutions suggested by these tests, we reached a good signal quality. In Figure 8 is shown a measurement on a final serial link design. The Time Interval Error Histogram (TIEH) [20] has a non-Gaussian behavior. This effect is due to a big component of deterministic jitter (Dj) mixed with the random jitter (Rj). The Dj depends on data sequence and differences between rise time and falling

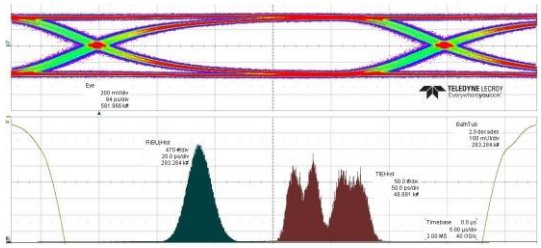

Figure 8. Serial data link analysis.

time. However, the total jitter (Tj) is low (~ 130 ps) and good enough to meet the tolerance of the AM chips receiver. Moreover, the Eye BER is estimated ~ 10^{-51}, if we take also into account the voltage noise.

We also directly tested the BER of all the serial links on the board with a PRBS-7 pattern generator, using the pattern checker inside the AM chip receiver or in the Artix FPGAs, depending on the receiver of the transmission. We sent PRBS-7 data for 48 hours at 2 Gbit/s with no errors on any link. This test translates into a BER smaller than 3×10^{-15}.

In order to check the quality of the differential lines, we also took Time Domain Reflectometry (TDR) measurements on all the lines for serial transmission. The error on the measured differential impedance value, according with the design, was always under 10%.

Finally, we also tested the quality of the tens of low jitter clock generators placed on the boards. The largest Tj is roughly ~ 60ps, good enough to meet the specifications of the AM chip.

Our single link speed is lower than today trends: high speed serial interconnects are dominant in almost every embedded systems design, from consumer and mobile devices to the high end routers and switches that power the wired Internet backbone, where data rates are 10 Gbit/s or more [21]. We have on purpose chosen the 9U VME standard to spread around computing elements and I/O resources on the large area provided by this standard. We implement a large number (~850) of medium frequency (2 Gbit/s) serial links to keep low the consumption per unit area on the PCB. However, the total traffic on the board is above the Tbit/s, for a total of more than 100 Tbit/s in the whole AM system used by the ATLAS experiment. This approach is not common in the literature, where the trend is to realize higher speeds and performances on small areas.

C. Pattern Matching Performances

Our hardware satisfies the system requirements. Candidate tracks are found exploiting the detector readout time, few clock cycles after the arrival of the corresponding detector channels belonging to the track.

This powerful highly parallel dedicated hardware has been demonstrated using the experiment simulation [7] to provide excellent performance, reaching resolutions, efficiencies and fake track rejection typical of the best tracking algorithms. For this reason, the use of the system in offline reconstruction applications has also been proposed [14], with the advantage of a low power usage (250 W/board). The system in fact is very compact and requires simplified infrastructures [22] compared to the ones necessary for the huge CPU farms executing an equivalent task. Four racks of electronics, for a total power of ~40 kW, are able to reconstruct events with an average latency of ~100 μs [7], while offline tracking requires several seconds when performed on events containing 60 p-p collisions [6].

One interesting technology which recently has attracted the attention of the high energy physics community for real-time applications is graphic processing. Both ATLAS ([23], [24]) and CMS [25] are studying the performance of real time tracking at LHC executed on modern Graphic Processing Units (GPUs). Even if the comparison with the CPU performances is promising, the latency to execute tracking is at least tens of milliseconds for simplified algorithms and reduced detector occupancies, with a fast grow above hundreds of milliseconds when the occupancy increases. In conclusion, our hardware dedicated approach is today thousands of times faster than any available commercial computing device.

The short latencies, reachable by the parallelized AM system, push both CMS and ATLAS to study its possible application at L1 [26] for the future accelerator upgrades, when the LHC luminosity will cause hundreds of pile-up collisions and will require much faster and more efficient trigger selections.

V. Conclusions

The presented powerful, highly parallelized pattern matching system exploits dedicated hardware to provide excellent timing performance, reaching resolutions, efficiencies and precision typical of algorithms executed on CPU farms. The system has been thoroughly tested and has been proved to be robust from both algorithmic and hardware point of view. It has been developed for the ATLAS real-time event processing for particle tracking from proton-proton collisions, but it can be adapted to generic image processing applications.

The planned future evolution includes miniaturization for its use as a coprocessor in any kind of image reconstruction, included high precision reconstruction of LHC events. Such a coprocessor can target any data processing system that is based on pattern recognition on massive data throughput (namely “big data” problems).

References

[1] The ATLAS Collaboration, “The ATLAS Experiment at the CERN Large Hadron Collider,” *Journal of Instrumentation* 3 S08003, 2008.
[2] The CMS Collaboration, “The CMS experiment at the CERN Large Hadron Collider,” *Journal of Instrumentation* 3 S08004, 2008.
[3] L. Evans and P. Bryant, “LHC machine,” *Journal of Instrumentation* 3 S08001, 2008.
[4] The ATLAS Collaboration, “ATLAS high-level trigger, data-acquisition and controls: Technical Design Report”, ATLAS-TDR-16, CERN-LHCC-2003-022, available online: http://cds.cern.ch/record/616089?ln=en.
[5] W. Smith, “Triggering at LHC Experiments”, *Nucl. Instr. and Meth. A*, vol. 478, pp. 62–67, 2002.
[6] The CMS Collaboration, “Description and performance of track and primary-vertex reconstruction with the CMS tracker”, *Journal of Instrumentation* 9 P10009, 2014.

[7] A. Andreani, et al., "The FastTracker Real Time Processor and Its Impact on Muon Isolation, Tau and b-Jet Online Selections at ATLAS," IEEE Trans. on *Nuclear Science*, vol.59, no.2, pp. 348-357, April 2012.
[8] M. Dell'Orso and L. Ristori, "VLSI Structures Track Finding", *Nucl. Instr. and Meth. A*, vol. 278, pp. 436-440, 1989.
[9] K. Pagiamtzis and A. Sheikholeslami, "Content-addressable memory (CAM) circuits and architectures: A tutorial and survey," in *IEEE Journal of Solid-State Circuits*, vol.41, no.3, pp. 712-727, March 2006.
[10] C.-L. Sotiropoulou, S. Gkaitatzis, A. Annovi, M. Beretta, P. Giannetti, K. Kordas, P. Luciano, S. Nikolaidis, C. Petridou and G. Volpi "A Multi-Core FPGA-based 2D-Clustering Implementation for Real-Time Image Processing", in IEEE Trans. on *Nuclear Science*, vol. 61, no. 6, pp. 3599 - 3606, December 2014.
[11] C. Wissing, et al., "Performance of the H1 Fast Track Trigger Operation and Commissioning Results," in IEEE *RT Conference (14th IEEE-NPSS)*, 2005, Stockholm,
[12] W. Ashmanskas et al., " The CDF online Silicon Vertex Tracker", *Nucl. Instr. and Meth. A*, vol. 485, pp. 178-182, 2002.
[13] A. Annovi, et al., "VLSI Processor for Fast Track Finding Based on Content Addressable Memories", in IEEE Trans. on *Nuclear Science*, vol. 53, no. 4, pp. 2428-2433, August 2006.
[14] A. Annovi, et al., "Associative memory design for the Fast TracK processor (FTK) at Atlas," in IEEE *NSS/MIC*, 2009, Orlando, pp. 1866 – 1867.
[15] A. Andreani, et al., "Characterisation of an Associative Memory Chip for high-energy physics experiments," in *Proc. I2MTC*, 2014, Montevideo. pp. 1487 – 1491.
[16] IDT, "IDT Thermal Considerations in Package Design and Selection," Application Note, available online: http://www.idt.com/document/apn/842-thermal-considerations-package-design-and-selection
[17] F.D. Mbairi, W.P. Siebert, H. Hesselbom, "On The Problem of Using Guard Traces for High Frequency Differential Lines Crosstalk Reduction," in IEEE Trans. on *Components and Packaging Technologies*, vol. 30, no. 1, pp. 67-74, March 2007.
[18] Xilinx Inc, "7 Series FPGAs GTP Tranceivers," User Guide, available online: http://www.xilinx.com/support/documentation/user_guides/ug482_7Series_GTP_Transceivers.pdf
[19] Teledyne Lecroy, "SDA, ASDA and SDM, SDA Serial Data Analyzer and SDM, Serial Data Mask Package, Theory of Operation," White Paper, available online: http://cdn.teledynelecroy.com/files/whitepapers/sda_theory.pdf
[20] Teledyne Lecroy, "Operator's Manual: SDAIII-CompleteLinQ Software," Manual, available online: http://cdn.teledynelecroy.com/files/manuals/sdaiii-completelinq_operators_manual.pdf
[21] B. Cole, "High Speed Serial Links are Here to Stay," available online: http://www.embedded.com/electronics-blogs/cole-bin/4404171/Editor-Note-High-speed-serial-links-are-here-to-stay, January 2013
[22] D. Calabro, et al., "The Associative Memory Boards for the FTK processor at ATLAS," in IEEE *NSS/MIC*, 2013, Seoul, pp. 1 – 5.
[23] D. Emeliyanov, et al., "GPU-based tracking algorithms for the ATLAS high-level trigger," in *Journal of Phys. Conf.*, Ser. 396, 012018, 2012.
[24] J. Mattmann, et al., "Track finding in ATLAS using GPUs," in *Journal of Phys. Conf.*, Ser. 396, 022035, 2012.
[25] V. Halyo, et. al., "GPU Enhancement of the Trigger to Extend Physics Reach at the LHC," *Journal of Instrumentation* 8 P10005, 2013.
[26] A. Annovi, et al., "Associative Memory for L1 Track Triggering in LHC Environment," in IEEE Trans. on *Nuclear Science*, Vol. 60, No. 5, pp. 3627 – 3632, 2013.