

Highly Parallelized Pattern Matching Execution for ATLAS Event Real Time Reconstruction

N. Biesuz, S. Citraro, S. Donati,
M. Piendibene, E. Rossi
C.-L. Sotiropoulou, G. Volpi
University of Pisa
Largo B. Pontecorvo 3, 56127
Pisa, Italy

A. Annovi, A. Andreani, M.
Berreta, P. Giannetti, A. Lanza,
V. Liberali, P. Luciano,
D. Magalotti, S. Shojaii,
A. Stabile
INFN
Via Enrico Fermi 40, 00400
Frascati, Italy

R. Beccherle, F. Crescioli
LPNHE
Couloir 12-22, 4rth floor, Place
Jussieu, 75005
Paris, France

C. Gentsos, N. Kimura,
K. Kordas, S. Nikolaidis
Aristotle University of Thessaloniki
Department of Physics, 54124
Thessaloniki, Greece

D. Dimas, A. Sakellariou
Prisma Electronics SA
El. Venizelou 128,
Nea Smyrni, 17123
Athens, Greece

W. Billereau, J.M. Combe,
P. Vulliez
CERN
CH-1211, Geneva 23,
Switzerland

ABSTRACT

In this paper a high performance "pattern matching" system is presented. The system is based on the concept of Associative Memory (AM), designed to solve the track finding problem that is typical of high energy physics experiments executed in hadron colliders. It is powerful enough to process data produced from 80 overlapping proton-proton collisions at a 100 kHz rate, in a time span of a few microseconds, even very high multiplicity events. The AM is designed for massive parallelism in data correlation searches. This system is implemented as a large array of custom VLSI chips (AM chips), based on Content Address Memory (CAM). All the chips are identical and each one of them stores a preset number of "patterns". All the patterns in all the chips are compared in parallel to the incoming data from the detector while the detector is being read out. Data are distributed to the AM chips through a huge network of high speed serial links. The complexity of the "pattern matching" problem is one that increases exponentially when CPU-based algorithms are used. With the proposed system the complexity increase is reduced to linear and the problem is solved by the time data are loaded in the system.

General Terms

Algorithms, Measurement, Performance, Design, Experimentation.

Keywords

Pattern matching, Associative Memory, ASIC, FPGA, ATLAS, Trigger

1. INTRODUCTION

In recent years there has been a great 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 applications 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 constitute a predefined pattern. Pattern matching is not limited to image processing, but is extended to other fields such as data servers (e.g. search engines, data) and all types of data processing that require identification of patterns. The presented system can execute 1 million comparisons on a single chip every 10ns, while 64 chips work in parallel on each system board. The complete system can integrate as many boards as required, all working in parallel. Such high performance requirements can be found in high energy physics experiments executed in hadron colliders.

These high energy physics experiments executed in hadron colliders search for extremely rare processes hidden in much larger background levels. The experiments are performed by overlapping proton-proton collisions that produce particles that leave traces to the detector's millions of detecting elements (100 million detector elements are used in ATLAS). Each one of these overlapping proton-proton collisions is called an "event". The data flow is so massive that only a very small fraction of the produced collisions can be stored to tape. A drastic real-time data reduction must be obtained with minimal loss of useful information.

A multi-level trigger is an effective solution for an otherwise impossible problem. The level-1 (L1) trigger is historically based on custom processors and reduces the rate from the machine event production down to tens of kHz. With the current upgrade of the Large Hardon Collider (LHC) the level-1 trigger will reduce the event rate to 100 kHz. The level-2 (L2) has been implemented with dedicated hardware in the past, and with standard CPUs more recently at LHC. The L2 output rate is usually few kHz. The level-3 (L3) selection has always been performed by CPU farms and its output is the one required for data storage on tape.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

Conference '10, Month 1–2, 2010, City, State, Country.

Copyright 2010 ACM 1-58113-000-0/00/0010 ...\$15.00.

These multilevel triggers work by quickly identifying interesting events. These are events that provide useful information for future analysis. Tracking devices, and in particular Silicon devices 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 each particle, individually detected even in very high occupancy conditions, and they can discriminate most of the different paths of particles produced in the same set of overlapping collisions in the same recorded image. However these detectors contain hundreds of thousands or millions of channels, so they require huge computing power for full tracking. They make the problem of complete tracking a formidable challenge even for off-line analysis. As a consequence, complete high-quality tracking for real time event selection at very high rates (low trigger levels) has been considered impossible in LHC experiments. Real-time tracking was planned for a limited detector region or on a small subset of events, previously selected using other detectors. With the presented system we overcome this problem by providing real-time tracking by using a massively parallel high performance system.

2. PATTERN MATCHING FOR THE ATLAS FAST TRACKER

The presented implementation was developed for the Fast Tracker Processor (FTK) [1] that is an approved ATLAS upgrade. The implemented strategy was based on the optimal mapping of a complex algorithm in different technologies. The target is to get the best results by combining the high performances of rigid dedicated hardware with the distinctive flexibility of general-purpose but lower-performance CPUs. The architecture's key role is played by high-level field programmable gate arrays (FPGAs), while most of the computing power is provided by cooperating full-custom ASICs named Associative Memories (AM). Powerful highly parallelized dedicated hardware is built to provide excellent performances, reaching resolutions, efficiencies and fake rejections typical of offline algorithms, short latencies (few tens of microseconds), energy saving (the AM chip, a device able to execute 1 Million of comparisons each 10 ns, has a power consumption below 3 W), and small occupation of space (4 racks of electronics is able to perform a task that would need a farm of thousands of commercial CPUs).

The AM, the central device of our system, shares some features with the Content-Addressable Memory (CAM) [2], usually used in very high speed searching applications. Even if AMs and CAMs are similar devices, there are conceptual design differences in our proposed AM chip design. The innovation in the Associative Memories used in our system is that each pattern is stored in a single memory location like in the commercial CAM, but it consists of 8 independent words of 16 bits each. Each word refers to a particular item to be identified in a flux of data that is private of the words that occupy that position in the pattern. In fact data are sent on 8 parallel buses, one for each word of the pattern. Each word is provided with reserved hardware comparators and a match flip-flop. All words in the AM can make independent and simultaneous comparisons with the data serially presented to its own bus. Any time a match is found, the match flip-flop is set. A pattern matches when a majority of its flip-flops are set. FPGAs control, configure and handle the AM providing the flexible computing power to process the selected shapes. Distributed debugging and monitoring tools suited for a pipelined, highly parallelized structure and high degree of configurability

have been developed to cope with different applications with the best possible efficiency.

AMs and CAMs have been used in the past for real time tracking. Pattern matching has been adopted in different ways, depending on the trigger level where it was used. Commercial CAMs have been used in the H1 experiment [3]. In the H1 experiment each bit of a CAM word corresponded to a detector channel. The whole event, made of a single large word, had to be submitted to the memory bank in the same clock cycle. In order to limit the number of channels to the largest CAM widths, usually smaller than 1000 bits, only a small detector section was analyzed. Detector data came in the form of a sequence of addresses of "hit channels" that are simply called "hits". Thus, additional hardware was needed to reformat the incoming data before sending them to the CAM. When the used detector section is sizable, the number of bits per word becomes prohibitively large for this method (15 bits to address a channel on each layer of an 8-layer detector would require a chimerical 23x215 bit wide CAM). The first produced AM device [4] has been applied without problems to this case [5]. A full-custom VLSI technology was used in this context to produce the first AM for the CDF experiment. Each of the words of one 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. All words in the AM could make independent and simultaneous comparisons with the hit addresses serially presented to their common buses. Layer matches could happen at different times, since they are stored in flip-flops and continuously checked for coincidence with the other layers to produce a track match.

The presented system was developed and tested for high energy physics detectors but the problem is essentially an image processing problem. Therefore the system can be adapted to be used by more generic image processing applications.

3. IMPLEMENTATION

We have developed a new Associative Memory system for the ATLAS experiment. It is organized into 128 Processing Units (PUs) that process the tracker data in parallel, working on different sections (towers) of the detector. The whole AM system stores 1 billion (10^9) AM patterns. The PU is made of a 9U VME card, the AM board assembled with 64 AM chips, and a Rear Transition Module (RTM), named AUX card, which is placed in the same slot of the VME core crate. The AUX card communicates with the AM board through a high density high speed connector providing the input data and collecting the fired patterns.

The design of the AM system is a challenging task, due to the following factors: (1) the high pattern density (8 million patterns per board), which requires a large silicon area; (2) the I/O signal congestion at the board level, which requires the use of serial links; and (3) the power limitation due to the cooling system: as we are fitting 8 000 AM chips in 8 VME crates and 4 racks, the power should not exceed 250 W per AM board.

3.1 The AM chip

A critical figure of merit for a AM-based track reconstruction system is the number of patterns that can be stored in the data bank. In the past, the request to maximize available patterns forced a full-custom VLSI approach, which implied a big development effort and a difficult upgrade path to more recent and denser micro-electronic technologies, as they eventually become available. After that experience, very high density silicon technologies made it possible to build a very large number of

transistors inside a reasonably large silicon area (say, $\sim 1\text{cm}^2$). It was therefore appropriate to reconsider the best trade-off between pattern density and ease of design (and eventually re-design). While the full-custom approach obviously maximizes pattern density, an FPGA-based design gives the fastest development time at the cost of a drastically reduced pattern density. This option has been considered in [6]. Despite the recent FPGA progress, these devices were and are still not convenient for our application. Midway between the two approaches, a standard-cell based design brings substantial advantages, as discussed in details in the paper [7] that describes the design of AMchip03 for the CDF experiment.

The requirements for the ATLAS FTK application, however, are more demanding than those for CDF: a bigger silicon detector with higher granularity requires more patterns and higher trigger frequency requires higher operating frequency, while the total power consumption must be contained. The next generation of AM devices for ATLAS introduced a mixed architecture: full custom blocks for the CAM cells, standard cell logic for everything else, in particular the control logic. The chosen technology is TSMC 65 nm. The use of full custom CAM cells enabled a higher pattern density with respect to AMchip03 and also the use of advanced techniques to reduce power consumption, more than what expected from simple node scaling from 180 nm to 65 nm. The full custom design effort was anyhow limited to a small piece of the large memory, a cell that could be replicated many times in the very structured area of the chip, occupying the largest fraction of the die. The control logic instead was totally implemented with standard cell, easily handled and simulated by the development software. With this method the design effort, the degree of reliability and the chip consumption could be maintained inside the desired limits.

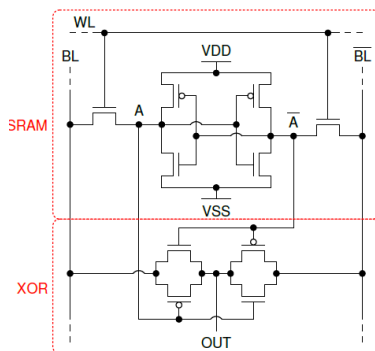


Figure 1. XORAM schematics.

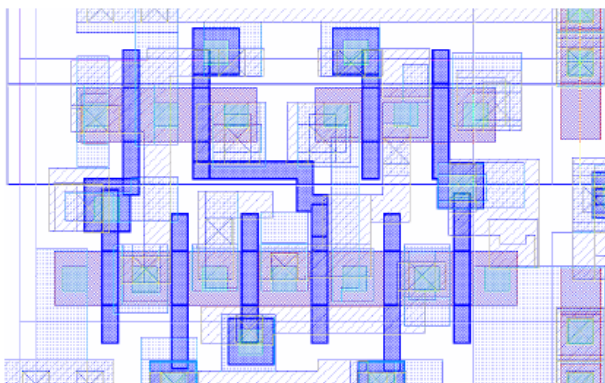


Figure 2. Layout of the XORAM block in 65 nm CMOS technology.

Another very important feature was introduced in the new AM chip: ternary logic bits. Some bits in the CAM cell can store ternary values (1, 0, don't care) and they can be used to achieve a variable resolution pattern. The idea of variable resolution pattern is essential in ATLAS to have a high efficiency pattern bank without increasing the capacity of the AM system over the foreseen one billion patterns [1].

The full custom designed CAM cell has been described in [8]. It is based on the XOR logic function, and it is made of a conventional 6T SRAM cell merged with a pass-transistor XOR gate. Figure 5 shows the CMOS schematic diagram, and Figure 6 illustrates the layout of a 1-bit cell. The single bit cell output (OUT) is equal to zero when the stored bit (A) matches the bit-line (BL), and is equal to one when they are different. The comparison on the 18-bit words is made by taking the logic NOR of the 18 AM cell output bits.

The AM chip used parallel busses for I/O in the past. This led to extreme complexity in the design of the mezzanine boards to host the AM chips, each board hosting 16 or 32 of them. Furthermore for the new device is it foreseen to use different power domains (1.0 V for the AM core and the standard cells, 1.2 V and 2.5 V for I/O) increasing again the routing complexity of the board. In order to solve this board routing issue we decided to switch from parallel busses to high speed serial busses. The package of the AM chip also changed from TQFP208 to BGA 23x23 in order to use a modern flip-chip technology, including a heat slug for high dissipation capability, many pins for the many power domains and a small number of pins, optimally routed, 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 has 11 serial links.

The main features required for the AM chip serial links (SERDES) are:

- data rate at least 2 Gb/s to match 16 bit @ 100 MHz
- 8b/10b encode/decode capabilities
- separate serializer and deserializer macro (the AM chip has many input busses but one output bus for patterns)
- 32bit input/output bus
- driver and receiver circuits compatible with LVDS standard
- comma detection and word alignment
- BIST capabilities for fast debugging
- Low power

We have bought SERDES IP by Silicon Creations meeting all our requests. We have produced 200 AM chips (MPW run) with the final functionality but a much smaller bank, only 2000 patterns.

Figure 3 shows the test stand setup we have built to test the chips using a zip socket, to select the good ones for the final system.

3.2 The boards: Putting Chips Together

A 9U-VME board filled with 64 AM chips can allocate 8 Millions of patterns. To simplify input/output operations, the AM chips are grouped into AM units composed of 16 chips each, called Little Associative Memory Boards (LAMB, Figure 4). A 9U-VME board has been implemented to allocate 4 of such units. Figure 5

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 characterizes the data distribution from the input (the high density connector in the green box on the bottom-right side of Figure 5, called P3) to the 64 AM chips and back to the connector, for a total of ~750 point-to-point connections. Twelve input serial links (in yellow) provide the silicon data from the P3, 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 to P3.

The data traffic is handled by 2 Xilinx FPGAs. They are 2 Xilinx-Artix7 which have 16 Gigabit Transceivers (GTP) each providing ultra-fast data transmission. The FPGA in the yellow box in Figure 5 handles the input data, while the FPGA in the red box near the P3 handles the output data. Two separate Xilinx Spartan-6 FPGAs implement the data control logic. 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 data rate is really challenging. A huge number of silicon data must be distributed at high rate (2 Gb/s on each serial link, for a total of 24 Gb/s maximum rate), with extremely large fan-out. Events can enter the board with a maximum rate of 100 kHz. Each 10µs in average, 8 thousand words (16 bits) have to reach the patterns through 8 buses and a similarly large number of output words must be collected and sent back to the P3 (32 Gb/s maximum output rate). Each input word has to reach the 8 million patterns of the board.

The large input fan-out is obtained through 3 levels of serial fan-out chips to reach each of the 64 devices and a very powerful data distribution tree inside each device itself. The AM chip compares 8 input words with 128k locations each 10 ns. The first level of 1:2 fan-out is visible inside the 2 yellow boxes of Figure 5, providing each of the 8 buses to the 4 LAMBs. The other two levels are placed on the LAMBs and are visible in Figure 6.

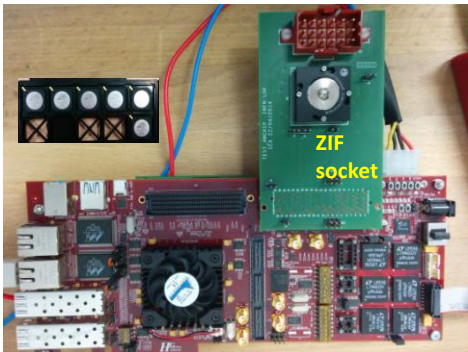


Figure 3. The AM chips and the test setup



Figure 4. The LAMB assembled with 16 AM chips.

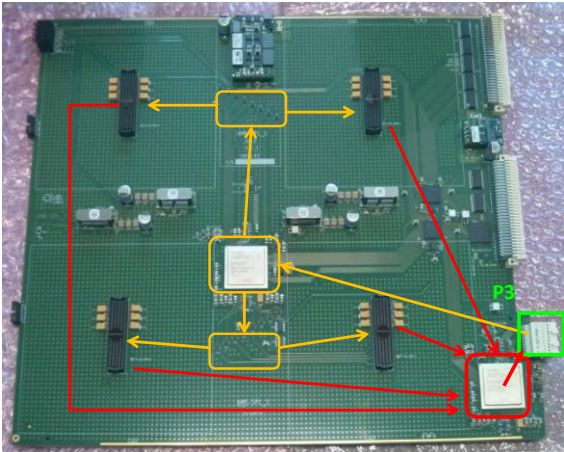


Figure 5. The data traffic in the motherboard.

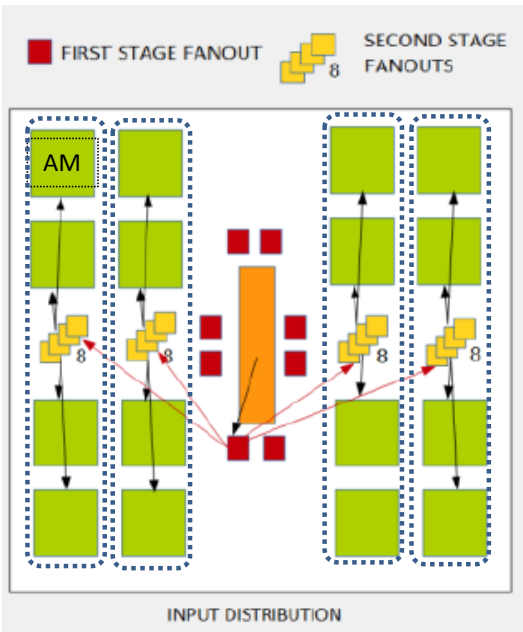


Figure 6. Input data distribution to AM chips.

Each LAMB has 40 1:4 fan-outs. The 8 red ones around the central connector (orange box) replicate 4 times each of the 8 incoming buses to make them available to a quartet of AM chips. For the input data distribution AM chips are organized into

vertical quartets as shown by the blue dotted lines in Figure 6. The second level of fan-outs (yellow little squares) replicates again the bus 4 times, one for each single AM device in the quartet. The placement of chips on the LAMB has been studied and optimized with the goal of minimizing the crossing of the serial links.

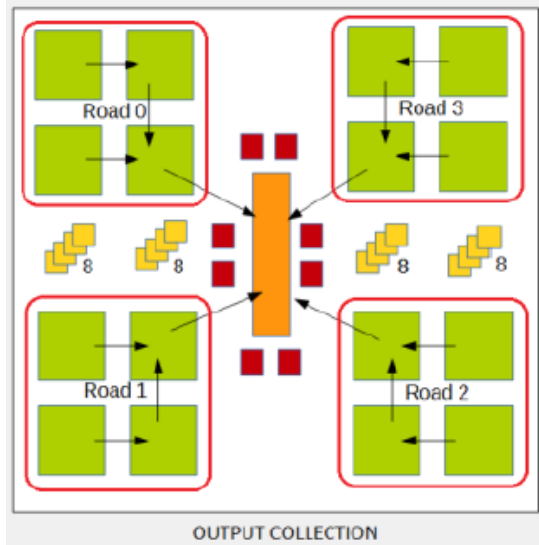


Figure 7. Output data collection from AM chips.

Figure 7 shows how the output words are collected from the 16 AM chips, connected in 4 daisy chains. Each AM device has the capability to receive outputs from other two AM chips and merge them internally with fired patterns found in the chip itself. Each daisy chain has a single output that goes directly to the connector. Each quartet shares also a 100 MHz low jitter clock necessary for the 11 serial links handled by each AM chip. The oscillator and the 1:4 fan-out for its output distribution are placed exactly in the middle of the quartet in the red boxes.

Particular care has been devoted to the PCB routing, in particular for the many serial links (~200 links), to keep the relative impedance fixed at $100\ \Omega$ and to minimize the cross talk. It is a 12 layer PCB where signal planes 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 they are shielded from other lines in the same plane by metal ground fill.

3.3 System Control and configuration

The AM system is hosted in 9U VME crates and it is fully controlled and monitored using the VME standard. The VME slave interface, implemented in a Spartan6 FPGA, allows writing/reading functions to/from registers, memories and FIFOs, using random access or block transfer modes.

The most important implemented function is the configuration of the AM chips, in particular the upload of patterns that have to be stored in the memory.

The AM chips are configured through JTAG port. The 64 chips are organized into 32 chains of 2 AM chips each. The chains are handled in parallel to limit configuration time. The VME 32-bit wide data transfer is segmented into 4 bytes, each one assigned to a LAMB. On each LAMB 8 JTAG chains are handled by a small Spartan 6 FPGA (Blue box in Figure 8).

The VME slave interface and the FPGA on LAMBS allow write/read of the JTAG registers contained in the AM chips. Pattern downloading time was measured to be ~20 seconds.

Another important part to be configured is the very large number of serial I/O interfaces. The AM chips alone, use 640 receivers and 64 transmitters that require proper initialization.

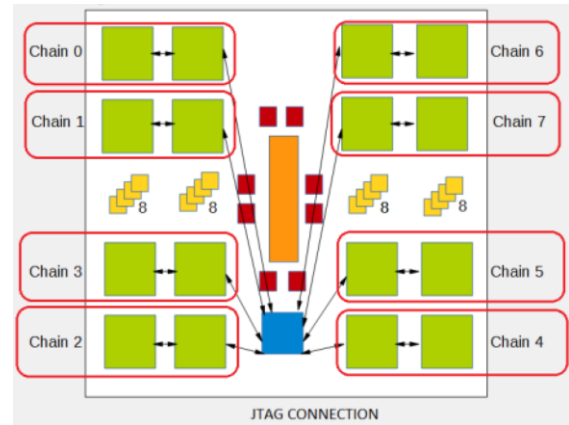


Figure 8. JTAG control of AM chips.

3.4 Data Flow and Event Synchronization

The AM system is part of a data driven pipeline where a large number of devices are connected by thousands of links: 16400 dedicated custom chips (AM chips) that perform pattern matching and 2000 FPGAs for all other functions.

A simple communication protocol is used for data transfers. The data flow through serial links connecting one source to one destination. The protocol is a simple pipeline transfer driven by control words, for example idle words and alignment words. An 8b/10b encoding is used in the serial data stream in order to provide effective error detection, i.e: a 32-bit word is transmitted as 40 bits. The idle word is transmitted when no valid data is available. On each link the information is transmitted in data words whose format depends on the kind of information being processed in that portion of the pipeline. Alignment words are periodically transmitted between data words. Input FTK words in each processing step of the pipeline are pushed into a de-randomizing FIFO buffer. All the words that are 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. The source and destination devices are two separate logic functions in the pipeline which can be on separate boards or even be two functions in the same large FPGA.

To maximize speed, no handshake is implemented on a word-by-word basis. 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 the HOLD signal. The source responds to the HOLD signal by suspending data flow. Using Almost Full instead of Full gives the source enough time to stop. Since the source is not required to wait for an acknowledge signal from the destination device before sending the next data word, data can flow at the maximum rate compatible with the link bandwidth even when transit times are long. The standard clock frequency is 100 MHz for 16-bit words or 50 MHz for 32-bit words, which corresponds

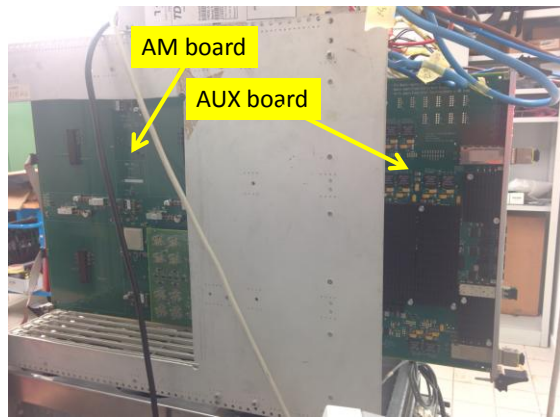


Figure 10. The test stand for debugging

- We generate random patterns and we download the bank in the chips.
- We generate random data, enriched of good words that fire patterns.
- We simulate the data flow of the AM system calculating the expected fired patterns keeping into account the knowledge of the bank and the data to be sent in input.
- We download the input words to the AUX through VME and we let them flow to the AM system at full speed.
- We read back by VME the real fired patterns received and stored back in the AUX.
- 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 on the experiment to take data for the first time at the end of 2015.

5. FUTURE EVOLUTION

The future evolution of the presented system targets two different goals:

- Adapting the existing system to be used by generic image processing applications.
- A technological effort to "miniaturize" the system's PU and make it suitable to be used as a coprocessor for speed up of offline tracking algorithms. Such an implementation can be used for the targeted generic image processing applications.

The AM system fundamentally executes a filtering function that can also target images of different nature. The AM-based processor can simulate the preliminary stages of image processing performed by the brain for vision, such as the identification of shape edges [9]. The most convincing models that try to validate brain functioning hypotheses are extremely similar to the real time architectures developed for High Energy Physics experiments. A multilevel model seems appropriate also to describe the brain organization to perform a synthesis certainly much more impressive than what done in HEP triggers. The AM pattern matching has proven to play a key role in high rate filtering/reduction tasks [9]. We can test the AM device capability as the first level of this process, dedicated to external stimuli pre-processing. We follow the conjecture that brain works by dramatically reducing input information by selecting for higher-level processing and long-term storage only the input data that match a particular set of memorized patterns. The double constraint of finite computing power and finite output bandwidth

determines to a large extent what type of information is found to be "meaningful" or "relevant" and becomes part of higher level processing and longer-term memory. The AM-based processor will be used for a real-time hardware implementation of fast pattern selection/filtering of the type studied in these models of human vision and other brain functions. Shapes extracted by the AM from the images would be analyzed exploiting the computing power of the FPGAs to identify clusters of contiguous pixels above a programmable threshold [10]. The AM, cooperating with the FPGAs could have a new nice application in the field of Smart Cameras. As a summary, this multi-chip system will try to reproduce the initial stages of the brain visual processing: the ASIC will extract object contours and the FPGA will analyse their shape.

The systems miniaturization will be achieved by producing a System In Package (SIP) where the FPGA, an external large memory and a single AM chip are packaged together [11].

6. CONCLUSIONS

In this paper a powerful, highly parallelized pattern matching system is presented. The system exploits dedicated hardware to provide excellent performances, reaching resolutions, efficiencies and false rejections typical of offline algorithms. The system achieves very short latencies (few tens of microseconds) and the system's core, the AM chip, a device able to execute 1 Million of comparisons each 10 ns, has peak power consumption below 3 W. The system itself is several magnitudes smaller than its CPU equivalent (4 racks of electronics is able to perform a task that would need a farm of thousands of commercial CPUs), Communication between chips is guaranteed by a powerful network of more than 750 2 Gb/s serial links. The system is developed for the ATLAS Fast Tracker Processor but it can be adapted to be used for generic image processing applications. The system is flexible and a planned future evolution is to be "miniaturized" in order to be used as a coprocessor for any kind of image reconstruction. Such a coprocessor can target any artificial intelligence process based on massive pattern recognition.

7. ACKNOWLEDGMENTS

The AM system project receives support from Istituto Nazionale di Fisica Nucleare; and the European community FP7 People grant FTK 324318 FP7-PEOPLE-2012-IAPP.

8. REFERENCES

- [1] Andreani A. et al. 2012. The FastTracker Real Time Processor and Its Impact on Muon Isolation, Tau and b-Jet Online Selections at ATLAS. *IEEE Transactions on Nuclear Science* 59, 2, 348-357. DOI= [10.1109/TNS.2011.2179670](https://doi.org/10.1109/TNS.2011.2179670)
- [2] Pagiamtzis, K. and Sheikholeslami, A. 2006. Content-addressable memory (CAM) circuits and architectures: A tutorial and survey. *IEEE Journal of Solid-State Circuits*. 41, 3, (Mar. 2006), 712-727. DOI=10.1109/JSSC.2005.864128
- [3] Wissing, C. et al. 2005. Performance of the H1 Fast Track Trigger Operation and Commissioning Results. In *Real Time Conference, 2005, 14th IEEE-NPSS*. DOI= [10.1109/RTC.2005.1547429](https://doi.org/10.1109/RTC.2005.1547429)
- [4] Amendolia, R. et al. 1992. The AMchip: a full-custom CMOS VLSI associative memory for pattern recognition. *IEEE Transactions on Nuclear Science*. 39, 4, pp. 795-797. DOI= 10.1109/23.159709

- [5] Jones, M. et al. 2008. The CDF II Level 1 Track Trigger Upgrade. *IEEE Transactions on Nuclear Science*. 55, 1, pp. 126-132. DOI= [10.1109/TNS.2007.911618](https://doi.org/10.1109/TNS.2007.911618)
- [6] Bardi, A. et al. 1998. A programmable associative memory for track Finding. *Nuclear Instruments and Methods in Physics Research A*. 413, (1998), 367-373.
- [7] Annovi, A., et al. 2006. VLSI Processor for Fast Track Finding Based on Content Addressable Memories. *IEEE Transactions on Nuclear Science*. 53, 4, (August 2006) 2428-2433. DOI= [10.1109/TNS.2006.876052](https://doi.org/10.1109/TNS.2006.876052)
- [8] Frontini, L., Shojaii, S., Stabile, A., and Liberali, V. 1994. A new XOR-based Content Addressable Memory architecture. in *Proceedings of the International Conference on Electronics, Circuits and Systems (ICECS), Seville Spain* (December 2012), 701-704. DOI=[10.1109/ICECS.2012.6463629](https://doi.org/10.1109/ICECS.2012.6463629)
- [9] Del Viva, M., Punzi, G., and Benedetti, D. 2013. Information and Perception of Meaningful Patterns. *PloS one* 8.7 (July 2013): e69154. DOI: 10.1371/journal.pone.0069154
- [10] Sotiropoulou, C.-L. et al. A Multi-Core FPGA-based 2D-Clustering Implementation for Real-Time Image Processing" In *IEEE Transactions on Nuclear Science*. 61, 6, (August 2006) 3599-3606. DOI= 10.1109/TNS.2014.2364183
- [11] Gentsos, C. et al. 2014. Future evolution of the Fast TracKer (FTK) processing unit. *Proceedings of Science*. 209