

Future evolution of the Fast Tracker (FTK) processing unit

C. Gentsos^{*a}, F. Crescioli^b, P. Giannetti^c, D. Magalotti^d, S. Nikolaidis^a

^a*Aristotle University of Thessaloniki, 54124 Thessaloniki, Greece*

^b*Laboratoire de physique nucléaire et de hautes énergies, Couloir 12-22, étage 4, Place Jussieu, 75005 Paris, Francia*

^c*Sezione di Pisa INFN, Largo Bruno Pontecorvo 3, 56127 Pisa, Italy*

^d*University of Modena and Reggio Emilia, Via Università, 4, 41121 Modena, Italy*

E-mail: cgentsos@physics.auth.gr, francesco.crescioli@pi.infn.it,
paola.giannetti@pi.infn.it, daniel.magalotti@pg.infn.it,
snikolaid@physics.auth.gr

The Fast Tracker (FTK) processor for the ATLAS experiment has a computing core made of 128 Processing Units that reconstruct tracks in the silicon detector in a 100 μ sec deep pipeline. The track parameter resolution provided by FTK enables the software-based High Level Trigger (HLT) trigger to efficiently identify and reconstruct significant samples of fermionic Higgs decays and other interesting physics processes. The data processing speed is achieved with custom VLSI pattern recognition, linearized track fitting executed inside modern FPGAs, pipelining, and parallel processing. One large FPGA executes full resolution track fitting inside low resolution candidate tracks found by a set of 16 custom ASIC devices, called Associative Memories (AM chips). For future fast tracking applications, the FTK dual structure which is based on the cooperation of VLSI dedicated AM and programmable FPGAs is maintained, but significant performance enhancements are achieved through miniaturization and tighter integration of the current state of the art prototypes. Those performance enhancements can be of use to new applications within and outside the High Energy Physics field. Specifically, we plan to increase the parallelism by associating one FPGA to each AM chip. The FPGA configures and handles the AM, and provides a flexible computing power to process the patterns selected by the AM. The benefits of using this new elementary unit made of 2 chips are: maximum parallelism exploitation, low power consumption, execution time at least 1000 times shorter than the best commercial CPUs, distributed debugging and monitoring tools suited for a pipelined, highly parallelized structure, and high degree of configurability to face different applications with maximum efficiency. We report on the design of the FPGA logic performing all the functions complementary to the pattern matching performed by the AM.

*International Conference on Technology and Instrumentation in Particle Physics
2-6 June 2014
Amsterdam, The Netherlands*

*Speaker.

Contents

1. Introduction	2
2. System Overview	3
3. Data Organizer	4
4. Track Fitter	4
5. FPGA firmware implementation results	6
6. Performance and latency	6
7. Conclusions	7

1. Introduction

The Fast TracKer (FTK) processor [1] for the ATLAS experiment [2] is a processing unit that increases efficiency in the software-based High Level Triggers (HLT) selection, by offering fast track reconstruction [3]. The algorithm is split in two parts; the first part performs pattern recognition on a lower resolution representation of the detector data; the second part computes linear fits of the tracks found in the matched patterns, using the full resolution version of the detector hits. The Fast TracKer system comprises a set of boards, two of which implement the main functionality. One is called Associative Memory Board (AMB), and mounts 4 Local Associative Memory Boards (LAMB), which in turn feature 16 Associative Memory Chips each; the pattern matching operation is performed in that board [4]. To each AMB, another board called the Auxilliary Board (AUX) is connected. Its main components are 4 FPGA units, corresponding to one LAMB each. They perform functions of data storage and the Track Fitting computations.

The goal is to integrate each Associative Memory Chip with its own FPGA device and an auxilliary RAM memory die into a single package which will be called AMSiP, essentially miniaturizing the FTK system. In that way, much more track fitting performance is available per pattern and much smaller latency figures are possible. The target of the upgraded system will be tracking to assist in first level (L1) trigger selection [5, 6]. As a first stage, the current FTK detector layout is chosen for the structural parameters of the architecture for reasons of practicality, but the resulting system will be parametric and possible to target to other applications. To test the design, a custom board consisting of the three discrete chips (the AM, FPGA and RAM) will be developed. Then, after the verification of the full design, the System-in-Package version is planned, and after its completion, new ATCA boards that will be based on the new AMSiP devices.

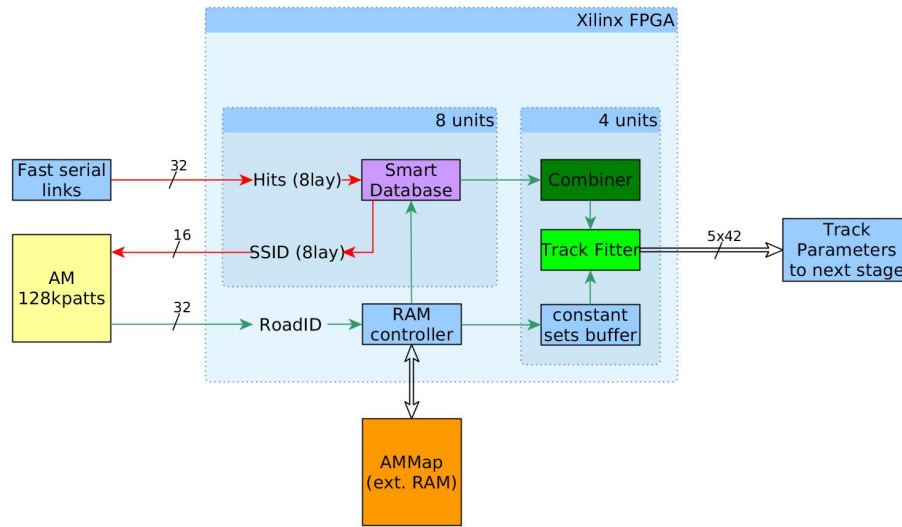


Figure 1: System block diagram

2. System Overview

The system comprises of three chips: an Associative Memory chip (AMChip), where the pattern matching is performed; an FPGA device, where the necessary logic for the hit storage and the Track Fitting calculations is implemented; and a RAM memory, connected to the FPGA, which is used as a Look-up table to decode the Road IDs - ID values of the patterns selected by the AMChip - to SuperStrip IDs - reduced resolution hit coordinates - utilized by the FPGA logic. Multi-gigabit serial links are used to carry the full resolution hits to the input, and provide the communication between the FPGA and the AMChip. The communication between the FPGA device and the AMChip, and system data input and output, are handled by fast Multi-Gigabit Transceivers. For a system-level block diagram, see Figure 1.

The event reconstruction is done in two steps. In the first step, after a full-resolution hit is received, its SuperStrip ID is calculated. That ID is a reduced-resolution coordinate of the hit, according to which pattern matching is done. The full-resolution hit coordinates are then stored in the Data Organizer according to its SuperStrip ID, which is sent to the AMChip via Multi-Gigabit transceivers for the pattern recognition stage of the algorithm. After all hits of an event have been received, stored in the DO and their SuperStrip IDs sent to the AMChip, the second step begins. Initially, the pattern recognition results from the AMChip, in the form of Road IDs (a matched pattern is called a Road), are read into the FPGA. The Road IDs are broken down in SuperStrip values per layer, using an external memory as a Look-Up Table. Those IDs are passed on to the Data Organizer, and all full-resolution hits contained in the coarse-resolution Roads are extracted. Afterward, all the possible combinations that can form actual tracks are produced, representing candidate tracks, and for each one a fast linear fit is performed. The χ^2 value and the track parameters are computed. Those tracks with a χ^2 value below a predetermined threshold are considered to be good quality fits, and the computed parameters are passed on as the system output to aid in the triggering decision.

3. Data Organizer

The Data Organizer (DO) implements a type of “smart” database. It operates in two modes; in *write* mode it stores the incoming full resolution hits according to their SuperStrip ID value; then in *read* mode it retrieves them according to the Road information obtained by the Associative Memory chip.

In write mode the hits arrive in a random manner and are stored according to their SuperStrip ID, with the only restriction being the maximum number of hits to store for a layer per event. In read mode, only the hits that belong to the specific SuperStrips suggested by the pattern matching step are extracted, in an orderly manner.

The core of the Data Organizer architecture is a set of memory elements that work together to construct a linked-list structure (for a simplified block diagram see Figure 2). Hits are stored sequentially in a dual-port memory, the “hit memory”, in the order in which they arrive. Another dual-port memory, “Hit List Pointer” (HLP), holds information on the location in the “hit memory” of the first hit to arrive for each SuperStrip. To avoid corruption of the HLP data between events, a register file is used in conjunction with a second, wider HLP memory port. The memory space is split in areas of 32 consecutive locations. The first time in an event data is written in a location in some area, the bit in the register file that corresponds to that area is toggled set to logical ‘1’, and the data is written while all the rest of the locations of that area are reset. The next time data is going to be written in that area, the logical “1” of the register file will indicate that the area has already been properly reset, and only the one location - where the data is meant to be written - is going to be written to. The whole register file is reset at the beginning of each event.

An additional dual-port memory, the “next memory”, having the same depth as the “hit memory” is used, for each hit of a SuperStrip, to point to the location of the next hit in the same SuperStrip - if there is one. Finally, to be able to write the contents of that “next memory”, another memory of the same depth is employed - the “last memory”, that points to the last hit of a SuperStrip and is utilized only in the writing phase, to provide the correct address when there is a need to write to the “next memory”.

In the reading phase, the Road ID received from the Associative Memory is translated to SuperStrip IDs per layer by performing a look-up in the external RAM, and with the help of the HLP and “next memory” the full-resolution hit contents of the selected SuperStrips are read out to the next stage of the algorithm, the Track Fitter.

4. Track Fitter

The first step of the Track Fitting process is to compute all combinations of hits across layers that can form tracks. The full-resolution hits from the Data Organizer are put in a memory structure divided in layers, and a pipelined module takes care of producing those hit combinations across layers. These hit combinations make up the candidate tracks, and the module passes them along to the Track Fitter computational pipeline.

The FPGA device provides fast dedicated DSP resources, able to perform multiplication and addition operations. To compute the χ^2 value and the track parameters, the Track Fitting stage performs scalar multiplications at a clock speed of 550MHz, making as much use as possible of

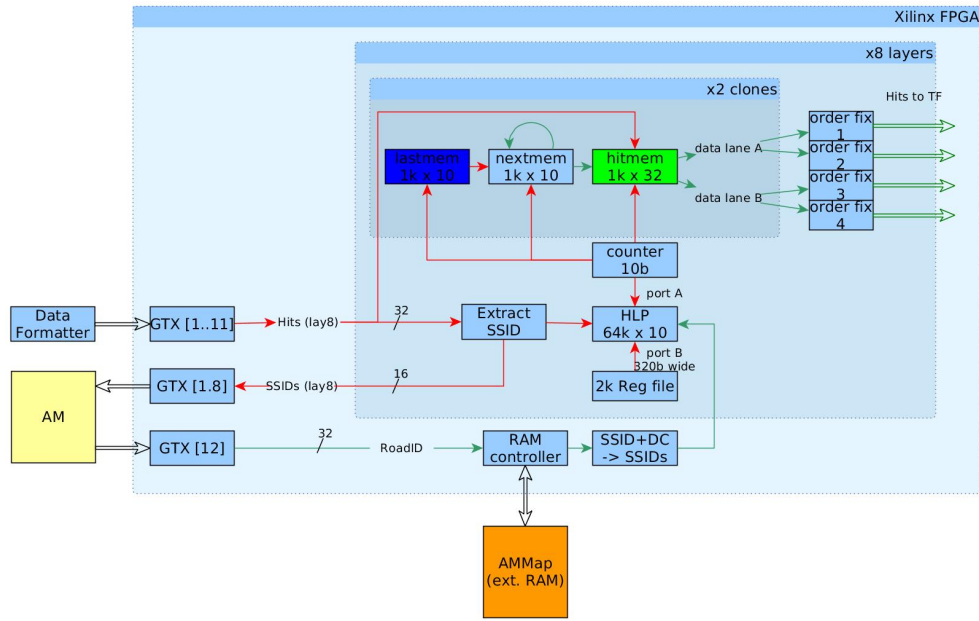


Figure 2: Simplified Data Organizer block diagram

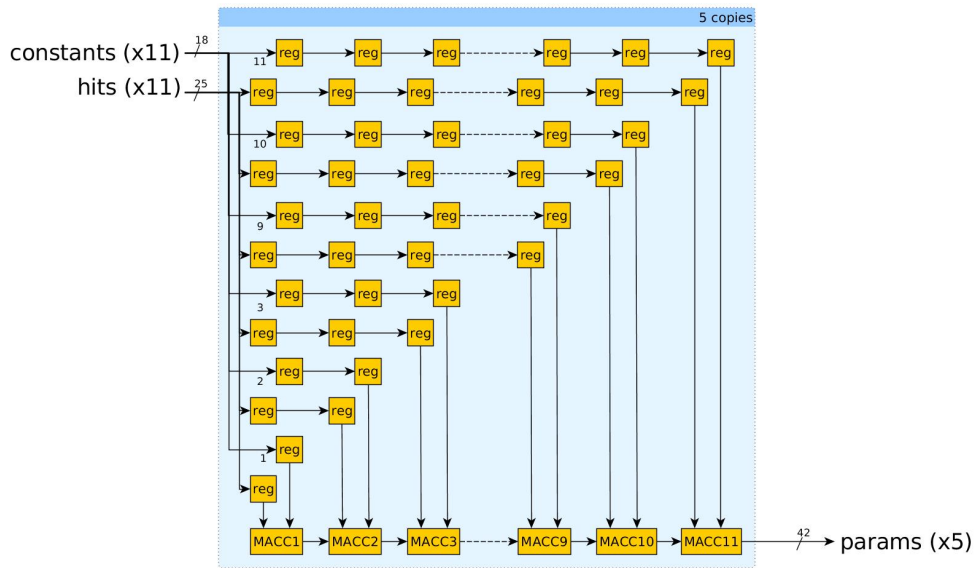


Figure 3: Track Fitter parameter calculation pipeline

those DSP units. Each DSP unit multiplier has a pipeline latency of 2 clock cycles, and can perform a new multiplication each clock cycle. The result of each multiplication goes in the dedicated adder unit of each DSP, to be added with the product of the previous unit. To account for each multiplier latency, the input data is fed to each DSP unit delayed by $2n$ clock cycles (see Figure 3). In that way, the latency of the track fitter pipeline is 24 clock cycles, which translate to 44ns.

To obtain the maximum performance possible, four Track Fitter units operate in parallel, making the system able to perform the full-resolution linear fits with a maximum rate of 2.2GFits/s.

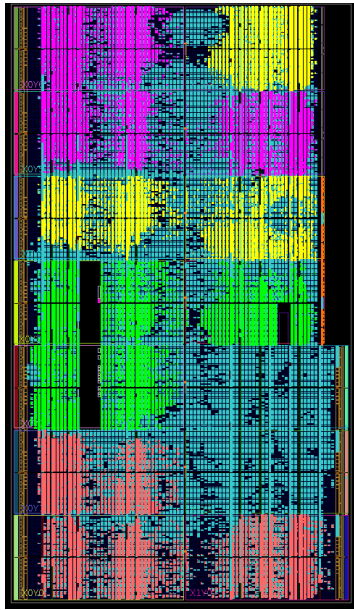


Figure 4: Placement of the Track Fitter modules in the FPGA device

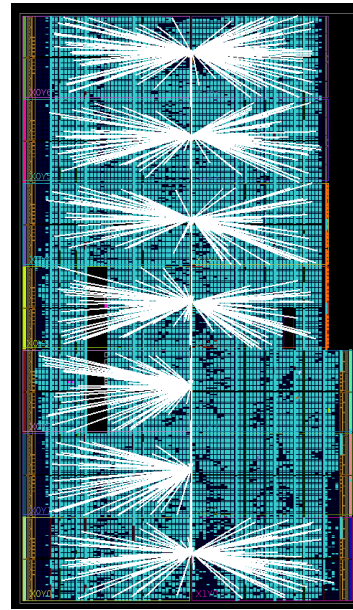


Figure 5: Clock distribution scheme for the Track Fitter modules

5. FPGA firmware implementation results

The key components of the design (Serial Transceivers, Data Organizer, Track Fitter) were implemented in a Xilinx 7-series Kintex device (xc7k325t-ffg900-3). To reach the frequency targets (450MHz for the Data Organizer and 550MHz for the Track Fitting module) several advanced design techniques were employed (e.g. study of the required pipeline depth in several places of the design, per-signal manual fan-out control where needed, manual clock buffer instantiation for the Track Fitter clocking).

A picture of the Track Fitter modules placed in the FPGA device can be seen in Figure 4, along with the respective clock buffer distribution in Figure 5. The design takes up 70% of the DSP units the device provides, 96% of the available Block RAM cells, 56% of the device registers and 13% of the LUT cells.

The power consumption is at 15.5W, which can be considered to be high. However, taking into account the fact that at this stage the design was not optimized for power consumption, this figure can be easily reduced by about 20% with minor modifications. Another 20% reduction can be expected by migrating to a newer device family, bringing the total expected power consumption to below 9W.

6. Performance and latency

The FPGA firmware was designed to support a future upgrade of the Associative Memory Chip, and thus supports higher data rates than the current AMChip does. The figure for the maximum input data rate is 225MHits/s per layer (AMChip currently supports 100MHits/s per layer),

whereas the maximum rate for handling Roads returned by the AMChip is 100M/sec (AMChip currently produces roads with a rate of 50M/s). The minimum latency of the system (the time interval from the last data received to the first result produced) is $\sim 0.3\mu s$.

To precisely evaluate the system performance, simulation input vectors that reflect the Phase II environment are needed. However, the required input vectors are not available yet.

We will have to consider the case of a typical event that consists of 500 hits per layer, and 50 roads returned from the pattern matching stage. A typical SuperStrip for this event could contain 1 hit on 4 of the layers and 2 hits on the rest 4 of the layers. Under those assumptions, the total event processing time would be $6.2\mu s$. If we consider the case of an upgrade to the AMChip to double its current performance, the total processing time for the same event would be reduced to $3\mu s$. Those are very satisfactory figures, considering the $8\mu s$ available for L1 tracking, which means that there is plenty of processing headroom to handle more complex events.

7. Conclusions

The main components necessary to miniaturize the FTK system, namely the Data Organizer and the Track Fitter, have been successfully implemented in the target FPGA device (Xilinx xc7k325-ffg900-3). The performance results are promising, although input vectors of the predicted L1 tracking environment in Phase II conditions are needed for the performance to be evaluated with increased confidence. Design power consumption figures have shown to be quite high; but a significant reduction in those figures is expected, especially after migrating to a newer device generation.

Acknowledgments

The Fast Tracker project receives support from Istituto Nazionale di Fisica Nucleare; the US National Science Foundation and Department of Energy; Grant-in-Aid for Scientific Research from the Japan Society for the Promotion of Science and MEXT, Japan; the Bundesministerium für Bildung und Forschung, FRG; the Swiss National Science Foundation; and the European Union's Seventh Framework Programme for research, technological development and demonstration under grant agreement no 324318.

References

- [1] M. Shochet, L. Tompkins, V. Cavaliere, P. Giannetti, A. Annovi, and G. Volpi, *Fast TracKer (FTK) Technical Design Report*, Tech. Rep. CERN-LHCC-2013-007. ATLAS-TDR-021, CERN, Geneva, Jun, 2013. ATLAS Fast Tracker Technical Design Report.
- [2] R. S. Bartoldus, C. M. C. Bee, D. C. Francis, N. R. Gee, S. L. R. George, R. M. S. Hauser, R. R. Middleton, T. C. Pauly, O. K. Sasaki, D. O. Strom, R. R. I. Vari, and S. R. I. Veneziano, *Technical Design Report for the Phase-I Upgrade of the ATLAS TDAQ System*, Tech. Rep. CERN-LHCC-2013-018. ATLAS-TDR-023, CERN, Geneva, Sep, 2013.
- [3] A. Andreani, A. Andreatza, A. Annovi, M. Beretta, V. Bevacqua, et al., *The Fast Tracker real time processor and its impact on muon isolation, tau and b-jet online selections at ATLAS*, *IEEE Trans.Nucl.Sci.* **59** (2012) 348–357.

- [4] F. Alberti, A. Andreani, A. Annovi, M. Beretta, M. Citterio, F. Crescioli, M. Dell’Orso, P. Giannetti, A. Lanza, V. Liberali, D. Magalotti, C. Meroni, M. Piendibene, I. Sacco, A. Stabile, and G. Volpi, *Performance of the AMBFTK board for the FastTracker processor for the ATLAS detector upgrade*, *Journal of Instrumentation* **8** (2013), no. 01 C01040.
- [5] Cerri, A, *Towards a Level-1 tracking trigger for the ATLAS experiment*, in proceedings of this conference.
- [6] A. Annovi, G. Broccolo, A. Ciocci, P. Giannetti, F. Ligabue, D. Magalotti, A. Nappi, M. Dell’Orso, R. Dell’Orso, F. Palla, E. Pedreschi, M. Piendibene, L. Servoli, S. Taroni, and G. Volpi, *Associative Memory for L1 Track Triggering in LHC Environment*, *Nuclear Science, IEEE Transactions on* **60** (Oct, 2013) 3627–3632.