The Future Evolution of the Fast TracKer Processing Unit

C. Gentsos[1,6], F. Crescioli[2], F. Bertolucci[3], D. Magalotti[4], S. Citraro[*,3,5], K. Kordas[6], S. Nikolaidis[6]*

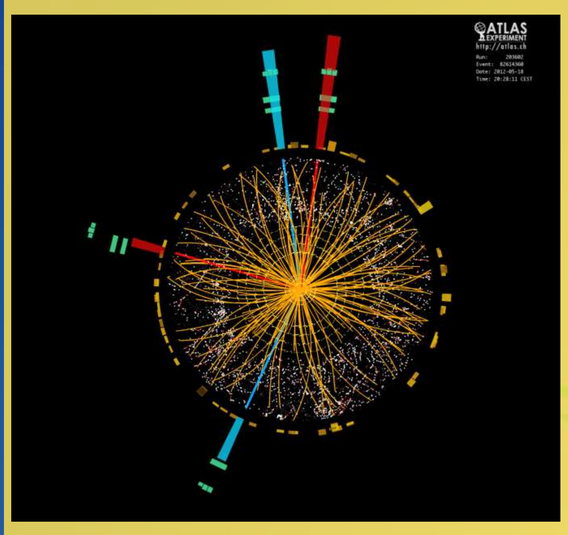
[1]Prisma Electronics S.A., [2]LPNHE CNRS, [3]University of Pisa,
[4]University of Modena and Reggio Emilia, [5]INFN-Pisa, [6]Aristotle University of Thessaloniki

[*]Presenter email: saverio.citraro@for.unipi.it

Abstract

Real time tracking is a key ingredient for online event selection at hadron colliders. The Silicon Vertex Tracker at the CDF experiment and the Fast Tracker (FTK) at ATLAS are two successful examples of the importance of dedicated hardware to reconstruct full events at hadron machines. We present the future evolution of this technology, for applications in the High Luminosity runs at the Large Hadron Collider (HL-LHC). Data processing speed is achieved with custom VLSI pattern recognition and linearized track fitting executed inside modern FPGAs, exploiting deep pipelining, extensive parallelism, and efficient use of available resources. In the current system, one large FPGA executes track fitting in full resolution inside low resolution candidate tracks found by a set of custom ASIC devices, called Associative Memories (AM chips). The FTK dual structure, based on the cooperation of VLSI AM and programmable FPGAs, is maintained, but we plan to increase the FPGA parallelism by associating one FPGA to each AM chip. Implementing the two devices in a single package would achieve further performance improvements, plus miniaturization and integration of the state of the art prototypes. We present the new architecture, the design of the FPGA logic performing all the complementary functions of the pattern matching inside the AM, the tests performed on hardware integrating the FPGA and the ASIC, the power consumption and performance running Monte Carlo events.

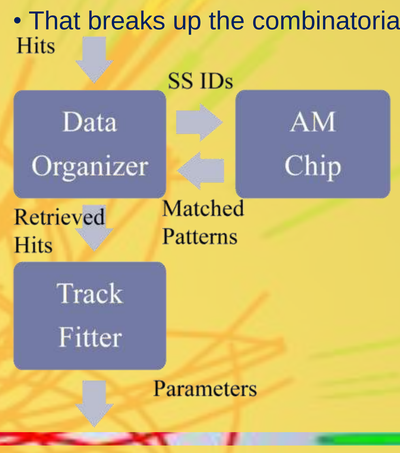
ATLAS Fast Tracker evolution



- The Fast TracKer has been successfully used to perform efficient online track reconstruction in ATLAS simulated events
- The tracking process is split in two steps, the first of which takes place in custom ASICs (AMChips), and the second in reconfigurable FPGA devices
- For future applications the system needs to be adapted to be able to reconstruct tracks within some μs
- To keep the latency to a minimum and at the same time increase the performance, we plan to increase the parallelism, while at the same time assigning one FPGA device to each AMChip and packaging them together
- At the same time, optimizations on the implementation of the functions complementary to the pattern matching are required, in order to deal with ~500KHz of events

Data flow

- In FTK the tracking process is split in two steps
- The AMChip first finds low resolution track candidates called “roads”
- Then, within each road, all real track candidates are examined using a localized linear fit and approved or rejected according to their computed χ^2 value.
- That breaks up the combinatorial problem by a huge factor



The main components that perform those functions are

- The AM Chip which does the low-resolution pattern matching, a separate ASIC designed for exactly that purpose
- The Data Organizer, which stores the full resolution hits according to their low-resolution representation
- Units that produce track-forming combinations of the hits
- The Track Fitter units, which perform very fast linear fits
- Everything except the actual pattern matching is implemented on one FPGA

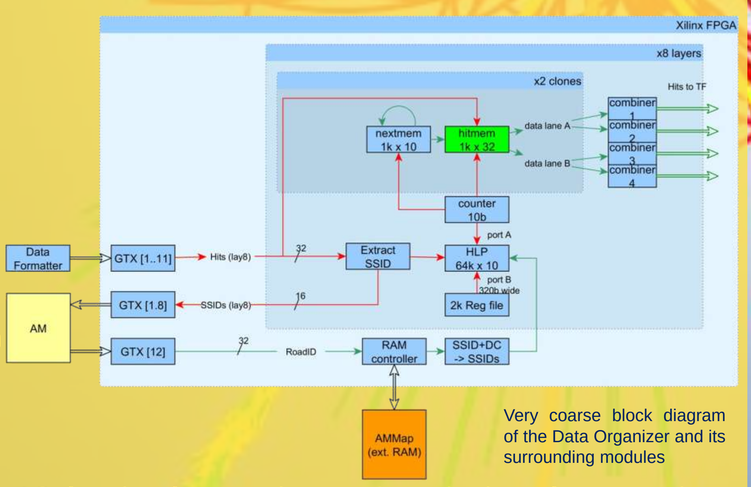
Data Organizer module

The Data Organizer architecture has to support the following features

- Store full resolution hits according to 16-bit SSIDs (coarse resolution coordinates for inner detector layers)
- Decode the AMChip output to SSIDs and Don’t Care bits (see below) per each layer
- Retrieve the full resolution hits based on that information

Also has to support

- AMChip Don’t Care function, a pattern can be associated to up to 8 SSIDs/layer
- No hard limit on the number of hits/SSID
- Deterministic performance
- Maintain a very high bandwidth, especially on the reading stage



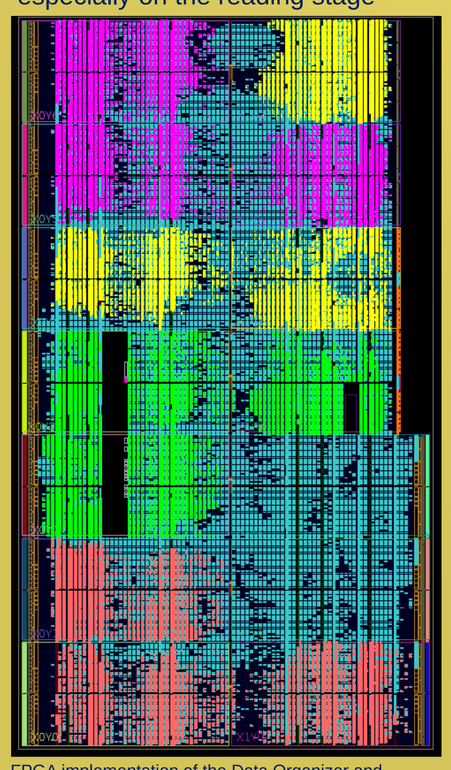
Very coarse block diagram of the Data Organizer and its surrounding modules

The DO function for one detector layer is that of a somewhat more complicated linked-list. The full resolution hits are written in a memory in the order in which they arrive. Then the Hit List Pointer (HLP) memory, whose every location corresponds to an SSID, is updated with the address of the last hit to arrive for that specific SSID. At the same time, another memory is updated with the invalidated information the HLP held, so the address of the last hit stored can point to the previous hit of the same SSID (if there is one).

To account for the case of a missing layer and support the Don’t Care function, for each event the system must know if data has been written in any SSID address possible. That is made possible with the use of a register file. To keep that register file small (2k instead of 64k), and in the same time boost read performance, the HLP memory has been widened so that each location covers 32 SSIDs. The impact on performance is that it is possible to read the possible hit locations of 8 SSIDs in one clock cycle, releasing the HLP to fetch the next road’s locations.

Moreover, the dual-port hit memory is duplicated, so there are 4 channels reading at the same time, providing the necessary memory bandwidth to support the fast parallelized HLP output.

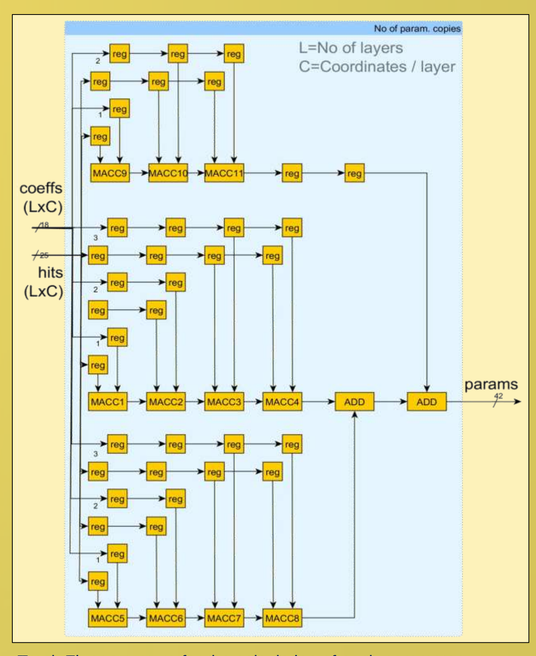
The operating frequency of the Data Organizer is 400MHz. When writing it can accept hits every 2 clock cycles, thus the maximum input rate is 200MHits/layer/s, and when reading data the maximum output rate is 1600MHits/layer/s.



FPGA implementation of the Data Organizer and 4 Track Fitting units (each unit is highlighted in color)

Track Fitter module

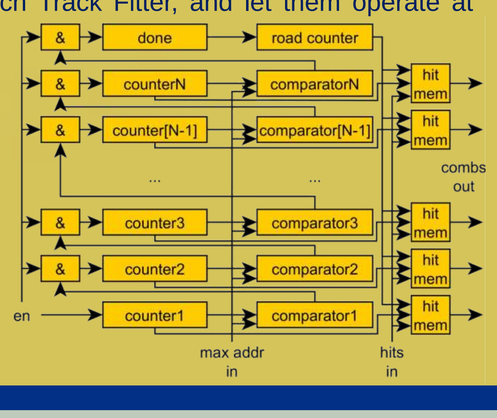
- The helix parameters are extracted from the local hit coordinates using linear functions
- The resolution of the fit is close to that of the full helical fit in a narrow sector of the detector
- Using dedicated DSP units found in modern FPGA devices very fast implementation can be realized
- We exploited parallelism and high operating frequency due to deep pipelining using the architecture shown
- The implementation strikes a balance between DSP usage and register utilization
- Each Track Fitting unit operating @500MHz can perform a fit every 2ns, after an initial latency of 90ns
- Four units can fit in a modern mid-grade device making the maximum performance of the system 2GFits/s
- That kind of performance can help in keeping the trigger efficiency up in high occupancy situations



Track Fitter structure for the calculation of each parameter

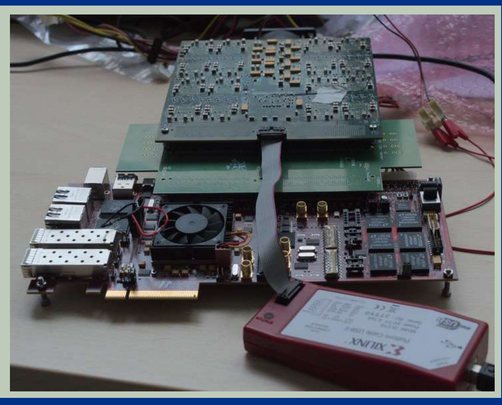
Combiner module

- Between the Data Organizer module and each Track Fitter unit, another type of module that computes all the possible combinations that form valid tracks is necessary
- The Combiner function seems simple but the realization in hardware has a long critical path, which keeps the frequency low
- The solution was to assign two combiner units for each Track Fitter, and let them operate at exactly half the frequency (250MHz)
- Each clock cycle, one combination is produced
- No additional latency between roads
- The hit memories are split into sectors, each one is filled with hits from a detected road
- The sectors are written asynchronously lin relation to the read operation, in a cyclic way
- FIFO-like flags are generated to prevent data from being overwritten
- Easy-to-adapt architecture



Results

A firmware has been designed, that makes efficient usage of the device resources to allow for a future-proof implementation, maintaining a significant performance overhead to allow for AMChip upgrades. Track fitting performance can exceed 2GFits/s, and the rest of the architecture (namely the Data Organizer and Combiner modules) can sustain that performance. The design has already been implemented on an FPGA device and is currently being integrated as a full system on a mezzanine PCB, to be seated on a PULSAR board. After the integration process, work has to be done to test the actual performance on the board using simulated events on real-time tests, and do the necessary simulation studies to evaluate the impact the increased performance will have on the achievable efficiency of the tracking system.



Acknowledgements

This work receives support from Istituto Nazionale di Fisica Nucleare; Grant-in-Aid for Scientific Research from the Japan Society for the Promotion of Science and MEXT, Japan; and the European community FP7 People grants ARTLHCFE 254410 FP7-PEOPLE-2009-IOF and FTK 324318 FP7-PEOPLE-2012-IAPP.