Available on CMS information server

CMS CR -2015/307

The Compact Muon Solenoid Experiment

Conference Report

Mailing address: CMS CERN, CH-1211 GENEVA 23, Switzerland

17 November 2015 (v2, 27 November 2015)

Track Finding in CMS for the Level-1 Trigger at the HL-LHC

Fabrizio Palla, Mark Pesaresi and Anders Ryd for the CMS Collaboration

Abstract

The High Luminosity LHC (HL-LHC) will deliver luminosities of up to 5×10^{34} Hz/cm^2, with an average of about 140 overlapping proton-proton collisions per bunch crossing. These extreme pileup conditions place stringent requirements on the trigger system to be able to cope with the resulting event rates. A key component of the CMS upgrade for HL-LHC is a track trigger system which would identify tracks with transverse momentum above 2 GeV/c already at the first-level trigger. This paper presents the status of proposals for implementing the L1 tracking in conjunction with the planned upgrade for the silicon tracker of the CMS experiment.

Presented at *TWEPP 2015 TWEPP 2015 - Topical Workshop on Electronics for Particle Physics*

Preprint typeset in JINST style - HYPER VERSION

Track Finding in CMS for the Level-1 Trigger at the HL-LHC

F. Palla[a], M. Pesaresi[b*] and A. Ryd[c] for the CMS Collaboration

[a]*INFN, Sezione di Pisa*
56127 Pisa, Italy
[b]*Blackett Laboratory, Imperial College,*
London SW7 2AZ, UK
[c]*Cornell University,*
Ithaca NY 14850, US
E-mail: mark.pesaresi@imperial.ac.uk

ABSTRACT: The High Luminosity LHC (HL-LHC) will deliver luminosities of up to 5×10^{34} Hz/cm^2, with an average of about 140 overlapping proton-proton collisions per bunch crossing. These extreme pileup conditions place stringent requirements on the trigger system to be able to cope with the resulting event rates. A key component of the CMS upgrade for HL-LHC is a track trigger system which would identify tracks with transverse momentum above 2 GeV/c already at the first-level trigger. This paper presents the status of proposals for implementing the L1 tracking in conjunction with the planned upgrade for the silicon tracker of the CMS experiment.

KEYWORDS: L1 Tracking, Track Finding, Track Trigger.

[*] Corresponding author.

Contents

1. Introduction

The High Luminosity LHC (HL-LHC) will deliver luminosities of up to 5×10^{34} Hz/cm^2, with an average of about 140 overlapping proton-proton collisions per 25 ns bunch crossing (BX). These extreme pileup conditions place stringent requirements on the trigger system to be able to cope with the resulting event rates. One of the goals of CMS for the high luminosity upgrade is to maintain the physics performance achieved during Run 1 in 2012. While the Level-1 (L1) trigger will be upgraded to provide a maximum trigger rate of around 750 kHz (compared to $<$ 100 kHz in Run1), even with this increase in rate the thresholds for basic objects (muons, electrons, jets etc.) will have to be tightened if no new information can be provided towards the L1 trigger making decision [1]. Therefore a L1 trigger that can make use of reconstructed data from the silicon strip tracker is desirable, thanks to its superior momentum and spatial resolution for charged particles.

In order to provide tracking information to the L1 trigger, the outer tracker will utilise a novel type of module that transmits hits consistent with charged particles above a chosen transverse momentum (p_T) threshold, at the full 40 MHz beam crossing rate. Since about 99% of all tracks in CMS are below $\sim$2 GeV/c, and are of no relevance to most practical triggers, this reduces the volume of data the trigger will receive down to O(20) Tbps. The basic concept [2, 3] is to compare the binary pattern of hit strips on upper and lower sensors of a two-layer module to reject patterns that are consistent with a low transverse momentum track. Hit combinations in the two sensors consistent with a high-p_T track segment are known as stubs.

The role of the tracking trigger is to deliver track objects to the L1 trigger within about 5 μs, in order to allow this information to be merged with that from other sub-detectors. Given that a 40 MHz silicon-based tracking trigger on the scale of the CMS detector has never been built, it is essential to demonstrate the feasibility of such system. This paper presents the status of three different approaches with state of the art technologies: two using a fully FPGA-based approach, and another using a combination of FPGAs and ASICs to perform real time pattern recognition. All three approaches provide full coverage of the Tracker detector, segmenting it into several trigger

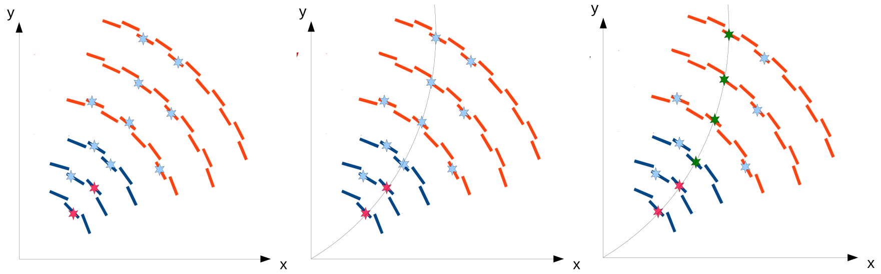

Figure 1: The steps of the tracklet algorithm are illustrated starting with a) where pairs of stubs in neighboring layers are combined to form the seeds, tracklets. In b) the tracklets are projected to the other layers and matching stubs are found. In the last step c) the matched hits are included in the final track fit.

towers in pseudo-rapidity and azimuth, optimized to provide full coverage and minimal amount of redundancy. In addition, all three approaches have a first stage of off-detector electronics to format the data and time-multiplex them out to a second stage of processor cards, over the course of the time-multiplexing period.

A more detailed description of the CMS detector, together with a definition of the coordinate system used and the relevant kinematic variables, can be found in Ref. [4].

2. FPGA Based Tracklet Approach

The tracklet based approach implements the L1 track finding using a traditional road search technique. This approach is commonly used in software based track finding algorithms. Here we are exploring how to implement this on modern FPGAs for applications in the L1 trigger.

Figure 1 schematically describes how the algorithm works. The seeds, tracklets, are formed from pairs of stubs in adjacent layers. The figure illustrates a seed formed between layers 1 and 2; but the algorithm in parallel forms seeds in other pairs of layers as well. We use the impact parameter (IP) in the r-ϕ plane as a constraint to calculate the helix parameters of the trajectory in both the r-ϕ and r-z planes. These parameters are used to project the trajectory to other layers. We then search for matches in the other layers. If a tracklet has two or more matching stubs, i.e., a total of at least 4 stubs, we perform a linearized χ^2-fit. The linearized χ^2-fit is very simple; we have a good seed from the initial tracklet and we have calculated the residuals from these seeds to the stubs in the matching stage and hence the final fit simply involves a set of multiply-and-add operations since the derivatives have been precomputed. Since we perform seeding in multiple layers in parallel we often find the same tracks multiple times and we need to remove the duplicate tracks.

The challenge is the implementation of the algorithm on an FPGA. We have a baseline project that we are working towards implementing. The main features are:

- The detector is divided into 28 sectors in ϕ, each of these sectors will be handled by one FPGA. We have a small data duplication between nearby sectors in order to be able to form the tracklets locally in a sector. However, tracklets that project to the neighboring sectors are sent there for matching with stubs and any matches are then returned to the sector where the tracklet was found for the final trackfit.

- We implement seeding in the following combinations of layers: L1+L2, L3+L4, and L5+L6; and the following combination of disks D1+D2 and D3+D4; as well as in the overlap region where we combine L1+D1 and L2+D1.

- We assume that the project has a time multiplexing of a factor of 4. However, we can increase this to a factor of 8 and stay within the latency requirement of about 4 μs.

We have a first implementation of the firmware for this baseline project. We initially focused on the barrel part; but the disk implementation is very similar and is being implemented now. In Fig. 2 is shown a high-level overview of the firmware for a smaller project that covers 1/4 of the barrel. The data arrive from the front end via the DTC (Data Trigger and Control) system on the right. The first two stages of processing routes the stubs by layer and organize them by their ϕ and z position into 'Virtual Modules', VMs, that correspond to smaller areas of the detector. Having organized the stubs into the VMs we then perform the combinatorics to look for stub pairs that are consistent with $p_{\mathrm{T}} > 2$ GeV and $|z_0| < 15$ cms. These consistency cuts are implemented at this stage as a fast lookup table. After the pairs of stubs are formed they are processed in the next step where the full tracklet parameters are evaluated and the projections to the other layers are performed. Projections to neighboring sectors are flagged and in the next step these are sent over serial optical links to the neighboring sectors. Similar to how the stubs are organized into VMs, the projections are routed into the corresponding ϕ and z regions; in the first step we form candidate matches based on approximate positions and in the second step we calculate the exact residuals between the projections and stubs. Matches made to projections from neighboring sectors are sent back to the 'home' sector of the tracklet for the linearized χ^2 fit in the last step.

The firmware for these steps have been implemented and tested. First we ran it on Xilinx VC709 evaluation boards. This allowed us to demonstrate the processing steps but did not allow the sector-to-sector communication. During the summer of 2015 we have established a test stand, Fig. 3, based on the CTP7 boards developed at the U. of Wisconsin for the CMS Phase-1 trigger upgrade. The test stand is fully operational. We have implemented the communication between the boards, using the link protocol developed for the Phase-1 trigger. With the VC709 boards we had not used the sector-to-sector communication, the focus recently has been to implement this functionality and demonstrate that the test stand produces results identical to our C++ emulation of the track finding.

We have the test stand operational now and have good agreement between the emulation and firmware for tracking in simple events without pileup (PU). The next steps involve using more complicated events with PU and to scale the project to a larger part of the detector than the 1/4 barrel sector currently used. The project for the 1/4 barrel sector is using 25% or less of the FPGA resources so we should be able to scale the project to a substantially larger fraction of a sector. With the Virtex 7 FPGA we are using today it is not likely that we can implement the full sector for the full baseline project. But with the additional resources available in newer generation FPGAs (Ultrascale+) we should easily be able to handle a full sector.

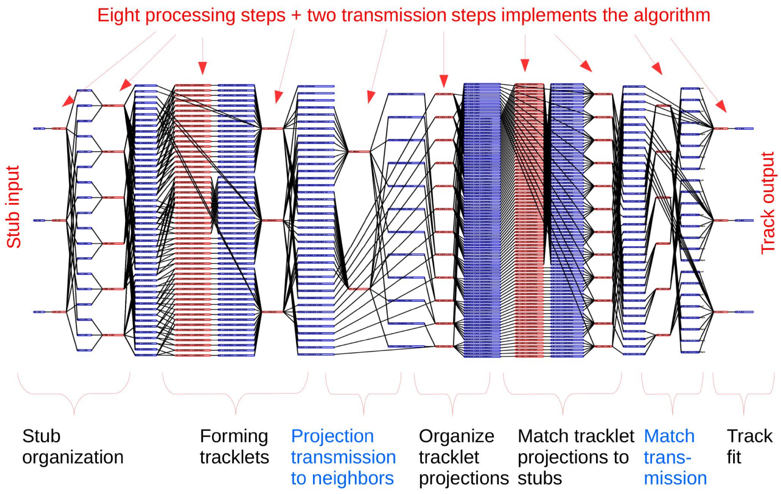
Eight processing steps + two transmission steps implements the algorithm
Stub input
Track output
Stub organization
Forming tracklets
Projection transmission to neighbors
Organize tracklet projections
Match tracklet projections to stubs
Match trans- mission
Track fit

Figure 2: The implementation of the tracklet algorithm in firmware consists of 10 processing steps, see text for more detail. In this diagram the stubs come in from the left and are processed through the 10 pipelined processing stages until we get the final tracks produced at the right. (This does not yet include the duplicate removal). Each processing step (red) is separated by a memory module (blue).

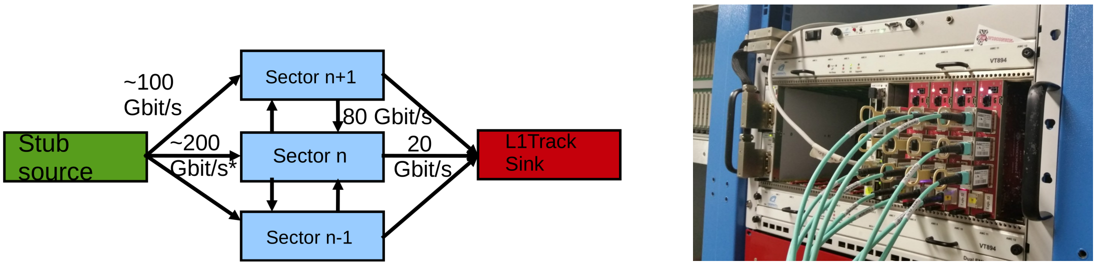

Figure 3: Left is a schematic view of the test stand. The goal is to demonstrate the full functionality of the central sector processing board. To the right is the implementation. We used four CTP7 boards; three are used for the sector boards and one of the CTP7 boards is used as the source of the stubs as well as for reading out the L1 tracks.

3. FPGA Based Projective Binning Approach

This approach makes use of the concept of the Hough Transform [5], a well known line detection algorithm, to coarsely group together stubs consistent with a high transverse momentum track into candidates, before a second stage fitting process. To simplify both the system architecture and the implementation of the Hough Transform track finder, a fully time multiplexed design using FPGA based hardware is considered. A fully time multiplexed design is advantageous for a number of reasons [6], particularly for the fact that a very small number of nodes is needed to demonstrate an entire trigger, since each processor is carrying out identical processing, delayed by one LHC clock cycle.

The track finder will require the Hough Transform to be applied in two, daisy-chained, two

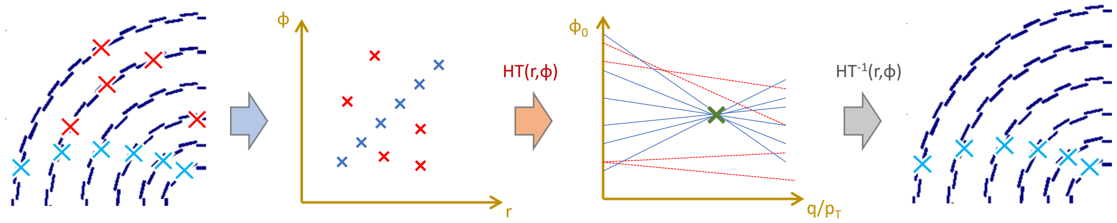

Figure 4: Illustration of the Hough Transform concept for identifying stubs consistent with a high transverse momentum track in the (r,ϕ) projection.

dimensional projections, for example (r,ϕ) and (r,z). Figure 4 illustrates the principle for the (r,ϕ) projection. As required in the non-generalised Hough Transform, high p_T tracks define straight lines in (r,ϕ) according to the equation $\phi = mr + \phi_0$, where m is a constant proportional to q/p_T of the track (where $q = \pm 1$), and ϕ_0 is the ϕ of the track at $r = 0$[1]. A stub with defined (r,ϕ) traces a line in Hough space (m,ϕ_0), indicating all possible 2D track parameters that are consistent with this point. Consequently a set of stubs that belong to a high p_T track will be represented as a set of lines in Hough space, and the intersection of these lines allows the parameters of the 2D track to be determined. By quantising Hough space into bins and histogramming the stubs according to the transform, one can group stubs consistent with a physical track as a single candidate, avoiding expensive combinatorial computations, and simultaneously extract an estimate of the track parameters $(q/p_T,\phi_0)$.

Typically the line parameters in Hough space are determined by a simple voting procedure in each bin. The Hough Transform track finder however can make use of additional information relating to the Tracker geometry to eliminate combinatorial background in high pileup events. Continuing with the example of the r-ϕ transform, one can impose a condition that, given a track will cross at least 6 barrel or disk layers, at least 5 stubs from unique radii are present, and that their local $\Delta\phi$ (coarse vector information provided by the Tracker) is consistent with the parameters of that bin.

Figure 5 provides an overview of a tracker division into regions and related system architecture, based on currently available technology [7]. A minimum of two layers of processing boards are required: at Layer-1, data from the tracker are received, buffered, formatted and sent to the HLT in the case of L1 triggered data, or time-multiplexed and forwarded to the Layer-2 track finder in the case of stubs; at Layer-2 stubs are grouped together into track candidates using the Hough Transform before passing through a final fitting stage. Since the Layer-1 processor and DTC can be the same physical board, the number of Layer-1 processors required is essentially determined by the number of links emerging from the tracker and fibre routing constraints within CMS. However if for reasons of practicality a separate DTC and Layer-1 stage is required then this could be easily accommodated at the expense of more boards. Segmenting the tracker into 5 independent trigger regions, for example in η, is sufficient to allow all data from a region to be processed by a single processor board provided a time multiplexing period of 24-36BX from Layer-1 to Layer-2.

A demonstrator system for this concept is underway, with the first test stand assembled and

[1] Here, the small angle approximation is used for high p_T tracks, although one could equally use an alternative projection.

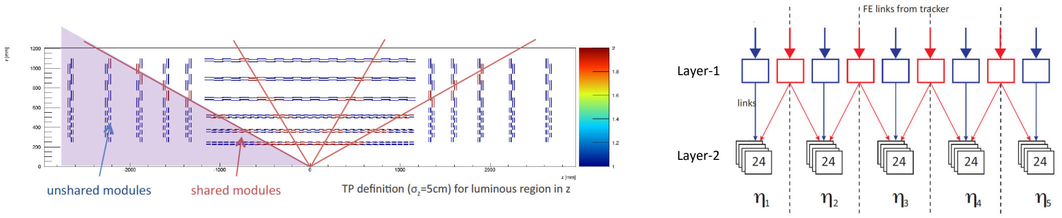

Figure 5: Left: Division of the tracker into 5 regions in η highlighting shared regions (red) and non-shared regions (blue); right: A schematic allocation of links and processors to the five sectors, separated by dashed vertical lines, for Layer-1 and Layer-2 where the Layer-2 processors are time multiplexed nodes. Note that the Layer-2 track finders act as independent processors on the data they receive, therefore no boundary sharing between regions or processors is required at this stage.

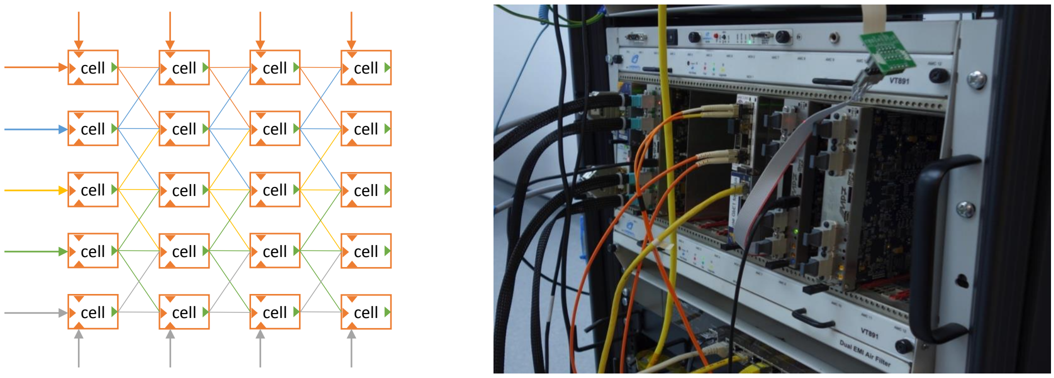

Figure 6: Left: Block diagram of the systolic array concept, where each cell is an intelligent storage element, able to pass data to the adjacent column; right: the demonstrator test stand.

implementing parts of the track finder. As explained earlier, a time multiplexed architecture can be demonstrated with only a limited fraction of the final hardware, since one Layer-2 processor is sufficient to emulate one time slice and runs the same algorithm as all other nodes (at least for the same η region), independently of all others. The hardware for the demonstrator system is based on the MP7 [7] μTCA card, which features a Xilinx Virtex-7 V690 FPGA and Avago MiniPOD optics providing a total optical bandwidth of 0.9 + 0.9 Tbps (transmit + receive) from 72 bi-directional links operating at up to 12.5 Gbps. It currently implements Layer-2 of the upgraded CMS L1 calorimeter trigger, which is also a time multiplexed system. The demonstrator envisaged is scaleable, allowing additional Layer-2 MP7 boards to be daisy-chained together if further processing power is required. A Layer-1 stage can also be implemented since one or two MP7s would have sufficient output links to emulate the data from an entire trigger region.

The demonstrator test stand, consisting of two MP7-XEs so far, allows us to test elements of the Hough Transform track finder in hardware. The system makes use of both the existing MP7 calorimeter trigger infrastructure and link firmware, and the online software and utilities designed during commissioning of the MP7 and trigger system to make the demonstrator accessible to users for testing and validating different algorithms simply and rapidly. One implementation of the Hough Transform algorithm in firmware, illustrated in Figure 6, is based on the concept of the pipelined systolic or self-filling array. Stubs are presented to the outside edge of the 2D array and are stored within the cell block RAM if they are compatible with the cell parameters in Hough space. Stubs are propagated to the next column (west to east), if within the bounds of the array, and the process is repeated. Buffering and selection logic is required to process stubs coming from multiple sources in a cell (north-west, west, south-west). Additional filters can be implemented in these intelligent cells to reduce the number of stubs binned per cell. A cell is marked for readout if at least 5 stubs from unique radii have been stored. In the current non-optimised implementation, each cell requires 4 DSPs, 2 block RAMs and 2k LUTs, although we expect that this can be reduced further.

At present a reduced r-ϕ Hough array of 50 cells is implemented in the demonstrator for validation against the C++ emulation of the track finder using simple events in Monte Carlo without pileup. The events are injected into the MP7 input link buffers and candidates collected on the output link buffers. Once the algorithm is validated in hardware, more complex events can be loaded at the input while the array can also be expanded in size to increase the granularity and therefore performance of the Hough Transform algorithm. Simulations suggest that an array approximately 30×30 in dimension could identify track candidates with high efficiency under conditions of 140 interactions per BX.

4. Associative Memory (AM) Plus FPGA Based Approach

The L1 AM + FPGA approach makes use of a two-stage procedure. The detector is subdivided into $8(\phi)\times6(\eta)$ trigger sectors. In a first stage, data are formatted with coarser resolution than the detector pitch, and the pattern recognition is performed by matching compatible sequences of low resolution stubs in 6 detector layers with pre-computed pattern banks, residing in the AM chips. The AM is a massively parallel system in that each hit is compared with all patterns almost simultaneously. The high-resolution stub data belonging to the matched pattern are then retrieved and a second step of track reconstruction is performed on a FPGA, thus dealing with a smaller combinatorial problem. In events with 140 pile-up plus 4 top quarks, each sector receives on average about 300 stubs, divided into 6 detector layers. The number of patterns required in each sector depends on the stub resolution and the minimum p_T of the track trigger tracks: coarser stub resolution as well as higher p_T threshold lower the number of needed patterns, but complicates the subsequent step of track finding in the FPGA, which in turn affects the latency of the trigger algorithm and the purity of the tracks. Viceversa, too precise stub information increases the number of patterns required, thus resulting in an unpractical number of AM chips. These parameters are now under a global optimization, and preliminary indications result in about 1-2 Million patterns for a p_T threshold of 2 GeV/c.

The filtered stubs from the matched patterns are then available for track reconstruction. The first stage is the selection of the combinations of stubs compatible to come from the same track. This can be accomplished in several ways: from selecting the combinations below a given chi-square cut, adding stub-bending compatibility, or by forming track seeds in the innermost pixellated modules and then selecting the stubs compatible with its extrapolation. The final fit is performed

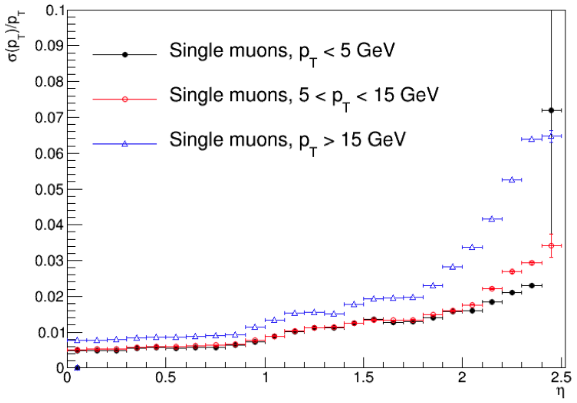

Figure 7: Relative transverse momentum (p_T) resolution as a function of the track pseudorapidity for single muons in several p_T bins in simulated events as obtained from the PCA track fit.

by a Principal Component Analysis (PCA) algorithm over a set of narrow regions of the detector, and in several p_T bins, where the track parameters are linear in the local stub coordinates, through a series of constant terms, unique in each one of these regions. About 20,000 constants are needed for the entire tracker. Figure 7 shows the precision for the p_T as a function of the pseudorapidity for single muons, in the simulation.

In order to demonstrate the concept, a vigorous R&D has started with the aim to build a Vertical Slice Demonstration System, making use of state of the art technologies. This system will comprise a full tracking trigger path and will be used with simulated high-luminosity data to measure trigger latency and efficiency, to study overall system performance, and to identify potential bottlenecks and appropriate solutions.

The architecture under study is based on Advanced TCA with a full-mesh backplane as shown in Figure 8. The large inter-board communication bandwidth provided by the full-mesh backplane is used to time multiplex the high volume of incoming data in such a way that the I/O bandwidth demands are manageable at the board and chip level, making it possible for an early technical demonstration with existing technology. The resulting architecture is scalable, flexible and open. The data coming from the modules of a given sector are processed by a series of Pulsar II boards [8], developed at FNAL, each capable of handling about 10% of the modules, and formatted and routed to a target node (another Pulsar) which is processing the actual event. The data of the second event are received by the same boards, but are routed to a different target node, in a time-multiplexing fashion. Each Pulsar board hosts 4 FMC mezzanines equipped with AM chips and an FPGA performing the pattern recognition and track fitting of a given particular event. Depending upon the number of mezzanines in each Pulsar, each Pulsar can effectively accept several concurrent events multiplexed into several mezzanine boards.

There are currently two different types of mezzanine boards under development: a first one developed by INFN is a 14.9×14.9 cm^2 card hosting two FMC connectors designed to satisfy the VITA 57.1 standard specification, a Kintex 7 FPGA (XC7K355T) and 16 AM chips, which have been produced for the ATLAS FTK [9, 10]. The board has been first developed to host AM05

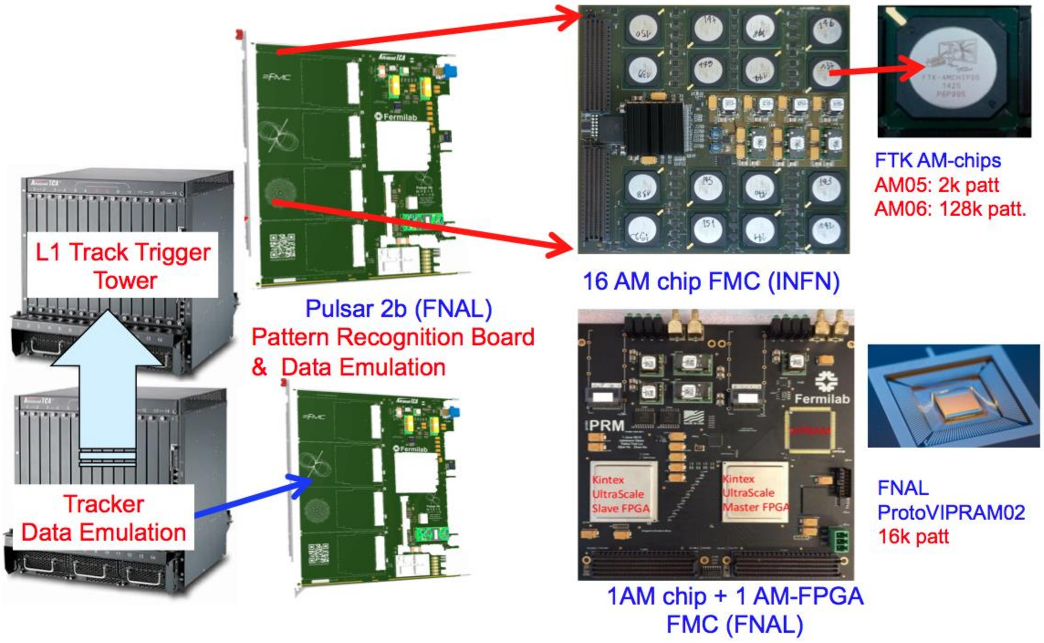

Figure 8: Demonstrator architecture.

chips, to test the basic functionality of the serial links, even if the number of patterns stored in each chip will be modest (2,048). A second version will use a newer AM chip version (AM06) where the number of patterns per chip will be increased to ∼130k. The FPGA has the role of distributing the hits to AM chips, collecting the candidate tracks, performing a track fitting on them using different algorithms and then send out the results to the following trigger level. The direct connection between the FPGA and the AM chips has been chosen to eliminate daisy chains and to reduce as much as possible the latency of the Level-1 trigger decision and to integrate all the functionalities in one single board. The most important test was the measurement of the eye-diagrams of the serial links between the FPGA and the AM chips, those between the FPGA and the FMC connectors, as well as the integration with the Pulsar board. The serial links between the FPGA and an external Ultrascale evaluation board have been tested using an IBERT tool by Xilinx. The signal integrity has been checked up to 8 Gbps using an Ultrascale evaluation board. The serial links between the FPGA and the 16 AM chips were tested using a serial link data analyzer sending PRBS-7 sequence on the links. The up links (see Figure 9) and the down links were successfully tested. The links showed no errors in 8 hours continuous running, leading to a Bit Error Rate (BER) of less than 10E-14. The mezzanine has passed all integration tests with the Pulsar board, including electrical, as well as LVDS and Xilinx GTH-GTX high-speed links.

A second mezzanine has been developed by FNAL, also with two FMC connectors, but with two Kintex UltraScale KU040 FPGAs, and a vertically integrated AM chip (VIPRAM). One FPGA is used as master FPGA, identical to the INFN mezzanine. The second FPGA instead is used to implement an associative memory, despite the lower pattern density compared to an ASIC. This allows easy modification and reconfiguration of the design, providing an ideal R&D test bench to develop the specification of the associative memory for a CMS L1 tracking trigger application. The

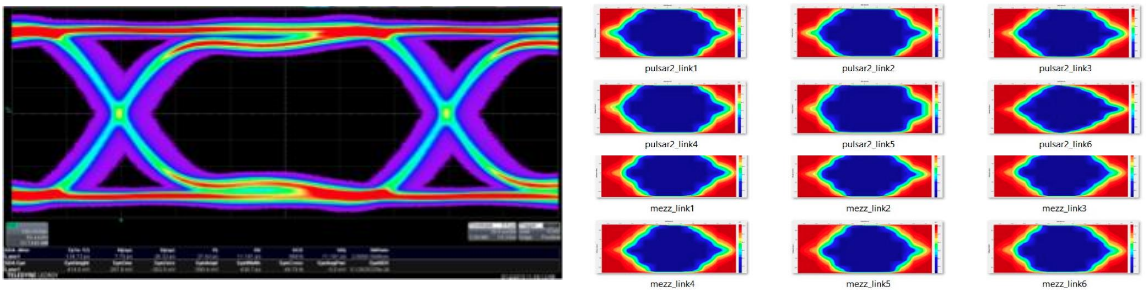

Figure 9: Left: AM05 to FPGA link at 2 Gbps; and right: Pulsar-mezzanine links at 10 Gbps.

initial tests show excellent performance, reaching a BER $<$ 10E-14 at 10 Gbps between the Pulsar and the mezzanine, and more than 15 Gbps open eye-diagrams between the inter-FPGA bus.

5. Summary

CMS is undertaking a programme of R&D to investigate the feasibility of a hardware based track finder for providing useable tracking information to the L1 trigger at HL-LHC. System slices and test benches based on the three parallel concepts described here are being assembled to demonstrate track finding using technology and hardware available today, with the aim of demonstrating at least one viable option by the end of 2016. These demonstrator systems will allow evaluation of track finder performance, including the total latency required, answer questions of scalability and cost, and help understand the impact on the final L1 trigger design.

Acknowledgments

We would like to acknowledge the U.K. Science and Technology Facilities Council (STFC) for funding contributions to this research. We also gratefully acknowledge funding from the U.S. DOE and U.S. NSF (grants NSF-PHY-1306801 and NSF-PHY-1307256), the European Union's seventh Framework Program for Research, Technological Development and Demonstration under Grant agreement no. 31744 (INFIERI), and from the Progetto PRIN MIUR DM 28.12.2012 n.957 (H-TEAM).

This paper is presented on behalf of the CMS Tracker Collaboration.

References

[1] CMS Collaboration, *Technical Proposal for the Phase-II Upgrade of the CMS Detector*, *CERN-LHCC-2015-010* (2015).

[2] J. Jones, G. Hall, C. Foudas and A. Rose, *A Pixel Detector for Level-1 Triggering at SLHC*, 11th Workshop on Electronics for LHC and Future Experiments, Heidelberg, *physics/0510228* (2005).

[3] M. Pesaresi and G. Hall, *Simulating the performance of a pT tracking trigger for CMS*, 2010 *JINST* **5** C08003.

[4] CMS Collaboration, *The CMS experiment at the CERN LHC*, 2008 *JINST* **3** S08004.

[5] R. O. Duda and P. E. Hart, *Use of the Hough Transformation to Detect Lines and Curves in Pictures*, Comm. ACM, **15** (1972) pp. 11-15.

[6] G. Hall et al., *A time-multiplexed track-trigger for the CMS HL-LHC upgrade*, Nucl. Instrum. Meth. A, *doi:10.1016/j.nima.2015.09.075* (2015).

[7] K. Compton, A. Rose et al., *The MP7 and CTP-6: multi-hundred Gbps processing boards for calorimeter trigger upgrades at CMS*, 2012 *JINST* **7** C12024.

[8] J. Olsen et al., *A full mesh ATCA-based general purpose data processing board*, 2014 *JINST* **9** C01041.

[9] 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. Nucl. Sci., **59** *doi:10.1109/TNS.2011.2179670* (2012) pp. 348-357.

[10] A. Andreani et al., *The Associative Memory Serial Link Processor for the Fast TracKer (FTK) at ATLAS*, 2014 *JINST* **9** C11006.