The Compact Muon Solenoid Experiment

Conference Report

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

20 July 2014 (v2, 24 July 2014)

Implementation of FPGA-based Level-1 Tracking at CMS for the HL-LHC

Jorge Enrique Chaves for the CMS Collaboration

Abstract

A new approach for track reconstruction is presented to be used in the all-hardware first level of the CMS trigger. The application of the approach is intended for the upgraded all-silicon tracker, which is to be installed for the High Luminosity era of the LHC (HL-LHC). The upgraded LHC machine is expected to deliver a luminosity on the order of $5 \times 10^{34} \text{cm}^{-2} \text{s}^{-1}$. This expected luminosity means there would be about 125 pileup events in each bunch crossing at a frequency of 40 MHz. To keep the CMS trigger rate at a manageable level under these conditions, it is necessary to make quick decisions on the events that will be processed. The timing estimates for the algorithm are expected to be below $5 \mu\text{s}$, well within the requirements of the L1 trigger at CMS for track identification. The algorithm is integer-based, allowing it to be implemented on an FPGA. Currently we are working on a demonstrator hardware implementation using a Xilinx Virtex 6 FPGA. Results from simulations in C++ and Verilog are presented to show the algorithm performance in terms of data throughput and parameter resolution.

Presented at WIT2014 WIT2014 Workshop on Intelligent Trackers

Implementation of FPGA-based Level-1 Tracking at CMS for the HL-LHC

J. Chaves

*Cornell University,
245 East Avenue, Ithaca, NY, 14853, U.S.A.
E-mail: jec429@cornell.edu*

ABSTRACT: A new approach for track reconstruction is presented to be used in the all-hardware first level of the CMS trigger. The application of the approach is intended for the upgraded all-silicon tracker, which is to be installed for the High Luminosity era of the LHC (HL-LHC). The upgraded LHC machine is expected to deliver a luminosity on the order of $5 \times 10^{34} \text{cm}^{-2} \text{s}^{-1}$. This expected luminosity means there would be about 125 pileup events in each bunch crossing at a frequency of 40 MHz. To keep the CMS trigger rate at a manageable level under these conditions, it is necessary to make quick decisions on the events that will be processed. The timing estimates for the algorithm are expected to be below $5 \mu\text{s}$, well within the requirements of the L1 trigger at CMS for track identification. The algorithm is integer-based, allowing it to be implemented on an FPGA. Currently we are working on a demonstrator hardware implementation using a Xilinx Virtex 6 FPGA. Results from simulations in C++ and Verilog are presented to show the algorithm performance in terms of data throughput and parameter resolution.

KEYWORDS: Pattern recognition, cluster finding, calibration and fitting methods; Performance of High Energy Physics Detectors; Simulation methods and programs; Data reduction methods.

Contents

1. Tracklet Algorithm	1
1.1 Algorithm overview	2
1.2 Algorithm timing	2
1.3 System architecture	3
2. Algorithm performance	3
2.1 Occupancy	3
2.2 Parameter resolution	5
3. Hardware implementation	5
3.1 System scale	6
4. Summary	7

1. Tracklet Algorithm

Over the next decade, the LHC accelerator complex, as well as the detectors, will be going through major upgrades to increase the instantaneous luminosity to $5 \times 10^{34} \text{ cm}^{-2}\text{s}^{-1}$. These upgrades will lead to the era of the High-Luminosity LHC (HL-LHC), where the CMS detector is expected to collect the majority of 3000 fb^{-1} of data. The data collected will be used to study the properties of the newly discovered Higgs boson and for other Standard Model precision measurements. It will also allow to probe yet unexplored areas of phase space in searches for new physics. At the proposed luminosity, the rates for muons, electrons, and jets exceed the front-end capabilities and increasing the trigger thresholds no longer reduce the trigger rates. In particular, muons with low energies are poorly measured at the time of the level 1 decision to have higher transverse momentum and pass the trigger. In order to keep the rates at an acceptable level at the upgraded LHC, we must incorporate tracking information at the level-1 trigger.

By the time of the scheduled Long Shutdown 3 for the LHC, the tracker will be at the end of its useful life. The proposed replacement for the tracker consists of four inner pixel layers and ten pixel disks that will cover a region of pseudo-rapidity up to 4. The pixels are not used in our approach for level-1 tracking. There are also six layers of modules in the outer barrel and five pairs of disks in the end-cap regions. The three inner layers of modules are made up of Pixel-Strip sensors while the outer layers are Strip-Strip sensors [1]. These double stack modules form pairs of hits called 'stubs'. The stub formation is the first level of selection. As a track bends in the magnetic field of the CMS detector, it deposits hits on the double stack modules. The positions of the hits on the sensors can be correlated to provide a coarse estimate of the track's transverse momentum. To reduce the data input, the stubs must be consistent with tracks with p_T of at least 2 GeV. This coarse requirement selects approximately 10,000 stubs from nearly 600,000 hits.

1.1 Algorithm overview

The approach presented here is similar to offline methods for track reconstruction used by the CMS and ATLAS experiments [2] [3]. With advances in integrated circuit technology, these complex algorithms can now be used in an online environment such as the level-1 trigger at the CMS detector. The algorithm is divided into sequential steps that allow for parallelism in the track processing. The track finding is seeded by pairs of stubs from adjacent layers in the tracker that are combined to form 'tracklets'. Then, using the detector origin as a constraint in the $r - \phi$ plane, we calculate an initial estimate of the track parameters for the tracklet. We select tracklets with p_T of at least 2 GeV and a longitudinal impact parameter of $|z_0| < 15$ cm. The tracklets are then projected to other layers using a uniform magnetic field (both outside-in and inside-out), depending on the layers used for seeding. For example, if a track was seeded in layers 1 and 2, we would project to layers 3, 4, 5, and 6. Instead if the track was seeded in layers 3 and 4, we would project in to layers 1 and 2 and project out to layers 5 and 6. Then we look in the projected layers for stubs consistent with the trajectory of a high- p_T track. If a stub is found within a given window of the expected track projection, we include it in the track candidate and store this difference in position. Along with precalculated derivatives and the differences from the previous step, we perform a linearized χ^2 fit to correct the initial track parameters. Since we seed in multiple pairs of layers, a track can be reconstructed more than once. In the last step we remove the duplicates found from the multiple seeding. Finally we pass the tracks to the global trigger. The global trigger can now associate the level-1 tracks to muons, electrons, and jets and make a better decision for which events to keep.

1.2 Algorithm timing

The space available to store each event in a buffer limits the time allowed for the trigger decision. With the bunch crossing rate of the LHC of 40 MHz, the level-1 trigger decision must be made in less than 10 μ s. Thus the track reconstruction must be made in approximately 5 μ s. This is to account for time needed by the global trigger to associate tracks to the other physics objects and make a decision. To increase the time available for processing, we consider time multiplexing the system by a factor of four. Each copy of the system receives a new event every 100 ns. The hardware implementation of the algorithm can be done in eight steps for a total of 800 ns. These steps are:

1. Sort the input stubs by their corresponding layer. Each stub stores the coarse estimate of the p_T and the coordinates in r , ϕ , and z ¹.
2. Sort the stubs into bins of z and ϕ and store a reduced version of the data. The reduced version of the data only contains a few of the most significant bits for the coordinates of the stub.
3. Select possible tracklets from allowed stub pairs. A lookup table is used to check for consistency of the stubs with a high p_T track coming from the origin.

¹The CMS coordinate system is defined with the positive x -axis in the direction of the center of the LHC ring, the y -axis is directed upwards, and the z -axis is in the direction of the beam. In the xy plane, the azimuthal angle ϕ is measured from the x -axis and r corresponds to the radial coordinate in this plane.

4. For the tracklets selected in the previous step, calculate an initial estimate of their parameters and their position in the extrapolated layers.
5. The projections are routed into bin of z and ϕ as in step 2.
6. Find matching stubs to the projected tracklet with the reduced versions.
7. Calculate the difference in position between the stubs and the projected tracklet.
8. Calculate the corrections to the initial estimates of the track parameters using the differences obtained in the previous step and lookup tables for precalculated derivatives.

Every 100 ns a new event is received and the previous event moves onto the next step. We are then processing a different event in each step of the algorithm. The number of objects that can be processed will depend on the specific clock speed that can be achieved by the hardware used.

1.3 System architecture

We divide the detector into 28 sectors in the $r - \phi$ plane (Figure 1), such that all tracks with $p_T > 2$ GeV are fully contained in at most two sectors. This reduces the need for large data transfers between sectors to more than the two nearest neighbors. We also divide the barrel into four z regions, which will be associated with the input links from the detector. We define a 'virtual module' as a subdivision of a sector in bins of z and ϕ . Each z region is subdivided into two virtual modules for even and odd layers. The odd layers are divided into three virtual modules in ϕ while the even layers are divided into four. In the even layers, the first and last virtual modules are shared between adjacent sectors. The subdivision into virtual modules has been optimized so that a stub in a virtual module can only form a tracklet with stubs in two virtual modules in the neighboring layer. In Figure 2 we can see an example of a high momentum track ($p_T > 2$ GeV) that is seeded from virtual modules 2 in the inner layer and 3 in the outer layer. We also see that a low energy track would be seeded from modules 2 and 4 and thus not consider this combination of virtual modules. Subdividing ϕ and z into virtual modules reduces the number of possible tracklet combinations even before any processing is done.

2. Algorithm performance

2.1 Occupancy

In a minimum-bias event, the large majority of the tracks in a given event seen at CMS comes from very soft interactions. There is already a requirement on the input stubs to be consistent with tracks with p_T greater than 2 GeV. Most of the stubs that pass this initial requirement come from combinatorics and do not correspond to real stubs. There are still on average 65 and 55 stubs in a ϕ sector in the innermost two layers respectively as seen in Figure 3 (left). As a result, we would need to process on average $65 \times 55 \approx 3600$ stub pairs per sector with seeding in just the innermost layers. Most of these are fake combinations and there would be many more if we counted the seeding in other layers. Figure 3 (right) shows the average number of stubs in a virtual module for the two innermost layers. When we consider only allowed combinations from the

Figure 1. $r - \phi$ schematic view of the upgraded CMS tracker. We divide it into 28 sectors so that a bending track with $p_T \geq 2$ GeV will be contained in at most 2 sectors. The inner layers (blue) are made up of PS modules, while the outer layers (red) are made up of 2S modules.

Figure 2. Virtual module subdivision of a ϕ sector. A track (green) with $p_T > 2$ GeV is shown to be seeded from stubs in virtual modules 2 and 3 in the inner and outer layers. A lower energy track (red) has stubs in modules 2 and 4, therefore we do not consider this pair for reconstruction.

virtual modules, a stub pair is known as a tracklet candidate. Figure 4 (left) shows the average number of tracklet candidates per virtual module pair. By imposing requirements on the tracklet candidates of $p_T > 2$ GeV and $|z_0| < 15$ cm, we can further improve the tracklet selection. Figure 4 (right) shows the average number of tracklets per layer. These studies have been done using Monte Carlo simulation of single muon events with a pileup of 140. The muons are uniformly distributed in the transverse momentum range $0.2 < p_T < 100$ GeV, covering the pseudorapidity range $|\eta| < 3.0$.

Figure 3. (left) Average number of stubs in the two innermost layers for a ϕ sector. (right) Average number of stubs in the two innermost virtual modules.

Figure 4. (left) Average number of tracklet candidates per virtual module pair for layers 1 and 2. (right) Average number of tracklets in an event in layers 1 and 2.

2.2 Parameter resolution

We estimate the resolution of the track parameters using a version of the algorithm implemented in a separate C++ code with both floating-point and integer precision. Figure 5 shows the resolution of the track parameters from the floating point and integer implementations. The parameter resolution is calculated as the difference between the generated values from the Monte Carlo event simulations and the results of the algorithm.

Figure 5. Resolution of the track parameters in emulation with floating point precision compared with integer precision.

3. Hardware implementation

The algorithm presented here can be implemented in commercial hardware, taking advantage of the flexibility offered by Field Programmable Gate Arrays (FPGAs). We have simulated the algorithm in Verilog as the first step towards a complete hardware implementation. The implemented simulation reproduces the same results as the C++ integer version. We are using the Gigabit Link Interface Board (GLIB) [4] as a testing system, as it provides a relatively easy interface through the use of IPBus. This board is based on a Xilinx Virtex 6. Currently we are in the process of porting the algorithm to the FPGA and we see the expected results so far for the calculations in the algorithm.

3.1 System scale

With the available resources in current FPGAs and certainly with the technology that will come before the HL-LHC era, we think that with a small number of FPGAs we can process an entire ϕ sector. These chips will be placed on the same board to minimize the need for board-to-board communication. Accounting for the factor of four time-multiplexing, we believe the whole system could be built with approximately 100 boards. We hope to benefit from experience already gained by other systems in CMS, such as the trigger or the DAQ.

4. Summary

Including tracking at the level-1 trigger is a requirement for the CMS experiment at the HL-LHC in order to deal with the challenging environment at the increased luminosity. We have presented a possible approach for tracking based on seeding tracklets and extrapolating to other layers where we look for matching hits. Simulations have shown that this method is viable for an integer implementation using commercial hardware. The work is ongoing for a slice test using the GLIB, which will provide a more realistic estimate for the processing time as well as the resource usage.

Acknowledgments

I would like to thank the CMS collaboration and contributors to this project at Cornell University for their advise preparing this presentation and these proceedings. The work presented here is partly funded by the National Science Foundation Graduate Research Fellowship Program as well as the Cornell University grant (NSF-PHY-1307256).

References

- [1] D. Abbaneo and A. Marchioro, *A hybrid module architecture for a prompt momentum discriminating tracker at HL-LHC*, 2012 *JINST* **7** C09001.
- [2] CMS Collaboration, *Tracking and Vertexing Results from First Collisions*, CMS PAS TRK-10-001.
- [3] ATLAS Collaboration, *Performance of the ATLAS Inner Detector Track and Vertex Reconstruction in the High Pile-Up LHC Environment*, ATLAS-CONF-2012-042.
- [4] P. Vichoudis et al., *The Gigabit Link Interface Board (GLIB), a flexible system for the evaluation and use of GBT-based optical links*, 2010 *JINST* **5** C11007.