PROCEEDINGS OF SCIENCE

Upgrade of the CMS Barrel Muon Track Finder for HL-LHC featuring a Kalman Filter algorithm and an ATCA Host Processor with Ultrascale+ FPGAs

M. Bachtis[a] ,C. Foudas[b], P. Katsoulis[b], T. Lam[a], S. Mallios[b],G. Karathanasis[c]*, I. Papavergou[c], S. Regnard[a], M. Tepper[a], P. Sphicas[c,d], C. Vellidis[c]

[a] *University of California, Los Angeles*
[b] *University of Ioannina, Greece*
[c] *University of Athens, Greece*
[d] *European organization of Nuclear Research (CERN)*

On behalf of the CMS Collaboration

The Barrel Muon Track finder of the CMS experiment at the Large Hadron Collider uses custom processors to identify muons and measure their momenta in the central region of the CMS detector. An upgrade of the L1 tracking algorithm is presented, featuring a Kalman Filter in FPGAs, implemented using High Level Synthesis tools. The matrix operations are mapped to the DSP cores reducing resource utilization to a level that allows the new algorithm to fit in the same FPGA as the legacy one, thus enabling studies during nominal CMS data taking. The algorithm performance has been verified in CMS collisions during 2018 operations. The algorithm is also proposed for standalone muon tracking at the High Luminosity LHC. The algorithm development is complemented by ATCA processor R&D featuring a large ZYNQ Ultrascale+ SoC with high speed optical links.

Topical Workshop on Electronics for Particle Physics (TWEPP2018)
17-21 September 2018
Antwerp, Belgium

*Speaker.

1. Introduction

The Barrel Muon Track Finder (BMTF) of the CMS [1] experiment at the LHC consists of 12 custom MP7 [3] processors featuring XILINX Virtex 7-690T FPGAs and is responsible for identifying tracks originating from muons and measuring their transverse momenta with a precision of about 10% in the central (barrel) region of the detector. Each BMTF processor uses 10 Gbps optical links to receive detector data in the form of muon stubs from the TwinMux [2] processor. Each muon stub, combining information from drift tube (DT) chambers and resistive plate chambers (RPC), provides 22 bits of coordinate information in the bending $r-\phi$ plane (azimuthal angle ϕ and bending angle ϕ_b), 7 bits of information in the $r-\theta$ plane, and quality bits. The momentum assignment is performed using lookup tables (LUT) implemented with Block RAM (BRAM). Due to the limited address space provided by Block RAM, the momentum assignment is performed using information from only two stations. In addition, the LUTs are filled assuming that the track originates from the center of the detector. This beamspot constraint improves momentum resolution since it effectively adds one more point in the center exploiting the full bending power of the CMS solenoid.

In High Luminosity LHC (HL-LHC), the DT electronics will be upgraded, providing a drift time resolution of 2 ns. Possible improvements in track reconstruction are also studied. The first goal is to improve momentum resolution by including information from more than two stations in the fit.Such improvement is not possible with the memory lookup approach since it would require lookup tables with address space of 88 bits. The second goal is to implement momentum assignment without the beamspot constraint. This measurement is motivated by physics searches for displaced particles. To achieve both goals, a new tracking paradigm is implemented exploiting a Kalman Filter algorithm [4] in an FPGA.

2. A Kalman Filter for the Barrel Muon Track Finder

Kalman Filtering (KF) is a technique used very widely in track reconstruction in hadron colliders. Recently a Kalman filter has been implemented for the CMS HL-LHC track trigger[5] with excellent performance but with a latency much larger than 250 ns which is the limit for the current muon trigger. This section describes a KF implementation tuned especially for the barrel muon trigger and optimized for low latency. The basic steps of a Kalman filter are described here. The full implementation of a Kalman Filter for tracking can be found elsewhere [4].

The track parameters at each detector station are given by the state vector $x_n = (k, \phi, \phi_b)$, where $k = q/p_T$. A track is seeded by a stub in the outer available station, and the track parameters and their uncertainties are propagated inwards. After neglecting the energy loss in the muon system, the new state and the old station are related as following:

$$\begin{pmatrix} k \\ \phi \\ \phi_b \end{pmatrix}_{n+1} = \begin{pmatrix} 1 & 0 & 0 \\ a & 1 & b \\ c & 0 & d \end{pmatrix} \begin{pmatrix} k \\ \phi \\ \phi_b \end{pmatrix}_{n}, \tag{2.1}$$

where the propagation matrix F consisting of parameters (a, b, c, d) is derived by the detector geometry and simulation. The state uncertainties expressed by a covariance matrix P are also propagated to the next station by the transformation $P_{n+1} = FP_nF^T + Q$ where Q is an additional covariance matrix corresponding to multiple scattering in the return yoke between layers. After propagating the state to the new layer, the closest stub is selected and the track parameters are updated based on the values and uncertainties of the measurement. Each measurement consists of a vector $z_k = (\phi, \phi_B)$ and the corresponding uncertainty is expressed by a 2x2 covariance matrix R, corresponding to the position resolution of the detector.

The state update is a matrix manipulation process involving matrix inversion. The result of this matrix algebra [4] is a Kalman gain matrix G that can be used to update the state as following:

$$x_n = x_n + Gr_n, \tag{2.2}$$

where r_n is the residual between the propagated state and the measurement.

To reduce the number of calculations, we studied the values of the Kalman Gain for different tracks in simulation and found out that the gain matrix can be precalculated for different values of curvature (implying different multiple scattering) and different combinations of the hits used at each station (implying different uncertainty of the already reconstructed track at each station). Therefore, the approximate Kalman filter implemented here propagates the track from station to station and updates it using a precalculated Kalman gain that depends on a 4-bit hit pattern (defining which of the four stations have hits), and the value of the curvature at the given point of the track.

After reaching the first station, a measurement without the beamspot constraint is stored and then the track is propagated to the center of CMS. While the energy loss was neglected for the station-to-station propagation, for the propagation to the vertex it is taken into account since the material budget of the calorimeters, the magnet and the tracker is significant. After the vertex propagation, the impact parameter of the track to the beamspot is estimated and a Kalman update is performed, requiring that the track should pass from the origin. This final update provides a beamspot constrained measurement. This approximate KF algorithm decreases the trigger inefficiency by 25% for muons from the beamspot at the same rate and provides an efficiency increase by a factor of four for displaced particles by more than 50 cm from the center of CMS.

3. Firmware implementation

The algorithm consists of two major operations to be implemented in firmware: propagation and update. Track propagation is implemented by Equation 2.1. This matrix operation instatntiated many times results in about 800 multiplications that are relatively expensive in firmware. The resource utilization is substantially reduced by exploiting the DSP slices present in modern FPGAs. Each Virtex 7 690T FPGA contains 3.6k of such slices where each slice can perform the operation $x + y * z$ with no LUT resource usage. Therefore all matrix operations are mapped to DSP minimizing resource utilization. A well defined track requires at least two stubs, and given that there are four stations with maximum two stubs each, there are 22 possible tracks with two, three, or four stubs that can exist. All those combinations are implemented in parallel. Each track update in those 22 track chains corresponds to a different Kalman gain to be precalculated which is mapped in one BRAM. Then, the state update in Equation 2.2 is also implemented in DSPs. Then, tracks that are sharing hits are cleaned based on the quality defined by the number of hits and the χ^2 of the

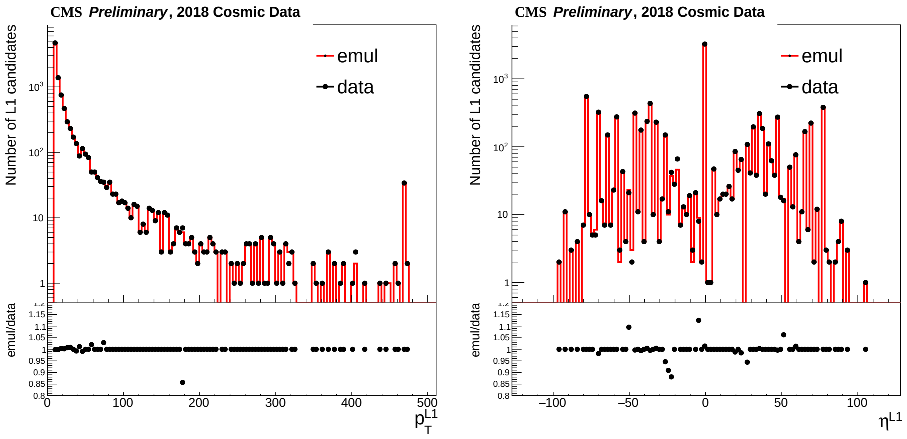

Figure 1: Agreement between data and emulator for the transverse momentum p_T and pseudorapidity η for events collected by CMS in 2018.

fit. Cleaned tracks are sorted and the top four with highest p_T are forwarded to the Global Muon Trigger.

The firmware is implemented using Vivado High Level Synthesis (HLS). Vivado HLS compiles C code to HDL, optimally pipelining the algorithm for the specific chip and the required clock frequency and by mapping calculations to DSP cores. As the clock frequency is increased, the latency is reduced as expected. However, for very high frequencies, the timing constraints require too many pipeline steps, resulting in increased latency. Therefore, there is an optimal clock frequency for each design and chip. For the KF implementation discussed here, optimal clock frequency is 200 MHz, however 160 MHz is used to be identical to the clock running on the current BMTF algorithm.

4. Integration in CMS data taking

The firmware is integrated in CMS data taking running in parallel with the current BMTF algorithm. We implemented both algorithms into the same FPGA. Both KF and BMTF algorithms take the same data. The BMTF is used for trigger, while the KF algorithm is readout in the DAQ for each collected event out of any trigger path in CMS. This implementation exploits real data to study the algorithm with a plan to deploy it online as the default track finder in Run III.

Algorithm	LUT	FF	BRAM	DSP
Legacy BMTF	43%	23%	35%	0%
Kalman Filter	16%	11%	15%	25%

Table 1: Resource Utilization for both track finders in a Virtex 7 690T FPGA

Table 1 shows a comparison of the resource utilization in a Virtex 7 690T FPGA. Exploiting 25% of the DSP budget results in a smaller utilization for the Kalman filter. Both track finders occupy about half of the FPGA.

The performance of the algorithm is studied using CMS data. Figure 1 shows the agreement between data and a C++ emulator mimicking the Kalman filter firmware measured to be better than

99%. The total latency of the KF algorithm is 9.5 bunch crossings which fits in the latency budget of the current trigger. At HL-LHC, the latency budget is much larger and the modern FPGAs are larger and can run at higher clock frequencies. Synthesizing the KF algorithm in a Kintex Ultrascale+ FPGA (KU15P-2) results in a resource utilization of about 10% and latency of about 4 bunch crossings.

5. Hardware platforms for HL-LHC

Based on the low resource utilization observed during the implementation of the Kalman filter in the current system, and the expected low utilization needed to combine muon information with tracks from the track trigger at HL-LHC, several R&D efforts have started towards ATCA processor boards with cost effective FPGAs. The OCEAN platform, pursued at UCLA, is an ATCA baseboard with the basic services that has a mezzanine connector (interposer). A processing mezzanine, including processing FPGA and optics connects to the interposer. The first version of the processing mezzanine will feature a ZYNQ Ultrascale+ MPSoC. This cost effective chip, contains FPGA logic (2x the Virtex 7) 74 transceivers up to 28 Gbps and a set of application processors and real time processors running commercial Linux. The processing system is interconnected to the FPGA logic through an AXI interface enabling applications such as real time data analysis, exhaustive monitoring and memory lookup in the FPGA using the DDR memory connected to the processor.

6. Summary

An implementation of a Kalman filter algorithm for tracking in the CMS Barrel Muon system has been implemented in the CMS data taking in Run II. The algorithm was implemented in High level Synthesis. Using the DSP slices of the Virtex 7 FPGA, the resource utilization is less than 25% of a Virtex 7 690T FPGA, enabling implementation of the current algorithm and the KF algorithm in the same chip. Given the low resource utilization achieved, targeted R&D for the same algorithm is performed on cost effective FPGAs. The OCEAN platform is one of those efforts investigating possible gains by using a System on Chip as the main FPGA processor.

References

[1] CMS Collaboration, “The CMS experiment at the CERN LHC”, JINST 3 S08004

[2] CMS Collaboration, “Performance of the CMS Muon Detectors in 2016 collision runs”, CERN-CMS-DP-2016-46

[3] K. Compton et al. “The MP7 and CTP-6: multi-hundred GBPS processing boards for calorimeter trigger upgrades at CMS”, JINST 7 C12024

[4] R. Fruhwirth, “Application of Kalman filtering to track and vertex fitting”, Nucl. Instrum. Meth. A 262 444

[5] M. Pesaresi et al, “An FPGA based track finder at Level 1 for CMS at the High Luminosity LHC”, TWEPP 2016