

An FPGA implementation of the Hough Transform tracking algorithm for the Phase II upgrade of ATLAS

F. Alfonsi, on behalf of the ATLAS Collaboration

INFN Bologna, Italy

E-mail: fabrizio.alfonsi@cern.ch

Abstract. The LHC High Energy Physics experiments will face challenging trigger requirements in the next decade. The increase of the peak of luminosity to $5\text{-}7.5 \times 10^{-34} \text{ cm}^{-2} \text{ s}^{-1}$ will push the major experiments such as ATLAS to exploit the online tracking for their inner detector to achieve 10 kHz of events from 1 MHz. In this paper is presented the proposal for a tuned Hough Transform algorithm implemented on high-end FPGA, versatile and adaptable to different tracking situations. The platform developed allows to study different dataset, including ATLAS simulations, using a software simulating the firmware. Xilinx FPGA have been selected for this implementation, exploiting currently the VC709 commercial board and its PCI Express Generation 3 technology. The system provides the features to possibly process a 200 pile up event of ATLAS Run4 in the order of 10 μs averagely, with the possibility to run two events at a time. Preliminary tests showed a tracking efficiency $> 95\%$ for single muon. The project plans to be proposed for the Event Filter TDAQ ATLAS Upgrade of Phase-II.

1. Introduction

The Large Hadron Collider (LHC) [1] accelerator located at CERN will go through several upgrades in the near future. These will allow to reach a peak luminosity of $5\text{-}7.5 \times 10^{34} \text{ cm}^{-2} \text{ s}^{-1}$. The LHC experiments, such as ATLAS (A Toroidal LHC Apparatus) [2], are preparing important detector and infrastructure changes to exploit it and cope with the new conditions, to perform the desired physics studies such as the physics beyond the Standard Model. In particular for ATLAS, these upgrades will lead to a pile-up of 140 and to 1 MHz of events to handle in the final stage of the trigger chain, called Event Filter (EF). The EF is under development to reduce the event rate down to 10 kHz [3, 4]. To withstand 1 MHz of rate, ATLAS plans to integrate the CPU Farm dedicated to the tracking operations, the one which will cost more computing power, with a hardware farm composed of commercial accelerators such as FPGA and GPU boards [4]. In particular this infrastructure should speed-up the tracking operations related to the ATLAS charged particle inner detector, the Inner Tracker (ITk). The hardware farm is planned to accelerate the tracking operations by providing a filter of the hits from ITk to reduce the Processor Farm amount of processes, within a trigger time window of $\sim 10 \mu\text{s}$. ITk is planned to be separated in 1280 regions alongside the azimuth angle ϕ and the pseudo-rapidity coordinate η , to perform the tracking functionalities. The hardware farm will cover all the regions, with a number of cards per region decided by their future performance.

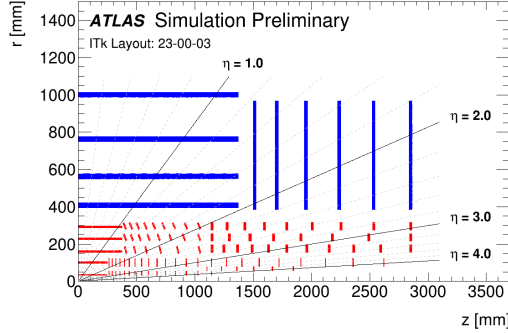


Figure 1. Schematic representation of the ITk sub-detector.[4]

2. Proposal of Filtering Algorithm Implementation

The requests of the ATLAS trigger led to explore peculiar research areas, such as the implementation of tracking algorithms on hardware accelerators. One selected solution was the Hough Transform [5] (HT) for straight lines. In this algorithm, described in a simplified way in Figure 2, the points $x_{0,1}, y_{0,1}$, part of a straight line, are processed through all the possible m, b values to plot them as lines in a new parameter space, called Accumulator [6].

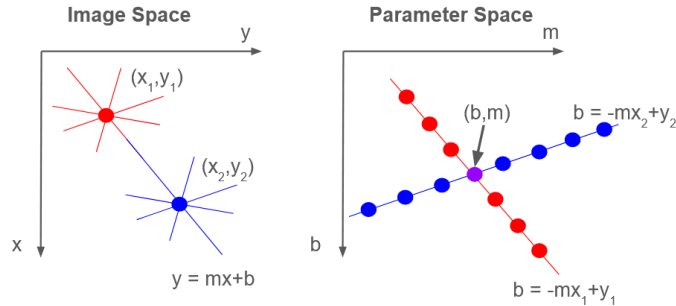


Figure 2. Schematic description of the Hough Transform concept applied to straight lines.

This concept is applied to the ATLAS case study by firstly transforming the ITk track's hits from the cartesian to the polar coordinate system with r as radius and ϕ as the azimuth angle. Then, approximating the electromagnetic field of the detector and applying simplification for small angles, the formula of the Hough Transform applied to ATLAS is $qA/p_t = (\phi_0 - \phi)/r$, referring to $m = (y - b)/x$ for the regular straight lines formula. Here p_t represents the particle's transverse momentum, q represents the particle's charge, A is the constant $0.0003 \text{ GeV mm}^{-1}$, and ϕ_0 is the azimuth angle of the track [5]. In more details, r and ϕ are recovered by processed groups of particle hits detected by ITk, called "cluster", a more precise position information for the track's points. In this paper are presented two versions of the HT algorithm implemented on high-end FPGAs of the Xilinx vendor. The goal is to develop a proposal for the hardware accelerator part of the EF. This study started in 2020 and previous updates are summarized here [7],[8]. The targeted FPGAs-based Xilinx commercial boards are the VC709 and the VCU1525. Low-level Hardware Description Language has been used to develop the firmware architecture. The algorithm concept has been built to balance the physics performance and the firmware capabilities. Several other firmware blocks and further operations are planned to compose the final filter for the EF tracking task, many of them still under preliminary development stages.

The further results are related to clusters of one of the 1280 ITk regions, [0.3:0.5] rad in ϕ and [0.1:0.3] in μ .

2.1. Flexible version

The approach of this version is to develop a versatile HT targeting low processing time with respect to low resources utilization. This is achieved through high parallelization and high clock frequency. The conceptual operations are shown in Figure 3. On the left, the Accumulator

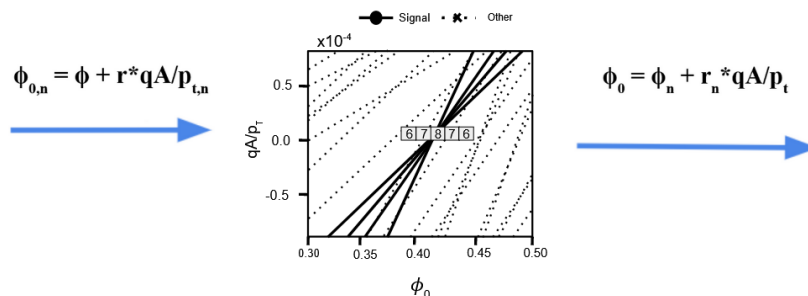


Figure 3. Schematic representation of the Flexible HT approach. The formula on the left is to build the accumulator and the operation is done looping over qA/p_t . On the right the same formula is used to extract the candidate clusters for the track.

is built using the original HT formula. One cluster per layer, for all layers, are processed by applying more HT formula concurrently. On the right, first the candidate tracks pairs $qA/p_t:\phi_0$ are extracted by checking the bins with crossing lines; second the clusters which generated that pair are recovered by applying the original HT formula “again” through all the clusters $r_n:\phi_n$ of the event. This second operation is done one $qA/p_t:\phi_0$ pair per time. Two main firmware design techniques have been exploited: the first is to separate the firmware in different clock domains. This allows the firmware compiler to place and to route it into the FPGA device more freely. The second applies a different method to draw the lines generated by the clusters in the Accumulator. First, the regular HT formula is applied for a part of the Accumulator, drawing a truncated line. Then this is monotonically copied and pasted to the rest of the Accumulator. This second method is done to use fewer multiplication logic blocks. Table 1 shows preliminary

Table 1. Physics performance for 200 pile-up events from software analysis.

Z slices	μ 1-2 GeV	μ 2-4 GeV	$\mu > 4$ GeV	π 1-2 GeV	π 2-4 GeV	$\pi > 4$ GeV
19 Z-slices	95.9 %	100 %	98.6 %	88.8 %	92.7 %	95.2 %
6 Z-slices	96.6 %	100 %	98.6 %	89.3 %	93 %	95.9%

physics performance for 200 pile-up events from software analysis. The efficiency is extracted by the % of truth tracks reconstructed. “Z slices” means to separate the 200 pile-up event for a single region of the sub-detector ITk alongside the Z coordinate of ATLAS, to avoid the saturation of the Accumulator. This is performed by partially overlapping the different slices alongside Z to avoid to break the tracks. The hardware implementation of this HT on Xilinx VCU1525 commercial board reached 350 MHz for an Accumulator of $216 qA/p_t \times 32 \phi_0$ bins, eight layers of the outer-most barrel region of ITk and 160 clusters stored per layer, allowing averagely a processing time per event and per card of $< 10 \mu s$. The resource utilization are shown in Table 2.

Table 2. Screenshot summarizing the resources occupied by the Flexible HT on the VCU1525 Xilinx card.

Resource	Utilization	Available	Utilization %
LUT	296990	1182249	25.12 %
FF	448979	2364480	18.99 %
BRAM	183	2160	8.47 %
DSP	1816	6840	26.55 %

2.2. Low-Resources version

The approach of this version is a highly-tuned HT keeping low resource utilization. The process concept is to optimize the firing of the bins in the Accumulator by making preliminary complex and complete studies in software where the range of r and ϕ of the events run generates pattern lines which will be drawn in the HT parameter space of $216 qA/p_t \times 64 \phi_0$ bins (Figure 4). Because this method approximates r , it can be exploited only in the barrel region of the detector. Then on firmware, the value of r is considered constant per layer, and the input ϕ is compared with predefined ranges to decide which pre-generated and pre-saved pattern to draw in the Accumulator (Figure 4) [9]. Table 3

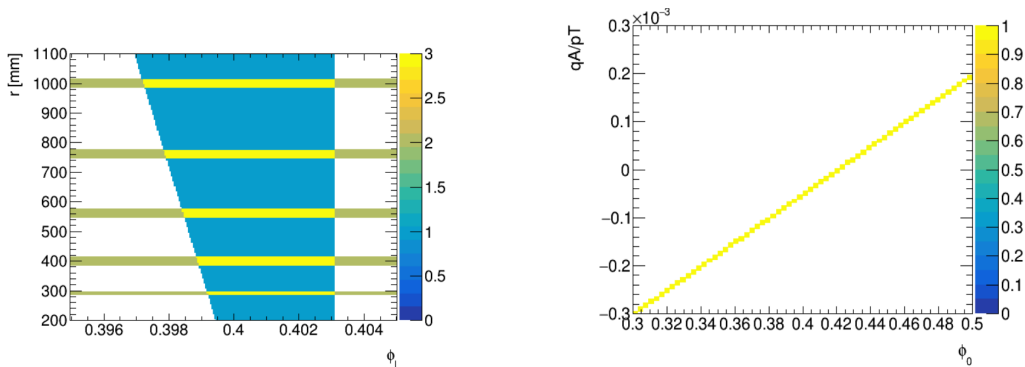


Figure 4. On the left there is a picture showing the cluster range studied in the software operations of the Low-Resources HT, where the blue area represents the r and ϕ range to activate specific bins of the accumulator while the yellow area the ITk detector. The information on the left picture generate the right one, describing how one single cluster is represented in the Accumulator in the firmware operations.

shows feasibility studies of the tracking efficiency and the resource occupied by the firmware. These latter are achieved by implementing a logically compatible function as software. "Scan r " and "Fix r " represent the two methods tested: the first is about considering all possible ranges of ϕ , assuming r as a floating value between the range of the layer. The second considers r fixed per layer with the centre of the distribution for each layer. These different approaches lead to two different results: high performance but high output amount of candidate tracks and fits for the Scan r and vice versa for the Fix r . The possibility to apply a duplicate removal is under study for the first case. The results of the feasibility study of the firmware implementation are shown in Table 4, with the resources occupied by the firmware on the VC709 Xilinx commercial board. The maximum frequency stably reached is 250 MHz.

Table 3. Feasibility studies of the physics performance and the resource occupied by the firmware for the Low Resources HT. "Flexible" HT resources are related to a safe amount of Look-Up-Tables of the FPGA occupied, "math" represents the physics performance for the pure Hough Transform algorithm. These resources are for the accumulator builder block only.

	Flexible	math	Fix r	Scan r
#LUTs	400k	-	31k	108k
< #fits>	-	2375	2709	292k
< #candidate tracks>	-	242	270	4881
Inefficiency	-	4.5 %	6.5 %	0.36 %

Table 4. Screenshot summarizing the resources occupied by the Low Resources HT on the VC709 Xilinx card.

Resource	Utilization	Available	Utilization %
LUT	82131	433200	18.96 %
FF	180584	174200	20.84 %
BRAM	154	1470	10.48 %
DSP	104	3600	2.89 %

3. Conclusion

The ATLAS HEP experiment will face several challenges in the following years. The expected luminosity will require fast tracking within a $< 10 \mu\text{s}$ time windows. In this paper are presented two FPGA implementations of the Hough Transform algorithm to propose a feasible and performing solution for fast tracking in HEP experiments. The firmware development is mature, it reached functionalities satisfying and working frequency of at least 250 MHz. The preliminary efficiency results are promising, reaching in few cases value $> 95 \%$.

4. Acknowledgments

Copyright 2020 CERN for the benefit of the ATLAS Collaboration. CC-BY-4.0 license

References

- [1] Evans L. and Bryant P. *LHC Machine*, JINST 3, S08001 (2008).
- [2] ATLAS Collaboration, *The ATLAS experiment at the CERN Large Hadron Collider*, JINST 3 (2008) S08003
- [3] ATLAS Collaboration, *Technical Design Report for the Phase-II Upgrade of the ATLAS TDAQ System*, <https://cds.cern.ch/record/2285584?ln=it>
- [4] ATLAS Collaboration, *Technical Design Report for the Phase-II Upgrade of the ATLAS Trigger and Data Acquisition System - EF Tracking Amendment*, <https://cds.cern.ch/record/2802799?ln=it>
- [5] Mikael Martensson, *A search for leptonquarks with the ATLAS detector and hardware tracking at the High-Luminosity LHC*, ISSN 11651-6214, ISBN 978- 91-513-0707-7 (2019).
- [6] Allam Shehata Hassanein et al, *A survey on Hough Transform, Theory, Techniques and Applications*, IJCSI Vol 12, I 1, No 2, ISSN 1694-0784 (2015).
- [7] Alessandro G. et al, *Hardware Implementation Study of Particle Tracking Algorithm on FPGAs*, MDPI Electronics 2021, 10(20), 2546, <https://doi.org/10.3390/electronics10202546>
- [8] Alessandro G. et al, *Simulated Hough Transform Model Optimized for Straight-Line Recognition Using Frontier FPGA Devices*, MDPI Electronics 2022, 11(4), 517, <https://doi.org/10.3390/electronics11040517>
- [9] K. Todome on behalf of the ATLAS TDAQ community, "Hough transform implementation on FPGA for event filtering of HL-LHC", ICHEP 2022, PoS(ICHEP2022)216