Journal of Instrumentation

PAPER • OPEN ACCESS

Hough Transform FPGA solution for High Energy Physics online fast tracking

To cite this article: F. Alfonsi *et al* 2024 *JINST* **19** C02070

Published by IOP Publishing for Sissa Medialab

Received: *November 24, 2023*
Accepted: *January 2, 2024*
Published: *February 27, 2024*

16th Topical Seminar on Innovative Particle and Radiation Detectors
Siena, Italy
25–29 September 2023

Hough Transform FPGA solution for High Energy Physics online fast tracking

F. Alfonsi,[a] **F. Del Corso**[a] **and A. Gabrielli**[a,b]

[a] *Istituto Nazionale di Fisica Nucleare, Bologna, Italy*
[b] *Physics and Astronomy Department, University of Bologna, Bologna, Italy*

E-mail: falfonsi@bo.infn.it

Abstract: In the coming years, significant upgrades are planned for ATLAS and other High Energy Physics experiments at CERN. Both the technologies and methodologies employed will undergo changes for the scheduled runs at the end of the decade. The LHC accelerator itself will also undergo multiple modifications, allowing it to achieve a peak of instantaneous luminosity up to 5–7.5 $\times$ 10^{34} cm^{-2} s^{-1}. These enhancements will necessitate the experiments to handle a greater number of events at the conclusion of the data acquisition chain. For instance, ATLAS will be compelled to employ online tracking for its inner detector, aiming to achieve a final event rate of 10 kHz from the 1 MHz originating from the Calorimeters and the Muon Spectrometer trigger discrimination. Among the architectures explored to expedite fast tracking, there is consideration of a "hardware accelerator" farm, an infrastructure made of interconnected accelerators such as GPUs and FPGAs, designed to accelerate the tracking processes. The project presented here proposes a tuned Hough Transform algorithm implementation on high-end FPGA technology, specifically designed to adapt to various tracking situations. A development platform comprising software and firmware tools has been created to study different datasets. This platform utilizes software to simulate the firmware and to perform hardware tests. AMD-Xilinx FPGAs were chosen to implement and asses the system, with specific boards such as the VC709, the VCU1525 and the Alveo U250. Strategies such as low-level design for the firmware architecture, leveraging the card's features like PCI Express data transfer, and the > 1 million gates array available have been exploited. The system underwent testing using internally simulated events generated within the ATLAS environment. Simulated 200 pile up events were used to evaluate the algorithm effectiveness. The average processing time was estimated to be below 5 µs, with the capability to concurrently process two events per algorithm instance. Internal efficiency tests have shown conditions where track finding performance for single muon tracking exceeded 95%.

Keywords: Accelerator Applications; Particle tracking detectors; Trigger algorithms

https://doi.org/10.1088/1748-0221/19/02/C02070

2024 JINST 19 C02070

Contents

2024 JINST 19 C02070

1 Introduction

In recent years, there has been a growing focus on the hardware acceleration of algorithms applied in the research domain of triggers for data acquisition systems. The user-friendliness of several high-level tools, such as HLS4ML (High-Level Synthesis for Machine Learning), especially those leveraging python language, has attracted Machine Learning and Neural Network developers to implement their ideas in hardware accelerators. The primary technologies employed for these research fields are GPU (Graphical Processor Unit) and FPGA (Field Programmable Gate Array), utilizing programming languages as CUDA, OpenCL, VHDL, python and complex tools as HLS4ML, TensorFlow, etc. These development strategies are also currently being explored in Physics, in particular in the trigger and data acquisition systems of High Energy Physics experiments. Here, the imperative is to filter the events to store in the final repositories. An example is the ATLAS experiment (A Toroidal LHC ApparatuS) [1] located at the CERN laboratories of Geneve, Switzerland. This detector uses a complex trigger system to discriminate and select events to be saved, incorporating both hardware and software online trigger operations. ATLAS is scheduled to begin the so-called Run 4 in 2029, exploiting the Large Hadron Collider [2] (LHC) circular accelerator and its proton-proton bunches collisions occurring at intervals of 25 ns. For Run 4, the LHC is expected to generate collisions capable of reaching a center of mass energy of 14 TeV and a peak of instantaneous luminosity of $5\text{–}7.5 \times 10^{34}\,\mathrm{cm}^{-2}\,\mathrm{s}^{-1}$, allowing on average up to 140 p-p interactions per bunch crossing. These new features will impact the volume of events to be processed in the ATLAS Trigger and Data Acquisition (TDAQ) [3] system. To cope with these challenges, the TDAQ plans to employ new technologies and strategies to achieve an event discrimination from 40 MHz to 10 kHz, as shown in figure 1. Among the critical requirements for the TDAQ, especially for the last level of triggering, the Event Filter, are the tracking operations, currently managed by the Event Filter Processor Farm (PF). The PF is currently in the development phase, aiming to identify the most suitable and sustainable solution while preserving the physics performance. One proposal involves the utilization of a hardware accelerator farm, which, when combined with the Processor Farm, would facilitate the tracking operations by filtering a portion of the

hits originating from the tracker detectors described in [4], in particular from the Inner Tracker (ITk), the innermost ATLAS detector. To achieve this goal, several groups inside the ATLAS collaboration are investigating GPU, FPGA and high performing CPU acceleration. These efforts encompass a spectrum of ideas, ranging from specific tracking algorithms to apply a pre-tracking operation, Neural Networks, Graph Neural Networks and more. In this paper the proposal of a Flexible Hough Transform FPGA (FHTF) is presented, detailing the firmware design techniques and the development tools; some preliminary results for both the firmware and physics analysis are also presented. The objective of the FHTF is to be proposed as a FPGA accelerated filter algorithm for tracking within the TDAQ system of ATLAS, with a targeted processing time less than 5 µs.

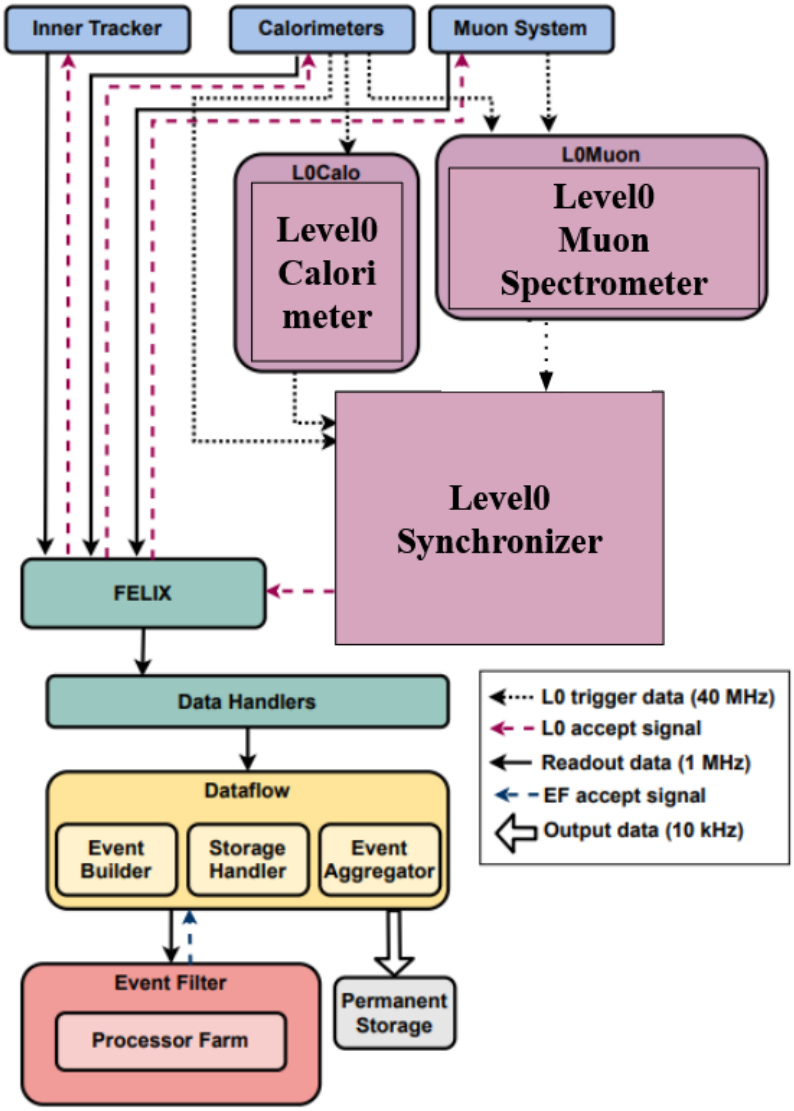

Figure 1. Scheme of the ATLAS TDAQ system. Reproduced from [3]. CC BY 4.0.

2024 JINST 19 C02070

2 Hough Transform

The Hough Transform (HT) [5] is a versatile tracking algorithm which applies a change in the coordinate system of the track under test, creating a new parameter space. Figure 2 shows an example of the HT applied to bi-dimensional straight lines. In this representation, two points in the X:Y coordinate system are transformed into two straight lines in the parameter space usually named Hough space. The two points in X:Y are connected by the straight line formula $Y = X \cdot m + b$. By converting the formula for m, a new parameter space is generated, where the crossing point of straight lines represents the m:q pair describing the line generated in the original coordinate system. This method is highly parallelizable, making it well-suited for hardware accelerators such as FPGA based cards. The parallelizability comes from the possibility to do all the necessary processes of the HT concurrently, because the mathematical operations aren't interdependent.

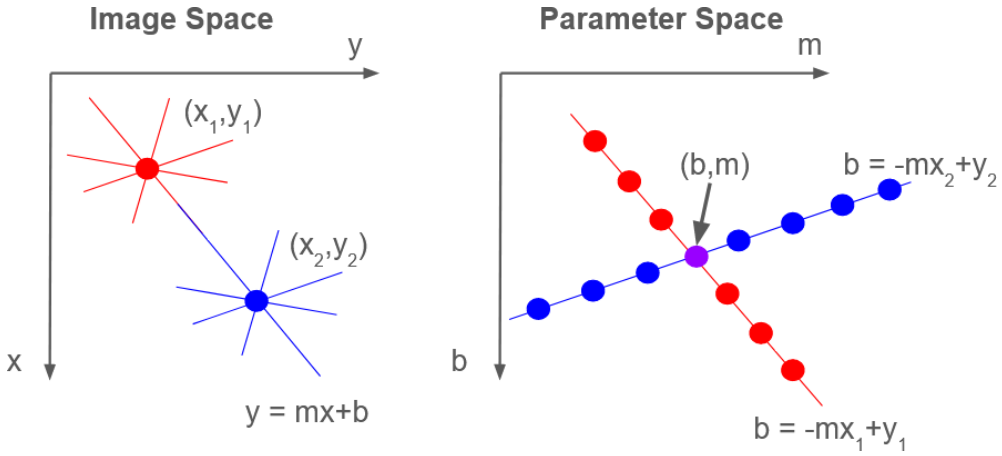

Figure 2. Image showing an example of the Hough Transform for bi-dimensional straight lines. The parameter space shown in the right figure is the so-called Hough space.

A first proposal of the HT for the Event Filter was introduced in the study [6]. This proposal suggests to consider the electromagnetic field as invariant and applying the polar coordinate system to the ATLAS cartesian X:Y:Z space. Then, simplifying for small angles, the ITk hits, defined by their radius r and their azimuth angle ϕ, can be described by the straight line formula

$$\frac{qA}{p_T} = \frac{(\phi_0 - \phi)}{r} \tag{2.1}$$

where q is the charge of the particle, p_T the transverse momentum, A is a constant which depends on the magnetic field and has here the value of 0.0003 GeV mm^{-1}, and ϕ_0 the track's azimuth angle. Equation (2.1) is used to construct the most important component of the HT, the *Accumulator*, the Hough parameter space which gathers all the hits, points in the basic coordinate space, transformed in lines. The Accumulator is a 2D histogram made of bins, bins representing a range of possible tracks with a specific transverse momentum and a given azimuth angle. It is used to identify intersection points between straight lines to select the line features. In the ATLAS case study the feature is the pair qA/p$_T$:ϕ_0. The ITk architecture, shown in figure 3 [3], consists of superimposed internal cylindrical layers in the so-called barrel region, the one covering the pseudorapidity (η) range [−1:1]. The proposed HT alternative implements one Accumulator per ITk layer. The complete HT core

comprises several superimposed Accumulators. It's important to remind that, for the ATLAS ITk use case, this algorithm is proposed as a filter, not as a complete tracking algorithm, as additional complex operations are required for full track reconstruction. The qA/p_T:ϕ_0 bins in the several superimposed Accumulators, if enough of them are active because crossed by lines, are considered a candidate track. In other words a qA/p_T:ϕ_0 pair is valid if a number of superimposed pairs are activated (crossed by lines) through several Accumulators, intended as several layers. A candidate track is made of the qA/p_T:ϕ_0 information and the hits that generated that pair, and it requires operations in the Event Filter Processor Farm to confirm it as a track to activate or not a trigger signal in the TDAQ.

The HT proposed in this paper is termed as Flexible because it is configurable in various aspects, such as the size of the Accumulator, input/output/internal variable resolution, and the trade-off between a faster algorithm occupying more firmware resources and a smaller but slower one. The lines are drawn in the Accumulator by applying the straight line formula bin by bin, looping over ϕ_0 for one input pair r:ϕ. Another versatile characteristic is the option to use the reverse formula of equation (2.1) for ϕ_0 if it yields better performance. The Flexible Hough Transform presented here is designed for FPGA hardware. In the development and optimization of the algorithm, the focus has leaned more toward enhancing processing time rather than maximizing physics performance. This means that in the development and optimization of the algorithm, the increase of already optimal processing time is preferred with respect to already optimal physics performance. To achieve this, a specific method was employed to recover the hits that generated a qA/p_T:ϕ_0 pair selected as a candidate track. This involves reapplying the equation (2.1) after extracting a valid Accumulator bin, keeping the bin value fixed and looping over all the hits of the event. This process allows the recovery, at each clock cycle, of all hits that enabled the firing of that bin. The equation (2.1), or its reverse, must be respected to select a generative hit, as it was used during the Accumulator building to draw the lines which crossed and fired a valid bin. Figure 4 summarizes the algorithm's design. Starting from the left, one r:ϕ pair is processed for all qA/p_T or ϕ_0 bins. Following that, the Accumulator is built and the FHTF iterates over the hits of the event with the selected pair qA/p_T:ϕ_0 fixed for all. This final operation recovers the generative hits.

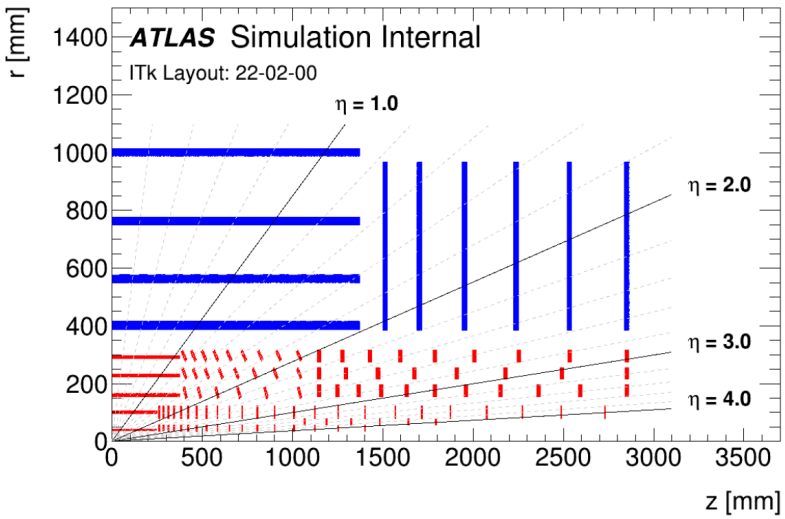

Figure 3. Scheme of the charged particle detector ITk. Reproduced from [3]. CC BY 4.0.

2024 JINST 19 C02070

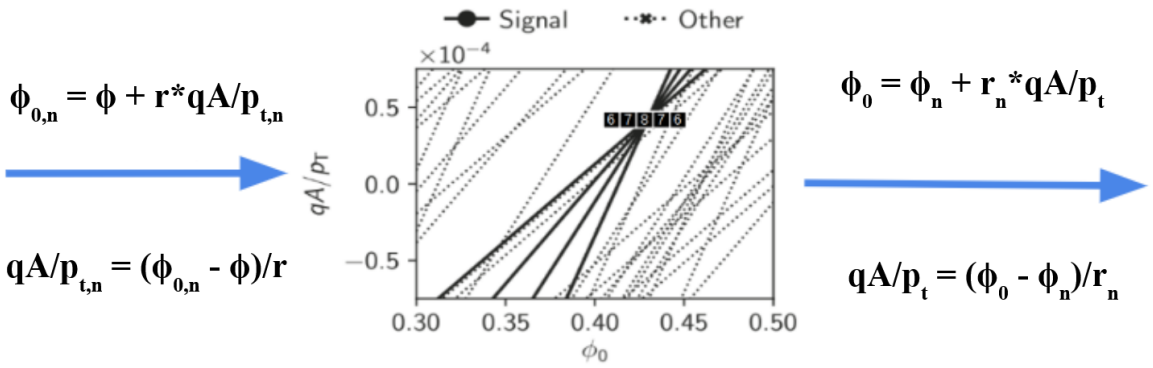

Figure 4. Scheme of the FHTF algorithm. From the left an input $r : \phi$ is acquired and a scan over all the qA/p_T bins or all the ϕ_0 bins is made to build the lines in the Accumulator. In the center the Accumulator is built and, comparing the superimposed active bins, the qA/p_T:ϕ_0 pair passing the threshold is selected. The numbers 6-7-8-7-6 signify here that five bins are checked to decide if the central one is valid. On the right the loop over all the hits for one qA/p_T:ϕ_0 is done to recover the generative hits.

3 Development tools

To implement the algorithm a development platform tool has been made. It includes a python code to simulate the firmware and perform data analysis, bash files to configure the latter, other software to manage the tests, and a hardware setup. A series of service bash files is dedicated for selecting which parameters to use in the python code, reconfiguring the firmware design with those parameters, creating the firmware project, compiling it, and analyzing the obtained results. Three AMD-Xilinx FPGA based cards are currently available for the hardware tests: the VC709, the VCU1525 and the Alveo U250. The analysis of firmware outputs involves utilizing PCI Express data transmission and a loop-back method. The hardware setup is shown in figure 5. The algorithm development tool is still under debug to perfect it. The python code is under debug to reach 100% of equality with respect to the firmware outputs. Furthermore, it is under rebuilding to speed it up. Currently, one preliminary simulated event, compatible with ATLAS, is analyzed averagely in 0.20 ± 0.05 s.

4 Firmware design and performance

In section 2, we discussed the algorithmic strategies employed to achieve low processing times in FPGA. Additionally, methods suggested in the AMD-Xilinx manuals ([7] and [8]) were implemented, and firmware design techniques were devised to achieve higher event processing rate. These techniques are essentially three and will be listed in the following paragraphs.

One strategy is shown in figure 6. It is based on the separation of the firmware in several clock domains, all managed by distinct clock sources internally generated in the FPGA by Multi-Mode Clock Managers or Phase Locked Loops. This separation affects positively the capability to operate the firmware at higher frequencies, as it facilitates better signal passage through the FPGA Super Logic Regions. Furthermore, it provides greater flexibility to the compiler in arranging the firmware within the chosen FPGA. This separation enables more efficient placement and routing operations, as each block running on a specific clock source is considered separate from others in terms of the clocking network, allowing for more effective arrangement and routing.

2024 JINST 19 C02070

Figure 5. Photo of one of the setups used for the hardware tests. It shows a PC with inserted the VC709 (up) and the VCU1525 (down) in the PCI Express slots.

The second technique exploits the monotonic behavior of the lines drawn in the Accumulator. Instead of calculating all the bins to draw the line by using equation (2.1), only an initial portion of it is computed. The remaining part is constructed by copying the initial segment and pasting it in a different position, following the monotonic direction. This copy-paste operation is executed by an addition, which is less resource consuming with respect to the bare implementation of equation (2.1) or its reverse. By implementing this, several factors of Digital Signal Processors (the FPGA components used for the multiplication) can be saved, depending on the configuration of the overall structure.

The third technique exploits the amount of resources available in the FPGA targeted and the necessary Flip Flops and internal memory RAMs to store one event. This method involves in duplicating the Accumulator and the hit's storage, allowing to process two events concurrently. This happens because the FPGA's memory stores the Accumulator and hits of event n while event $n+1$ is being processed. This method is described in figure 7 showing a simulation of the firmware.

These techniques combined contributed to the current firmware performance. The minimum achievable frequency with any major configuration in the Alveo U250 is 400 MHz. The configuration shown in this paper has $168\,\mathrm{qA/p}_T \times 48\,\phi_0$ Accumulator bins, with eight Accumulators superimposed. The hits storage is configured for 1600 hits, 200 per Accumulator. Figure 8 displays the real scheme of the Alveo U250 filled with the placed and routed firmware. Notably, only half of the FPGA is used (two Super Logic Regions), and the components activated to function (highlighted in yellow, red, blue and green) are visibly separated instead of being a very big spot in the middle of the FPGA. Figure 9 shows in its upper part a screenshot demonstrating that setup and hold times constraints of the clock dependent components are actually satisfied, while the lower screenshot shows the FPGA resources utilization. The second firmware design technique described above allows saving 760 Digital Signal Processors (DSPs).

2024 JINST 19 C02070

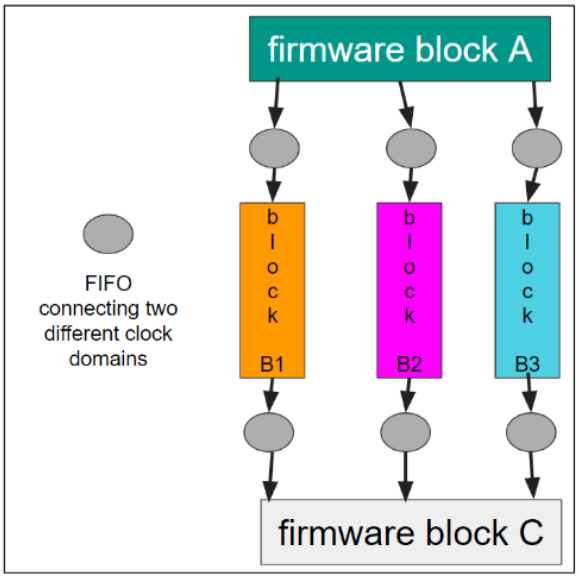

Figure 6. Scheme representing the use of a firmware design technique. Here the firmware blocks A, B1, B2, B3 and C are run by 5 different clock sources and connected by an independent clock First In First Out (FIFO).

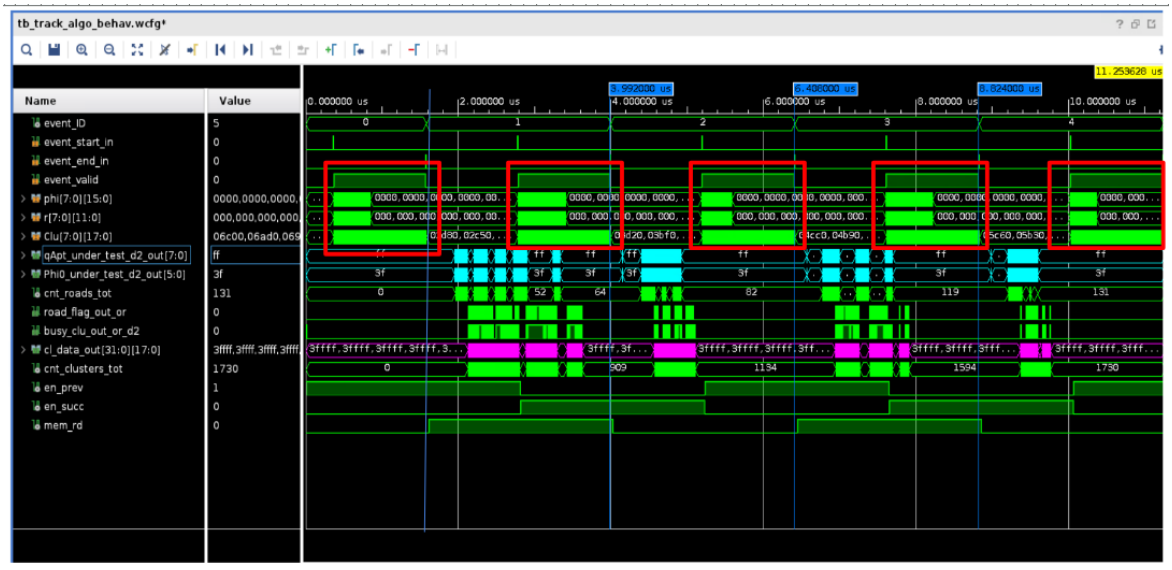

Figure 7. Screenshot of a simulation of the FHTF firmware. The red squares highlight the acquisition of the inputs from the firmware while the violet signal represents the outputs. It can be seen that the inputs of the event n+1 overlap the output of the event n, demonstrating the possibility to run two events concurrently.

2024 JINST 19 C02070

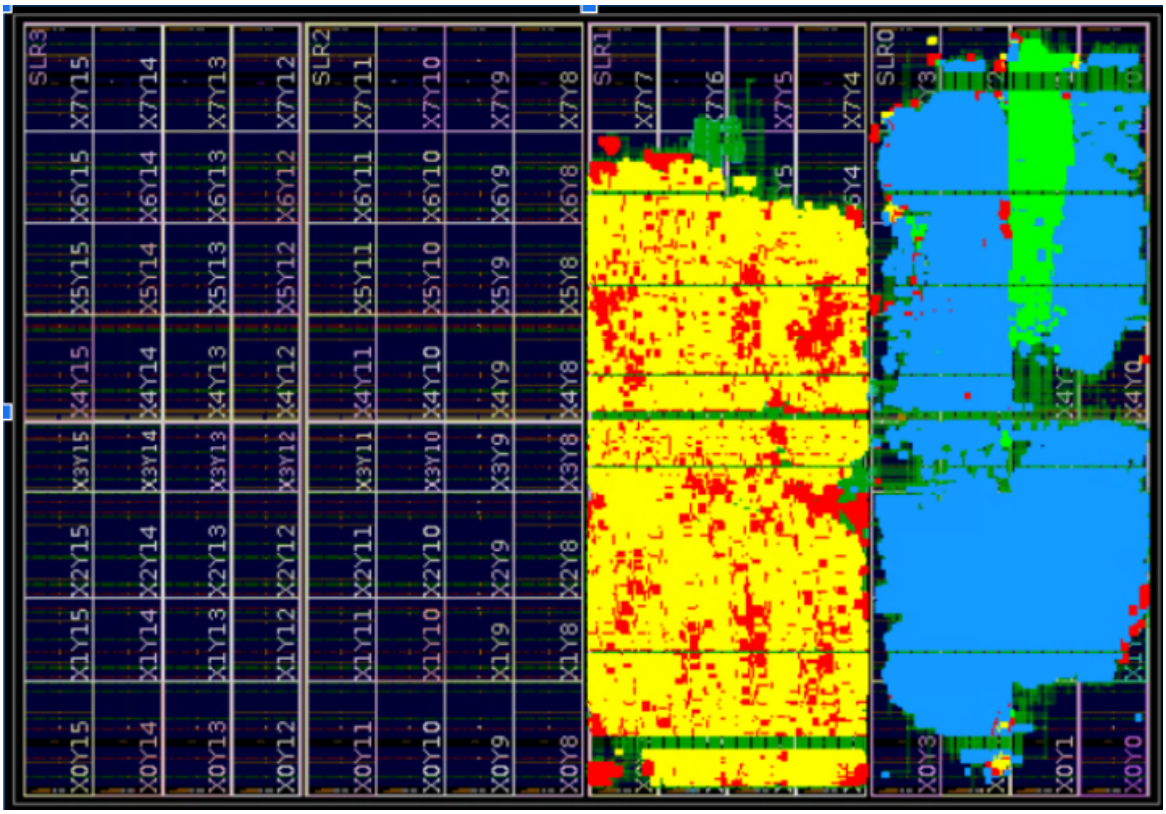

Figure 8. Scheme of the Alveo U250 FPGA with the FHTF firmware placed and routed, matching the timing constraints of setup and hold times at 400 MHz. In the upper left part of the scheme can be found the letters SLR stating the four Super Logic Regions.

Worst Negative Slack (WNS):	0.01 ns
Total Negative Slack (TNS):	0 ns
Number of Failing Endpoints:	0
Total Number of Endpoints:	1367922

Resource	Utilization	Available	Utilization %
LUT	290023	1728000	16.78
LUTRAM	11754	791040	1.49
FF	649060	3456000	18.78
BRAM	547	2688	20.35
DSP	2184	12288	17.77
IO	2	676	0.30
BUFG	22	1344	1.64
MMCM	5	16	31.25

Figure 9. Two screenshots showing in the upper part the respect of the setup and hold times constraints and in the lower part the summary of the resources occupied in the Alveo U250.

5 Preliminary physics performance

The algorithm was tested preliminary with several random generated events. The first complex utilization was with unofficial simulated events compatible with ATLAS. These particular events are from two supposed regions of the barrel area of ITk, the regions in ϕ [0.3:0.5] rad and η [0.1:0.3] and [0.7:0.9]. The eight outermost layers, out of thirteen available for ITk, have been selected. In particular, regarding figure 3, the outermost red line and the four blue lines, using a separation of upper and lower part for these last four, have been exploited to generate the simulated events. The threshold was set at seven layers active. The q/p_T range is $[-1.0571758563701927 : 1.0571758563701927]$. This unconventionally large number of digits is required because the integer only operations done in firmware could loose an important amount of information if the operands are not almost integer (for example 7.00000000001), due to the fact that the software simulating the firmware, and the firmware itself, round at the lower integer value. Table 1 summarizes the preliminary tracking efficiency results of the ITk areas studied separated in η regions. Two particle types have been studied: the muons and the pions. The efficiency is defined as the amount of candidate tracks found with respect to the total present in the simulated event. The simulated events are separated per type of particle, meaning that only one particle type candidate tracks can be searched in a single event. Considering the average size (amount of hits) of the events and the produced candidate tracks, the average estimated processing time per event is 3 µs for the Accumulator building and 2.7–4.5 µs for the hits recovery part.

Table 1. Table summarizing the preliminary physics efficiencies of tracking for muons and pions.

Region η	µ	π
0.1–0.3	$> 96.5\%$	$> 90\%$
0.7–0.9	$> 97\%$	$> 82\%$

6 Conclusion

The High Energy Physics experiments will face several challenges in the next years. The increase of interactions per bunch crossing that the experiments at LHC will face will result in the need to process a higher event rate in the final trigger operations. In particular, in the ATLAS Run 4 there will be an increase of one order of magnitude of event rate to manage by the Event Filter. This will affect in particular the tracking operations, leading to necessitate a higher amount of processing power or the adoption of hardware acceleration strategies. Fast tracking within small trigger windows is an important requirement for the future. We are developing an FPGA implementation of the Hough Transform algorithm to propose a feasible and performing solution for fast tracking in HEP experiments. This work started in 2020 ([9]) and it has been carried forward in the last four years ([10–12]). Firmware design has been defined and consolidated, capable to run at 400 MHz with compact resources utilization. The preliminary physics analysis studies are promising, suggesting the possibility to achieve interesting performance of simulated muons tracking efficiency $> 95\%$ and processing time < 5 µs. The next steps will be focused on the completion of the full detector performance.

2024 JINST 19 C02070

References

[1] ATLAS collaboration, *The ATLAS Experiment at the CERN Large Hadron Collider*, 2008 *JINST* **3** S08003.

[2] L. Evans and P. Bryant, *LHC Machine*, 2008 *JINST* **3** S08001.

[3] ATLAS collaboration, *Technical Design Report for the Phase-II Upgrade of the ATLAS TDAQ System*, CERN-LHCC-2017-020 (2017) [DOI:10.17181/CERN.2LBB.4IAL].

[4] ATLAS collaboration, *Technical Design Report for the Phase-II Upgrade of the ATLAS Trigger and Data Acquisition System — Event Filter Tracking Amendment*, CERN-LHCC-2022-004 (2022).

[5] A.S. Hassanein, S. Mohammad, M. Sameer and M.E. Ragab, *A Survey on Hough Transform, Theory, Techniques and Applications*, *IJCSI* **12** (2015) 139 [arXiv:1502.02160].

[6] M. Mårtensson, *A search for leptonquarks with the ATLAS detector and hardware tracking at the High-Luminosity LHC*, Ph.D. Thesis, Uppsala University (2019), https://s3.cern.ch/inspire-prod-files-1/12c8ec58e941b7a3385a4e6491d74a9c.

[7] AMD-Xilinx, *UltraScale Architecture Configuration User Guide*, Xilinx manual UG570 (2022), https://docs.xilinx.com/v/u/en-US/ug570-ultrascale-configuration.

[8] AMD-Xilinx, *UltraFast Design Methodology Guide for FPGAs and SoCs*, Xilinx manual UG949 (2022), https://docs.xilinx.com/r/en-US/ug949-vivado-design-methodology.

[9] F. Alfonsi, *Study and Optimization of Particle Track Detection via Hough Transform Hardware Implementation for the ATLAS Phase-II Trigger Upgrade*, Ph.D. Thesis, Università Di Bologna (2021), https://amsdottorato.unibo.it/9774/.

[10] ATLAS TDAQ collaboration, *Hough transform implementation on FPGA for event filtering of HL-LHC*, *PoS* **ICHEP2022** (2022) 216.

[11] A. Gabrielli, F. Alfonsi and F. Del Corso, *Hough Transform Proposal and Simulations for Particle Track Recognition for LHC Phase-II Upgrade*, *Sensors* **22** (2022) 1768.

[12] A. Gabrielli et al., *Simulated Hough Transform Model Optimized for Straight-Line Recognition Using Frontier FPGA Devices*, *Electronics* **11** (2022) 517.

2024 JINST 19 C02070