Hough Transform implementation on FPGA for event filtering of HL-LHC

Kazuki Todome[a,*] and Fabrizio Alfonsi[a] on behalf of the ATLAS TDAQ collaboration

[a] *Università e INFN, Bologna,*
V.le C. Berti Pichat 6/2 40127 Bologna Italy

E-mail: kazuki.todome@cern.ch, fabrizio.alfonsi@cern.ch

The ATLAS experiment plans to upgrade its Trigger DAQ system dedicated to HL-LHC. Due to the expected large amount of data, one of the key upgrades is how to filter the events in a short time. Part of the filtering is performed based on calorimeter and muon spectrometer information, and then further event filtering is done in the Event Filter (EF) system with data including the ones from the inner tracker (ITk). From the ITk, O(10^{5-6}) clusters are expected per beam-crossing. Within those clusters, EF needs to perform regional tracking at 1 MHz. In this report, we will introduce one of the proposals for this event filtering using an FPGA-based solution. In this setup, we adopt a Hough Transform algorithm on FPGA to filter the cluster candidates associated with the track. The algorithm has been implemented on the VC709 board which features a Virtex-7 FPGA. In order to evaluate its performance, we used simulated hit clusters from a single muon event together with 200 pileup events. Compared to the previous study, resource utilization is reduced down to 8%, while keeping inefficiency within 6.5%.

41st International Conference on High Energy physics
6-13 07 2022
Palazzo della Cultura e dei Congressi, Bologna, Italy

*Speaker

1. Introduction

The Large Hadron Collider (LHC) at CERN is the world's highest energy particle accelerator. In order to probe more comprehensive physics, a significantly large amount of data is to be collected during the next decade after the LHC project, in the High Luminosity LHC (HL-LHC) project. At the HL-LHC, the instantaneous luminosity is planned to be increased by a factor of 5 to 7.5 with respect to the LHC design. Accordingly, the pileup is expected to be increased to 200 on average. The detectors used for LHC, including ATLAS [1], need to be upgraded to operate in such extreme environments. As part of the upgrade of the ATLAS detector, the current inner detector is going to be replaced by new silicon-based inner trackers (ITk).

The ITk consists of novel pixel and SemiConductor Tracker (SCT) designed with a high-granularity to achieve such resolution required in a high pileup environment. Figure 1 shows the current ITk design in the $R - z$ representation due to the cylindrical ATLAS symmetry. The region where the detector is arranged parallel to the beam axis is called the barrel region, and the other region is the end-cap region. Because of the symmetric collision, more tracks are expected in the barrel region, which is covered by the inner five pixel layers and the outer physical four SCT layers. Each physical SCT layer consists of two layers of strip sensors. Because of the broad coverage and the high granularity of the sensor size, O(10^{5-6}) clusters are expected from ITk per beam-crossing. Therefore, the data acquisition (DAQ) system of the ITk detector is one of the challenging developments. The DAQ system is aimed at scaling the 40 MHz beam-crossing rate

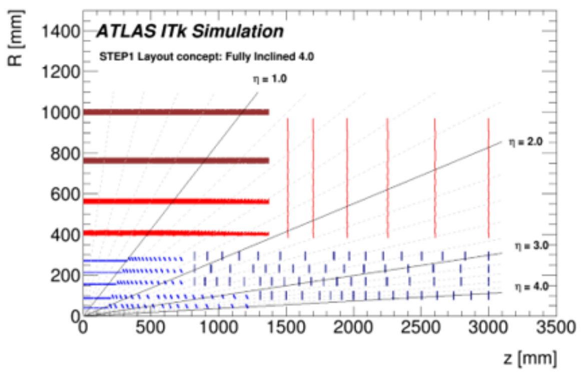

Figure 1: A design of the ITk geometry [2]. Blue (Red) lines describe pixel (SCT) positions.

down to 1MHz. For this, the trigger system selects useful events using calorimeters and muon spectrometers. The events selected by the trigger are transferred to the event filtering (EF) system, which further filters the events down to 10 kHz. Since the EF system is the first DAQ system that processes the ITk output, efficient design is one of the key prospects of EF system. On the EF system, first, online hit clusters are formed from the ITk hits. Second, the clusters that can construct tracks are grouped. In the end, online fitting is performed for each obtained group, and the fitted track is used for event skimming. The second step is called pattern recognition, and this function is discussed in this report.

The pattern recognition function receives a bunch of clusters according to the ATLAS geometry. The Hough Transform (HT) is a promising algorithm for extracting groups of clusters which may construct a track from those inputs [3]. The basic idea of HT is to convert the cluster parameters to the phase space of the track and to search for probable track parameters. Figure 2 shows the input clusters parameterized with r and ϕ used in this application. Due to the constant magnetic field in the detector, each cluster can be converted to the set of parameters $(qA/p_T, \phi_0)$ of tracks passing the given cluster, using

$$qA/p_T = (\phi_0 - \phi)/r \tag{1}$$

where q is the charge of the particle, p_T is the transverse momentum of the track, ϕ_0 is the initial track angle, and A is the constant term. This relation is used to convert all clusters on a layer to a plane describing possible track parameters. Then, by adding each plane and searching for track parameters activated in enough planes, it is possible to extract track parameters that are likely to be reconstructed as a track, as shown in Figure 3. The summarised plane is called the accumulator. Using equation 1, possibly associated clusters can also be extracted from the found track parameters. This extracted set of clusters is called a road. In the later stage, online fitting is performed only on the clusters contained in a road. Thus significant reduction in the fitting process is possible.

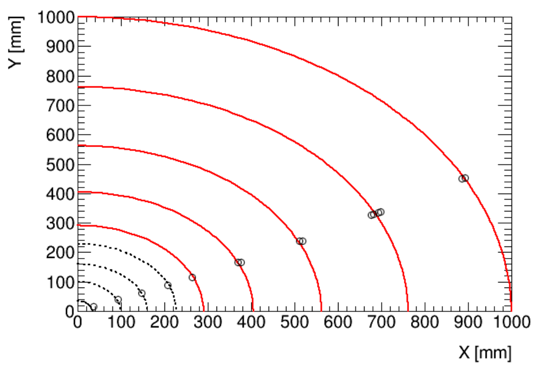

Figure 2: The clusters distributed on the ALTAS geometry. The clusters are distributed only on the detector planes shown as lines.

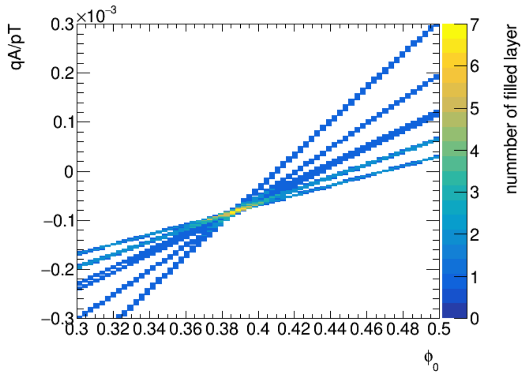

Figure 3: Accumulator generated from the clusters shown in the figure 2. A road is seen around the center of the accumulator.

In order to adopt HT for the pattern recognition in the HL-LHC environment, there are some requirements. The most basic requirement is to keep the system with low latency, as the system needs to function at 10 kHz. Because of this reason, hardware-based implementations, including FPGA, are the ideal approach. In addition, while keeping high efficiency of the essential track, the system needs to keep a low number of roads and associated clusters to suppress processing time in the consequent block. Furthermore, those functions need to be constructed with as low resources as possible to save the number of hardware required for this function. The developed demonstrator had been confirmed for basic functionalities in the previous study [4], but the expected resource size was not feasible. This report introduces a method adopted to solve this issue and the obtained results.

2. Overview and limitation of the previous study

Figure 4 describes the overview of the logic blocks designed in the previous study [4]. In block A, provided inputs are converted to the accumulator and propagated to block B. Block B is dedicated

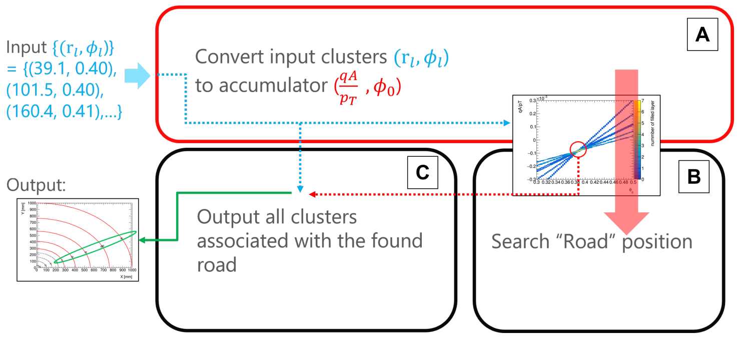
Input $\{(r_l, \phi_l)\}$ = {(39.1, 0.40), (101.5, 0.40), (160.4, 0.41),...}
A
Convert input clusters (r_l, ϕ_l) to accumulator $(\frac{qA}{p_T}, \phi_0)$
B
Search "Road" position
C
Output all clusters associated with the found road
Output:

Figure 4: Overview of the logic blocks designed in the previous study.

to search of roads and provides the road positions to block C. Block C compares the found road with the input clusters and extract associated clusters. In this implementation, most resource-consuming block A needs 400k Look Up Tables (LUTs), which is more than 90% of the available resource. In this implementation, mathematical evaluations using equation 1 are performed on an FPGA. This process requires many resources on an FPGA board. Thus, this point is improved in a new low-resource implementation, as introduced in the following sections.

3. Approach of low resource implementation

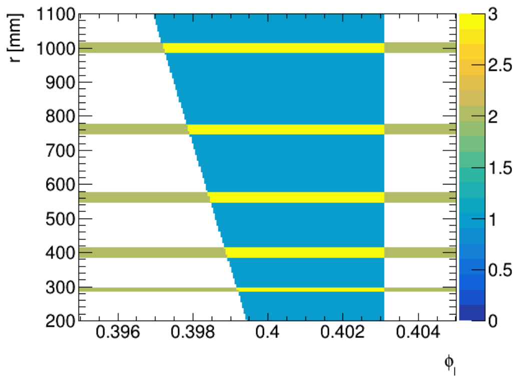

Figure 5: Example of the range of input cluster parameter for a given accumulator bin. The blue region shows the range of (ϕ, r) to activate the given accumulator bin. The yellow region shows the range of r where ITk is placed. The light yellow region is the range to activate the given accumulator bin for each layer.

This implementation allows reducing resources by splitting functions into software and firmware. First, in the software, all possible mathematical calculations are performed as follows. From a given configuration, such as the numbers of the accumulator bins, the range of track parameters $(qA/p_T, \phi_0)$ represented by one accumulator bin is defined. The given track parameter range is interpreted as the range of input cluster parameters (r, ϕ) to fire the given bin, using the equation 1. Figure 5 shows an example of an evaluated range of input cluster parameters for a given bin. As shown in the figure, each physics layer has an almost identical range of ϕ to activate the given bin independently from the r value, suggesting that the acceptance range of ϕ is also independent. Therefore, only the ϕ range is evaluated in the following discussion. This evaluation is done for each accumulator bin in the software before the firmware implementation. Since the evaluation takes long time, this software cannot be adopted as the DAQ system, but this step reduces the processing time and resources on firmware. Second, the evaluated range of ϕ is recorded for each accumulator bin on the firmware, then all the bins check whether input ϕ is in the recorded range. Since this comparison is low cost on FPGA this conversion from the input ϕ to the corresponding accumulator bins is a pretty efficient process in FPGA.

This approach is promising from a resource-reduction point of view. However, effects on the other performance points, such as efficiency and amount of outputs, are not obvious. In order to confirm those performances, we report a feasibility study in the following sections.

4. Feasibility studies

In order to evaluate performance, some configurations were assumed following the previous study. First, accumulator size is defined as $qA/p_T = [-3 \times 10^{-4}, 3 \times 10^{-4}]/\mathrm{mm}$ in 216 bins and $\phi_0 = [0.3, 0.5]$ in 64 bins. Single muon events under 200 pileup events are used as the input dataset. The input is interpreted as 18 bits of ϕ for each layer. Not all the ITk layers are used, but the outermost pixel layer and all SCT layers except the second layer are used for this process. Thus in total, 1 pixel and 7 SCT layers are used. For the resource point of the discussion, the Xilinx Virtex-7 VC709 board, which has 433k LUTs, is considered a demonstration board. Under those assumptions, software and firmware-based feasible studies are reported in the following subsections.

4.1 Software-based study

The purpose of the software-based study is to evaluate performance with the operation that is logically compatible with the firmware design. First, the software described in the section 3 is used as designed. In order to optimize the performance, two approaches to the treatment of r are considered. One is to fix the r value to the center of the r distribution for each layer (Fix-r). The other is to consider all possible ranges ϕ, assuming that r is a floating value in the range in which r is distributed (Scan-r). The evaluated ranges are used in another dedicated software mimicking firmware, to evaluate performance and resource utilization. The performance of purely mathematical operations ignoring the firmware resource is also evaluated for comparison purposes.

	Previous method	mathematical operation	Fix-r	Scan-r
#LUTs	400k	—	31k	108k
<#Fit>	—	2375	2709	292k
<#Road>	—	242	270	4881
Inefficiency	—	4.5%	6.5%	0.36%

Table 1: Results of the software-based study

Table 1 shows the evaluated performance. For the Fix-r option, LUTs utilization has been reduced to less than 8% of the previous implementation of 400k LUTs, without increasing inefficiency drastically. For the Scan-r option, inefficiency is surprisingly decreased without consuming too many LUTs resources. However, the expected output in this option is too large, because of duplicated roads. Therefore, this option is unrealistic unless a clever idea could be adopted to remove the duplicated roads. The Fix-r option is adopted in the subsequent firmware-based study.

4.2 Firmware-based study

The purpose of the firmware-based study is to confirm that this approach can be implemented and that the logic is feasible to match the timing requirements. The evaluated range of ϕ with the Fix-r option is imported into the firmware and loaded on a VC709 board.

The implementation has been achieved with the order of 82k of total LUTs, which is 19% of the available resources. In the previous implementation, block A consumed 400k LUTs, while this implementation is achieved with exactly the expected number of LUTs. The most significant component of the LUTs is consumed by logic block B in this implementation.The loaded firmware was tested with a few events and confirmed that all blocks work in the 250 MHz clock domain, and the newly implemented block works as designed by the software. Thus, we conclude that this approach is promising for the pattern recognition function for the EF in the HL-LHC environment.

5. Summary

In the challenging environment of HL-LHC, pattern recognition is one of the essential functions in Event Filtering. HT is one of the promising algorithms for this function. In the previous studies of HT, the function block to build accumulator consumes 400k LUTs on FPGA, which is more than 90% of available resources. This report introduces an idea to pre-calculate logic using software. This approach allows for reducing resource utilization of the concerned block down to less than 8% of the original implementation while keeping its inefficiency low enough. As a feasibility study, the pre-calculated logic has been imported on the firmware targeting Xilinx Virtex-7 VC709 working with 250 MHz CLK and the implemented block is confirmed to work as designed. This idea is not only promising for the pattern recognition function for the EF toward the HL-LHC environment but also feasible for any other mathematical operations on the firmware to reduce resource utilization.

References

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

[2] ATLAS collaboration, *Expected tracking and related performance with the updated ATLAS Inner Tracker layout at the High-Luminosity LHC*, Tech. Rep. , CERN, Geneva (2021).

[3] R.O. Duda and P.E. Hart, *Use of the hough transformation to detect lines and curves in pictures*, *Commun. ACM* **15** (1972) 11–15.

[4] A. Gabrielli, F. Alfonsi, A. Annovi, A. Camplani and A. Cerri, *Hardware implementation study of particle tracking algorithm on fpgas*, *Electronics* **10** (2021) .