Journal of Physics: Conference Series **2374** (2022) 012099 doi:10.1088/1742-6596/2374/1/012099

# Muon trigger with fast Neural Networks on FPGA, a demonstrator

**M. Migliorini[1], J. Pazzini[1,2], A. Triossi[3], M. Zanetti[1,2] and A. Zucchetta[1]**

[1] Istituto Nazionale di Fisica Nucleare (INFN) Sezione di Padova, Padova, Italy
[2] Dipartimento di Fisica e Astronomia "G.Galilei", Università degli studi di Padova, Padova, Italy
[3] Institut Pluridisciplinaire Hubert Curien, CNRS, Université de Strasbourg, Strasbourg, France

E-mail: matteo.migliorini@pd.infn.it, jacopo.pazzini@unipd.it, andrea.triossi@cern.ch

**Abstract.** The online reconstruction of muon tracks in High Energy Physics experiments is a highly demanding task, typically performed on reconfigurable digital circuits, such as FPGAs. Complex analytical algorithms are executed in a quasi-real-time environment to identify, select, and reconstruct local tracks in often noise-rich environments. A novel approach to the generation of local triggers based on a hybrid combination of Artificial Neural Networks and analytical methods is proposed, targeting the muon reconstruction for drift tube detectors. The proposed algorithm exploits Neural Networks to solve otherwise computationally expensive analytical tasks for the unique identification of coherent signals and the removal of geometrical ambiguities. The proposed approach is deployed on state-of-the-art FPGA and its performances are evaluated on simulation and on data collected from cosmic rays.

## 1. Introduction

Muon detectors are pivotal in a large number of particle physics experiments and apparatus. For example, the imaging of inaccessible or hidden volumes based on muon tomography has its foundations on the efficient detection and tracking of muons. In high energy physics collider experiments muon final states are often considered as golden channels for the study of rare processes, with muon trigger logics playing a decisive role in the efficient selection of such events. Indeed, depending on the application, local muon trigger algorithms are among the first stages of online event selection, often having to cope with demanding conditions such as high level of background noise, high detector occupancy and short available time for the trigger decision.

Machine Learning (ML) methods are often deployed in the final stages of data processing and have been demonstrated capable of outperforming traditional methods in many different tasks across particle physics. Recently, the development of flexible tools such as HLS4ML[5] has allowed to deploy neural networks on COTS FPGA, opening the possibility of executing complex tasks such as classification and reconstruction in hardware, close to the detector, taking advantage of the short evaluation time which can be of the order of $\mathcal{O}(100)$ ns.

Taking advantage of the capability of artificial neural networks (ANN) to solve complex problems and their short evaluation time, a novel trigger algorithm for the fast muon

identification and track parameter estimation has been developed. The method implements a combination of neural networks, aiming at removing noise and reducing the complexity of the problem, inherently dependent on the combinatorial induced by the number of signals released in the detectors, and analytical methods that are applied on the filtered signals. The latter are strongly dependent on the specific detector for which the algorithm is implemented. A complete demonstrator of the algorithm has been implemented on a Xilinx Kintex Ultrascale FPGA and tested using cosmic muons with a dedicated setup described in the next Section.

## 2. Experimental setup

The algorithm developed in this work aims at processing the digitized signals produced by a muon telescope composed by a set of Drift Tube (DT) detectors. The DTs used in this work were built at the Legnaro National INFN Laboratories (LNL) and inspired by those used in the CMS experiment [1], with which they share the same underlying design and configuration. These detectors (referred here as *mini-DTs*) are designed to provide small-footprint, roughly $70 \times 70\,\mathrm{cm}^2$, muon tracking detectors, and be deployed in a number of different configurations, as for instance in the case of the test-beam of the LEMMA collaboration [2]. Each mini-DT chamber consists of 4 layers of 16 cells, totaling 64 cells per chamber. The signals produced by each cell (referred to as *hit*) are amplified, discriminated and shaped according to the LVDS standard. Two Xilinx VC707 evaluation boards are used to implement the time-to-digital conversion (TDC) of the hits, each VC707 receiving data from 128 DT channels. The data stream of each VC707 boards is serialized with the GBTx-FPGA protocol [3] and transmitted via optical links to a Xilinx KCU1500 evaluation board mounted on the PCI express Gen-3 bus of a data server for final transfer and storage of the data. Its firmware (FW) processes the stream of the entire set of TDC hits of the muon telescope and hosts the FW implementation of the trigger algorithm, which is described in the next Section.

## 3. Algorithm overview

The aim of the trigger primitive generator (TPG) logic is to define the local position and crossing angle for muons traversing the detectors, which can be well approximated with a straight line path across the volume of a chamber. For each DT cell the drift time, i.e. the time difference between the production of primary ionization and the signal deposition on the anode wire, can be translated to a spatial distance by correcting for the electron cloud drift velocity. An inherent left-right ambiguity does however still persist, as no information on the side of the electron cloud production with respect to the wire is a priori available. Moreover, in all most common applications of the DT chambers the absolute time at which the primary ionization occurs is not an information available from external references such as independent detectors, and has to be identified in situ.

The absolute time of passage of a particle can be obtained applying the well-established mean-timer technique [6], which exploits geometrical relations associated to the staggering of cells in adjacent layers. The resulting equations are related to both the geometrical pattern of active channels, and the left-right side to which a hit is assigned inside each cell, preventing to associate a unique equation to each combination of cells. For this reason, typical TPG algorithms implement iterative approaches testing all possible combinations of cell groups and hits laterality. However, in noisy environments the number of possible combinations can increase significantly, thus creating a large number of alternatives that needs to be tested, which in turn might severely impact the performances of the trigger algorithm.

Several ML methods, and specifically various implementations of ANNs, are commonly exploited to perform denoising tasks and pattern recognition. Among the features common to many ANN-based models, the fast evaluation time is one of their most appealing features for a TPG task. While analytical approaches to the noise rejection can take several iterations over

a set of finite points, the evaluation time of an ANN is almost instantaneous. In this algorithm ANNs are used to perform noise rejection and pattern recognition on clusters of compatible TDC hits, followed by analytical relations for evaluating the time pedestal and the track parameters. The goal of the ANNs is to solve the combinatorial prior to applying the mean-timer equations. A diagram of the algorithm is shown in Figure 1.

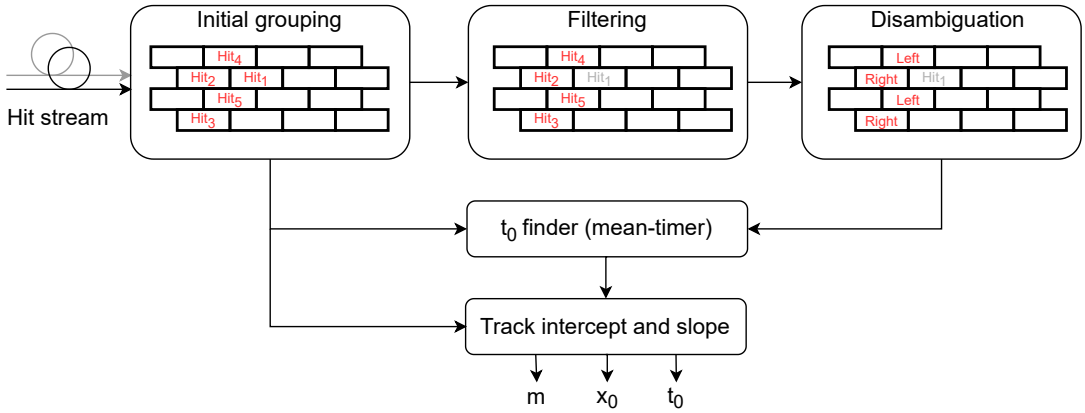



Figure 1: Schematic representation of the algorithm for each macro-cell. In the initial grouping hits are collected from the stream and positioned in the macro-cell. This is then fed to filtering and disambiguation blocks where noise is rejected and the left-right ambiguity is solved. This information is finally used to compute the muon crossing time and the track parameters.

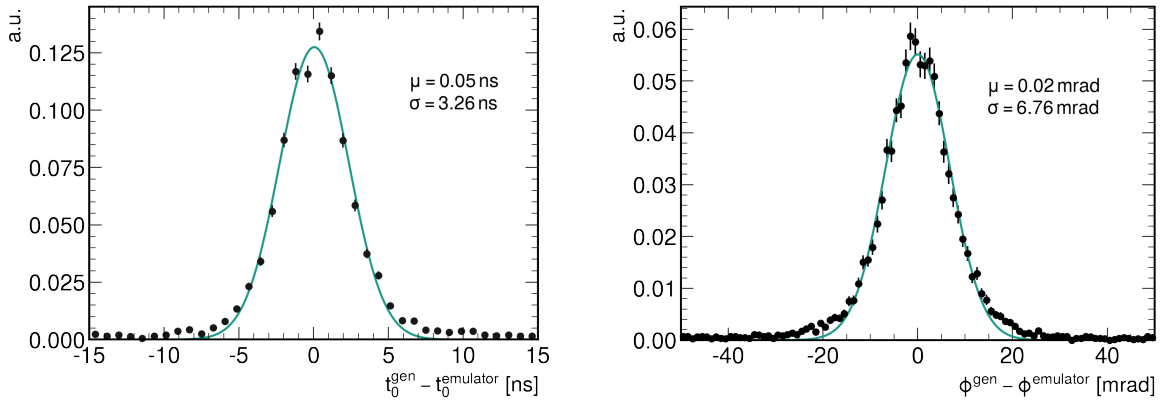
The first module, called *initial grouping*, collects hits from the stream and organizes them in time-coherent sets for each macro-cell. A macro-cell is a set of $4 \times 4$ adjacent cells. The grouped hits are then fed to the two ANN blocks. The first module contains an ANN devoted to the *filtering* of spurious hits: only the three or four hits compatible to the passage of a genuine muons are retained, while all others are rejected. Hits passing this filtering stage are fed to a second ANN performing the *disambiguation* step, i.e. predicting the side of the muon passage inside the cell with respect to the wire. Once the correct set of three or four hits, along with their lateralities, is known then the mean-timer equation related to this specific configuration can be applied to compute the muon crossing time $t_0$. All filtered hits are then mapped in the coordinate space of the detector and the track parameters are finally obtained by performing a linear regression.

The algorithm has been implemented on the KCU1500 evaluation board of the LNL testbed. The ML models have been trained usign Qkeras [4] and the correspondent HLS code generated using HLS4ML [5] package. The optimization of the ANNs through pruning and weight quantization allows for a total usage of 11k LUTs for each macro-cell, accounting for less than 2 % of the available resources of the XCKU115 FPGA used in this study. No DSP is used for either ANN block. The total latency of the two ANNs blocks is measured in 5 clock cycles at 40 MHz, and the execution of the entire algorithm can be fully pipelined.

## 4. Performance Evaluation

The performance of the algorithm are first evaluated on a private muon simulation, also used for the training of the ANNs, which include generated muons passing through the detector volume, as well as injected noise and simulated detector inefficiencies. All hits associated with a generated muons are processed by a software framework emulating the FW implementation of the algorithm. The SW framework also implements the processing of the real data collected by the muon telescope read-out, thus enabling a direct comparison of the SW and FW implementation

of the algorithm. An excellent agreement is observed between the two: the estimated emulator efficiency, i.e. the fraction of triggers produced by the FW for which a SW equivalent is produced, is 99.9% A time resolution of $3.2 \pm 0.1$ ns is estimated, as reported in Figure 2a. In Figure 2b the angle resolution of the trigger emulated trigger primitives is displayed, which is estimated as $6.8 \pm 0.1$ mrad. Both values are compatible with the expected performance of this kind of detectors. An overall trigger efficiency of $\epsilon = 99.0 \pm 0.1\%$ is estimated from the simulated sample.



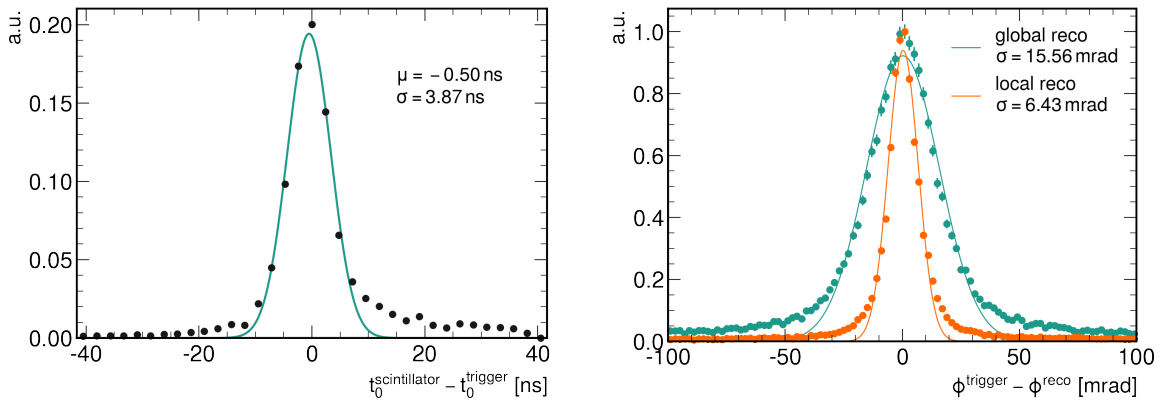

(a) (b)

Figure 2: Time pedestal (a) and trigger primitive angle (b) resolutions evaluated on the simulated sample.

The performance of the FW implementation of the algorithm is evaluated on real data with cosmic muons. Four mini-DT chambers are stacked in the muon telescope configuration. Three mini-DT chambers, installed at roughly 80 cm apart, are used for this scope. The combined information from the two most external ones is used to reconstruct the *global* muon track, and the result is compared with the FW output of the middle "probe" chamber. A *local* track is also reconstructed by only using the hits collected from the probe chamber. The coincidence of an additional pair of scintillator palettes included in the setup provides an external estimate of the muon crossing time $t_0$. The offline reconstruction of the muon tracks is performed selecting only those hits whose TDC timestamp is compatible with the external scintillator coincidence within the time window of the maximum allowed drift time ($\approx 390$ ns). A constant time calibration, specific to each chamber, accounts for the delay of the coincidence logic, the cable length, the signal digitization time, and the muon time of flight.

The measured time difference with respect to the external time reference is $3.9 \pm 0.2$ ns, as reported in Figure 3a. The bulk of the distribution is comparable with the results of the simulation, whereas a slight asymmetry in the left tail, accounting for roughly 10% of the total events, is present. This may be related to a combination of ill-defined events and instrumental effects.

The resolution on the trigger primitive angular parameter, reported in Figure 3b, is measured to be $15.56 \pm 0.1$ mrad with respect to the global track, and $6.43 \pm 0.1$ mrad with respect to the local track. The resolution with respect to the local track is comparable to one obtained in the simulation reported in Figure 2b. This is expected as both the offline reconstruction and the online FW perform a linear regression on the same hits. On the other hand, for the global track two chambers distant from each other roughly 1.6 m are used, independent from the hits used by the FW to produce a trigger. The trigger efficiency on cosmics is measured to be $\epsilon = 98.8 \pm 0.7\%$.




(a) (b)

Figure 3: (a) Time resolution between the trigger $t_0$ and the external scintillator time. (b) Resolution of the trigger track angle with respect to the global and local tracks.

## 5. Concluding Remarks and Future Outlook

A novel approach to the generation of local trigger primitives for drift tubes muon detectors has been presented, based on an hybrid model integrating artificial neural networks and analytical methods. The key feature of the proposed algorithm is the efficient use of resources to perform the identification and disambiguation of the signals thanks to two ANN layers. A demonstrator of the algorithm is deployed on a COTS FPGA for a single $4 \times 4$ macro-cell configuration. The performances of the algorithm are evaluated both on a private simulated sample and on data collected from cosmic rays with an experimental telescope testbed, and found to be competitive with the figures expected from the offline reconstruction. An extension of the proposed algorithm is foreseen to generate trigger primitives from an entire mini-DT detector geometry. A horizontal extension to cover a larger number of channels over the same chamber is under study. By replicating a series of macro-cell entities in an array, it is possible to span any 4-layer chamber configuration. An overlapping set of 4 channels across two consecutive macro-cells is expected to maximize the trigger primitive efficiency and acceptance.

As the processing of all trigger blocks can be pipelined, it is expected to be able to funnel hits from multiple macro-cells into a smaller number of ANN blocks, multiplexing all FPGA stages down to a single track parameter estimation block for an entire macro-cell array, thus optimizing the resource utilization without impacting the latency of the algorithm.

## References

[1] S. Chatrchyan *et al.*, The CMS Collaboration: *The CMS experiment at the CERN LHC*, JINST 3 (2008) S08004.
[2] N. Amapane *et al.*, The LEMMA Collaboration: *Study of muon pair production from positron annihilation at threshold energy*, JINST 15 (2020) P01036.
[3] S. Baron, J. P. Cachemiche, F. Marin, P. Moreira and C. Soos: *Implementing the GBT data transmission protocol in FPGAs*, Proceeding of Topical Workshop on Electronics for Particle Physics (TWEPP09).
[4] C. N. Coelho *et al.*: *Automatic heterogeneous quantization of deep neural networks for low-latency inference on the edge for particle detectors*, Nat Mach Intell 3 (2021)
[5] J. Duarte *et al.*: *Fast inference of deep neural networks in FPGAs for particle physics*, *JINST* 13 (2018) 07, P07027.
[6] F. Gasparini *et al.*, The RD5 Collaboration: *Bunch crossing identification at LHC using a mean-timer technique*, Nucl. Instrum. Meth. A 336 (1993) 91-97.