Journal of Physics: Conference Series **2374** (2022) 012092 doi:10.1088/1742-6596/2374/1/012092

A muon tracking algorithm for the Level 1 trigger in the CMS barrel muon chambers during HL-LHC

Jose Manuel Cela Ruiz[1] **for the CMS collaboration**

[1] CIEMAT, Av. Complutense 40, 28040 Madrid, Spain

E-mail: josemanuel.cela@ciemat.es

Abstract.
This contribution presents results on the Analytical Method (**AM**) algorithm for trigger primitive (TP) generation in the CMS Drift Tube (**DT**) chambers during the High Luminosity LHC operation (**HL-LHC** or LHC Phase II). The algorithm has been developed and validated both in software with an emulation approach, and through hardware implementation tests. The obtained performance on Phase II simulated data shows timing and position resolutions close to the ultimate performance of the **DT** chambers, with resilience to potential ageing situations. The firmware version has been implemented in the so-called AB7 (*TwinMux*), spare μTCA boards from the present **DT** system which host Xilinx Virtex 7 FPGAs, and included in a prototype chain of the HL-LHC electronics operated with real **DT** chambers during cosmic data taking. Agreement between the software emulation and the firmware implementation has been verified using different data samples, including a sample of real muons collected during 2016 data taking.

1. Introduction

To determine the trajectory of the muons generated in the hadron interaction point of CMS, the external barrel region is covered with gaseous Drift Tube chambers (**DT**) capable of detecting ionization signals caused by the particles traveling through them. When those signals are received by the read-out electronics, they are digitized and a time tag is associated to the identifier of the cells that collected the ionization avalanches, getting *hits*.

It is possible to reconstruct each muon trajectory, using this *hits* information, by computing chamber segments defined in local coordinates (horizontal position and trajectory angle, see Fig. 2 in [4]) and extrapolating them to the global coordinates of the detector. In addition, the LHC bunch-crossing (BX) where each muon was created, can be identified.

The Analytical Method (**AM**) is an algorithm designed to determine the trajectory of the muons by performing these calculations in hardware, it also detects and rejects any other spurious signals coming from different effects within the cells.

It has been implemented in VHDL and currently is executed in a Xilinx FPGA. This method is an improvement of the original idea, evolved from the mean-timer algorithm, that was developed as a *C++* software emulator [1].

2. Firmware algorithm architecture

2.1. Global architecture

AM core comprises three main building blocks: two ϕ-*angle* super-layer (SL-ϕ) [2] segment processors and one segment matching processor, all of them connected by 'First-In First-Out' (*FIFO*) buffers following the schemas shown in Fig. 1.

Each block is built with operational modules and data filters, also linked with *FIFO* buffers or *FIFO*-like ad-hoc components. The latter components are mainly groups of register chains and multiplexers that emulate the standard interface and the behaviour of a *FIFO* module, but with a reduced use of resources and designed to optimize the FPGA timing.

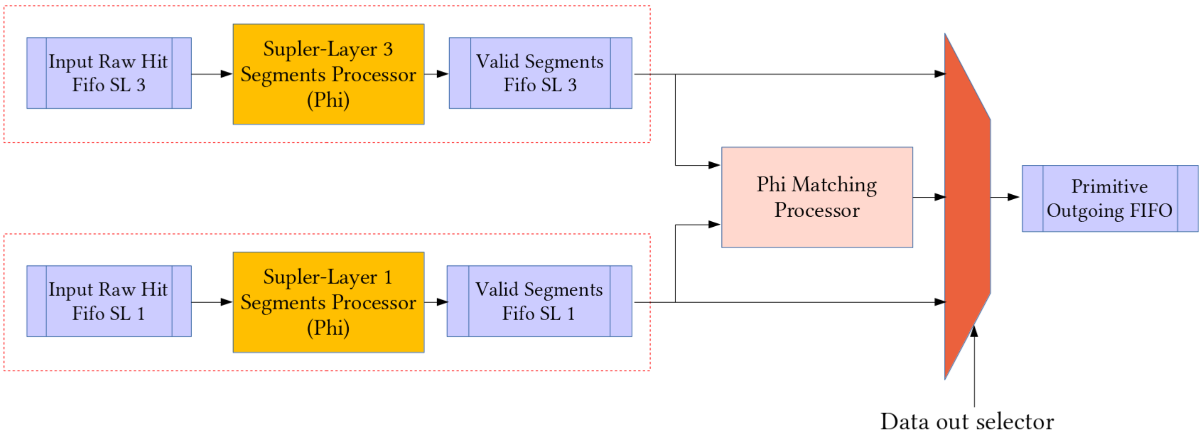

Figure 1: Algorithm architecture overview.

2.2. The Super-layer processor components

The first element in the super-layer chain (Fig. 2), the **Mixer** processor, receives *hits* and, for each one, prepares groups of segment candidates using previously stored items that were buffered in temporary memory blocks (*grouping* stage). The *grouping* is carried out according to geometrical criteria and joins the super-layer cells into triangular regions as described in [1]. *Hits* considered too old are periodically removed from those buffers.

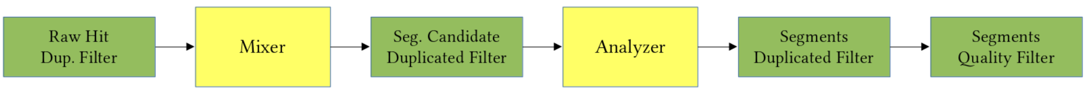

Figure 2: Building blocks for a super-layer processor.

After the **Mixer**, a filter removes duplicate segment candidates from the processing pipeline that were generated by the very nature of the *grouping* procedure.

The **Analyzer** module retrieves the remaining segment candidates and computes all possible muon trajectories (horizontal position in the super-layer and local ψ-*angle*, see Fig. 2 in [4]), the so-called valid *lateralities* [1], the possible BXs and a quality parameter (a χ^2 error estimator) and, based on them, it sends to the output or rejects each candidate. The ψ-*angle* value can also be used as a rejection criterion by limiting it to a configurable maximum. For a given event, this analysis procedure can also leave duplicate valid segments, requiring a filter after it.

Along the super-layer processing stages a mix of similar, but not equal, segments is generated. At the end of the super-layer processor chain, a final segment quality filter compares every item with the previous ones discarding those with lower relative quality.

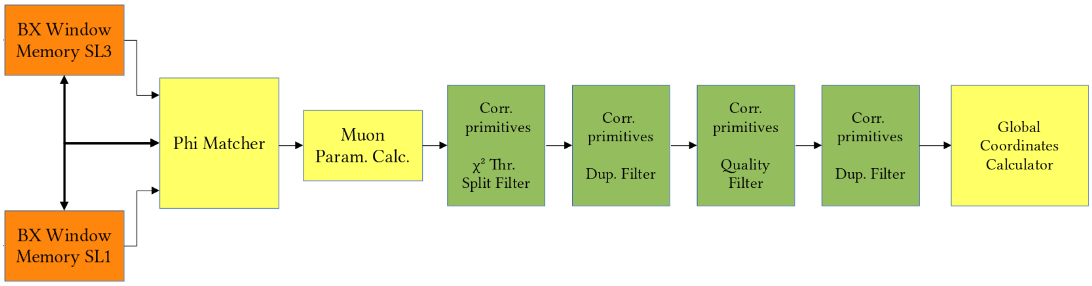

Figure 3: Super-layer ϕ segment matcher blocks.

2.3. The Phi-Matcher processor components

Valid segments generated by both *ϕ-angle* super-layer processors are periodically extracted from the middle *FIFO* buffers (Fig. 1) and sorted according to their *BX* values, storing them into a pair of memory blocks, one per *SL-ϕ* (Fig. 3). When the information in the memory buffers reaches certain time and *BX* limits, the extraction process is stopped and the **Matcher** core module retrieves them from those memory blocks, trying to find pairs of segments (one per *SL-ϕ*) whose *BX*s differ as much as one unit and the difference between their *ψ-angle* is lower than a maximum limit (they are considered collinear), discarding any other pair with larger differences. One pair of *SL-ϕ* segments per *BX* is accepted as valid and sent out of the module.

For every outgoing item from the **Matcher** core, new chamber-level trajectory parameters, another χ^2 error estimator, and a new quality tag are computed. This information, packed together with the original segment data, results in a so-called *primitive*. Following a similar schema as in the case of the super-layer processor, an additional chain of filters rejects *primitives* based on no-duplicity, quality level and χ^2 criteria.

A final module (Fig. 3) converts the computed muon-trajectory local coordinates into the global coordinates of the sector, sending the *primitives* outside the **AM** module together with a technical trigger signal.

3. Algorithm performance

3.1. Validation procedures

Two different methods have been developed to validate the algorithm performance and the quality of the trigger *primitives* that it generates.

The first one (software method) is an emulator-firmware comparison that uses an **AM** software version, implemented in *C++* and integrated into the CMS Software Framework (*CMSSW*), to reproduce the hardware behaviour. The tests are performed by injecting *hits*, obtained from real collisions, in a hardware test platform, as well as using them as input for the *CMSSW* code. The results from both are compared.

The second validation method (a hardware based one) takes advantage of the **DT** Slice Test [3] that allowed to take data from cosmic muons along several weeks using, simultaneously, the electronics of the Phase I legacy system as well as a setup for the LHC Phase II electronics running the AM algorithm. The signals taken by the **DT** chambers front-end electronics were split and injected in both systems. Their results have been also compared event-by-event.

3.2. Results

Figure 4 shows a comparison between the global *ϕ-bending* angle computed by the *CMSSW* emulation and by the **AM** hardware test setup, after injecting *hits* data coming from a particle-gun sample (2 GeV $\leq$ pT $\leq$ 100 GeV) with a pile-up level of 200 collisions per BX. The compared *primitive* pairs (around 4000 items) have the same *hits* and the same *lateralities*.

Figure 5 shows the **AM** reconstruction efficiency for the identified muon creation *BX* (also called *t0*) with respect to the values obtained by the offline Phase I legacy system. The figure was obtained by using the hardware validation method.

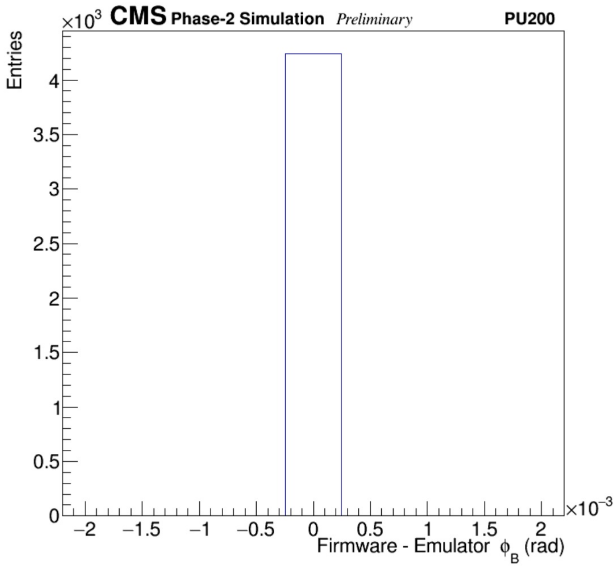

Figure 4: Emulator vs firmware ϕ-bending angle comparison.

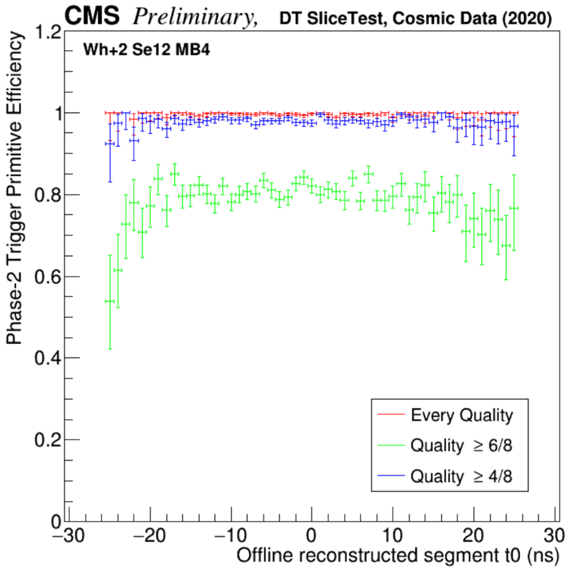

Figure 5: Phase I offline system vs **AM** results for muon creation *BX* value.

Efficiencies for different *primitive* quality levels are depicted in different colors. In red the curve every *primitive* identified by the Phase I offline is considered, while in the blue curve *primitives* are built with more than 4 *hits* and in the green one the *primitives* have six or more *hits* (3 or more per super-layer). All selected segments were built with more than 4 *hits*, which also have an inclination in the radial coordinate smaller than 30° with respect to the direction perpendicular to the chamber.

4. Conclusions

- The **AM** proposed trigger algorithm [1] for the High Luminosity LHC has been successfully migrated to a hardware implementation using VHDL.
- It has been extended to two super-layers with a subsequent matching mechanism and a converter to global coordinates.
- The agreement between firmware and emulated versions is high, as well as the efficiency in segment reconstruction when tested with cosmic muons.

References

[1] J.M.Cela Ruiz, 2017, FPGA-based algorithms for the new trigger system for the Phase II upgrade of the CMS drift tubes detector, JINST 12 (2017), no. 01, C01033, doi:10.1088/1748-0221/12/01/C01033

[2] G. Ceminara, 2007, The Drift Tube System of the CMS Experiment, JINST Vol. 172 (2007), pages 71-74, doi:10.1016/j.nuclphysbps.2007.07.016

[3] C. F. Bedoya, 2020, Implementation in a sector of the CMS Drift Tube chambers of a muon tracking algorithm for Level 1 trigger during HL-LHC, IEEE-NSS 2020

[4] R. S. Romaniuk, 2010, Performance of the CMS drift-tube chamber local trigger with cosmic rays, JINST (March 2010), 5(3): T03003, doi:10.1088/1748-0221/5/03/T03003