The Compact Muon Solenoid Experiment
Conference Report

Mailing address: CMS CERN, CH-1211 GENEVA 23, Switzerland

11 November 2016 (v4, 16 November 2016)

FPGA based algorithms for the new trigger system for the phase 2 Upgrade of the CMS Drift Tubes detector

Jose Manuel Cela Ruiz for the CMS Collaboration

Abstract

The Phase 2 upgrade of the CMS Drift Tubes detector aims at moving all the readout and trigger electronics from the inner detector to outside the cavern. Trigger algorithms need to be redesigned to handle direct timing information and remove present bottlenecks of resolution and deadtime, approaching to present high level trigger performance. In the present contribution we describe the work that has been performed to emulate the firmware that process 1 ns TDC hits from one DT chamber with the combinatorial problematic of the arrival time uncertainty in a detector with up to 400 ns of drift time.

Presented at *TWEPP-16 Topical Workshop on Electronics for Particle Physics*

PREPARED FOR SUBMISSION TO JINST

TOPICAL WORKSHOP ON PARTICLE PHYSICS 2016

26-30 SEPTEMBER 2016

KARLSRUHE INSTITUTE OF TECHNOLOGY (KIT)

FPGA-based algorithms for the new trigger system for the phase 2 upgrade of the CMS drift tubes detector

Jose-Manuel Cela-Ruiz, on behalf of the CMS Collaboration

Ciemat,

Avda. Complutense 40, Madrid, Spain

E-mail: josemanuel.cela@ciemat.es

ABSTRACT: The new luminosity conditions imposed after the LHC upgrade will require a dedicated upgrade of several subdetectors. To cope with the new requirements, CMS drift tubes subdetector electronics will be redesigned in order to achieve the new foreseen response speed. In particular, it is necessary to enhance the first stage of the trigger system (L1A). In this document we present the development of a software algorithm, based on the mean-timer paradigm, capable of reconstructing muon trajectories and rejecting spurious signals. It has been initially written in C++ programming language, but designed with its portability to a FPGA² VHDL³ code in mind.

KEYWORDS: Drift tubes; Trigger algorithms; Trigger concepts and systems (hardware and software)

¹Corresponding author.

²Type of programable digital electronic device.

³Hardware description language for configurable electronic devices programming.

Contents

1	Introduction	1
2	Principles and methods	1
3	Algorithm description and architecture	5
3.1	Algorithm overview	5
3.2	Algorithm architecture	6
3.3	Algorithm processes description	7
3.3.1	Time-window discriminator	7
3.3.2	Channel splitter	7
3.3.3	Multi-channel data mixer	7
3.3.4	Primitive analyzer	8
4	Test results	8
5	Conclusions	9

1 Introduction

A CMS Drift Tube Chamber (DT) is composed of two or three independent measuring units called super-layers, each one of them with four layers of rectangular drift cells staggered layer to layer by half a cell. Their purpose is the measurement, trigger and identification of muons in the central part of CMS.

Their current associated read-out electronics is unable to process fast enough the foreseen amount of signals that will be generated after LHC reaches its new luminosity level. In particular the on-chamber readout electronics (ROB boards) will have to be replaced, forcing to design a new type of TDC and a new way of reconstructing muon trajectories, which is necessary to be able to supply particle candidates information to the trigger system.

This work presents a muon-trajectory reconstruction algorithm based on the mean-timer paradigm [1, 2]. Unlike the previous electronic system, based on the Bunch and Track Identifier (BTI) ASIC [2] which used directly the electronic digital pulses provided by the front-end electronics, this new model uses only the time-stamp information supplied by the new TDCs and the chamber position of cells involved in each event.

2 Principles and methods

Drift tube chambers provide information about the drift time of a crossing particle after it interacts with the filling gas. When the particle crosses one chamber tube it ionizes the gas. The ionisation

electrons drift towards the central anode wire, where a charge avalanche is produced due to the high gradient electric field, generating an electrons' avalanche due to the high-voltage electric field. This avalanche is collected in the anodic central wire generating a digital signal (*hit*), whose detection time is measured regarding a certain arbitrary origin which is periodically set to zero by the “bunch crossing 0” signal (BX#0).

Gas pressure and electric field are designed to guarantee that the ionisation electrons propagate at a constant drift speed. This way it is possible to correlate directly the horizontal particle's position, inside the tube, with the signal drift time: $x = v_d \cdot t_d$

On the other hand, because drift tubes are far enough from the CMS solenoid center and are inserted inside the yoke barrel, the residual magnetic field is weak enough inside each chamber to consider negligible its effect on the particle trajectory, so it can be assumed that it is a straight line (see, figure 1).

Figure 1. Muon trajectory parameters inside a chamber considering only two random cells. In this picture, L corresponds to half a cell length, h to a cell height, while X_l and X_u are the particle's positions inside the *lower* and *upper* cells respectively.

Figure 2. Lateral ambiguity if only two cells are involved in the interaction.

Using the straight line equation and the x_i positions from two of the cells could be possible to evaluate the muon trajectory parameters, but because the drift time gives only information about the absolute distance between the particle interaction point and the anode, there is a left-right

ambiguity, giving four possible trajectories (see, figure 2) compatible with the measured drift times. In this situation, and thanks to the half-cell staggering, the combination of signals from at least three layers allows to resolve this ambiguity. This is the principle on which the mean-timer algorithm is based [1].

To obtain an useful equation, initially we can apply the straight line equation to the geometry shown in figure 1, but considering all possible left-right combinations –*lateral combinations*–. This gives four possible equations shown in equation 2.1, where H_{ul} is the horizontal distance between the considered cells, measured in half-cell width units (L); V_{ul} is the vertical distance in cell height units (h); ϕ is the trajectory slope; and x_i are the particle's position on the upper and lower considered layers. All of them are very similar, and differ only in two signs regarding the considered *lateral combinations*.

$$\begin{cases} L_{low}L_{up} \rightarrow H_{ul} \cdot L - x_u + x_l = V_{ul} \cdot h \cdot tg(\phi) \\ R_{low}R_{up} \rightarrow H_{ul} \cdot L + x_u - x_l = V_{ul} \cdot h \cdot tg(\phi) \\ L_{low}R_{up} \rightarrow H_{ul} \cdot L + x_u + x_l = V_{ul} \cdot h \cdot tg(\phi) \\ R_{low}L_{up} \rightarrow H_{ul} \cdot L - x_u - x_l = V_{ul} \cdot h \cdot tg(\phi) \end{cases} \quad (2.1)$$

Starting from previous equations, and applying them to three cells in different layers, it is possible to remove the angular dependency by subtracting two of them, and taking into account proper *lateral combinations*. The resulting equation involves only the particle positions inside the cells and a few chamber geometry parameters. Moreover this equation can be generalized to any other *combination*, by only changing the positive or negative signs of the particle coordinates in the proper way¹.

$$\begin{cases} [R_{mid}R_{up}] \rightarrow H_{um} \cdot L + x_u - x_m = V_{um} \cdot h \cdot tg(\phi) \\ [L_{low}R_{mid}] \rightarrow H_{ml} \cdot L + x_m + x_l = V_{ml} \cdot h \cdot tg(\phi) \end{cases}$$

$$V_{um} \cdot (H_{ml} \cdot L - x_m + x_l) - V_{ml} \cdot (H_{um} \cdot L - x_u + x_m) = 0 \quad (2.2)$$

Finally, because the information received from the front-end electronics is temporary data from the TDCs, it will be useful to convert equation 2.2 into a time-dependent one.

Before doing this, it is necessary to understand that TDC time values comprise not only cell drift-time values, but also other components like electronic delays, time-of-flight from the interaction point, and an additional time due to the TDC time-counters origin (BX#0) (see figure 3). Some of this time components depend only on the position of the specific cell in the detector, so they can be determined as calibration parameters. BX#0, which is the time between the TDC start and the collision, has to be evaluated to judge if the particle indeed originates at the considered bunch crossing.

Considering the next definitions:

$$\begin{aligned} \tau &\equiv \text{TDC time} & x_{cell} &\equiv v_{drift} \cdot t_{drift} \\ Bx_t &\equiv \text{Bx time since BX\#0} & L &\equiv v_{drift} \cdot T_{max} \end{aligned} \quad (2.3)$$

¹It is easy to prove by making the eight possible combinations of subtractions.

and replacing in equation 2.2 spatial distances by their time equivalent values, one obtains:

$$V_{um} \cdot (H_{ml} \cdot T - (\tau_m - Bx_t) + (\tau_l - Bx_t)) - V_{ml} \cdot (H_{um} \cdot T - (\tau_u - Bx_t) + (\tau_m - Bx_t)) = 0 \quad (2.4)$$

This equation is generalized to its final form by changing all the signs with generic coefficients (a, b, c, d), depending on the *lateral combination* considered on each layer (see table 1), and reordering its terms:

$$Bx_t = \frac{V_{um} \cdot (H_{ml} \cdot T + a\tau_m + b\tau_l) - V_{ml} \cdot (H_{um} \cdot T + c\tau_u + d\tau_m)}{V_{um} \cdot (a + b) - V_{ml} \cdot (c + d)} \quad (2.5)$$

Figure 3. Muon time components measured by the TDC from its internal counters start (BX#0) until *hit* signal is received.

Table 1. Generic coefficients for all lateral combinations involving three cells.

Cell laterality per layer				Coefficients			
Low	Mid	Up		a	b	c	d
L	L	L	→	1	-1	1	-1
L	L	R	→	1	-1	1	1
L	R	L	→	1	1	-1	-1
L	R	R	→	1	1	-1	1
R	L	L	→	-1	-1	1	-1
R	L	R	→	-1	-1	1	1
R	R	L	→	-1	1	-1	-1
R	R	R	→	-1	1	-1	1

Equation 2.5 provides a method to determine if the time values involved in the calculation conform a good particle candidate, as well as gives a way to calculate the associated bunch crossing where the muon-generating collision took place. It will be done by imposing the next group of rules:

- Denominator: $V_{um} \cdot (a + b) - V_{ml} \cdot (c + d)$ should not be null

- The result (Bx_t) of equation 2.5 must be positive
- After evaluating equation 2.5, and using the obtained Bx_t value into equation 2.4, the resulting value, in general, is not equal to zero due to rounding errors, but it should be lower than a certain tolerance that can be configured as needed.

3 Algorithm description and architecture

3.1 Algorithm overview

The designed algorithm solves equations 2.4 and 2.5 and applies them on each processing stage to a set of 10 cells at a time like the ones shown in figure 4. They are identified by the index of the base cell that belongs to the lower layer. After one set is processed, the algorithm selects another one which base cell, at the bottom layer, is consecutive to the previous one. This procedure is repeated along all cells of that layer until the end of the chamber is reached.

Since the aim of the algorithm is to be implemented in a FPGA, this approach allows to parallelize it as much as needed, being only limited by the available amount of resources in the selected device.

Figure 4. Cell and trajectories considered by the algorithm on each processing stage.

For each set of cells (on each processing stage), the algorithm creates 4-cell groups belonging to four different layers. These groups of cells are selected in a “reasonable” way to guarantee that they can contain straight muon trajectories of particles flowing from a unique bottom-layer cell. It is done by considering the cells crossed by arrows depicted in figure 4.

Figure 5. Different groups of cells considered to apply algorithm equations. The most grayed cell on each group is discarded, while equations are applied over the other three.

As the algorithm equations involve TDC time-stamps from only three cells, previous 4-cell groups are analyzed in subgroups of three, excluding one layer at a time, like in the examples in light shadow shown in figure 5. The numerical results obtained after each group analysis must be equal to their equivalent partners, in order to select the associated TDC-time group of values as belonging to a muon candidate.

Additionally, because not all cells involved in the analysis have a *hit* in each event, the algorithm imposes the restriction that at least three of each 4-cell considered group must have a signal. If not, the group under test is discarded.

3.2 Algorithm architecture

The algorithm has been written in C++, but designed using a multi-thread pipelined processing approach, with memory buffers (FIFO, ring-buffers, ...) as storage for data between consecutive processes. This way, all algorithm components can be translated into a FPGA code in a straight way.

Figure 6. Algorithm general architecture. In this figure *DTP* and *DT Prims* are acronyms for *Drift Tube Primitives*, while *DT Hits* corresponds to *Drift Tube Hits*.

The global architecture is shown in figure 6, where the boxes labeled with *DT Hits* and *DT Prims* represent FIFO buffers, those labeled from *Ch 0* to *Ch n* are a special type of ring-buffers that are able to retrieve many times the same data items until they receive a command to delete them, and the rest of boxes are processing threads.

3.3 Algorithm processes description

3.3.1 Time-window discriminator

At the beginning of the algorithm, a process reads *hits* data² from a file (optionally could read them from a network data stream) formatting them into an object structure, and inserting those objects into a FIFO buffer.

This buffer is continuously polled by the *time-window discriminator* process, which is in charge of rejecting *hits* that have a time-stamp with a value too near to the previous one. It is done in a channel-by-channel basis, keeping in memory a table with the last accepted value of each cell under analysis. The accepted hits are sent to another FIFO buffer.

This process avoids the overflow of the next stages of the algorithm due to an excessive amount of incoming data, mainly due to spurious signals inside the chambers. The rejection time-window value used in this process, is a configurable parameter.

3.3.2 Channel splitter

The only restriction that is imposed to all accepted or rejected *hits* that are sent to the algorithm is that, for a given cell they must be time-ordered, but they can reach the analyser from any cell of the chamber mixed with those from other cells in an unsorted way. So it is mandatory to classify them by channels before any other further processing.

This is done by the *channel splitter* component, retrieving them from its FIFO input buffer, and storing the *hits* in different outgoing buffers, one per channel.

3.3.3 Multi-channel data mixer

This component mixes TDC time-stamps from four different channels, according to the criteria explained in section 3.1, to build 4-cell groups of *hits* forming a new type of objects called “*primitives*”.

For each bottom cell of a super-layer it calculates the indices of the other 9 cells that should be analyzed in one processing stage. Considering these 10 cells, it extracts from their associated input buffers a preconfigured amount of *hit-objects*³, packaging them in groups of 4 *hits*. Since a buffer can be empty, at the time of trying to retrieve data from it, some of this potential *primitives* could have less than 4 data, in which case the discarding criteria described in section 3.1 is applied.

Besides, because each of these 10-cells groups share a few overlapping cells with other adjacent groups, the values extracted from any of the buffers should not be deleted immediately (like happens in an ordinary FIFO). Instead of this, they must be kept until certain amount of time has elapsed and the contained data are considered obsolete, in which case, the *multi-channel data mixer* component sends a command to the buffers to remove the older values.

After building every *primitive*, this component stores them in an outgoing FIFO buffer.

²This data contains the drift-time value from certain cell, and the necessary information to identify this specific cell: chamber id, super-layer id, layer number inside the super-layer, and a unique cell index in its layer.

³The number of hits that should be obtained per channel can be configured individually.

3.3.4 Primitive analyzer

The last process in the chain of threads, the *analyzer*, retrieves candidate *primitives* from the previous buffer and, considering the four *hits* stored in each *primitive* object under analysis, creates those 3-cell subgroups indicated in figure 5.

With every subgroup, and taking into account the relative positions of the cells where these hits have been generated, the *primitive analyzer* applies the principles and equations from section 2 to every possible *lateral combination* shown in table 1. Some of this combinations would not need to be analyzed as they are incompatible with a straight line trajectory. To avoid this, the *analyzer* evaluates a pseudo second derivative of the potential muon path, considering all of these combinations and discarding those with a non-null curvature.

Finally, the position inside the chamber and the slope of the muon trajectory are calculated and stored in the *primitive*. At the same time, a quality level is assigned to the object, depending on the number of *hits* (3 or 4) that have been used to satisfy all criteria summarized in section 2. The resulting object is stored in a final buffer, read by a component writer that records them in a file.

4 Test results

The algorithm has been tested by generating random trajectories that simulate particles crossing a chamber-equivalent geometry over different bunch crossings. The corresponding TDC values were calculated and injected in the algorithm. Its results have been compared with the original data⁴ and are shown on the next two graphics (figures 7 and 8), where it is possible to see that the agreement is good.

Figure 7. Original vs calculated trajectory slopes. The horizontal axis shows the angle values from the test tracks used as algorithm input. The vertical axis shows the corresponding values obtained by the algorithm after the analysis.

⁴Data mixer is not included in these tests.

Figure 8. Original vs calculated position in the chamber. Similar to the previous figure, here is shown the test-track horizontal-coordinate values (algorithm’s input data) on the horizontal axis, versus those obtained after the calculation process on the vertical axis.

5 Conclusions

It has been demonstrated that this algorithm is able to determine trajectories, using DT *hits* time measurements, in reference to the $BX = 0$, as a starting point.

Using a fast procedure it solves left-right ambiguities and is also able to qualify the detected DT *primitives* valid candidates.

Due to a multi-threading software architectural approach, translation to FPGA should be straight forward.

References

- [1] C.F. Bedoya, *Diseño, construcción y validación del sistema de adquisición de datos de las cámaras de deriva del experimento CMS*, Universidad Complutense de Madrid, Spain (2009)
- [2] L. Castellani et al., *BTI Reference Manual*, I.N.F.N. di Padova, Italy (Feb. 1999)