Available on CMS information server **CMS CR -2023/231**

The Compact Muon Solenoid Experiment

Conference Report

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

25 October 2023 (v3, 03 November 2023)

Adaptability and efficiency of the CMS Level-1 Global Trigger firmware implementation for Phase-2

Gabriele Bortolato, Maria Cepeda, Jaana Heikkilae, Benjamin Huber, Elias Leutgeb, Dinyar Rabady, Hannes Sakulin on behalf of the CMS Collaboration

Abstract

We present details on the new Level-1 Global Trigger at CMS for the upcoming high-luminosity operation of the LHC. Our focus is on the newly developed firmware, which employs a bottom-up generic approach to enhance menu adaptability and accommodate the increase in upstream data. We also highlight our efficient pipelining strategy that ensures excellent routability at 480 MHz. Furthermore, we discuss the firmware implementation for three prototypes targeting Serenity boards, together with their current and future testing and validation endeavours.

Presented at *TWEPP2023 Topical Workshop on Electronics for Particle Physics*

Prepared for submission to JINST

Topical Workshop on Electronics for Particle Physics
2–6 October 2023
Geremeas, Sardinia, Italy

Adaptability and efficiency of the CMS Level-1 Global Trigger firmware implementation for Phase-2

G. Bortolato,[a,b] M. Cepeda,[c] J. Heikkilä,[d] B. Huber,[a,e,*] E. Leutgeb,[a,e] D. Rabady,[a] H. Sakulin[a] on behalf of the CMS Collaboration

[a] *Experimental Physics Department, CERN,*
1211 Genève 23, Switzerland

[b] *Department of Physics and Astronomy "Galileo Galilei",*
Padova University, Via Marzolo 8, 35131 Padova, Italy

[c] *Centro de Investigaciones Energéticas, Medioambientales y Tecnológicas (CIEMAT),*
Av. Complutense, 40, 28040 Madrid, Spain

[d] *Universität Zürich,*
Winterthurerstrasse 190, 8057 Zürich, Switzerland

[e] *Technische Universität Wien,*
Karlsplatz 13, 1040 Wien, Austria

E-mail: benjamin.huber@cern.ch

Abstract: We present details on the new Level-1 Global Trigger at CMS for the upcoming high-luminosity operation of the LHC. Our focus is on the newly developed firmware, which employs a bottom-up generic approach to enhance menu adaptability and accommodate the increase in upstream data. We also highlight our efficient pipelining strategy that ensures excellent routability at 480 MHz. Furthermore, we discuss the firmware implementation for three prototypes targeting Serenity boards, together with their current and future testing and validation endeavours.

Keywords: Trigger concepts and systems (hardware and software), Trigger algorithms

[*] Corresponding author.

Contents

1 Introduction

In preparation for the Phase-2 operation at the High-Luminosity LHC, scheduled to commence in 2029, the CMS detector is undergoing significant upgrades to its detectors and readout electronics [1, 2]. The increased luminosity also poses additional challenges for the CMS trigger system. To ensure that the physics performance is maintained — and even improved — under the new pile-up conditions, a completely new Field Programmable Gate Array (FPGA) based Level-1 trigger system is being designed as part of the upgrade [3]. The new system will process high-granularity information from the calorimeters, the muon systems as well as reconstructed tracks from the silicon tracker. An increased latency budget of 12.5 μs, compared to 3.5 μs in the current trigger, will allow for sophisticated algorithms such as vertex finding, particle flow reconstruction and the extensive use of neural networks, which were previously only possible at later software-driven stages of the trigger system [4].

In this process the final stage of the Level-1 trigger pipeline, the Global Trigger is being entirely redesigned, including a modern continuous-integration-driven development and testing infrastructure and a completely rewritten firmware. Distributed over up to 12 processing boards featuring Xilinx Ultrascale+ FPGAs, the upcoming system will be tasked with handling the processing of over 26 trigger object collections from four different upstream trigger systems (figure 1) to ultimately decide whether an event is to be passed on to the software-driven stages of the trigger [5].

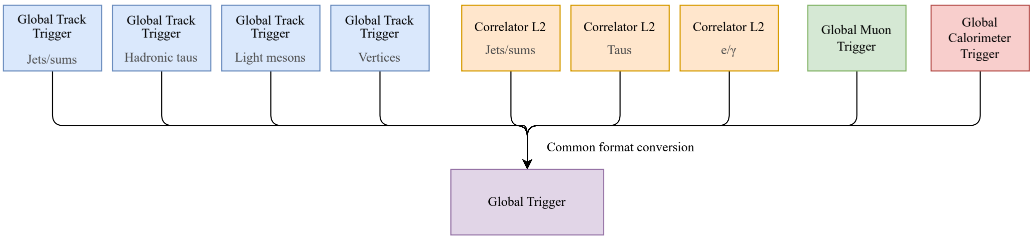


Figure 1. Upstream trigger systems interfacing with the Level-1 Global Trigger [5].

2 Algorithms & adaptability

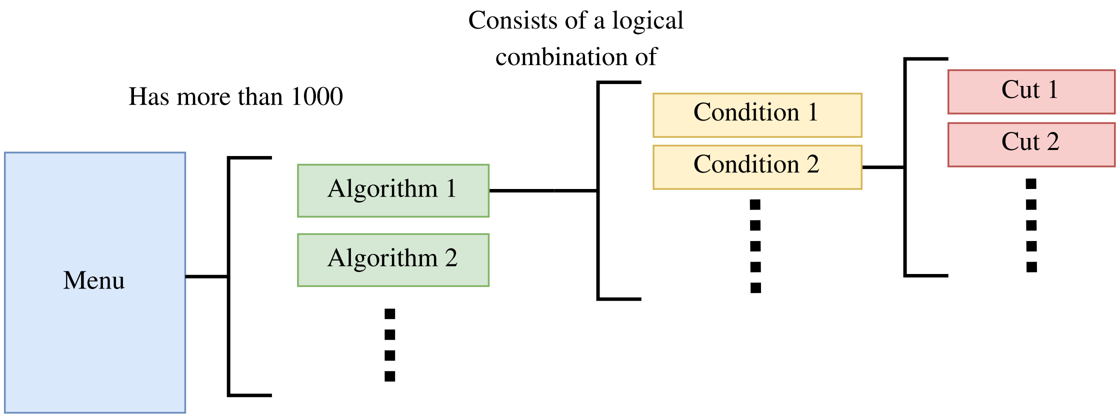


Figure 2. Schematic drawing of the menu building blocks. Conditions are combined with logical operations ("and", "or", "not") to form algorithms, while conditions themselves consist of a defined number of cuts on various quantities.

In order to conduct simultaneous and efficient searches for numerous physics signatures within the extensive data produced at the CMS detector, it is necessary to partition the so called trigger menu into more manageable units that correspond to individual signatures or clusters of signatures (figure 2). These more manageable units are referred to as "algorithms." An algorithm is essentially a logical combination of conditions, where each condition enables a specific set of cuts on various quantities. These include greater or less than comparisons on individual quantities like p_T, η, ϕ [1] as well as more complex correlational cuts that involve calculations, such as invariant mass, ΔR, transverse mass, combined P_T and others. Currently cuts listed in tables 1–3 are implemented. The four condition types, single-, double-, triple- and quad-object condition differ by the number of objects to search for in any given bunch-crossing event. A triple-object condition consuming muons for example, would search for three muons with desired properties configured via its cut values.

Table 1. Currently implemented cuts on trigger object quantities by condition type. For double- triple- and quad-object conditions, these cuts can be individually configured for each trigger object to search for. Cuts on p_T, isolation/p_T and quality can also be specified η-region dependent.

Quantity	Single	Double	Triple	Quad
p_T	✓	✓	✓	✓
η	✓	✓	✓	✓
ϕ	✓	✓	✓	✓
z_0	✓	✓	✓	✓
isolation/p_T	✓	✓	✓	✓
$\lvert z_0 - z_{0,\mathrm{vtx}} \rvert$	✓	✓	✓	✓
quality	✓	✓	✓	✓

Table 2. Currently implemented correlational cuts by condition type. For triple- and quad-object conditions, correlational cuts can be applied to any double-object subset.

Correlational cuts on double-object subsets	Double	Triple	Quad
$q_1 = q_2$	✓	✓	✓
$q_1 \neq q_2$	✓	✓	✓
$\Delta\eta$	✓	✓	✓
$\Delta\phi$	✓	✓	✓
$\lvert\Delta z_0\rvert$	✓	✓	✓
ΔR (= $\sqrt{\Delta\eta^2 + \Delta\phi^2}$)	✓	✓	✓
M (invariant mass)	✓	✓	✓
M_T (transverse mass)	✓	✓	✓
P_T (two particle transverse momentum)	✓	✓	✓
$M/\Delta R$ (invariant mass over ΔR)	✓	✓	✓

Table 3. Currently implemented 3-body cuts by condition type. For quad-object conditions, 3-body cuts can be applied to any triple-object subset.

3-body cuts on triple-object subsets	Triple	Quad
M (=$\sqrt{M_{1,2}^2 + M_{1,3}^2 + M_{2,3}^2}$)	✓	untested
M_T (=$\sqrt{M_{T,1,2}^2 + M_{T,1,3}^2 + M_{T,2,3}^2}$)	✓	untested

Only a subset of cuts listed in tables 1–3 were utilised by a simplified menu for Phase-2 [3], designed to cover most of the existing CMS physics program. Nevertheless, it has been ensured that all cuts present in the current LHC Run-3 system [6] are likewise available in the new Phase-2 system, albeit with slightly different implementations, enabling a more extensive menu for Phase-2 to uphold and possibly improve the existing physics coverage.

2.1 Adaptability

There are four handles available to configure cut-based algorithms.

- The trigger objects used (electrons, track-matched muons, photons, etc.).
- The condition types involved (single-, double-, triple-, quad-object).
- The logical expression combining multiple conditions.
- The cuts set on each individual condition.

To abstract those handles, separate Very High-Speed Integrated Circuit Hardware Description Language (VHDL) modules are used for each of the four condition types. The resulting condition decisions are logically combined with the decisions of other condition modules to form algorithm results. It’s worth noting here that algorithms comprising only one condition are also permitted. Due to our approach of converting all upstream trigger objects to an internal superset object [5], the selection of the trigger objects can be performed by simply passing generics to the VHDL condition module. The essential aspect here is to ensure that the condition modules are highly adaptable, accommodating all possible combinations of cuts and trigger objects. This adaptability is achieved by using arrays as generic inputs to the VHDL condition modules for each cut. These arrays facilitate single cuts on individual object quantities as well as correlational cuts via a correlation indexing scheme. For certain cuts like η-regional ones, we are leveraging "unbound arrays as generics", a feature of VHDL-2008 [7]. This ensures that any number of η-regions can be configured.

2.2 Calculation accuracy

A notable challenge arose in the calculation of the invariant mass when attempting to achieve acceptable relative accuracy throughout the entire phase space while adhering to available latency constrained computation techniques. The issue stems from the vast discrepancy in magnitudes between the mathematical functions involved: $\cos(\Delta\phi)$, where $\Delta\phi \in [0, 2\pi]$, is confined to the range of [-1, 1], while $\cosh(\Delta\eta)$, with $\Delta\eta \in [0, 4\pi)$, can be five orders of magnitude larger ($\cosh(4\pi) \approx 143\,376$). To strike a balance between minimizing latency and conserving FPGA resources, we employ Look-Up Tables (LUTs) for both $\cos(\Delta\phi)$ and $\cosh(\Delta\eta)$ computations. Additionally, to limit the number of Digital Signal Processor (DSP) tiles used, both LUTs share the same scale factor C_s, allowing this factor to be absorbed into the cut value. Adhering to these restrictions while maintaining acceptable relative accuracy within different orders of magnitude, we thus had to introduce a split into two distinct calculation regimes at $\Delta\eta = 2\pi$ ($\cosh(2\pi) \simeq 268$), where $\cos(\Delta\phi)$ can be neglected in the upper $\Delta\eta$-regime (figure 3).

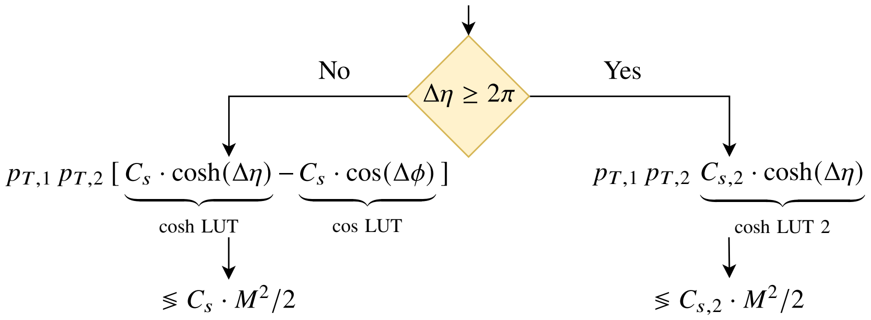

No
$\Delta\eta \geq 2\pi$
Yes
$p_{T,1}\, p_{T,2}\, [\, C_s \cdot \cosh(\Delta\eta) - C_s \cdot \cos(\Delta\phi)\,]$
cosh LUT
cos LUT
$\lessgtr C_s \cdot M^2/2$
$p_{T,1}\, p_{T,2}\; C_{s,2} \cdot \cosh(\Delta\eta)$
cosh LUT 2
$\lessgtr C_{s,2} \cdot M^2/2$

Figure 3. Flowchart of the invariant mass calculation split at 2π ($\cosh(2\pi) \simeq 268$).

The maximum relative invariant mass error in the transition region, resulting from the omission of $\cos(\Delta\phi)$ for $\Delta\eta \geq 2\pi$ can be calculated as

$$\frac{\Delta M_{\text{err}}}{M}(\Delta\eta = 2\pi, \Delta\phi = 0) = \sqrt{\frac{\cosh(2\pi) - [\cosh(2\pi) - \cos(0)]}{\cosh(2\pi) - \cos(0)}} = 6.1\,\%\,. \tag{2.1}$$

3 Hardware implementation

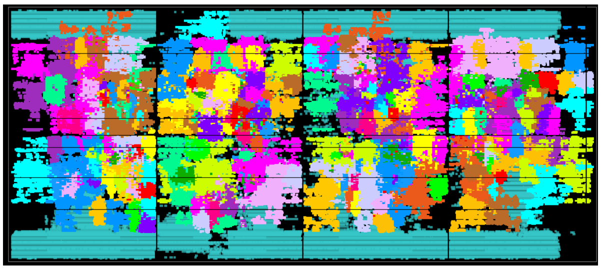
We opted to implement the algorithm part of the Global Trigger firmware to operate at 480 MHz, a frequency that is twelve times the LHC clock of 40 MHz. This choice was made in order to effectively reuse comparator and calculation logic for the up to twelve objects in each trigger object collection that we receive per bunch-crossing event. Consequently, the primary challenge encountered, is ensuring that the design maintains routability at this high clock frequency across all possible configurable cut combinations. To address this challenge we implemented an efficient pipelining strategy, ensuring that a limited number of operations are performed within each clock cycle. In particular, we want to highlight two of our pipelining techniques used throughout the firmware design.

To enable data parallelization for correlating a streamed and a parallel trigger object collection, our first step involves successively populating a buffer with all the objects from one of the collections. Once we have processed the entire collection, we transfer the data to a second buffer, where it will remain for the subsequent 12 clock cycles, allowing us ample time to perform all correlations with the 12 objects in the streamed collection.

A second technique involves our method of combining cut decisions as soon as they are ready. This is achieved by first building a decision tensor of rank N, where N corresponds to the number of objects to search for (i.e. for the double-object condition $N = 2$). Once this tensor is available we can execute logical "AND" operations to aggregate the results of individual comparisons and calculations, effectively accommodating latency differences. A collapse of this tensor by executing a logical "OR" of all elements, yields the final condition decision. In certain scenarios, this collapse may be distributed across multiple clock cycles to improve the routability of the design.

4 Prototypes

The CMS Collaboration has designed new generic ATCA boards leveraging state-of-the-art Xilinx Ultrascale+ FPGAs, intended to accommodate various trigger subsystem. Our target board for the final algorithm implementation is designated to be one such board, the Serenity [8], equipped with a Xilinx VU13P FPGA. At the time of writing, one such board is available in the CMS integration facility for which we have successfully tested a firmware build with 368 individual algorithms (figure 4). In this design, each cut (cf. section 2) appears in at least eight different algorithms, with the exception of 3-body cuts. Furthermore, we also implemented similar designs with fewer algorithms on Serenity boards equipped with a Xilinx VU9P and KU15P. The former design is mainly maintained for tests involving multiple boards within the CMS integration facility due to the limited availability of boards featuring a VU13P. Meanwhile, the latter design is planned for an LHC Run-3 test at LHC Point 5 (CMS) together with parts of the already installed Phase-2 muon trigger system.



Resource usage:

LUTs: 33.9 % abs.

Flip-Flops: 36.6 % abs.

Block-RAMs: 87.4 % abs.

DSPs: 52.8 % abs.

Figure 4. Floor-plan of the Global Trigger firmware implemented on a Xilinx VU13P FPGA. The cyan blocks at the edges correspond to the Extensible, Modular (data) Processor (EMP) framework [9] provided by the Serenity developers, while each color in the center represents one of 368 implemented algorithms.

5 Summary

The development of the entirely redesigned Phase-2 Global Trigger firmware is steadily progressing, already incorporating a large variety of individual cuts, both on single quantities and correlations. Simultaneously, it retains its high adaptability, allowing these cuts to be utilized in various combinations, all while ensuring the design's routability. This is exemplified by a prototype featuring 368 implemented algorithms, requiring only three ATCA boards for the cut-based menu of more than 1000 algorithms while leaving nine boards for advanced algorithms featuring novel techniques like neural networks [3].

References

[1] CMS Collaboration, *The CMS experiment at the CERN LHC*, Journal of Instrumentation 3 (2008) S08004.

[2] CMS Collaboration, *Development of the CMS detector for the CERN LHC Run 3*, submitted to JINST (2023), arXiv:2309.05466, `https://cds.cern.ch/record/2870088`

[3] CMS Collaboration, *The Phase-2 Upgrade of the CMS Level-1 Trigger*, CERN-LHCC-2020-004, CMS-TDR-021 (2020).

[4] M. Jeitler, *The Phase-2 Upgrade of the Hardware Trigger of CMS at the LHC*, JINST 15 C09009 (2020).

[5] H. Sakulin et al., *Architecture and prototype of the CMS Global Level-1 Trigger for Phase-2*, JINST 18 C01034 (2023).

[6] J. Wittmann et al., *Design and performance of the phase I upgrade of the CMS Global Trigger*, JINST 12 C01046 (2017).

[7] *1076-2008 - IEEE Standard VHDL Language Reference Manual*, IEEE Std 1076-2008 (2009).

[8] A. Rose et al., *Serenity: An ATCA prototyping platform for CMS Phase-2*, PoS TWEPP2018 (2019) 115.

[9] Serenity Collaboration, *EMP framework*, `https://serenity.web.cern.ch/serenity/emp-fwk`