Cluster reconstruction in the HGCAL at the Level 1 trigger

Bruno Alves[1,*], for the CMS Collaboration

[1]Laboratoire Leprince-Ringuet, École Polytechnique, Institut Polytechnique de Paris, Palaiseau, France

Abstract. The CMS collaboration has chosen a novel High Granularity Calorimeter for the endcap regions as part of its planned upgrade for the High Luminosity LHC. The calorimeter will have fine segmentation in both the transverse and longitudinal directions, and its data will be part of the Level 1 trigger of the CMS experiment. The trigger has tight constraints on latency and rate, and will need to be implemented in hardware. The high granularity results in around six million readout channels in total, reduced to one million that are used at 40 MHz as part of the Level 1 trigger, presenting a significant challenge in terms of data manipulation and processing; the trigger data volumes will be an order of magnitude above those currently handled at CMS. In addition, the high luminosity will result in an average of 140 (or more) interactions per bunch crossing. This leads to a huge rate by background processes which must be efficiently rejected by the trigger algorithms. Furthermore, reconstruction of the particle clusters to be used for particle flow in events with high hit rates is also a complex computational problem for the trigger. The status of the cluster reconstruction algorithms developed to tackle these major challenges, as well as the associated trigger architecture, is presented. Methods developed to mitigate the known issue of cluster splitting are described, incuding an iterative algorithm which has no impact on firmware resources.

1 The High Granularity Calorimeter

The High Luminosity LHC (HL-LHC) will start taking data in 2029, achieving unprecedented instantaneous luminosities of $\sim 5 \times 10^{34}\,\mathrm{cm}^{-2}\,\mathrm{s}^{-1}$ (more than twice of LHC's current value) and a pile-up (PU) of up to 200 (currently ~50 on average [1]). An integrated luminosity of $\sim 3\,\mathrm{ab}^{-1}$ should be reached over a period of 10 years [2], while current CMS endcap calorimeters are designed to sustain up to $500\,\mathrm{fb}^{-1}$. A dramatic performance degradation is foreseeable for higher luminosities. CMS [3] is thus developing the High Granularity Calorimeter (HGCAL) [4]: a novel endcap sampling calorimeter with an extremely fine granularity. About 6 million channels will enable particle identification and high resolution measurements of the position, energy and time of high-energy collision products. The proposed design of HGCAL includes two sections measuring the properties of different types of particles. The electromagnetic (EM) section covers the first 26 layers, closer to the interaction point. There, silicon (Si) sensors organized in hexagonal modules act as active material, in order to sustain the expected radiation. Layers of Si are interleaved with absorbers. The following 21 layers, comprising the hadronic (HAD) section, are split into 8 Si-only layers

*e-mail: bruno.alves@cern.ch

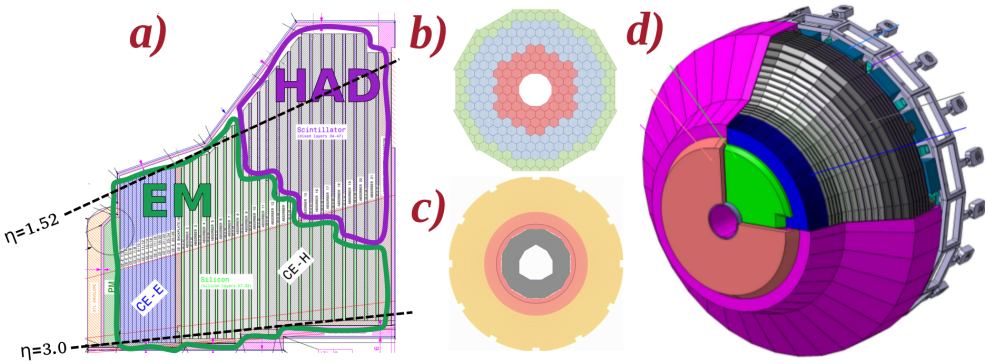

Figure 1. Schematic views of HGCAL. *a)* Longitudinal profile of positive endcap with highlighted η range and EM and HAD sections *b)* Transversal view of a Si-only layer, with different colors representing different sensor thicknesses *c)* Same as *b* for a hybrid layer *d)* 3D view of HGCAL.

followed by 14 hybrid layers, with Si closer to the beamline and cost effective plastic scintillator at lower pseudorapidity (η) values (see fig. 1). Each endcap weighs 215 t and measures 2 m (2.3 m) in the longitudinal (radial) direction. HGCAL will be integrated with the online firmware trigger system put in place by CMS, the Level-1 (L1) [5], which precedes the High-Level Trigger (HLT) running on standard servers. L1 performs an online selection of interesting physics processes, whose cross sections are typically orders of magnitude lower than the total proton-proton cross section.

These proceedings describe the HGCAL L1 reconstruction chain, from raw energy deposits to the creation of trigger primitives (TPs), which are detector-specific inputs to the L1. Special emphasis is placed on the study of cluster splitting, which represents a known and so far unstudied shortcoming of the TP chain. The azimuthal angle (ϕ) in the transverse plane, the radial coordinate R and the z-axis lying parallel to the beam-line form the binned projective (ϕ, R/z) coordinate system used in this work [3]. Since $\tan(\theta) = R/z$, where θ is measured from the z-axis, for a constant angle θ corresponds a constant R/z. Energy deposits of neutral particles spanning several layers will thus lie in a single R/z bin.

2 The dataflow of HGCAL trigger primitives

TPs are detector-specific quantities preceding and serving as input to the CMS L1 trigger, hence the name. In HGCAL, they consist on coarse (η, ϕ) towers and fine-grained cluster-related variables, such as energy, positions and shapes. TPs provide valuable information to L1 at 40 MHz within limited time and bandwidth budgets. Specifically, L1 has an allocated latency of 12.5 µs, ~5 µs of which for creating HGCAL TPs. The generation of TPs spans several data processing steps (fig. 2) running on the front-end (FE) chips and in the back-end (BE) electronics [4]. Data throughput is reduced as much and as soon as possible, and pipelined algorithms are exploited whenever feasible.

In the FE, dedicated radiation-hard read-out chips (HGCROCs) [6] measure and digitize the ionization and scintillation signals at ~100 TB s^{-1} [4] in a power-, cost- and space-constrained environment [6]. Each HGCROC linearizes the deposited charge in the Si and scintillator sections. It then reduces the prohibitive data throughput by grouping channels into 4 or 9 trigger cells (TCs), with 48 TCs per Si module. Only odd layers in the EM section are used for the trigger for further bandwidth reduction. The HGCROC finally compresses

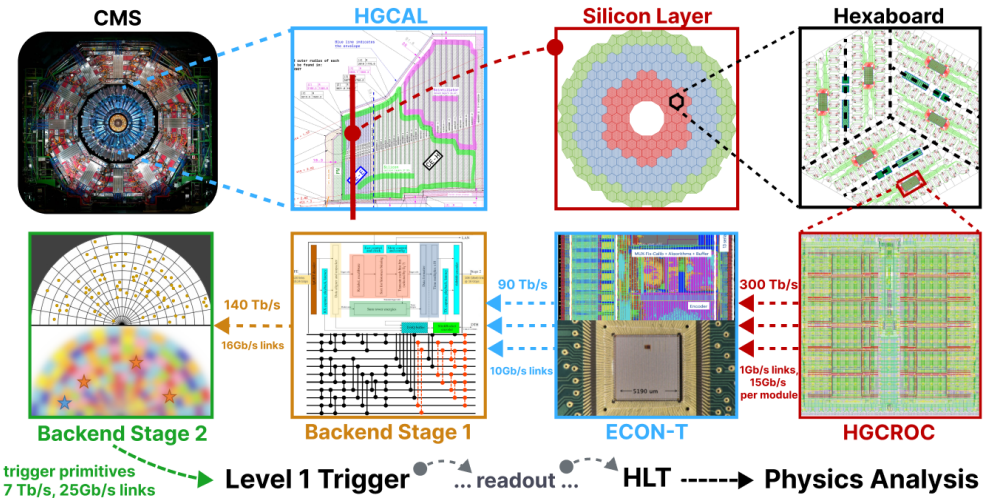

Figure 2. Simplified schematic of TP processing in HGCAL, following the data flow in a Si layer through the FE and BE up to L1, including expected approximate bandwidths. Trigger decisions at this stage will impact the HLT and, consequently, physics analysis.

energy values to a 7-bit floating point representation, and provides a measurement of the time of arrival of pulses down to a few tens of ps. However, time information cannot be exploited in the trigger path due to bandwidth constraints. The TC data is then sent to the on-detector ECON-T chip via 1.28 Gbit s^{-1} e-links (see, for instance, [7]). Each ECON-T processes data coming from a single module (3 or 6 HGCROCs). Dedicated algorithms reduce TC data, either by aggregation or by selection: all TCs above a certain energy threshold or only the most energetic ones. The ECON-T also builds *module sums*, where the energies of TCs in a module are summed without applying any threshold. The remaining data is sent via e-links to lpGBT ASICs [8], serialized to 10.24 Gbit s^{-1}, and sent via optical-links [9] to the off-detector BE.

The BE is composed of two processing stages, both running on Serenity boards with 128-transceivers Xilinx VU13P FPGAs. Their assigned latency budget is ~2.5 µs. FPGAs in Stage 1 (S1) cover ~2 % only of one endcap and, just like Stage 2 (S2) boards, do not communicate with eachother. Handling boundaries thus requires data duplication. The S1 receives ECON-T data, unpacks and calibrates it, and routes and sorts TCs in energy into projective 2 ϕ vs. 42 R/z bins per 120° sector. The sorting uses batcher odd-even sorting networks [10–12], where on-the-fly truncation reduces the total number of firmware comparators required. Modules sums are here partially summed into module towers, and time multiplexing [13] with a 18 bunch-crossing period is applied before sending the data to S2. The latter enables more time for single event processing at S2. The S2 unpacks the data from a single bunch-crossing coming from S1. It accumulates partial tower energies into (η, ϕ) bins and it builds clusters, where the following pipelined steps are run (fig. 3):

- **Histogramming**: TCs passing the ECON-T selection are mapped to a (ϕ, R/z) space with (216, 42) bins. This further reduces spatial granularity and facilitates vectorized/parallel processing in the firmware due to its grid-like structure. A histogram is constructed where each bin contains the energy sum of all its TCs, together with their mip_T[1]-weighted x/z and y/z positions.

[1] $\text{mip}_\text{T} := \text{mip} / \cos(\theta)$, where one mip is the energy deposited by a minimum ionizing particle [14, §34.2.3].

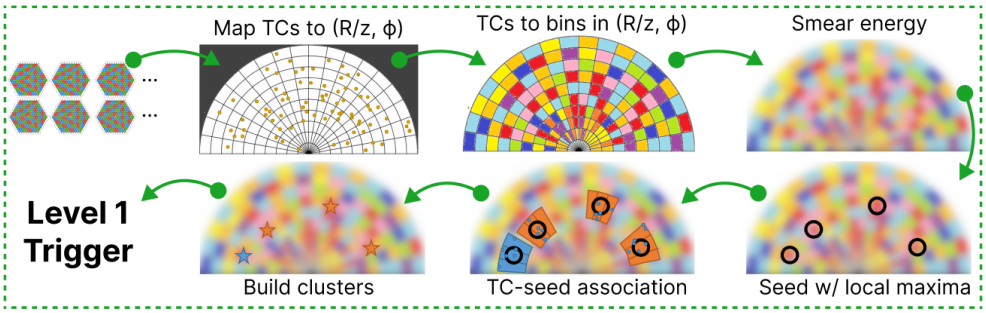

Figure 3. Schematic flowchart of S2's reconstruction chain. TCs from S1 are unpacked and processed in a pipelined fashion up to the creation of cluster-related variables, which are fed to L1. The description of the steps can be found in the text.

- **Smoothing**: An energy smoothing step is applied to (ϕ, R/z) bins to decrease overall variations in their energy distribution. A *smoothing kernel* is applied to all bins, where to each bin's energy a fraction of the energy of its neighbours is added. The fraction decreases with distance. The kernels are shown in eq. (1), along ϕ (left) and R/z (right):

$$\left[\cdots \quad \tfrac{1}{16} \quad \tfrac{1}{8} \quad \tfrac{1}{4} \quad \tfrac{1}{2} \quad 1 \quad \tfrac{1}{2} \quad \tfrac{1}{4} \quad \tfrac{1}{8} \quad \tfrac{1}{16} \quad \cdots \right] \qquad \begin{bmatrix} \frac{1}{2} \\ 1 \\ \frac{1}{2} \end{bmatrix} \tag{1}$$

  Variations are more proeminent along ϕ since the binning is finer. The kernel along ϕ is thus R/z-dependent, as illustrated by the dots in eq. (1). The ϕ kernel collects the energy from more bins for lower R/z rows. The energy of each bin is normalized after applying the kernel, to ensure no energy is artificially added to the event.

- **Seeding**: Seeds are local mip_T maxima in the histogram. They are found using a *seeding window* which, for each bin, spans immediately adjacent bins and checks whether their mip_T energy is lower. If it is, and if its energy lies above a threshold, the bin is promoted to a seed.

- **Clustering**: TCs are associated to seeds and used to calculate cluster properties. Every seed originates a cluster. Contrary to previous steps, the clustering uses a (x/z, y/z) projective space. Two clustering algorithms are defined, one associating TCs to their closest seed, the other prioritizing association based on seed energy. The former is used in this work.

The S2 reconstruction, previously only available in C++ within CMSSW [15], has been ported to a standalone `Python` code[2]. It enables exponentially faster prototyping, testing and optimization, and it includes event displays supported by a simplified version of HGCAL's geometry. It was used for all studies that follow.

3 Cluster splitting

The projective (ϕ, R/z) bins do not have a fixed size in the (x, y) plane. Bins located at higher (lower) R/z values will be physically larger (smaller), their size being inversely proportional to the expected occupancy. Due to the lack of alignment of detector elements with (ϕ, R/z)

[2]https://github.com/bfonta/bye_splits

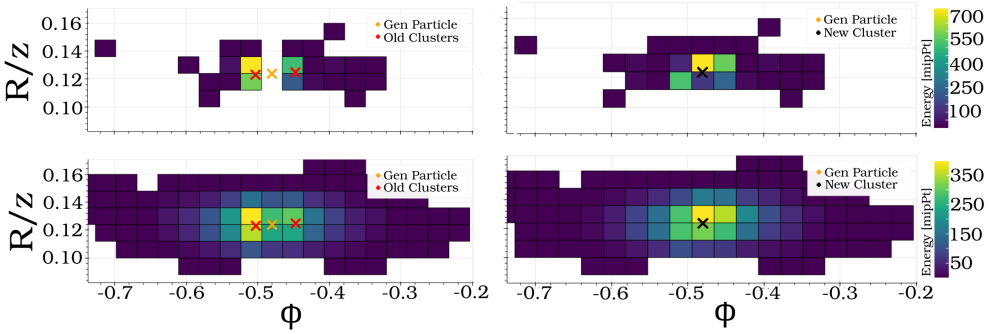

Figure 4. Cluster split example for a single event in the (ϕ, R/z) space, where colors represent energy deposited per bin in mip_T units. The orange cross shows the position of the generated photon. The top (bottom) row shows the same event before (after) applying the smoothing step. The left (right) column displays the event not considering (considering) the `bye-splits` algorithm, where the red (black) crosses show the position of the reconstructed clusters. Generated and reconstructed clusters become superimposed after running `bye-splits`.

bins, there are stark differences in TC multiplicities between adjacent bins along ϕ. In other words, the assignment of TCs to bins is non-uniform. This introduces nonphysical biases, since the distribution of deposited energy in (ϕ, R/z) bins might not follow the one in the detector. In fact, single particles occasionally deposit their energy such that two energy maxima along ϕ can be observed (left column of fig. 4). This happens due to the lack of TCs in the intermediate ϕ bin. When the seeding step is run on these events, two seeds are found. These events are referred as *cluster splits*, since they artificially originate more than one cluster per particle. They are overwhelmingly located in the high η (low R/z) region, where bins are smaller and TC counts are less homogeneous along ϕ. A degradation of the detector's energy response and position resolution is expected.

To study this effect, we use generated unconverted photons without pile-up, with a uniform transverse momentum (p_T) distribution between 0 and 100 GeV. We consider only events where a split very likely happens by requiring an energy response of $(E_\text{Cluster} - E_\text{Gen})/E_\text{Gen} < -0.35$, where the cut captures events forming a peak at around $-0.5/-0.6$ (around half the energy is reconstructed). Only photons with $\eta \in [1.7, 2.8]$ are retained, avoiding HGCAL boundaries, where showers might be transversally cut. Within this sample, around ~1 % of events suffer from cluster splitting.

3.1 Bye-splits iterative algorithm

In order to remove cluster splits, we introduce the `bye-splits` algorithm, which is run independently for each R/z row. Its goal is to modify the mapping of TCs to bins along ϕ. This is done to reduce the variance of the number of TCs per bin, and consequently the number of splits. A sliding window is defined around three consecutive ϕ bins. The algorithm computes, for each group of three bins, the differences $D_\text{left} = C_2 - C_1$ and $D_\text{right} = C_3 - C_2$ between their TC counts C, where the indexes 1 to 3 refer to the left, middle and right bin positions in the sliding window. A pseudo-random number x is sampled from an uniform distribution $\mathcal{U}(0, 1)$ to decide whether the TC position migration occurs on the left or right

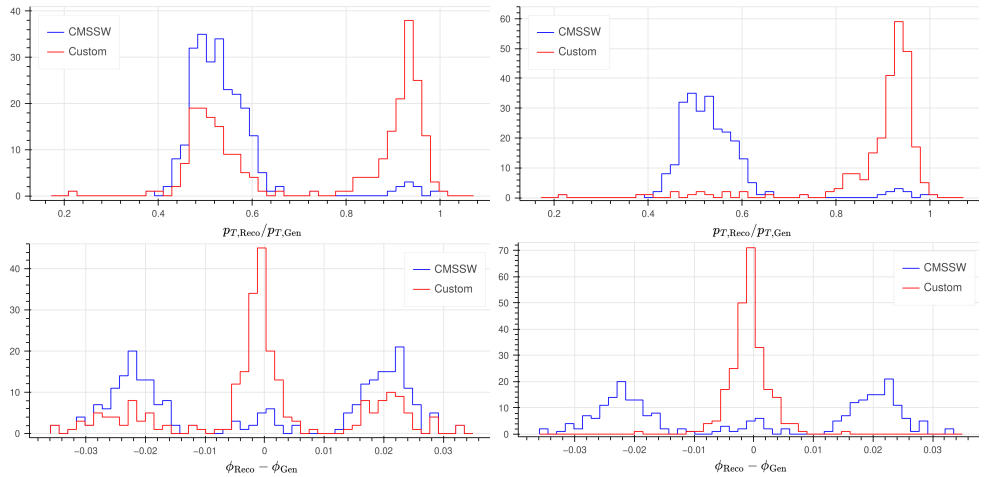

Figure 5. Default (blue, labeled CMSSW after CMS' software [15]) and custom (red) reconstructions. All events displayed satisfy $(E_{\mathrm{Cluster}} - E_{\mathrm{Gen}})/E_{\mathrm{Gen}} < -0.35$ before running the `bye-splits` algorithm. The top (bottom) row shows event multiplicities for the p_{T} response (ϕ resolution), and the left (right) column includes results after running `bye-splits` (seeding step with a window of size 2 along ϕ).

side of the window:

$$\mathrm{Side} = \begin{cases} \text{left}, & \text{if } x \sim \mathcal{U}(0,1) < \frac{D_{\mathrm{left}}}{|D_{\mathrm{left}}|+|D_{\mathrm{right}}|} \\ \text{right}, & \text{otherwise} \end{cases} \tag{2}$$

The randomness in eq. (2) ensures the overall shape of the distribution of TC counts along ϕ is kept, while setting TC migrations on one side to be more likely when differences are larger. Once a side has been chosen, the shift of a TC is done by taking into account the relative distribution of TC counts in the sliding window (there are four possibilities for a sliding window of size 3). TC bin migrations should be minimized, with only one ϕ bin shift per iteration. Indeed, despite wanting to reduce the splitting, the final mapping should still reflect the original physical distribution of TCs. After the shift, the sliding window moves with unitary stride. The algorithm is run for all possible windows, forming one *epoch*. Circular boundary conditions are taken into account. After each epoch, the following termination condition is checked for every ϕ bin i:

$$\left|D_{\mathrm{left},i}\right| + \left|D_{\mathrm{right},i}\right| \leq \max\left\{1, \lambda \times \left(\left|D^0_{\mathrm{left},i}\right| + \left|D^0_{\mathrm{right},i}\right|\right)\right\} \tag{3}$$

where $\lambda \in [0,1]$ is a tunable parameter controlling the final TC count variance, and D^0 refers to the differences before the algorithm was run. The max operator ensures convergence for low-λ (more aggressive) runs. For $\lambda = 0$, we verify that all TCs move less than 2 cm along ϕ, which implies they moved to their immediately adjacent bins. The algorithm removes a significant portion of splits, and in fig. 4 we visualize one such event. Significant improvements in energy response and position resolution are obtained (fig. 5). We have also validated `bye-splits` by verifying that it does not impact the reconstruction of a sample where no splits are present. Importantly, `bye-splits` is run offline, decoupled from the online firmware reconstruction, and its TC-to-bin output mapping can be encoded in a Look-Up Table (LUT). This implies that `byes-splits` does not impact firmware resources.

3.2 Alternative approaches to mitigate cluster splitting

We propose alternative approaches which, despite requiring more resources, are simple to implement. We expect their usage to be considered once available resources are clearly defined. These methods achieve better results than the resource-agnostic `bye-splits` algorithm.

- **Size 2 seeding window**: we consider the seeding step with a window of size 2 along ϕ. The window has access to 14 neighbours instead of 8. As expected, given the nature of cluster splitting, virtually all splits are removed (bottom row of fig. 5). However, each window requires six additional firmware comparators, increasing resource consumption.
- **Flat smoothing kernel**: we keep the standard size 1 seeding window and change instead the smoothing step that is executed immediately before. The current smoothing kernel applies a larger weight to the central bin. It is thus natural to consider a "flat" weigth scheme around the central bin to reduce the number of splits. This is equivalent to share the bin energy between central bins. We choose the following kernel along ϕ:

$$\left[\ \ldots \quad \tfrac{1}{8} \quad \tfrac{1}{4} \quad \tfrac{1}{2} \quad 1 \quad 1 \quad 1 \quad \tfrac{1}{2} \quad \tfrac{1}{4} \quad \tfrac{1}{8} \quad \ldots \ \right] \tag{4}$$

  which can still be implemented in the firmware using powers of 2. The width of the kernel is R/z-dependent, following what was done for the default kernel. The new kernel strongly reduces splits, having an effect extremely similar to the size 2 seeding window. Contrary to that method, however, changing the smoothing kernel does not impact firmware resources, as long as the size of the kernel for each R/z row remains constant.
- **Energy prioritization**: we run the reconstruction chain with the energy prioritization clustering algorithm. The latter associates TCs to the most energetic seed, after the seeding step takes place. TCs are additionally selected within a given radius. By construction, the method concentrates TCs into the same cluster, strongly mitigating cluster splits in samples without pile-up and for large enough cluster radiae.
- **Maximum shower region**: we assess the impact of considering only the region of the detector where the largest fraction of energy is deposited. We run `bye-splits` on TCs between the 8^{th} and 15^{th} layers, where EM showers usually display their longitudinal maxima. We observe the results to be identical to the ones using all TCs. This suggests that, at least for EM showers, the optimization of FPGA resource consumption is possible via the development of future algorithms focusing on specific detector regions.

4 Conclusions and Next Steps

In the context of the HL-LHC, we presented the HGCAL TP reconstruction chain, as currently planned to be used by the CMS online L1 trigger system. It includes multiple algorithms which development is driven by physics results and firmware resource constraints. The BE S2 results were obtained using a dedicated simulation of the algorithms to be deployed on firmware. The simulation allows quick prototyping, testing and optimization. We gave particular emphasis on fixing cluster splits, and presented several methods to do so.

The `bye-splits` and alternative algorithms presented in sections 3.1 and 3.2 were so far tested with photons only, and without considering PU. More realistic scenarios are already under study, namely considering 200 PU, where kinematical distributions and data throughput are expected to be very different. In addition, we are starting the process of systematically assessing the impact of the algorithms when reconstructing hadrons and long-lived particles. This will require mixing the information coming from the EM and HAD sections of the detector, and might imply the adaptation and optimization of current algorithms. To give an

example of differences we might encounter, tau leptons naturally create cluster splits (one and three pronged decays). An additional way to remove cluster splits and possibly simplify channel routing from bins to TCs is to use detector coordinates directly, instead of (ϕ, R/z) bins. Near future steps also include algorithms where only specific detector regions are looked at, to further decrease the amount of data processed, and the development of 2D and 3D event displays for quick inspection of single events.

One of the major challenges of TP reconstruction is implementing the simulated algorithms in firmware. The latter has clearly defined resources and data bandwidth constraints. We are therefore in the process of porting some of the BE algorithms to firmware to assess their feasibility in the demanding HL-LHC environment.

References

[1] CMS Collaboration, *Public CMS Luminosity Information* (2023), Twiki. Accessed: 2023-08-18, https://twiki.cern.ch/twiki/bin/view/CMSPublic/LumiPublicResults

[2] O. Aberle, I. Béjar Alonso, O. Brüning, P. Fessia, L. Rossi, L. Tavian, M. Zerlauth, C. Adorisio, A. Adraktas, M. Ady et al., *High-Luminosity Large Hadron Collider (HL-LHC): Technical design report*, CERN Yellow Reports: Monographs (CERN, Geneva, 2020), https://cds.cern.ch/record/2749422

[3] CMS Collaboration (CMS), JINST **3**, S08004 (2008)

[4] CMS Collaboration, Tech. Rep. CMS-TDR-019, CERN (2018), https://cds.cern.ch/record/2293646

[5] CMS Collaboration, Tech. Rep. CMS-TDR-021, CERN (2020), https://cds.cern.ch/record/2714892

[6] F. Bouyjou, G. Bombardi, F. Dulucq, A.E. Berni, S. Extier, M. Firlej, T. Fiutowski, F. Guilloux, M. Idzik, C.D.L. Taille et al., Journal of Instrumentation **17**, C03015 (2022)

[7] N. Strobbe, *The overall electronics chain (powering and readout) of the CMS HGCAL* , https://indico.cern.ch/event/847884/contributions/4833234/ (2022), CALOR 22

[8] lbGBT team, *The lpgbtv1 manual*, https://lpgbt.web.cern.ch/lpgbt/

[9] J. Troska et al., PoS **TWEPP-17**, 048 (2017)

[10] K.E. Batcher, *Sorting Networks and Their Applications*, in *Proceedings of the April 30–May 2, 1968, Spring Joint Computer Conference* (1968), AFIPS '68 (Spring), p. 307–314, ISBN 9781450378970

[11] I. Skliarova, Electronics **11**, 1029 (2022)

[12] L. Portalès, Instruments **6** (2022)

[13] A. Zabi, Habilitation à diriger des recherches, Laboratoire Leprince Ringuet Ecole Polytechnique (2016), https://hal.science/tel-03030251

[14] R.L. Workman, Others (Particle Data Group), PTEP **2022**, 083C01 (2022)

[15] K. Bloom, Proceedings of 38th International Conference on High Energy Physics (2017)