# Readout Firmware of the Vertex Locator for LHCb Run 3 and Beyond

Karol Hennessy, Antonio Fernández Prieto, Pablo Vázquez Regueiro, Jan Buytaert, Martin Van Beuzekom, Edgar Lemos Cid, Lars Eklund, Kristof de Bruyn, Sneha Naik, Manuel Schiller, Dónal Murray, Alexander Leflat, Giovanni Bassi, Giovanni Punzi, Federico Lazzari, Michael J. Morello, Oscar Boente García, Abraham Gallas Torreira, Beatriz García Plana, Themis Bowcock, Francesco Dettori, Karlis Dreimanis, Vinicius Franco Lima, David Hutchcroft, Kurt Rinnert, Tara Shears, Oscar Augusto, Victor Coco, Paula Collins, Tim Evans, Massi Ferro-Luzzi, Heinrich Schindler, Kazu Akiba, Elena Dall' Occo, Cristina Sanchez Graz, Wouter Hulsbergen, Daniel Hynds, Igor Kostiuk, Marcel Merk, Aleksandra Snoch, Dana Seman Bobulska, Silvia Borghi, Stefano de Capua, Deepanwita Dutta, Marco Gersabeck, Chris Parkes, Peter Svihra, Mark Williams, Galina Bogdanova, Vladimir Volkov, Pawel Kopciewicz, Maciej Majewski, Agnieszka Oblakowska-Mucha, Bartlomej Rachwal, Tomasz Szumlak, Lucas Meyer Garcia, Franciole Marinho, Larissa Helena Mendes, Irina Nasteva, Juan Otalora, Gabriel Rodrigues, Jaap Velthuis, Pawel Jalocha, Malcolm John, Nathan Jurik, Luke Scantlebury-Smead, John Back, Tim Gershon, Tom Latham, and Andrew Morris

***Abstract*—The new LHCb Vertex Locator (VELO) for LHCb, comprising a new pixel detector and readout electronics, will be installed in 2021 for data taking in Run 3 at the LHC. The electronics centers around the "VeloPix" ASIC at the front-end operating in a trigger-less readout at 40 MHz. A custom serializer, called gigabit wireline transmitter (GWT), and associated custom protocol have been designed for the VeloPix. The GWT data are sent from the serializers of the VeloPix at a line rate of 5.12 Gb/s, reaching a total data rate of 2–3 Tb/s for the full VELO detector. Data are sent over 300-m optic-fiber links to the control and readout electronics cards for deserialization and processing in Intel Arria 10 FPGAs. Because of the VeloPix trigger-less design, latency variances up to 12 $\mu$s can occur between adjacent datagrams. It is therefore essential to buffer and synchronize the data in firmware prior to onward propagation or suffer a huge CPU-processing penalty. This article will describe the architecture of the readout firmware in detail with focus given to the resynchronization mechanism and techniques for cauterization. Issues found during readout commissioning, and scaling resource utilization, along with the their solutions, will be illustrated. The latest results of the firmware data-processing chain can be presented as well as the verification procedures employed in simulation. Challenges for the next generation of the detector will also be presented with ideas for a readout processing solution.**

Manuscript received November 9, 2020; revised February 15, 2021 and March 29, 2021; accepted May 23, 2021. Date of publication May 31, 2021; date of current version October 18, 2021. This work was supported in part by CERN and the National Agencies: CAPES, CNPq, FAPERJ, and FINEP (Brazil); in part by INFN (Italy); in part by NWO (Netherlands); in part by MEiN and NCN (Poland) under Grant UMO-2018/31/B/ST2/03998; in part by MSHE (Russia); in part by MICINN (Spain); and in part by STFC (U.K.).

Please see the Acknowledgment section of this article for the author affiliations.

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TNS.2021.3085018.

Digital Object Identifier 10.1109/TNS.2021.3085018

***Index Terms*—DAQ, firmware, LHCb, readout, vertex locator (VELO).**

## I. Introduction

THIS article describes the readout architecture of the LHCb vertex locator (VELO) [1]–[3] currently being constructed and commissioned for operation for LHC Run 3 in 2022. The VELO is a silicon hybrid pixel detector operating in vacuum and very close to the LHC beams (5.1 mm), and therefore must cope with a very high radiation environment (the maximum fluence is expected to be $8 \times 10^{15} \cdot 1\ \text{MeV} \cdot n_{\text{eq}}/\text{cm}^2$). Furthermore, LHCb will have no hardware trigger, and so the ASICs must readout every bunch crossing at the full LHC machine rate. A new front-end ASIC, VeloPix [4], was designed to meet these requirements. The VeloPix and its supporting readout electronics will be described. Some of the challenges to commission the full VELO high-speed readout system and the solutions developed are presented.

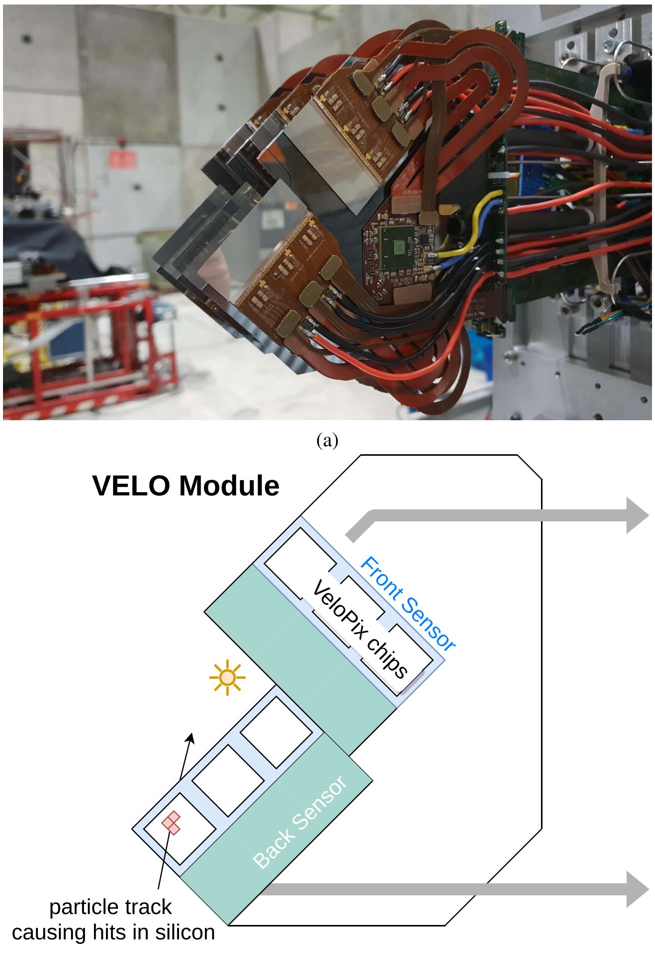
Several key features drive the design of the data-acquisition architecture. Foremost is the data rate-the expected data rates during nominal LHC conditions are summarized in Table I. Each VeloPix is equipped with up to four readout links enabled, running at a line-rate of 5.12 Gb/s. The particle hit rate for the VeloPix chips closest to the beam line is almost an order of magnitude greater than those at the exterior. Therefore, fewer links are enabled for those exterior chips, as the bandwidth requirements are less. A VELO module has twelve VeloPix chips and totals twenty readout links in the configuration (4,2,1,1,1,1,4,2,1,1,1,1).

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

TABLE I

PEAK DATA RATES CORRESPONDING TO THE VELOPIX ASIC WITH THE HIGHEST EXPECTED OCCUPANCY, AND THE RESULTANT BANDWIDTHS

| | |
|---|---|
| Peak hit rate | 900 Mhits/s/ASIC |
| Max data rate | 19.2 Gb/s |
| Total VELO | 2.85 Tb/s |



(a)

(b)

Fig. 1. (a) Photograph of three VELO Modules with the reflective sensor tiles seen to the left and the data tapes (in green) to the right. (b) Schematic showing the items relevant to the data acquisition.

Fig. 1(a) shows three prototype VELO modules and Fig. 1(b) shows a schematic of the components driving the data acquisition. The VELO modules operate in vacuum, and significant effort has been made to limit the amount of material in each module. This is done to minimize multiple scattering interactions that occur in the detector, which degrade the overall tracking and vertexing performance. Given the aforementioned bandwidth requirements, the end result is a thin detector producing a lot of heat with little place to go. $CO_2$ microchannel cooling [5] has been employed to extract the heat from the module. The VeloPix uses a custom transceiver, gigabit wireline transmitter (GWT) [6], which has been designed for low power to limit the heat budget used by the ASIC. GWT uses a custom protocol for data transmission. The VeloPix sends binary hit information (i.e., no measure of signal amplitude) in a data-driven mode. This means that hits above a configurable threshold produce data packets to be transmitted (and those below threshold are suppressed and yield no data). The chip is designed with a columnar readout path, where data fragments traverse the length of the column from their origin, are subsequently assembled at the end-of-column logic, and routed toward the center for transmission. This means that data fragments produced at the top and bottom of the pixel matrix will have significantly different readout latencies. Data fragments are timestamped such that they can be reordered in time at a later stage. Simulation studies [7] performed early in the design of the VeloPix showed that the bandwidth requirements of the chip could be reduced by approximately 30% by aggregating the data into groups of pixels ($2 \times 4$) called SuperPixels. If one or more pixels fire in the group, the binary hit information for the eight pixels is sent.

The high data rate, the custom GWT protocol, the time reordering of the data and the processing of the SuperPixels are the driving factors for real-time processing of the data using an FPGA. The platform used for LHCb is the PCIe40 [8]. A description of the readout board and the VELO firmware running therein, which is designed to meet these requirements using the PCIe40 architecture, is described in Section II.

## II. DESCRIPTION OF THE VELO DAQ ARCHITECTURE

The VELO detector consists of two sets of 26 VELO modules arranged horizontally around the LHC beam line. In terms of the data acquisition for the detector, it is sufficient to describe a readout slice comprising a single VELO module and its supporting hardware and firmware. Data from a VELO module are sent over flexible readout tapes to an optopower board on the exterior of the VELO vacuum tank. There, the data are converted to optical signals using CERN's versatile link modules [9]. The signals are sent over 300-m optical fiber to the PCIe40 readout board for processing. The processed data fragment is then sent to the LHCb event-building farm to be combined with data fragments from the other VELO modules and data fragments of all other detectors of LHCb to create complete events. These events are examined in software in the high-level trigger farm to search for interesting physics signatures.

The PCIe40, shown in Fig. 2, is (as the name suggests) a PCIe Gen3 card developed by the "Centre de Physique des Particules de Marseille" for LHCb. It is a single control and readout board for the entire experiment. The FPGA used is an Intel® Arria 10 (10AX115), and it has 48 bidirectional optical links and a PCIe bandwidth of approximately 100 Gb/s [10]. It can be used for timing, slow control, DAQ or all three at once-making it an excellent tool for lab use, obfuscating the need for multiple devices. The PCIe40 firmware [11] defines the functions of the device. The device is given a different moniker for each of its functions, depending on whether it is slow control, data acquisition, or timing and fast control (TFC). For the purposes of this article, only the data-acquisition variant (known as "TELL40") will be described, and that term will be used for both the firmware and the card itself.

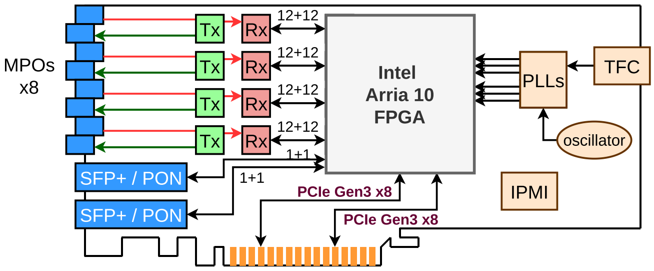
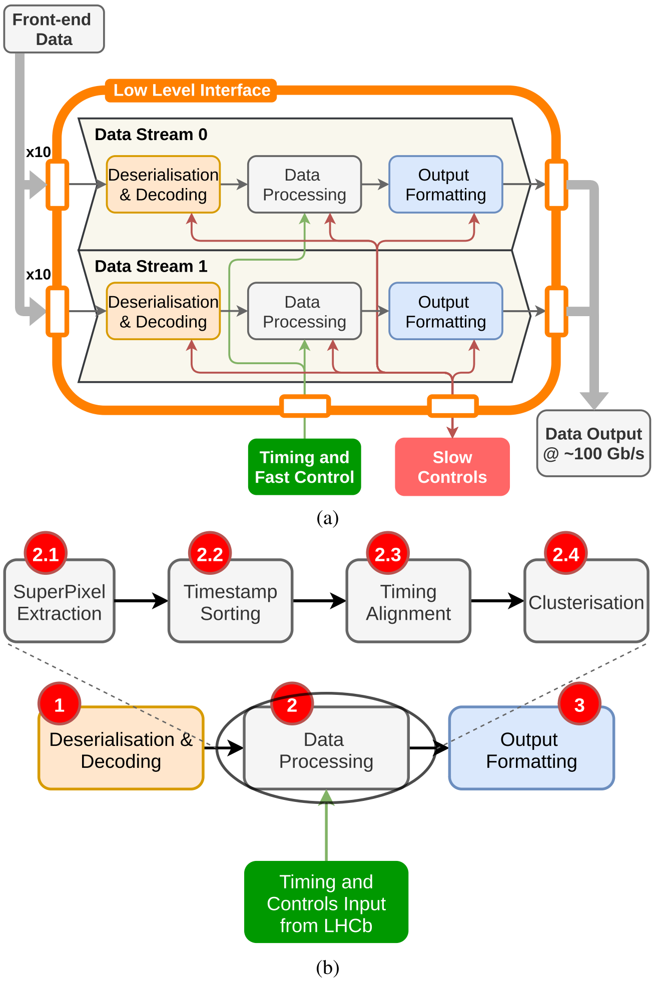


Fig. 2. Diagram of the PCIe40 board and its primary pathways. The Arria 10 FPGA is located in the center of the card with MiniPod optical modules driving the I/O to/from the detector. Eight MPO-12 connectors are used for the fibers. Two PCIe Gen3 x8 lanes are used for data output. Two dedicated SFP+ modules are used for the timing interface.



(a)

(b)

Fig. 3. Schematics of the TELL40 firmware. (a) Input data flow from the front-end and is fed to two independent data streams, where it is decoded (1), processed (2), formatted (3), and finally sent to the PCIe output. (b) Processing is subdivided into four parts. The detail of each is described in the text.

A schematic overview of the TELL40 firmware for LHCb is shown in Fig. 3. For reasons of bandwidth optimization of the PCIe bus, the firmware is divided into two discrete and identical PCIe lanes, with input data links split between the two. Ten optical links from the VELO module are sent to each data stream. The split is fixed such that the overall data bandwidth to each stream is the same (this is achieved by exploiting detector geometry and expected number of hits per link). Shown in orange are several low-level interface functions of the firmware, such as the transceiver interfaces for the optical links and PCIe bus, the slow controls memory interface for communicating with the registers in the firmware, and the interface to the TFC system. These components are common to the experiment. The central "Data Processing" component (in gray) is custom for each detector of LHCb, tailored to the front-end data coming from said detector. For VELO, the first component "Decoding and Deserialization" is also custom, in order to handle the custom GWT data protocol. More detail is given in Section II-A. For the other detectors, a generic component is used. The last component (in blue) formats the data into LHCb event fragments, adding a header with an event ID (a monotonically increasing number with accepted events), a source ID (unique per TELL40 data stream), a fragment size in bytes, and format version number. This header information is used to assemble individual data fragments from all TELL40s into complete events in the LHCb event-building software.

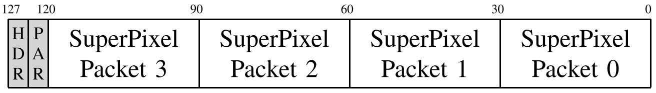


Fig. 4. GWT data frame. The most significant bits arrive at the transceiver first, with the four header bits (HDR) leading, following by four parity bits (PAR), and the four 30-bit SPPs. The SPPs are scrambled for transmission to ensure bit balancing on the wire.

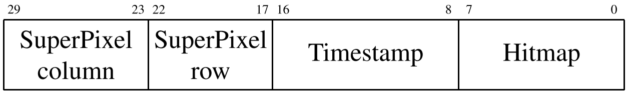


Fig. 5. SuperPixel data format. The most significant bits represent the address of the SuperPixel in the VeloPix matrix, in column, row order. Next is the 9-bit timestamp, followed by the pixel hit-map. A high bit signifies that pixel was "hit."

The introduction outlines the set of requirements of the VELO firmware, namely, 1) handling the VeloPix custom GWT protocol; 2) reordering of the data fragments in time; and 3) processing the SuperPixel data content. Each of these will be described in detail in Section II-A–II-C.

### A. GWT Deserialization and Decoding

Figs. 4 and 5 describe the VeloPix data. The former, shows the GWT data frame of 128 bits, with header, parity information, and four SuperPixel Packets (SPPs). The SPPs come in two variants: data and special. A data SPP is expanded in Fig. 5. A special SPP is denoted by an empty pixel hitmap (i.e., 0 × 00). Although the VeloPix performs zero suppression, the GWT transmitter maintains an active link, constantly sending SPPs. Because four SPPs need to be sent every clock cycle, special idle SPPs are used to fill the gaps in the absence of data SPPs. Other special SPPs are sent in response to control signals received by the VeloPix. The upper bits [29:26] of the special SPP are used to distinguish the different special types.

The GWT data arrive at the TELL40 receivers as a serial stream. It is converted to a series of 32-bit words and a search is performed for the 4-bit 0xA header pattern. This is achieved with a bit-slip operation until the header is in the right location (“HDR” in the figure). Then a lock is asserted signifying that the GWT data frame has been found and aligned. The lock can be lost if the 0xA header pattern has not been seen for a number of clock cycles. Both the lock assertion and deassertion times can be configured. At this point the data are considered “word aligned.”

After word alignment, the four parity bits of the GWT frame can be used to check for the presence of errors in the frame. This is insufficient to support error recovery, however, a significant number of parity errors in the data stream can indicate a degradation of the link quality. Frames with parity errors are marked as invalid, and such frames are dropped. A threshold for an acceptable error rate has yet to be defined, however, a high rate would raise a TELL40 error, thereby halting data taking. Continuous monitoring of the parity error counts throguh registers will be employed to signal early warnings of poor signal integrity. User-issued VeloPix reset signals can be issued to attempt recovery of a failed link. Should the problem persist, the link can be excluded from data taking. Error detection is followed by descrambling of the four SPPs—each 30-bit wide, expanded in Fig. 5. The scrambling algorithm is implemented as a linear-feedback shift register described by the feedback polynomial in (1), where the power of $x$ represents the bits that are tapped in the feedback circuit

$$x^{30} + x^{16} + x^{15} + x + 1. \qquad (1)$$

The output of the VELO “Deserialization and Decoding” block is a fully descrambled GWT frame sent to the data-processing component.

### B. Time Ordering of SPPs

The “data-processing” component represents the dominant workload of the VELO firmware and consumes the most significant fraction of the Arria 10 resources. Current estimates show that data processing consumes approximately 80% of both the logic resources and the block memory resources. However, the firmware is not in its final incarnation, and these numbers are expected to change. The data processing [Fig. 3(b)] can be subdivided into several subcomponents: 1) SuperPixel extraction; 2) timestamp sorting; 3) time alignment; and 4) clustering. The last of these, 4), is dealt with in Section II-B1 on processing the SuperPixel content.

*1) SuperPixel Extraction:* The GWT frames arrive at the input of the data processing subsequent to their deserialization and decoding/descrambling. The four SuperPixels are extracted from the frame and retimed from their 40-MHz input rate (driven by the LHC bunch crossing rate) to 160 MHz. By doing so, the SPPs can be dealt with individually rather than in the groups of four that arrive. Empty SPPs are indicated by an empty pixel hit-map (i.e., all eight pixels in the SPP are zero) and are discarded. Next, the timestamps of the remaining valid data are converted from Gray code (used to minimize digital switching in the VeloPix) to binary.

Ten input links are processed in parallel in the SuperPixel Extraction block. The input link mapping is fixed at compile time. However, in the blocks following (described in Section II-B2), the fixed ordering of input links is lost.

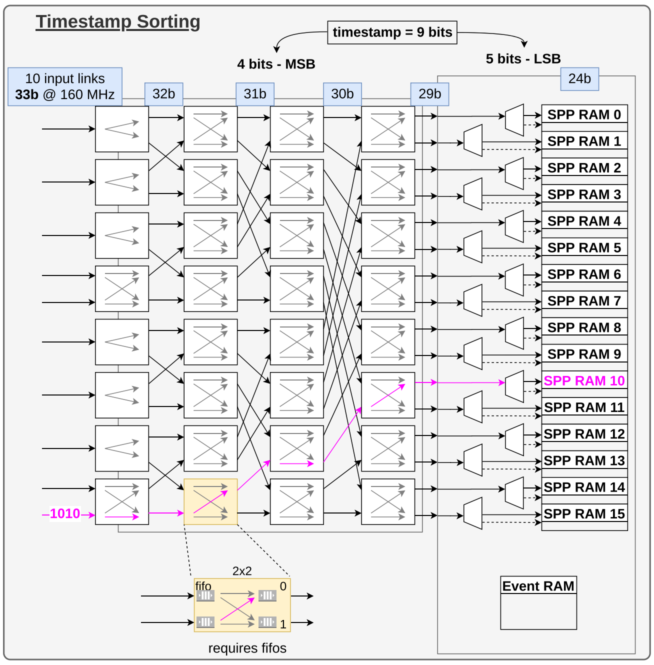


Fig. 6. Dataflow through the Timestamp Sorting component. Detailed explanation is given in the text.

Therefore, in order to keep track of which SPPs come from which VeloPix chips, a chip identifier (3 bits) is prepended to the data, extending the size from 30 to 33 bits. Each data stream processes half of the VELO module, or six chips. To map an SPP to its originating chip, the 3 bits of the chip ID are combined with the data stream number (identified as 0 or 1 in the output data) giving 4 bits, which are sufficient to uniquely identify the 12 chips on the full VELO module.

*2) Timestamp Sorting:* The SPPs have a 9-bit timestamp (known as a bunch-crossing ID in LHC parlance) and therefore can represent 512 time ticks of the LHC 40-MHz clock cycle domain before wrapping around to zero. In fact, the LHC timestamp is 12 bits, and only the lower 9 bits are sent from VeloPix. The completion of the full timestamp is dealt with in the next section. The consequence of this is that the latency of the data coming from the VeloPix to the TELL40 must not exceed 512 clock cycles. Otherwise, for example, it is impossible to identify if a particle hit happened 100 clock cycles ago or 512 + 100 clock cycles ago. Monte-Carlo studies of LHCb simulation data [7] show that the 9-bit timestamp should be sufficient at the expected particle rates at LHCb. However, it is also required that the firmware does not add a significant amount of latency to the incoming data. Simply using multiplexers to sort the timestamped data was found to be too complex for the firmware compilation.

Fig. 6 shows a schematic overview of the timestamp sorting component. A switching router was employed to sort the four most significant bits of the timestamp into sixteen timestamp ranges. Sorted SPPs are stored in RAMs. Multiplexers are employed to use the five least significant bits to define 32 address ranges in RAM at which the SPP is stored. SPP counts are tracked in a separate “event count RAM” for all 512 time bins. These counts are used to define the final address within the RAM address range. The switching router works as follows: SPPs from ten input links are sorted 1 bit per column for the first 4 MSBs. Eight switching blocks are used in each column, with two inputs and outputs at each stage, with FIFOs on both sides to deal with congestion. Because there are only ten input links, six of the switching blocks in the first column require only one input. As the SPPs get sorted, their timestamp bits are removed, because at end of the sorting, the address in RAM corresponds to the timestamp of the original data. An example “1010” (10 in decimal) for the 4 MSBs is given in pink in the figure. Zeroes go to the upper output of each block and ones down. The SPP correctly arrives at RAM 10.

As a bandwidth optimization, the RAMs are split into two, such that reading and writing can be performed simultaneously. Data are written to the RAMs for 512 clock cycles (to accommodate the variable latency of the incoming data); after which the read and write locations are swapped, and previously written data are read out. In total, there are two data streams with 16 RAMs of 2 times 32 time bins of 512 SPPs of 24 bits each, resulting in 25 Mbit of RAM usage, representing almost half of the available RAM in the FPGA.

*3) Aligning to the LHCb Timing System:* Once the data are sorted in time, they need to be aligned to the timing system of LHCb-known as the TFC system [11]. The TFC provides a global clock to the experiment along with the current timestamp, and metadata associated with that timestamp. Such metadata can be a fast reset signal, a synchronization flag, or a data veto. The first stage of time alignment involves extending the 9-bit VELO timestamp to the full 12-bit LHC timestamp. This is done using synchronization commands from the TFC to both the VeloPix and the TELL40. The VeloPix sends special SPPs with the full 12-bit timestamp when it sees these commands. The special packets received in the TELL40 are stored and used to complete the 9-bit timestamps of the subsequent SPPs. Prior to synchronization, data are not considered valid.

The TFC metadata is buffered in the TELL40 and must be matched to the VELO data such that a consistent set of data fragments corresponding to a given timestamp can be combined downstream in the event builders. This matching is performed by reading the next instance of timing metadata from the TFC buffer and reading the corresponding timestamp address from the VELO RAMs. The VELO data and TFC metadata are then propagated forward synchronously through the rest of the firmware.

### C. Processing of SuperPixel Content

Pixel data need to be extracted from the SuperPixels in order to reconstruct the most accurate x,y position of the ionizing particle from which they originate. A single particle can “hit” a cluster of pixels. Combining the individual pixels hits into clusters is required to determine the geometric center of the particle trajectory at the sensor, and perform accurate tracking. One of the goals of the VELO firmware is to perform this clustering on the FPGA and reduce the processing load on the CPU farm.

Clustering starts with classification of the SuperPixels (see Fig. 7). SuperPixels are classified into two types for the purposes of producing clusters-isolated SuperPixels and nonisolated SuperPixels. The first stage of the clustering component is a search for isolated SuperPixels. The incoming SPs are buffered for an event, and then a search is performed for neighbors in any of the eight adjacent locations around the each SP. SPs with no neighbors are tagged as isolated, and, as such, the final cluster will be produced from the SuperPixel itself. Once the SuperPixel has been classified, the outcome is used to decide the next stage of processing-isolated or nonisolated clustering.

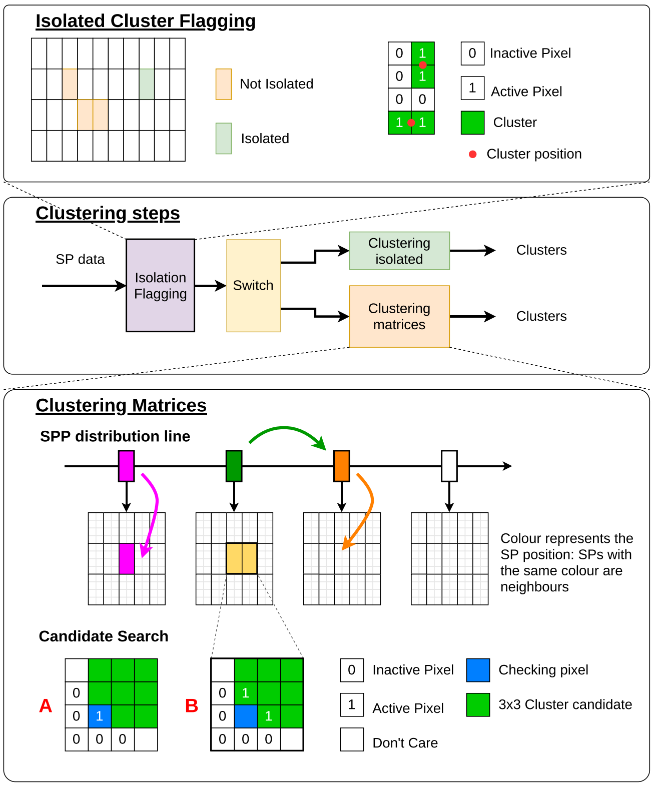


Fig. 7. Diagrams outlining cluster creation from SuperPixels. The top box shows the steps to produce clusters with the processing selection based on whether the SP is isolated or not. The bottom box shows the more complex treatment required for nonisolated SPs. A distribution line groups SP neighbors, and a candidate check is performed on each pixel to match condition A or B.

For isolated clustering, a lookup table is used to determine the final cluster from the 255 possible arrangements of hits in the eight pixel bits, and this form of clustering is complete.

For nonisolated clustering, more complex processing is required. First, a cluster candidate search is performed on the SPs. Forty clustering matrices are employed to perform the task per data stream. Each matrix can hold up to 3 × 5 SPs (or 12 × 10 pixels). Cluster candidates identify 3 × 3 pixels. SuperPixels are delivered to these matrices along a distribution line. Empty matrices have no predefined geometrical position on the VELO module. Their position is initialized when the first SP is added, and it is added to the central matrix position. Subsequent SPs are checked against the initialized matrix to determine whether they can be placed. If not, they continue along the distribution line until a match or an uninitialized matrix is found.

TABLE II

CURRENT VELO TELL40 RESOURCE USAGE

| | Logic | Memory blocks |
|---|---|---|
| Timestamp Sorting & Alignment | 32% | 59% |
| Clustering | 31% | 11% |
| Other (non-VELO) | 20% | 15% |
| Total | 83% | 85% |

Values are not final.

Operating in parallel to the distribution, the search for candidates is performed. In the bottom of Fig. 7, the two conditions (A or B) for candidate flagging are shown. In a matrix of 120 pixels, each pixel is verified (in parallel) to determine whether it is a “checking pixel” (highlighted in blue). It matches condition A if it has empty pixels south and west of it (the “zero L pattern”), and an active pixel within. It matches condition B if it has a “zero L pattern,” and an active pixel both north and east of it.

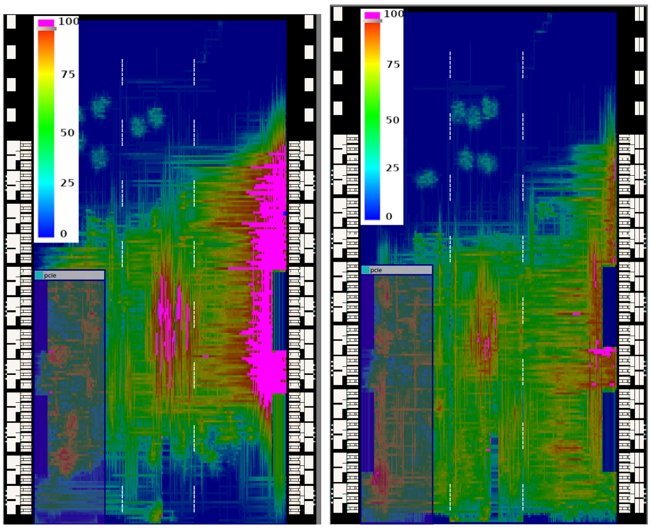
The search patterns A and B have been studied with smaller and larger “zero L patterns” with different active pixel positions. Performances were studied as a function of number of clusters produced, cluster splitting, clusters not found, tracking efficiencies, and FPGA resource utilization. From these studies, the patterns shown in Fig. 7 were chosen.

If a match is found, a 3 × 3 cluster candidate is formed (the green square in the figure). The cluster candidate can be defined as self-contained (all pixels are contained within the 3 × 3 matrix) or not self-contained, indicating that some pixel information is lost. The “checking pixel” used to seed the candidate is marked as done, so double-counting does not occur. When there are no more checking pixels left in the matrix, the process stops and the matrix is reset to its initial “free” state. The last stage produces clusters from the candidates. This is performed using a lookup table (similar to the isolated SPs). The resultant cluster is defined by a pixel position in VeloPix row, column coordinates, and a row and column fraction (in one-eighth steps). The VELO Sensor ID is included from the SPs, and some extra information flags to show whether the cluster is isolated, self-contained, or at the matrix edge. The final cluster definition is still under review at the time of writing.

On the whole, the clustering algorithm leverages the parallelism of the FPGA by performing matrix distribution, candidate search, and cluster formation simultaneously, and sufficient throughput can be achieved. The total resource usage is summarized in Table II. The values should be taken as approximate, as the firmware is still in development. Additional monitoring is needed, and resource optimization has not been performed. A reduced firmware without clustering will be considered if resources are insufficient for full functionality (downstream GPU-based clustering will be performed in that case).

## III. TESTING THE VELO FIRMWARE

The LHCb firmware is built with the Intel Quartus software (v18.1 SE) and simulated using Mentor Graphics QuestaSim (v10.6c). CERN’s gitlab is used as code repository for the firmware and an automatic “simulate, build, test” chain has been developed. This allows every new release of the firmware to be adequately tested and changes to be easily tracked.

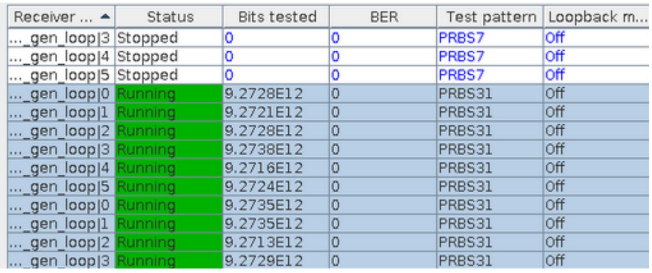


(a) (b)

Fig. 8. Quartus Chip Planner software showing the routing congestion in a compiled firmware. The color shows the level of routing congestion with pink areas over 95% congested. (a) Firmware with a data bypass. The GWT deserialization and decoding component shows the highest degree of congestion. (b) Using smaller data sizes and a faster clock, along with some reduced code complexity, the compiler is able to significantly reduce the congestion.

A simulation checker has been developed using this framework. LHCb Monte-Carlo simulation data have been used as test input for the firmware. The simulation tests both VELO and TFC functionality, requiring correct matching of the VELO data to the TFC metadata. The data output of the firmware simulation is cross-checked against the input using the simulation checker. Missing or corrupted data will cause the checks to fail and indicate the disparity in the source and output data files. Current tests of several thousand events pass the checker indicating good performance of the firmware logic. Unit tests are implemented to ensure full test coverage of the firmware.

The Questasim simulation is a behavioral check of the firmware and does not check the performance after placement and routing of the firmware on-chip. The same type of tests can be performed in the Arria 10 to test the performance after full compilation. The LHCb firmware has a feature to inject files into the memory of the firmware in a similar manner to the simulation, and the output files can be checked against the source too. This has not yet been performed as the compilation shows a degree of routing congestion that must be resolved first. The routing congestion results in failure to achieve timing closure of the firmware. This is typical of a large firmware project and is a task to be completed in the coming months. An example of this congestion in the deserialization and decoding component can be seen in Fig. 8(a) and the reduction after study and optimization in Fig. 8(b). This technique will be applied throughout the firmware code where necessary.



| Receiver ... | Status | Bits tested | BER | Test pattern | Loopback m... |
|---|---|---|---|---|---|
| ..._gen_loop\|3 | Stopped | 0 | 0 | PRBS7 | Off |
| ..._gen_loop\|4 | Stopped | 0 | 0 | PRBS7 | Off |
| ..._gen_loop\|5 | Stopped | 0 | 0 | PRBS7 | Off |
| ..._gen_loop\|0 | Running | 9.2728E12 | 0 | PRBS31 | Off |
| ..._gen_loop\|1 | Running | 9.2721E12 | 0 | PRBS31 | Off |
| ..._gen_loop\|2 | Running | 9.2728E12 | 0 | PRBS31 | Off |
| ..._gen_loop\|3 | Running | 9.2738E12 | 0 | PRBS31 | Off |
| ..._gen_loop\|4 | Running | 9.2716E12 | 0 | PRBS31 | Off |
| ..._gen_loop\|5 | Running | 9.2724E12 | 0 | PRBS31 | Off |
| ..._gen_loop\|0 | Running | 9.2735E12 | 0 | PRBS31 | Off |
| ..._gen_loop\|1 | Running | 9.2735E12 | 0 | PRBS31 | Off |
| ..._gen_loop\|2 | Running | 9.2713E12 | 0 | PRBS31 | Off |
| ..._gen_loop\|3 | Running | 9.2729E12 | 0 | PRBS31 | Off |

Fig. 9. Quartus Transceiver Toolkit showing the PRBS tests of the input links. The number of bits tested and bit error rate (BER) are shown. VELO links are required to have a BER of less than $10^{-12}$.

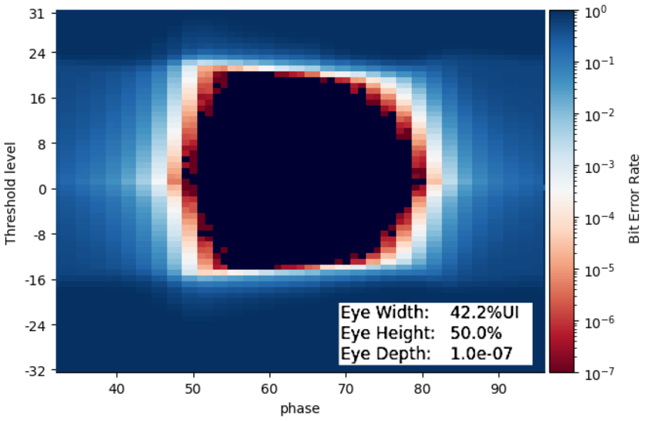


Fig. 10. Example eye diagram produced with VeloPix GWT data. An "open eye" (the region in the center where no transitions can be seen) signifies that sufficient distinction can be made between the "0" and "1" bit levels to recover the data encoded in the incoming stream.

In lieu of a fully working firmware, an interim solution was developed to allow adequate testing of the VeloPix and supporting hardware. A bypass component was created as an alternative to the full data-processing component. This simply buffers the GWT input data and passes it directly to the PCIe output stream. Load balancing is performed among the ten input links to the output. The data content is untouched. The data are time sorted offline. This also provides an essential test of the VELO deserialization and decoding block. This deserialization cannot be adequately tested in simulation because of the analog nature of the optical input stream. A known number and pattern of test signals are injected into the VeloPix, and these are checked in the output data. Several problems can occur to cause this test to fail: poor signal integrity on the optical links; incorrect clocking in the deserialization and decoding component; overflowing buffers in the firmware; etc. Readback and counter registers used in the firmware are exposed to show the loss of signal lock, parity error counts. Link quality is determined using pseudorandom bit stream (PRBS) tests (an example is shown in Fig. 9). The VeloPix has a special PRBS mode for this kind of link test. It does not test the GWT protocol, but it does signify that reliable analog settings have been set on the transmission (VeloPix) and receiving (the TELL40 transceivers) ends of the data link. Eye diagrams can be made for each link to test with the GWT protocol-an example is as shown in Fig. 10. Successful test signal patterns have been injected into VeloPix and reconstructed using the bypassed output data.

TABLE III
COMPARISON OF FPGA RESOURCES FOR THE CURRENT VELO AND A CANDIDATE FOR ITS NEXT UPGRADE

| | Arria 10 | AGI-027 | Factor Increase |
|---|---|---|---|
| Process (nm) | 20 | 10 | 2.0 |
| Logic Elements (k) | 1150 | 2692 | 2.3 |
| M20k Memory (Mbits) | 53 | 259 | 4.9 |
| DSP | 1518 | 17056 | 11.2 |

## IV. Next-Generation Data-Acquisition Planning

The high-luminosity phase of the LHC, due to start in 2028, will provide approximately 7.5 times the luminosity at LHCb [12]. The number of tracks and hits is expected to increase a similar amount. This poses a significant challenge for the tracking software for a detector such as VELO, as the number of combinations of hits increases exponentially. One proposed solution is to use more precise timing to separate the proton-proton interactions that occur. By breaking up the event into smaller time quanta, there are fewer hits to process and the combinatorics become more manageable. Integrated, there is no escaping the fact that more bandwidth is required for more data. In fact, the extra bits required to store a more precise timestamp, and an expected increase in spatial precision (2×–4×), coupled with the luminosity increase, result in a minimum expected bandwidth of ~200 Gb/s to be required from a VeloPix-like ASIC. Ignoring the rather formidable challenge of producing such a front-end ASIC and supporting electronics, one can take this future VELO as a test case to study the requirements for a cutting-edge data-acquisition system for the HL-LHC era of detectors. Given the time frames required to prototype and produce a readout board such as the PCIe40 (est. 6+ years), one can look at the latest FPGA technology available today, and expect not more than one new generation advance in the technology by the time production is necessary.

For the sake of comparison, the Intel Agilex I-Series is at the top end of Intel's current catalog. Table III summarizes a comparison of the current Arria 10 (AX115) used in the PCIe40 and the Agilex I-Series AGI-027. With the process change from 20 to 10 nm, nominally one would expect to run at twice the clock frequency (this is an oversimplification, but not too ambitious given the frequencies currently used in most of the firmware are 160 MHz and below). Combining this with the increase of logic elements (2.3×) gives almost a factor of 5× increase in processing bandwidth, whereas our requirement of 200 Gb/s is 10× the current generation, leaving a 2× shortfall. The same calculation can be done for the amount of M20k memory, with a similar result. The outlier in the change of resources is the increase in digital signal processors (DSPs) (mostly used for mathematical calculation), which increase by a factor of 11.2×. This significant DSP upgrade can easily be explained by the current market trend for FPGAs as tools for machine/deep-learning networks, which rely heavily on DSP use. The downside of this is that DSPs are a significantly underused resource in the current generation of FPGA algorithms used for data processing. Therefore, if the same approach is taken for the next generation of LHCb, the resource wastage becomes significantly worse (and a cheaper no-DSP FPGA does not seem to be an option offered by the main vendors).

The increased processing required in the online data-acquisition system is easily matched in the processing required to reconstruct the physics events. Reconstructing the tracks in the VELO is typically a CPU-intensive task. Any pattern recognition preprocessing that can be performed to produce tracks or partial tracks will reduce the CPU load in the computing farms. LHCb's Real-Time Analysis group is exploring the use of compute accelerators (GPUs and FPGAs) to perform this pre-processing task. A new approach currently under study in LHCb employs machine-learning techniques on FPGAs to perform VELO tracking. The research could capitalize on the DSPs currently underutilized in the data acquisition. Further study is required to determine how to marry the different FPGA use cases for the experiment.

## V. Conclusion

The readout firmware for the LHCb VELO is currently under active development. Reception and data recovery from the custom GWT protocol designed for the VeloPix has been demonstrated in the lab, with an error rate below the requirements of the experiment. Time ordering and alignment of the data from the VeloPix has been demonstrated in firmware simulation. Clusterization has also been tested to work in simulation. Congestion and timing closure mark the current challenges to complete the full compilation for a working on-chip firmware. These challenges will be met utilizing the techniques employed for the GWT deserialization and decoding, in time for data taking in 2022.

The VELO has been taken as an example case of how to prepare for the High Luminosity Phase of the LHC (due to start in 2028). A significant increase in the online processing requirement is foreseen. The addition of more precise timing to the data is under study to both improve the detector performance and lighten the compute load. To the same end, VELO tracking using FPGAs is also under study to make better use of online processing resources. Adequate planning is required at the design stage of data-intensive experiments to best manage computational resources required for the next generation of physics discovery.

## Acknowledgment

Karol Hennessy, Themis Bowcock, Francesco Dettori, Karlis Dreimanis, Vinicius Franco Lima, David Hutchcroft, Kurt Rinnert, and Tara Shears are with the Department of Physics, Liverpool University, Liverpool L69 7ZE, U.K. (e-mail: karol.hennessy@cern.ch).

Antonio Fernández Prieto, Pablo Vázquez Regueiro, Edgar Lemos Cid, Oscar Boente García, Abraham Gallas Torreira, and Beatriz García Plana are with the Instituto Galego de Física de Altas Enerxías (IGFAE), Universidade de Santiago de Compostela, E-15782 Santiago de Compostela, Spain (e-mail: antonio.fernandez.prieto@cern.ch).

Jan Buytaert, Oscar Augusto, Victor Coco, Paula Collins, Tim Evans, Massi Ferro-Luzzi, and Heinrich Schindler are with European Organization for Nuclear Research (CERN), 1211 Geneva, Switzerland.

Martin Van Beuzekom, Kristof de Bruyn, Kazu Akiba, Elena Dall' Occo, Cristina Sanchez Graz, Wouter Hulsbergen, Daniel Hynds, Igor Kostiuk, Marcel Merk, and Aleksandra Snoch are with the National Institute for Subatomic Physics (Nikhef), 1009 DB Amsterdam, The Netherlands.

Lars Eklund, Sneha Naik, Manuel Schiller, and Dana Seman Bobulska are with the Department of Physics and Astronomy, University of Glasgow, Glasgow G12 8QQ, U.K.

Dónal Murray, Silvia Borghi, Stefano de Capua, Deepanwita Dutta, Marco Gersabeck, Chris Parkes, Peter Svihra, and Mark Williams are with the Department of Physics and Astronomy, The University of Manchester, Manchester M13 9PL, U.K.

Alexander Leflat, Galina Bogdanova, and Vladimir Volkov are with the Skobeltsyn Institute of Nuclear Physics, Moscow State University (MSU), 119991 Moscow, Russia.

Giovanni Bassi and Michael J. Morello are with the Scuola Normale Superiore, 56126 Pisa, Italy, and also with the INFN Sezione di Pisa, 56127 Pisa, Italy.

Giovanni Punzi is with the Department of Physics, University of Pisa, 56126 Pisa, Italy, and also with the INFN Sezione di Pisa, 56127 Pisa, Italy.

Federico Lazzari is with the Department of Physical Sciences, Earth and Environment, University of Siena, 53100 Siena, Italy, and also with the INFN Sezione di Pisa, 56127 Pisa, Italy.

Pawel Kopciewicz, Maciej Majewski, Agnieszka Oblakowska-Mucha, Bartlomej Rachwal, and Tomasz Szumlak are with the Department of Particle Interactions and Detection Techniques, AGH University of Science and Technology, PL-30059 Kraków, Poland.

Lucas Meyer Garcia, Franciole Marinho, Larissa Helena Mendes, Irina Nasteva, Juan Otalora, and Gabriel Rodrigues are with the Instituto de Fisica, Universidade Federal do Rio de Janeiro (UFRJ), 21941-972 Rio de Janeiro, Brazil.

Jaap Velthuis is with the School of Physics, University of Bristol, Bristol BS8 1TL, U.K.

Pawel Jalocha, Malcolm John, Nathan Jurik, and Luke Scantlebury-Smead are with the Department of Physics, University of Oxford, Oxford OX1 3PU, U.K.

John Back, Tim Gershon, Tom Latham, and Andrew Morris are with the Department of Physics, University of Warwick, Coventry CV4 7AL, U.K.

## References

[1] The LHCb Collaboration *et al.*, "The LHCb Detector at the LHC," *J. Instrum.*, vol. 3, Aug. 2008, Art. no. S08005.

[2] The LHCb Collaboration, "LHCb VELO upgrade technical design report," CERN, Meyrin, Switzerland, Tech. Rep. CERN-LHCC-2013-021, 2013. [Online]. Available: https://cds.cern.ch/record/1624070

[3] A. F. Prieto *et al.*, "Phase I upgrade of the readout system of the vertex detector at the LHCb experiment," *IEEE Trans. Nucl. Sci.*, vol. 67, no. 4, pp. 732–739, Apr. 2020, doi: 10.1109/TNS.2020.2970534.

[4] T. Poikela *et al.*, "VeloPix: The pixel ASIC for the LHCb upgrade," *J. Instrum.*, vol. 10, no. 1, Jan. 2015, Art. no. C01057.

[5] O. A. de Aguiar Francisco *et al.*, "Evaporative $CO_2$ microchannel cooling for the LHCb VELO pixel upgrade," *J. Instrum.*, vol. 10, no. 5, 2015, Art. no. C01014.

[6] V. Gromov *et al.*, "Development of a low power 5.12 Gbps data serializer and wireline transmitter circuit for the VeloPix chip," *J. Instrum.*, vol. 10, no. 1, 2015, Art. no. C01054.

[7] T. Poikela, "Readout architecture for hybrid pixel readout chips," CERN, Meyrin, Switzerland, Tech. Rep. CERN-THESIS-2015-111, 2015, p. 140. [Online]. Available: https://cds.cern.ch/record/2042198/files/CERN-THESIS-2015-111.pdf

[8] M. Bellato *et al.*, "A PCIe Gen3 based readout for the LHCb upgrade," *J. Phys., Conf. Ser.*, vol. 513, no. 1, Jun. 2014, Art. no. 012023.

[9] F. Vasey *et al. Versatile Link Specifications*. Accessed: Oct. 1, 2020. [Online]. Available: https://edms.cern.ch/project/CERN-0000090391

[10] P. Durante, N. Neufeld, R. Schwemmer, G. Balbi, and U. Marconi, "100 Gbps PCI-express readout for the LHCb upgrade," *J. Instrum.*, vol. 10, no. 4, Apr. 2015, Art. no. C04018.

[11] The LHCb Collaboration, "LHCb upgrade trigger and online technical design report," CERN, Meyrin, Switzerland, Tech. Rep. CERN-LHCC-2014-016, 2016. [Online]. Available: https://cds.cern.ch/record/1701361/files/LHCB-TDR-016.pdf

[12] The LHCb Collaboration, "Physics case for an LHCb upgrade II—Opportunities in flavour physics, and beyond, in the HL-LHC era," CERN, Meyrin, Switzerland, Tech. Rep. CERN-LHCC-2018-027, 2018. [Online]. Available: https://cds.cern.ch/record/2636441