High-throughput packet aggregator for the back-end DAQ of CERN CMS HGCAL detector

Martim Rosado*†‡, Pedro Tomás‡, Nuno Roma‡, and André David*†
*On behalf of the CMS Collaboration
† Experimental Physics Department, CERN, Switzerland
‡INESC-ID, Instituto Superior Técnico, Universidade de Lisboa, Portugal

***Abstract*—The Phase 2 upgrade of CERN LHC accelerator requires the CMS detector to replace its endcap calorimeters with the new HGCAL. Each of the 96 back-end DAQ FPGAs will process LHC collision data from 108 input optical fibre pairs operating at 10 Gb/s and will have to route the aggregated data into 12 output optical fibre pairs operating at 24 Gb/s. This paper proposes an architecture of a multi-FIFO readout system, to be integrated into the highly space-constrained FPGA design of the back-end DAQ system of the HGCAL. This readout solution aggregates and conveys the processed data to the system's outputs and minimises the detector dead time by load-balancing the data distribution between sources and destinations. Experimental results demonstrate that the proposed circuit is able to cope with the final detector's bandwidth and configuration requirements, as well as several test systems needed for the assembly and commissioning of the detector.**

***Index Terms*—Data acquisition systems, Field-Programmable Gate Array, Testing and validation, Configurable digital electronic circuits.**

I. Introduction

The Phase 2 upgrade of the Large Hadron Collider (LHC) at the European Organisation for Nuclear Research (CERN) is expected to be concluded and in operation in 2029 and requires the replacement of the endcap calorimeters of the Compact Muon Solenoid (CMS) [1] detector with the new High-Granularity Calorimeter (HGCAL) [2] detector, as the current cannot cope with the new and harsher radiation environment. The new HGCAL Back-end Data Acquisition (DAQ) system must be able to process 750 thousand events per second [2], triggered by a Level 1 Accept (L1A) signal received from the CMS Level-1 Trigger system.

To cope with the data bandwidth required by the HGCAL back-end DAQ system [3], each of its 96 readout modules (implemented in a Field Programmable Gate Array (FPGA)) processes the input data received through 108 optical fibre pairs operating at 10 Gb/s (see fig. 1). This is accomplished through 54 Capture Blocks (CB) in each readout module, responsible for processing the data received from the LHC collisions and transmitting them out to the central CMS DAQ system [4]. The outgoing path consists of 12 optical fibre pairs operating at 24 Gb/s using the SLink protocol [5].

The demanding data rates involved in this processing stage and the need to adopt a versatile and fully scalable implementation infrastructure have led the HGCAL back-end DAQ to be implemented using FPGA technology. In particular, the CMS collaboration decided to implement the whole HGCAL

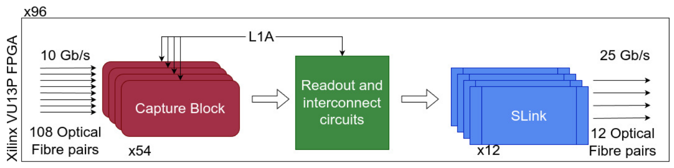

Fig. 1: Simplified dataflow of the HGCAL back-end DAQ system.

back-end electronics using solely AMD-Xilinx VU13P FPGA devices hosted in Serenity [6] Advanced Telecommunications Computing Architecture (ATCA) boards [2].

The main share of resources in each of the FPGAs is allocated to the capture blocks, SLink modules, and related IOs. Consequently, the resulting device occupancy considerably limits the resources available for the interconnect and readout circuits, as depicted in fig. 2. This is aggravated by the need to distribute these elements across FPGA chiplets to connect the data sources to the respective destinations. Since the VU13P FPGA device is composed of 4 Super Logic Regions (SLRs), the capture block data packets are forced to cross these SLRs when being transmitted to the respective SLink modules, greatly limiting the input-output interconnectivity datapath. Moreover, to reduce the strain on the FPGA routing resources, the readout circuits must follow a design that minimises area without sacrificing throughput.

The main contribution of this work is a new design (and the corresponding hardware testing) of a packet aggregation circuit to combine input data from a programmable number of capture blocks into a single SLink output stream. To prevent any data loss, the proposed architecture can buffer in-transit data while handling backpressure events. To fit within the rather limited hardware resources that are still available in

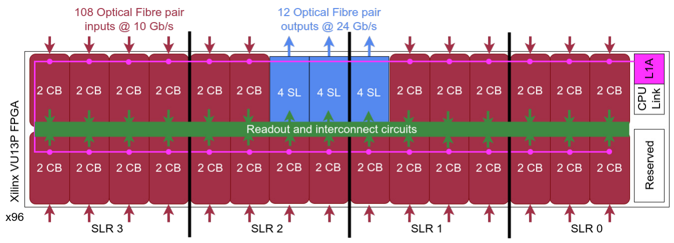
108 Optical Fibre pair inputs @ 10 Gb/s
12 Optical Fibre pair outputs @ 24 Gb/s
2 CB
4 SL
Readout and interconnect circuits
L1A
CPU Link
Reserved
Xilinx VU13P FPGA
x96
SLR 3
SLR 2
SLR 1
SLR 0

Fig. 2: Floorplanning of the HGCAL Back-end DAQ datapath implemented in a VU13P FPGA device. The readout and interconnect circuits-the focus of this paper-connect signals from the 54 Capture Blocks (CB) to the 12 SLinks (SL) across the 4 SLRs of this device.

the used FPGA device, this backpressure handling mechanism must be implemented with LUT-based FIFOs. Furthermore, the data aggregation must ensure load balancing of the data transmitted to the SLink outputs to minimise the system's deadtime.

II. Readout circuit requirements and operation

A. Requisites and Operation

The readout circuit illustrated in fig. 1 extracts collision (event) data from the capture blocks and merges them into a single data stream to be fed to an SLink output module. One data packet is sent whenever an L1A signal is asserted and each packet includes N capture block sub-packets. The capture blocks must be read in order and in full because no data interleaving is possible among capture blocks. This allows the offline software to unambiguously identify the sources with no additional data overhead.

Although the readout circuit is being specifically designed for the final detector system, it is important to recall that this is not the only element of the HGCAL needing this functionality, as several other components will also integrate this same readout circuit. As a consequence, the designed architecture must support a varying number of input capture blocks and two possible data widths (64-bit or 128-bit). Furthermore, to comply with the required output bandwidth of the HGCAL DAQ, this circuit must ensure a 24 Gb/s throughput. In fact, since the readout circuit acts as the aggregation point connecting the capture blocks to the SLink outputs, a smaller throughput would bottleneck the entire system. Consequently, the readout circuit must be clocked at 380 MHz and 190 MHz when handling 64-bit and 128-bit data words, respectively.

Another restriction that should be taken into account is the fact that no BRAM cells can be used by the readout as these are already fully allocated to the capture blocks and IOs. Furthermore, the already existing routing complexity (due to the FPGA occupancy) and the severe penalties incurred when data are transmitted across SLRs (see Figure 2) motivates a design aiming to minimise the area of the readout circuit.

To ensure a perfect commitment of all timing requirements (which is aggravated by the spatial spread of the capture blocks across the SLRs), the connection between the capture blocks and the readout circuits should be performed by means of a delay chain. Such flexible interconnect comprises a bi-directional pipeline to transfer the collision data (from capture blocks to the readout circuit) and the required control signals (from the readout circuit to the capture blocks). Moreover, due to constraints imposed by the used VU13P physical routing, it is not possible to implement an any-to-any design. Hence, each capture block shall be connected to only a subset of the readout circuits. Some overlap between readout circuit datapaths shall be considered in order to balance the data load between SLinks [3].

B. Data loss prevention

In the event of an SLink module not being immediately serviced by the central CMS DAQ system, or if it is fed with a large amount of data, the involved data payload will accumulate and it will eventually assert a backpressure signal connected to the readout circuit. To prevent data loss, data transmission is immediately stopped and upstream buffers will start being filled. In this circumstance, the stages of the delay line between the readout circuit and the capture blocks must be carefully controlled, as it is not possible to directly act on them as they are distributed across the FPGA.

To stop data transmission, two possible solutions arise: either the readout is stopped at the end of the packet, buffering the remaining words, or the readout is stopped mid-packet, buffering the words already in the datapath. Buffering a complete packet requires a considerable amount of resources, as the maximum packet length is in the order of $N \times 2^{12}$ 64-bit words, with N being the number of capture blocks interfaced by a readout circuit and the worst-case scenario would need to be accounted for. As the worst-case scenario is the buffering of a complete readout packet, this solution is not feasible in terms of resources. For this reason, partial buffering must be implemented.

C. Load Balancing

The 108 optical fibres entering the capture blocks have different data rates that depend on the region of the detector that they service. As the HGCAL back-end DAQ system is designed to avoid any data loss, it can throttle the CMS detector trigger if the data rate becomes overwhelming at the cost of an undesired detector deadtime. To avoid such deadtime, a load balancing strategy needs to be implemented, both inter-FPGA and intra-FPGA.

The inter-FPGA load balancing strategy consists of grouping the input fibre pairs to minimise the spread of the total input data throughput across different FPGAs. As a single design replicated in all devices is desirable for maintenance reasons, the intra-FPGA load balancing is implemented using a fixed interconnect and configurable readout circuits. This allows to balance the data transmission across the 12 SLink outputs with a single design while avoiding the complexity of an any-to-any routing and still complying with the routing and resource constraints of the device.

III. Readout Circuit Design

The block diagram of the designed readout circuit is presented in Figure 3, inserted in the broader context of the CMS detector. It can be divided into control and datapath, whose main components are the two high-throughput input multiplexers, a controller, a backpressure FIFO, and a delay module. Software-accessible registers allow to configure which capture blocks are to be read by the circuit, by using the IPBus protocol [9] to act on the capture block pointer. For optimisation reasons, it is up to the software to ensure that no two readout circuits are programmed to read the same capture block. This configuration allows the readout circuits to be instantiated in the final detector system and balances the data load among the 12 existing SLink outputs.

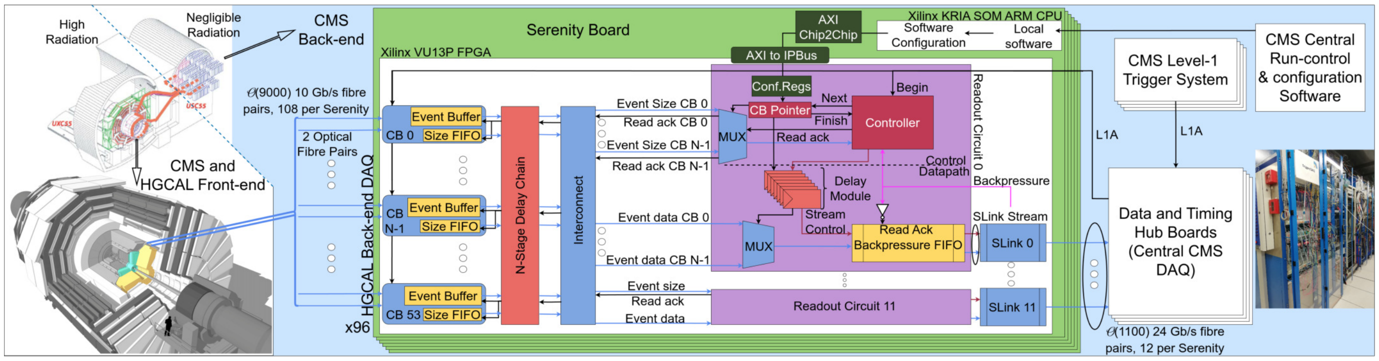

Fig. 3: Readout circuit block diagram (in purple) inserted in the context of the CMS detector. The HGCAL datapath is represented in blue with the FIFOs of the system highlighted in yellow and the delay elements in orange. The readout controller, CB pointer, and respective control signals are represented in red. The backpressure signal is highlighted in pink. The CPU connection modules are represented in dark green. CMS detector diagrams from [7], [8].

Each readout circuit is connected to each capture block it interfaces using 2 FIFOs that interface the bidirectional delay chain. The main one, called Event Buffer, receives the 64-bit data words that are to be transmitted onwards. The smaller FIFO, called Size FIFO, contains a 12-bit word per packet in the Event Buffer FIFO, indicating the packet length.

The backpressure FIFO buffers all in-transit data, ensuring no data loss when the backpressure signal is asserted, using a similar technique as the one used in [10] to absorb pipeline stalls. This significantly simplifies the controller by allowing it to handle backpressure in a fully transparent way (backpressure becomes an enable signal from the controller's perspective) provided that the FIFO can buffer all the words in the datapath between the capture blocks and the readout circuit. By connecting the read acknowledge port of the backpressure FIFO to the negated backpressure signal transmitted by the SLink, the FIFO behaves like a register, as it is being drained every clock cycle when the backpressure is absent. To guarantee the buffering of all in-transit data, the depth of the FIFO must be twice the pipeline depth, in order to account for the already extracted data words coming from the capture blocks, and for the read acknowledge signals travelling to the capture blocks that will, in their turn, originate more data.

The controller module is responsible for the generation of the capture block pointer which controls the multiplexers that select which capture block should be read. It also issues the signals that drain the capture block FIFOs and the control signals of the SLink stream. Due to the latency of the several stages of the datapath, the control signals that are controlling the data flow from the capture blocks need to be delayed to ensure that they are active when the capture block data arrives. This delay function is also ensured by a convenient delay chain, which delays these signals by twice the pipeline depth, i.e. the time taken between the issuing of a read acknowledgement from the controller and the arrival of the corresponding capture block data at the readout circuit.

The introduced delay of the control signals, as well as the transparent handling of backpressure, renders the controller fully agnostic with respect to the datapath that separates the readout and the capture blocks. This means that the datapath depth can be changed (according to the system's needs) requiring no modification to the readout controller circuit. This is highly desirable since this same circuit will be integrated into several other systems.

A. *Shift Register LUT cells*

The depth of both the backpressure FIFO and the delay module is tied to the datapath latency. Given the small magnitude of this latency, the implementation of these structures using solely the Shift Register LUTs [11] (as the ones used in [12]) provides a significant reduction of hardware resources.

In fact, this primitive allows to use each single LUT cell either as a 1-bit shift register with a depth of 32 words, or two 1-bit shift registers with a depth of 16 words. As the number of stages in the datapath is expected to be smaller than 8, it follows that only half of a LUT is required per bit in the data width of the words to be buffered. If N is the number of datapath stages and W is the width of the data bus, the spatial complexity of these structures will be $\mathcal{O}(W/2)$ LUTs, instead of $\mathcal{O}(2NW)$ flip-flops.

B. *Controller module*

The software configuration specifies which capture blocks the controller must read among the readout circuit inputs. Upon reception of an L1A signal, the controller registers the size of the first capture block packet to be read (as the packet size is variable). Afterwards, it starts reading the event data by issuing read acknowledge signals to the event buffer FIFO of the selected capture block. Upon finishing a complete packet from a capture block, the same process follows in a round-robin fashion for every capture block to be read. When the last capture block specified by the software configuration is read, the controller stops and waits for the next L1A trigger. The process followed by the controller is described in fig. 4. To prevent any loss in system synchronisation, capture blocks are only read once new event data is available. Otherwise, the controller waits for the availability of data. In addition, the assertion of the backpressure signal received from the SLink can stop the readout, resuming its normal operation after being cleared.

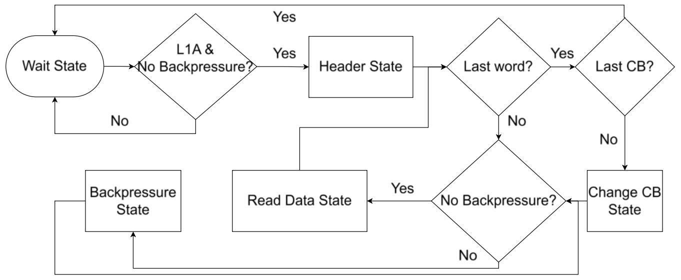

Fig. 4: Controller flowchart.

IV. Experimental Results

To validate and evaluate the proposed aggregation system, the designed readout circuit was implemented in a Xilinx VU7P FPGA device (xcvu7p-flvb2104-1-e, which uses the same technology as the target VU13P device) and integrated into a test system based on a Serenity Z board prototype, as the final Serenity prototypes are still under development.

The first testing procedure measures the circuit's maximum throughput and tests its correct operation under backpressure conditions. When integrated into the final HGCAL detector, it is envisaged that each readout circuit is connected to a maximum of 10 capture blocks. However, the current testing infrastructure is not able to populate such an amount of capture blocks using the existing prototype of the front-end detector hardware that will generate the input data in the final system. However, since the readout circuit is agnostic to the data contents, capture block emulators were used instead. These emulators include software-programmable registers that allow to change the generated packet size and the L1A trigger signal frequency. Hence, the considered test system comprises 10 such emulators connected to one readout circuit that feeds one SLink module. A 64-bit data bus and a 380 MHz clock were used for the datapath connecting the emulators to the SLinks through the readout circuit.

The measured readout circuit throughput for several L1A triggering frequencies of up to 1 MHz is depicted in fig. 5. The used packet size was fixed to 500 64-bit words per L1A trigger. As can be observed, the required throughput of 24 Gb/s was successfully achieved. The circuit operated without transmission errors and was able to saturate the test system readout, proving not only its ability to achieve the required throughput at the detector's nominal L1A frequency but also to correctly operate in the presence of backpressure.

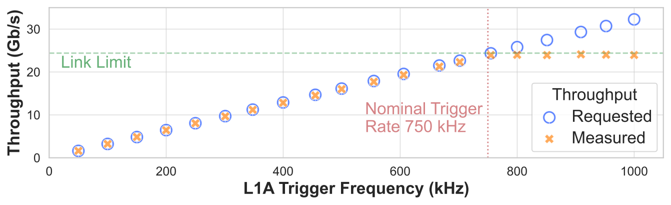

Fig. 5: Readout circuit throughput measurement. The packet size was fixed to 500 64-bit words.

Table I presents the required hardware resources for all readout and interconnect circuits in the VU13P device of the final detector system, with and without the backpressure FIFO. The amount of resources that are estimated for this circuit bode well for the circuit's integration in the overall HGCAL back-end DAQ system. Since this circuit extends to many distant FPGA endpoints, its small footprint enhances the probability of successful routing and timing closure. Furthermore, it is possible to assess the overhead the backpressure FIFO introduces in the system as negligible (less than 0.05%).

TABLE I: Resource estimation of the readout and interconnect circuits in the HGCAL back-end DAQ system when implemented in a VU13P device with and without the backpressure FIFO.

Backpressure FIFO	**LUTs**	**(%)**	**Flip-Flops**	**(%)**
Yes	5234	0.30%	28094	0.81%
No	4383	0.25%	28500	0.82%

V. Conclusion and Future Work

Each Serenity board in the HGCAL back-end DAQ system will process data from 108 optical fibre pairs, each working at a 10 Gb/s data rate. As each board will replicate the same design loaded into its VU13P FPGA device, the interconnect and designed readout circuit perform the intra-FPGA load balancing of the data among the SLink outputs of each board. This article detailed the design and hardware testing of this readout circuit, which can interface several FIFOs and convert their data into an SLink stream while buffering in-transit data in the event of SLink output backpressure, preventing data loss. The use of a small backpressure FIFO and a scalable delay module for control signals simplifies the design of the controller by rendering it agnostic to the depth of the datapath. Furthermore, software-accessible registers specify which FIFOs are read, allowing several instances of the same circuit to balance the data load at the FPGA level.

The obtained results confirm the correct operation of the designed circuit, when operated at the target detector nominal operating frequency of 750k events per second, satisfying the required throughput of 24 Gb/s. Moreover, its significantly reduced resource usage is in line with the requisites to allow a distributed implementation across the FPGA's SLRs.

Future (and ongoing) work comprehends the evaluation of the conceived architecture and of the load balancing mechanisms with real detector data and the scaling of the HGCAL back-end DAQ system towards its perspective dimension, to be achieved by the HGCAL until 2029.

Acknowledgment

Work supported by national funds through Fundação para a Ciência e a Tecnologia (FCT) under project 2022.06780.PTDC (DOI: 10.54499/2022.06780.PTDC) and grant PT/BD/154723/2022, as part of a doctoral studentship at CERN. We also acknowledge the contributions from project UIDB/50021/2020 (DOI: 10.54499/UIDB/50021/2020).

References

[1] CMS Collaboration, "The CMS experiment at the CERN LHC," *Journal of Instrumentation 3*, (2008) S08004, DOI:10.1088/1748-0221/3/08/S08004.

[2] ——, "The Phase-2 Upgrade of the CMS Endcap Calorimeter," CERN, Geneva, Tech. Rep., Nov 2017. [Online]. Available: https://cds.cern.ch/record/2293646

[3] S. Mallios, P. Dauncey, A. David, and P. Vichoudis, "Firmware architecture of the back end DAQ system for the CMS high granularity endcap calorimeter detector," *Journal of Instrumentation*, vol. 17, no. 04, p. C04007, apr 2022. [Online]. Available: https://doi.org/10.1088/1748-0221/17/04/c04007

[4] CMS Collaboration, "The Phase-2 Upgrade of the CMS Data Acquisition and High Level Trigger," CERN, Geneva, Tech. Rep., Mar 2021. [Online]. Available: https://cds.cern.ch/record/2759072

[5] H. van der Bij, R. McLaren, O. Boyle, and G. Rubin, "S-link, a data link interface specification for the LHC era," *IEEE Transactions on Nuclear Science*, vol. 44, no. 3, pp. 398–402, 1997.

[6] CMS Collaboration, "Serenity — an ATCA prototyping platform for CMS Phase-2." [Online]. Available: https://cds.cern.ch/record/2646388

[7] P. Paulitsch, "The silicon sensors for the high granularity calorimeter of cms," *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, vol. 978, p. 164428, 07 2020.

[8] K. Gill, S. Dris, R. Grabit, I. Pedraza, D. Ricci, J. Troska, and F. Vasey, "Integration of cms tracker optical links."

[9] C. Ghabrous Larrea, K. Harder, D. Newbold, D. Sankey, A. Rose, A. Thea, and T. Williams, "IPbus: a flexible Ethernet-based control system for xTCA hardware," *Journal of Instrumentation*, vol. 10, no. 02, p. C02019, 2015.

[10] M. N. Emas, A. Baylis, and G. Stitt, "High-frequency absorption-FIFO pipelining for stratix 10 hyperflex," in *2018 IEEE 26th Annual International Symposium on Field-Programmable Custom Computing Machines (FCCM)*, 2018, pp. 97–100.

[11] AMD Xilinx, "SRL16E primitive documentation." [Online]. Available: https://docs.amd.com/r/en-US/ug974-vivado-ultrascale-libraries/SRL16E

[12] T. Garg, S. Wasly, R. Pellizzoni, and N. Kapre, "Hoplitebuf: FPGA NoCs with provably stall-free FIFOs," in *Proceedings of the 2019 ACM/SIGDA International Symposium on Field-Programmable Gate Arrays*, ser. FPGA '19. New York, NY, USA: Association for Computing Machinery, 2019, p. 222–231. [Online]. Available: https://doi.org/10.1145/3289602.3293917