



Scalable FPGA-based controller of a thousand-ASIC system in the CERN CMS HGCAL detector

Pedro Martim Duarte Rosado

Thesis to obtain the Master of Science Degree in

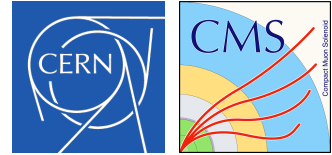
Electrical and Computer Engineering

Supervisors: Prof. André David Tinoco Mendes
Prof. Nuno Filipe Valentim Roma

Examination Committee

Chairperson: Prof. António Manuel Raminhos Cordeiro Grilo
Supervisor: Prof. André David Tinoco Mendes
Member of the Committee: Prof. Horácio Cláudio De Campos Neto

November 2022



I declare that this document is an original work of my own authorship and that it fulfils all the requirements of the Code of Conduct and Good Practices of the Universidade de Lisboa.

Acknowledgments

This work and involved experiences comprise one of the most difficult challenges I ever had the pleasure to face. Although I quite enjoyed the process and feel quite fulfilled, like every challenge, it was demanding and had its ups and downs. In fact, it is my belief the support and encouragement I received throughout all the process was a necessary condition that lead to the successful and satisfactory conclusion of this thesis and degree. Hence, I can only hope to adequately acknowledge this invaluable help within the following lines.

First and foremost, I would like to express my deepest appreciation to my dissertation supervisors Prof. André David and Prof. Nuno Roma not only for their insight, support, sharing of knowledge, and patience that has made this thesis possible but also for providing me with the opportunity to work with HGCAL.

Further, I would like to thank Prof. Pedro Tomás and Dr. Stavros Mallios for their support and involvement in this work, whose insight proved extremely helpful.

Additionally, this work would not have been possible without the opportunity to undertake a technical studentship at CERN, for which I would like to extend my sincere thanks.

I am also thankful for being able to work with a team of colleagues whose presence and help also provided guidance and made our pleasant work environment a reality.

Special thanks to my friends that were always there for me through thick and thin despite the distance: Catarina, Carlos, Rita, Leonor, António, Rodrigo, etc. My deepest gratitude.

Lastly, I would like to mention my family for their caring and support over all these years.

To each and every one of you – Thank you,

Martim Rosado

Abstract

The upcoming of the European Organization for Nuclear Research (CERN) High-Luminosity Large Hadron Collider (LHC) (HL-LHC) phase motivates the replacement of the endcap calorimeters of the Compact Muon Solenoid (CMS) detector with the High-Granularity Calorimeter (HGCAL). To read out its 6 million channels, the HGCAL uses a complex electronic readout chain that comprises a front-end and a back-end. The front-end is located in the experimental cavern comprising about 150 000 radiation-tolerant Application Specific Integrated Circuits (ASICs). The back-end is shielded from radiation and consists of about 100 Field Programmable Gate Arrays (FPGAs). Each FPGA is connected to 108 optical links, each providing a 10.24 Gbit/s transmission rate, and is responsible for configuring up to 3500 ASICs.

This dissertation reports on the work contributed to the control system implemented in the back-end of HGCAL to configure the front-end electronics, known as asynchronous (slow) control. By using development boards to emulate the HGCAL hardware still under development, it was possible to prototype the slow control FPGA hardware and validate the communication with the target front-end ASICs while ensuring complete portability between the prototype and the final detector systems. Furthermore, the configuration time of all the HGCAL electronics was roughly estimated at around 1 minute.

Another contribution of this work to HGCAL is the design of an accumulator system to accelerate the computation of the mean and standard deviation of several metrics in the testing of the HGCAL Readout Chip (ROC) (HGCROC) ASIC, allowing up to four times faster analysis in testing.

Keywords

Fast Prototyping, System Emulation, Testing and Validation, Field-Programmable Gate-Array

Resumo

A actual fase de melhoramento do *High-Luminosity Large Hadron Collider (LHC) (HL-LHC)* do *European Organization for Nuclear Research (CERN)* contempla a substituição de cada um dos calorímetros de *endcap* do detetor *Compact Muon Solenoid (CMS)* pelo *High-Granularity Calorimeter (HGCAL)*. De modo a permitir a leitura dos seus 6 milhões de canais, o *HGCAL* usa uma complexa cadeia electrónica que se divide no *front-end* e *back-end*. O *front-end* está localizado na caverna experimental e é composto por 150 000 *Application Specific Integrated Circuits (ASICs)*, enquanto o *back-end* consiste em aproximadamente 100 *Field Programmable Gate Arrays (FPGAs)*. Cada *FPGA* está ligada a 108 fibras ópticas, cada uma a funcionar a um ritmo de transmissão de 10.24 Gbit/s, e é responsável pela configuração de até 3500 *ASICs*.

Esta dissertação reporta o contributo que foi realizado no seio do sistema de controlo implementado no *back-end* do *HGCAL* para configurar a electrónica do *front-end*, conhecido como controlo assíncrono (*slow control*). Utilizando placas de desenvolvimento que permitiram emular o *hardware* que ainda está em fase de desenvolvimento, foi possível prototipar o *slow control* e validar os seus protocolos de comunicação, garantindo a completa portabilidade entre os sistemas de prototipagem e do detetor final. O tempo de configuração de toda a electrónica do *HGCAL* foi estimado em 1 minuto.

Outra contribuição deste trabalho para o *HGCAL* é o desenvolvimento de uma estrutura de acumulação para acelerar o cálculo da média e desvio padrão de diversas métricas utilizadas na testagem do *ASIC HGCAL Readout Chip (ROC) (HGCROC)* permitindo uma análise quatro vezes mais rápida.

Palavras Chave

Prototipagem Rápida, Emulação de sistemas, Testagem e validação, *Field-Programmable Gate-Array*

Contents

1	Introduction	1
1.1	Particle Accelerators and Detectors	3
1.2	The LHC accelerator and the CMS detector	4
1.3	The CMS High-Granularity Calorimeter	6
1.4	Objectives	8
1.5	Requirements	8
1.6	Contributions	9
1.7	Organisation of this document	10
2	HGCAL Electronic Systems	11
2.1	HGCAL Front-end (on detector) electronics	14
2.1.1	Silicon Section electronics tree	16
2.1.2	Scintillator Section electronics tree	18
2.2	HGCAL Back-end (off detector)	18
2.2.1	Slow Control block	19
2.2.2	The IpGBT Protocol	20
2.3	Contributions to HGCAL electronics prototyping	21
2.3.1	Accelerating HGCROC Testing	22
2.3.2	Slow control architecture validation and testing	23
2.4	Summary	23
3	Multiplexed Accumulator for RocS	25
3.1	HGCROC Calibration	27
3.2	Architectural Considerations	28
3.3	Initial Architecture	30
3.4	Pre-synthesis Configuration	33
3.5	Optimised Architecture	34
3.5.1	Read multiplexer optimisation	34
3.5.2	Technology Adaptation	35

3.6	Resource Usage and Performance Results	38
3.7	Summary	40
4	Slow-Control	41
4.1	Slow-Control Architecture	43
4.1.1	Software Interface	45
4.1.2	Features added to the slow-control block	47
4.1.3	Further improvements and optimisations	49
4.2	Slow-Control Testing and Validation	49
4.2.1	Prototyping System	50
4.2.2	Validation Testing Criteria	52
4.2.3	Validation Results	52
4.3	Performance Evaluation	53
4.3.1	Transaction Throughput	53
4.3.2	HGCAL Configuration Time	53
4.4	Hardware Resource Usage	56
4.4.1	Resource Usage of the Prototyping System	57
4.4.2	Resource Usage and Optimisation of the Slow-Control	57
4.5	Slow-Control Portability Evaluation	58
4.6	Summary	59
5	Conclusion	61
5.1	Conclusions	63
5.2	Future Work	64
	Bibliography	65
A	Publications	69

List of Figures

1.1	The Large Hadron Collider (LHC) and its four collision points. Other four access points are used for the accelerator proper, namely the accelerating cavities, beam cleaning (in space and energy), and the beam dump. The Compact Muon Solenoid (CMS) detector is hosted at the LHC Point 5, on the far left of the image [1].	4
1.2	Global view of the CMS detector and its main sub-detector systems. These systems are hosted in five large wheels in the barrel region and three endcap disks at either end of the barrel [2].	5
1.3	Summary of the main physical High-Granularity Calorimeter (HGCAL) parameters and transverse cut view of the top half of the detector layers. The Electromagnetic Endcap Calorimeter (CE) (CE-E) and Hadronic CE (CE-H) sections can be seen to use only radiation-hard silicon sensors (CE-E) or a mix of both silicon and scintillating tiles (CE-H). Scintillating tiles are used in the volumes with lower radiation fields. From [3].	7
2.1	High level architecture of the HGCAL electronics. The most important division is between the on-detector Front-end (FE) electronics in a high radiation area and the off-detector Back-end (BE) electronics in a zone with a negligible radiation field. The two are connected via optical fibre runs of about 100 m. From [4].	14
2.2	Data Acquisition (DAQ) data and slow command flow on the HGCAL FE electronics for the two detector technologies. The use of the GigaBit Transceiver (GBT) Slow Control Adapter (GBT-SCA) in the scintillator section reflects the presence of additional Application Specific Integrated Circuits (ASICs) in this part of the detector related to the bias voltage of the silicon photomultiplier technology used to convert the scintillating light into electrical signals (not depicted).	15
2.3	Slow control command architectures for the HGCAL silicon FE. The differences between the Low Density (LD) and High Density (HD) are not qualitative, but quantitative, as the same components are arranged differently in the two trees.	17

2.4	Slow control command architectures for the HGCAL scintillator FE. The main difference to the silicon FE is the presence of the GBT-SCA due to the presence of many peripherals in a tileboard and the larger distance between Low-Power GBT (lpGBT) and HGCAL Readout Chip (ROC) (HGCROC).	18
2.5	The lpGBT downlink data frame format from BE to FE. One can see the Internal Control (IC), External Control (EC), and Scalable Low Voltage Signalling (SLVS) electrical link (e-link) fields that in HGCAL are used to communicate with the FE ASICs. From [5]. . . .	21
3.1	HGCROC DAQ data packet structure for one HGCROC half. In total, there are 41 words of 32 bit transmitted when a Level-1 Accept (L1A) trigger fast command is received. The data corresponds to all the relevant measurements in all channels. From [6].	29
3.2	HGCROCv3 word format for regular channels, namely <code>ch0</code> through <code>ch35</code> and <code>Calib</code> . From [6].	29
3.3	HGCROCv3 word format for the <code>Common Mode (CM)</code> channels. From [6].	29
3.4	Simplified Multiplexed Accumulator for ROCs (MARS) architecture [7]. Control signals are highlighted in blue and data signals are in red.	31
3.5	Simplified flowchart of the accumulation process done by the MARS Finite State Machine (FSM). 1024 events are defined as the default maximum number of events. However, this value is controllable at synthesis time through a pre-synthesis parameter.	32
3.6	Vivado Graphical User Interface (GUI) showing the MARS Intellectual Property (IP) instantiation in block design mode. The number of links and maximum number of events to process can be selected, making the IP block easier to adapt to the different systems it is used in. Own work [7].	33
3.7	Initial readout scheme used in MARS. Two big multiplexers are used: one to select the registers to accumulate and another for the readout.	34
3.8	Optimised readout scheme used in the MARS. Only one large multiplexer is used for both the computation and readout phases. A smaller multiplexer selects the control signals to apply to the big multiplexer.	35
3.9	SRLC32E equivalent circuit. The shift register has a control signal, $A[4 : 0]$, that controls the depth of the shift register by multiplexing to the output, Q, the data of the Flip-Flop (FF) corresponding to the value of A thus changing the depth of the shift register. The value of A can be changed at run time. This circuit is implemented using one single Look Up Table (LUT). From [8].	36
3.10	MARS architecture based on the Shift Register LUT (SRL) primitives. Data signals are represented in red whereas control signals are in blue.	37

4.1	Slow-Control architecture showing the main architectural aspect that regards the fact that the IpGBT and GBT-SCA ASICs used in the FE have different protocols. One can also see the software-facing Advanced eXtensible Interface (AXI) interface. The multiplicity of cores reflects the expected number in a BE DAQ Field Programmable Gate Array (FPGA) of a BE DAQ Advanced Telecommunications Computing Architecture (ATCA) board. Own work [4].	44
4.2	Block diagram of the developed prototyping system to test and validate the slow-control block. The corresponding physical setup is depicted in Figure 4.3. Own work [4].	50
4.3	Photograph of the hardware used for the prototyping system developed to test and validate the slow-control block. The ZCU102 board acts as the HGCAL DAQ BE and is connected via optical fibres to the Versatile Link Plus Demo Board (VLDB+). The VLDB+ hosts a IpGBT ASIC and is further connected to a Versatile Link Demo Board (VLDB) board hosting a GBT-SCA ASIC. This set of boards allows to test all the slow-control interfaces of the HGCAL FE.	51

List of Tables

3.1	MARS synthesis resource comparison as reported by Vivado 2019.2 for a 1024-event accumulation of 1 HGCROC DAQ link.	36
3.2	MARS synthesis resource usage as reported by Vivado 2019.2 for a 1024-event 2-link test system. Comparison of the new architecture between versions of the SRL design with and without “Time to Digital Converter (TDC) efficiency” mode included in the IP. . .	38
3.3	MARS synthesis resource usage as reported by Vivado 2019.2 for a 1024-event test system as a function of the number of parallel inputs links it can process.	38
4.1	lpGBT Transmit (TX) frame format of the transactor developed in this work.	45
4.2	lpGBT Receive (RX) frame format of the transactor developed in this work.	46
4.3	GBT-SCA TX frame format of the transactor developed in this work.	47
4.4	Encoding of the Command ID field of the GBT-SCA TX frame.	47
4.5	GBT-SCA RX frame format of the transactor developed in this work.	47
4.6	lpGBT transactor error flags.	48
4.7	GBT-SCA transactor error flags.	48
4.8	Bit fields of the slow-control status registers.	49
4.9	Hardware resources required by the slow-control prototyping system when implemented in the ZCU102 development board using Vivado 2019.2. From [4].	57
4.10	Hardware resources required by the developed slow-control when implemented in a VU13P FPGA using Vivado 2021.2 toolchain. 16 lpGBT and 16 GBT-SCA cores were instantiated.	58

Acronyms

ADC	Analogue to Digital Converter
ASIC	Application Specific Integrated Circuit
ATCA	Advanced Telecommunications Computing Architecture
AXI	Advanced eXtensible Interface
BE	Back-end
BRAM	Block Random Access Memory (RAM)
BX	Bunch Crossing
CDC	Clock Domain Crossing
CE	Endcap Calorimeter
CE-E	Electromagnetic CE
CE-H	Hadronic CE
CERN	European Organization for Nuclear Research
CM	Common Mode
CMS	Compact Muon Solenoid
CPU	Central Processing Unit
DAC	Digital to Analogue Converter
DAQ	Data Acquisition
DMA	Direct Memory Access
DSP	Digital Signal Processor
EC	External Control
ECON	e-link Concentrator
ECON-D	DAQ e-link Concentrator (ECON)

ECON-T	Trigger ECON
e-link	SLVS electrical link
FE	Front-end
FEC	Forward Error Correction
FF	Flip-Flop
FIFO	First In First Out
FPGA	Field Programmable Gate Array
FSM	Finite State Machine
GBT	GigaBit Transceiver
GBT-SCA	GBT Slow Control Adapter
GPIO	General-Purpose Input/Output
GUI	Graphical User Interface
HD	High Density
HDLC	High-level Data Link Control
HGCAL	High-Granularity Calorimeter
HGCROC	HGCAL ROC
HL-LHC	High-Luminosity LHC
IC	Internal Control
IP	Intellectual Property
I2C	Inter-Integrated Circuit
LD	Low Density
LHC	Large Hadron Collider
IpGBT	Low-Power GBT
LUT	Look Up Table
L1A	Level-1 Accept
MARS	Multiplexed Accumulator for ROCs
MGT	Multi-Gigabit Transceiver
MMCM	Mixed-Mode Clock Manager
MPSoC	Multiprocessor System On a Chip (SoC)
PCB	Printed Circuit Board

PL	Programmable Logic
PS	Processing System
RAM	Random Access Memory
ROC	Readout Chip
RX	Receive
SFP+	Enhanced Small Form-factor Pluggable
SLVS	Scalable Low Voltage Signalling
SoC	System On a Chip
SRL	Shift Register LUT
TDC	Time to Digital Converter
TOA	Time of Arrival
TOT	Time over Threshold
TX	Transmit
VHDL	Very High Speed Integrated Circuit Hardware Description Language
VLDB	Versatile Link Demo Board
VLDB+	Versatile Link Plus Demo Board
VTRX+	Versatile Link Plus Transceiver

1

Introduction

Contents

1.1 Particle Accelerators and Detectors	3
1.2 The LHC accelerator and the CMS detector	4
1.3 The CMS High-Granularity Calorimeter	6
1.4 Objectives	8
1.5 Requirements	8
1.6 Contributions	9
1.7 Organisation of this document	10

Curiosity triggers scientific research. The need to understand the world motivates the study of the environment around us. The various questions arising from this need have evolved into different scientific fields. High-energy physics still remains closely connected with fundamental scientific research and core questions about what surrounds each and every one of us: “What is matter made of?” or “How did the universe come to be?”. Particle accelerators are devices that play a very important role in the field, allowing the study of particles not otherwise observable. Despite being developed with fundamental research as a goal, the applications of particle accelerators also contribute to other fields and applications such as oncological therapy, ion implantation for semiconductor fabrication, and isotope production for medical diagnosis.

1.1 Particle Accelerators and Detectors

Particle accelerators increase the energy and speed of a group of charged particles constrained in a beam through the use of electromagnetic fields. Besides accelerating the particles in a beam, accelerators also have to focus the beam to achieve the highest particle density, also known as beam brightness. Particle accelerators can be linear or circular depending on their structure. In a circular accelerator, the particle beam needs to be bent to comply with the circular trajectory. The accelerators used in high-energy physics collide these beams with a certain target of interest or another beam of particles. If enough energy is involved in a collision, new exotic particles may be the obtained result. After measuring the collision products, the resulting data must be analysed to study the nature of the observed particles.

Particle accelerators, that focus on delivering the particle beams, do not measure most of the data from the particle collisions. There are dedicated particle detectors responsible for measurements regarding different properties of particles produced in the collisions, such as their origin, direction, timing, electric charge, and energy. These measurements allow physicists to identify what particles were present in the collision. Modern detectors are made of several layers of detectors themselves. Each of those sub-detectors is custom-designed to measure specific properties of particles. Some types of detectors include:

1. Trackers, with the main purpose of uncovering charged particle trajectories.
2. Calorimeters, with the main purpose of measuring energy. Usually, calorimeters try to stop the particles leading to the deposition of all their energy. This energy deposition is measured as the particle travels across the calorimeter.
3. Particle-identification detectors, with the purpose of determining a particle’s identity based on its velocity.

The data from the different sub-detector layers are then combined to recreate the collision moment.

1.2 The LHC accelerator and the CMS detector

The European Organization for Nuclear Research (CERN) hosts the Large Hadron Collider (LHC), the world's largest particle accelerator, near Geneva, Switzerland. The LHC covers a 27 km circumference colliding beams of particles¹ that circulate at 99.999 999 1 % of the speed of light, or 13.6 TeV of energy, every 25 ns. These collisions allow the creation and measurement of elementary particles that do not occur naturally in the universe since instants after the Big Bang. Around the accelerator there are four points, shown in Figure 1.1, in which particle beams collide. At each of these points particle detectors gather data from the collisions, contributing to expanding human understanding of topics such as the nature of matter and the beginning of the universe. This thesis' work contributed to one of the two larger detectors of the LHC, the Compact Muon Solenoid (CMS) detector.

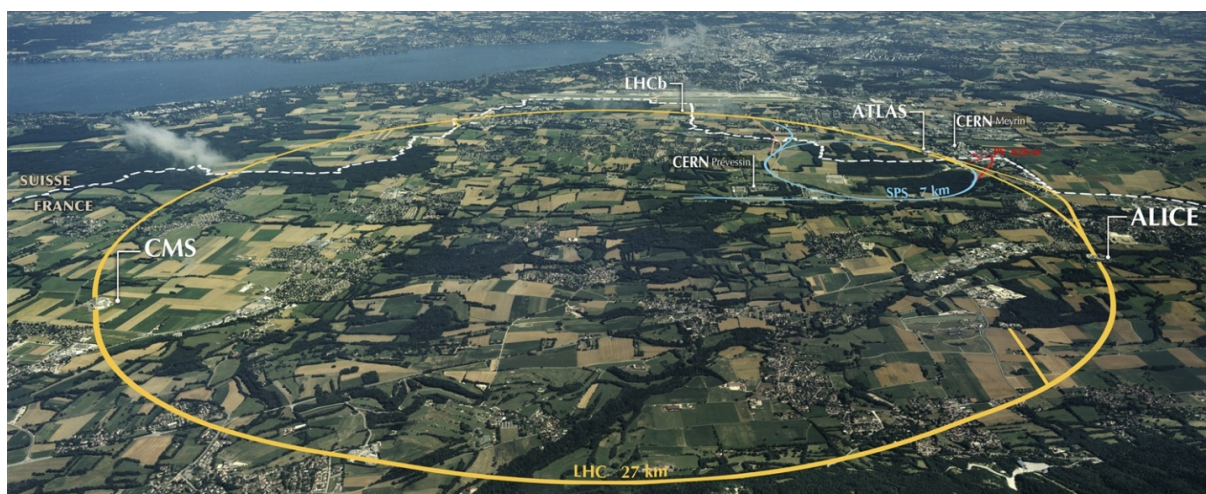


Figure 1.1: The LHC and its four collision points. Other four access points are used for the accelerator proper, namely the accelerating cavities, beam cleaning (in space and energy), and the beam dump. The CMS detector is hosted at the LHC Point 5, on the far left of the image [1].

The CMS detector [9], shown in Figure 1.2, is a general-purpose detector, composed of several concentric layers, designed to observe all possible particles. A magnetic field of 3.8 T, about 100 000 times stronger than our planet's, bends the trajectory of electrically-charged particles produced in the collision point, allowing the determination of their momentum, energy, and electric charge. Each of CMS's sub-detectors performs a specific task to accomplish this goal:

1. The silicon tracker, the innermost layer, takes precise position measurements of charged particles to reconstruct their trajectories.
2. The electromagnetic and hadronic calorimeters measure the energy of particles by stopping them.

¹ Mostly proton-proton collisions, but lead and other ion species are also collided.

3. The muon chambers, the outermost layer, track muon trajectories, as this particle is not stopped by the calorimeters, allowing to measure momentum.

CMS DETECTOR

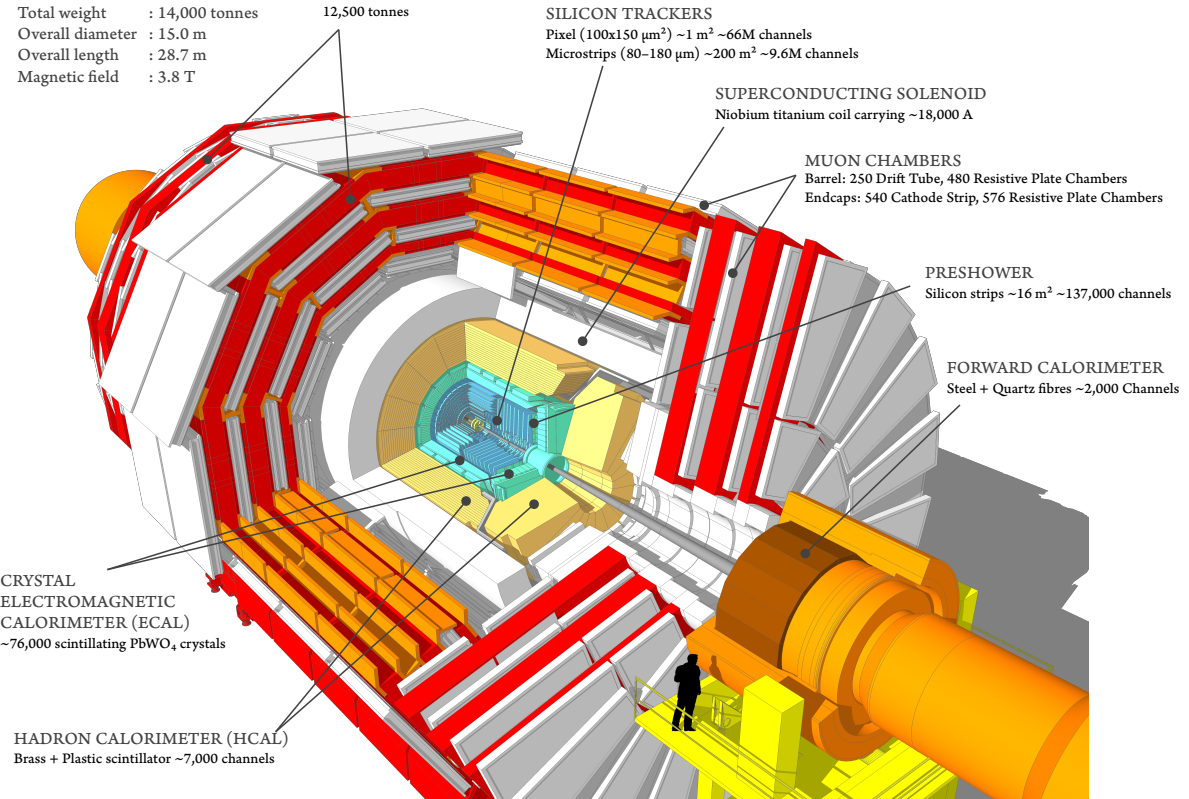


Figure 1.2: Global view of the CMS detector and its main sub-detector systems. These systems are hosted in five large wheels in the barrel region and three endcap disks at either end of the barrel [2].

Since the beginning of its operation, CMS (and the remaining experiments of the LHC) already contributed to significant advances in high-energy physics. One particular (and notable example) was the observation of a particle consistent with the Higgs Boson, which later led to the award of the Nobel Prize in physics in 2013 to François Englert and Peter Higgs.

Presently in its third run, the LHC has increased the energy of its collisions and particles since 2010, when it was first operated. However, the LHC and its detectors are reaching their respective performance limit regarding this energy. Hence, the upcoming High-Luminosity LHC (HL-LHC) project will upgrade the LHC to allow further increases in the number of collisions. The HL-LHC upgrade of the LHC is expected to be finished by the end of 2028.

However, the aimed increase in the number of collisions that will be provided by the HL-LHC will cause the detectors to be subjected to higher radiation levels and to have to process and communicate larger amounts of data. Therefore, to keep up with the HL-LHC's more demanding operation condi-

tions, CMS is preparing upgrades to some of its sub-detectors. As an example, the current endcap calorimeters of CMS will not be able to operate under the HL-LHC conditions without an unacceptable performance decrease.

Accordingly, this thesis was developed in the scope of one of such upgrade projects, the High-Granularity Calorimeter (HGCAL) [10] [11]. This new detector will replace the current CMS endcap calorimeters, both electromagnetic and hadronic, providing measurements of the timing, position, and energy of particles produced in the collisions with unprecedented high resolution.

1.3 The CMS High-Granularity Calorimeter

The HGCAL will have more than 6 million active channels spread across 47 layers grouped into two sections: the Electromagnetic Endcap Calorimeter (CE) (CE-E) and Hadronic CE (CE-H). The different radiation field intensities across the detector volume motivated the choice of two different sensitive materials: silicon sensors for the high radiation volume, and plastic scintillator tiles for the lower radiation volumes. Hence, while the CE-H will use both materials, the CE-E will only use silicon sensors, as shown in Figure 1.3. With a minimum cell size of 0.5 cm^2 for the silicon sensors and 4 cm^2 for the plastic scintillator tiles, the HGCAL will measure the position, timing, and energy of incoming particles with unprecedented high spatial resolution.

The HGCAL was designed to gather high-resolution data of the collisions at a 40 MHz sampling rate that matches the rate at which proton bunches collide. As each event is expected to generate between 2.5 MB and 3.5 MB [12], it is not possible to store all the data being produced. Moreover, not all collisions contain interesting events that allow to gain new knowledge about nature. For this reason, a stream of lower-resolution data is processed at 40 MHz in order to decide within a latency of $12\mu\text{s}$ whether to store the high-resolution data that can be read out from the detector at a rate of up to 750 kHz. The high-resolution data is further processed by a software system and only up to 7500 events per second are sent to permanent storage. The two different resolution data have two separate readout paths:

1. The Data Acquisition (DAQ) path, responsible for the readout of the high-resolution data, and
2. The Trigger path, responsible for the readout of the lower-resolution data and for deciding whether to trigger the readout of the high-resolution data.

The two paths are split directly inside the Front-end (FE) electronics that are mounted on-detector, and the off-detector Back-end (BE) electronics architecture reflects that division.

The high radiation environment of the detector requires the usage of radiation-tolerant Application Specific Integrated Circuits (ASICs) in the FE. About 150 000 such ASICs will read out and process the sensors' data, performing digitisation and a first level of pre-processing. The obtained data is then

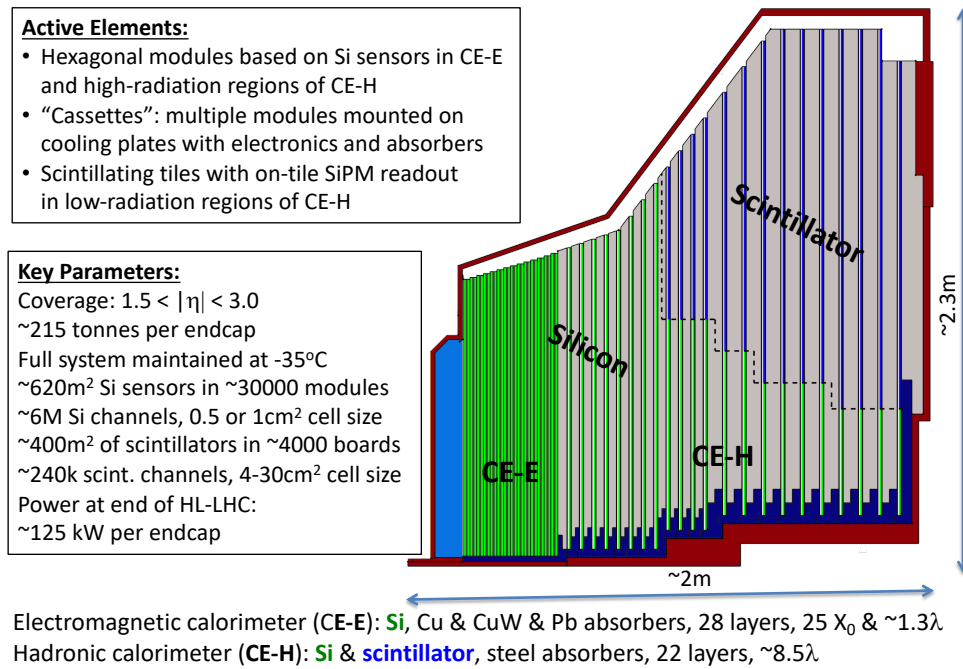


Figure 1.3: Summary of the main physical HGCal parameters and transverse cut view of the top half of the detector layers. The CE-E and CE-H sections can be seen to use only radiation-hard silicon sensors (CE-E) or a mix of both silicon and scintillating tiles (CE-H). Scintillating tiles are used in the volumes with lower radiation fields. From [3].

relayed via about 9000 optical links, each providing a 10.24 Gbit/s transmission rate from the detector cavern into the service cavern where the radiation field is of no concern.

The BE electronics are hosted in the service cavern and comprise a communication infrastructure composed of a significant number of Advanced Telecommunications Computing Architecture (ATCA) boards and crates. Each BE board will host one (or two) Virtex Ultrascale+ Field Programmable Gate Arrays (FPGAs), totalling about 100 FPGAs on the DAQ path alone. The BE is responsible for the reception and further processing of the data before conveying it to the central CMS processing system. Additionally, the DAQ path is responsible for configuring all the FE electronics during the detector operation. Two types of commands are necessary to configure and control the ASICs in the FE:

1. Synchronous commands (fast control) responsible for managing the data acquisition process on an event-by-event basis, e.g., ensuring the BE buffers do not overflow and conveying trigger path commands to start the high-resolution data acquisition. These commands are transferred in 320 Mbit/s serial bit streams interpreted as 8 bit codes at 40 MHz.
2. Asynchronous commands (slow control) perform the configuration of the parameters of the FE ASICs on a per-run basis, i.e., for sets of many events. These commands use 80 Mbit/s bit streams to drive many Inter-Integrated Circuit (I2C) masters.

Each FPGA will have to serve about 108 DAQ and control links, totalling about 1 Tbit/s of input data, and the slow control and fast control of up to 3500 FE ASICs.

1.4 Objectives

The large number of ASICs produced for the HGCAL puts tight constraints on the time budget for the corresponding production tests that need to be performed before the assembly of those same ASICs. Therefore, the test system used to perform these tests needs to be as efficient as possible to maximise the amount of testing that each ASIC can undergo within a fixed time budget. This is especially important for the HGCAL Readout Chip (ROC) (HGCROC), the most numerous ASIC, totalling about 100 000 in the HGCAL.

Considering the cost and complexity of the HGCAL, it was decided to vertically prototype the electronics systems to thoroughly verify the functionality of the system. This prototype will connect the sensors in the FE to the BE FPGA with all extant components, allowing to simulate the data acquisition process during the detector operation. As part of this effort, smaller parts of the detector are individually prototyped and validated. These include the slow-control block, the BE part responsible for the configuration of all FE ASICs during the detector commissioning and operation.

Accordingly, this thesis has two distinct objectives:

1. The acceleration of the characterisation testing and production testing of the HGCROC ASIC in the FE system, and
2. The implementation, integration, and validation of a final-like architecture for the BE slow-control block.

1.5 Requirements

Both objectives involved working with Virtex Ultrascale+ FPGA technology in the development of Intellectual Property (IP) cores. As these IPs are meant to be integrated into different systems, different resource requirements apply to each of them.

A new IP was developed to reduce the testing time of the HGCROC. As this IP is meant to be integrated with others inside a test system, it must not use more than about 2500 Look Up Tables (LUTs) or Flip-Flops (FFs).

Similarly, the slow-control block must be integrated in the BE FPGA design and implemented in a Virtex Ultrascale+ VU13P FPGA. Hence, it must not use more than 3% of the resources of the VU13P except for Block Random Access Memory (RAM)s (BRAMs). Moreover, at the beginning of a

physics run when the FE electronics are being calibrated, the HGCAL configuration time should be in the order of a few minutes to minimise the dead time of the detector. Hence, the slow-control block must have a performance able to comply with this constraint.

1.6 Contributions

To reduce the testing time of the HGCROC, a circuit was developed to perform real-time computations simplifying the calculation of the mean and standard deviation of HGCROC per-channel data. This circuit can be used in different FE test systems and can process data from up to 6 HGCROCs simultaneously. Its usage in the HGCROC test systems allowed to speed up the data readout and processing steps of the tests. Speed-ups of up to 4 times faster characterisation and testing were obtained and its integration in the production tests is an ongoing process as it allows for 4 times more detail in production testing for a constant test time budget.

To validate the functionality of the slow-control block, the previously existing slow-control architecture was tested, debugged, and additional features were added to equip this circuit with error-handling capabilities. By developing a prototyping system, it was possible to validate the different slow-control functionalities with real hardware. This includes the validation of the communication protocols between the slow-control block and the different FE ASICs interfaced by it.

This prototyping system was built ahead of the availability of the final system's FPGA and ATCA boards and surrogates were used for both BE and FE parts of HGCAL in the form of development boards. Despite the final detector boards being different from the ones used, the presence of the final detector ASICs allowed to reliably test and validate the respective communication chains. This work also enabled the use of the slow-control block in other HGCAL prototypes, allowing to employ the final detector framework for FE configuration. In turn, this is now allowing to develop the final software. Moreover, a comprehensive evaluation of the portability of the slow-control block was conducted and the configuration time of hundreds of thousands of ASICs in the full HGCAL detector was estimated.

The obtained results of the slow-control prototype system resulted in the publication of the following article in an international conference (see Appendix A):

- M. Rosado, S. Mallios, P. Tomás, N. Roma, and A. David, “Early prototyping and testing of CERN LHC CMS high-granularity calorimeter slow-control system,” in International Workshop on Rapid System Prototyping (RSP), Oct. 2022 [4].

1.7 Organisation of this document

The remainder of this thesis is organised as follows: Chapter 2 describes the HGCAL electronics chain, including both the FE and BE systems, providing further context of the work performed and the achieved objectives. Chapter 3 describes the challenges of the characterisation tests of the HGCROC and the circuit developed to speed up this process. Chapter 4 describes the work done regarding the slow-control block as well the improvements, prototyping, and validation. Finally, Chapter 5 presents conclusions and possible lines of future work.

2

HGCAL Electronic Systems

Contents

2.1 HGCAL Front-end (on detector) electronics	14
2.2 HGCAL Back-end (off detector)	18
2.3 Contributions to HGCAL electronics prototyping	21
2.4 Summary	23

Upon a particle collision (event), the HGCal sensors measure the position, timing, and energy of the involved particles. These data are then digitised and communicated to the central CMS processing system. However, with the 40 MHz sampling rate at which the data is being produced, it is not possible to store all of it as each event produces a minimum of 2.5 MB of data.

A possible solution would be to reduce the sampling rate. This is not the optimal solution because not all the events are considered to be of interest for later physics analysis. A better solution to this problem is to develop a system that decides, in real-time, which events to store, allowing the sampling rate to remain the same while discarding uninteresting events. This subdivides the HGCal electronic readout chain into two paths: the trigger and the DAQ.

The trigger path is used for this decision that “triggers” the DAQ path. To efficiently choose which events to store, the trigger path needs to make a time-constrained decision. Accordingly, it is not possible to make this decision based on the high-resolution data being produced. In the trigger path, the data is reduced via lossy compression. Afterwards, the trigger path processes these data and decides if the corresponding event is interesting and grants to collect the high-resolution data. When it is the case, the DAQ path is activated through a Level-1 Accept (L1A) fast command (see Section 2.1) and communicates the full-granularity data of that event to the central CMS processing system.

The DAQ path is responsible for the transmission of the full-resolution data to the central CMS processing system. The same path is used to configure and control the ASICs used in the detector. The DAQ must be able to read out all the 6 million channels of the detector (zero-suppressed) at a maximum L1A rate of 750 kHz. As a result, on average, the DAQ is expected to have a data rate of 15 Tbit/s. Accordingly, the work conducted in the scope of this thesis is focused on the DAQ path, through which the ASIC configuration is performed. As such, the trigger path will not be discussed in detail.

Due to the large amount of generated data, some amount of data processing and selection is required early in the chain. In fact, only part of this process is going to be done on the detector, where the radiation constraints are important. Nevertheless, due to the complexity of the processing, part of this process cannot be done with radiation-tolerant electronics. Accordingly, the readout chain is subdivided into FE (on-detector electronics) and BE (off-detector electronics). This division also applies to the trigger path. Figure 2.1 describes the HGCal high-level architecture and this separation.

The FE comprises the sensors and custom radiation-tolerant ASICs that will digitise, do a first level of processing, and transmit sensor data to the BE. This connection will be established via about 9000 optical links, each providing a 10.24 Gbit/s transmission rate. The BE, shielded from radiation, consists of a considerable amount of ATCA boards and crates hosting Multiprocessor System On a Chip (SoC) (MPSoC)s and large FPGAs. These devices will receive FE data, perform aggregation with very large fan-in fan-out ratios and more computationally intensive processing, and transmit data to the central CMS processing system. The BE is also responsible for controlling the FE ASICs according to external

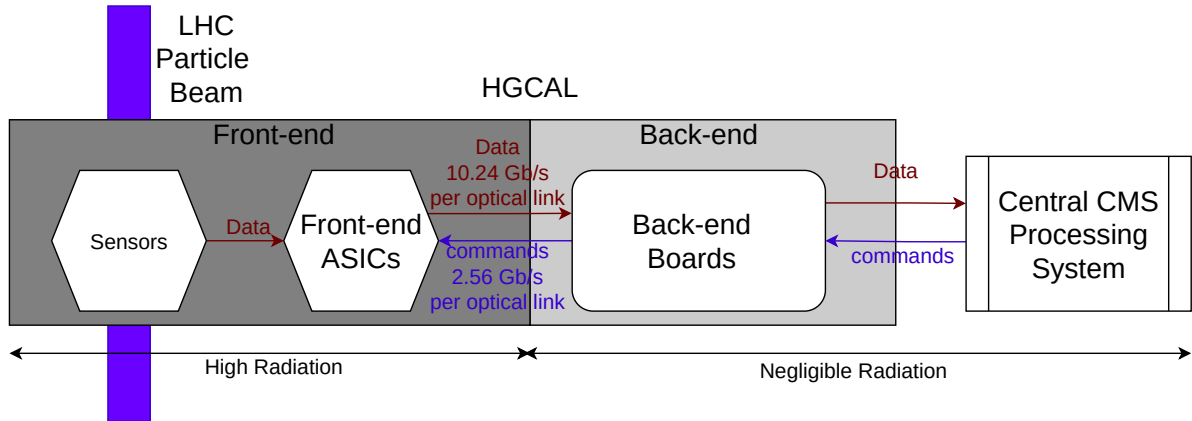


Figure 2.1: High level architecture of the HGCal electronics. The most important division is between the on-detector FE electronics in a high radiation area and the off-detector BE electronics in a zone with a negligible radiation field. The two are connected via optical fibre runs of about 100 m. From [4].

commands that reach the HGCal, coming from the central CMS system. This control connection is also established via optical links, working at a transmission rate of 2.56 Gbit/s.

2.1 HGCal Front-end (on detector) electronics

The LHC protons are grouped in bunches and every 25 ns a bunch of protons completes an orbit¹ and collide. The FE electronics [13] consist of a system of about 150 000 radiation tolerant ASICs which read the data from the sensors and do a first level of processing. This process also includes event building which consists in guaranteeing that the data from the same event is packed together. I.e., data needs to be coherent in three aspects: the triggered event, the LHC orbit, and the LHC Bunch Crossing (BX). Each of these has an associated counter that is used to “tag” the corresponding data.

To take advantage of the different radiation tolerance of materials, two different sensors will be used in the HGCal according to the radiation levels in the volume of the detector:

- Silicon sensors in volumes with high radiation.
- Plastic scintillator tiles in volumes with lower radiation.

Hence, the two different sensitive materials correlate two different FE electronics trees, as described in Figure 2.2.

The sensitive elements of the detector total about 6 million channels that are read out by the HGCROC ASICs. The HGCROCs perform three types of measurement of sensors’ signals: Analogue to Digital Converter (ADC) for small deposited charges, Time over Threshold (TOT) for very large charge deposits that saturate the pre-amplifier, and Time of Arrival (TOA) that measures the timing of the charge deposit.

¹An orbit corresponds to the duration of one full revolution of the LHC particles in the accelerator.

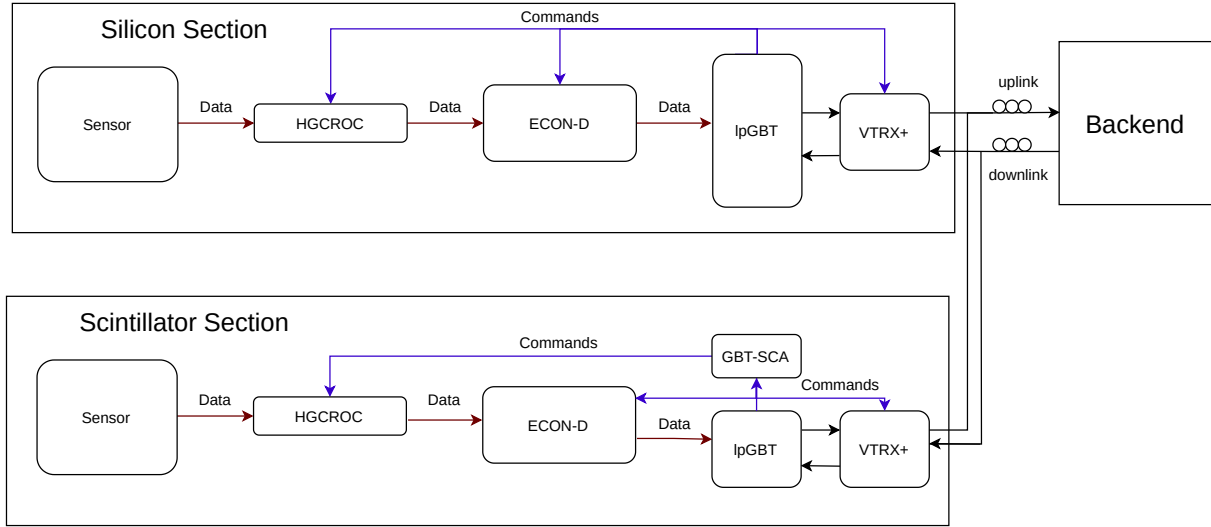


Figure 2.2: DAQ data and slow command flow on the HGCAL FE electronics for the two detector technologies. The use of the GigaBit Transceiver (GBT) Slow Control Adapter (GBT-SCA) in the scintillator section reflects the presence of additional ASICs in this part of the detector related to the bias voltage of the silicon photomultiplier technology used to convert the scintillating light into electrical signals (not depicted).

Each HGCROC is subdivided into two separate and equal halves each with 36 channels measuring the three mentioned types of data. Additionally, each half HGCROC has two common mode channels and a calibration channel. The HGCROC is the first link in the electronic readout chain described in Figure 2.2.

Then, the HGCROC high-resolution data are sent to the concentrator ASICs, the DAQ Scalable Low Voltage Signalling (SLVS) electrical link (e-link) Concentrator (ECON)s (ECON-Ds). The ECON-D performs zero suppression on the data and packages together data from the same event. The trigger path also has its concentrator, the Trigger ECON (ECON-T), that receives HGCROC trigger data. It can be noted here that due to the constraints on data transmission, the trigger path includes neither the full granularity nor the TOA information.

ECON-D output data is communicated to the DAQ BE. The frontend interface with the backend is done via custom ASICs, the Low-Power GBT (lpGBT) [5] that provides high-speed bidirectional communication: the downlink (BE to FE, used for control and configuration) and the uplink (FE to BE, used for ECON-D data). lpGBTs receive data from up to 7 ECON-Ds, serialise them, and send them to the BE using the uplink. The Versatile Link Plus Transceiver (VTRX+) ASIC [14] is used to convert the electrical signals from the lpGBT into optical signals and interface the BE via optical fibres completing the FE readout chain.

All these FE ASICs need to be configured. To fulfil this configuration, the downlink conveys configuration and control data from the BE to the FE. These commands can be divided into two types:

1. Fast Commands are synchronous commands with respect to the BXs, and thus the data taking. They control the data flow in the detector. The L1A signals are conveyed to the FE via the fast

command stream and other aspects of data acquisition are also managed (e.g., firing of calibration pulses, or the decision to not perform zero suppression on a given event's data). A 320 Mbit/s serial bit stream is used to convey fast commands from the BE to the FE.

2. Slow Control commands are asynchronous with respect to the BXs. They are used to configure and monitor the FE ASIC parameters. An 80 Mbit/s serial bit stream is used for these commands since ASIC configuration is not changed often during physics data-taking.

Depending on the detector section, the slow control commands can be conveyed to the target ASICs through the IpGBT alone or the IpGBT and the GBT-SCA [15], as shown in Figure 2.2. The GBT-SCA is an ASIC that conveys configuration and monitoring commands throughout the FE of particle detectors. The IpGBT, GBT-SCA, and VTRX+ are part of a family of common ASICs developed for all LHC experiments [16].

Depending on the considered section of HGCal, the architecture of the FE will have slight changes while maintaining this data flow.

2.1.1 Silicon Section electronics tree

On the silicon section of the FE [13], the HGCROCs are mounted in hexagonal Printed Circuit Boards (PCBs), called hexaboards, that are assembled into modules that include the sensitive elements as well. These hexaboards are subdivided into Low Density (LD) and High Density (HD) hexaboards [17]. LD full hexaboards have three HGCROCs while HD full hexaboards have six. Given how it is not possible to perfectly cover a circular area with full hexagons, each of these types also include “partial” hexaboards with a variable number of HGCROCs. The differences between the LD and HD hexaboards imply two different electronics trees, as described in Figure 2.3.

The silicon section FE electronics are organised in so-called trains, including engines and wagons as described below. Engines are complex, small, identical, and costly PCBs. Wagons, on the other hand, are simple, large, varied, and inexpensive PCBs.

The IpGBTs and VTRX+s are mounted on an engine to ensure communication. Engines also come in LD and HD varieties, associated with the respective hexaboards. The number of hexaboards connected to each engine type varies, with up to seven LD hexaboards per LD engine when compared to up to four hexaboards per engine in the HD regions.

Hexaboards are connected to the engines via a wagon. The ECON-D and ECON-T ASICs are mounted on the wagons in HD regions. In the LD regions, each LD hexaboard hosts a mezzanine with an ECON-D and an ECON-T, each responsible for processing the data from the HGCROCs on that LD hexaboard.

Each IpGBT transmits either trigger or DAQ data, according to their configuration and is associated

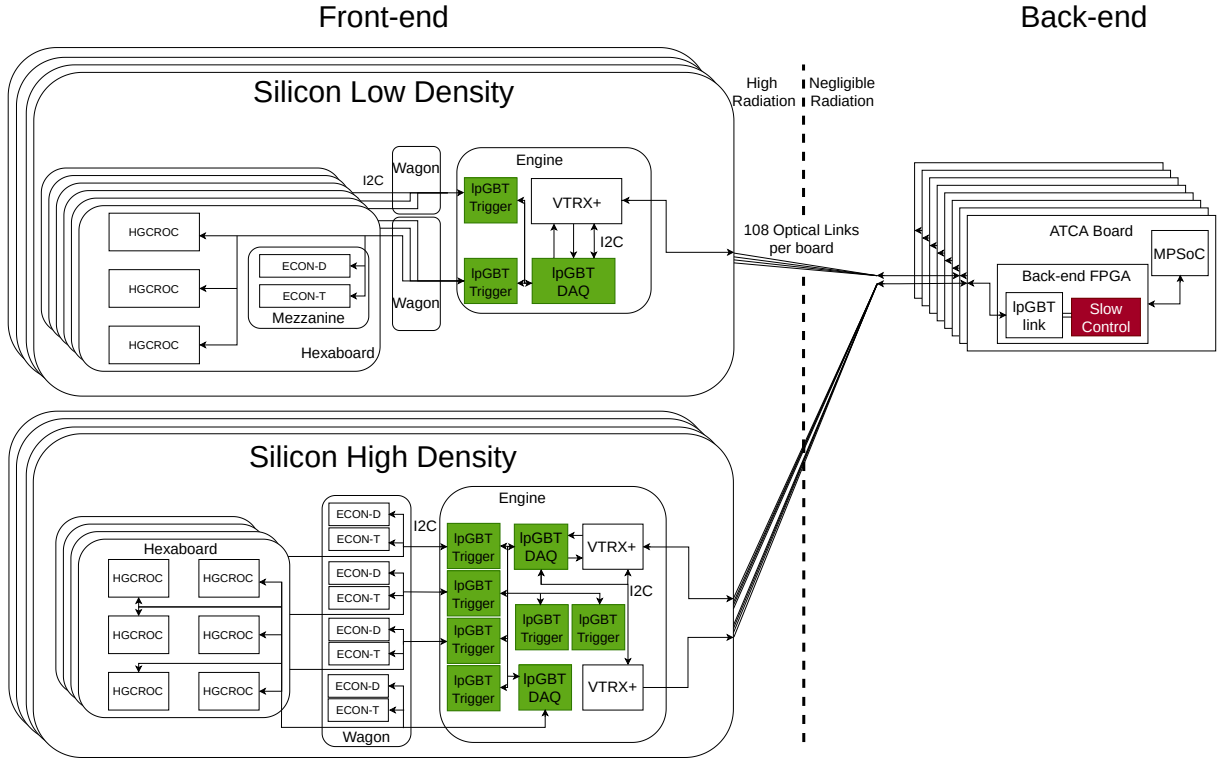


Figure 2.3: Slow control command architectures for the HGCAL silicon FE. The differences between the LD and HD are not qualitative, but quantitative, as the same components are arranged differently in the two trees.

with the respective data path. Therefore, we speak of trigger IpGBTs and DAQ IpGBTs. The DAQ IpGBTs also receive the control signals from the BE.

Each LD engine has one VTRX+ and three IpGBTs, namely two trigger and one DAQ. Each HD engine can have up to two VTRX+s and up to 8 IpGBTs, namely up to six trigger and up to two DAQ. An optical connection to the BE is associated with each VTRX+. The uplinks work at 10.24 Gbit/s and the downlinks at 2.56 Gbit/s.

The connections shown in Figure 2.3 describe the slow control command distribution chain. Slow control command data are received at the FE from the BE via the downlink and are distributed via the DAQ IpGBT. They can be directly sent to the target ASICs (HGCROC, ECON-D, ECON-T, or VTRX+) or to another IpGBT. The final distribution of data is done via I2C transactions, except for IpGBTs, which use their own internal protocol. An IpGBT other than the DAQ IpGBT can be considered to be directly interfaced by the BE as their data are treated as transparent by the previous IpGBT in the chain. Hence, only IpGBTs are directly interfaced by the BE and all other ASICs are indirectly interfaced. Responses to the slow control commands travel the same distribution chain in the opposite direction and are sent to the BE via the uplink.

2.1.2 Scintillator Section electronics tree

On the scintillator section, one HGCROC and one GBT-SCA are mounted on a PCB tileboard, associated with a set of sensitive scintillator tiles. Up to 5 tileboards can be connected to a scintillator motherboard with two IpGBTs, two ECON-Ts, and one ECON-D, as shown on Figure 2.4. Similarly to the silicon section, the number of tileboards associated with each motherboard is variable.

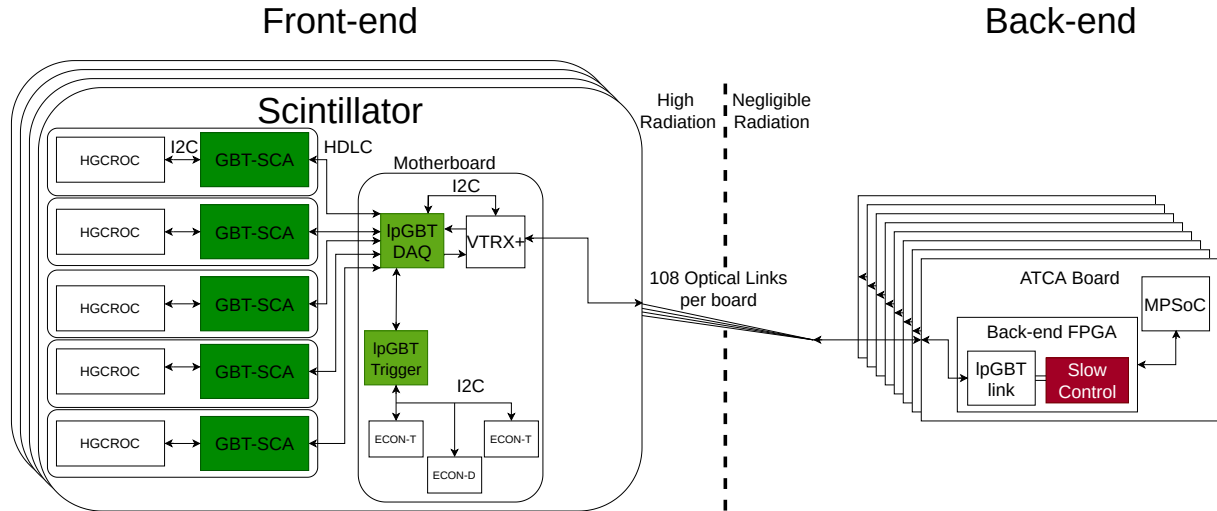


Figure 2.4: Slow control command architectures for the HGCAL scintillator FE. The main difference to the silicon FE is the presence of the GBT-SCA due to the presence of many peripherals in a tileboard and the larger distance between IpGBT and HGCROC.

The slow control command distribution on the scintillator section also uses GBT-SCAs to distribute data to the target ASICs. As in the silicon section, slow commands use the I2C protocol to be distributed, though there is a more pervasive use of General-Purpose Input/Output (GPIO) and other GBT-SCA features in the tileboards. The IpGBTs and GBT-SCAs use different protocols, with the IpGBTs using their own protocol and the GBT-SCAs using the High-level Data Link Control (HDLC) protocol [18]. Even if the slow control command data goes through a IpGBT, the GBT-SCAs can also be considered as directly interfaced by the BE since the IpGBTs also treats GBT-SCA data as transparent.

2.2 HGCAL Back-end (off detector)

The DAQ BE is located outside the detector cavern, in a negligible radiation environment, and consists of ATCA boards, each comprising one or two FPGAs and a MPSoC, for a total of about 100 FPGAs, as described in Figure 2.3 and Figure 2.4. Each BE FPGA performs further event building of the data received from up to 648 ECON-Ds, collating it together into larger packets. Event data are then packed and forwarded to the central CMS DAQ system, where the HGCAL data is merged with data from

other CMS detector systems. The BE also handles the transmission of slow and fast commands to the FE ASICs. To ensure this task, each FPGA implements an identical design [19] that performs these functionalities and interfaces the FE and central CMS DAQ.

As it was referred before, the FE is connected to the BE via optical fibres. Each ATCA FPGA will have 108 connections to the FE ASICs consisting of an uplink and a downlink. This totals about 1 Tbit/s of input data per BE FPGA, as each uplink connection works at 10.24 Gbit/s. Each FPGA outputs data to the central CMS DAQ through 12 optical fibre links operated at 25 Gbit/s.

It is worth noting that it is not required for the BE DAQ path to have a fixed latency, and data from each event must be processed before the arrival of the next. Hence, deep buffers are required within the FPGAs. As a result, due to the high volume of data per event arriving at the BE, it was decided to use Xilinx UltraScale+ VU13P FPGAs to implement this part as they have enough resources to implement the BE design. In fact, event building requires the largest share of resources of the FPGAs: about 40% of the device's LUTs and 30% of flip-flops. Moreover, this design is particularly demanding on routing resources due to its tight timing constraints, a consequence of having to route the data from 648 ECON-Ds into only 12 output links. Hence, other functionalities implemented in the same FPGA device, like slow control, fast control, or monitoring, end up being resource constrained since the event building limits the resource availability. In particular, the slow control block can only use up to 3% of the resources with the exception of BRAMs.

2.2.1 Slow Control block

As it was referred before, the slow control block monitors and configures all FE ASICs. This is achieved by conveying, via multiple channels, several transactions which consist of read and write operations to internal registers of the FE ASICs. This process consists of decoding and conveying slow control command data to the FE as well as receiving and processing the respective FE responses. All these data are propagated to the BE from software running on the MPSoC hosted on the BE board. Since this MPSoC is embedded in a Xilinx Zynq UltraScale+ device, the connection between the FPGA and the MPSoC is done via an Advanced eXtensible Interface (AXI) Chip2Chip [20] connection.

After decoding the slow-control data received from the MPSoC, the slow-control block has to directly interface two types of FE ASICs: IpGBTs and GBT-SCAs. If a transaction needs to be conveyed to another ASIC further down the FE, the I2C masters of either the IpGBT or GBT-SCA will transmit it. Hence, the two protocols handled by the slow-control block are the IpGBT protocol and the HDLC protocol. Figure 2.3 and Figure 2.4 highlight in green these ASICs and their place in the FE architectures. It should be noted that each slow-control transaction triggers a reply from the corresponding IpGBT or GBT-SCA. Each BE FPGA directly interfaces up to about 750 of these ASICs, behind which up to about 3500 ASICs are configured in total.

It should be noted that the mapping of FE fibre connections to BE FPGAs is not finished and still under development. Hence, there is the need to consider this worst-case scenario of 3500 ASICs connected to a FPGA to guarantee that the performance of the slow-control block meets the required configuration time for HGCal. It is considered this scenario involves 108 connections to LD engines. This is so as the scintillator section has fewer ASICs behind each fibre connection, and there are fewer HD engines which are also connected to fewer hexaboards. Furthermore, it is imposed to the optical fibre mapping that no more than 12 ECON-Ds can be connected per 2 optical fibres to each BE FPGA. Therefore, this worst-case scenario comprises a mean of 6 modules connected to each engine. As each module comprises 5 ASICs, each engine is connected to 30 ASICs plus its own 3 IpGBTs. With a FPGA connected to 108 modules, this totals about 3500 ASICs.

The following procedure is used to communicate commands between the BE and FE using the slow-control block:

1. The software running on the MPSoC transfers data to the slow-control block.
2. The slow-control block decodes and processes the received data issuing the necessary read and write transactions to the target IpGBT or GBT-SCA.
3. The replies from the FE are stored in the slow-control block.
4. The software on the MPSoC reads the reply data.

Each slow-control channel uses a pair of 80 Mbit/s streams to communicate with the FE, one for sending and another for receiving data. This data rate is slower when compared to the fast command control signals sent to the FE because ASIC configuration does not change often and thus there is no need for a faster communication link. Even if the configuration of the FE ASICs does not change often, this procedure is time-constrained, as the detector needs to be configured quickly at the beginning of a physics run to minimise dead time and not miss important events. Hence, the complete HGCal configuration should not take more than a few minutes to happen.

2.2.2 The IpGBT Protocol

The communication between each IpGBT connected to the BE uses the data frame described in Figure 2.5. The Internal Control (IC) and External Control (EC) fields are used to communicate slow-commands to the IpGBTs, each corresponding to a slow-control channel, encoded as a 2 bit signal clocked at 40 MHz. The IC field is used to internally address the IpGBT directly connected to the VTRX+, while the EC field is used to address other, external, IpGBTs. Both connections can be considered as direct to the BE since the EC data is transparent to the IpGBT relaying them. In each engine, only one IC stream and one EC stream are available so that the EC stream is usually shared by multiple IpGBTs.

To reach the GBT-SCAs, some IpGBT e-links [21] are used to relay data to the GBT-SCAs, transparently to the IpGBT.

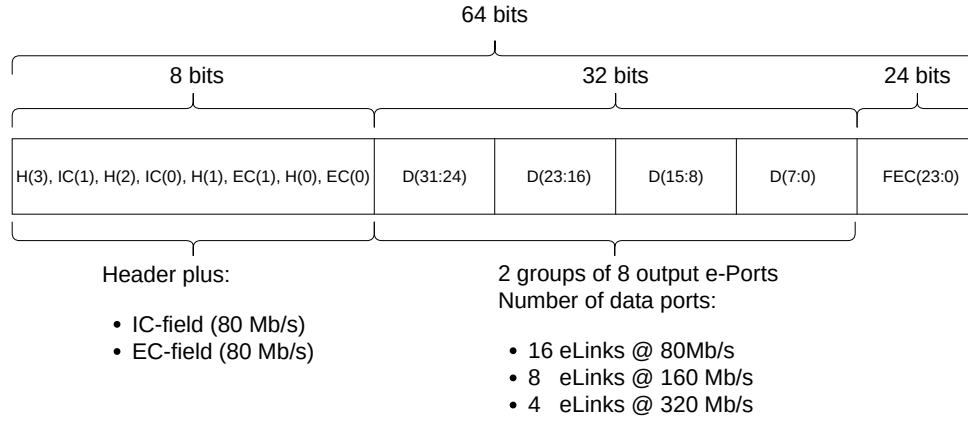


Figure 2.5: The IpGBT downlink data frame format from BE to FE. One can see the IC, EC, and e-link fields that in HGCal are used to communicate with the FE ASICs. From [5].

While the slow-control data is encoded to be compatible with these 80 Mbit/s streams, it cannot be directly connected to the IpGBT as it does not comprise the whole frame. The IpGBT link hardware block [22] is provided by the IpGBT group of CERN, and it is responsible for translating data into this protocol. As each optical link is associated with one IpGBT connection to the FE and thus one such data frame, one instance of the IpGBT link is required per connection. Hence, the outputs of the slow-control block must be connected to these hardware instances, as shown in Figure 2.3 and Figure 2.4 and highlighted in red.

The fields of the IpGBT frame that are not related to the slow-control correspond to data fields, the header and the Forward Error Correction (FEC) field. The latter is used to correct bit flips induced by the high radiation environment.

2.3 Contributions to HGCal electronics prototyping

The complexity of HGCal requires the development of testing and prototyping systems to validate the functionalities of its components before their final form is achieved. As such, to thoroughly validate the feasibility of the concept, it was required to prototype a vertical slice of the HGCal electronics system. This system comprises a simple connection with the minimum number of relevant components from the sensors on the FE to the BE output to the central CMS processing system. As the testing progresses, it is expected for this system to also grow horizontally and integrate more components of the same type up to the dimensions of the final HGCal prototype.

As it was referred in Chapter 1, this thesis contributes to two aspects involved in the implementation of

this prototype: the characterisation testing of the HGCROC ASICs, and the individual prototyping of the slow-control block. Both contributions involve the development of FPGA hardware in Very High Speed Integrated Circuit Hardware Description Language (VHDL) targeting a Xilinx UltraScale+ platform.

2.3.1 Accelerating HGCROC Testing

Due to intrinsic variability in the fabrication of its analogue blocks, each HGCROC needs to be put through a calibration procedure to perform the trimming of the analogue circuits. Around 100 000 HGCROCs will be used in HGCAL. This large amount of ASICs limits the time of the production tests that can be performed on each chip to one or two minutes. These tests are done to decide whether to assemble a given HGCROC on a hexaboard or tileboard. Moreover, characterisation tests use HGCROC data to characterise the HGCROC before its production phase and to prototype the hexaboard and tileboard systems in other test systems. All these tests rely on taking HGCROC data corresponding to a couple of thousand events to determine, for each HGCROC channel, the observed mean and standard deviation of measured quantities. E.g., the ADC pedestal and noise for a given channel correspond directly to the mean and the standard deviation of the set of ADC values.

The HGCAL FPGA-based test systems use Xilinx UltraScale+ Zynq devices to interface the HGCROCs under test. The custom hardware implemented at the Programmable Logic (PL) allows receiving and store data that is later read and processed by software. Such test systems will benefit from instantiating an IP block in the PL of the Zynq devices that will compute, in real-time, the mean and standard deviation values of the received HGCROC data. As these metrics are the main data of interest to the tests, there will be no need to read out the full data and do these computations over the full data in software. This will speed up the characterisation and production testing of the HGCROC by diminishing the reading overhead and accelerating computations.

Hence, the development of the Multiplexed Accumulator for ROCs (MARS) IP, that provides this acceleration, is the first contribution of this work to HGCAL and is further described in Chapter 3. The requirements for this IP are:

1. MARS must be connected to the Zynq Processing System (PS) via an AXI Lite interface.
2. The input data should be read as an AXI stream interface, as the data is already available in this format in the PL of the HGCROC test systems.
3. The expected input data is the output DAQ data of the HGCROC.
4. The output data must be provided for every HGCROC channel.
5. As MARS needs to be integrated with other IP blocks inside the PL, it must not use more than about 2500 LUTs or FFs.

6. Due to the use of MARS for different test systems handling different numbers of HGCROCs, MARS has to be configurable at synthesis time to adapt to the number of HGCROCs.

Besides being used in the characterisation tests of the HGCROC, this IP is meant to be also used in the production testing of the HGCROCs to be used in the final detector.

2.3.2 Slow control architecture validation and testing

Prototyping the slow-control block requires the validation of the connections to the MPSoC software via AXI, to the IpGBTs via the IpGBT protocol, and to the GBT-SCAs via HDLC. Before this work, the existing slow-control architecture was not tested, hence this thesis' second contribution to HGCAL, described in Chapter 4, is the prototyping of the slow-control block in such a test system that allows to:

1. Validate the slow-control interfaces using an AXI interface, and the interfaces connecting the IpGBT and GBT-SCA ASICs.
2. Debug and improve the existing architecture to ensure a reliable communication with both software and the FE.
3. Implement performance tests to the architecture to estimate the configuration time of the full HGCAL FE.

After this prototyping process, the slow-control block should comply with the following requirements:

1. Be compatible with both the design of the BE DAQ for the final detector system and other prototyping systems of HGCAL that use Xilinx FPGA technology.
2. Have an AXI Full interface to easily communicate with the MPSoC running the software.
3. Be compatible with the IpGBT link hardware block.
4. Use at most 3% of the VU13P FPGA resources, as specified by the BE DAQ FPGA design constraints.

If possible, the architecture should be also optimised during this testing, and additional features to prevent deadlocks and flag errors should also be added to the slow-control.

2.4 Summary

The brief introduction to the HGCAL electronic systems provided by this chapter explored the flow of data in this detector, the distribution of slow control commands across the FE and BE, and the intervening ASIC and FPGA components of HGCAL. Furthermore, this description allowed to better contextualise the contributions of Chapter 3 and Chapter 4.

3

Multiplexed Accumulator for RocS

Contents

3.1 HGCROC Calibration	27
3.2 Architectural Considerations	28
3.3 Initial Architecture	30
3.4 Pre-synthesis Configuration	33
3.5 Optimised Architecture	34
3.6 Resource Usage and Performance Results	38
3.7 Summary	40

Due to the fabrication process variability, each of the HGCROC's 78 channels has slightly different analogue properties that must be compensated for. Moreover, over the operational lifetime during which the ASICs will be irradiated, their analogue properties will evolve. The compensation is achieved using internal Digital to Analogue Converter (DAC) circuits that allow to bias each circuit individually with a tailored value that needs to be determined in order to trim the circuit. The trimmed parameter values are determined from the characterisation of each of the HGCROC's channels and their calibration. Both the characterisation and production tests to which the HGCROCs are subjected involve this per-channel parameter trimming procedure.

3.1 HGCROC Calibration

The large number of HGCROCs that need to be calibrated in production testing ($\sim 100\,000$) makes the overall process very long, given that there will be no more than 5 robots testing HGCROCs in parallel. In fact, each HGCROC has a testing time budget of no more than one or two minutes for the whole production testing process.

The calibration process involves running tests that acquire data from each HGCROC, allowing the determination of, the ADC pedestal values in each channel for example. The pedestal value can be tuned using the biases of the preamplifier and shaper, that in turn can be programmed through the use of a DAC internal to the HGCROC. This way, the analogue components can be calibrated and trimmed to minimise differences across channels.

Currently, these tests are run in different setups to accomplish different goals:

1. The prototyping of hexaboards: Depending on the type of hexaboard, data from either three (LD hexaboards) or six (HD hexaboards) HGCROCs is considered.
2. The prototyping of tileboards: The data of one HGCROC is considered.
3. The characterisation of a single HGCROC.
4. The production testing of HGCROCs on a large scale.
5. The production testing of several hexaboards: Up to 24 hexaboards will be tested comprising data from up to 72 HGCROCs.

Despite the differences between the tests of the different setups, all use Zynq SoCs hosted in custom boards and share the same calibration principle. The software running on the ARM processors configures the HGCROC using custom IP blocks in the Zynq PL and simulates sensor signals through injection of charge to the HGCROC inputs. The data from those signals are then read out upon a L1A fast command. Later, these HGCROC raw data are extracted from the Zynq, sent over a computer network, and

processed in a remote computer. A large part of the data processing is the computation of averages and standard deviations of ADC, TOA, or TOT values for each HGCROC channel. For a given channel we estimate these quantities following:

$$\text{Mean} = \frac{\sum_i X_i}{N} \quad (3.1)$$

$$\text{Variance} = \frac{\sum_i (X_i - \text{Mean})^2}{N - 1}, \quad (3.2)$$

with X_i being a particular event's datum and N the total number of events.

Another computation that is important in the calibration process is the per-channel count of non-zero measurements in the TOA and TOT measurements. These are used in determining the efficiency of those circuits as a function of threshold parameters, essential for trimming the thresholds.

Both types of computation involve data from a few thousand events.

This processing cannot be easily done in the Zynq MPSoC ARM cores due to the reading overhead of the current system. Although this could be mitigated through the use of a Direct Memory Access (DMA) readout, that would require significant changes to the software. Moreover, the ARM cores are already particularly overloaded running communication processes.

To accelerate this data processing step, the solution was to design a new IP, to embed in the HGCROC test systems. The IP uses Zynq PL resources to speed up this process by accumulating the values in real-time. This also reduces the communication and computation overhead, since only the final accumulated results are read out from the Zynq and the data for all events, that are not needed, are not ever transferred. This acceleration of the testing procedure will allow for more detailed tests to be implemented in the same time budget of each HGCROC.

3.2 Architectural Considerations

The MARS [7] is an IP described in VHDL language to speed up the HGCROC calibration procedures. To aid in the computation of the required metrics, the MARS separately accumulates either ADC, TOT, or TOA measurement from each channel over a couple of thousand events. The square of the data is also accumulated in parallel. Moreover, in another operation mode, "Time to Digital Converter (TDC) efficiency", MARS counts the amount of non-zero events for both TOA and TOT values in each channel. The IP is designed to be connected to a PS of a Zynq device via an AXI Lite interface.

The IP receives the data directly from the HGCROC and therefore parses the HGCROC DAQ data packet format, which is shown in Figure 3.1 for the HGCROC. The data is streamed in a 32 bit AXI stream bus, clocked at 40 MHz, for each HGCROC DAQ link.

Each half HGCROC has 36 channels plus 3 extra channels: one calibration channel (`calib`) and two common mode channels (`Common Mode (CM)`). The 32 bit words for each regular channel (`ch0` through

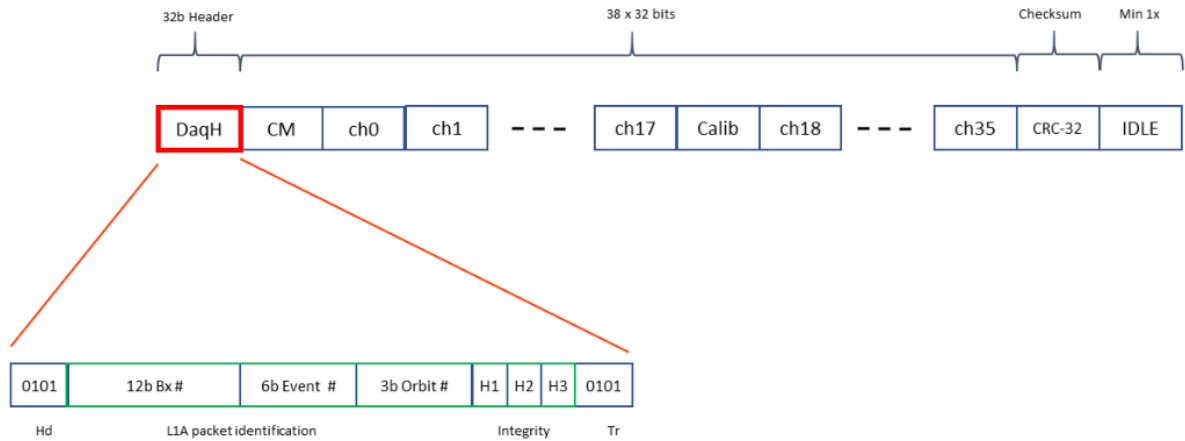


Figure 3.1: HGCROC DAQ data packet structure for one HGCROC half. In total, there are 41 words of 32 bit transmitted when a L1A trigger fast command is received. The data corresponds to all the relevant measurements in all channels. From [6].

ch35 and calib) can be decoded as shown in Figure 3.2. However, for the “special” CM channels, two ADC values for two channels are packed in a single word. These CM channels serve the purpose of estimating correlated noise affecting the ADC measurements in other channels. Therefore, the CM channels do not measure TOT or TOA, producing only a 10 bit ADC value each, so that the two can be compressed into a single word as shown in Figure 3.3.



Figure 3.2: HGCROCv3 word format for regular channels, namely ch0 through ch35 and Calib. From [6].



Figure 3.3: HGCROCv3 word format for the CM channels. From [6].

As described in Section 2.3.1, the MARS IP needs to process data from a varying amount of HGCROCs depending on which test system it is used. The data from a set of 39 HGCROC channels are transmitted in one HGCROC DAQ link. Hence, from Section 3.1, a MARS needs to process data from two to 12 HGCROC DAQ links. However, without optimisations to the existing baseline architecture, it is not possible to accumulate all links in parallel due to the FPGA resource constraint of Section 2.3.1.

The required bits to accommodate the arithmetic results increase in face of the original operands as a consequence of the involved operations. In particular, when multiplying two values the number of bits needed to store the result is the sum of the operands' widths. Therefore, when squaring a value its width

doubles. When adding N operands, $\log_2(N)$ additional bits are needed to store the sum.

Despite using only 10 bit in their encoding (see Figure 3.2), the arithmetic operations involving ADC, TOA, and TOT require a representation with at least 12 bit. The TOT measurement is compressed from an original width of 12 bit using a scheme that does not sacrifice its relative precision. In particular, if the most significant bit of the 10 bit TOT is 0, the two most significant bits of the 12 bit TOT are 0s followed by the 10 bit TOT word in the data packet. Otherwise, if the most significant bit of TOT 10 bit word is 1, the TOT 12 bit word is composed by the remaining 9 bit of the 10 bit TOT word in the most significant positions followed by three zeros in the least significant positions. Hence, the base width of the data to be accumulated in MARS should be 12 bit.

Assuming that the relative uncertainty of the measurements follows

$$\text{Relative Uncertainty} = \frac{1}{\sqrt{N}}, \quad (3.3)$$

an acceptable value for the minimum amount of accumulated number of events, N , was fixed at 1024, corresponding to $\sim 3\%$ relative uncertainty.

Therefore, one needs a minimum of

$$\text{Bit width for Accumulations} = 12 + \log_2(1024) = 22 \text{ bit, and} \quad (3.4)$$

$$\text{Bit width for Accumulations of Squares} = 12 \times 2 + \log_2(1024) = 34 \text{ bit} \quad (3.5)$$

for each channel. Hence, with a minimum of 1024 events to process, the minimum number of FFs required to implement the computation and data storage of this IP in FPGA is:

$$(22 \text{ bit Accumulation} + 34 \text{ bit Squared Accumulation}) * 39 \text{ channels} = 2184 \quad (3.6)$$

for one HGCROC DAQ link.

This result determines that only one link can be processed in MARS in order to comply with the constraints of the IP. Section 3.5.2 describes an optimisation that enables the parallel processing of all links.

3.3 Initial Architecture

The initial IP architecture shown in Figure 3.4 has three main components:

1. the datapath, that computes and stores data as the events data are streamed,
2. a state machine that controls the datapath and decodes the data packets, and

3. an AXI Lite interface allows the software to access data and control registers.

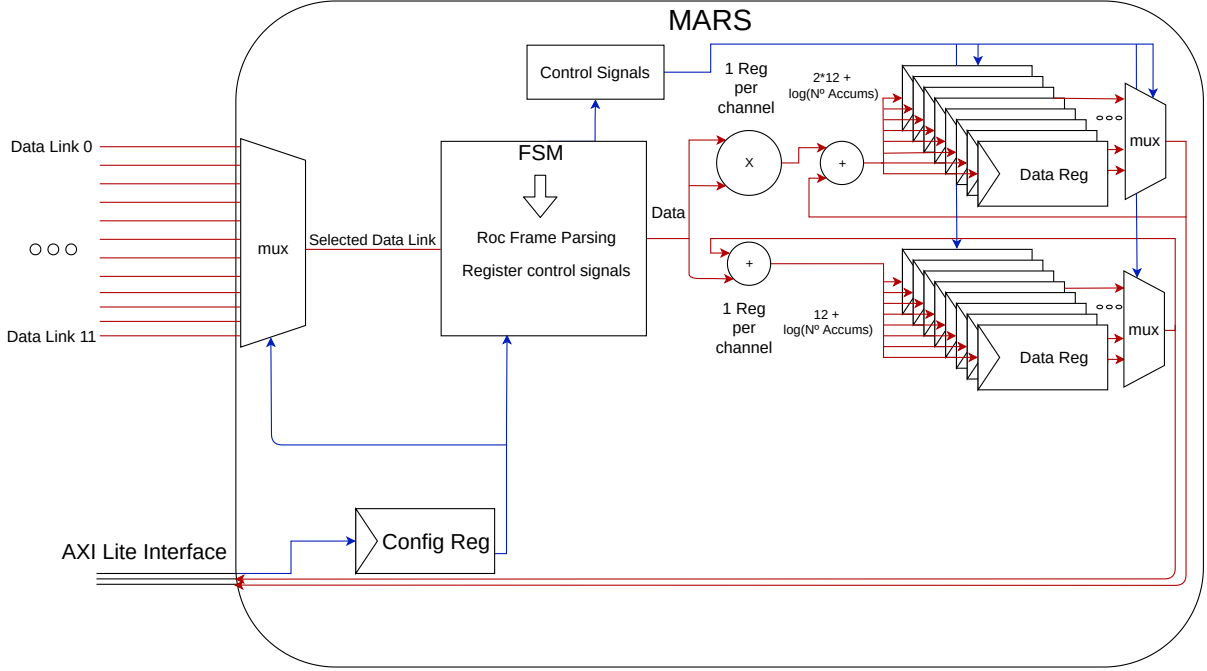


Figure 3.4: Simplified MARS architecture [7]. Control signals are highlighted in blue and data signals are in red.

The AXI Lite interface [23] translates the addresses coming from the PS to access data, and controls the reading and writing of the software-accessible registers. It includes several registers that control the IP (Config Reg in Figure 3.4) allowing to:

1. Reset the IP.
2. Query the Done bit.
3. Choose the operation mode: accumulate ADC, TOA, or TOT data.
4. Choose the input link to process.

The data accessible to the PS are stored in the datapath, since dedicated registers (“Data Reg”) to accommodate this information already exist there and thus no additional registers are required. Hence, the AXI Lite interface receives the data directly from the datapath and multiplexes the chosen register to the data output. Part of this address translation was moved to already existing multiplexers in the datapath, an optimisation described in Section 3.5.1, that allowed for a significant reduction of the overall LUT usage by over 40% [7]. However, sharing these multiplexers implies that the IP works in two mutually-exclusive modes: accumulation or readout. These modes can be checked by the PS by querying the Done bit. The readout mode is enabled when the IP has processed all events. After a reset, and until

the maximum number of events is processed, the IP is in accumulation mode and the data in the AXI Lite bus not corresponding to control register data is not valid.

Hence, following a reset, the Finite State Machine (FSM) parses the selected input link to find the HGCROC packet header. When it does, an accumulation cycle begins, synchronously with the arriving of data, at a frequency of 40 MHz, the operating frequency of MARS. This accumulation cycle is depicted in Figure 3.5. The state machine also selects the data to send to the datapath (ADC, TOA, or TOT) depending on what is specified by the user in the software-controlled register. Furthermore, in order not to overflow the sums, the state machine counts the number of events processed up to the maximum number of events to process specified via a pre-synthesis parameter, after which the IP stops accumulating values. When the maximum number of events is reached, the IP ignores the remaining input data, effectively stopping the accumulation of new values, asserts Done, and enters into the readout mode.

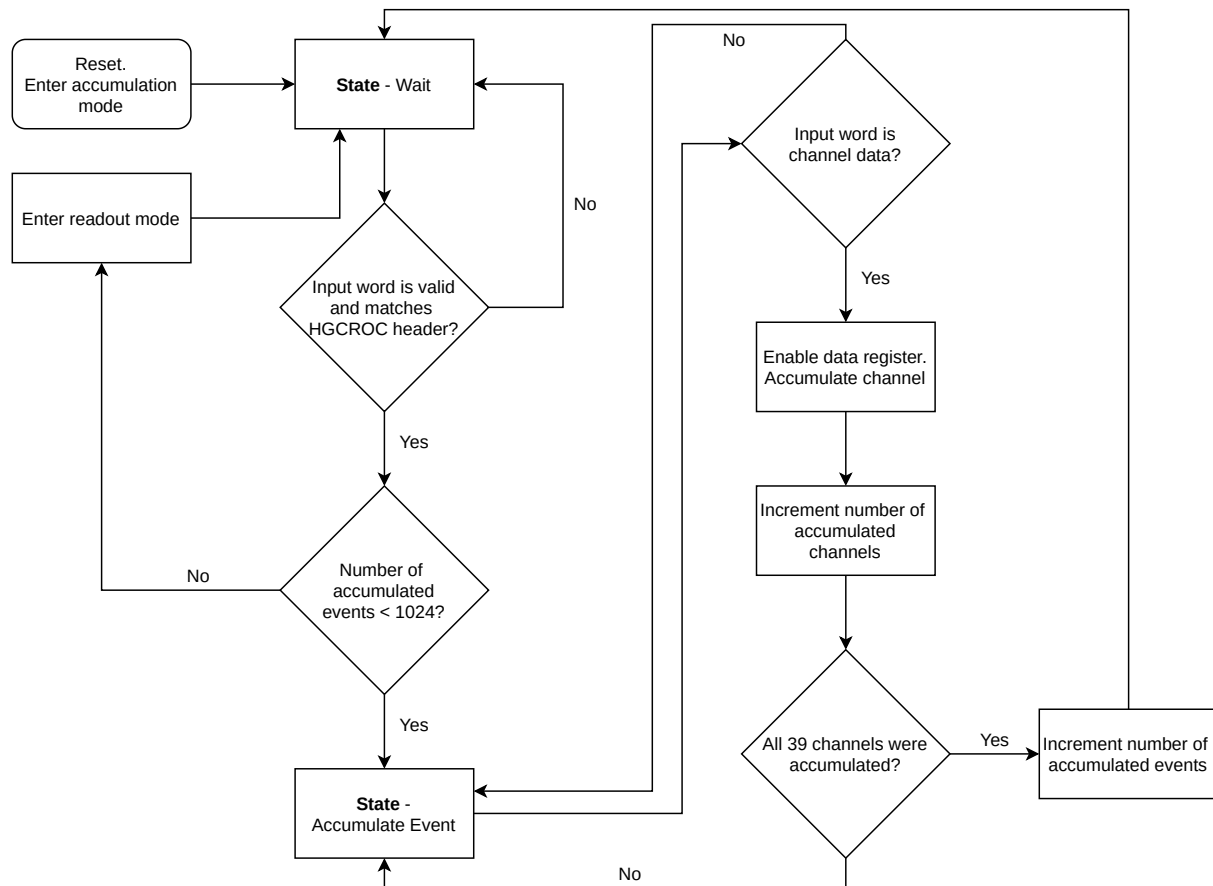


Figure 3.5: Simplified flowchart of the accumulation process done by the MARS FSM. 1024 events are defined as the default maximum number of events. However, this value is controllable at synthesis time through a pre-synthesis parameter.

The datapath comprises FFs, a multiplier, an adder, and multiplexers. This datapath performs the

arithmetic operations (sums and multiplication) needed to accumulate the received input values and their squares while being controlled by the FSM. The multiplexers, controlled by the FSM, select which data to accumulate with the data input, to save hardware resources. This datapath also outputs the stored data (after the computations are concluded) using the addresses received from the AXI Lite interface.

3.4 Pre-synthesis Configuration

Some MARS configurations are done via pre-synthesis parameters, namely the definition of the number of input links and the number of events to accumulate. To allow the possibility of reducing the statistical uncertainty, that scales with $1/\sqrt{N}$, the MARS allows to configure the maximum number of events to process.

Figure 3.6 shows the Vivado Graphical User Interface (GUI) that allows the user to change these parameters at instantiation time in block design mode.

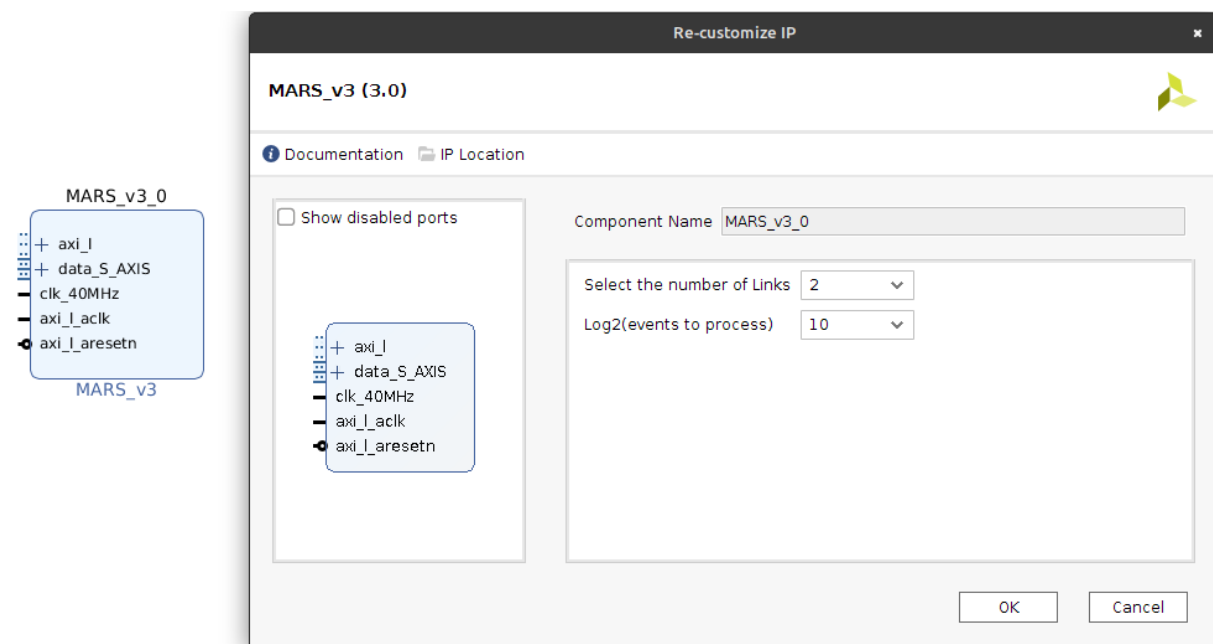


Figure 3.6: Vivado GUI showing the MARS IP instantiation in block design mode. The number of links and maximum number of events to process can be selected, making the IP block easier to adapt to the different systems it is used in. Own work [7].

The maximum number of events to accumulate is also controlled via a pre-synthesis parameter, that configures the width of the registers depending on the target number of accumulations that the IP should do. This parameter is the base two logarithm of the number of events to accumulate as it defines the width of registers. To not exceed the FPGA resources, the maximum number of accumulated events is

$2^{14} = 16384$. Following Equation (3.4) and Equation (3.5), this translates to

$$(26 \text{ bit Accumulation} + 38 \text{ bit Squared Accumulation}) * 39 \text{ channels} = 2496 \text{ bit.} \quad (3.7)$$

Hence, 2496 FFs are required to store the data. This resource usage of FFs is still acceptable as the total number of FFs is in the order of 2500.

3.5 Optimised Architecture

To optimise the resource usage of the architecture, two main optimisations were considered: one in the readout process and one in the data storage strategy. While the first exploited the working principle of the HGCROC testing, the latter exploited the FPGA target technology for MARS.

3.5.1 Read multiplexer optimisation

The HGCROC calibration procedures need the accumulated values corresponding to HGCROC data and their square. Hence, during the accumulation process, there is no need to read data from MARS as these data are not ready. Figure 3.7 shows the initial readout scheme implemented in MARS, where two similar multiplexers are used to perform the same function in different stages of the IP workflow.

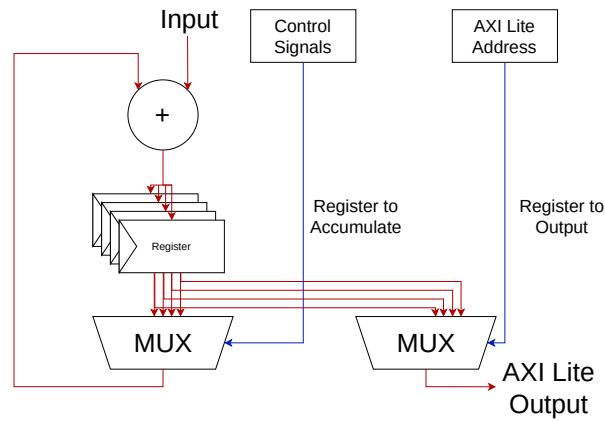


Figure 3.7: Initial readout scheme used in MARS. Two big multiplexers are used: one to select the registers to accumulate and another for the readout.

As the IP is not read during the accumulation and data is not changed during readout, only one multiplexer is being used at any given moment. Hence, the improved readout scheme described in Figure 3.8 was implemented, resulting in a decrease in LUT usage of over 40% [7].

This optimisation replaces one of the big multiplexers with a smaller one controlled by the Done bit. This was possible because a part of the AXI Lite address decoding could be done with the multiplexer

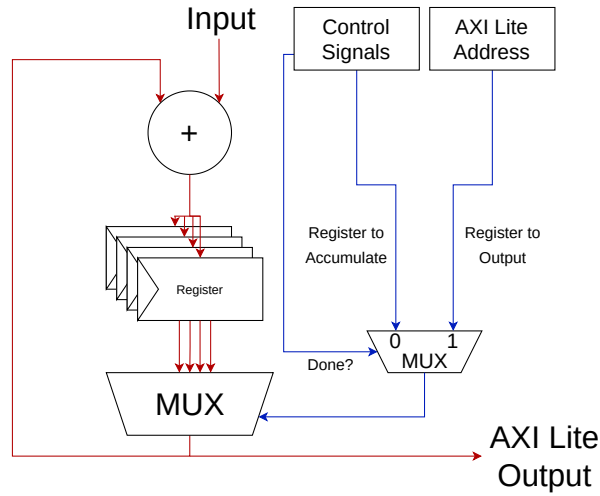


Figure 3.8: Optimised readout scheme used in the MARS. Only one large multiplexer is used for both the computation and readout phases. A smaller multiplexer selects the control signals to apply to the big multiplexer.

already in place for the datapath.

3.5.2 Technology Adaptation

Instead of using FPGA FFs to store the accumulation data (in registers), it is possible to use a shift register with the right depth, in this case 39, corresponding to the number of channels. The reason this can be done is that the order in which the channels enter into each accumulation step is always the same. By feeding the shift register with the new accumulated value, the channels' data would be accumulated in a round-robin fashion and the same results would be obtained.

The SRLC32E and SRL16E Xilinx primitives [8] allow to use LUTs not as combinatoric elements but as shift registers. Each such Shift Register LUT (SRL) is 1 bit wide and up to 16 or 32 bit deep, depending on the primitive. The internal architecture of the SRLC32E primitive is shown in Figure 3.9.

Accordingly, each LUT can implement a 1 bit 32-word-deep shift register or two 1 bit 16-word-deep shift registers. Hence, a 1 bit 39-word-deep shift register can be synthesised to match the number of HGCROC channels using only 1.5 LUTs. By using 56 of these shift registers in parallel, it is possible to synthesise both a 22 bit and a 34 bit 39-word-deep shift registers with only 84 LUTs. Such shift registers allow to store all data corresponding to 1024 accumulations of one HGCROC DAQ link without using 2184 FPGA FFs.

Figure 3.10 presents the MARS architecture whose data storage is now based on SRL primitives. As MARS is envisioned to be only implemented in Xilinx devices, the use of such primitives poses no compatibility issues with any HGCROC test system device. The resource usage of this optimised design is compared in Table 3.1 to the initial design described in Section 3.3. As it can be observed, for a single

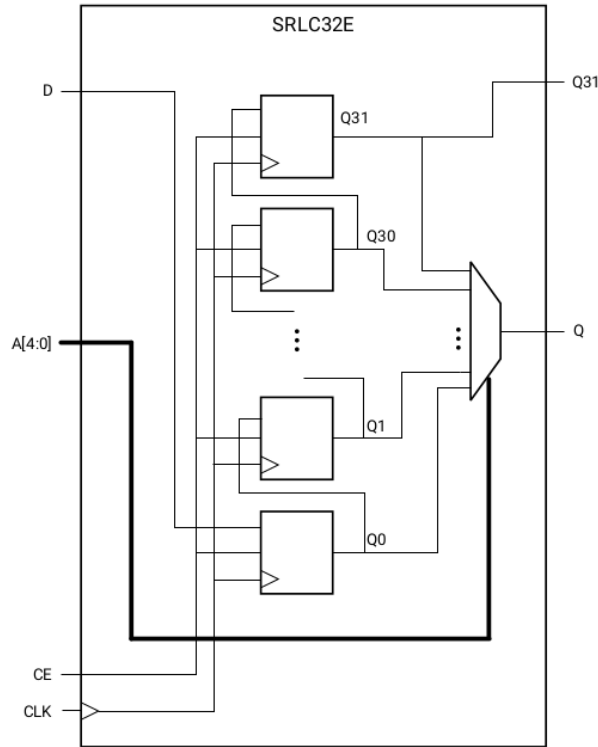


Figure 3.9: SRLC32E equivalent circuit. The shift register has a control signal, $A[4 : 0]$, that controls the depth of the shift register by multiplexing to the output, Q , the data of the FF corresponding to the value of A thus changing the depth of the shift register. The value of A can be changed at run time. This circuit is implemented using one single LUT. From [8].

input link the LUT usage decreases by 67%, while the FFs reduce 95% in the SRL design.

Table 3.1: MARS synthesis resource comparison as reported by Vivado 2019.2 for a 1024-event accumulation of 1 HGCROC DAQ link.

Data Storage Primitive	LUTs	Flip-Flops	Carry8	F7 Muxes	F8 Muxes	Digital Signal Processors (DSPs)
FF, Section 3.3	855	2273	3	123	42	1
SRL, Section 3.5.2	275	97	3	1	0	1

Due to this considerable decrease in resource usage, up to all the maximum 12 HGCROC DAQ links can now be accumulated in parallel in the SRL MARS design, while still complying with the resource usage constraint. This also makes the HGCROC calibration faster, as all the links can be calibrated in parallel. Thus, there is no need for the input multiplexer and the FSM and AXI interface are shared among all data paths.

Nevertheless, while there is no need to select the link to accumulate anymore, it is now needed to select which link will be parsed by the FSM to trigger the accumulation process, as some links might not have data during some tests. The software-accessible register that controlled which link would be accumulated was repurposed to select which link will be parsed. This poses no problem as data are streamed in all links synchronously.

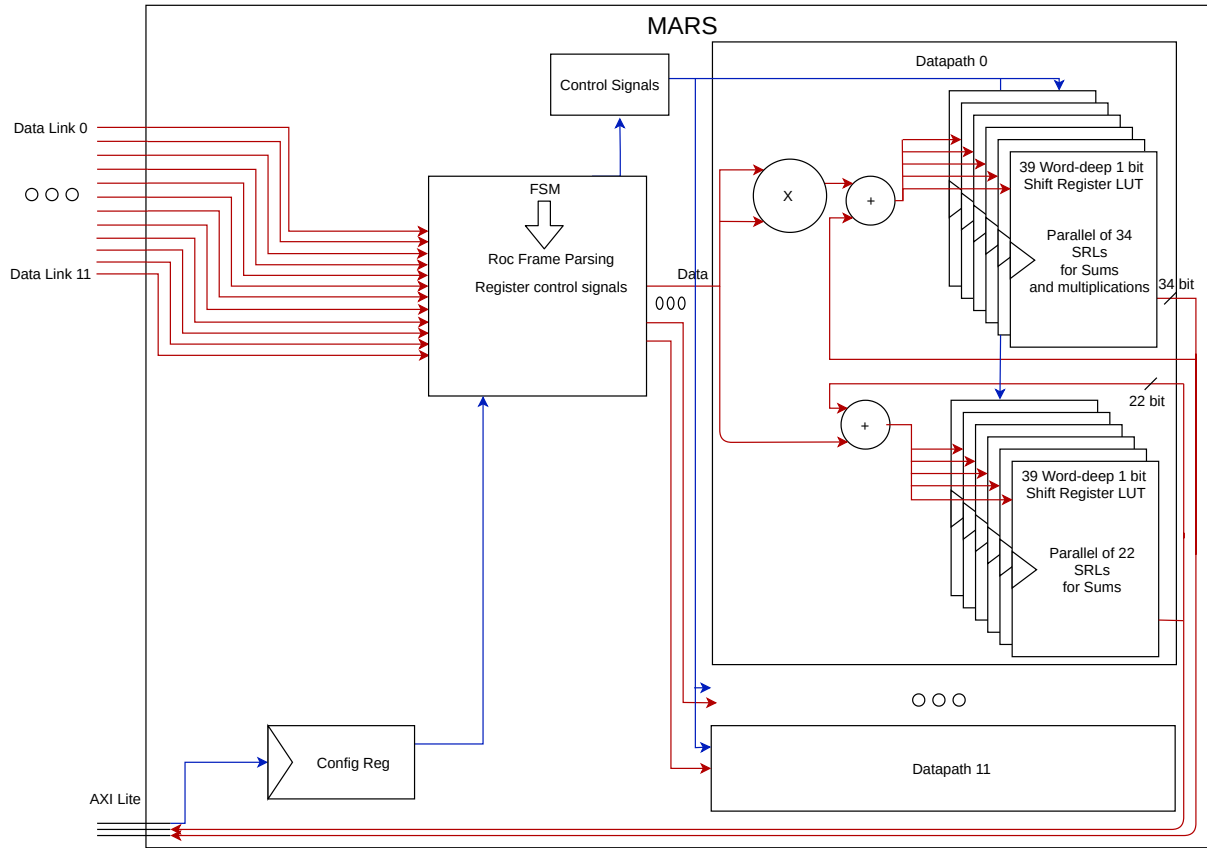


Figure 3.10: MARS architecture based on the SRL primitives. Data signals are represented in red whereas control signals are in blue.

Another advantage of this design is the multiplexer included in the SRL primitives. This multiplexer is used to control the depth of the shift register. During computation, this multiplexer must have static control bits to keep the shift register with fixed depth and not cause data corruption. However, in reading mode, this multiplexer can select the output from the several registers inside the shift register without the use of additional hardware, provided that the clock enable of the SRL is disabled, keeping the data static. This further reduces resource usage as the previously instantiated multiplexers are no longer needed.

It should be noted that only this design supports the “TDC efficiency” operation mode. Due to the request for this feature having come after the initial experience with the MARS, the initial architecture was no longer in use and this new feature was not back-ported to that design. Adding support for the “TDC efficiency” operation mode does not imply a significant change in the comparison of the results regarding resource usage, as the results in Table 3.2 show. The increase in LUT usage, due to the addition of the “TDC efficiency” mode is ~5% and the FF increases by ~4%.

Table 3.2: MARS synthesis resource usage as reported by Vivado 2019.2 for a 1024-event 2-link test system. Comparison of the new architecture between versions of the SRL design with and without “TDC efficiency” mode included in the IP.

“TDC efficiency” mode	LUTs	Flip-Flops	Carry8	F7 Muxes	F8 Muxes	DSPs
Included	505	107	6	3	0	2
Not included	481	103	6	0	0	2

3.6 Resource Usage and Performance Results

Table 3.3 summarises the Vivado 2019.2 synthesis resource utilisation report for shift-register design of MARS for up to 12 parallel input links with support for “TDC efficiency” mode. As shown, this IP fulfils the FPGA resource usage constraints of Section 2.3.1. The other constraints are also satisfied as the IP has the required input and output AXI interfaces.

Table 3.3: MARS synthesis resource usage as reported by Vivado 2019.2 for a 1024-event test system as a function of the number of parallel inputs links it can process.

Parallel inputs	LUTs	Flip-Flops	Carry8	F7 Muxes	F8 Muxes	DSPs
1	275	97	3	1	0	1
2	505	107	6	3	0	2
3	763	117	9	0	0	3
4	1000	127	12	0	0	4
5	1094	137	15	80	0	5
6	1443	147	18	67	0	6
7	1521	157	21	94	7	7
8	1711	167	24	110	1	8
9	1918	179	27	116	58	9
10	2146	188	30	114	20	10
11	2361	197	33	104	8	11
12	2523	207	36	174	51	12

The configuration capabilities of the developed IP allows its integration in different test systems with different needs and hardware resource scenarios. Besides covering all possible numbers of input links, if the hardware resources available at a certain system are not enough to process all HGCROC DAQ links in parallel, it is possible to instantiate a smaller MARS and serialise some of the processing chain, while still taking advantage of the hardware acceleration.

MARS is working as expected both in Vivado simulation and FPGA. The performed tests were successful and the IP is already integrated into HGCROC test systems [24], namely in the prototyping system of a LD hexaboard that has six HGCROC DAQ links. Each characterisation test takes several DAQ runs in order to scan the HGCROC parameters. Additionally, further software processing of the data is involved. Each of the DAQ runs consists in the following steps:

1. Configure the HGCROCs.
2. Start the readout process for 1024 events. For each event:
 - (a) Send a L1A fast command to trigger the data readout.

- (b) Read the event data from the Zynq to a remote computer.
- (c) Unpack the data on the remote computer. This step involves the accumulation of the data per HGCROC channel, the step that MARS accelerates.

With the addition of MARS to the test system, the equivalent procedure is slightly simpler:

1. Configure the HGCROCs.
2. Configure MARS mode (accumulation of ADC, TOA, or TOT; or “TDC efficiency” mode).
3. Reset MARS.
4. Trigger the readout process for 1024 events:
 - (a) Send 1024 L1A fast commands to trigger the data readout.
 - (b) Per half-HGCROC read out the 39 direct sums and 39 sums of squares from MARS to the software.

After each DAQ run, the resulting data are directly used by the software and further post-processing is not needed as the computations were performed in the MARS.

When considering a single DAQ run, MARS already accelerates the readout process since each DAQ run with MARS takes 2.24 s, a speed-up of about 1.65 with respect to the acquisition process without MARS that takes 3.70 s. This acceleration comes from the removal of the data transmission step as the data is already accumulated per HGCROC channel. Not only the amount of data transferred is reduced, but also the processing involved in the unpacking step is no longer necessary to be done in software.

The acceleration provided by MARS becomes more evident when evaluating a full set of characterisation measurements, comprising several DAQ runs. Such a test takes 250 s to run using a remote computer with 4 cores and without using MARS. The addition of MARS to the system reduces the test time to 143 s already providing a speed-up of about 1.75. An important difference in this test is that using MARS only uses one remote computer core, as the code running in parallel on the 4 cores was the unpacking step for several events and several runs. With MARS, there is no need to unpack data as they are read already accumulated and ready for subsequent software processing.

Although this speed-up already presents a good result from the inclusion of MARS, due to the large amount of HGCROCs to test, it is not possible to allocate 4 cores per HGCROC or hexaboard during production testing. Hence, a fair comparison to evaluate the speed-up that the MARS can provide to the HGCROC test systems is to compare the performance of MARS with the performance of the normal procedure constrained to use a single Central Processing Unit (CPU) core in the remote computer. The same set of characterisation measurements was executed in the single-core configurations yielding the larger execution time of 595 s. This implies a speed-up of about 4.15 when using the MARS.

These results demonstrate that not only the MARS reduces the execution time of the HGCROC characterisation tests but also makes the test systems more scalable for production testing. Hence, with the integration of this IP in the HGCROC test systems, it is possible to conduct more detailed tests of the ASICs in the same amount of time.

3.7 Summary

This chapter described the design of the MARS, the optimisations made to its architecture, and its positive impact on the characterisation and production testing of the HGCROC ASICs, the first contribution of this work to HGCAL. Chapter 4 will report on the second contribution, the development of a prototyping system for the slow-control block.

4

Slow-Control

Contents

4.1	Slow-Control Architecture	43
4.2	Slow-Control Testing and Validation	49
4.3	Performance Evaluation	53
4.4	Hardware Resource Usage	56
4.5	Slow-Control Portability Evaluation	58
4.6	Summary	59

As part of the effort to build a vertical HGCal prototype, it was decided to individually prototype the slow-control block and validate its functionality. The slow-control block, that will be instantiated in each BE FPGA of HGCal, is responsible for configuring all FE electronics, as described in Section 2.2.1, with each FPGA configuring up to 3500 ASICs. To validate its functionality it is needed to establish a reliable communication link with two types of ASICs: the IpGBTs and GBT-SCAs. Such connections must allow the slow-control to perform read and write operations to the internal registers of those ASICs (slow-control transactions). Furthermore, the software-facing AXI interfaces of the slow-control also need to be validated.

It is important to note that the slow-control architecture had already started being developed before the beginning of this work [19]. However, those early ideas had not been fully developed and had not been tested in order to validate the functionality. Hence, this chapter describes a contribution to HGCal consisting of the development of a prototyping system to test and validate this block, as well as implement some of its functionality.

The results obtained with this prototyping system were published in an article in the 33rd International Workshop on Rapid System Prototyping (RSP) in October of 2022 [4].

4.1 Slow-Control Architecture

The slow-control architecture, shown in Figure 4.1, is based on a significant amount of small cores, each associated with an ASIC type to be directly interfaced in the FE, namely the IpGBT and the GBT-SCA. Each such core receives data from the software with slow control command transactions. After a decoding step, the data are conveyed to the targeted ASIC in the FE and that ASIC's reply is stored for the software to later read out. Each core can execute independently such that some parallelism can be exploited in the configuration of the HGCal system.

Due to FPGA resource limitations, it is not possible to implement a fully parallel structure and instantiate one core for every FE ASIC [19]. Hence, each core needs to multiplex its output to drive more ASICs. This multiplexing factor can be adjusted allowing the flexibility to balance the amount of parallelism against FPGA resource usage. Moreover, some software constraints also need to be taken into consideration, namely the availability of independent software threads to service these cores and take advantage of the hardware parallelism.

The slow-control block is accessible to the controlling software through two AXI interfaces connected to the memory and control module. This component of the slow-control hosts data buffers and software-accessible registers, for control and status signals. Each core is associated with an independent set of buffers and registers, rendering each core independent of other cores.

The data buffers comprise 8 BRAMs configured in true dual-port mode, allowing to store 2048 128 bit

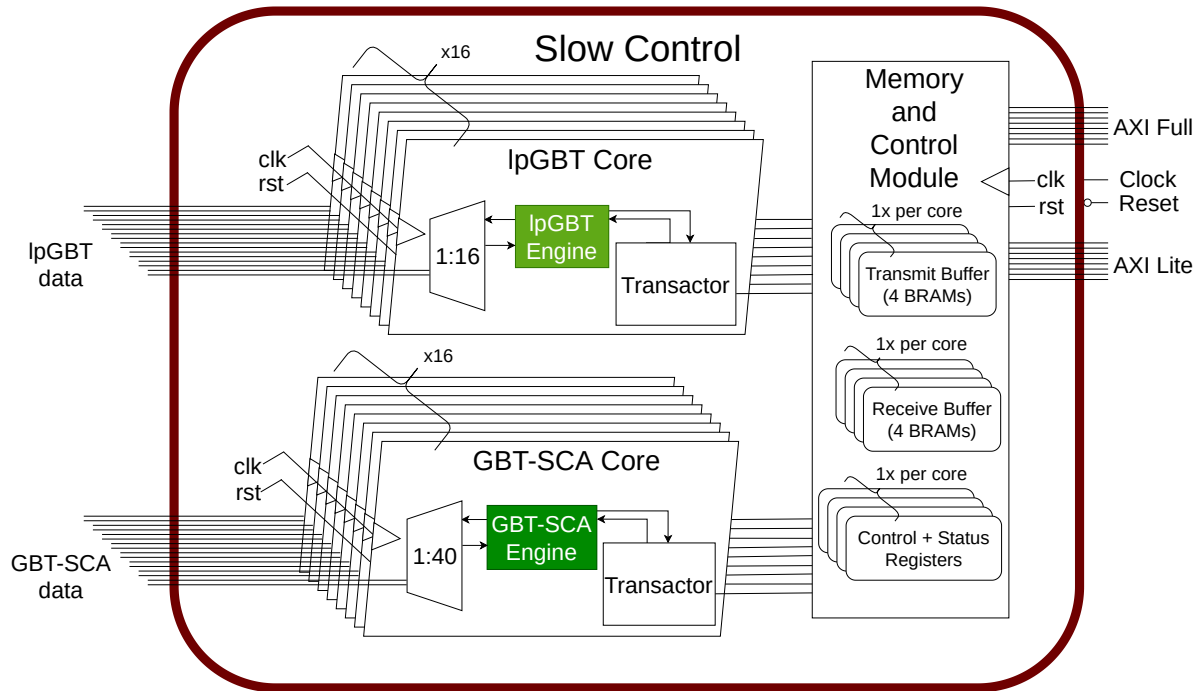


Figure 4.1: Slow-Control architecture showing the main architectural aspect that regards the fact that the IpGBT and GBT-SCA ASICs used in the FE have different protocols. One can also see the software-facing AXI interface. The multiplicity of cores reflects the expected number in a BE DAQ FPGA of a BE DAQ ATCA board. Own work [4].

words per core. Half of those words are dedicated to data to be sent to the FE (transmit buffer) and the other half is dedicated to holding the respective replies meant to be read by the software (receive buffer). These buffers are connected to the AXI Full interface of the slow-control with one of the BRAM ports. The other port is connected to the transactor of each core.

A set of status and control registers per core are accessible via the AXI Lite interface and allow the software to control the data flow on each core, i.e., how many transactions are to be processed and when to start their processing. Each core processes transactions in its send buffer sequentially, once the start bit on the respective control register is set. A transaction must be fully processed, including receiving its reply, before the next transaction is executed.

The structure of the slow-control cores is shared between the IpGBT and GBT-SCA cores. Each is composed of three components: a transactor, an engine, and a multiplexing circuit (cross-point), see Figure 4.1. The transactor is a state machine that interfaces the BRAM and controls the data flow inside the respective slow-control core. When sending transactions to the FE, the transactor reads data from the buffers, decodes it, and sends the output to the engine in the core.

The engines (IpGBT or GBT-SCA) are provided by the CERN IpGBT group and translate the data provided by the transactor into the internal IpGBT protocol or the HDLC protocol [18] used by the GBT-SCA. As the slow-control is clocked at 40 MHz, the output of both engines is a 2 bit signal encoding an 80 Mbit/s

signal.

Engine data is then fed to the cross-point multiplexing circuit. This circuit can either multiplex or broadcast the transaction to the FE. The working principle of this circuit involves a masking vector that allows to send (broadcast) the transaction to multiple ASICs in the FE. The multiplexing factor of each slow-control core is defined in the cross-point as a pre-synthesis parameter. By default, 16 lpGBTs can be driven from each lpGBT core and 40 GBT-SCAs can be driven from each GBT-SCA core. These parameters were chosen in order to have 16 instances of each type of core in the HGCAL BE DAQ FPGA design. These parameters are still subject to change as the result of fine-tuning to be done in the final BE design. Finally, each 80 Mbit/s signal output from the cross-point corresponds to a slow-control channel and needs to be connected to a lpGBT link block. These link blocks are instantiated inside the FPGA and are responsible for encoding full lpGBT frames, i.e., including other data that are conveyed via the lpGBT, such as fast commands.

The data flow that was described above corresponds to a full sending procedure. The reverse flow takes place when each slow-control core receives and processes a reply from the FE. The reply from the selected FE channel is routed by the cross-point back to the engine. The decoded data is output by the engine and read by the transactor that stores it in the receiver buffer. These replies can then be accessed by the software via the AXI bus.

4.1.1 Software Interface

The different number of ASICs that need to be serviced and, as a consequence, the different protocols to consider imply a different interface from the controlling software for words to be written in the transmit buffers of the slow-control.

The slow-control lpGBT cores use the Transmit (TX) word frame described in Table 4.1. The bits contained in this word convey the data to transmit to the lpGBT as well as the control bits necessary to configure the transactor and the cross-point.

Table 4.1: lpGBT TX frame format of the transactor developed in this work.

Bit Mask	Description
127:112	Broadcast Masking Vector.
111:100	Reserved.
99:96	Reply Address of the lpGBT.
95:88	Chip address of the lpGBT[7:1] + operation type R/W[0].
87:72	Address of the 1st lpGBT register to read/write.
71:68	Reserved.
67:64	Number of bytes/registers to read/write. Up to 8.
63:0	Bytes to write. Up to 8.

The broadcast masking vector and the reply address of the lpGBT fields configure the cross-point.

While the first specifies to which IpGBTs the data is sent, the latter selects which channel's reply is stored since only one IpGBT reply can be stored.

Unlike the HDLC protocol used in the GBT-SCA, the IpGBT protocol allows for the writing of consecutive registers in one transaction. To exploit this possibility, the slow-control allows to write up to 8 consecutive registers in a single transaction. In these cases, only the first register to be written is specified in the TX frame. The number of bytes to send to the IpGBT also needs to be specified. This number will correspond to the number of registers to write since the registers have a width of one byte in the IpGBT. The byte to be written in the first register should be inserted in the least significant position of the data field (bytes to write).

Upon the reading of the reply buffers of the slow-control IpGBT cores, the controlling software will be provided with data formatted according to the Receive (RX) frame described in Table 4.2. The IpGBT returns the reply data, a parity word, and a command word. The last bit of the command word validates the parity word when set to one. These data can be used by the software to determine the success or failure of each transaction individually. Similarly to the data field of the TX frame, the data corresponding to the first register is expected in the least significant position of the data field (written/read bytes).

Table 4.2: IpGBT RX frame format of the transactor developed in this work.

Bit Mask	Description
127:125	Error Flags.
124:120	Reserved.
119:112	Chip address of the IpGBT[7:1] + operation type R/W[0].
111:104	IpGBT Command. Last bit parity ok.
103:92	Reserved.
91:88	Number of bytes/registers to read/write. Up to 8.
87:72	Address of the 1st IpGBT register to read/write.
71:64	Parity Word from the IpGBT.
63:0	Written/Read bytes. Up to 8.

On the other hand, the GBT-SCA cores expect the TX word frame format described in Table 4.3. The broadcast masking vector and reply address fields perform the same functionality as their counterparts in the IpGBT cores.

The transaction ID, SCA channel address, and command fields correspond to internal GBT-SCA configuration words, following the GBT-SCA communication protocol specification [16]. The Command ID field allows the transactor to configure the engine to send three different types of commands: reset, connect, and start, according to the encoding of Table 4.4. The start command corresponds to a normal command, while the connect and reset commands are special command words that perform fixed functions requiring no additional data.

Upon the reading of the reply buffers of the slow-control GBT-SCA cores, the controlling software will find data in the RX frame format described in Table 4.5. Unlike the IpGBT RX frame, only the transaction

Table 4.3: GBT-SCA TX frame format of the transactor developed in this work.

Bit Mask	Description
127:88	Broadcast Masking Vector.
87:74	Reserved.
73:68	Reply Address of the GBT-SCA.
67	Reserved.
66:64	Command ID.
63:56	GBT-SCA Chip address.
55:48	Transaction ID.
47:40	GBT-SCA channel address.
39:32	Command.
31:0	Data.

Table 4.4: Encoding of the Command ID field of the GBT-SCA TX frame.

Bit Code	Description
000	Reserved.
001	Send Connect CMD to GBT-SCA.
010	Send Reset CMD to GBT-SCA.
011	Reserved.
100	Send Start CMD to GBT-SCA.
101	Reserved.
110	Reserved.
111	Reserved.

ID and address fields correspond to data previously sent to the GBT-SCA.

Table 4.5: GBT-SCA RX frame format of the transactor developed in this work.

Bit Mask	Description
127:83	Reserved.
82:80	Error Flags.
79:72	GBT-SCA address.
71:74	Control.
63:56	Transaction ID.
55:48	GBT-SCA channel address.
47:40	Number of bytes in Data.
39:32	Error.
31:0	32 bit data word.

4.1.2 Features added to the slow-control block

The slow-control block architecture that existed at the beginning of this thesis [19] was not ready to be tested without relocation of its control registers, as they were located in a dedicated control register block containing all control registers on the BE DAQ FPGA design. Only after the development of an AXI Lite interface that contained only the control and status registers that concern the slow-control block, it was possible to package the block as a Vivado IP ready to be tested.

Apart from the requisite validation, the envisaged testing of the slow-control block aims to provide to the other HGCAL prototyping systems with the capability of using the final slow-control infrastructure as a Vivado IP. This specific packaging of the FPGA hardware is required as these prototyping systems use IPs in their top-level design files in the Vivado block design environment.

Additionally, as some of these coexisting systems use unrelated clocks to drive different IP blocks in the same FPGA design, it was necessary to ensure that memory elements (BRAMs and registers) of the slow-control design that interfaced the AXI clocks would be compatible with Clock Domain Crossing (CDC). Xilinx CDC primitives were used for the control registers and no changes were needed for the BRAMs as they are dual clock devices when configured in dual port mode.

The controlling software interface described in Section 4.1.1 was also modified to provide error flags, not only for debugging purposes but also to prevent the hardware from getting blocked. As a result, the transactors had to be improved to add the error flags listed in Table 4.2 and Table 4.5. The error flags for the IpGBT, described in Table 4.6 and for the GBT-SCA presented in Table 4.7, provide several data regarding the reception of replies from the FE and about the occurrence of incorrect data in the transactions to send to the FE.

Table 4.6: IpGBT transactor error flags.

Bit	Description
127	Timeout. When active, no reply was received.
126	When active, a write transaction was specified with no data to send.
125	When active, error in broadcast masking vector. No target ASIC was selected.

Table 4.7: GBT-SCA transactor error flags.

Bit	Description
82	Timeout. When active, no reply was received.
81	When active, signals an invalid Command ID field.
80	When active, error in broadcast masking vector. No target ASIC was selected.

The timeout field of the error flags prevents the blocking of the transactors, and therefore the slow-control cores, in addition to its use in debugging. After sending a transaction to the FE, the transactors wait for 1024 clock cycles, after which the timeout flag is asserted and the next start transaction starts to be processed. Although the 1024 clock cycles are hardcoded, this amount of wait time should be fine-tuned upon the integration of the slow-control in the final detector system. The other flags serve debugging purposes and may be removed in a final version of the slow-control though they can catch software mistakes.

In addition to the error flags providing information about each transaction, the status registers were also modified to provide information about a set of transactions as a whole. Table 4.8 describes the bit fields of the introduced status registers associated with each slow-control core. Only the busy bit had

been originally implemented in the provided architecture at the beginning of this work.

Table 4.8: Bit fields of the slow-control status registers.

Bit	Description
12:3	Number of successful transactions.
2	zero_cmd. Indicates if upon start the number of transactions set on the control register was zero.
1	Busy. If asserted, the BRAM contents must not be changed.
0	Negated Timeout. If 0, some transaction failed due to timeout.

As the timeout bit refers to all the issued transactions on the buffers, it is possible to quickly confirm if all transactions were successful, i.e., none timed out. Furthermore, if some (or all) transactions fail, the field corresponding to the number of successful transactions allows determining how many. These data provide quick coarse information to the controller software as it does not need to read and process all the data in the reply buffers. The zero command field is a debug value that indicates a bad configuration of the control registers, namely the trigger of the slow-control core with zero transactions envisaged to be issued to the FE.

4.1.3 Further improvements and optimisations

During the process of debugging the system in order to achieve a reliable connection with the FE that is compliant with the lpGBT and HDLC protocols, some components of the slow-control were also changed, namely both transactors as well as an internal First In First Out (FIFO) buffer of the lpGBT engine.

Besides debugging and optimisations, the modifications made to the lpGBT transactor also include the introduction of a pre-synthesis parameter that allows to specify the version of the lpGBT chip being interfaced. Since the lpGBT has two versions (called 0 and 1) with slight changes in the protocol, it is very important to distinguish between the two. Using one version may invalidate transactions when using the other chip version. The pre-synthesis parameter allows the system designer to choose between two similar state machines. The two different versions of the lpGBT are not expected to be used in the same system and HGICAL will only use lpGBTs of version 1. However, as some test systems still use the older version of the lpGBT, the slow-control must support both ASIC versions.

4.2 Slow-Control Testing and Validation

This section describes the prototyping system that allowed the testing, validation, and debugging of the slow-control block.

This prototyping system aims to validate the AXI interfaces of the slow-control and its connection to the lpGBT and GBT-SCA ASICs. Since the final hardware for the detector system is still under development, this test system emulates the BE and FE of HGICAL at the system level.

Furthermore, the current unavailability of the ATCA board hosting the VU13P FPGAs and Zynq MPSoC motivated the choice of the Xilinx ZCU102 development board [25] to emulate the HGCAL BE. The Enhanced Small Form-factor Pluggable (SFP+) connectors and Multi-Gigabit Transceivers (MGTs) of this board allow the connection of the optical links to the FE using the required data rates. Moreover, the Zynq hosted in this platform shares the Ultrascale+ FPGA technology of the VU13P and allows the use of AXI interfaces as in the final system.

The FE is also emulated using CERN development boards specifically designed to prototype those systems that need to interface with IpGBTs and GBT-SCAs. These boards, denoted as the Versatile Link Plus Demo Board (VLDB+) [26] and the Versatile Link Demo Board (VLDB) [27], contain either a IpGBT or a GBT-SCA, allowing the prototyping system to use the ASICs that will be used in the final HGCAL system.

4.2.1 Prototyping System

Figure 4.2 presents a block diagram of the developed prototyping system and Figure 4.3 shows the hardware on the bench. The PS of the Zynq FPGA runs a CentOS 7 Linux distribution, the same that is used in other HGCAL prototyping systems.

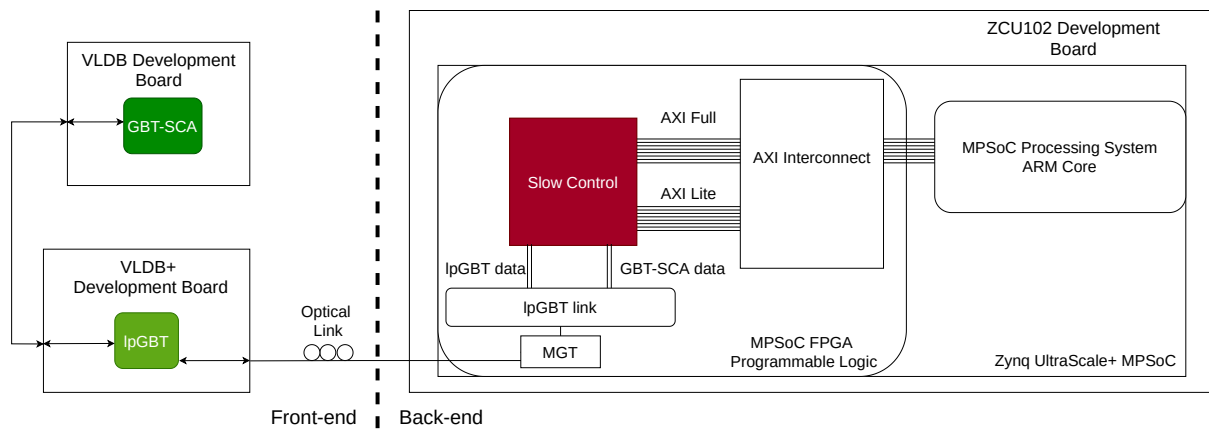


Figure 4.2: Block diagram of the developed prototyping system to test and validate the slow-control block. The corresponding physical setup is depicted in Figure 4.3. Own work [4].

The slow-control block is instantiated in the PL part of the Zynq FPGA considering only a single core of each type since the system will only have two ASICs to test: one IpGBT and one GBT-SCA. Through the use of a Xilinx AXI interconnect IP, the slow-control is connected to the PS of the Zynq MPSoC.

To establish the connection with the FE, two slow-control channels are connected to the IpGBT link hardware block using the IC part of the IpGBT protocol to convey data to the IpGBT and the EC part to convey data to the GBT-SCA. The IpGBT link is connected to a MGT, which in turn is connected to a SFP+ opto-electrical transceiver plugged into the ZCU102 board.

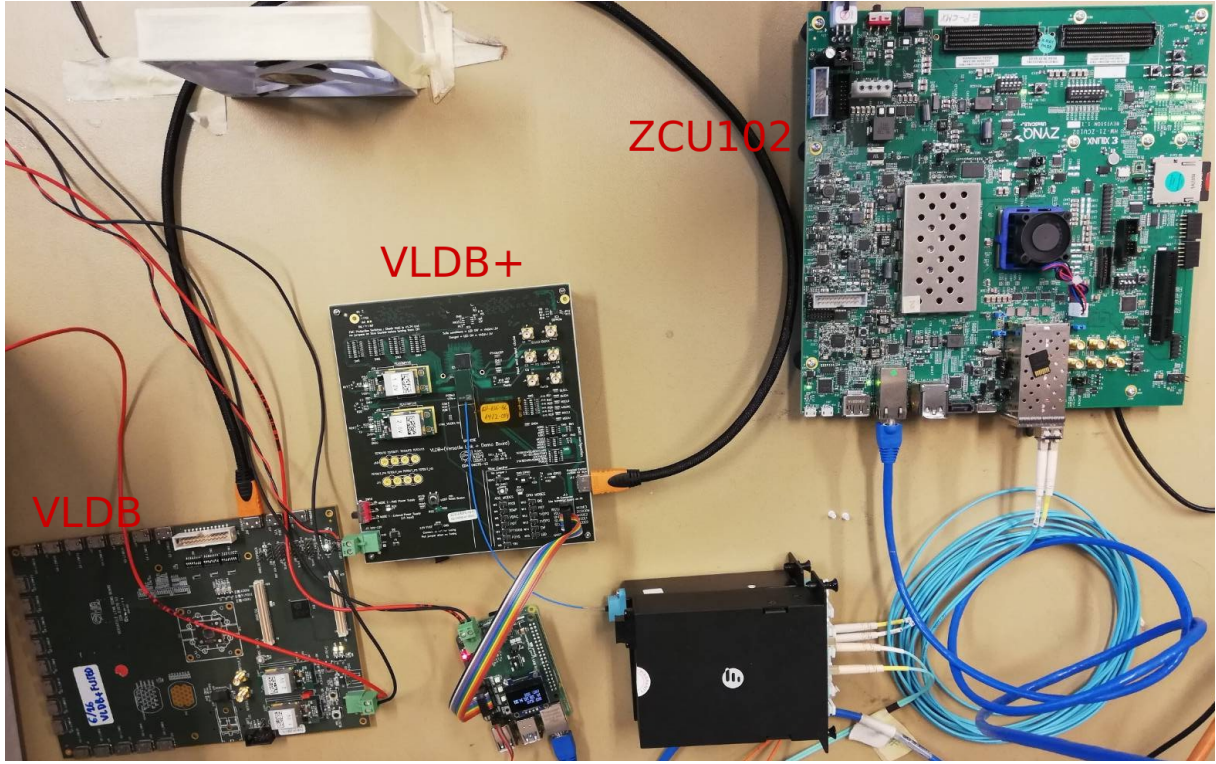


Figure 4.3: Photograph of the hardware used for the prototyping system developed to test and validate the slow-control block. The ZCU102 board acts as the HGCAL DAQ BE and is connected via optical fibres to the VLDB+. The VLDB+ hosts a IpGBT ASIC and is further connected to a VLDB board hosting a GBT-SCA ASIC. This set of boards allows to test all the slow-control interfaces of the HGCAL FE.

Hence, two optical fibres are connected to the SFP+ connector, one for the uplink connection and the other for the downlink connection. These fibres are connected to a VLDB+ development board containing a VTRX+ and one IpGBT. To complete the prototyping system, the VLDB+ is connected to a VLDB board containing a GBT-SCA assigned to the EC channel of the IpGBT. Although the field of the IpGBT protocol assigned to the GBT-SCAs is not the EC field in the final system, this choice for prototyping purposes is acceptable as the data remains transparent to the IpGBT, hence not affecting the validation of the HDLC communication.

Three clock sources are involved in this system. The data rate of 10.24 Gbit/s, imposed by the IpGBT, requires the MGT to be clocked at 320 MHz. This clock is driven from the MGT clocks of the ZCU102 board and also provides the clock to the IpGBT link. The slow-control logic is clocked at 40 MHz, with a clock driven from this source through the use of a Mixed-Mode Clock Manager (MMCM) IP to reduce its frequency. The Xilinx IP used to configure the MGT also requires another clock, corresponding to the system clock. This clock is driven from the free-running clock of the ZCU102 board which has a frequency of 125 MHz. The last clock source comes from the PS and is used to clock the AXI interfaces of the slow-control at 40 MHz.

Vivado 2020.2 was the original software toolchain used to develop this prototyping system and the slow-control architecture. However, as other HGCAL prototyping systems use Vivado 2019.2 and the slow-control is to be used as an IP in those systems, the slow-control prototyping system was ported to the Vivado 2019.2 version to avoid compatibility issues.

4.2.2 Validation Testing Criteria

The AXI interfaces in the slow-control block can be validated through the writing of data in the local memory followed by the read-back of the same data from those addresses. Therefore, the validation procedure consists of an exhaustive test that writes all available addresses in the address space of both AXI interfaces, followed by the read-back of those addresses and comparison of the values written and read back.

The connections to the IpGBT and GBT-SCA are validated by the successful writing and reading of data to the internal registers of these ASICs. These transactions allow to change the configuration of the IpGBT and GBT-SCA (e.g., their operation mode, enabling I2C masters, or control GPIOs). Hence, the validation procedure involves the writing and reading of data to these registers and subsequent validation. This also involves the analysis of the RX data frames received from the FE and validation of the functionality of all fields.

4.2.3 Validation Results

The validation tests were executed in two separate environments: first in a bare-metal configuration (without an operating system) and then from within Linux. The use of Linux as a second step of the testing process allowed to initially debug the system without the layer of abstraction provided by the operating system, simplifying the debug process from an electronics engineering point of view. The bare-metal tests were performed using a C application, while the tests in Linux used a python script.

The AXI interfaces were validated with a procedure that wrote data in all available addresses, read back the same addresses, and compared the written and read-back data, to ensure their correct operation.

Afterwards, to validate the connection to the IpGBT, several internal registers of the ASIC were changed and read back to verify the new correct values. As some of these changes concerned specific configuration registers, it was possible to change the internal state of the IpGBT and to enable the EC channel. As the state of the IpGBT changed to “ready”, it was possible to verify the success of this change via a LED state change on the VLDB+ board.

The connection with both versions of the IpGBT was validated through the use of two different VLDB+ boards, each with a different version of the ASIC. However, only one of the boards was used at any given

time.

The expected behaviour of the EC channel of the lpGBT was checked through the validation of the GBT-SCA transactor. As with the lpGBT, the communication with the GBT-SCA was validated through writing and reading back its internal registers. Moreover, the LEDs in the VLDB board allowed to validate the control of the GBT-SCA's GPIOs.

The data read from the slow-control buffers, not only allowed to validate the correct communication with the FE ASICs, but also allowed to validate the correct implementation and functionality of the slow-control software in interpreting the data frames.

4.3 Performance Evaluation

Following the validation tests, and after a reliable connection was achieved between the slow-control and the ASICs, several stress tests were conducted under the Linux environment, to gauge and quantify the performance of the slow-control cores.

To estimate the transaction throughput of the slow-control cores, the slow-control send buffers were loaded to their full capacity (1024 transactions) after which the slow-control issued the corresponding data to the FE. This procedure was executed with a C++ script to increase software performance and repeated 10 000 times comprising $\sim 10^7$ transactions. All the conducted tests executed successfully.

4.3.1 Transaction Throughput

The fraction of the stress test corresponding to the slow-control was completed by the lpGBT core in 28.58 s, yielding a throughput of $\sim 358\,000$ transactions per second, one transaction executed on average every $2.79\,\mu\text{s}$. The GBT-SCA core was able to complete the same amount of transactions in 44.08 s. The throughput obtained for the GBT-SCA core is $\sim 232\,000$ transactions per second with each transaction taking an average of $4.30\,\mu\text{s}$ to execute.

4.3.2 HGCal Configuration Time

The obtained results allow to roughly estimate the configuration time for all of HGCal's 150 000 ASICs. As the ~ 100 FPGAs in the BE will be configuring parts of the detector in parallel only the worst case scenario of the slowest FPGA needs to be considered.

For now, this estimate will be approximate, as there are no tests with the final system hardware or software. As an example, some software overheads are not accounted for (such as the access to the configuration database) and some hardware testing is not yet done, so the latency of I2C transactions cannot be exactly determined. However, the estimated configuration time that was obtained already

allows to evaluate the performance of this slow-control architecture and to determine if such performance can meet the requirement of a configuration time in the order of a few minutes.

Some known overheads of the final system can be taken into account when making this estimation including the optical fibre latency, an estimation for the I2C transaction time [28], and some of the software overheads.

In accordance, to determine the overall HGCAL configuration time, three values are needed:

1. How many slow-control transactions need to be issued.
2. How much time it takes to issue a slow-control transaction.
3. How much time should be accounted for the I2C overheads.

To determine how many slow-control transactions are needed, the worst-case scenario must be considered. That is determined by the BE FPGA that needs to issue the largest quantity of slow-control transactions to the FE. The architecture of the FE configured by this FPGA is considered to be such that in each of the 108 fibre connections there are six low-density hexaboards connected to the engine [28], see Figure 2.3. Note that in this architecture, no GBT-SCA is used. This scenario comprises the following amounts of ASICs to be configured per FPGA:

$$3 \text{ HGCROCs / hexaboard} \times 6 \text{ hexaboards} \times 108 \text{ optical fibres} = 1944 \text{ HGCROCs,} \quad (4.1)$$

$$(1 \text{ ECON-D} + 1 \text{ ECON-T}) / \text{hexaboard} \times 6 \text{ hexaboards} \times 108 \text{ optical fibres} = 1296 \text{ ECONs, and} \quad (4.2)$$

$$3 \text{ lpGBTs / engine} \times 108 \text{ optical fibres} = 324 \text{ lpGBTs,} \quad (4.3)$$

with each FPGA configuring a total of ~ 3500 ASICs.

Additionally, it is also assumed the worst-case scenario of having every register of every ASIC written once. Furthermore, as the order of writing cannot be guaranteed to be sequential, due to the specifications of each ASIC, the gains that could be had from writing up to 8 registers in one lpGBT transaction will not be considered. Hence, the worst-case number of slow-control transactions needed per ASIC equals the number of internal registers:

- Each HGCROC would require 1210 transactions.
- Each ECON-T would require 2407 transactions.
- Each ECON-D¹ would require 2407 transactions.
- Each lpGBT would require 493 transactions.

¹The ECON-D design is not finished and for the purpose of this estimation it is considered that it would require the same number of transactions as the ECON-T, 2407.

However, to determine how much time is needed to issue a full slow-control transaction, the results of Section 4.3.1 are not sufficient. In addition to these data, the time it takes the software to write and read through the AXI interfaces needs to be considered, as well as the optical fibre latency. The stress test performed on the slow-control block measured how long it took to write and read the AXI interfaces. The complete writing took 2.67 s and the reading 22.50 s for both the tests of the lpGBT and GBT-SCA cores. Adding this overhead to the previously obtained time of 28.58 s of the lpGBT core, the procedure to send $\sim 10^7$ lpGBT transactions to the FE is computed as taking 53.86 s. With this new value, it is possible to determine that a slow-control transaction takes about 5.26 μ s to be issued.

When installed in the CMS cavern, these transactions must travel about 200 m of optical fibre, 100 m for sending and 100 m for the replies. The transmission speed of the fibres is $\sim 70\%$ of the speed of light, $c = 299\,792\,458$ m/s.

However, to be in a worst-case scenario, only 50% of c will be considered and the fibre latency to be added to the time to issue a slow-control transaction can be obtained as

$$\text{Optical fibre latency} = \frac{200 \text{ m}}{50\% \times c} \sim 1.33 \mu\text{s}. \quad (4.4)$$

This means that each slow-control transaction will be considered as having a latency of

$$\text{Slow-Control Latency} = 5.26 \mu\text{s} + 1.33 \mu\text{s} = 6.59 \mu\text{s}. \quad (4.5)$$

However, this latency reflects an estimate corresponding to a slow-control transaction with hardware experimental results without the I2C transactions taking place in the FE. The I2C overheads have been estimated [28] as 14.8 ms for a single HGCROC and as 27.2 ms for a single ECON-T or ECON-D.

Considering a single lpGBT core in the BE issuing all the slow-control transactions in series, the total configuration time of the HGCAL is given by the following equation, as each register in an ASIC corresponds to one slow-control transaction:

$$\begin{aligned} \text{HGCAL configuration time} \leq & 108 \text{ optical fibres} \times \{ \\ & 3 \text{ lpGBTs} \times (\text{N}^\circ \text{ lpGBT registers} \times \text{Slow-Control Latency}) + \\ & 6 \text{ modules} \times [\\ & 3 \text{ HGCROCs / module} \\ & \quad \times (\text{N}^\circ \text{ HGCROC registers} \times \text{Slow-Control Latency} + \text{I2C latency}) + \\ & 2 \text{ ECONs / module} \\ & \quad \times (\text{N}^\circ \text{ ECON registers} \times \text{Slow-Control Latency} + \text{I2C latency}) \\ &] \\ & \}. \end{aligned} \quad (4.6)$$

Hence,

$$\begin{aligned}
\text{HGICAL configuration time} &\leq 108 \times \{3 \times (493 \times 6.59 \mu\text{s}) + 6 \times \\
&\quad [3 \times (1210 \times 6.59 \mu\text{s} + 14.8 \text{ ms}) + 2 \times (2407 \times 6.59 \mu\text{s} + 27.2 \text{ ms})] \\
&\quad \} \\
&\sim 101.13 \text{ s} \\
&= 1 \text{ min and } 41 \text{ seconds.}
\end{aligned} \tag{4.7}$$

Assuming no overheads from synchronisation, a perfect load balancing, and enough software threads, this number can be divided by the number of slow-control cores envisaged to be instantiated in the final BE DAQ FPGA design (16). Hence,

$$\text{HGICAL configuration time} \geq \frac{101.13 \text{ s}}{16} = 6.32 \text{ s.} \tag{4.8}$$

Nevertheless, this scenario is not likely to be precise, as the number of available threads will not be able to execute the software in parallel for every instantiated slow-control core. Moreover, there will be synchronisation overheads and the load balancing is not guaranteed to be perfect across all slow-control cores. Hence, the real HGICAL configuration time will be greater than 6.32 s.

Moreover, there are other unknown factors that will impact both negatively (and positively) this estimate of the configuration time of the detector. Positive contributions include the optimisation of the procedure of reading data from the slow-control AXI interfaces and exploiting the broadcast feature offered by the slow-control cores. Negative factors include additional read transactions sent to the FE to verify the correct configuration and overheads of loading the FE configuration from a database. Nevertheless, both types of contribution may balance each other so that the configuration of HGICAL remains around ~ 1 min. As there are still missing important components of both software and hardware, a more accurate estimation cannot be obtained at this time.

Nevertheless, the described procedure allows estimating the configuration time of HGICAL with the available experimental data. Although those data were not produced using final system hardware or software, they allow to validate that the implemented slow-control architecture meets the requirements in terms of the minimum detector configuration time in the order of a few minutes.

4.4 Hardware Resource Usage

This section examines the resource usage of the slow-control prototyping system (in Section 4.4.1) and of the slow-control infrastructure integrated into the BE DAQ FPGA design (in Section 4.4.2). While the prototyping system does not have a constraint for resource usage in the ZCU102 development board, the slow-control block cannot use more than $\sim 3\%$ of the targeted VU13P FPGA, except for BRAMs.

4.4.1 Resource Usage of the Prototyping System

When implemented in the ZCU102 development board, the developed prototyping system (described in Figure 4.2) uses the resources reported in Table 4.9. The reduced resource usage that is observed allows this prototyping system to be expanded in future works by considering the instantiation of other IP blocks in the FPGA fabric, capable of performing different functionalities, allowing a more comprehensive HGCAL prototype. As an example, such a prototype could also test data acquisition or emulate ASICs not yet available for testing, such as the ECON-D.

Table 4.9: Hardware resources required by the slow-control prototyping system when implemented in the ZCU102 development board using Vivado 2019.2. From [4].

Resource	Used	Available	ZCU102 fraction (%)
LUT	7718	274 080	2.82
LUTRAM	541	144 000	0.38
FF	10 986	548 160	2.00
BRAM	16	912	1.75
DSP	2	2520	0.08
IO	2	328	0.61
GT	1	24	4.17
BUFG	8	404	1.98
MMCM	1	4	25.00

This expansion of the current prototyping system can be conducted not only vertically (i.e., by the addition of new functionalities or features) but also horizontally, by considering the introduction of redundancy in already existing components. This would involve the addition of more lpGBTs and GBT-SCAs to obtain a larger FE, able to test parallel slow-control transactions with ASICs. Upon the availability of the final detector hardware, the current FE and BE emulation platforms can also be replaced by final detector parts.

4.4.2 Resource Usage and Optimisation of the Slow-Control

After the synthesis and implementation of the BE DAQ FPGA design, the resource usage of Table 4.10 was reported by Vivado 2021.2 for the validated slow-control block, highlighted in red in Figure 4.2. Apart from the BRAMs, the resource usage is under 3%, as required. This larger BRAM usage (9.52%) is already allocated for the slow-control block in the BE DAQ FPGA design, comprising enough storage for 16 lpGBT and 16 GBT-SCA cores. Since the BE DAQ FPGA design is being developed with the Vivado 2021.2 toolchain, the 2019.2 version used in the prototyping systems is not used to obtain the resource usage of the hardware required to implement this design.

During the described prototyping and testing procedures, other optimisations were introduced in the VHDL code of the slow-control block to reduce resource usage. A notable improvement was the use of DSPs in the FPGA to implement small counters needed by the transactors of each core. Despite

Table 4.10: Hardware resources required by the developed slow-control when implemented in a VU13P FPGA using Vivado 2021.2 toolchain. 16 lpGBT and 16 GBT-SCA cores were instantiated.

Resource	Used	Available	VU13P fraction (%)
LUT	14 232	1 728 000	0.82
FF	25 620	3 456 000	0.74
BRAM	256	2688	9.52
DSP	96	12 288	0.78

using DSPs, that are a big FPGA cell with high computational capabilities, to perform such a simple functionality, the BE DAQ FPGA design did not use DSPs anywhere else in the design. Therefore, even if the DSPs are used to perform such a simple functionality, this allows reducing the usage of both LUTs and FFs in the design, by utilising a cell not required by other components of the BE.

4.5 Slow-Control Portability Evaluation

After the presented validation of the slow-control block, and in addition to the expansion of the developed prototyping system, it is now possible to proceed with the integration of the slow-control into other existing HGCal systems such as more complex (and complete) prototyping systems [13] and the final detector design. These integrations comprise no change to the slow-control block as it is portable among those systems despite the involved emulation in its testing.

Although the hardware of the final detector is not yet available, the developed prototyping system allowed to test and validate the slow-control, which culminates in a reliable connection between the software running in a MPSoC and the FE electronics. Furthermore, the used technologies ensure portability between the prototyping systems and the final detector. In particular, the common AXI interfaces and a standardised Linux framework allow the usage of the software drivers that interface the slow-control. Similarly, compatibility in terms of hardware is guaranteed by the use of Xilinx Ultrascale+ FPGA fabric technology both in the prototyping and final detector systems.

Similar results are obtained when considering the expansion of the current prototyping system. Since the lpGBT and HDLC protocols have been validated, when the final detector hardware becomes available and replaces the VLDB+ and VLDB boards, no change should be required to the slow-control FPGA hardware or software. The same happens in the BE due to the adopted standardised AXI connection and common FPGA fabric hosted in the ATCA board. No changes shall be needed in the final implementation of the slow-control block.

4.6 Summary

This chapter describes the slow-control block and the prototyping system developed to validate its functionality. Moreover, it was possible to roughly estimate the configuration time of the detector to about 1 min. Full portability of the block among different prototyping systems and the final detector is ensured due to the chosen technologies.

5

Conclusion

Contents

5.1	Conclusions	63
5.2	Future Work	64

The HGCAL will replace and upgrade the current endcap calorimeters of the CMS detector at the CERN LHC. This new calorimeter will provide data from particles produced in collisions with a high spatial, energy, and time resolution using around 150 000 ASICs spread out through its FE electronic readout chain and about 1000 m² of area.

This work contributes to the control system used to configure these ASICs and to the testing of the HGCROC, of which there will be about 100 000 in HGCAL.

The validation of the slow-control block, responsible for the ASIC control and configuration, was possible through the development of a prototyping system that used surrogate hardware to replace the final detector hardware still under development. This part of the work was presented at the 33rd International Workshop on Rapid System Prototyping (RSP), Shanghai, China in October 2022 [4].

The MARS IP was developed to accelerate the HGCROC testing through hardware acceleration in FPGAs and was being deployed in the different test systems of HGCAL at the time of writing.

5.1 Conclusions

The MARS accelerates the computation of metrics used in the production and characterisation testing of the HGCROC ASIC in hardware. Its pre-synthesis configuration parameters and low FPGA resource usage allow for high flexibility regarding the amount of data to process. Hence, MARS has been integrated into several testing and prototyping systems with different requirements.

The HGCROC characterisation tests performed at CERN provided positive results of the integration of MARS on HGCROC characterisation tests. A speedup of up to 4.15 was obtained when using MARS allowing for more detailed testing and improving the scalability of the testing system. The MARS is in the process of being integrated into the production testing of the HGCROC where a positive impact of the usage of MARS is expected.

After production testing and after integration of the HGCAL detector system, all FE electronics will be controlled from the BE FPGAs through the slow-control block. Each FPGA interfaces directly up to 750 IpGBT or GBT-SCA ASICs, through which I2C transactions are conveyed to the remaining FE ASICs. Hence, up to 3500 ASICs are configured per FPGA.

Although the final system ATCA board for the BE is still under development, the surrogate hardware used in this work allowed to test and validate the functionality of the slow-control architecture for the whole system. This was possible through the establishment of a reliable connection between the software running on a MPSoC and the ASICs that will be used in the final system.

Besides the validation of the protocols used in the slow-control, this prototype also allowed to ensure full portability of the slow-control between the different platforms used in test systems and the final detector due to the use of shared technology.

Furthermore, the measurements performed in the slow-control prototype already allow to roughly estimate the configuration time of the HGCal FE to be of the order of about 1 min. Although uncertainties still exist regarding other detector components, such as part of the FE latency and software, the estimate is considered to be accurate enough to ensure a configuration time that meets the required performance.

5.2 Future Work

Despite the obtained success and performance regarding slow-control transactions, bottlenecks might be hidden in the software under development to operate the HGCal. Should the slow-control scheme need a performance improvement, some architectural aspects can still be explored, such as the introduction of new modes of operation that allow sending transactions to the FE without waiting for the corresponding replies. This new operation mode would minimise the impact of latency in the system through the pipelining of the transactions. This new feature would come at a cost of using more FPGA resources. However, the low resource usage of the current architecture makes this strategy promising and worth exploring should it be required.

As critical transactions could require the stopping of the sending process to wait for a specific FE reply, these new architectures should also support new memory words that would allow controlling the sending procedure.

The low resources of the slow-control prototyping system also allow further work in this system to evolve into the integration of new components with a possible scope exchange. Besides the upgrade of either the BE or FE components to use final hardware, this expansion can evolve in different directions, either horizontally or vertically, as described below.

The horizontal expansion involves the addition of more components already in use to widen the slow-control tree. These modifications involve establishing more communication channels with IpGBTs and GBT-SCAs.

The vertical expansion involves the addition of different components to test more interfaces of the HGCal either in the BE or FE. As the communication protocols of the slow-control have been validated, these changes should be transparent, allowing the focus of the system testing to shift to the validation of other aspects.

Either one of these options will increase the complexity of the current system, continuing the effort to test the detector system up to the implementation of a complete HGCal prototype.

Bibliography

- [1] M. Brice, “Aerial View of the CERN taken in 2008.” 2008. [Online]. Available: <https://cds.cern.ch/record/1295244> (cited in pp. ix and 4).
- [2] T. Sakuma, “Cutaway diagrams of CMS detector,” 2019. [Online]. Available: <http://cds.cern.ch/record/2665537> (cited in pp. ix and 5).
- [3] D. Barney, “Overview slide of CE with main parameters,” june 2021. [Online]. Available: https://cms-docdb.cern.ch/cgi-bin/DocDB/RetrieveFile?docid=13251&filename=OverviewDrawing_October2019.pdf&version=9 (cited in pp. ix and 7).
- [4] M. Rosado, S. Mallios, P. Tomás, N. Roma, and A. David, “Early prototyping and testing of CERN LHC CMS high-granularity calorimeter slow-control system,” in *International Workshop on Rapid System Prototyping (RSP)*, Oct. 2022. (cited in pp. ix, xi, xiii, 9, 14, 43, 44, 50, 57, 63, and 69).
- [5] P. Moreira, S. Baron, S. Biereigel, J. Carvalho, B. Faes, M. Firlej, T. Fiutowski, J. Fonseca, R. Francisco, D. Gong, N. Guettouche, P. Gui, D. Guo, D. Hernandez, M. Idzik, I. Kremastiotis, T. Kugathasan, S. Kulis, P. Leitao, P. Leroux, E. Mendes, J. Mendez, J. Moron, N. Paulino, D. Porret, J. Prinzie, A. Pulli, Q. Sun, K. Swientek, K. Wyllie, D. Yang, J. Ye, T. Zhang, and W. Zhou, *IpGBT documentation: release*, Mar 2022. [Online]. Available: <https://cds.cern.ch/record/2809058> (cited in pp. x, 15, and 21).
- [6] CMS Collaboration, “HGCROC3 Spec Working Document 2.0,” April 14, 2021. (cited in pp. x and 29).
- [7] M. Rosado, “MARS BLOCK (Multiplexed Accumulator for RocS),” DAQ and FE coordination meeting, September 13, 2021. Internal note. [Online]. Available: <https://indico.cern.ch/event/1066025/> (cited in pp. x, 28, 31, 33, and 34).
- [8] *UltraScale Architecture Libraries Guide (UG974)*, Xilinx. [Online]. Available: <https://docs.xilinx.com/r/en-US/ug974-vivado-ultrascale-libraries> (cited in pp. x, 35, and 36).

- [9] CMS Collaboration, “The cms experiment at the cern lhc,” *Journal of Instrumentation*, vol. 3, no. 08, p. S08004, aug 2008. [Online]. Available: <https://dx.doi.org/10.1088/1748-0221/3/08/S08004> (cited in p. 4).
- [10] —, “The Phase-2 Upgrade of the CMS Endcap Calorimeter,” CERN, Geneva, Tech. Rep., Nov 2017. [Online]. Available: <https://cds.cern.ch/record/2293646> (cited in p. 6).
- [11] C. Ochando, “HGCAL: A High-Granularity Calorimeter for the endcaps of CMS at HL-LHC,” *J. Phys.: Conf. Ser.*, vol. 928, no. 1, p. 012025, 2017. [Online]. Available: <https://cds.cern.ch/record/2311394> (cited in p. 6).
- [12] 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> (cited in p. 6).
- [13] N. Strobbe and on behalf of the CMS collaboration, “Readout electronics for the cms phase ii endcap calorimeter: system overview and prototyping experience,” *Journal of Instrumentation*, vol. 17, no. 04, p. C04023, apr 2022. [Online]. Available: <https://dx.doi.org/10.1088/1748-0221/17/04/C04023> (cited in pp. 14, 16, and 58).
- [14] J. Troska, A. Brandon-Bravo, S. Detraz, A. Kraxner, L. Olanterä, C. Scarcella, C. Sigaud, C. Soos, and F. Vasey, “The VTRx+, an optical link module for data transmission at HL-LHC,” *PoS, vol. TWEPP-17*, p. 048. 5 p, 2017. [Online]. Available: <https://cds.cern.ch/record/2312396> (cited in p. 15).
- [15] A. Caratelli, S. Bonacini, K. Kloukinas, A. Marchioro, P. Moreira, R. De Oliveira, and C. Paillard, “The GBT-SCA, a radiation tolerant ASIC for detector control and monitoring applications in HEP experiments,” *JINST*, vol. 10, p. C03034, 2015. [Online]. Available: <https://cds.cern.ch/record/2158969> (cited in p. 16).
- [16] CERN EP-ESE Group. The GBT Project. [Online]. Available: <https://espace.cern.ch/gbt-project/> (cited in pp. 16 and 46).
- [17] M. Noy and on behalf of the CMS collaboration, “The cms hgc al silicon region architecture specification and optimisation,” *Journal of Instrumentation*, vol. 17, no. 03, p. C03010, mar 2022. [Online]. Available: <https://dx.doi.org/10.1088/1748-0221/17/03/C03010> (cited in p. 16).
- [18] “ISO/EIC 13239 : 2002(E) Information technology — Telecommunications and information exchange between systems — High-level data link control (HDLC) procedures,” HDLC standard. (cited in pp. 18 and 44).

- [19] 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> (cited in pp. 19, 43, and 47).
- [20] *AXI Chip2Chip v5.0 LogiCORE IP Product Guide*, Xilinx. [Online]. Available: <https://docs.xilinx.com/r/en-US/pg067-axi-chip2chip/AXI-Chip2Chip-v5.0-LogiCORE-IP-Product-Guide> (cited in p. 19).
- [21] S. Bonacini, K. Kloukinas, and P. Moreira, “E-link: A Radiation-Hard Low-Power Electrical Link for Chip-to-Chip Communication,” 2009. [Online]. Available: <https://cds.cern.ch/record/1235849> (cited in p. 21).
- [22] J. M. Mendez, S. Baron, S. Kulis, and J. Fonseca, “New LpGBT-FPGA IP: Simulation model and first implementation,” *PoS*, vol. TWEPP2018, p. 059, 2019. [Online]. Available: <http://cds.cern.ch/record/2710381> (cited in p. 21).
- [23] ARM, “AMBA® AXI™ and ACE™ Protocol Specification,” AXI4 protocol specifications from ARM. [Online]. Available: <https://developer.arm.com/documentation/ih0022/e/AMBA-AXI3-and-AXI4-Protocol-Specification> (cited in p. 31).
- [24] A. Steen, “Hexa-controller test software : updates and MARS,” DAQ and FE coordination meeting, May 16, 2022. Internal note. [Online]. Available: <https://indico.cern.ch/event/1157474/> (cited in p. 38).
- [25] *ZCU102 Evaluation Board User Guide*, Xilinx. [Online]. Available: <https://docs.xilinx.com/v/u/en-US/ug1182-zcu102-eval-bd> (cited in p. 50).
- [26] D. Montesinos, S. Baron, N. Guettouche, and J. Mendez, “The versatile link+ demo board (VLDB+),” *Journal of Instrumentation*, vol. 17, no. 03, p. C03032, mar 2022. [Online]. Available: <https://doi.org/10.1088/1748-0221/17/03/c03032> (cited in p. 50).
- [27] R. Martin Lesma, F. Alessio, J. Barbosa, S. Baron, C. Caplan, P. Leitao, C. Pecoraro, D. Porret, and K. Wyllie, “The Versatile Link Demo Board (VLDB),” *JINST*, vol. 12, p. C02020. 12 p, 2017. [Online]. Available: <https://cds.cern.ch/record/2275133> (cited in p. 50).
- [28] J. Wilson, “HGCAL slow control time,” DAQ and FE coordination meeting, November 23, 2020. Internal note. [Online]. Available: <https://indico.cern.ch/event/871877/> (cited in pp. 54 and 55).



Publications

This appendix collects the publication presented at the 33rd International Workshop on Rapid System Prototyping (RSP), resulting of the prototyping system for the slow-control block [\[4\]](#), an important complement to Chapter 4.

Early prototyping and testing of CERN LHC CMS high-granularity calorimeter slow-control system

Martim Rosado^{*†}, Stavros Mallios^{*}, Pedro Tomás[†], Nuno Roma[†] and André David^{*}

^{*}on behalf of the CMS Collaboration, CERN, Geneva, Switzerland

[†]INESC-ID, Instituto Superior Técnico, Universidade de Lisboa, Portugal

Abstract—The Compact Muon Solenoid (CMS) high-granularity calorimeter (HGCAL) upgrade for CERN's Large Hadron Collider (LHC) high-luminosity phase is a detector with more than 6 million channels that will provide precise sensing and measurement of position, timing, and energy of the particles produced in the collisions of the beams. The HGCAL electronics are a large and complex set of processing systems split into front-end and back-end. The front-end, located in the experimental cavern, consists of ≈ 150 thousand radiation tolerant ASICs. The high-density FPGA-based back-end is housed away from the radiation area in a set of Advanced Telecommunications Computing Architecture (ATCA) boards and crates hosting ≈ 100 FPGAs. Each ATCA back-end board will comprise one (or two) FPGAs, managing up to ≈ 120 optical links, each providing a transmission rate of 10.24 Gb/s between the back-end and the front-end electronics. Each back-end FPGA is responsible for configuring and monitoring up to ≈ 3500 front-end ASICs and will be controlled by software running on a back-end MPSoC that provides the entry point for the whole control procedure. This paper presents the design and implementation of the prototyping infrastructure deployed to test and validate the slow-control block of the HGCAL back-end electronics, together with the related interfaces with the controller MPSoC and the front-end transceiver ASICs. The required functionalities have been validated with a ZCU102 Xilinx Ultrascale+ development board, which emulated the back-end elements that are still under development and not yet available for this comprehensive test. This development board was connected to other custom ASIC development boards via optical links, emulating the front-end side of the system, also still under development. Besides providing reliable testing and validation of the operation of the whole infrastructure, the prototyping platform also allowed to attain the required software/hardware portability that ensures easy integration/replacement of all the (still) emulated components with their final implementations.

Index Terms—Fast Prototyping, System Emulation, Testing and Validation, Field-Programmable Gate-Array

I. INTRODUCTION

The High Luminosity phase of the Large Hadron Collider (HL-LHC) at CERN (European Organization for Nuclear Research) motivates the replacement and upgrade of multiple sub-detectors of the Compact Muon Solenoid (CMS) detector. The upgrade is needed because the current detector will not have an acceptable performance under HL-LHC operation conditions with the higher levels of radiation. The High Granularity Calorimeter (HGCAL) [1] is a new detector that will replace the current CMS endcap calorimeter systems, which comprise both the electromagnetic and the hadronic

calorimeters. With more than 6 million channels, the HGCAL will measure position, timing and energy of particles produced in collisions of the LHC beams with unprecedented resolution.

The intense radiation environment necessitates the use of a resilient electronics chain which is based on the use of radiation-tolerant ASICs (Application Specific Integrated Circuit) in the front-end electronics of the HGCAL detector. These ASICs read the more than 6 million active channels, then digitise and process the signals acquired from silicon sensors and from plastic scintillator tiles read out by silicon photomultipliers. That information is conveyed using 10.24 Gb/s optical links to ≈ 100 large (radiation sensitive) FPGAs (Field Programmable Gate Array) mounted on Advanced Telecommunications Computing Architecture (ATCA) boards in the back-end (off detector). Each back-end FPGA is responsible not only for processing data from up to ≈ 120 of these links but also for controlling and monitoring up to 3.5 thousand front-end ASICs producing those data [2]. Two types of commands are needed to control the front-end ASICs:

- 1) Synchronous commands (fast-control) responsible for triggering and managing data acquisition, e.g. preventing buffer overflows in the FPGAs. These commands are sent to the front-end in a 320 Mb/s serial bit stream.
- 2) Asynchronous commands (slow-control) responsible for configuring and monitoring the front-end ASIC parameters. These commands are issued asynchronously to the data-taking and are transmitted via an 80 Mb/s serial bit stream; the ASIC configuration does not need to be changed often, thus, these commands do not need to be sent at a high frequency.

Considering the cost and magnitude of the overall detection and processing system, the feasibility of the concept must be thoroughly tested and prototyped. Consequently, performing rapid prototyping of a vertical slice of the HGCAL global system was required, i.e. a system in which the front-end sensors are connected to the back-end electronics with the minimum number of components. The system will then grow horizontally, integrating more components of the same type, up to the size of the final prototype. As part of this effort, it was decided to prototype the FPGA hardware block that will handle sending slow-control commands to the front-end ASICs, ahead of the availability of the final FPGAs and ATCA modules.

The main contribution of this work is the conception of

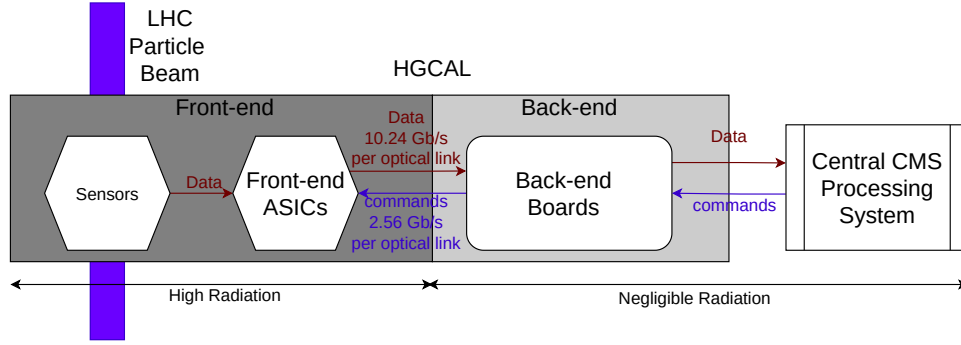


Fig. 1. Schematic overview of the HGCal electronics systems, including those on-detector in the high-radiation environment and those off-detector in a shielded underground cavern.

a prototyping system used to test and validate the interface between the slow-control block and the radiation-tolerant ASICs of the final detector. The validation of the slow-control block allows for other HGCal test systems and prototypes to use the final detector framework of front-end configuration.

Since both the final back-end infrastructure and ATCA communication boards are still under development and not available for testing, a ZCU102 Xilinx FPGA development board [3], equipped with a Zynq MPSoC, was used to emulate the back-end platform. The front-end infrastructure was also emulated with custom development boards containing the target ASICs. This way, even though the final detector boards might be different from those used in this prototyped emulation, it is possible to test the communication infrastructure with the detector ASICs that will be used in the final system, validating the whole configuration chain.

Moreover, this prototyping infrastructure allowed a comprehensive portability evaluation of both the software and the FPGA-deployed hardware elements that will be present in the final system. Furthermore, this setup allowed for the parallel development of the slow-control FPGA hardware, its interface software, and the corresponding feasibility studies for its integration into the final back-end system (to be implemented in FPGA).

The remainder of this paper is organized as follows: Section II briefly describes the HGCal detector system; Section III describes the slow-control block and its interactions and interfaces within the broader system; The developed test system is the subject of section IV along with a discussion of the test procedures; Results are presented in section V; The conclusion in section VI; Finally, section VII discusses future work.

II. CMS HIGH GRANULARITY CALORIMETER

The High Granularity Calorimeter of CMS is a particle detector being developed in the scope of the HL-LHC upgrade, expected to be completed by the end of 2027. This detector will sample data (timing, energy, and position) of particle collisions with unprecedented high resolution. The HGCal electronics system is divided into two distinct parts: the front-end and the back-end, as shown in Fig. 1.

The front-end electronics are located in the CMS detector and hence operate in a high radiation area. With the use of silicon sensors and plastic scintillator tiles as sensitive materials, data gathered from the particle collisions are acquired through more than 6 million channels and undergo initial processing before being conveyed to the back-end electronics. The front-end electronics are based on radiation-tolerant ASICs to deal with the high radiation environment.

The back-end is connected to the front-end via optical links and is located off the detector, safe from radiation. It receives the front-end data at 10.24 Gb/s per link. The back-end incorporates several ATCA communication boards, each with one (or two) VU13P (Virtex Ultrascale+) Xilinx FPGAs. The back-end not only receives data from the front-end but also pre-processes it before sending the data to the central CMS processing system. Each back-end FPGA manages data from up to ≈ 120 of these links while also configuring and monitoring up to ≈ 3500 front-end ASICs.

III. SYSTEM BEING PROTOTYPED

The slow-control block is located in the back-end FPGA on the ATCA boards (see Fig. 2) and is responsible for configuring and monitoring the front-end ASICs. This is achieved with multiple channels through which transactions are performed, consisting of several read/write operations to the front-end ASICs' internal registers. The information is communicated to the slow-control via a software layer running on a Zynq MPSoC, also hosted on the back-end ATCA board. This MPSoC provides the software with an entry point that opens a communication line with the back-end FPGA via Advanced eXtensible Interface (AXI) Chip2Chip [4], [5].

Each slow-control block interfaces with up to 756 front-end ASICs of two different types:

- 1) The Low Power Giga Bit Transceiver (lpGBT) [6] ASIC that provides high-speed bidirectional communication, and
- 2) The Giga Bit Transceiver - Slow-Control Adapter (GBT-SCA) [7] ASIC, which was specifically developed for sending configuration and monitoring commands throughout the front-end of particle detectors.

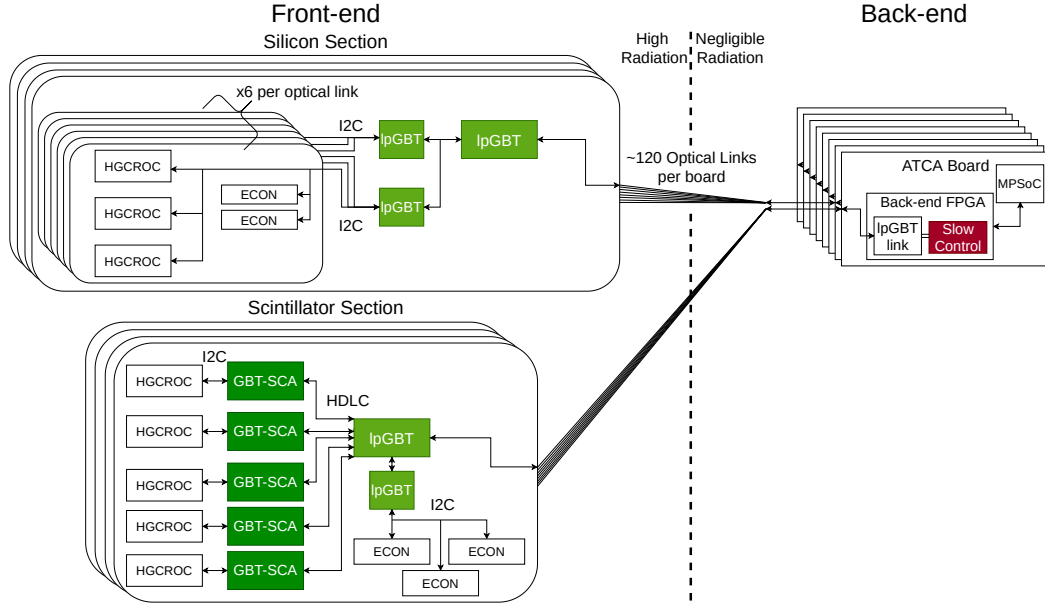


Fig. 2. Main interfaces and front-end endpoints of the back-end slow-control block being prototyped in this work.

A. Slow-control Connection to the Front-end

The HGCAL front-end is divided into two sections by the type of sensitive material used; silicon sensor (silicon section) or plastic scintillator (scintillator section), leading to two architectures, shown in Fig. 2.

After particle collisions, the sensors' signals are first digitized by the HGCAL readout chip (HGCROC). Next, the data are transmitted to the Elink concentrator ASICs (ECON) and pre-processed before being sent to the back-end. Both the HGCROC ASICs and ECON ASICs are configured via I2C using either an IpGBT or GBT-SCA that have I2C masters that are controlled by the slow-control block in the back-end.

The silicon section does not use GBT-SCAs and has a higher ASIC density per back-end FPGA than the scintillator section. For communication, each back-end FPGA will service ≈ 120 optical links, each connected to a single IpGBT. Each IpGBT is subsequently connected to:

- 1) 2 other IpGBTs, further connected to ≈ 18 HGCROCs and ≈ 12 ECONs (I2C targets), totalling up to ≈ 3500 ASICs to configure per back-end FPGA in the silicon section, and
- 2) 1 other IpGBT, up to 5 GBT-SCAs and up to ≈ 8 I2C targets, totalling up to ≈ 1600 ASICs to configure per back-end FPGA in the scintillator section.

Although the GBT-SCA data is transmitted through the IpGBTs, the GBT-SCA transactions are transparent to the IpGBT; therefore, the GBT-SCAs can be considered to be directly interfaced to the slow-control. The same applies to the IpGBTs not connected directly to the optical links.

To establish communication between the back-end and

front-end through the slow-control block, the following procedure is undertaken:

- 1) The software running in the ATCA MPSoC transfers the data to the slow-control block in the back-end FPGA;
- 2) The slow-control block processes these data and issues the required read/write transactions to the target IpGBTs and/or GBT-SCAs;
- 3) The replies from the target IpGBTs and/or GBT-SCAs are read by the slow-control block and stored;
- 4) The MPSoC software reads the data in the replies from the slow-control block.

To fully understand the requirements, it is important to observe that, although the front-end configuration is not changed very often, it must be done with low latency so as to minimize the detector dead time when configuring the front-end electronics, as important data may be missed while the beams are colliding.

B. Communication Infrastructure and Protocols

To reach the front-end, the slow-control block implements the communication protocols of both the IpGBT and GBT-SCA transceivers [2]. The IpGBTs use a custom protocol [6], and the GBT-SCAs use the HDLC (High-level Data Link Control) protocol. These protocols allow the reading from and writing to registers in the IpGBTs and GBT-SCAs, such as configuration and status registers of their I2C masters. It should be noted that each transaction using these protocols triggers a reply from the corresponding IpGBT or GBT-SCA.

For each IpGBT and GBT-SCA interface, the slow-control block produces an 80 Mb/s output signal and receives another

80 Mb/s input signal. These bitstreams are implemented internally in the back-end FPGA as 2-bit encoded signals at 40 MHz and correspond to a subset of the full lpGBT data frame presented in Fig. 3, namely the Internal Control (IC) [6] and External Control (EC) [6] fields. Therefore these signals cannot be connected directly to the transceiver that sends data to the lpGBT and need to be provided to an intermediate block (lpGBT link) that is in charge of encoding the complete lpGBT data frames. The GBT-SCA stream is also connected to this lpGBT interface and uses part of the eLinks field [6].

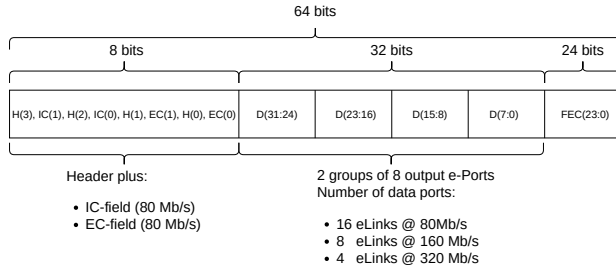


Fig. 3. The lpGBT data frame, from [6]. The slow-control block makes use of both the IC and EC fields that are dedicated to this type of communication and, in the scintillator section, also uses some of the eLinks to reach up to 5 GBT-SCAs in total.

In each optical link, the front-end lpGBT that is directly connected to the back-end is reached via the IC bitstream while the other lpGBTs are reached via the EC bitstream. As an example, in the silicon section, this EC channel is shared by two lpGBTs. Hence, per optical link, two 80 Mb/s output streams are required to interface up to 3 lpGBTs, and one lpGBT link must be instantiated in the back-end FPGA.

C. Slow-control Architecture

Because of FPGA resource limitations [2], it is not possible to implement a fully parallel structure that allows the transmission of simultaneous transactions to all lpGBT and GBT-SCA interfaces, so it is necessary to multiplex the transactions between several front-end lpGBTs or GBT-SCAs. To achieve a compromise between FPGA resource usage and detector configuration time, several small blocks with multiplexing capabilities, denoted as “lpGBT cores” or “GBT-SCA cores” [2], are used to achieve some level of parallelism. Fig. 4 shows the slow-control block architecture.

1) *Memory and Control Module:* A memory module is needed because each of these cores has to receive transaction data from the back-end MPSoC ARM core and store the front-end replies. This functionality is implemented through the use of BRAMs in the FPGA programmable logic. Each lpGBT or GBT-SCA core has a send buffer and a receive buffer, and its own set of status and control registers. The data word size for each transaction is 128 bits and it was decided to store data from 1024 transactions in each buffer. To accommodate this, both buffers are built with 4 BRAMs and have “AXI Full” connections to communicate with the MPSoC ARM cores.

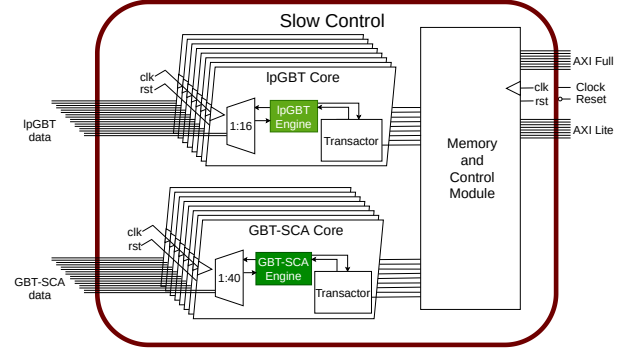


Fig. 4. Simplified block diagram of the slow-control architecture, showing how a single FPGA holds a number of slow-control cores, each multiplexing signals to multiple lpGBT or GBT-SCA interfaces.

The control and status registers are accessible via an “AXI Lite” interface.

2) *The lpGBT and GBT-SCA Cores:* Each core is associated with the front-end lpGBT or GBT-SCA communication protocol. The core is responsible for reading the transaction data from the respective buffer, processing it, and multiplexing the data to the respective 80 Mb/s stream, which corresponds to the correct lpGBT or GBT-SCA ASIC in the front-end. Afterwards, the core receives and stores the reply in the receive buffer. The current design [2] has a multiplexing factor of 16 in the lpGBT core and of 40 in the GBT-SCA core. With these configurations, it is possible to achieve a full detector coverage with 16 units of each core in each back-end FPGA, as depicted in Fig. 4.

Both core types are similar in structure: a small state machine (the transactor) interfaces with the buffers and controls the sending and receiving of data. It was chosen to not support concurrent transactions in each lpGBT and GBT-SCA core, meaning that one transaction only starts after the previous one is finished. This way, except in a timeout scenario, each transactor waits for the response of the current transaction before starting the next.

Each transactor output is fed to the respective engine (lpGBT or GBT-SCA). The engine is responsible for the decoding of software data into lpGBT and GBT-SCA transactions and encoding back the corresponding response. The engine output is routed to the channel connected to the target lpGBT or GBT-SCA.

The described chain corresponds to a complete sending procedure. A similar, but reverse, operation happens when receiving front-end replies. The response from the target lpGBT or GBT-SCA is sent to the engine that encodes the data into the data format expected by the MPSoC ARM core. The engine output is fed into the transactor, which stores the data into the respective receive buffer and then starts processing the next transaction.

IV. SYSTEM EMULATION, VERIFICATION, AND VALIDATION

Preliminary prototyping and subsequent verification of the slow-control block requires careful validation of both its internal operation and its interfaces, namely the AXI interface with the MPSoC ARM core, and the connection with the lpGBTs and GBT-SCAs. This section describes what was done to achieve such a comprehensive validation.

A. Emulation and Testing Infrastructure

The dimension and complexity of the sensing and data acquisition system that supports this CMS detector upgrade, allied with the fact that a significant amount of the intervening parts and components of the HGCAL are still under development and not already available for a complete deployment and system integration mean that certain modules to which the slow-control block will be interfaced are implemented in surrogate hardware, shown in Fig. 5. Such surrogates ensure accurate stimulus of the several elements, together with the reception and validation of the received replies.

Due to the current unavailability of a Zynq-controlled ATCA board prototype, a Xilinx ZCU102 development board was selected as the surrogate for the back-end platform. This board allows to connect the incoming optical links from the front-end at the required data rates to its SFP+ connectors and MGTs (Multi-Gigabit Transceiver). The ZCU102 is equipped with a Zynq MPSoC based on the same technology as the target back-end FPGA, the VU13P. In particular, both have Ultrascale+ FPGA fabric. Furthermore, the ARM processing system on the MPSoC also allows the use of AXI connections as in the final system, alongside a Linux installation, also used in the final system. The test system uses the CentOS 7 Linux distribution.

In the back-end board's programmable logic (PL), the slow-control block is connected to the MPSoC processing system (PS) through its two AXI interfaces and a Xilinx AXI interconnect IP. Its 80 Mb/s data streams to/from the front-end are connected to the lpGBT link module, which is responsible for parsing the complete lpGBT data frames and ensuring a reliable link between the back-end and front-end. The bitstreams corresponding to the lpGBT data were connected to the IC field of the lpGBT frame and the bitstreams corresponding to the GBT-SCA were packed in the EC field. Although in the final system the GBT-SCA connection will not use this EC field of the lpGBT frame, this setup is acceptable for slow-control prototyping purposes since it does not affect the validation of the communication protocol as GBT-SCA data remains transparent to the lpGBT.

The output of the lpGBT link is then connected to the ZCU102 MGTs, allowing an optical fibre to be connected to the board SPF+ connectors. The GTH MGT in the ZCU102 board was configured using a Xilinx IP to operate at a transmission rate of 10.24 Gb/s. Data decoding and encoding when interfacing the GTH are done by the lpGBT link as it is implementing its own custom protocol designed to withstand radiation conditions [6].

To fulfil these prototyping conditions, the ZCU102 settings were changed to clock the MGT at 320 MHz as required by the lpGBT transmission rates. This clock was also used to generate the clock signal that drives the slow-control block at 40 MHz since the phase of the clocks of the MGT and the slow-control block need to be related. Both AXI interfaces are also clocked at 40 MHz but are driven from a different clock source from the PS. The design of the FPGA logic was done with the Xilinx Vivado 2019.2 software and using the VHDL hardware description language.

The surrogate for the front-end side of the prototyping system was a VLDB+ development board [8] that contains an lpGBT device like the HGCAL front-end detector boards. The VLDB+ was connected to the optical fibre coming out of the back-end board.

A similar procedure was followed to add a GBT-SCA to the system. A VLDB development board [9], which includes a GBT-SCA, was connected to the VLDB+ board and assigned the EC channel of the lpGBT, completing the testing infrastructure.

B. Validation Criteria and Tests

The AXI interface between the slow-control block and the MPSoC ARM core can be validated by writing a predefined sequence of data words to system memory followed by a subsequent reading of the same addresses. The validation procedure for this specific interface consisted of an exhaustive test that checked every available address on the specified address spaces of both the AXI Full and AXI Lite interfaces of the slow-control block.

Similarly, communication with the front-end was validated by issuing a sequence of write and read operations to the lpGBT and GBT-SCA internal registers. These operations can be used to change the configuration of the ASICs such as changing the operation mode, enabling I2C masters, controlling general purpose input/output pins (GPIOs), etc. The outcome of such operations was validated by the reply data stored in the receiving buffers.

V. RESULTS

To comprehensively validate and test the operation of the system in Fig. 4, a sequence of testing procedures targeting each individual part of the infrastructure was devised and is described in this section. To ensure the proper operation of the AXI interfaces, a Python script was run on the prototyping MPSoC ARM core that writes data words to all address positions on the memory and control module of the slow-control block and then reads back the same positions.

To test the communication with the front-end lpGBT, the values of several lpGBT internal registers were changed and read back to verify that the changes were successfully applied. Some changes concerned specific configuration registers of the lpGBT, such as changing the operation state of the lpGBT or configuring the EC channel. By monitoring an LED on the VLDB+ board that indicates whether the operation of the lpGBT is in its ready state, it was possible to confirm the

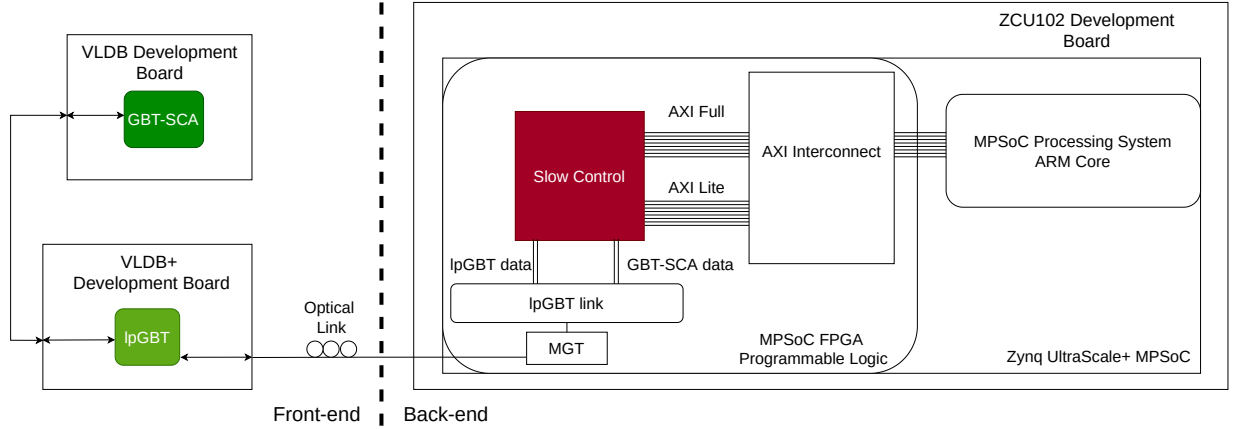


Fig. 5. Slow-control block testing infrastructure using surrogate hardware. The ZCU102, VLDB+, and VLDB boards present the same functionality and interface as the final hardware even if the latter is not yet available.

requested change in the lpGBT operation state. The correct behaviour of the configured EC channel was tested by conveying correct information to the GBT-SCA.

Similarly to the lpGBT, the GBT-SCA HDLC protocol was validated by writing and reading configuration registers of the GBT-SCA. Moreover, control of the GBT-SCA's GPIOs was validated using the connected LEDs on the VLDB board.

Although not all of the internal registers of the lpGBT and GBT-SCA were accessed in these tests, the functionality of the slow-control block was stress-tested with a specific procedure. First, the send buffers were loaded to full capacity. Next, the slow-control issued the corresponding transactions to the front-end and the MPSoC ARM core read all the front-end replies. This stress test was repeated 10000 times in a row and all (≈ 10 million) transactions were successful.

Both lpGBT and GBT-SCA cores achieved a transaction rate greater than 230000 transactions per second. The observed rates can be used to roughly extrapolate the total HGCAL configuration time. In order to have a reasonably accurate estimate, other contributions have to be considered as well, including the I2C transaction time, the optical fibre latency, and software overheads. With rough estimates of these other contributions, it was possible to estimate that the full HGCAL can be configured in ≈ 1 minute.

Although this result yields a reasonable configuration time for the detector, there are still uncertainties in several contributing factors such as front-end experimental results and software bottlenecks. In particular, there are both unknowns that can increase the configuration time (e.g. when taking into account the configuration verification read-back) and opportunities for optimisation that remain unexplored (e.g. in terms of concurrent software access). Overall, these likely cancel each other out, and the order of magnitude for the time to configure the HGCAL is expected to remain on the order of minutes. In this sense, the performance of the conceived slow-control architecture is expected to allow for an acceptable

configuration time of the detector.

Once the front-end communication protocols were validated, the slow-control block is ready to be integrated into the HGCAL global prototype and other HGCAL test systems, fulfilling the purpose of the developed test system.

Table I lists the hardware resources required by the slow-control block testing system developed for the ZCU102 board, which is shown on the right side of Fig. 5. The small size of this system allows for further expansion, with a possible change in scope to test other detector functionality, like fast commands, data acquisition, etc.

TABLE I
HARDWARE RESOURCES REQUIRED BY THE SLOW-CONTROL TESTING SYSTEM WHEN IMPLEMENTED FOR THE ZCU102 DEVELOPMENT BOARD IN VIVADO 2019.2.

Resource	Used	Available	Fraction of ZCU102 (%)
LUT	7718	274080	2.82
LUTRAM	541	144000	0.38
FF	10986	548160	2.00
BRAM	16	912	1.75
DSP	2	2520	0.08
IO	2	328	0.61
GT	1	24	4.17
BUFG	8	404	1.98
MMCM	1	4	25.00

Although the final detector back-end boards are not yet available, the testing infrastructure allowed the prototyping of an accurate model of the final detector system, which validates the configuration chain between the MPSoC ARM core and the front-end.

Moreover, the employed technologies ensure complete portability of the slow-control block from the testing to the final systems. In particular, by sharing a standardized Linux framework and AXI interfaces, the software drivers that were developed to access the slow-control buffers and registers are completely portable and can be used in the final detector. Similarly, the slow-control block is compatible with the back-

end FPGA of the final system as they are both deployed on Xilinx Ultrascale+ technology. The abstraction provided by the AXI interface not only facilitates the development of the MPSoC software but also eases the implementation of further improvements and adjustments to the FPGA hardware modules.

VI. CONCLUSION

Each ATCA board on the HGCAL back-end system is responsible for configuring and monitoring up to ≈ 3500 radiation-tolerant ASICs on the front-end. This is performed through the slow-control block that interfaces up to 756 transceiver ASICs, lpGBTs and GBT-SCAs, which house I2C masters that then communicate with the remaining front-end ASICs.

Although the final ATCA board and corresponding back-end FPGA are still under development and not available for testing, the infrastructure described in this work allowed for the validation of the functionality of the whole system by modelling an accurate interface between the slow-control block, the software part running at the MPSoC, and the front-end.

Despite its reduced size and simplicity when compared to the final infrastructure, this test system was able to provide a reliable validation of the communication protocol with the front-end ASICs. This allowed the simultaneous development of the final system software and FPGA hardware, and their use in other test systems that are presently in use. The shared technology between the MPSoC of the back-end board and the final system back-end FPGA also ensures full portability between the platforms used by the test and final detector system.

VII. FUTURE WORK

The presented prototyping and validation of the communication infrastructure between the slow-control block at the back-end and both the lpGBT and GBT-SCA at the front-end allows not only the integration of the slow-control block in other HGCAL test systems but also the expansion of the current test system by expanding its scope vertically (by adding additional functionality and features of either the front-end or the back-end domain) and horizontally (by adding already existing components to increase the size of the prototype). In both cases, the hardware can be upgraded to use the final detector components, either the back-end FPGA hardware or the front-end ASICs. Both can be done in parallel, using different test systems.

The front-end expansion will involve the establishment of more communication channels with the back-end by adding more ASICs. It will also involve the replacement of the VLDB+ and VLDB boards with custom HGCAL hardware parts, targeted for the final detector. This process should be straightforward for the slow-control, since the communication is already validated with both lpGBTs and GBT-SCAs, only requiring that more channels be instantiated. Moreover, any further vertical integration of I2C targets on the test systems

does not require the slow-control block to change because the interface with such ASICs is done via the lpGBTs or GBT-SCAs.

In the back-end domain, the FPGA hardware has to be updated with more back-end functionalities and, upon availability, the ZCU102 board will be replaced by the final ATCA detector board, based on a VU13P FPGA. These back-end migrations are also straightforward from the perspective of the slow-control block due to its modular development and shared technology with the VU13P FPGA. The involved software also does not need any significant changes since the Linux platform is the same, as well as the AXI connection from the FPGA fabric to the ARM core in the MPSoC.

ACKNOWLEDGEMENTS

This work was supported by national funds through Fundação para a Ciência e a Tecnologia (FCT), under projects UIDB/50021/2020 (MR, PT, NR); by Instituto Superior Técnico, under project 1018P.04875.1.01; and it was undertaken during a technical studentship at CERN (MR).

REFERENCES

- [1] CMS Collaboration, "The Phase-2 Upgrade of the CMS Endcap Calorimeter," CERN, Geneva, Tech. Rep., Nov 2017. [Online]. Available: <https://cds.cern.ch/record/2293646>
- [2] 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>
- [3] *ZCU102 Evaluation Board User Guide*, Xilinx. [Online]. Available: <https://docs.xilinx.com/v/u/en-US/ug1182-zcu102-eval-bd>
- [4] ARM, "AMBA® AXI™ and ACE™ Protocol Specification," AXI4 protocol specifications from ARM. [Online]. Available: <https://developer.arm.com/documentation/dhi0022/e/AMBA-AXI3-and-AXI4-Protocol-Specification>
- [5] *AXI Chip2Chip v5.0 LogiCORE IP Product Guide*, Xilinx. [Online]. Available: <https://docs.xilinx.com/r/en-US/pg067-axi-chip2chip/AXI-Chip2Chip-v5.0-LogiCORE-IP-Product-Guide>
- [6] P. Moreira, S. Baron, S. Biereigel, J. Carvalho, B. Faes, M. Firlej, T. Fiutowski, J. Fonseca, R. Francisco, D. Gong, N. Guettouche, P. Gui, D. Guo, D. Hernandez, M. Idzik, I. Kremastiotis, T. Kugathasan, S. Kulis, P. Leita, P. Leroux, E. Mendes, J. Mendez, J. Moron, N. Paulino, D. Porret, J. Prinzie, A. Pulli, Q. Sun, K. Swientek, K. Wyllie, D. Yang, J. Ye, T. Zhang, and W. Zhou, *lpGBT documentation: release*, Mar 2022. [Online]. Available: <https://cds.cern.ch/record/2809058>
- [7] A. Caratelli, S. Bonacini, K. Kloukinas, A. Marchioro, P. Moreira, R. De Oliveira, and C. Paillard, "The GBT-SCA, a radiation tolerant ASIC for detector control and monitoring applications in HEP experiments," *JINST*, vol. 10, p. C03034, 2015. [Online]. Available: <https://cds.cern.ch/record/2158969>
- [8] D. Montesinos, S. Baron, N. Guettouche, and J. Mendez, "The versatile link+ demo board (VLDB+)," *Journal of Instrumentation*, vol. 17, no. 03, p. C03032, mar 2022. [Online]. Available: <https://doi.org/10.1088/1748-0221/17/03/c03032>
- [9] R. Martin Lesma, F. Alessio, J. Barbosa, S. Baron, C. Caplan, P. Leita, C. Pecoraro, D. Porret, and K. Wyllie, "The Versatile Link Demo Board (VLDB)," *JINST*, vol. 12, p. C02020. 12 p, 2017. [Online]. Available: <https://cds.cern.ch/record/2275133>

