

Design and FPGA Implementation of the Barrel Calorimeter Trigger Algorithms for the High Luminosity LHC

DISSERTATION SUBMITTED IN PARTIAL FULFILMENT OF THE
ACADEMIC REQUIREMENTS
FOR THE AWARD OF THE DEGREE OF

Master of Technology

Microelectronics and VLSI Design

by

Abhinav Mohan

(Reg. No. 23PMMT11)



Center for Advanced Studies in Electronics Science and Technology

School of Physics

UNIVERSITY OF HYDERABAD

Hyderabad - 500046, India

June, 2025



Certificate

This is to certify that the dissertation entitled “**Design and FPGA Implementation of the Barrel Calorimeter Trigger Algorithms for the High Luminosity LHC**”, submitted by **Abhinav Mohan** to the University of Hyderabad, for the award of the degree of **Master of Technology in Microelectronics and VLSI Design**, is a record of the original, bona fide research work carried out by him under our supervision and guidance.

The results contained in this dissertation have not been submitted in part or in full to any other University or Institute for the award of any degree or diploma.

B. Gomer
26/6/2025

Dr. Bhawna Gomber
(Supervisor)
CASEST,
School of Physics
Assistant Professor
CASEST, School of Physics
University of Hyderabad
Hyderabad-500046, Telangana.

Samrat Sabat
30.06.2025

Prof. Samrat Sabat
(Head of Department)
CASEST,
School of Physics

S. PK
30/6/25

Prof. Suresh PK
Dean, School of Physics

संकाय अध्यक्ष / Dean
भौतिकी संकाय / School of Physics
हैदराबाद विश्वविद्यालय
UNIVERSITY OF HYDERABAD
हैदराबाद - 500 046, भारत / INDIA.

Declaration

I declare that this written submission represents my ideas in my own words. Where others' ideas and words have been included, I have adequately cited and referenced the original source. I declare that I have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated, or falsified any idea/data/fact/source in my submission.

A handwritten signature in black ink, appearing to read 'Abhinav', is written over a horizontal dotted line.

Abhinav Mohan

Roll No.: 23PMMT11

Date: June 2025

Place: University of Hyderabad

Acknowledgements

The journey to this thesis has been a remarkable expedition. It is a privilege to express my deepest gratitude to the incredible individuals who guided, supported, and inspired me along the way.

My foremost gratitude goes to my supervisor, Dr. **Bhawna Gomber**. It was her faith in my abilities that opened the door to this incredible opportunity. Her mentorship provided both the freedom to explore and the guidance to stay on course. Her passion and strong work ethic were and continues to be a constant source of motivation.

I extend my sincere thanks to Prof. **Sridhara Dasu** of the University of Wisconsin-Madison for arranging my per diem and integrating me into the Calo L1 group, making my stay both seamless and productive.

At CERN, I had the immense fortune of working under the direct supervision of Dr. **Alexander 'Sascha' Savin**. Sascha's invaluable experience and calm, insightful guidance were a beacon in the complex landscape of this project. His mentorship was crucial in navigating the daily challenges and turning them into learning opportunities.

I also extend my thanks to Dr. **Varun Sharma** and **Ales Svetek**, senior scientist and senior engineer at UW Madison for their guidance and support.

This entire endeavor would not have been possible without the foundational support of my home institution. I am grateful to Prof. **Ghanshyam Krishna**, Director of the Institute of Eminence (IoE) at the University of Hyderabad, and the IoE for their support in covering my travel expenses.

My thanks also extend to Prof. **K.C. James Raju** the then Dean, School of Physics, and all the faculty and support staff at the Centre for Advanced Studies in Electronics Science and Technology (CASEST) for fostering an environment of learning and academic excellence.

Abhinav Mohan

Abstract

This thesis presents the design, implementation, and validation of the Regional and Global Calorimeter Trigger (RCT and GCT) firmware for the CMS Phase-2 Level-1 Calorimeter Trigger system. The RCT firmware was developed with a new 17x6 architecture, incorporating High-Level Synthesis (HLS) IP cores for processing ECAL and HCAL inputs, including cluster finding, shower shape determination, and energy calibration. Real-time, software-controlled threshold calibration was implemented to allow e/γ identification without reprogramming the FPGA.

For the GCT Barrel, IP cores were developed to process aggregated RCT data and compute objects such as Particle Flow clusters, jets, taus, and energy sums. A Time Multiplexing (TMUX) scheme was introduced for the TMUX18-based GCT barrel architecture that has superior interconnect efficiency. Both RCT and GCT designs were optimized for 360 MHz operation with successful timing closure and efficient resource utilization.

The development workflow was improved through automated physical optimization techniques, incremental synthesis, and modular Out-of-Context synthesis flows. A virtual system-level validation environment with FEAST was employed, along with a suite of custom GUI-based tools to automate test vector generation, floorplanning, and MGT configuration. The final integrated system achieved an end-to-end processing latency of 1.01 μs , well within the 12.5 μs CMS Level-1 trigger budget,

Contents

Certificate

Declaration

Acknowledgements

Abstract

Contents

List of Figures

List of Tables

Abbreviations

Symbols

1	Introduction	1
1.1	The Large Hadron Collider	1
1.1.1	The CMS Detector	1
1.1.2	CMS Coordinate System	2
1.1.3	Detector Layout	3
1.1.4	The CMS Trigger System	6
1.1.4.1	Limitations of the Phase-1 Level 1 Trigger	7
1.1.4.2	The Phase-2 Upgrade	9
2	Design of the Regional and Global Calorimeter Triggers	11
2.1	Phase-2 Calorimeter Trigger Architecture	11
2.2	Hardware Platform	13
2.3	Regional Calorimeter Trigger (RCT) Design	14
2.3.1	RCT 17×6 Architecture	15
2.3.2	Design Requirements	17
2.3.3	Threshold Calibration	19

2.3.4	RCT 17×6 Algorithms	22
2.3.4.1	IP1 (5×6 Processing Core)	22
2.3.4.2	IP1 (2×6 Processing Core)	24
2.3.4.3	IP21 (Cluster Aggregation, Stitching, Sorting, and Calibration)	25
2.3.4.4	IP22 (Energy Reintegration)	27
2.3.4.5	IP3 (HCAL Integration and Output Packing)	27
2.4	Global Calorimeter Trigger (GCT) Design	29
2.4.1	GCT Barrel Architecture	30
2.4.2	Design Requirements	32
2.4.2.1	Time Multiplexing Scheme	33
2.4.3	GCT Barrel Algorithms	34
2.4.3.1	IP1 Algorithm (processing data for Correlator)	35
2.4.3.2	IP2 Algorithm (processing data for GCT Sum)	37
3	Implementation	40
3.1	Regional Calorimeter Trigger (RCT) Implementation	40
3.1.1	Threshold Calibration	41
3.1.2	Floorplanning	42
3.1.3	Timing Closure and Resource Utilization	43
3.2	Global Calorimeter Trigger (GCT) Implementation	46
3.2.1	Time Multiplexing (TMUX) Implementation	46
3.2.1.1	TMUX6	47
3.2.1.2	TMUX18	49
3.2.2	TMUX6-based GCT Barrel	51
3.2.2.1	Floorplan (TMUX6 GCT)	52
3.2.3	TMUX18-based GCT Barrel (Final Design)	53
3.2.3.1	Floorplan (TMUX18 GCT)	53
3.2.3.2	Timing Closure and Utilization (TMUX18 GCT)	53
3.3	Optimizations to the Vivado Implementation Flow	55
3.3.1	Physical Optimization (PhysOpt) Looping	55
3.3.2	Incremental Synthesis and Implementation	57
3.3.3	Out-of-Context (OOC) Synthesis for HLS IP Cores	58
4	Testing and Validation	60
4.1	Methodology	60
4.2	Standalone Single-Board Hardware Tests	61
4.2.1	RCT Standalone Tests	62
4.2.2	GCT Barrel Standalone Tests	63
4.3	Virtual Multi-Board System Test using FEAST	64
4.3.1	RCT + GCT System Tests using FEAST	67
4.4	Testing Summary	70

5	Conclusion and Future Work	71
5.1	Summary of Contributions	71
5.2	Future Developments	73
A	Automation Tools	76
A.1	CHEF: Configuration Helper for Easy FEASTing	76
A.2	MGT Mapping Utility	78
A.3	Visual Floorplanning Constraint Generator	79
A.4	Test Vector Generation and Formatting Tool	80
	Bibliography	82
	List of Publications	85

List of Figures

1.2	Schematic layout of the Compact Muon Solenoid (CMS) detector, showing its various sub-detector systems arranged in concentric layers around the interaction point.	4
1.3	Overview of the Phase-2 Calorimeter Trigger subsystems. The dotted rectangle indicates the RCT and GCT subsystems relevant to this thesis, which are responsible for processing data from the barrel region[13].	10
2.1	High-level architecture of the Phase-2 Level-1 Calorimeter Trigger, showing data flow from detector back-ends through the RCT and GCT stages for barrel, endcap, and forward regions.	12
2.2	The APx-F 'Falcon' Board: An ATCA-based platform utilizing the Xilinx VU13P-2 FPGA.[12]	14
2.3	The $17\eta \times 6\phi$ tower region processed by each RCT card. It generates four output links: three carrying tower information and one carrying information for the top nine e/γ clusters.	16
2.4	Overview of the RCT system, where 24 RCT cards cover the entire ECAL barrel region of $34\eta \times 72\phi$. Each RCT's $17\eta \times 6\phi$ region is processed in four logical parts, mapped across the four SLRs within the VU13P FPGA.	16
2.5	(a) Usable area for algorithm logic on the FPGA, with the red region reserved for DAQ firmware and the blue region for the firmware (FW) shell. (b) The conceptual floorplan of the initial RCT 17×6 architecture that failed to meet the DAQ floorplan constraint. (c) An approximate representation of the floorplan after integrating the DAQ firmware alongside the FW shell.	18
2.6	Block diagram of the initial RCT 17×6 design, illustrating the data flow between IP1 (sub-region processing), IP2 (stitching and sorting), and IP3 (HCAL processing and output packing) across SLR0, SLR1, and SLR2.	19
2.7	Revised architecture for the RCT 17×6 , incorporating changes to IP1, the split of IP2 into IP21 and IP22, and their relocation. This design addresses the DAQ floorplan constraints and other design requirements.	20

2.8	Illustration of the P_T -dependent shower shape threshold (red line) derived from Monte Carlo simulation studies. Points above this line represent e/γ candidates with good shower shape characteristics. This function is implemented in hardware using calibration constants. . . .	21
2.9	Conceptual subdivision of a $5\eta \times 6\phi$ tower region's crystal data into smaller, 11×30 crystal regions for parallel processing within an IP1 (5×6) core. 3 e/γ clusters are extracted from each such crystal region.	24
2.10	Illustration of cross-boundary stitching principles for e/γ clusters. Stitching is applied both within IP1 cores (between their internal crystal processing blocks) and in IP21 (between the main 5×6 and 2×6 sub-regions).	24
2.11	Key properties and associated information for e/γ clusters, including core energy, shower shape variables ($E_{2 \times 5}$, $E_{5 \times 5}$), Bremsstrahlung flags, and subsequently, H/E ratio and isolation metrics.	25
2.12	High-level architecture of the Global Calorimeter Trigger (GCT) system, showcasing its constituent parts: HF/HGCal, GCT Barrel, and GCT Sum unit. (TMI6 giving 6×24 links out of GCT Barrel, red) [16]	29
2.13	GCT Barrel processing illustrating independent IP1 processing across 6 SLRs. IP1 and IP2 are shown, with IP1 outputs (red) being time-multiplexed before transmission.[16]	31
2.14	Representation of the barrel detector geometry, indicating regions processed by different trigger cards and illustrating the origin of data overlap requirements for seamless object reconstruction across boundaries.[16]	31
2.15	Conceptual GCT Barrel architecture based on the TMUX6 time-multiplexing scheme.	34
2.16	Conceptual GCT Barrel architecture based on the TMUX18 time-multiplexing scheme.	35
3.1	Mechanism for real-time updates of calibration constants. Software writes constants to memory-mapped registers via AXI4-Lite, which are then read by the IP21 core. The wrapper structure around the IP core, facilitating this interface and other connections, is also depicted.	42
3.2	Comparison of the target floorplan (left) and the achieved floorplan (right) for the RCT 17×6 design. The placement of IP cores respects the DAQ keep-out regions and SLR boundaries.	43
3.3	Summary of timing closure results and total FPGA resource utilization for the implemented RCT 17×6 design.	44
3.4	Per-SLR resource utilization breakdown for the RCT 17×6 design. SLR1 shows the highest CLB utilization at 54%, within the design target. [16]	45
3.5	Flowchart illustrating the operational logic of the TMUX6 module. .	47
3.6	Flowchart illustrating the operational logic of the TMUX18 module, highlighting its construction from TMUX9 instances and a delay line.	50

3.7	Output waveform from TMUX18 module, showing successive data bunches emerging from the output links.	51
3.8	Floorplan of the exploratory TMUX6-based GCT Barrel design. This partial layout includes two IP1 cores and eight TMUX6 instances and achieved timing closure.	52
3.9	Final achieved floorplan of the GCT Barrel design utilizing the TMUX18 scheme. This full design, including IP1, IP2, and TMUX18 modules, successfully closed timing.	54
3.10	Summary of timing closure results and total FPGA resource utilization for the final TMUX18-based GCT Barrel design.	54
3.11	Per-SLR resource utilization for the TMUX18-based GCT Barrel design. The maximum CLB usage is noted at 68% in the most utilized SLR.[16]	55
3.12	Flowchart illustrating the iterative physopt looping strategy employed in the Vivado flow. This involves repeated application of physical optimization steps, combined with timing analysis, to improve WNS before and after routing.[21]	57
4.1	Illustration of output matching of hex words across HLS, RTL and Bitfile levels for 1bx data for RCT.	63
4.2	Conceptual illustration of the FEAST environment, showcasing its ability to emulate a multi-FPGA system using as little as a single physical FPGA. Users define the system configuration, and FEAST sequentially evaluates each FPGA's role, managing data flow between virtualized boards [12].	66
4.3	CHEF visualization of the small-scale FEAST test: 8 virtual RCT cards providing input to 1 virtual GCT Barrel card [16].	68
4.4	CHEF visualization of the full-scale FEAST test: 24 virtual RCT cards providing input to 3 virtual GCT Barrel cards, emulating the entire barrel trigger system [16].	69
A.1	CHEF for FEAST [20]	77
A.2	MGT Configuration Utility [20]	79
A.3	Floorplanning Tool for VU13P-2 [20]	80
A.4	Test vector generation tool [20]	81

List of Tables

2.1	Latency of RCT HLS IP cores. Total Latency of RCT processing = 0.572 μs	28
2.2	Latency of GCT Barrel HLS IP cores. Total GCT Barrel Latency = 0.444 μs	39

Abbreviations

CMS	C ompact M uon S olenoid
LHC	L arge H adron C ollider
RCT	R egional C alorimeter T rigger
GCT	G lobal C alorimeter T rigger
FPGA	F ield- P rogrammable G ate A rray
MGT	M ulti- G igabit T ransceiver
CHEF	C onfiguration H elper for E asy FEAST ing
FEAST	F PGA E nvironment for A lgorithm S lice T ests

Symbols

η	Pseudorapidity
ϕ	Azimuthal angle
E_T	Transverse energy

Chapter 1

Introduction

1.1 The Large Hadron Collider

The Large Hadron Collider (LHC), located at CERN, is the world's foremost particle accelerator, designed to collide beams of protons or heavy ions at unprecedented energies. Its primary mission is to probe the fundamental structure of matter and the forces governing the universe.^[1] The Compact Muon Solenoid (CMS) experiment is one of the two large, general-purpose detectors situated at the LHC's interaction points. CMS is engineered to observe and record the outcomes of these high-energy collisions, enabling a broad spectrum of physics research, from precision measurements of the Standard Model, including detailed studies of the Higgs boson, to searches for new physics phenomena such as supersymmetry, extra dimensions, and dark matter candidates.^[2]

1.1.1 The CMS Detector

The CMS detector employs a complex, multi-layered design centered around a powerful 3.8 Tesla superconducting solenoid magnet. Each layer, or sub-detector, is

specialized to identify and measure the properties of different types of particles emerging from the collisions. Understanding how particles interact with matter is fundamental to this design:

- Charged particles traversing the detector leave measurable trajectories (tracks).
- Electrons and photons deposit their energy entirely within the Electromagnetic Calorimeter (ECAL).
- Hadrons (particles composed of quarks, like protons and neutrons) deposit most of their energy in the Hadronic Calorimeter (HCAL).
- Muons, being highly penetrating, traverse most detector material and are identified by the outermost muon system.
- Neutrinos interact very weakly and escape direct detection, but their presence is inferred from an imbalance in the measured transverse energy of an event.

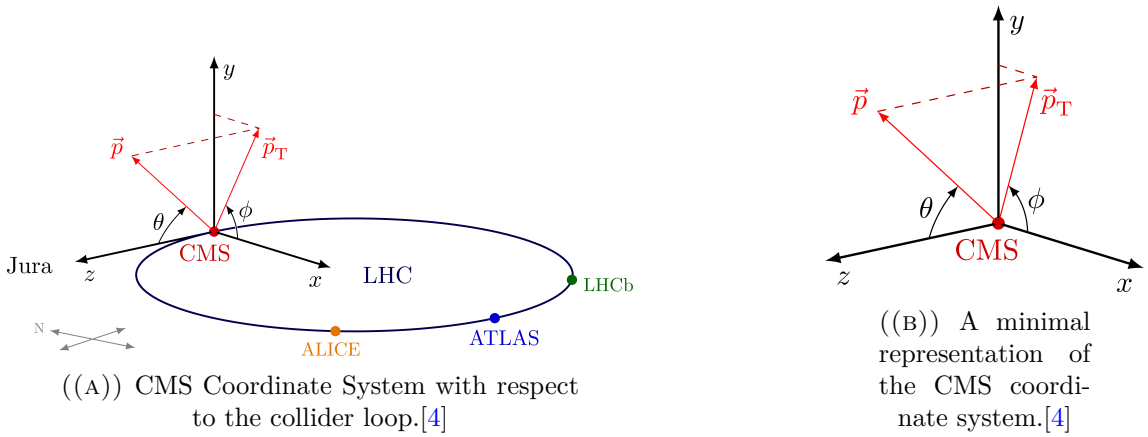
A schematic overview of the CMS detector layout is presented in Figure 1.2 [3].

1.1.2 CMS Coordinate System

The CMS experiment utilizes a right-handed Cartesian coordinate system with its origin defined at the nominal interaction point within the detector[4]. The positive y-axis points vertically upwards, while the positive x-axis is directed radially inward towards the center of the LHC ring. Consequently, the z-axis is aligned with the beam direction, pointing counter-clockwise along the collider loop.

In this framework, the azimuthal angle, ϕ , is measured from the x-axis in the x-y plane, which is transverse to the beam. The polar angle, θ , is measured from the

positive z-axis. An important kinematic variable derived from the polar angle is the pseudorapidity, defined as $\eta = -\ln[\tan(\theta/2)]$. Pseudorapidity is a preferred coordinate in high-energy physics as differences in η are invariant under Lorentz boosts along the z-axis. Physical quantities are often projected onto the transverse plane, such as the transverse momentum (p_T) and transverse energy (E_T), which are calculated from their respective x and y components. This coordinate system is used for reconstructing particle trajectories and analyzing collision event kinematics.



1.1.3 Detector Layout

The CMS detector is a cylindrical apparatus, approximately 21.6 meters long and 14.6 meters in diameter, with a total mass of around 12,500 tonnes. Its design follows a "hermetic" philosophy, aiming to detect and measure nearly all stable particles emerging from the high-energy proton-proton collisions. The detector is composed of several concentric sub-detector systems, each specialized for a particular task.

- **Inner Tracking System:** The innermost component of CMS, the tracker, is designed to provide highly efficient and precise measurements of charged particle trajectories. It is constructed entirely from silicon sensors and is the largest of its kind, with an active area of nearly 200 m². It consists of a pixel

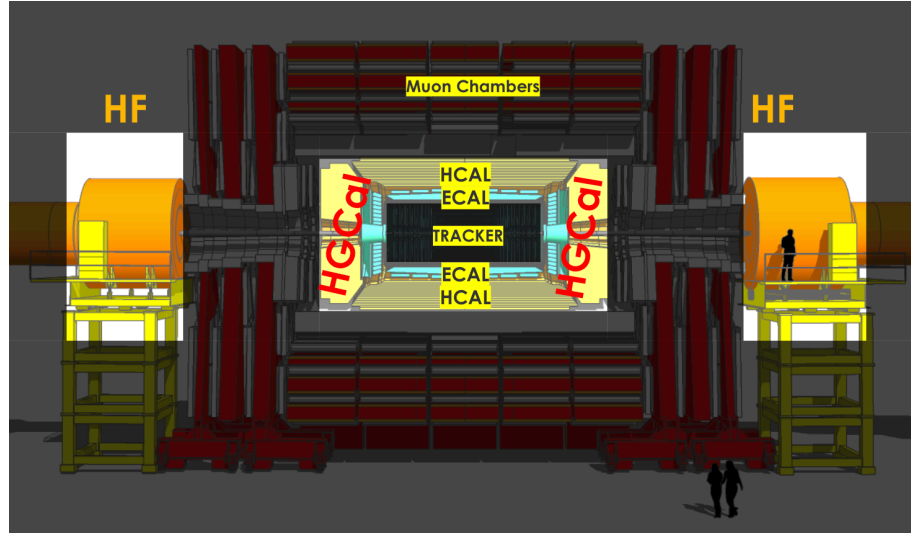


FIGURE 1.2: Schematic layout of the Compact Muon Solenoid (CMS) detector, showing its various sub-detector systems arranged in concentric layers around the interaction point.

detector with three barrel layers at radii between 4.4 cm and 10.2 cm, and an outer strip tracker with ten barrel layers extending to a radius of 1.1 m. This barrel structure is complemented by endcap disks to ensure coverage up to $|\eta| < 2.5$. The high granularity of the pixel detector is crucial for reconstructing the primary collision vertex and identifying secondary vertices from decaying particles [5]. The tracker's design balances the need for precise measurements against the requirement to minimize the material budget, thereby reducing effects like bremsstrahlung and photon conversions that can degrade subsequent energy measurements in the calorimeters.

- **Calorimeter System:** Located outside the tracker, this system is designed to absorb particles and measure their energy. It is composed of:
 - **Electromagnetic Calorimeter (ECAL):** A homogeneous, hermetic calorimeter responsible for the precise energy measurement of electrons and photons. It is constructed from 75,848 lead tungstate (PbWO_4) scintillating crystals. This material was chosen for its high density (8.28

g/cm³), short radiation length (0.89 cm), and small Molière radius (2.2 cm), which allow for a compact and fine-grained detector. The scintillation light is detected by Avalanche Photodiodes (APDs) in the barrel region (EB, $|\eta| < 1.479$) and Vacuum Phototriodes (VPTs) in the endcaps (EE, $1.479 < |\eta| < 3.0$). A preshower detector, consisting of lead absorbers and silicon strips, is placed in front of the endcaps to help distinguish single high-energy photons from pairs of lower-energy photons originating from π^0 decays.[6]

- **Hadronic Calorimeter (HCAL):** Surrounding the ECAL, the HCAL is a sampling calorimeter that measures the energy of hadrons (particles composed of quarks and gluons). It uses layers of brass (an alloy of 70% copper and 30% zinc) or steel as the absorber material and plastic scintillators as the active medium. Light from the scintillators is collected by wavelength-shifting fibers and read out by photodetectors. The HCAL is divided into a barrel section (HB), endcaps (HE), an outer section (HO) situated outside the solenoid, and a forward component (HF) that extends coverage to $|\eta| \approx 5.2$. Its granularity is designed to match that of the ECAL, with typical tower sizes of $\Delta\eta \times \Delta\phi = 0.087 \times 0.087$.[8]
- **High Granularity Calorimeter (HGCAL):** As part of the Phase-2 upgrade for the High-Luminosity LHC (HL-LHC), the HGCAL is a novel endcap calorimeter system set to replace the existing EE and HE detectors. It is engineered to perform robustly in the future high-radiation and high-pileup environment.[7] The HGCAL will offer unprecedented three-dimensional granularity for both electromagnetic and hadronic showers, enabling advanced techniques like particle-flow reconstruction and providing precise timing information to mitigate the effects of multiple overlapping collisions.

- **Superconducting Solenoid:** At the heart of the CMS detector lies a large superconducting solenoid, 13 meters in length and 6 meters in inner diameter. It generates a powerful and uniform 4 Tesla magnetic field in the central region. This field is essential for momentum measurement, as it bends the paths of charged particles traversing the inner tracker. The momentum of a particle is determined by measuring the radius of curvature of its track. The solenoid's large bore encloses the entire tracking system and both the electromagnetic and hadronic calorimeters.
- **Muon System:** The outermost layer of the CMS detector is the muon system, which is embedded within the large steel return yoke of the magnet. Muons are penetrating particles that pass through the calorimeters, making them uniquely identifiable. The muon system's primary role is to identify these muons and provide a robust and independent measurement of their momentum. It employs three types of gaseous detectors over an area of approximately 25,000 m²: Drift Tubes (DT) in the barrel region, Cathode Strip Chambers (CSC) in the endcap regions where the particle flux is higher, and Resistive Plate Chambers (RPC) in both barrel and endcaps to provide fast and precise timing information for the trigger system.^[9]

1.1.4 The CMS Trigger System

The LHC generates proton-proton collisions at a design rate of 40 million bunch crossings per second. This rate far exceeds what can be stored or analyzed in full. Therefore, CMS experiment relies on a real-time decision-making system, the trigger, to select a small fraction of these events (approximately 1-2 kHz) for permanent storage and offline analysis. The system is structured in two primary levels to manage this data reduction challenge.

The first level, the **Level-1 (L1) Trigger**, is a hardware-based system built on custom-designed electronics, with FPGAs and ASICs as its central processing elements. In the current Phase-1 configuration, the L1 trigger processes coarse-granularity data from the calorimeters and muon systems while the full-granularity data is temporarily held in pipelined buffers in the front-end electronics. The processing is layered: regional processors receive trigger primitives (TPs) calorimeter energy deposits or muon track segments and reconstruct trigger objects like electrons, photons, muons, and jets. These are then passed to global triggers that sort all candidates and make the final L1 decision. For the calorimeter trigger, this involves a two-layer architecture of Calorimeter Trigger Processor (CTP7) and MP7 boards, both based on Virtex-7 FPGAs. This system must make a decision within a few microseconds, reducing the event rate from 40 MHz down to approximately 100 kHz.

The second level, the **High-Level Trigger (HLT)**, is a software-based system operating on a large computer farm. It receives the events accepted by the L1 trigger and performs a more detailed analysis using high-granularity information from all sub-detectors. By executing complex reconstruction algorithms, similar to those used in offline analysis, the HLT further refines the selection and reduces the rate to the final 1-2 kHz written to storage.

1.1.4.1 Limitations of the Phase-1 Level 1 Trigger

The High-Luminosity LHC (HL-LHC) program, scheduled to begin operations after 2030, will dramatically increase the instantaneous luminosity to a baseline of $5 \times 10^{34} \text{ cm}^{-2}\text{s}^{-1}$ and ultimately to $7.5 \times 10^{34} \text{ cm}^{-2}\text{s}^{-1}$. This seven-fold increase over the original LHC design luminosity is essential for studying rare physical phenomena and making precise measurements of particles like the Higgs boson. However, it

introduces a significant challenge: an extreme level of pileup, with an average of 140 to 200 simultaneous proton-proton interactions occurring in each bunch crossing.

This high-pileup environment exposes the fundamental limitations of the Phase-1 L1 trigger architecture:

- **Coarse Granularity and Pileup Contamination:** The Phase-1 L1 trigger relies on trigger towers with coarse spatial granularity. With up to 200 simultaneous interactions, energy deposits from pileup can easily be misidentified as or overlap with signatures from the primary interaction of interest. This makes it very difficult to isolate objects like electrons or photons and accurately measure transverse energies, leading to a high rate of fake triggers.
- **Lack of Tracking Information:** The Phase-1 L1 trigger has no access to data from the silicon tracker. This prevents it from distinguishing charged from neutral particles, associating energy deposits in the calorimeter with a specific interaction vertex, or using track momentum to sharpen trigger thresholds. This capability is crucial for rejecting pileup, which is distributed across many vertices.
- **Limited Algorithmic Complexity:** The processing latency of the Phase-1 system (a few microseconds) and the logic capacity of its Virtex-7 FPGAs restrict the complexity of the algorithms that can be implemented. More sophisticated techniques required for pileup mitigation, such as particle-flow-like reconstruction, are not feasible within these constraints.

Without a significant upgrade, the Phase-1 L1 trigger would be overwhelmed by the HL-LHC conditions, forcing either an unacceptably high trigger rate that saturates the HLT or an increase in energy thresholds that would compromise the experiment's sensitivity to key physics processes.

1.1.4.2 The Phase-2 Upgrade

To meet the challenges of the HL-LHC, a complete redesign of the L1 Trigger system is a central component of the CMS Phase-2 upgrade. This new system is engineered not only to survive the new environment but also to improve physics performance by leveraging technological advancements and a new architectural philosophy.

A key enabler for this upgrade is the decision to extend the L1 processing latency from approximately $3.8\ \mu\text{s}$ to **$12.5\ \mu\text{s}$** . This expanded time budget is not a compromise but an advantage, providing the necessary processing window to execute far more complex algorithms. Correspondingly, the L1 output rate will be increased from 100 kHz to **750 kHz**, providing the HLT with a richer event stream for its final selection.

The advantages of the Phase-2 L1 trigger are built on three pillars: advanced hardware, higher-granularity data, and improved algorithms. The system will be built using the Xilinx UltraScale+ VU13P, which offer more than double the logic resources and serial bandwidth of the Virtex-7 FPGAs used in Phase-1. This hardware upgrade is essential to handle the enormous input data bandwidth, projected to be around 75 Tbps from the calorimeter systems alone.

The upgraded L1 trigger will process data at its native, high granularity. For the barrel calorimeter, this means access to the energy of each individual crystal. This fine-grained information, combined with the extended latency and powerful FPGAs, allows for the implementation of advanced reconstruction algorithms directly in hardware. These include particle-flow-like techniques that correlate information across sub-detectors to build a more complete picture of the event, enabling superior object identification and pileup mitigation.

One of the most transformative aspects of the Phase-2 upgrade is the **incorporation of track information** from the upgraded Outer Tracker directly into the L1 decision. For the first time, L1 algorithms will be able to use high-precision track vectors to confirm and refine calorimeter-based objects, apply track-based isolation criteria, and crucially, link objects to the primary interaction vertex to reject pileup.

This thesis focuses on the design, implementation, and validation of key components of this upgraded Phase-2 L1 Calorimeter Trigger, specifically for the barrel region. As shown in Figure 1.3, this work concerns the Regional Calorimeter Trigger (RCT) and Global Calorimeter Trigger (GCT) subsystems, which are responsible for processing the high-granularity data from the barrel electromagnetic and hadronic calorimeters.

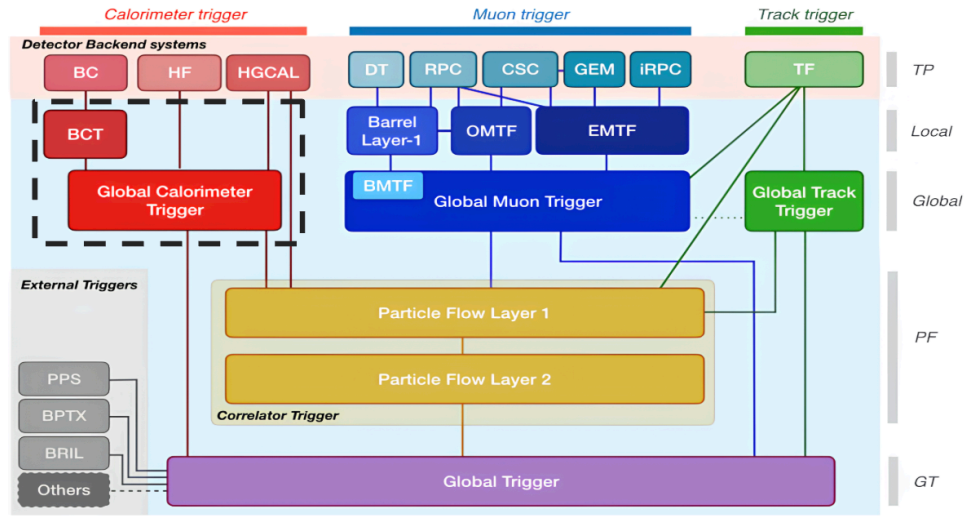


FIGURE 1.3: Overview of the Phase-2 Calorimeter Trigger subsystems. The dotted rectangle indicates the RCT and GCT subsystems relevant to this thesis, which are responsible for processing data from the barrel region[13].

Chapter 2

Design of the Regional and Global Calorimeter Triggers

2.1 Phase-2 Calorimeter Trigger Architecture

The Phase-2 L1 Calorimeter Trigger processes data from the upgraded barrel (ECAL/H-CAL) and new endcap (HGCAL) calorimeters. The system architecture, depicted in Figure 2.1, involves two primary processing stages: the Regional Calorimeter Trigger (RCT) and the Global Calorimeter Trigger (GCT) [11]. These stages are implemented on Xilinx Virtex UltraScale+ VU13P-2 FPGAs.

- **Regional Calorimeter Trigger (RCT):** The RCT system is responsible for the initial processing of barrel calorimeter data. Twenty-four identical RCT boards, each featuring a VU13P-2 FPGA, cover the entire barrel. Each board processes a specific geometric region ($17\eta \times 6\phi$ in trigger tower granularity), receiving 102 optical links with ECAL crystal data and 4 links with HCAL tower data. In total, 2544 optical links feed the RCT system. Each RCT

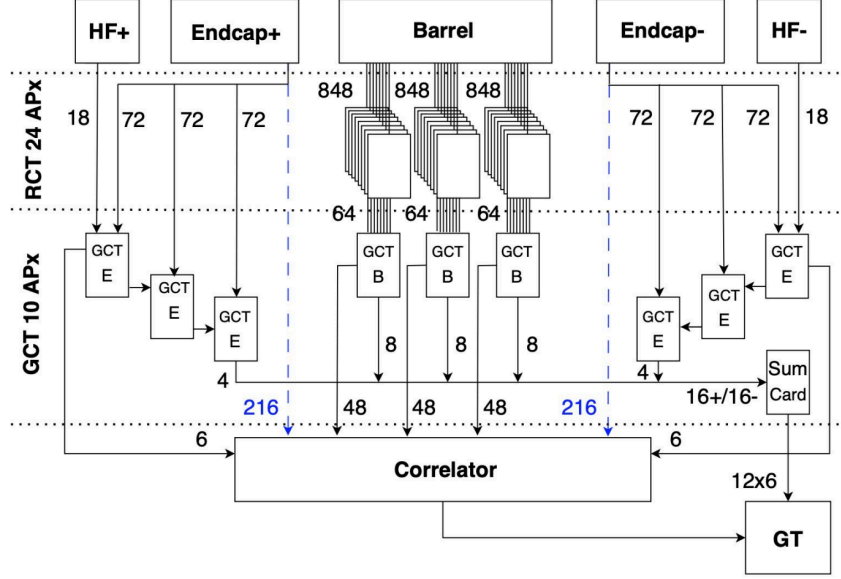


FIGURE 2.1: High-level architecture of the Phase-2 Level-1 Calorimeter Trigger, showing data flow from detector back-ends through the RCT and GCT stages for barrel, endcap, and forward regions.

board outputs 8 links carrying reconstructed e/γ objects and tower energy sums. This results in 192 links from the RCTs being sent to the GCT Barrel boards.

The RCT 17x6 design presented in this thesis refines a prior 17x4 architecture (P. Kumar et al.) [13] that utilized 3-SLR VU9P FPGAs, requiring a total of 36 devices. To reduce system cost and complexity, the present work migrates the design to the 4-SLR VU13P FPGA, lowering the requirement to 24 devices. This hardware change from a 3-SLR to a 4-SLR FPGA necessitated the redesign of the algorithms and architecture detailed herein.

- **Global Calorimeter Trigger (GCT):** The GCT system integrates information from all calorimeter regions (barrel, endcaps, forward).
 - **GCT Barrel:** Three boards receive data from the RCTs. They reconstruct higher-level objects like jets and taus for the barrel, refine e/γ and particle-flow cluster information, and compute barrel energy sums. They

also send detailed cluster information (via time multiplexed links) to the downstream Correlator Layer 1.

- **GCT Endcap/HF:** Six boards process data from the new HGCal (endcaps) and the Hadronic Forward (HF) calorimeter. They perform similar object reconstruction and energy sum calculations for these forward regions. Input data from the detector back-ends for these regions also utilizes time multiplexing (TMI18).
- **GCT Sum Card:** One board receives e/gamma and supertower information from both the GCT Barrel and GCT Endcap/HF boards. It performs jets, taus and energy sums calculations and prepares the final calorimeter trigger summary.

The GCT Sum card then transmits 12 copies of the global L1 calorimeter objects and energy sums to the Global Trigger (GT), which combines this with information from other L1 sub-triggers (muon, track) to make the final L1 Accept decision.

2.2 Hardware Platform

The computational intensity and I/O demands of the Phase-2 L1 trigger necessitate state-of-the-art FPGAs. The Xilinx Virtex UltraScale+ VU13P-2 is the selected device. It features four Super Logic Regions (SLRs), providing enough logic resources and Multi-Gigabit Transceivers (MGTs) for high-speed data links (25 Gbps per lane). SLRs 1-3 have 32 MGTs each, while SLR0 has 28 .

These FPGAs are hosted on custom-designed electronics boards like the APx-F 'Falcon' platform (Figure 2.2). The APx-F is an ATCA-based (Advanced Telecommunications Computing Architecture) board, ensuring robustness and standardization. It includes a Kria System-on-Module for board management (IPMC functions) and Samtec FireFly optical modules for high-density, high-speed optical I/O [12] [13].

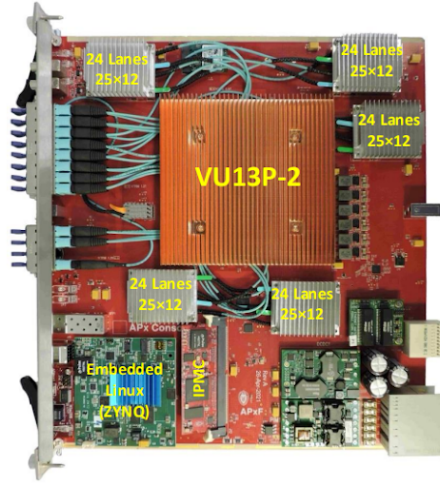


FIGURE 2.2: The APx-F 'Falcon' Board: An ATCA-based platform utilizing the Xilinx VU13P-2 FPGA.[12]

2.3 Regional Calorimeter Trigger (RCT) Design

The Regional Calorimeter Trigger (RCT) processes information from the Electromagnetic Calorimeter (ECAL) barrel backend cards, transforming low-level crystal data into higher-level electron/gamma (e/γ) objects. The system also incorporates Hadron Calorimeter (HCAL) data to aid in distinguishing hadronic interactions. Within this framework, clusters are formed from ECAL hits, their energies are calibrated, and a shower shape flag is computed for each. Any ECAL energy not assigned to clusters is accumulated per tower and calibrated. Similarly, HCAL energy deposits are summed by tower.

2.3.1 RCT 17×6 Architecture

The RCT constitutes the first-level trigger for the barrel calorimeter system. The entire detector barrel is processed by 24 APx boards, each tasked with an identical detector region spanning $17\eta \times 6\phi$ towers from ECAL and $16\eta \times 6\phi$ towers from HCAL (as depicted in Figure 2.3). This standardized regional division allows a single reference design for all 24 boards. Each RCT board employs a Xilinx Virtex UltraScale+ VU13P FPGA, which features four Super Logic Regions (SLRs) and provides sufficient capacity for the 106 input links (102 from ECAL and 4 from HCAL). The primary role of the RCT is the preparation of tower data and e/γ clusters for subsequent processing by the Global Calorimeter Trigger (GCT) barrel. A key design principle is the use of identical algorithms across all RCT boards, which simplifies system development and maintenance. Figure 2.4 illustrates how the 24 RCT cards cover the ECAL barrel and how each RCT region is processed across the FPGA's SLRs.

The VU13P FPGA has a significant number of Multi-Gigabit Transceivers (MGTs) distributed across its SLRs (32 MGTs per SLR). This MGT distribution is a critical factor in determining how algorithm processing is partitioned within the board. To manage the processing load and data flow, each $17\eta \times 6\phi$ region is divided into four sub-regions: three $5\eta \times 6\phi$ regions and one $2\eta \times 6\phi$ region. The $5\eta \times 6\phi$ sub-regions each utilize 30 input links, while the $2\eta \times 6\phi$ sub-region uses 12 input links. These initial processing stages, implemented as independent HLS IP cores, operate in parallel, with no direct inter-SLR communication required between them. The detailed functionality of these IPs is presented in Section 2.3.4. Subsequent to this sub-region processing, the tower and cluster information is forwarded to a second-stage IP (IP2 in the initial design, later IP21/IP22) for stitching across sub-region

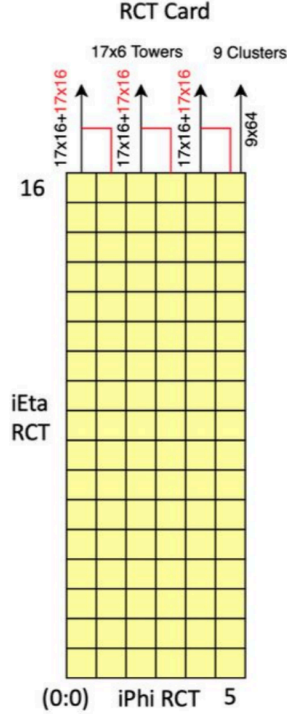


FIGURE 2.3: The $17\eta \times 6\phi$ tower region processed by each RCT card. It generates four output links: three carrying tower information and one carrying information for the top nine e/γ clusters.

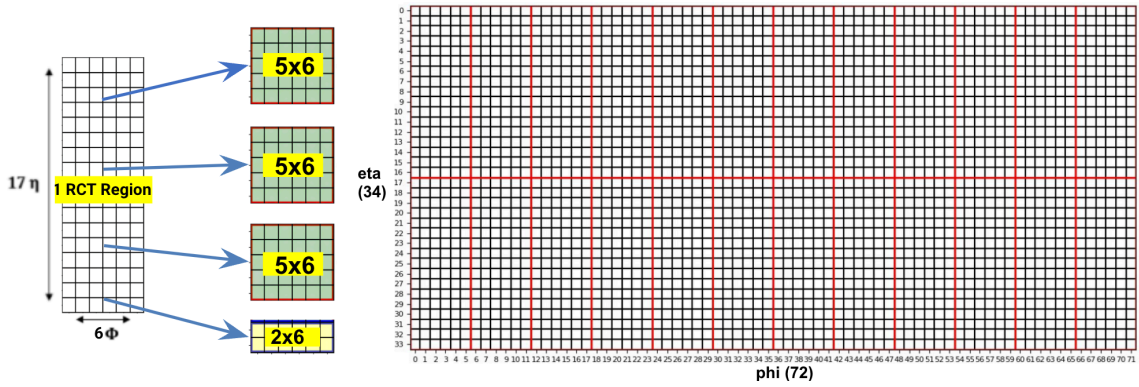


FIGURE 2.4: Overview of the RCT system, where 24 RCT cards cover the entire ECAL barrel region of $34\eta \times 72\phi$. Each RCT's $17\eta \times 6\phi$ region is processed in four logical parts, mapped across the four SLRs within the VU13P FPGA.

boundaries, cluster sorting, and threshold calibration. A final IP block (IP3) incorporates HCAL data to calculate H/E (hadronic energy / electromagnetic energy) for the top clusters and packs the information for transmission to the GCT.

2.3.2 Design Requirements

The firmware design for the RCT 17×6 system is governed by several major requirements:

1. **Functionality:** Accurate implementation of the trigger algorithms.
2. **Timing Closure and Resource Utilization:** The design must achieve timing closure at a clock frequency of 360 MHz. Resource utilization should not exceed 70% of the available resources within each SLR to ensure routability and accommodate future modifications. The total latency of RCT processing should be affordable within the total L1 trigger latency budget of $12.5 \mu\text{s}$.
3. **Floorplan Constraints:** Specific regions of the FPGA must be reserved. As illustrated in Figure 2.5(a), a perimeter of one clock region width must remain unutilized by the algorithm payload, separating it from the firmware shell. This area is designated for the Data Acquisition (DAQ) subsystem firmware.
4. **Reproducibility:** The design must demonstrate consistent timing closure, resource utilization, and floorplan across multiple implementation runs. A benchmark of achieving timing closure in 10 out of 10 builds is used to ensure design stability.

Functional equivalence must be maintained across C++ high-level synthesis models, RTL simulations, and hardware tests.

An initial version of the RCT 17×6 firmware (block diagram in Figure 2.6) encountered difficulties in satisfying the DAQ floorplan constraint. In this version, each initial processing IP (for 5×6 and 2×6 sub-regions) sent two links (one for clusters, one for towers) to a central IP2 block, where stitching and sorting occurred. Four

tower links and one sorted cluster link were then passed to IP3. IP3 integrated HCAL tower information from four HCAL input links, calculated H/E for sorted clusters, and packed data for the GCT. This architecture, whose floorplan is depicted in Figure 2.5(b), did not implement threshold calibration or the duplication of RCT outputs at that stage.

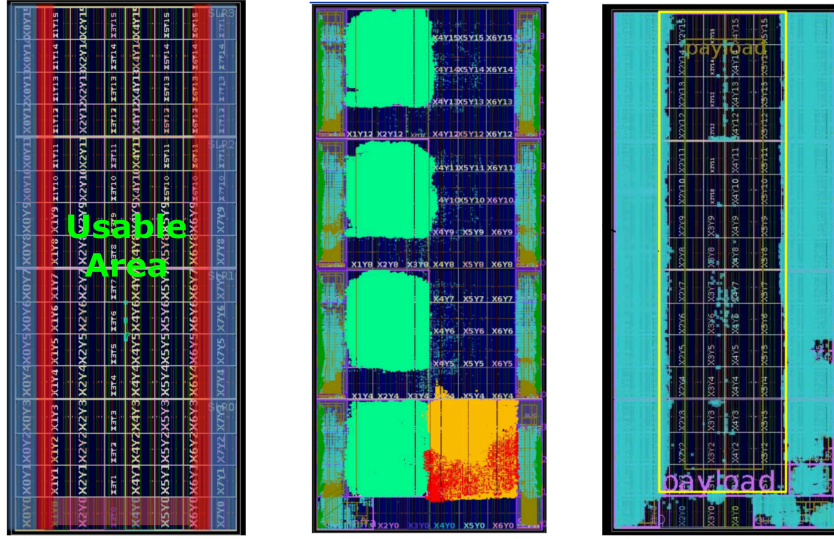


FIGURE 2.5: (a) Usable area for algorithm logic on the FPGA, with the red region reserved for DAQ firmware and the blue region for the firmware (FW) shell. (b) The conceptual floorplan of the initial RCT 17×6 architecture that failed to meet the DAQ floorplan constraint. (c) An approximate representation of the floorplan after integrating the DAQ firmware alongside the FW shell.

The strict DAQ floorplan requirement necessitated a revision of the initial RCT architecture. In the original design, each 5×6 and 2×6 IP core occupied nine clock regions; this had to be reduced to eight clock regions to create the necessary peripheral space. Moreover, the architecture of IP2, especially its placement and resource footprint within SLR0, required modification.

Considering these constraints, alongside the need for improved resource utilization and the implementation of previously deferred features, a new architecture for the RCT 17×6 was developed, as shown in Figure 2.7. Key changes included:

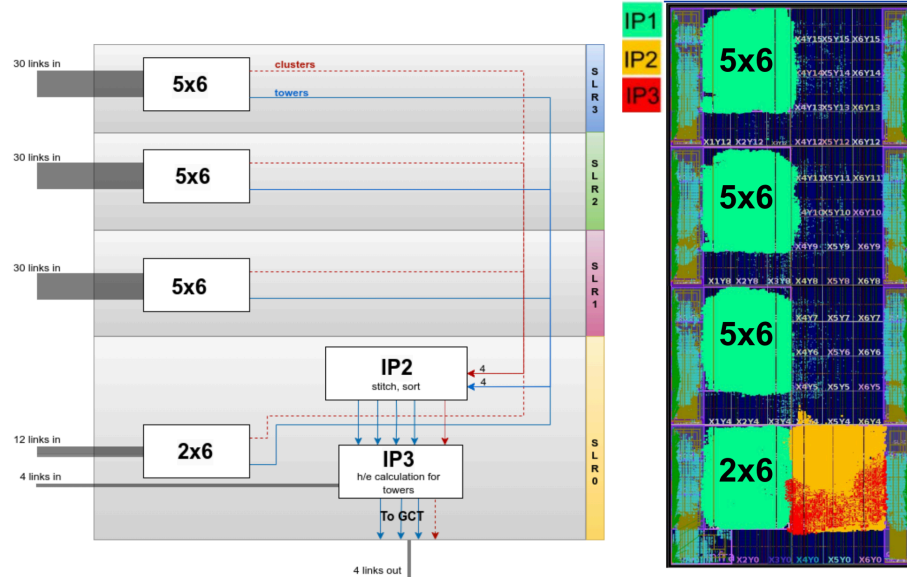


FIGURE 2.6: Block diagram of the initial RCT 17×6 design, illustrating the data flow between IP1 (sub-region processing), IP2 (stitching and sorting), and IP3 (HCAL processing and output packing) across SLR0, SLR1, and SLR2.

- Modifications to the IP1 C++ HLS code to reduce resource utilization.
- The functionality of IP2 was partitioned into two smaller IPs: IP21 and IP22. IP21 handles cluster stitching, sorting, and threshold calibration. IP22 re-integrates the energy of rejected clusters back into the corresponding towers.
- IP21 and IP22 were relocated to SLR1 to optimize the floorplan.
- Duplication of the IP3 outputs (to meet GCT overlap requirements) was implemented in the wrapper logic surrounding the core IPs.

2.3.3 Threshold Calibration

Efficient identification of electron and photon (e/γ) candidates within the CMS Electromagnetic Calorimeter (ECAL) is important for a wide range of physics analyses. This requires reliable methods to distinguish genuine electromagnetic signatures

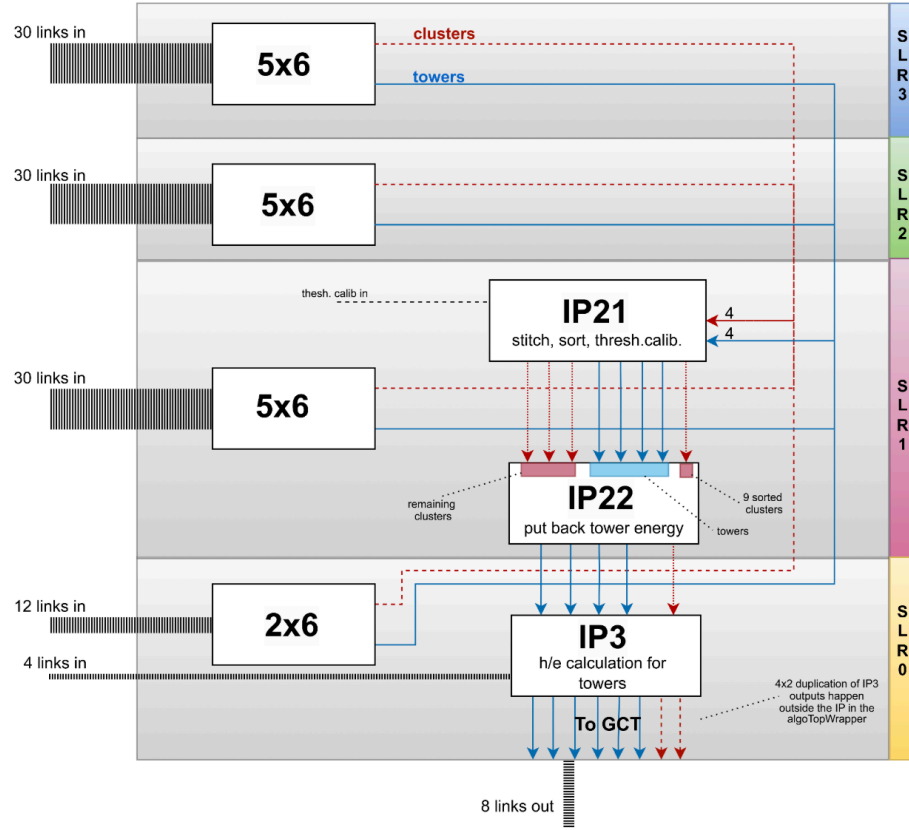


FIGURE 2.7: Revised architecture for the RCT 17×6 , incorporating changes to IP1, the split of IP2 into IP21 and IP22, and their relocation. This design addresses the DAQ floorplan constraints and other design requirements.

from backgrounds, primarily hadronic jets. The differing topological development of electromagnetic showers compared to hadronic interactions provides a basis for discrimination. Electromagnetic showers are typically narrow, depositing most of their energy within a compact region around the shower axis. In comparison, hadronic showers or overlapping photons from neutral pion decays ($\pi^0 \rightarrow \gamma\gamma$) tend to be broader, having a higher molière radius [14].

This characteristic difference is exploited by calculating a shower shape variable based on the energy distribution within ECAL crystals. A commonly used shower shape variable quantifies energy concentration by comparing the energy deposited in a narrow 2×5 crystal matrix centered on the most energetic crystal ($E_{2 \times 5}$) to the

energy in a wider 5×5 crystal matrix ($E_{5 \times 5}$) surrounding the same central crystal. This ratio, termed Shower Shape (SS), is defined as:

$$SS = \frac{E_{2 \times 5}}{E_{5 \times 5}} \quad (2.1)$$

Genuine electrons and photons typically exhibit SS values close to 1 due to their compact energy deposition, whereas background processes often yield lower SS values. However, the distribution of this SS variable, and therefore the optimal discriminating threshold, is not constant; it exhibits a dependence on the transverse momentum (P_T) of the calorimeter cluster. Simulation studies, using tools like GEANT4 [15], are performed for accurately modeling this P_T dependence for both signal and background processes.

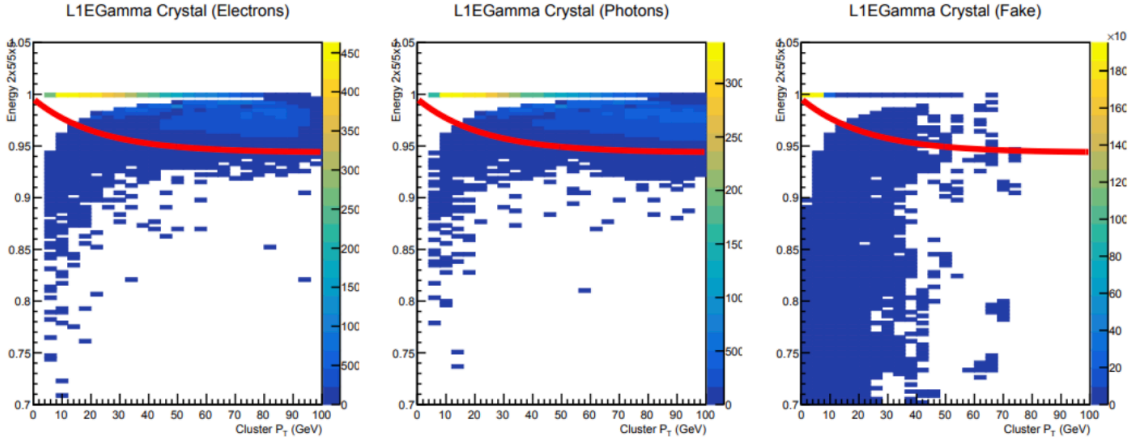


FIGURE 2.8: Illustration of the P_T -dependent shower shape threshold (red line) derived from Monte Carlo simulation studies. Points above this line represent e/γ candidates with good shower shape characteristics. This function is implemented in hardware using calibration constants.

To accommodate this variation, a P_T -dependent threshold calibration is performed. This establishes a functional relationship, often visualized as a curve in the SS versus Cluster P_T plane (as indicated by the red curve in Figure 2.8), which defines the minimum SS value required for a cluster to be considered an e/γ candidate at a given P_T . Clusters with an SS value *above* this dynamically determined threshold

are tagged as potential e/γ candidates, thereby improving selection purity across a broad momentum range. For practical implementation within the constraints of hardware trigger systems, this continuous threshold function is discretized. The relevant P_T range is divided into a number of bins (here, 25 bins), and a specific SS threshold value is determined and stored as a configurable constant for each bin. This approach allows straightforward adjustments to the e/γ identification sensitivity by updating these constants, foregoing the need for time-consuming firmware rebuilds of the RCT. This significantly reduces potential deadtime and enhances the system's online performance during data acquisition.

2.3.4 RCT 17×6 Algorithms

The $17\eta\times 6\phi$ tower region assigned to each RCT card is processed by dividing its ECAL crystal data into four logical sub-regions: three corresponding to $5\eta\times 6\phi$ tower areas and one corresponding to a $2\eta\times 6\phi$ tower area. The processing for each of these sub-regions is primarily handled by distinct High-Level Synthesis (HLS) IP cores, distributed across the FPGA's SLRs for maximum parallel operation. The overall data processing pipeline involves three instances of an IP1 (5×6) core, one instance of an IP1 (2×6) core, followed by an IP21 core for aggregation and calibration, an IP22 core for energy reintegration, and finally an IP3 core for HCAL data incorporation and output packing. The functionality of each IP stage is detailed below.

2.3.4.1 IP1 (5×6 Processing Core)

Each of the three IP1 (5×6) cores is responsible for processing crystal-level information from an ECAL area corresponding to a $5\eta\times 6\phi$ tower region.

1. **Input Data Processing:** The core receives ECAL crystal data for its designated $5\eta \times 6\phi$ area. To allow for maximum parallelism, this crystal data is further partitioned into smaller blocks (as depicted in Figure 2.9). These blocks are processed concurrently.
2. **Cluster Identification:** Within each internal processing block, a fixed maximum number of e/γ clusters are identified. This process involves:
 - Locating a high-energy "seed" crystal.
 - Forming an e/γ cluster around this seed (e.g., a 3×5 crystal matrix) and calculating its total energy by summing constituent crystal energies.
 - Computing shower shape characteristics, including energies in 2×5 ($E_{2 \times 5}$) and 5×5 ($E_{5 \times 5}$) crystal matrices for shower shape determination, as well as indicators for Bremsstrahlung activity (see Figure 2.11).
 - Masking the energies of crystals assigned to a found cluster to prevent their re-selection within the same processing block.
3. **Local Stitching and Selection:** After clusters are identified in all internal processing blocks, a local stitching procedure is applied. This step eliminates potential double-counting of clusters that might span the boundaries of these internal blocks, based on proximity criteria (boundary conditions is shown in Figure 2.10). From the resulting set of unique clusters, nine clusters per IP1 (5×6) core are obtained.
4. **Tower Energy Aggregation:** Energy not assigned to these clusters is aggregated on a per-tower basis for the $5\eta \times 6\phi$ region.
5. **Output Data Preparation:** Each IP1 (5×6) core sends its processed information via two links: one carrying the nine identified e/γ cluster objects, and the other carrying the ECAL tower data for its $5\eta \times 6\phi$ region.

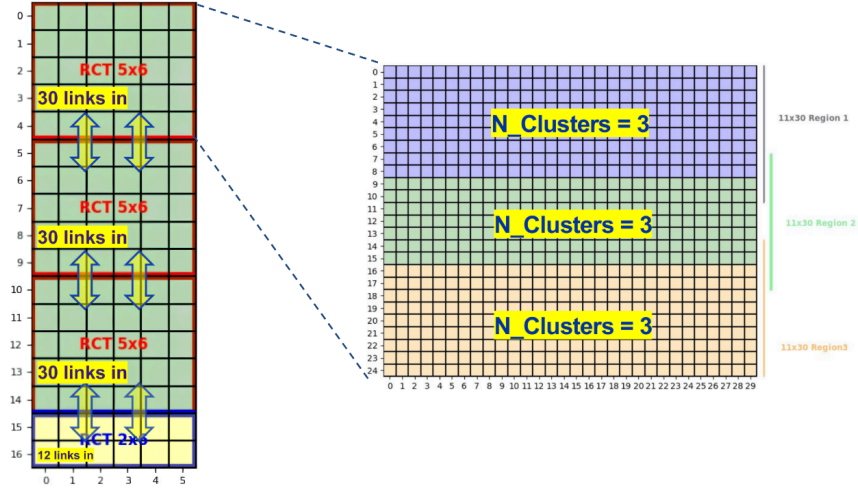


FIGURE 2.9: Conceptual subdivision of a $5\eta \times 6\phi$ tower region's crystal data into smaller, 11×30 crystal regions for parallel processing within an IP1 (5×6) core. 3 e/γ clusters are extracted from each such crystal region.

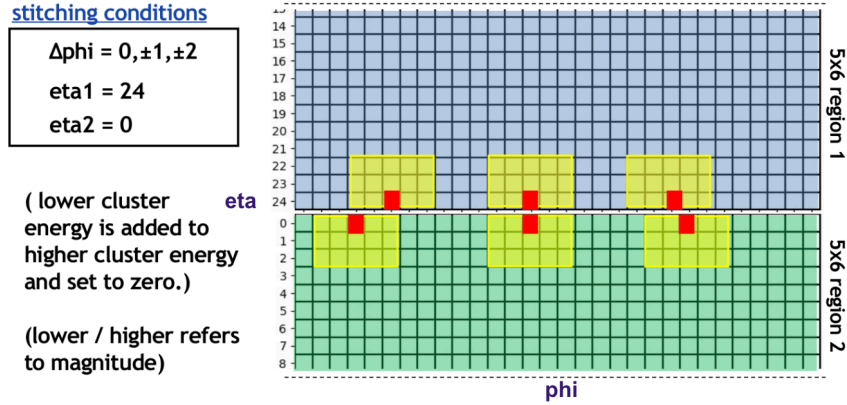


FIGURE 2.10: Illustration of cross-boundary stitching principles for e/γ clusters. Stitching is applied both within IP1 cores (between their internal crystal processing blocks) and in IP21 (between the main 5×6 and 2×6 sub-regions).

2.3.4.2 IP1 (2×6 Processing Core)

The IP1 (2×6) core processes crystal-level information from an ECAL area corresponding to a $2\eta \times 6\phi$ tower region.

1. **Input Data and Cluster Identification:** The core receives ECAL crystal data for its $2\eta \times 6\phi$ area. Given the smaller input region, internal sub-partitioning for cluster finding is not employed. Instead, 3 e/γ clusters are

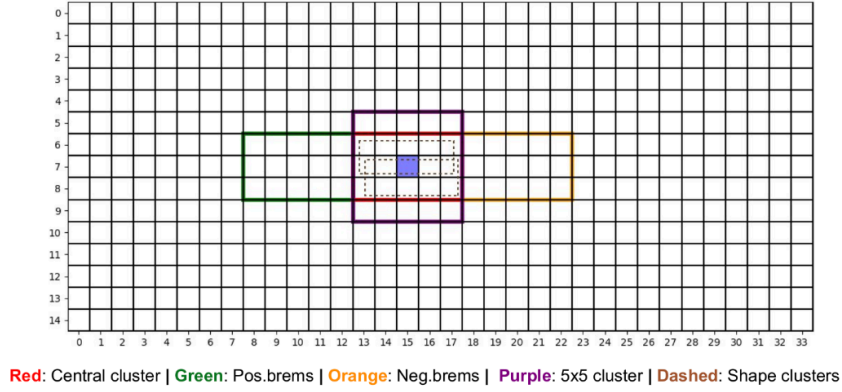


FIGURE 2.11: Key properties and associated information for e/γ clusters, including core energy, shower shape variables ($E_{2\times5}$, $E_{5\times5}$), Bremsstrahlung flags, and subsequently, H/E ratio and isolation metrics.

identified directly from the entire $2\eta \times 6\phi$ crystal data, using similar seed-finding and characterization logic as the 5×6 cores.

2. **Tower Energy Aggregation:** Energy not assigned to these clusters is aggregated on a per-tower basis for the $2\eta \times 6\phi$ region.
3. **Output Data Preparation:** The output structure mirrors that of the IP1 (5×6) cores, with one link dedicated to the three cluster objects and another to the ECAL tower data for its $2\eta \times 6\phi$ region.

Collectively, the three IP1 (5×6) cores and the single IP1 (2×6) core produce up to 30 e/γ clusters (27 from the 5×6 cores and 3 from the 2×6 core) and corresponding tower data for the full $17\eta \times 6\phi$ region processed by the RCT card.

2.3.4.3 IP21 (Cluster Aggregation, Stitching, Sorting, and Calibration)

The IP21 core serves as the central aggregation point for data from all four IP1 instances (three 5×6 and one 2×6). Its primary functions are to stitch clusters across IP1 boundaries, sort them, select the top candidates, and apply threshold calibration.

1. **Input Data Aggregation:** IP21 receives four links carrying cluster information (up to 30 clusters in total) and four links carrying the corresponding ECAL tower data from the IP1 cores.
2. **Inter-Region Stitching:** A stitching procedure is applied to clusters located near the boundaries of the four main sub-regions defined by the IP1 instances. This step merges energy from clusters that might be fragmented across these boundaries or eliminates any remaining duplicates (refer to Figure 2.10 for stitching conditions).
3. **Sorting and Selection:** The resulting set of unique e/γ clusters is then sorted by energy using an efficient, hardware-optimized sorting algorithm (bitonic-sort32). From this sorted list, the top nine highest-energy clusters are selected for further processing and eventually sending to the Global Calorimeter Trigger (GCT). Clusters are also tagged based on the originating SLR.
4. **Threshold Calibration:** These top nine clusters undergo the P_T -dependent shower shape threshold calibration, as detailed in Section 2.3.3. The shower shape parameters calculated in IP1 for each cluster are compared against P_T -binned thresholds (derived from the function shown in Figure 2.8), which are stored as configurable constants. This comparison assigns a quality flag to each cluster, indicating whether it passes the e/γ identification criteria.
5. **Output Data Preparation (for IP22):** IP21 generates several outputs:
 - One link carrying the top nine calibrated e/γ clusters.
 - Three links carrying data for the clusters that were not selected among the top nine (these are termed rejected clusters).
 - Four links passing through the original ECAL tower data received from the IP1 cores.

2.3.4.4 IP22 (Energy Reintegration)

The IP22 core is primarily responsible for energy conservation within the ECAL tower data.

1. **Input Data:** It receives the top nine selected clusters, the set of rejected clusters, and the complete ECAL tower data from IP21.
2. **Energy Reintegration:** The main function of IP22 is to add the energy of the rejected clusters back into the ECAL tower sums. Using the seed crystal position and originating sub-region identifier of each rejected cluster, its energy is added to the appropriate tower(s) in the corresponding tower data array.
3. **Output Data Preparation (for IP3):** The top nine e/γ clusters (which are passed through IP22 without modification) and the ECAL tower data, now updated with the reintegrated energy from rejected clusters, are forwarded to the IP3 stage.

2.3.4.5 IP3 (HCAL Integration and Output Packing)

The IP3 core finalizes the RCT objects by integrating Hadron Calorimeter (HCAL) information and preparing the data payload for the GCT.

1. **Input Data:** IP3 receives the top nine e/γ clusters and the updated ECAL tower data from IP22. Additionally, it ingests HCAL tower energy information from four dedicated HCAL input links, which correspond to the $16\eta \times 6\phi$ HCAL region geometrically mapped to the RCT card.
2. **HCAL Data Integration and H/E Calculation:** Within IP3, the ECAL and HCAL tower data are combined. For each calorimeter tower, an H/E

(Hadronic Calorimeter energy / Electromagnetic Calorimeter energy) ratio is calculated. This ratio, along with the sum of ECAL and HCAL energies, forms the updated tower information.

3. **Cluster H/E Assignment:** For the top nine e/γ clusters, an H/E value is also determined. This is done by associating each cluster with the H/E characteristics of the calorimeter tower(s) it predominantly occupies. This cluster-specific H/E metric is added to the data for each of the top nine clusters.
4. **Output Formatting and Packing:** Finally, IP3 formats and packs the processed e/γ cluster data and the combined ECAL+HCAL tower data into four output links, as shown in Figure 2.3. These consist of one link for the top nine e/γ clusters (now including H/E information and calibration flags) and three links for the tower data.
5. **Output Duplication (Wrapper Logic):** As a final step in the firmware, implemented in the wrapper logic external to these core IPs, these four output links are duplicated to eight. This duplication provides the necessary overlap required for processing by downstream GCT algorithms, which operate on data from adjacent RCT regions.

IP	Latency min (cycles)	Latency max (cycles)	Latency absolute	II (cycles)
IP1 5x6	131	131	0.364 μs	9
IP1 2x6	123	123	0.342 μs	9
IP21	48	48	0.133 μs	9
IP22	27	27	0.075 μs	9

TABLE 2.1: Latency of RCT HLS IP cores.
Total Latency of RCT processing = 0.572 μs .

2.4 Global Calorimeter Trigger (GCT) Design

The Global Calorimeter Trigger (GCT) system aggregates and processes information from the RCT barrel, the High Granularity Calorimeter (HGCAL), and the Hadronic Forward (HF) calorimeter. Its main role is to identify and sort high-level physics objects such as electrons/photons (e/γ), jets, taus, and to compute global energy sums for the Level-1 trigger decision. An overview of the GCT system architecture and its major components, is presented in Figure 2.12. GCT uses 10 APx boards, of these 3 are for the GCT Barrel logic, which is to be discussed here.

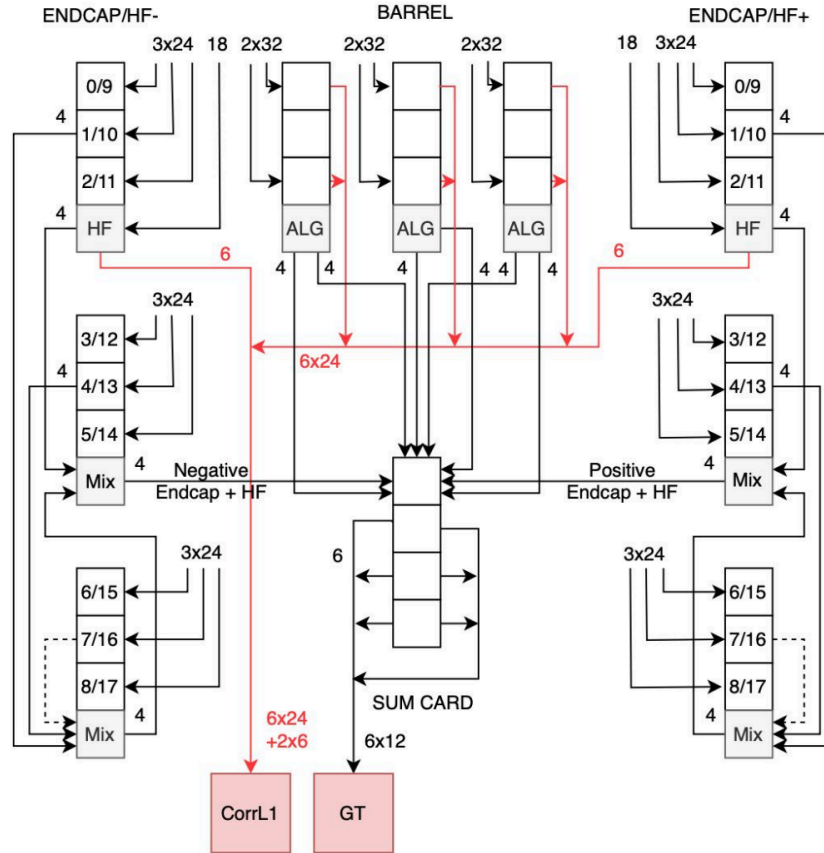


FIGURE 2.12: High-level architecture of the Global Calorimeter Trigger (GCT) system, showcasing its constituent parts: HF/HGCAL, GCT Barrel, and GCT Sum unit. (TMI6 giving 6x24 links out of GCT Barrel, red) [16]

2.4.1 GCT Barrel Architecture

Each GCT Barrel board is designed to receive input from eight RCT cards. The processing of data from these RCT cards is distributed across two SLRs within the GCT Barrel FPGA. Each SLR is responsible for processing a core region corresponding to four RCT cards, while also utilizing overlapping data from an additional four adjacent RCT cards to ensure seamless object reconstruction across boundaries. The processing of RCT data within these two SLRs proceeds independently. Each SLR instantiates an identical IP core (IP1) responsible for processing the incoming RCT card information, generating Particle Flow (PF) clusters (48 per SLR), and stitching e/γ objects (36 per SLR) across the boundaries of its effective four-RCT-card region. This processing stage prepares data for sending to the correlator layer 1.

The Correlator Layer 1 accepts data in a time-multiplexed format. Consequently, the outputs from the IP1 cores in both SLRs are directed to a Time Multiplexing (TMUX) module, implemented in VHDL. The time-multiplexed data is then transmitted to the correlator. Simultaneously, a second IP core (IP2) processes information for the GCT Sum card. IP2 receives 12 links from each of the two IP1 instances, containing e/γ cluster and SuperTower (ST) information. It then calculates jets, taus, and energy sums, which are then forwarded to the GCT Sum card. Thus, the GCT Barrel board's logic is composed of two IP1 instances, one IP2 instance, and one TMUX module. The independent processing across SLRs and the subsequent multiplexing are illustrated in Figure 2.13, while the detector geometry highlighting the need for data overlap is shown in Figure 2.14.

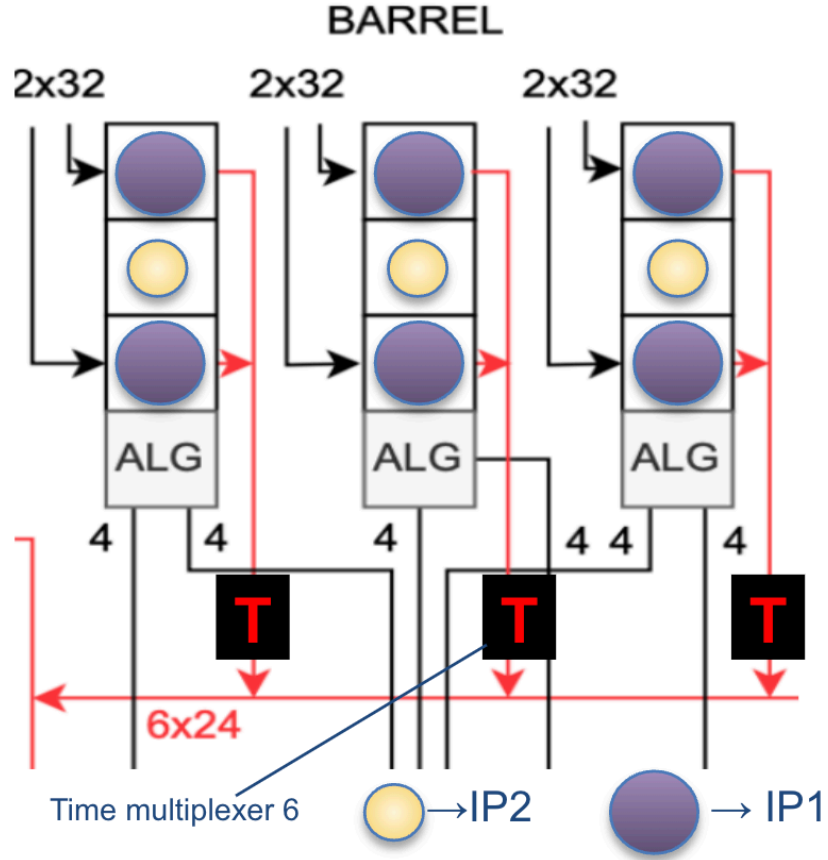


FIGURE 2.13: GCT Barrel processing illustrating independent IP1 processing across 6 SLRs. IP1 and IP2 are shown, with IP1 outputs (red) being time-multiplexed before transmission.[16]

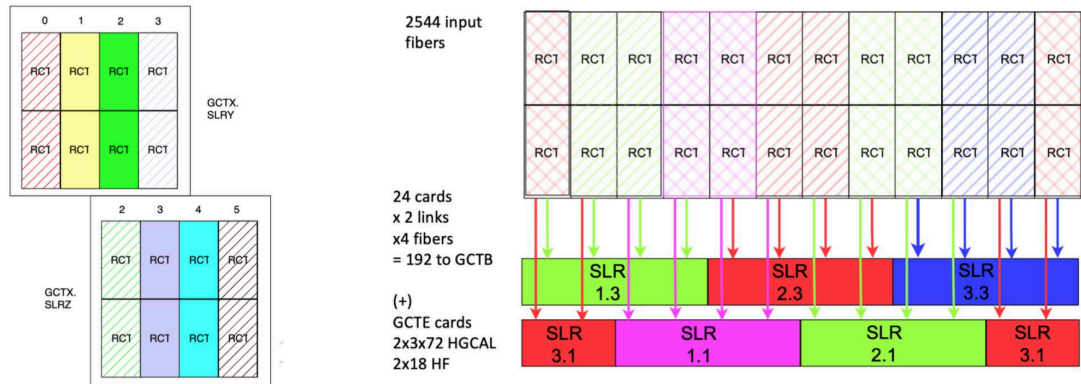


FIGURE 2.14: Representation of the barrel detector geometry, indicating regions processed by different trigger cards and illustrating the origin of data overlap requirements for seamless object reconstruction across boundaries.[16]

2.4.2 Design Requirements

The principal design requirements for the GCT Barrel firmware are analogous to those of the RCT system, emphasizing:

1. **Functionality:** Correct implementation of the GCT barrel trigger algorithms.
2. **Timing Closure and Resource Utilization:** The design must achieve timing closure at the target clock frequency of 360 MHz. Resource utilization within each FPGA SLR must remain below 70% of total availability to ensure routability and accommodate future modifications. The total latency of GCT processing should be affordable within the total L1 trigger latency budget of $12.5 \mu\text{s}$.
3. **Floorplan Constraints:** Adherence to floorplan constraints for DAQ system firmware similar to the RCT.
4. **Reproducibility:** The design must demonstrate stability and reproducibility across multiple implementation builds, achieving consistent timing closure, resource utilization, and floorplan results.

A main aspect of the GCT Barrel design arises from its method of processing RCT information independently across two SLRs (per card). To accurately stitch e/γ objects and other physics primitives across the boundaries between regions handled by different RCT cards (and subsequently different GCT SLRs), a significant data overlap is necessary. Each IP1 core within a GCT Barrel SLR, therefore, requires information from eight RCT cards in total: four corresponding to its primary processing region and four providing overlapping data from adjacent regions. This ensures that objects near the edges of an RCT's coverage can be fully reconstructed. See Fig. 2.14.

2.4.2.1 Time Multiplexing Scheme

The transmission of processed data from the GCT Barrel to the Correlator Layer 1 necessitates time multiplexing due to bandwidth constraints on the input side of correlator layer 1. An initial design consideration involved a TMUX6 scheme (Figure 2.15), wherein six time-slots would be utilized for data transmission. Under this scheme, for each SLR processing 36 e/γ clusters and 48 PF clusters, the data would be structured into 9 words per object type, fitting into 54 words per fiber (9 words \times 6 Time Multiplexing Intervals (TMI)). This would require one fiber for e/γ objects (36 EG + 18 spare words) and one fiber for PF clusters (48 PF + 6 spare words). This scheme necessitated duplication of data to handle overlaps, leading to $(1 \text{ } e/\gamma \text{ fiber} + 1 \text{ PF fiber}) \times 2 \text{ (for overlap)} \times 6 \text{ (SLRs across the system, assuming 3 cards with 2 SLRs each)} = 24 \text{ fibers per SLR processing block, totaling 144 fibers for the entire GCT barrel processing, which go towards correlator layer 1.}$

Subsequent evaluation led to the exploration of a TMUX18 scheme (Figure 2.16), using eighteen time-slots. With TMUX18, the 36 e/γ clusters and 48 PF clusters per SLR could be accommodated within 162 words per fiber (9 words \times 18 TMI). This allowed for the transmission of both e/γ objects and PF objects from both SLRs on a single fiber, assuming we can reduce the number of e/γ objects to 32 per SLR (32 EG + 48 PF = 80 words per SLR, so 160 words for two SLRs, fitting within 162 words). This would result in 1 fiber per 2 SLRs (or 1 fiber per GCT Barrel card times 18). For a system with 3 GCT Barrel cards, this could lead to a total of 54 fibers for the system, resulting in a reduced number of physical fibers going to correlator layer 1 compared to TMUX6. Plus the added advantage of not needing duplication for overlap in downstream logic, since now we have all information in the same fibre.

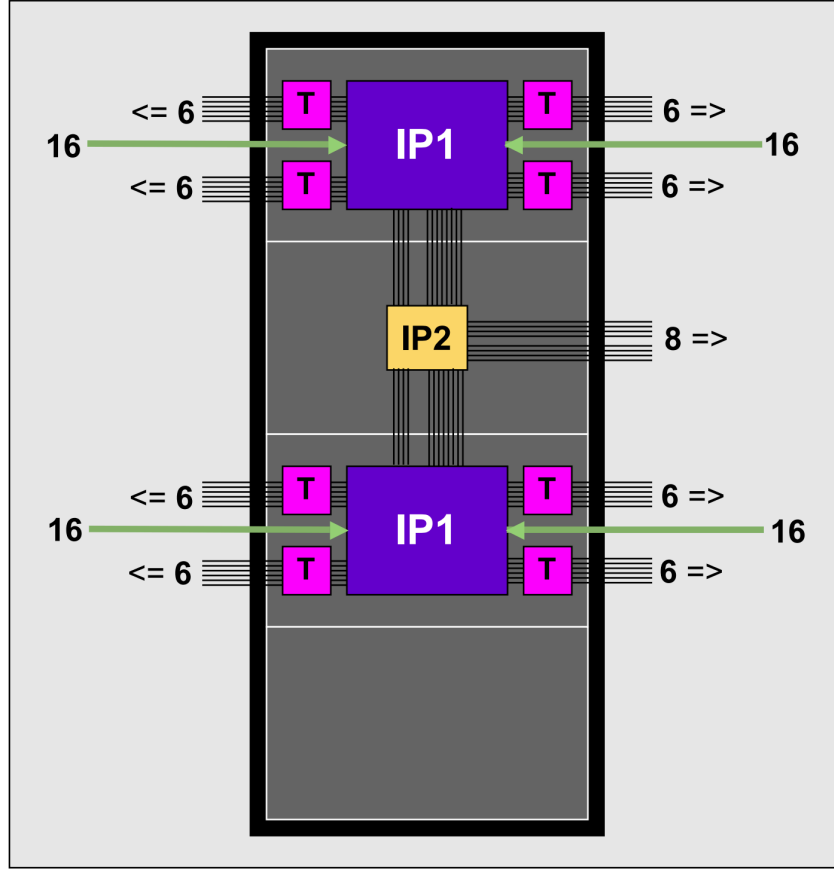


FIGURE 2.15: Conceptual GCT Barrel architecture based on the TMUX6 time-multiplexing scheme.

Following comparative analysis of interconnection efficiency and no necessity for overlap, the TMUX18 scheme was adopted as the baseline for further development due to its more favorable characteristics regarding fiber count and handling of data capacity.

2.4.3 GCT Barrel Algorithms

The GCT Barrel firmware implements algorithms within its IP1 and IP2 HLS cores to process RCT data and generate outputs to be sent to correlator layer 1 and GCT sum card respectively.

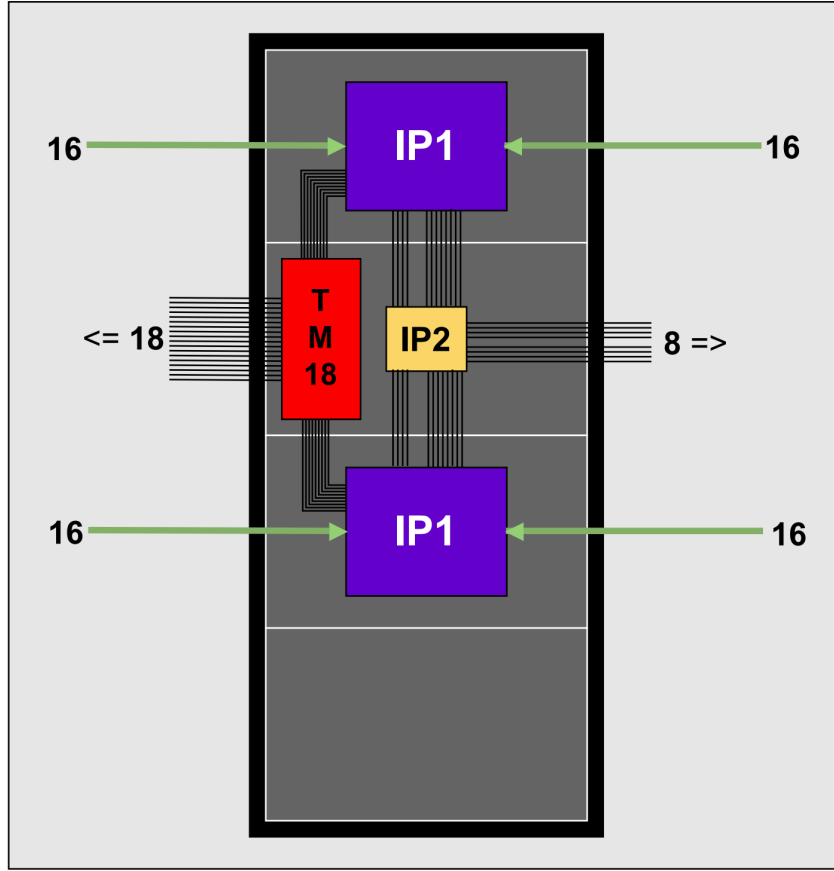


FIGURE 2.16: Conceptual GCT Barrel architecture based on the TMUX18 time-multiplexing scheme.

2.4.3.1 IP1 Algorithm (processing data for Correlator)

The IP1 core in each GCT Barrel SLR processes data from its assigned set of RCT cards (including overlaps) to produce e/γ objects, Particle Flow (PF) clusters, and SuperTowers (ST). The key algorithmic steps are:

1. **Input Data Unpacking:** Raw data, consisting of ECAL clusters and calorimeter tower energies from the input RCT links, is unpacked and formatted into internal data structures. Each of the eight RCT regions (four unique to the SLR's core processing area, four providing overlap) contributes its set of objects.

2. **Tower Energy Calculation3:** The energies of ECAL clusters are incorporated into the energies of their corresponding calorimeter towers, for consistent energy accounting within the GCT's view of the tower grid.
3. **ECAL Cluster Stitching:** To form complete e/γ objects across RCT card boundaries (which now define internal boundaries within the GCT IP1's processing region), a stitching procedure is applied. Clusters located near the η or ϕ edges of adjacent input RCT regions are evaluated. If they meet proximity criteria, their energies (including $E_{T,5x5}$) are combined into the higher-energy cluster, and the lower-energy contributor is suppressed. This is performed for all relevant boundaries among the eight input RCT data streams.
4. **Tower Overlap Propagation:** Calorimeter tower information from the overlapping RCT regions is used to populate overlap regions in the local tower map processed by IP1. This ensures that subsequent clustering algorithms operating near the boundaries of the core four-RCT-card region have access to complete energy deposit information.
5. **Particle Flow (PF) Cluster Generation:** PF clusters are formed from the refined calorimeter tower energies. For each of the input RCT regions, the tower map is divided into several three overlapping 10×10 tower sub-grids. Within each sub-grid, an iterative algorithm identifies PF cluster candidates:
 - A seed tower (highest remaining energy) is identified.
 - A 3×3 tower cluster is formed around this seed, and its total energy and position are recorded.
 - The energy of towers participating in the newly formed cluster is effectively zeroed or masked to prevent their reuse in subsequent iterations within that sub-grid.

This process is repeated a 5 times per sub-grid, yielding a set of PF clusters. The clusters from all sub-grids within an RCT region are then collected, sorted by energy, and 12 highest-energy PF clusters are selected per RCT region.

6. **SuperTower (ST) Formation:** SuperTowers, which are coarser-granularity energy sums (sums of 2×2 or 2×1 towers), are calculated from the tower grid.
7. **Output Data Preparation:** The stitched e/γ clusters, selected PF clusters, and calculated SuperTowers are formatted and packed into output links. These links are directed towards the TMUX module for transmission to the Correlator Layer 1 and also to the IP2 core for further processing for sending to gct sum card.

2.4.3.2 IP2 Algorithm (processing data for GCT Sum)

The IP2 core processes the e/γ cluster and SuperTower information received from both IP1 instances (representing the two halves of the GCT barrel i.e., positive and negative η regions). Its primary function is to generate jets, taus, and global energy sums.

1. **Input Data Aggregation:** Stitched e/γ clusters and SuperTower data from the two IP1 cores are received and organized.
2. **SuperTower Overlap Stitching at $\eta = 0$:** If the IP1s process distinct η hemispheres, SuperTower information near the $\eta = 0$ boundary is exchanged and stitched between the two sets of inputs for continuity in jet and tau finding across the central region of the detector.

3. **Jet Reconstruction:** Jets are reconstructed from the SuperTower grid. An iterative sliding window or seed-based algorithm is employed, independently for each effective input IP1 region:

- A seed SuperTower (highest remaining energy) is identified.
- A jet is formed by summing energies in a 3×3 window of SuperTowers centered on the seed.
- The SuperTowers contributing to this jet are masked or their energy zeroed to prevent re-clustering.

This process is repeated to find 2 jets per effective IP1 input region, leading to four overall from IP2.

4. **Tau Reconstruction:** Tau candidates are identified from the SuperTower grid, using the energy remaining after jet reconstruction or operating on a copy of the original SuperTower data :

- A seed SuperTower (highest remaining energy) is identified.
- The energy of this single seed SuperTower (or a very small local region, 1×1 or 2×1 STs) is taken as the tau candidate's energy.
- The seed SuperTower(s) are then masked.

This process is repeated to find a predetermined number of tau candidates (here, two per effective IP1 input region, leading to four overall).

5. **Final e/γ Object Selection:** From the collection of e/γ clusters received from the IP1s, a final selection of the highest energy candidates (top six overall) across both input streams is made and sorted.

6. **Global Energy Summation:** Scalar sums, such as total transverse energy (E_T^{tot}), missing transverse energy (MET, from E_x and E_y sums), and total

hadronic transverse energy (H_T^{tot}), are computed by summing the appropriate energy components from all SuperTowers across both IP1 input regions.

7. **Output Data Preparation:** The reconstructed jets, taus, selected final e/γ objects, and global energy sums are formatted and packed into output links for transmission to the GCT Sum card.

IP	Latency min (cycles)	Latency max (cycles)	Latency absolute	II (cycles)
IP1	132	132	$0.367 \mu s$	9
IP2	28	28	77.784 ns	9

TABLE 2.2: Latency of GCT Barrel HLS IP cores.
Total GCT Barrel Latency = $0.444 \mu s$

Chapter 3

Implementation

3.1 Regional Calorimeter Trigger (RCT) Implementation

The successful deployment of the RCT 17×6 algorithms involves several key implementation steps beyond the core logic design in HLS. These include the integration of the HLS-generated IP cores with VHDL wrappers, careful floorplanning to meet physical constraints, and ensuring timing closure through appropriate pipelining and data alignment strategies.

The HLS IP cores, which encapsulate the algorithmic logic described in Section [2.3.4](#), are integrated into the larger FPGA design using VHDL wrappers. These wrappers serve multiple purposes:

- **Firmware Shell Interface:** They provide the necessary interface logic to connect the IP cores to the standardized APx-F firmware shell, which handles board-level functionalities such as clock distribution, reset management, and

external data links. FW Shell also implements board to board communication protocol (CMS Standard Protocol).

- **Inter-IP Communication:** For data passed between successive IP cores (e.g., from IP1 instances to IP21, then to IP22, and finally to IP3), the wrappers manage the axistream links and ensure proper data synchronization. This often involves the insertion of pipeline stages or alignment buffers within the wrapper logic to compensate for varying latencies of the IP cores and routing delays, thereby ensuring that data arrives at the inputs of subsequent IPs correctly aligned in time.
- **Control and Status Register Interface:** Wrappers also facilitate access to control and status registers within the IP cores, via a standardized bus protocol like AxiStreams.

3.1.1 Threshold Calibration

A significant feature implemented in the RCT is the real-time update capability for threshold calibration constants used in IP21 (see Section 2.3.3). To achieve this, the calibration values are written to memory-mapped registers within the FPGA. This write access is managed through a User Space I/O (UIO) driver, communicating with the FPGA via the AXI4-Lite protocol. The IP21 core is designed to read these register values through an `ap_ctrl_none` interface [17] during its operation. This mechanism allows for the dynamic adjustment of calibration constants by software without requiring a full FPGA reprogramming, which is crucial for minimizing detector deadtime during calibration adjustments. The general scheme for this real-time update, including the wrapper structure around the IP core facilitating this access,

is illustrated in Figure 3.1. Eventually this same infrastructure will be implemented in future versions of GCT Barrel.

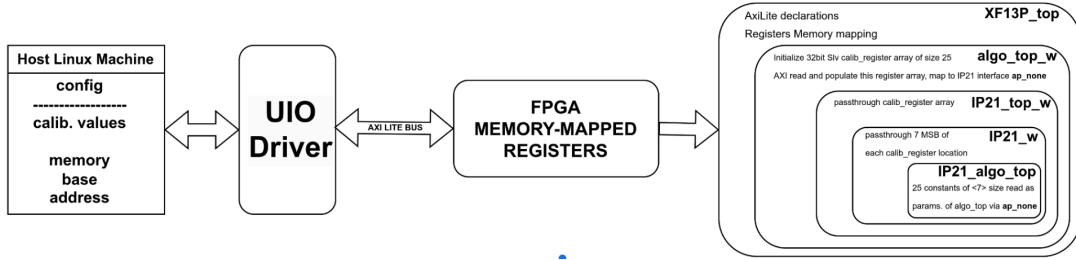


FIGURE 3.1: Mechanism for real-time updates of calibration constants. Software writes constants to memory-mapped registers via AXI4-Lite, which are then read by the IP21 core. The wrapper structure around the IP core, facilitating this interface and other connections, is also depicted.

3.1.2 Floorplanning

Adherence to the floorplan constraints, particularly the reservation of peripheral space for DAQ firmware (as discussed in Section ??), is critical. Physical constraints were applied to the IP cores using placement blocks (pblocks) during the FPGA implementation process. These pblocks define specific rectangular regions on the FPGA die where the logic for each IP core (or groups of related IPs) must be placed and routed. The goal was to arrange the IP cores (IP1s, IP21, IP22, IP3) within their designated SLRs in a manner that respects the DAQ keep-out zones while also optimizing for timing and routability. The achieved floorplan, as realized in the implemented design, closely matched the target layout, successfully accommodating all algorithmic and infrastructure logic within the specified constraints, as shown in Figure 3.2.

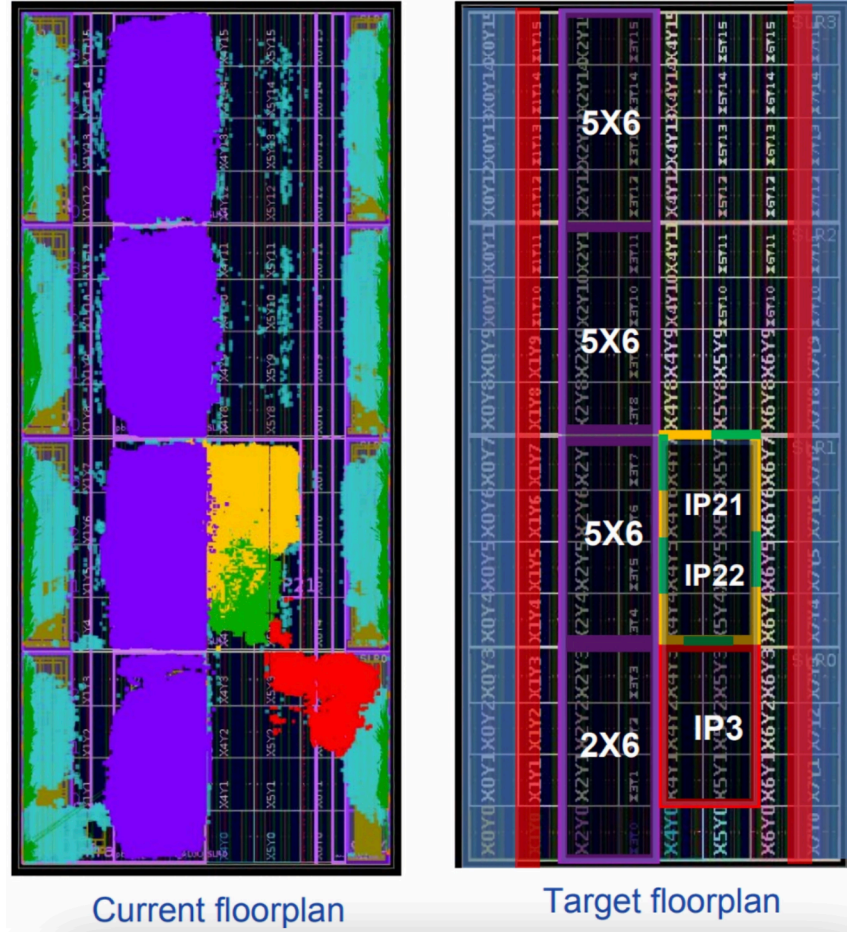


FIGURE 3.2: Comparison of the target floorplan (left) and the achieved floorplan (right) for the RCT 17×6 design. The placement of IP cores respects the DAQ keep-out regions and SLR boundaries.

3.1.3 Timing Closure and Resource Utilization

Achieving timing closure at the target operating frequency of 360 MHz was a primary objective. This involved careful HLS coding practices, insertion of pipeline stages within the IP cores and VHDL wrappers, and optimized placement and routing guided by the floorplan. The stability of the timing closure was verified by performing multiple (ten) complete build cycles of the design. This process confirmed that the design consistently met timing requirements and that the floorplan and resource utilization figures remained stable across builds, indicating a robust

and non-metastable implementation.

The overall timing closure results and total resource utilization for the RCT 17×6 firmware are summarized in Figure 3.3. The design successfully met the 360 MHz timing target.

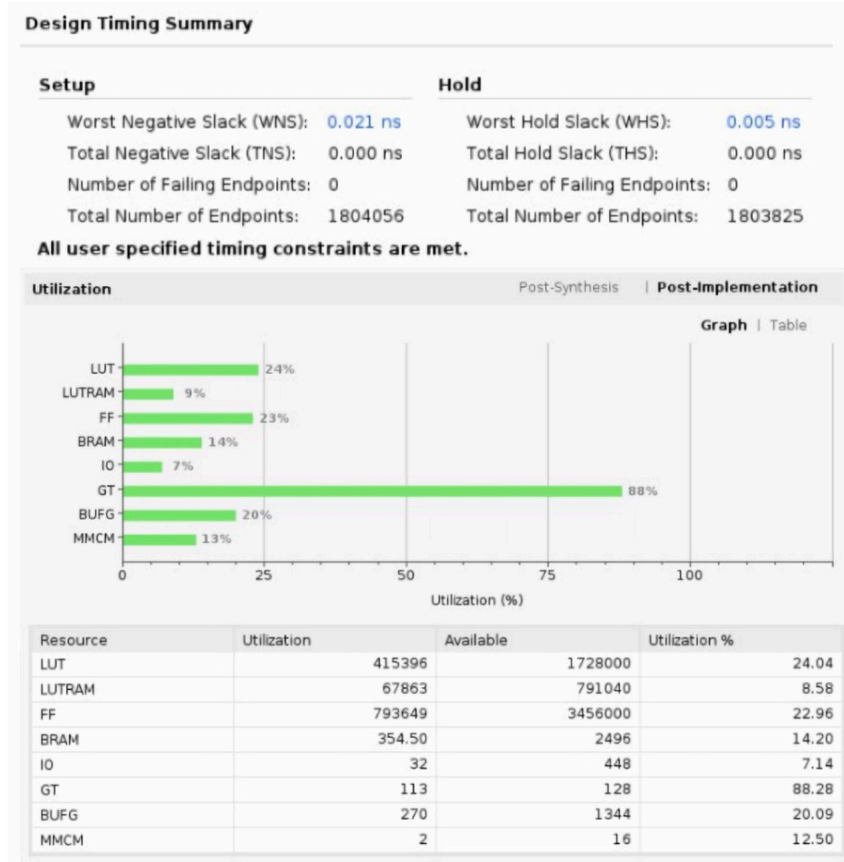


FIGURE 3.3: Summary of timing closure results and total FPGA resource utilization for the implemented RCT 17×6 design.

An analysis of resource utilization on a per-SLR basis (Figure 3.4) reveals that Configurable Logic Blocks (CLBs) were the most heavily utilized resource, which is expected given the significant amount of combinational logic in the calorimeter trigger algorithms. However, even in the most utilized SLR (SLR1, which hosts IP21 and IP22 in the revised architecture), the CLB utilization remained at approximately 54%. This is well within the design requirement of not exceeding 70% utilization per

SLR, leaving adequate margin for routing and potential future enhancements. Other resources, such as DSP slices and Block RAMs, were utilized at lower percentages.

3. SLR CLB Logic and Dedicated Block Utilization

Site Type	SLR0	SLR1	SLR2	SLR3	SLR0 %	SLR1 %	SLR2 %	SLR3 %
CLB	27715	29292	22544	22885	51.32	54.24	41.75	42.38
CLBL	15080	15946	12151	12388	51.50	54.46	41.50	42.31
CLBM	12635	13346	10393	10497	51.11	53.99	42.04	42.46
CLB LUTs	90341	127885	100916	100507	20.91	29.60	23.36	23.27
LUT as Logic	85884	102560	77664	77227	19.88	23.74	17.98	17.88
using O5 output only	1541	1611	1450	1438	0.36	0.37	0.34	0.33
using O6 output only	72182	80079	61865	61406	16.71	18.54	14.32	14.21
using O5 and O6	12161	20870	14349	14383	2.82	4.83	3.32	3.33
LUT as Memory	4457	25325	23252	23280	2.25	12.81	11.76	11.77
LUT as Distributed RAM	151	0	0	0	0.08	0.00	0.00	0.00
LUT as Shift Register	4306	25325	23252	23280	2.18	12.81	11.76	11.77
using O5 output only	1	3	0	0	<0.01	<0.01	0.00	0.00
using O6 output only	3979	18939	18325	18381	2.01	9.58	9.27	9.29
using O5 and O6	326	6383	4927	4899	0.16	3.23	2.49	2.48
CLB Registers	183655	238807	182022	180704	21.26	27.64	21.07	20.91
CARRY8	3026	4289	3043	3043	5.60	7.94	5.64	5.64
F7 Muxes	680	895	801	646	0.31	0.41	0.37	0.30
F8 Muxes	21	2	3	4	0.02	<0.01	<0.01	<0.01
F9 Muxes	0	0	0	0	0.00	0.00	0.00	0.00
Block RAM Tile	84.5	90	90	90	12.57	13.39	13.39	13.39
RAMB36/FIFO	84	90	90	90	12.50	13.39	13.39	13.39
RAMB18	1	0	0	0	0.07	0.00	0.00	0.00
URAM	0	0	0	0	0.00	0.00	0.00	0.00
DSPs	0	0	0	0	0.00	0.00	0.00	0.00
Unique Control Sets	5197	5961	5455	5306	4.81	5.52	5.05	4.91

* Note: Available Control Sets based on CLB Registers / 8

FIGURE 3.4: Per-SLR resource utilization breakdown for the RCT 17×6 design. SLR1 shows the highest CLB utilization at 54%, within the design target. [16]

3.2 Global Calorimeter Trigger (GCT) Implementation

The implementation of the Global Calorimeter Trigger (GCT) Barrel firmware, similar to the RCT, involves integrating HLS-generated IP cores (IP1 and IP2) with VHDL wrappers, managing floorplanning, ensuring timing closure, and specifically, implementing the Time Multiplexing (TMUX) functionality critical for data transmission to the Correlator Layer 1.

3.2.1 Time Multiplexing (TMUX) Implementation

Time multiplexing is a core requirement for the GCT Barrel outputs that feed the Correlator Layer 1. The fundamental principle is to take data from multiple parallel input streams, representing different Bunch Crossings (BX) or logical data segments processed in parallel, and interleave them onto a smaller number of physical output links over successive time slots.

Consider a generic TMUX- N scheme with N input links and N output links, operating synchronously with the algorithm clock. The goal is to ensure that data from the k^{th} logical input segment (e.g., k^{th} BX, or data from the k^{th} parallel processing unit) across all N input streams at a specific time T_0 is routed to the k^{th} output link at a subsequent time $T_0 + \delta_k$. After N such time intervals, the pattern repeats. For example, data corresponding to the 0^{th} BX from all input links would be directed to output link 0. Then, data for the 1^{st} BX from all input links would go to output link 1, and so on. After the $(N - 1)^{th}$ BX data is sent to output link $N - 1$, the $(N)^{th}$ BX data would again be routed to output link 0. This effectively serializes parallel data segments onto the output links in a round-robin fashion.

3.2.1.1 TMUX6

The TMUX6 module is designed to multiplex 6 input AXI streams onto 6 output AXI streams. Each input 576-bit stream delivers 64-bit data per clock cycle over 9 clocks to complete 1BX. The core of the TMUX6 logic, as illustrated in the flowchart in Figure 3.5, revolves around a 2D memory buffer and a set of counters that manage write and read addressing.

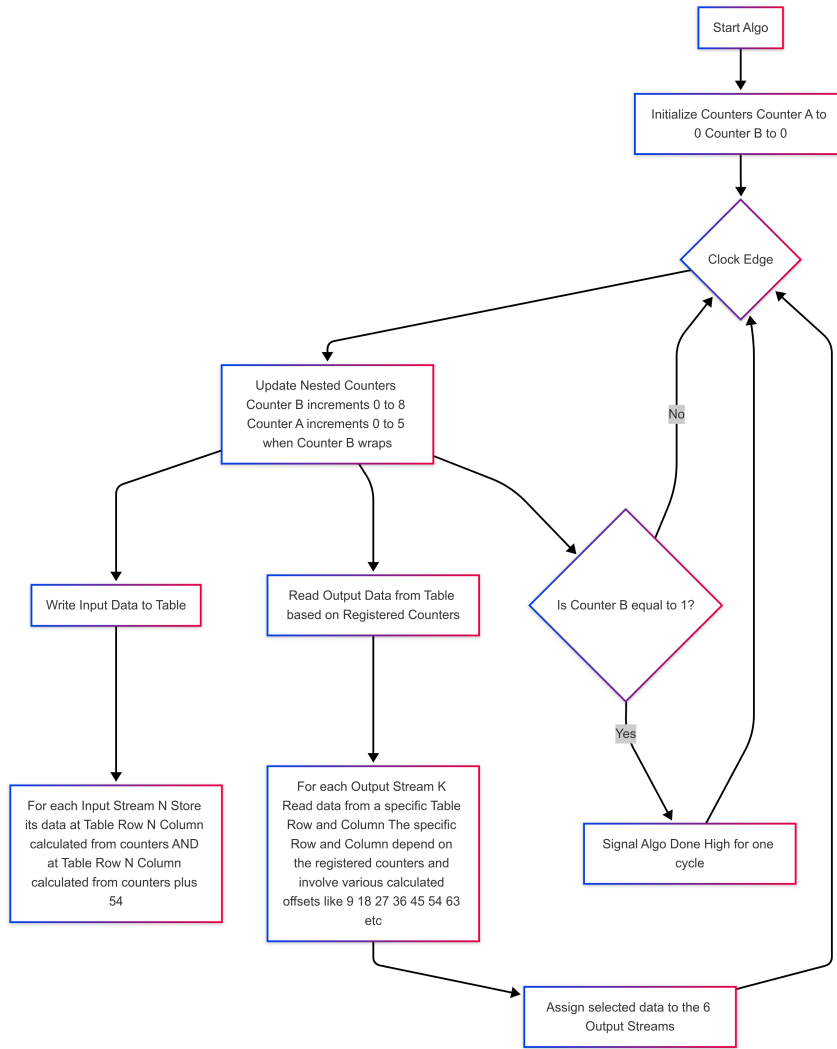


FIGURE 3.5: Flowchart illustrating the operational logic of the TMUX6 module.

The internal buffer can be conceptualized as a table, for instance, of size 6×108 elements, where each element is a 64-bit word. The 6 rows correspond to the 6 input

streams. The 108 columns are logically divided into blocks (12 blocks of 9 elements each).

- **Write Operation:** Incoming data from the 6 input streams is written into this table based on a primary write counter system. This system might consist of a fast counter ('count_b' from 0 to 8) indexing positions within a time slot, and a slower counter ('count_a' from 0 to 5) indexing the time slots themselves. To ensure data availability for the complex read patterns, input data is written to two locations simultaneously: a primary location and a "mirror" location at a fixed offset within the buffer.
- **Read Operation:** The read operation is governed by a separate set of output counters which are delayed versions of the write counters. The key complexity lies in the read addressing logic, which permutes the data from different input streams and buffer locations onto the output streams. The specific permutation changes with each time slot (as indexed by 'count_out_a'). For example, in time slot 0, output stream 0 might receive data from input stream 0 (from its mirror block), output stream 1 from input stream 5 (from a primary block), and so on. This pattern rotates or shifts for subsequent time slots, achieving the desired time-multiplexed output where each output link carries a sequence of data segments originally from different input streams at different logical time points.

The TMUX6 effectively implements a reconfigurable delay and routing network, ensuring that data segments are correctly ordered and timed for downstream processing.

3.2.1.2 TMUX18

The TMUX18 module, designed to multiplex 18 input streams onto 18 output streams, was developed with a focus on resource optimization and reusing existing logic. As shown in the flowchart in Figure 3.6, its architecture is based on hierarchical composition using two instances of a TMUX9 module. The TMUX9 module itself is an extension of the TMUX6 logic, adapted for 9 input/output streams but following similar principles of buffered writing and permuted reading.

The TMUX18 implementation operates as follows:

1. **Input Splitting:** The 18 input AXI streams are divided into two groups of 9 streams each.
2. **Parallel TMUX9 Processing:**
 - The first group of 9 input streams (inputs 0-8) is fed into the first TMUX9 instance (TMUX9a).
 - The second group of 9 input streams (inputs 9-17) is fed into the second TMUX9 instance (TMUX9b).

Both TMUX9 instances operate in parallel, performing the 9-stream time multiplexing internally.

3. **Delayed Path for TMUX9b Output:** The 9 output streams from TMUX9b are passed through a delay line, specifically an 81-clock-cycle delay. This delay is crucial for the final interleaving stage. The delay line is implemented as a shift register for each of the 9 AXI streams from TMUX9b.
4. **Output Switching:** A counter (cycling from 0 to 80, then resetting) and a toggle signal control the final output stage.

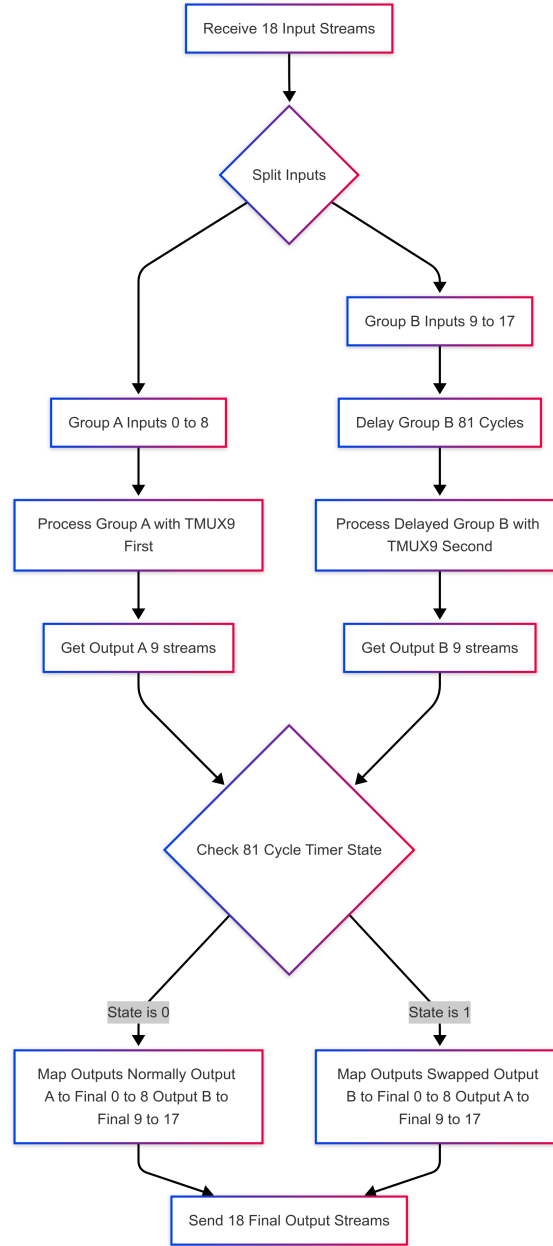


FIGURE 3.6: Flowchart illustrating the operational logic of the TMUX18 module, highlighting its construction from TMUX9 instances and a delay line.

- For the first 81 clock cycles (`toggle = '0'`), output streams 0-8 of the TMUX18 are sourced directly from TMUX9a's outputs, and output streams 9-17 are sourced from the (now 81-cycle delayed) outputs of TMUX9b (i.e., `'delay_line_out(0)'`).

- For the next 81 clock cycles (toggle = '1'), the sources are swapped: output streams 0-8 of the TMUX18 are sourced from the delayed TMUX9b outputs, and output streams 9-17 are sourced directly from TMUX9a's outputs.

This interleaving, combined with the internal multiplexing of the TMUX9 instances, achieves the full TMUX18 functionality. The 81-cycle period corresponds to 9×9 , suggesting that each TMUX9 completes a full cycle of its 9 input segments over 9 of its internal time slots, and the interleaving ensures that segments from TMUX9a and TMUX9b are correctly ordered over the larger 18-segment cycle of TMUX18.

This hierarchical design allows for reuse of the TMUX9 logic and simplifies the implementation and verification of the more complex TMUX18. The waveform illustrating the sequential output of data bunches from the designed TMUX18 is shown in Figure 3.7.

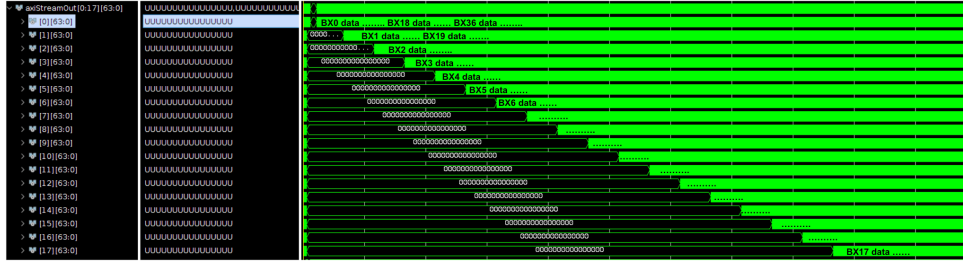


FIGURE 3.7: Output waveform from TMUX18 module, showing successive data bunches emerging from the output links.

3.2.2 TMUX6-based GCT Barrel

An initial exploration of the GCT Barrel architecture was based on the TMUX6 scheme. This design involved instantiating the two primary IP1 cores (each processing data equivalent to four RCT cards plus overlaps) and multiple TMUX6 instances

to handle their outputs. A partial implementation focusing on the two IP1 cores and eight TMUX6 instances was developed before it was replaced in favor of the TMUX18 based architecture.

3.2.2.1 Floorplan (TMUX6 GCT)

The floorplan for this partial TMUX6-based GCT design is shown in Figure 3.8. The two IP1 cores were placed in their respective SLRs, and the TMUX6 instances were distributed as required. Even in this partial state, the design successfully achieved timing closure, indicating the viability of the TMUX6 module itself. However, as discussed in Section 2.2.2.1, the overall system implications (e.g., fiber count due to overlap handling) led to the TMUX18 scheme being favored for the final GCT Barrel design.

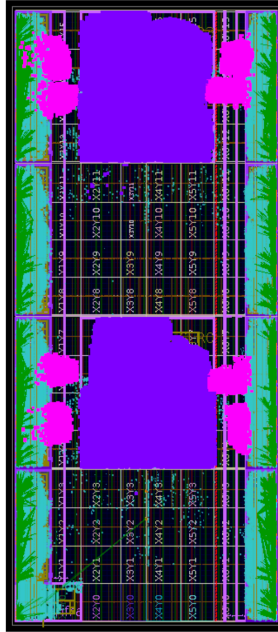


FIGURE 3.8: Floorplan of the exploratory TMUX6-based GCT Barrel design. This partial layout includes two IP1 cores and eight TMUX6 instances and achieved timing closure.

3.2.3 TMUX18-based GCT Barrel (Final Design)

The definitive GCT Barrel architecture was developed using the TMUX18 scheme for its outputs to the Correlator Layer 1. This design incorporates the two IP1 cores, the IP2 core, and the TMUX18 module.

3.2.3.1 Floorplan (TMUX18 GCT)

The final floorplan for the TMUX18-based GCT Barrel is presented in Figure 3.9. The IP1 cores are placed in two different SLRs to process their respective detector regions. The IP2 core, which aggregates data from both IP1s, and the TMUX18 module are placed in SLR2 sandwiched in between SLR3 and SLR1 with IP1 instances. The floorplan was carefully managed to ensure adherence to DAQ constraints and to optimize for timing closure.

3.2.3.2 Timing Closure and Utilization (TMUX18 GCT)

The TMUX18-based GCT Barrel design successfully achieved timing closure at the target operating frequency of 360 MHz. The stability of this timing closure was verified through multiple build iterations. The overall timing results and total FPGA resource utilization are summarized in Figure 3.10.

A per-SLR resource utilization breakdown is provided in Figure 3.11. Similar to the RCT, Configurable Logic Blocks (CLBs) were the most utilized resource due to the complex data processing algorithms in IP1 and IP2, and the logic within the TMUX18 module. The maximum CLB usage in any single SLR reached approximately 68%. This utilization level is within the non-conservative design target of

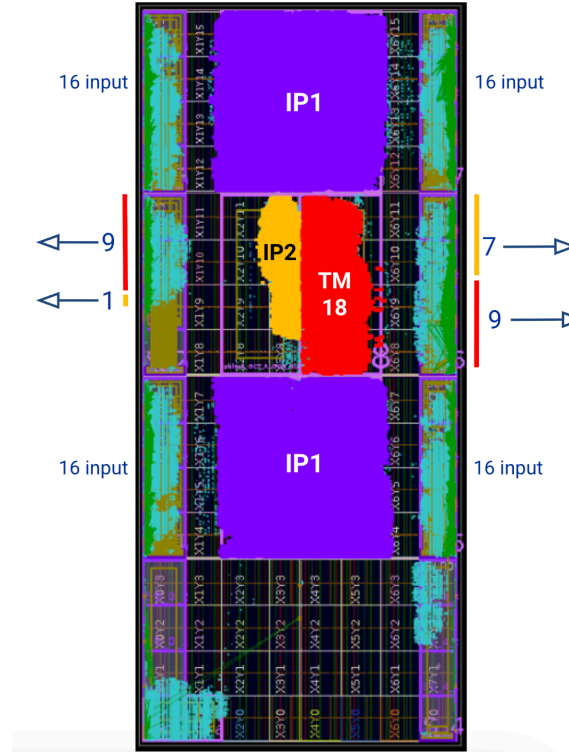


FIGURE 3.9: Final achieved floorplan of the GCT Barrel design utilizing the TMUX18 scheme. This full design, including IP1, IP2, and TMUX18 modules, successfully closed timing.

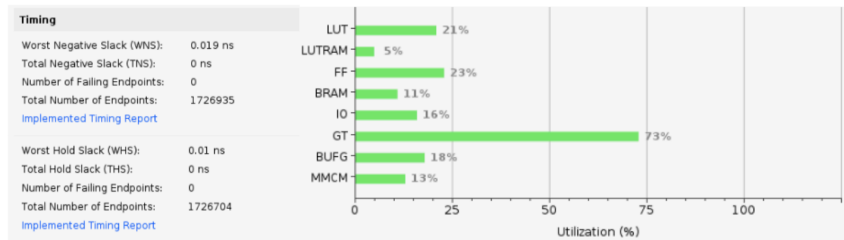


FIGURE 3.10: Summary of timing closure results and total FPGA resource utilization for the final TMUX18-based GCT Barrel design.

70%. Other resources like DSPs and BRAMs were consumed at lower rates, reflecting the nature of the implemented algorithms. The consistent achievement of timing closure across multiple builds, along with manageable resource utilization, confirmed the robustness of the final GCT Barrel design.

Site Type	SLR0	SLR1	SLR2	SLR3	SLR0 %	SLR1 %	SLR2 %	SLR3 %
CLB	5188	36976	21346	36632	9.61	68.47	39.53	67.84
CLBL	2872	20207	11170	19923	9.81	69.01	38.15	68.04
CLBM	2316	16769	10176	16709	9.37	67.84	41.17	67.59
CLB LUTs	14926	160114	55723	159862	3.46	37.06	12.90	37.01
LUT as Logic	14692	133373	52401	133302	3.40	30.87	12.13	30.86
using O5 output only	657	2970	2314	4090	0.15	0.69	0.54	0.95
using O6 output only	10385	90648	42012	88711	2.40	20.98	9.73	20.53
using O5 and O6	3650	39755	8075	40501	0.84	9.20	1.87	9.38
LUT as Memory	234	26741	3322	26560	0.12	13.52	1.68	13.43
LUT as Distributed RAM	155	0	0	0	0.08	0.00	0.00	0.00
LUT as Shift Register	79	26741	3322	26560	0.04	13.52	1.68	13.43
using O5 output only	0	0	0	0	0.00	0.00	0.00	0.00
using O6 output only	79	20857	2130	20812	0.04	10.55	1.08	10.52
using O5 and O6	0	5884	1192	5748	0.00	2.98	0.60	2.91
CLB Registers	28668	277307	133362	273487	3.32	32.10	15.44	31.65
CARRY8	220	6873	587	6873	0.41	12.73	1.09	12.73
F7 Muxes	228	1009	5943	810	0.11	0.47	2.75	0.38
F8 Muxes	0	2	2	2	0.00	<0.01	<0.01	<0.01
F9 Muxes	0	0	0	0	0.00	0.00	0.00	0.00
Block RAM Tile	12.5	96	78	96	1.86	14.29	11.61	14.29
RAMB36/FIFO	12	96	78	96	1.79	14.29	11.61	14.29
RAMB36E2 only	10	96	78	96	1.49	14.29	11.61	14.29
RAMB18	1	0	0	0	0.07	0.00	0.00	0.00
URAM	0	0	0	0	0.00	0.00	0.00	0.00
DSPs	0	0	0	0	0.00	0.00	0.00	0.00
PLL	0	0	0	0	0.00	0.00	0.00	0.00
MMCM	0	0	0	0	0.00	0.00	0.00	0.00
Unique Control Sets	1506	7259	3652	7245	1.39	6.72	3.38	6.71

FIGURE 3.11: Per-SLR resource utilization for the TMUX18-based GCT Barrel design. The maximum CLB usage is noted at 68% in the most utilized SLR.[16]

3.3 Optimizations to the Vivado Implementation Flow

Achieving consistent timing closure, resource utilization, and floorplan stability for complex FPGA designs like the RCT and GCT Barrel often requires more than standard synthesis and implementation runs. Several optimizations were integrated into the Xilinx Vivado Design Suite flow to enhance design performance, reproducibility, and reduce iteration times.

3.3.1 Physical Optimization (PhysOpt) Looping

Physical optimization (‘phys_opt_design’ command in Vivado) plays a vital role in resolving timing violations that persist after initial placement. Standard ‘phys_opt_design’ performs a series of optimizations such as high-fanout net optimization, critical

cell optimization, DSP/BRAM/URAM register retiming, and placement-based optimization [18]. However, for challenging paths, a single pass of ‘phys_opt_design’ might not be sufficient.

To address this, a "physopt looping" strategy was employed, as illustrated in the flowchart in Figure 3.12. This iterative approach involves:

1. Running an initial ‘phys_opt_design’ pass post-placement.
2. Checking the Worst Negative Slack (WNS).
3. If WNS is still negative (indicating timing violations), specific ‘phys_opt_design’ directives targeted at the failing paths (e.g., focusing on logic replication, hold fixing, or specific types of retiming) can be re-applied, or a general ‘phys_opt_design’ can be run again.
4. This loop can be repeated a few times or until WNS becomes non-negative or improvements diminish.

This iterative application of physical optimization before routing helps to make WNS positive by aggressively tackling difficult timing paths early in the flow. A similar, typically less aggressive, pass of ‘phys_opt_design’ can also be beneficial after routing (‘route_design’) to clean up any minor negative slack introduced during routing or to further optimize hold times. This iterative use of physical optimization proved effective in improving timing closure.

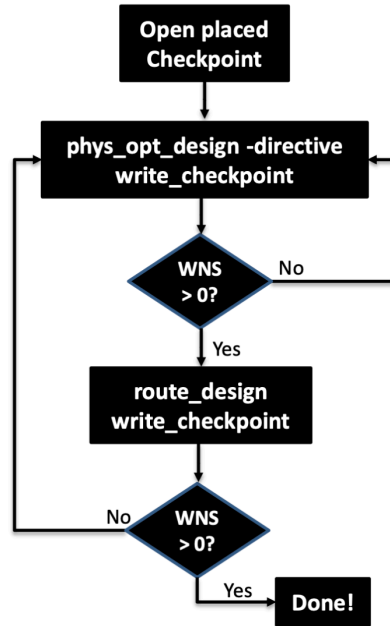


FIGURE 3.12: Flowchart illustrating the iterative physopt looping strategy employed in the Vivado flow. This involves repeated application of physical optimization steps, combined with timing analysis, to improve WNS before and after routing.^[21]

3.3.2 Incremental Synthesis and Implementation

For large designs, full synthesis and implementation runs can be time-consuming. Incremental compilation allows Vivado to reuse results from previous runs for unchanged portions of the design, significantly speeding up iterations when only minor modifications are made. ^[19]

- **Incremental Synthesis:** If only a small part of the RTL code (e.g., a single module) is modified and there are no top level design changes, incremental synthesis can reuse the synthesis results for the unchanged modules, recompiling only the affected hierarchy.
- **Incremental Implementation:** Similarly, if post-synthesis netlists have minor changes, or if only constraints are modified, incremental implementation can reuse placement and routing data for unchanged partitions or blocks. This

is particularly useful for floorplanned designs, where preserving the placement of critical, already-timed blocks is advantageous.

Utilizing incremental flows not only reduces build times but also contributes to design stability. By preserving the placement and routing of well-optimized and timing-closed sections of the design, it minimizes the chances of new timing issues arising from unrelated changes, thereby improving the predictability of timing closure.

3.3.3 Out-of-Context (OOC) Synthesis for HLS IP Cores

The RCT and GCT designs make extensive use of High-Level Synthesis (HLS) to generate IP cores from C++ descriptions. Synthesizing these HLS IP cores in Out-of-Context (OOC) mode [19] offers several benefits:

- **Modularity and Reduced Compilation Time:** Each HLS IP core is synthesized independently as a standalone block, separate from the top-level design. This allows for parallel synthesis of multiple IPs and significantly reduces the overall synthesis time for the full design, as the pre-synthesized IP netlists are simply instantiated.
- **Improved Reproducibility and Stability:** When an HLS IP core is synthesized OOC, its internal logic, resource usage, and interface timing characteristics are determined without the influence of the surrounding top-level logic or constraints (beyond its own interface definition). This leads to more predictable and reproducible results for the IP core itself across different top-level design builds. If the IP core's source code or HLS directives do not change, its OOC synthesis results will remain consistent.

- **Facilitates Bottom-Up Design and Floorplanning:** Having pre-synthesized, characterized IP blocks simplifies a bottom-up design approach. The known resource footprint and interface timing of OOC IPs make it easier to create accurate floorplans and to budget resources at the top level.

By synthesizing the HLS IP cores (like IP1, IP21, IP22, IP3 in RCT, and IP1, IP2 in GCT) in OOC mode, and then integrating these as black boxes (or grey boxes with timing models) into the top-level VHDL structure, better control over the implementation process was achieved, contributing to overall design stability and reproducibility. This approach aligns well with the use of pblocks for floorplanning, as the OOC IPs can be directly assigned to their respective pblocks.

Chapter 4

Testing and Validation

4.1 Methodology

The comprehensive testing and validation of the Regional Calorimeter Trigger (RCT) and Global Calorimeter Trigger (GCT) Barrel firmware encompassed a hierarchical strategy. This approach progressed from unit-level simulations of individual IP cores and High-Level Synthesis (HLS) C++/RTL co-simulations, through to standalone single-board hardware tests, and ultimately to multi-board system-level evaluations facilitated by the FEAST environment (as detailed in Section 4.3). This chapter specifically elaborates on the methodology employed for standalone single-board hardware tests, a critical phase for verifying the implemented firmware (bitfile) directly on the target FPGA hardware.

The overarching goal of these standalone tests is to confirm that the behavior of the firmware operating on the physical FPGA precisely matches the behavior predicted by prior (RTL and C) simulation stages. This involves programming the FPGA with the compiled bitfile, configuring operational parameters such as calibration

constants, supplying predefined input test vectors, capturing the resulting output data, and performing a comparison against established reference outputs for multiple BX data.

4.2 Standalone Single-Board Hardware Tests

Standalone single-board tests were conducted using the VU13P-2 based APx-F hardware platform. These tests used a structured test environment containing necessary scripts, board and buffer configuration files, input test vectors with specific test patterns, and corresponding reference output files.

The generalized workflow for conducting a standalone single-board hardware test can be summarized as follows:

1. **Test Environment Preparation:** A dedicated directory structure was established, housing all artifacts pertinent to the test. This included the specific firmware bitfile under scrutiny, configuration files for the FPGA board and its data buffer configuration files (MGT config), input test vector files, and the reference output files.
2. **Firmware Bitfile Loading:** The compiled bitfile corresponding to the RCT or GCT Barrel firmware version being validated was programmed onto the FPGA. This was accomplished using platform-specific scripts for the APx board series, which automate the low-level FPGA configuration process.
3. **Configuration and Calibration Data Loading:** For firmware designs incorporating run-time configurable parameters, such as the P_T -dependent

shower shape threshold calibration constants within the RCT's IP21 module, these parameters were loaded into the FPGA's memory-mapped registers post-bitfile programming. This step typically involved executing custom scripts that communicated with the FPGA, via an AXI4-Lite interface, to write and read-back the constants.

4. **Pattern Test Execution:** A scripted pattern test was initiated. This script orchestrated the configuration of the FPGA's input data buffers to stream the predefined test vectors into the firmware and simultaneously configured output buffers to capture the data processed by the firmware. The script managed the synchronization and execution of this data playback and capture sequence.
5. **Output Verification:** The data captured from the hardware execution, stored in an output file, was compared against the corresponding reference output file. The reference outputs themselves were the result of a consistent verification chain, originating from HLS C++ simulations and validated through RTL simulations.

4.2.1 RCT Standalone Tests

The standalone single-board tests for the RCT 17×6 firmware adhered to the general methodology described above. A notable element of these tests was the dynamic loading of threshold calibration constants for the IP21 module, enabling verification of this run-time update capability. Input test vectors were carefully crafted to exercise all significant algorithmic stages of the RCT, including e/γ cluster finding, inter-region stitching, sorting, shower shape calibration, energy reintegration from

rejected clusters, and the incorporation of HCAL data. The successful comparison of hardware outputs against simulation-derived reference data confirmed the functionality of the RCT firmware implementation.

4.2.2 GCT Barrel Standalone Tests

Similarly, the GCT Barrel firmware underwent standalone single-board hardware testing. This involved loading the GCT bitfile, providing input test vectors representative of the aggregated data from multiple RCTs, and capturing the GCT's output, which included reconstructed physics objects (e/γ candidates, PF clusters, jets, taus) and global energy sums. For certain developmental iterations of the GCT Barrel firmware where run-time calibration features were not yet fully integrated, the calibration loading step was bypassed. Nevertheless, the core functionality, encompassing complex data aggregation from multiple sources, object reconstruction algorithms within its IP1 and IP2 modules, and the critical TMUX data multiplexing was validated by comparing hardware outputs against reference data from RTL simulations.

MATCH

MISMATCH

Data for bx1

IP3 links from HLS sim

link_out[0]:

0x006811E412EE0002

0x06811E40CAE0002

0x06811E4066E0002

0x3C01A000CA80000

0x3C01A0006680000

0x3C01A0012E80000

0x600190014500170

0x60019000E100170

0x600190007D00170

link_out[1]:

0x0000000040001FFF

5FFF5FFF00004000

5FFF5FFF5FFF0000

40005FFF5FFF5FFF

000040005FFF4000

5FFF40001FFF1FFF

5FFF400000005FFF

40005FFF00000000

5FFF40005FFF0000

00005FFF40005FFF

00005FFF40005FFF

link_out[2]:

0x0000000043FF5FFF

00005FFF5FFF5FFF

400000005FFF5FFF

5FFF400000005FFF

5FFF40001FFF1FFF

5FFF40001FFF1FFF

5FFF40001FFF1FFF

5FFF40001FFF1FFF

5FFF40001FFF1FFF

5FFF40001FFF1FFF

5FFF40001FFF1FFF

link_out[3]:

0x0000000000005FFF

40001FFF40000000

400040001FFF4000

0000400040001FFF

4000000040000000

000000005FFF4000

4000400000005FFF

4000400000000000

5FFF400040000000

5FFF400040000000

5FFF400040000000

RTL - SIMULATION OUTPUT (11 22 33 44)

0600190007D00170

0600190007D00170

00005FFF40005FFF

00005FFF40005FFF

1FFF1FFF5FFF1FFF

1FFF1FFF5FFF1FFF

5FFF400040004000

5FFF400040004000

5FFF400040004000

5FFF400040004000

5FFF400040004000

5FFF400040004000

060019000E100170

060019000E100170

5FFF40005FFF0000

5FFF40005FFF0000

1FFF5FFF1FFF4000

1FFF5FFF1FFF4000

4000400040000000

4000400040000000

4000400040000000

4000400040000000

4000400040000000

4000400040000000

0600190014500170

0600190014500170

40005FFF00000000

40005FFF00000000

5FFF1FFF40001FFF

5FFF1FFF40001FFF

4000400000005FFF

4000400000005FFF

4000400000005FFF

4000400000005FFF

4000400000005FFF

4000400000005FFF

03C01A0012E80000

03C01A0012E80000

5FFF000000005FFF

5FFF000000005FFF

5FFF40001FFF1FFF

5FFF40001FFF1FFF

000000005FFF4000

000000005FFF4000

000000005FFF4000

000000005FFF4000

000000005FFF4000

000000005FFF4000

03C01A0006680000

03C01A0006680000

000040005FFF4000

000040005FFF4000

5FFF5FFF40004000

5FFF5FFF40004000

4000000040000000

4000000040000000

4000000040000000

4000000040000000

4000000040000000

4000000040000000

03C01A000CA80000

03C01A000CA80000

40005FFF5FFF5FFF

40005FFF5FFF5FFF

5FFF400000005FFF

5FFF400000005FFF

0000400040001FFF

0000400040001FFF

0000400040001FFF

0000400040001FFF

0000400040001FFF

0000400040001FFF

06811E4066E0002

06811E4066E0002

5FFF5FFF5FFF0000

5FFF5FFF5FFF0000

400000005FFF5FFF

400000005FFF5FFF

400040001FFF4000

400040001FFF4000

400040001FFF4000

400040001FFF4000

400040001FFF4000

400040001FFF4000

06811E40CAE0002

06811E40CAE0002

5FFF5FFF00004000

5FFF5FFF00004000

00005FFF5FFF5FFF

00005FFF5FFF5FFF

40001FFF40000000

40001FFF40000000

40001FFF40000000

40001FFF40000000

40001FFF40000000

40001FFF40000000

06811E412EE0002

06811E412EE0002

0000000040001FFF

0000000040001FFF

0000000043FF5FFF

0000000043FF5FFF

0000000000005FFF

0000000000005FFF

0000000000005FFF

0000000000005FFF

0000000000005FFF

0000000000005FFF

Bitfile test OUTPUT (11 22 33 44)

0x0600190007D00170

0x0600190007D00170

0x00005FFF40005FFF

0x00005FFF40005FFF

0x1FFF1FFF5FFF1FFF

0x1FFF1FFF5FFF1FFF

0x5FFF400040004000

0x5FFF400040004000

0x5FFF400040004000

0x5FFF400040004000

0x5FFF400040004000

0x5FFF400040004000

0x060019000E100170

0x060019000E100170

0x5FFF40005FFF0000

0x5FFF40005FFF0000

0x1FFF5FFF1FFF4000

0x1FFF5FFF1FFF4000

0x4000400040000000

0x4000400040000000

0x4000400040000000

0x4000400040000000

0x4000400040000000

0x4000400040000000

0x0600190014500170

0x0600190014500170

0x40005FFF00000000

0x40005FFF00000000

0x5FFF1FFF40001FFF

0x5FFF1FFF40001FFF

0x4000400000005FFF

0x4000400000005FFF

0x4000400000005FFF

0x4000400000005FFF

0x4000400000005FFF

0x4000400000005FFF

0x03C01A0012E80000

0x03C01A0012E80000

0x5FFF000000005FFF

0x5FFF000000005FFF

0x5FFF40001FFF1FFF

0x5FFF40001FFF1FFF

0x000000005FFF4000

0x000000005FFF4000

0x000000005FFF4000

0x000000005FFF4000

0x000000005FFF4000

0x000000005FFF4000

0x03C01A0006680000

0x03C01A0006680000

0x000040005FFF4000

0x000040005FFF4000

0x5FFF5FFF40004000

0x5FFF5FFF40004000

0x4000000040000000

0x4000000040000000

0x4000000040000000

0x4000000040000000

0x4000000040000000

0x4000000040000000

0x03C01A000CA80000

0x03C01A000CA80000

0x40005FFF5FFF5FFF

0x40005FFF5FFF5FFF

0x5FFF400000005FFF

0x5FFF400000005FFF

0x0000400040001FFF

0x0000400040001FFF

0x0000400040001FFF

0x0000400040001FFF

0x0000400040001FFF

0x0000400040001FFF

0x06811E4066E0002

0x06811E4066E0002

0x5FFF5FFF5FFF0000

0x5FFF5FFF5FFF0000

0x400000005FFF5FFF

0x400000005FFF5FFF

0x400040001FFF4000

0x400040001FFF4000

0x400040001FFF4000

0x400040001FFF4000

0x400040001FFF4000

0x400040001FFF4000

0x06811E40CAE0002

0x06811E40CAE0002

0x5FFF5FFF00004000

0x5FFF5FFF00004000

0x00005FFF5FFF5FFF

0x00005FFF5FFF5FFF

0x40001FFF40000000

0x40001FFF40000000

0x40001FFF40000000

0x40001FFF40000000

0x40001FFF40000000

0x40001FFF40000000

0x06811E412EE0002

0x06811E412EE0002

0x0000000040001FFF

0x0000000040001FFF

0x0000000043FF5FFF

0x0000000043FF5FFF

0x0000000000005FFF

0x0000000000005FFF

0x0000000000005FFF

0x0000000000005FFF

0x0000000000005FFF

0x0000000000005FFF

FIGURE 4.1: Illustration of output matching of hex words across HLS, RTL and Bitfile levels for 1bx data for RCT.

4.3 Virtual Multi-Board System Test using FEAST

The validation of the complex L1 trigger system with more than 50 FPGAs across multiple subsystems presents significant challenges, particularly in terms of hardware availability and system-level integration testing. The FPGA Environment for Algorithm Slice Tests (FEAST) has been developed [12] to address these challenges by providing a versatile environment for system validation, and virtualized testing on real hardware. FEAST enables the emulation of large-scale, multi-board installations using a minimal set of physical hardware, even a single FPGA board.

FEAST operates by allowing users to define a target system architecture and test conditions through configuration files. This definition includes the number and type of FPGA boards, the specific firmware (bitfiles) to be run on each virtualized board, and the logical interconnections (links) between them. The core capability of FEAST is its ability to sequentially evaluate the behavior of each FPGA position within the defined multi-board architecture, even if only one physical FPGA is available for testing.

The workflow is as follows:

1. **System Definition:** The user specifies the complete multi-layer system architecture, including the number of FPGAs, their roles (e.g., RCT, GCT Barrel), the firmware bitfiles designated for each, and the logical data links connecting them. This definition essentially creates a virtualized installation of the real system.
2. **Sequential Evaluation:** The FEAST engine takes this system definition and, if running on limited hardware (e.g., a single APx board), sequentially

configures the physical FPGA to emulate each board in the defined architecture.

- For the first board in the virtual system, its corresponding bitfile is loaded onto the physical FPGA. Input test vectors are supplied (from pre-defined files).
- The physical FPGA processes these inputs, and its outputs (which would normally go to other FPGAs via physical links) are captured by the FEAST framework.
- For the next board in the virtual system, its bitfile is loaded. The captured outputs from the previously emulated board(s) that serve as inputs to this current board are then fed into the physical FPGA.
- This process repeats, with FEAST managing the data flow between these sequential emulations, effectively mimicking the data propagation through the physical links of the full multi-board system.

3. **Mixed Physical and Virtual Links:** FEAST also supports test systems with a combination of physical and virtual links. If multiple physical boards are available, they can be interconnected as per the defined architecture, while FEAST manages any remaining virtual links or emulates parts of the system not physically present.
4. **Link Alignment and MGT Allocation:** The environment is designed to handle complexities such as link alignment, ensuring that data passed between virtually connected FPGAs (or between physical and virtual segments) maintains correct timing relationships. It also respects the target MGT channel allocations specified for the firmware bitfiles, ensuring that tests are representative of the final hardware deployment.

5. **Extensibility:** While initially designed with APx boards in mind, FEAST is architected to be extensible, allowing for its potential use in mixed-platform environments involving different types of FPGA boards.

As illustrated in Figure 4.2, FEAST provides a powerful abstraction layer. Users can construct and test full-scale, multi-layer systems (like a complete slice of the calorimeter trigger chain from RCTs through GCT) on actual FPGAs, irrespective of their physical location or the number of available boards, by leveraging the FEAST engine’s sequential evaluation and data management capabilities. This significantly accelerates firmware development, integration testing, and debugging by allowing comprehensive system-level validation much earlier in the design cycle and with reduced hardware dependency.

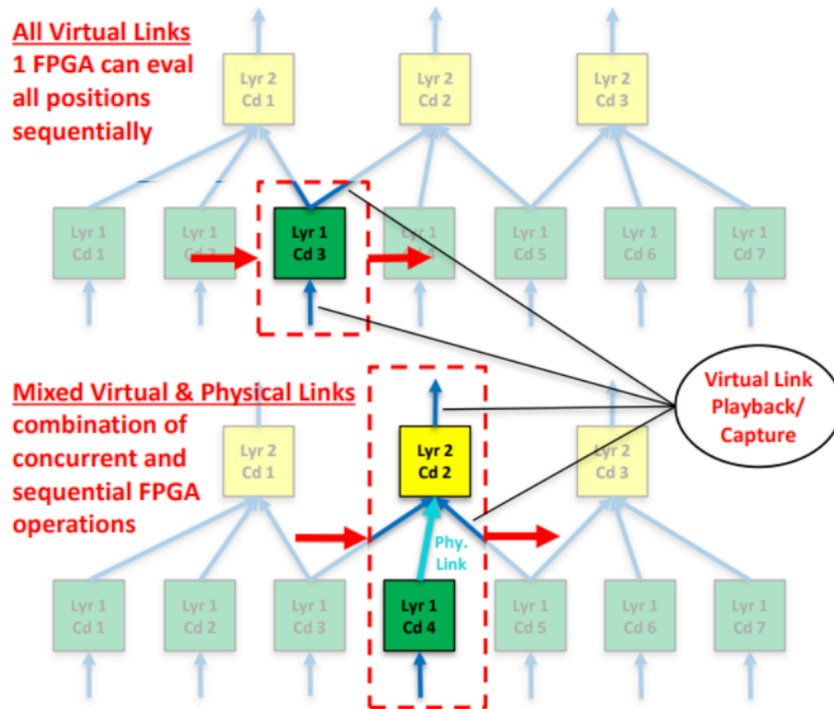


FIGURE 4.2: Conceptual illustration of the FEAST environment, showcasing its ability to emulate a multi-FPGA system using as little as a single physical FPGA. Users define the system configuration, and FEAST sequentially evaluates each FPGA’s role, managing data flow between virtualized boards [12].

4.3.1 RCT + GCT System Tests using FEAST

To validate the integrated functionality of the Regional Calorimeter Trigger (RCT) and Global Calorimeter Trigger (GCT) Barrel systems, comprehensive tests were conducted using the FEAST environment. These tests simulated the data flow from multiple RCT boards into the GCT Barrel boards, as depicted in the target architectures.

The testing methodology involved preparing input test vector files for each virtual RCT board in the defined system. These test vectors represent the ECAL and HCAL data that an RCT board would receive from the detector back-end. Within the FEAST configuration:

- Each virtual RCT board is "loaded" with its firmware and processes its assigned input test vector file.
- The output data links from these virtual RCT boards are then logically mapped as inputs to the appropriate virtual GCT Barrel board(s) according to the system architecture.
- The GCT Barrel board(s) process this aggregated RCT data, and their final outputs (to the Correlator Layer or GCT Summation unit) are captured for verification.

Two main scales of system tests were performed:

1. **Small-Scale Test (8 RCTs to 1 GCT):** An initial test focused on verifying the processing chain for a single GCT Barrel board. This configuration involved simulating eight RCT cards whose outputs fed into one GCT Barrel card, as illustrated conceptually in Figure 4.3. This test was executed using

FEAST in an "all virtual links" configuration, utilizing a single physical APx board to sequentially emulate all nine FPGA positions (8 RCTs + 1 GCT). The captured outputs from the virtual GCT Barrel board were successfully validated against expected results, confirming the correct data handling and algorithmic processing through this segment of the trigger chain.

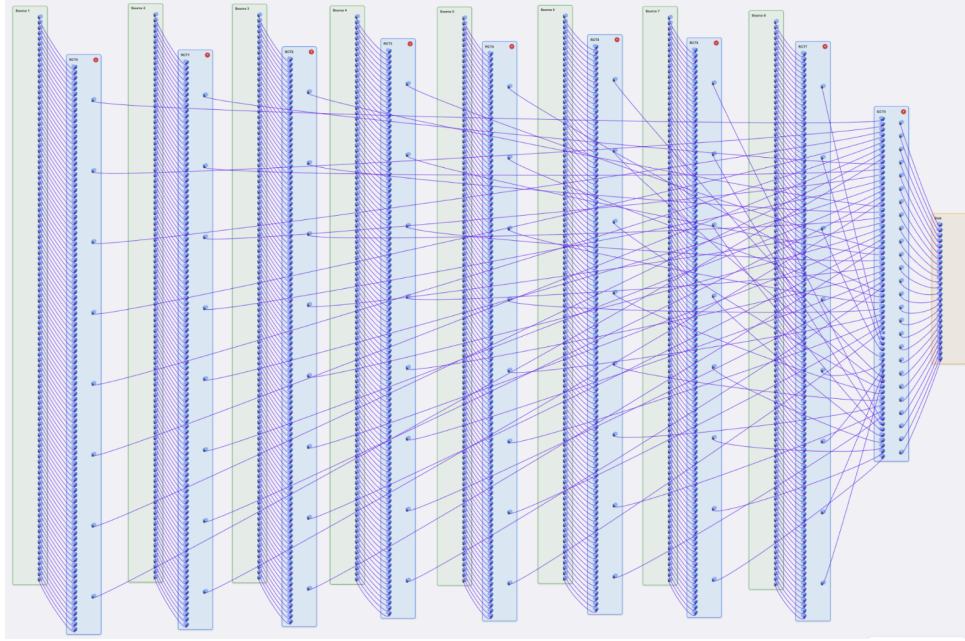


FIGURE 4.3: CHEF visualization of the small-scale FEAST test: 8 virtual RCT cards providing input to 1 virtual GCT Barrel card [16].

2. **Full-Scale Barrel Test (24 RCTs to 3 GCTs):** Following the success of the smaller test, a more comprehensive test emulating the full barrel calorimeter trigger path was conducted. This involved emulating all 24 RCT cards providing input to 3 GCT Barrel cards, representing the complete barrel coverage (Figure 4.4). This larger test was also run with all inter-FPGA links configured as virtual. To expedite the significantly longer runtime associated with sequentially emulating 27 FPGA positions, a pool of two physical APx boards was utilized by FEAST, allowing for some parallelization of the sequential evaluations. This full-scale system test was also successful, demonstrating

the scalability of the firmware and the capability of FEAST to manage and validate such large, interconnected systems. The results verified the correct data aggregation, object stitching across GCT board boundaries, and overall data integrity for the complete barrel trigger system.

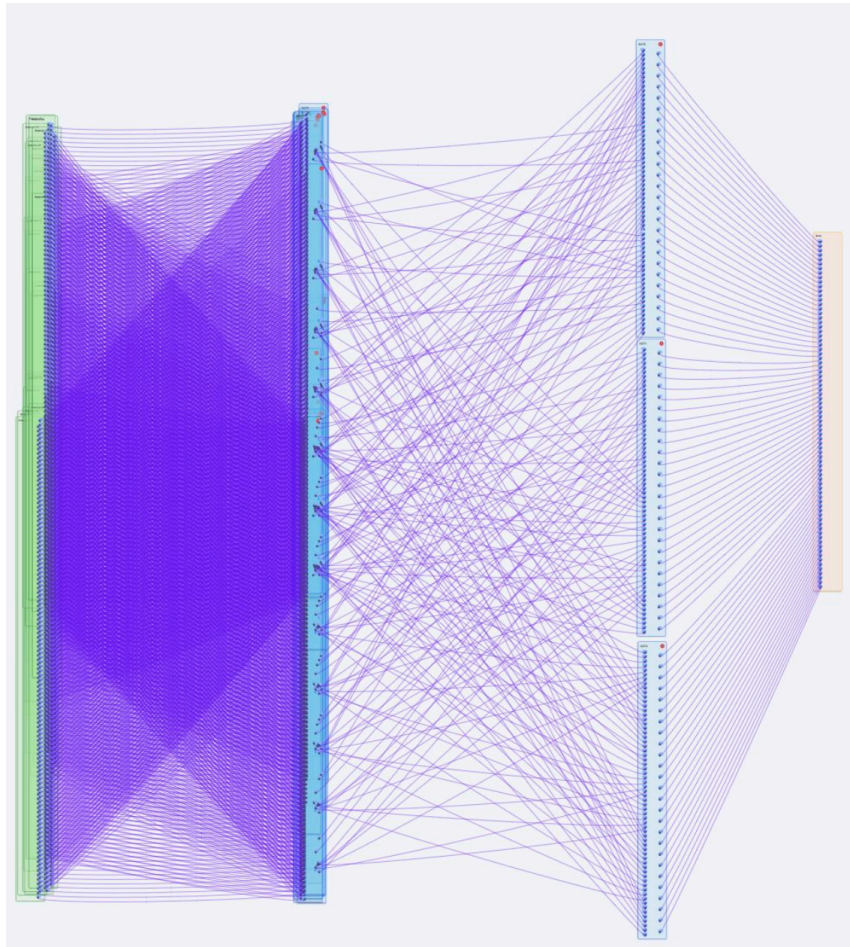


FIGURE 4.4: CHEF visualization of the full-scale FEAST test: 24 virtual RCT cards providing input to 3 virtual GCT Barrel cards, emulating the entire barrel trigger system [16].

These FEAST-based system tests were instrumental in verifying the inter-board communication protocols, data formats, and the algorithmic correctness of the combined RCT and GCT Barrel firmware before full hardware integration test was possible.

4.4 Testing Summary

In summary, the outputs from HLS simulations, RTL simulations, and standalone hardware (bitfile) tests were matched for both the RCT and GCT Barrel firmware individually. The combined RCT + GCT Barrel system tests conducted using the FEAST framework also demonstrated correct integrated behavior. This multi-layered validation approach provided high confidence in the functional correctness, timing performance, and hardware implementability of the developed calorimeter trigger firmware.

Chapter 5

Conclusion and Future Work

5.1 Summary of Contributions

The work presented in this thesis encompasses the design, implementation, validation, and optimization of the Regional Calorimeter Trigger (RCT) and the GCT Barrel firmware components for the CMS Phase-2 Level-1 Calorimeter Trigger system. The contributions span key areas, from algorithmic architecture and high-level synthesis to FPGA implementation, testing infrastructure development, and workflow optimization.

A main contribution was the development and refinement of the RCT 17×6 firmware architecture. This involved designing and implementing HLS IP cores for ECAL crystal processing, cluster finding, shower shape calculation, energy calibration, and HCAL data integration. To satisfy the floorplan constraints of the DAQ subsystem, the RCT firmware was re-architected through the optimization of IP core resources and the relocation of functional blocks across SLRs. A mechanism for real-time threshold calibration was also developed, which enables operational flexibility by

allowing e/γ identification parameters to be updated via software, thus eliminating the need for FPGA reprogramming.

For the GCT Barrel, the work included the design and implementation of IP cores for processing aggregated RCT data, performing inter-RCT object stitching, generating Particle Flow clusters, SuperTowers, jets, and taus, and calculating global energy sums. A Time Multiplexing (TMUX) scheme was developed, evolving from an initial TMUX6 to a final TMUX18 architecture. The TMUX18 design, implemented with hierarchical TMUX9 instances and delay lines, meets the requirements for the Correlator Layer 1 while optimizing interconnect efficiency.

Throughout the development of both RCT and GCT Barrel firmware, a strong emphasis was placed on achieving reliable timing closure at 360 MHz and maintaining resource utilization within specified limits. This was supported by contributions to the Vivado implementation flow, including the systematic application of iterative physical optimization looping, the use of incremental synthesis and implementation to improve iteration times and design stability, and the adoption of Out-of-Context (OOC) synthesis for HLS IP cores to improve reproducibility. These methodological improvements were instrumental in achieving stable and reproducible builds that met performance targets.

Contributions were also made to the development of the testing and validation infrastructure. This includes the utilization and support of the FEAST environment for virtualized multi-board system testing, which enables comprehensive system-level validation with limited hardware. To streamline the use of FEAST and other aspects of the firmware development workflow, several GUI-based helper tools were developed: CHEF, for automating the creation of complex FEAST configuration files; an MGT mapping utility for visual configuration of high-speed serial transceivers; a visual floorplanning constraint generator to simplify pblock definition; and a test

vector generation and formatting tool to automate the preparation of test vectors for HLS, RTL, and hardware testing. These tools collectively reduce manual effort, minimize errors, and accelerate the design and verification cycle.

In summary, this work has delivered functional, optimized, and validated firmware designs for the RCT and GCT barrel components of the CMS L1 Calorimeter Trigger. It has also contributed to the underlying methodologies and toolsets that support efficient FPGA firmware development. All designs closed timing reliably at 360MHz and the latency of RCT + GCT processing was $1.01\ \mu s$ which is within the total L1 trigger latency budget of $12.5\ \mu s$. The resulting firmware and infrastructure provide a good foundation for the further development of the Phase-2 trigger system.

5.2 Future Developments

The current designs for the Regional Calorimeter Trigger (RCT) and Global Calorimeter Trigger (GCT) Barrel provide a robust foundation for the Phase-2 trigger upgrade. However, several avenues for future development and enhancement are envisioned to further optimize performance, resource utilization, and maintainability, as well as to expand testing capabilities.

A first step in the near term will be the integration of the latest version of the APx firmware shell for both the RCT and GCT Barrel projects. This upgrade will ensure compatibility with the evolving board support package and provide access to the newest features and improvements in the underlying firmware infrastructure. Following this, the implementation of the Data Acquisition (DAQ) subsystem firmware

will be undertaken. This involves integrating the DAQ logic into the reserved floorplan regions on the FPGAs and establishing the necessary interfaces with the trigger algorithm payload.

A crucial ongoing effort will focus on reducing the Configurable Logic Block (CLB) utilization. While current designs meet the nominal target of less than 70% per Super Logic Region (SLR), further optimization to bring CLB usage consistently below 60% per SLR is desirable. This would provide greater margin for routing, accommodate potential late-stage design modifications more easily, and improve timing performance by reducing congestion. This will involve re-evaluating HLS directives, exploring alternative algorithmic implementations, and potentially a full overhaul of algorithm logic and subsystem architecture.

Another significant development will be the transition of the toolchain from the currently used Vivado HLS 2020.1 and Vivado 2021.x versions to the latest Vitis HLS and Vivado releases. This migration is essential for long-term support, access to new FPGA device features, and potential improvements in synthesis and implementation algorithms offered by newer tool versions. Such a transition will most likely require an extensive effort, involving refactoring, rewriting, or in some cases, completely overhauling portions of the HLS C++ code to leverage new tool capabilities effectively and produce better results.

The current algorithmic implementations for both RCT and GCT make minimal to no use of the dedicated Digital Signal Processing (DSP) blocks available in the VU13P FPGA. The VU13P-2 device offers over 12,000 DSP slices, which are currently an underutilized resource. Future work will explore opportunities to offload computationally intensive operations, such as multiplications, accumulations, or filtering tasks inherent in the calorimeter algorithms, to these DSP blocks. Judicious use of DSPs could significantly reduce CLB utilization.

In terms of testing and validation, the scope and scale of tests will be expanded. Current testing procedures will be augmented to include much larger datasets, specifically targeting tests involving 1000 or more bunch crossings (BX). These extensive, multi-BX tests are critical for uncovering subtle bugs related to data buffering, state machine behavior over long periods, and the handling of high-pileup conditions. Such large-scale tests will be conducted at multiple levels: for standalone bitfile validation on individual boards and within the integrated FEAST environment to simulate complex multi-board system behavior over extended operational periods. This comprehensive testing strategy will ensure the reliability of the trigger system under realistic experimental conditions.

Appendix A

Automation Tools

The development, implementation, and validation of the firmware for the Regional and Global Calorimeter Triggers involves a complex workflow often requiring non-insignificant amounts of manual attention in generating configuration files, preparing test vectors etc.. The automation tools discussed below were developed to streamline various stages of the design process, from configuration and constraint generation to system-level testing and test vector preparation, to enhance efficiency and reduce the likelihood of manual errors. [\[20\]](#)

A.1 CHEF: Configuration Helper for Easy FEAST-ing

As detailed in [Section 4.3](#), the FEAST environment facilitates virtualized multi-board system testing, which relies on a configuration file. This file encapsulates critical information such as Multi-Gigabit Transceiver (MGT) mappings, inter-board

dependencies, and the detailed connection topology of the system architecture under test. While manually creating this configuration file is feasible for smaller test scenarios involving a few boards and limited interconnectivity, the complexity escalates significantly for large-scale system tests. Such tests may involve numerous FPGAs and thousands of individual connections, making manual mapping not only a laborious task but also highly susceptible to human error. To address this challenge

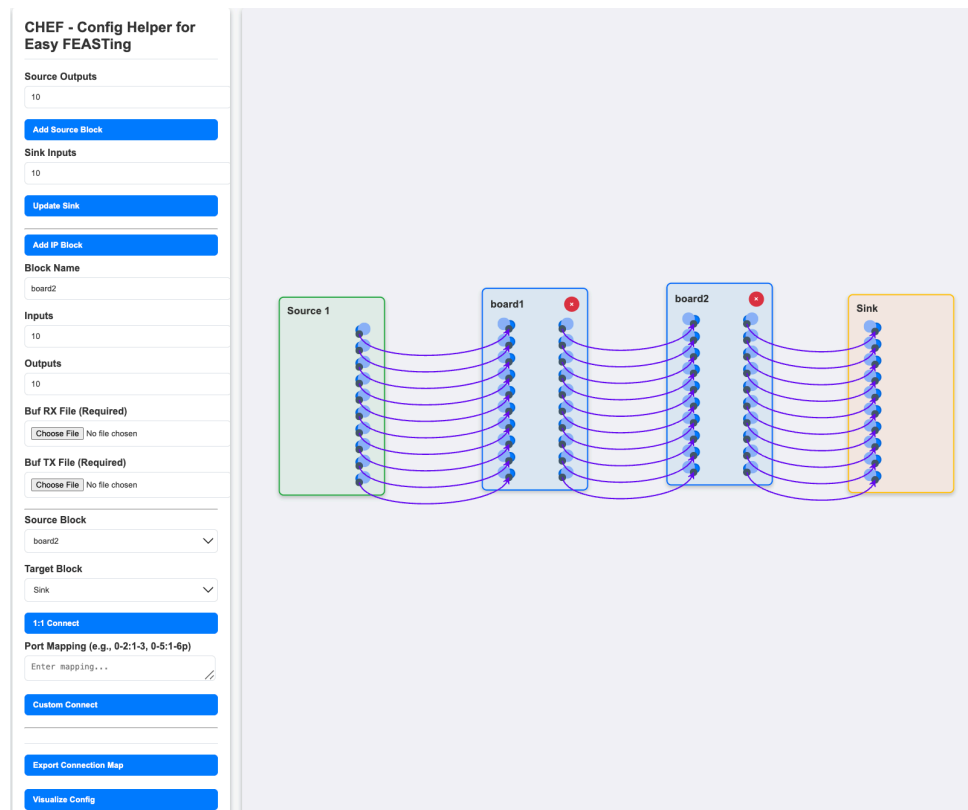


FIGURE A.1: CHEF for FEAST [20]

and automate the creation of FEAST configuration files, a dedicated tool named CHEF (Config Helper for Easy FEASTing) was developed. CHEF is a browser-based Graphical User Interface (GUI) tool designed to simplify and expedite the process of defining complex system architectures for FEAST. It offers intuitive drag-and-drop functionality, allowing users to visually construct their test setup by placing and connecting virtual boards. CHEF then automates the intricate task of connection

mapping, translating the visual representation into the precise syntax required by the FEAST configuration file. Furthermore, it provides visual verification of the established connection maps, enabling users to readily inspect and confirm the system topology before initiating tests. This automation significantly reduces the setup time for large-scale FEAST simulations and minimizes the risk of errors associated with manual configuration.

A.2 MGT Mapping Utility

The successful compilation and implementation of FPGA projects, particularly those involving high-speed serial links, require an accurate MGT map. This map, typically provided as a constraint file, details which MGT quads or individual channels are active, their operational mode (transmitter, receiver, or both), and their assignment to specific logical links in the design. Traditionally, creating or modifying this MGT map involves consulting FPGA schematics or pinout diagrams to identify the physical locations of MGTs and then manually editing the constraint file. This process can be tedious and prone to errors, especially for FPGAs with a large number of MGTs, such as the VU13P. To streamline MGT configuration, a browser-based GUI tool was developed. This utility provides a visual representation of the MGT layout on the target FPGA. Users can interactively select MGTs, designate them as active or inactive, and configure their operational status (TX, RX, or TX/RX). Once the desired MGT configuration is visually established, the tool automatically generates the corresponding MGT map in the required file format. This visual approach accelerates the MGT mapping process and significantly reduces the potential for manual errors in specifying MGT constraints, leading to more reliable and efficient design implementation.

MGT #	I/O INDEX	MGT FW LINK #	REFCLK FW #	QPLL FW #	DIRECTION	DIRECTION	QPLL FW #	REFCLK FW #	MGT FW LINK #	I/O INDEX	MGT #	SLR
X1Y03		59			TX	TX			123		X1Y03	
X1Y02		58			TX	TX			122		X1Y02	
X1Y01		57			TX	TX			121		X1Y01	
X1Y00		56			TX	TX			120		X1Y00	
X1Y59		55			TX	TX			119		X1Y59	
X1Y58		54			TX	TX			118		X1Y58	
X1Y57		53			TX	TX			117		X1Y57	
X1Y56		52			TX	TX			116		X1Y56	
X1Y55		51			TX	TX			115		X1Y55	SLR3
X1Y54		50			TX	TX			114		X1Y54	
X1Y53		49			TX	TX			113		X1Y53	
X1Y52		48			TX	TX			112		X1Y52	
X1Y51		47			TX	TX			111		X1Y51	
X1Y50		46			TX	TX			110		X1Y50	
X1Y49		45			TX	TX			109		X1Y49	
X1Y48		44			TX	TX			108		X1Y48	
X1Y47		43			TX	TX			107		X1Y47	
X1Y46		42			TX	TX			106		X1Y46	

FIGURE A.2: MGT Configuration Utility [20]

A.3 Visual Floorplanning Constraint Generator

Floorplanning, the process of assigning specific physical regions on the FPGA die for the placement of different design modules or IP cores, is crucial for achieving timing closure and managing resource distribution in complex designs. Effective floorplanning requires a detailed understanding of the FPGA's internal architecture, particularly the layout of clock regions, SLRs, and other physical resources. Manually crafting floorplan constraints (e.g., pblock definitions) can be complex and error-prone, as it involves specifying precise coordinates and resource ranges. To simplify this task, a visual GUI-based browser tool was created specifically for generating floorplan constraints for the VU13P-2 FPGA. This tool presents an interactive graphical layout of the FPGA, allowing users to visually define placement regions for their design modules. By selecting areas on the FPGA map, users can intuitively create pblocks. The tool then automatically translates these visual selections into the correct syntax for floorplan constraints, which can be directly copied and pasted into the project's constraint file. This visual aid demystifies the floorplanning process, reduces the likelihood of errors in constraint definition, and allows designers to more effectively guide the FPGA place-and-route tools.

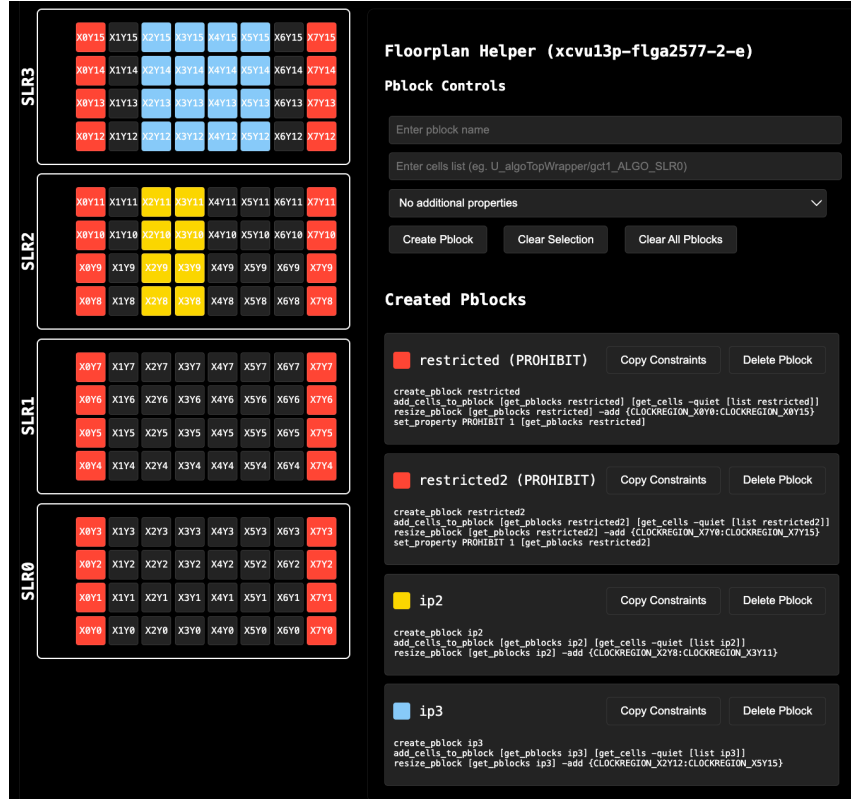


FIGURE A.3: Floorplanning Tool for VU13P-2 [20]

A.4 Test Vector Generation and Formatting Tool

Thorough verification of FPGA firmware requires appropriately formatted test vectors for various simulation and hardware testing stages. For the C++ HLS models, input and output data are typically represented as hexadecimal values corresponding to the link widths. However, for RTL simulation testbenches, these wide links (e.g., 576-bit links) often need to be padded to a specific width and then segmented into smaller, manageable words (e.g., 64-bit hexadecimal words) for each clock cycle or bunch crossing (BX). Furthermore, test vectors intended for on-hardware bit-file testing, which are loaded into FPGA receive buffers, must adhere to a specific format dictated by the test infrastructure. Manually performing these formatting operations for numerous test vectors and multiple bunch crossings is not only time-consuming but also a significant source of potential inconsistencies and errors. To

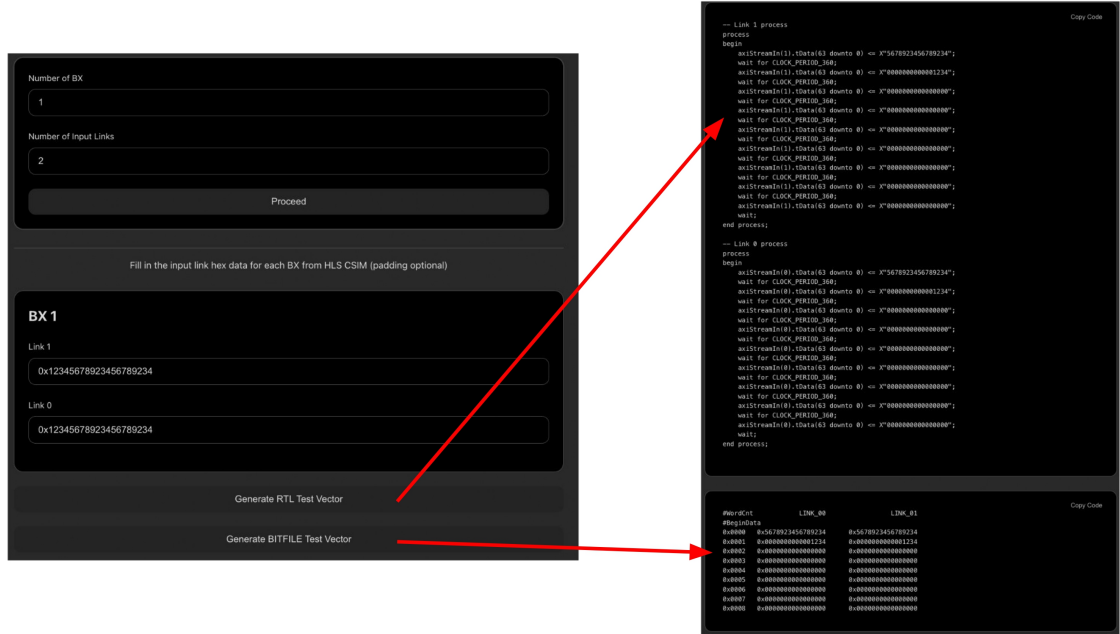


FIGURE A.4: Test vector generation tool [20]

automate this process and ensure consistency, a dedicated test vector generation tool was developed. This tool takes the raw hexadecimal data from C++ HLS simulation link outputs as input. It then automatically performs the necessary padding, segmentation, and formatting to generate:

Multi-bunch crossing test vectors suitable for RTL simulation testbenches, with data correctly arranged into the required word sizes and temporal sequence. Test vectors formatted for ingestion by the hardware test setup's receive buffers, ensuring compatibility with the bitfile testing environment.

By automating these formatting tasks, the tool significantly speeds up the test vector preparation process, eliminates manual formatting errors, and ensures that consistent test cases are applied across all stages of the verification flow.

References

- [1] L. Evans and P. Bryant, “LHC Machine,” *J. Instrum.*, vol. 3, no. 08, p. S08001, 2008, doi: 10.1088/1748-0221/3/08/S08001.
- [2] CMS Collaboration, “The CMS experiment at the CERN LHC,” *J. Instrum.*, vol. 3, no. 08, p. S08004, 2008, doi: 10.1088/1748-0221/3/08/S08004.
- [3] T. Sakuma and T. McCauley, “Detector and Event Visualization with SketchUp at the CMS Experiment,” *J. Phys.: Conf. Ser.*, vol. 513, no. 2, p. 022032, 2014, doi: 10.1088/1742-6596/513/2/022032.
- [4] CMS Collaboration, “Description and performance of track and primary-vertex reconstruction with the CMS tracker,” *J. Instrum.*, vol. 9, no. 10, p. P10009, 2014, doi: 10.1088/1748-0221/9/10/P10009.
- [5] CMS Collaboration, “Performance and operation of the CMS electromagnetic calorimeter,” *J. Instrum.*, vol. 5, no. 03, p. T03010, 2010, doi: 10.1088/1748-0221/5/03/T03010.
- [6] B. Acar *et al.*, “Response of a CMS HGCal silicon-pad electromagnetic calorimeter prototype to 20300 GeV positrons,” 2022, *arXiv:2111.06855*.

-
- [7] CMS Collaboration, “Performance of the CMS hadron calorimeter with cosmic ray muons and LHC beam data,” *J. Instrum.*, vol. 5, no. 03, p. T03012, 2010, doi: 10.1088/1748-0221/5/03/T03012.
 - [8] A. M. Sirunyan *et al.* (CMS Collaboration), “Performance of the CMS muon detector and muon reconstruction with proton-proton collisions at $\sqrt{s} = 13$ TeV,” *J. Instrum.*, vol. 13, no. 06, p. P06015, 2018, doi: 10.1088/1748-0221/13/06/P06015.
 - [9] P. Kumar and B. Gomber, “The CMS Level-1 Calorimeter Trigger for the HL-LHC,” in *EPJ Web of Conf.*, vol. 295, 2024, p. 02022, doi: 10.1051/epjconf/202429502022.
 - [10] P. Kumar and B. Gomber, “System Design and Prototyping of the CMS Level-1 Calorimeter Trigger at the High-Luminosity LHC,” *IEEE Trans. Nucl. Sci.*, vol. 72, no. 3, pp. 385391, 2025, doi: 10.1109/TNS.2024.3414999.
 - [11] C. Savard on behalf of the CMS Collaboration, “CHEP 2023 Overview of the HL-LHC Upgrade for the CMS Level-1 Trigger,” in *EPJ Web of Conf.*, vol. 295, 2024, p. 02022, doi: 10.1051/epjconf/202429502022.
 - [12] I. Ojalvo, “The APx Board for the CMS Phase 2 L1 Calorimeter trigger: Testing and Performance,” presented at TWEPP, 2024.
 - [13] P. Kumar, “System Design and Prototyping of the CMS Level-1 Calorimeter Trigger,” Ph.D. dissertation, Univ. of Hyderabad, Hyderabad, India, 2024.
 - [14] S. Piperov, “Geant4 validation with CMS calorimeters test-beam data,” 2008, *arXiv:0808.0130*. [Online]. Available: <https://arxiv.org/abs/0808.0130>

-
- [15] H. Abramowicz *et al.*, “Measurement of shower development and its Molière radius with a four-plane LumiCal test set-up,” *Eur. Phys. J. C*, vol. 78, no. 2, p. 135, 2018, doi: 10.1140/epjc/s10052-018-5600-4.
- [16] A. Mohan, “Phase 2 L1Calo Trigger Update 2025,” presented at the CMS Level-1 Trigger Workshop, Oviedo, Apr. 1-4, 2025. Available : <https://indico.cern.ch/event/1497887/contributions/6386283/>
- [17] Xilinx, “Vivado Design Suite User Guide,” UG1399 Port-Level Protocols for Vivado IP Flow, Available: <https://docs.amd.com/r/en-US/ug1399-vitis-hls/Port-Level-Protocols-for-Vivado-IP-Flow>.
- [18] Xilinx, “Vivado Design Suite User Guide,” UG835 Vivado Design Suite Tcl Command Reference Guide - phys_opt_design, Available : https://docs.amd.com/r/en-US/ug835-vivado-tcl-commands/phys_opt_design
- [19] Xilinx, “Vivado Design Suite User Guide,” UG896 Vivado Design Suite User Guide: Designing with IP, Available: <https://docs.amd.com/r/en-US/ug896-vivado-ip/Out-of-Context-Flow>
- [20] A. Mohan, "FPGA Automation Toolkit," 2022. [Online]. Available: <https://abhinavm2000.github.io/fpga-automation-toolkit/>.
- [21] G. Daughtry, "Top 5 Timing Closure Techniques," presented at the Club Vivado Users Group, Paris, France, 2015. [Online]. Available: https://www.xilinx.com/publications/prod_mktg/club_vivado/presentation-2015/paris/Xilinx-TimingClosure.pdf

List of Publications

[16] Presentation in CMS Level 1 Trigger Workshop 2025, Oviedo

A. Mohan and Phase2 L1Calo group, Phase 2 L1Calo Trigger Update, presented at the L1 Trigger Workshop, Oviedo, Apr. 1 – 4, 2025.

Available: <https://indico.cern.ch/event/1497887/contributions/6386283/>

Design and FPGA Implementation of the Barrel Calorimeter Trigger Algorithms for the High Luminosity LHC

by Abhinav Mohan

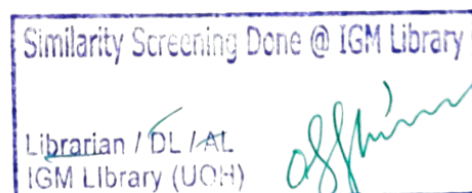
Submission date: 24-Jun-2025 11:34AM (UTC+0530)

Submission ID: 2705178581

File name: Abhinav_Mohan.pdf (20.75M)

Word count: 15824

Character count: 86870



Design and FPGA Implementation of the Barrel Calorimeter Trigger Algorithms for the High Luminosity LHC

ORIGINALITY REPORT

5%

SIMILARITY INDEX

3%

INTERNET SOURCES

2%

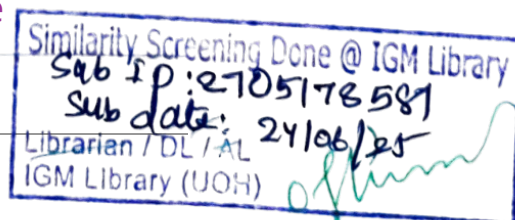
PUBLICATIONS

3%

STUDENT PAPERS

PRIMARY SOURCES

- | | | |
|---|---|-----|
| 1 | Submitted to University of Edinburgh
Student Paper | 1% |
| 2 | Submitted to The University of Manchester
Student Paper | <1% |
| 3 | publish.etp.kit.edu
Internet Source | <1% |
| 4 | Submitted to City University of Hong Kong
Student Paper | <1% |
| 5 | Daniel Salerno. "The Higgs Boson Produced With Top Quarks in Fully Hadronic Signatures", Springer Science and Business Media LLC, 2019
Publication | <1% |
| 6 | Michail Bachtis. "Heavy Neutral Particle Decays to Tau Pairs", Springer Science and Business Media LLC, 2014
Publication | <1% |
| 7 | Submitted to Vietnamese-German University
Student Paper | <1% |
| 8 | repositorio.unican.es
Internet Source | <1% |
| 9 | Sarin, P. "Heavy ion physics from the CMS collaboration", Journal of Physics Conference Series, 2013.
Publication | <1% |



10	Oliver Pooth. "Introduction", The CMS Silicon Strip Tracker, 2010 Publication	<1 %
11	Cécile Caillol. "Scalar Boson Decays to Tau Leptons", Springer Science and Business Media LLC, 2018 Publication	<1 %
12	www.escholar.manchester.ac.uk Internet Source	<1 %
13	Submitted to Brunel University Student Paper	<1 %
14	www.imperial.ac.uk Internet Source	<1 %
15	P Klabbers, M Bachtis, J Brooke, M Cepeda Hermida et al. "CMS level-1 upgrade calorimeter trigger prototype development", Journal of Instrumentation, 2013 Publication	<1 %
16	digitalcommons.unl.edu Internet Source	<1 %
17	Submitted to Kyungpook National University Student Paper	<1 %
18	conservancy.umn.edu Internet Source	<1 %
19	indico.cern.ch Internet Source	<1 %
20	Submitted to Chonnam National University Student Paper	<1 %
21	physphd.unideb.hu Internet Source	<1 %
22	iopscience.iop.org Internet Source	<1 %

Dr. Bhawna G.
Assistant Professor
CASEST, School of Physics
University of Hyderabad
Hyderabad-500046, Telangana

23

web.physik.rwth-aachen.de

Internet Source

<1 %

24

5dok.org

Internet Source

<1 %

Exclude quotes On

Exclude matches < 14 words

Exclude bibliography On



Dr. Bhawna G.
Assistant Professor
CASEST, School of Physics
University of Hyderabad
Hyderabad-500046, Telangana